

CONFERENCES IN RESEARCH AND PRACTICE IN
INFORMATION TECHNOLOGY

VOLUME 60

INFORMATION VISUALIZATION 2006



AUSTRALIAN
COMPUTER
SOCIETY

INFORMATION VISUALIZATION 2006

Proceedings of the
**Asia Pacific Symposium on
Information Visualization (APVIS2006)**,
Tokyo, Japan, February, 2006

Kazuo Misue, Kozo Sugiyama and Jiro Tanaka, Eds.

Volume 60 in the Conferences in Research and Practice in Information Technology Series.
Published by the Australian Computer Society Inc.



Published in association with the ACM Digital Library.

Information Visualization 2006. Proceedings of the Asia Pacific Symposium on Information Visualization (APVIS2006), Tokyo, Japan, February, 2006

Conferences in Research and Practice in Information Technology, Volume 60.

Copyright ©2006, Australian Computer Society. Reproduction for academic, not-for profit purposes permitted provided the copyright text at the foot of the first page of each paper is included.

Editors: **Kazuo Misue**
Department of Computer Science
University of Tsukuba
Tsukuba
Japan
Email: misue@cs.tsukuba.ac.jp

Kozo Sugiyama
School of Knowledge Science
Japan Advanced Institute of Science and Technology (JAIST)
Ishikawa
Japan
Email: sugi@jaist.ac.jp

Jiro Tanaka
Department of Computer Science
University of Tsukuba
Tsukuba
Japan
Email: jiro@cs.tsukuba.ac.jp

Series Editor: John F. Roddick,
Conferences in Research and Practice in Information Technology
Flinders University,
PO Box 2100, Adelaide 5001
South Australia.
crpit@infoeng.flinders.edu.au

Publisher: Australian Computer Society Inc.
PO Box Q534, QVB Post Office
Sydney 1230
New South Wales
Australia.

Conferences in Research and Practice in Information Technology, Volume 60.
ISSN 1445-1336.
ISBN 1-920-68241-4.

Printed, December 2005 by Flinders Press, PO Box 2100, Bedford Park, SA 5042, South Australia.
Cover Design by Modern Planet Design, (08) 8340 1361.

The *Conferences in Research and Practice in Information Technology* series aims to disseminate the results of peer-reviewed research in all areas of Information Technology. Further details can be found at <http://crpit.com/>.

Table of Contents

Information Visualization 2006. Proceedings of the Asia Pacific Symposium on Information Visualization (APVIS2006), Tokyo, Japan, February, 2006

| | |
|----------------------------|-----|
| Preface | ix |
| Programme Committee | x |
| Organising Committee | xi |
| Sponsors | xii |

Keynote Papers

| | |
|--|---|
| Cyber Security Through Visualization | 3 |
| <i>Kwan-Liu Ma</i> | |
| Japanese Landscape and My Environmental Design | 9 |
| <i>Taketo Kurokawa</i> | |

Contributed Papers

Visualization Techniques

| | |
|---|----|
| Polygons Labelling of Minimum Leader Length | 15 |
| <i>Michael A. Bekos, Michael Kaufmann, Katerina Potika and Antonios Symvonis</i> | |
| Visualization of multi-dimensional data of bioactive chemicals using a hierarchical data visualization technique "HeiankyoView" | 23 |
| <i>Takayuki Itoh and Fumiyoshi Yamashita</i> | |
| Mesh Simplification using Ellipsoidal Schema for Isotropic Quantization of Face-Normal Vectors - Short Paper | 31 |
| <i>Ganesan Subramaniam and Kenneth Ong</i> | |
| Visualization of a Closed Three-Dimensional Surface using Portal-based Rendering - Short Paper ... | 35 |
| <i>Michael Bui, Nick Lowe and Masahiro Takatsuka</i> | |

Applications 1

| | |
|---|----|
| Coordinated Perspectives and Enhanced Force-Directed Layout for the Analysis of Network Motifs .. | 39 |
| <i>Christian Klukas, Falk Schreiber and Henning Schwöebbermeyer</i> | |
| Generation of Relevance Maps and Navigation in A Digital Book | 49 |
| <i>Katsuhiro Ikeda, Kozo Sugiyama, Isamu Watanabe and Kazuo Misue</i> | |
| Knowledge Visualization in Hepatitis Study - Short Paper | 59 |
| <i>DucDung Nguyen, TuBao Ho and Saori Kawasaki</i> | |
| Integrated Visualization for Geometry PIG Data - Short Paper | 63 |
| <i>Bok Dong Kim, Sang Ok Koo, Hyok Don Kwon, Seong Dae Jung, Soon Ki Jung, Minhoo Lee, YongWoo Rho and SungJa Koo</i> | |

Graph Drawing 1

| | |
|--|----|
| Increasing the Readability of Graph Drawings with Centrality-Based Scaling | 67 |
| <i>Damian Merrick and Joachim Gudmundsson</i> | |
| Visualizing Multivariate Network on the Surface of a Sphere | 77 |
| <i>Yingxin Wu and Masahiro Takatsuka</i> | |
| An Experimental Study on Algorithms for Drawing Binary Trees - Short Paper | 85 |
| <i>Adrian Rusu, Radu Jianu, Confesor Santiago and Christopher Clement</i> | |
| Method for Drawing Intersecting Clustered Graphs and Its Application to Web Ontology Language - Short Paper | 89 |
| <i>Hiroki Omote and Kozo Sugiyama</i> | |

Systems

| | |
|--|-----|
| BiblioViz: A System for Visualizing Bibliography Information | 93 |
| <i>Zeqian Shen, Michael Ogawa, Soon Tee Teoh and Kwan-Liu Ma</i> | |
| Information Gathering Support Interface by the Overview Presentation of Web Search Results | 103 |
| <i>Takumi Kobayashi, Kazuo Misue, Buntarou Shizuki and Jiro Tanaka</i> | |
| NeL2: Network Drawing Tool for Handling Layered Structured Network Diagram | 109 |
| <i>Nagayoshi Nakazono, Kazuo Misue and Jiro Tanaka</i> | |
| Interactive Optimization in Cooperative Environments - Short Paper | 117 |
| <i>Joelma de Moura Ferreira, Hugo A. Dantas do Nascimento and Eduardo Simões de Albuquerque</i> | |

Applications 2

| | |
|---|-----|
| Enhancing the Visualization Process with Principal Component Analysis to Support the Exploration of Trends | 121 |
| <i>Wolfgang Müller, Thomas Nocke and Heidrun Schumann</i> | |
| STARMINE : A Visualization System for Cyber Attacks | 131 |
| <i>Yusuke Hideshima and Hideki Koike</i> | |
| Spatial analysis of centralization and decentralization in the population migration network - Short Paper | 139 |
| <i>Shinji Tomita and Yukio Hayashi</i> | |
| Visualization by information type on mobile device - Short Paper | 143 |
| <i>Hee Yong Yoo and Suh Hyun Cheon</i> | |

Visual Interfaces

| | |
|---|-----|
| FindFlow: Visual Interface for Information Search based on Intermediate Results | 147 |
| <i>Tomoyuki Hansaki, Buntarou Shizuki, Kazuo Misue and Jiro Tanaka</i> | |
| Ripple Presentation for Tree Structures with Historical Information | 153 |
| <i>Masaki Ishihara, Kazuo Misue and Jiro Tanaka</i> | |
| SnapShoot: Integrating Semantic Analysis and Visualization Techniques for Web-based Note Taking System | 161 |
| <i>Soichiro Iga and Makoto Shinnishi</i> | |

Graph Drawing 2

| | |
|---|-----|
| Drawing Bipartite Graphs as Anchored Maps | 169 |
| <i>Kazuo Misue</i> | |
| Mental Map Preserving Graph Drawing Using Simulated Annealing | 179 |
| <i>Yi-Yi Lee, Chun-Cheng Lin and Hsu-Chun Yen</i> | |
| Visual Analysis of Network Centralities | 189 |
| <i>Tim Dwyer, Seok-Hee Hong, Dirk Koschützki, Falk Schreiber and Kai Xu</i> | |

Visual Representation

| | |
|--|-----|
| How People Read Sociograms: A Questionnaire Study | 199 |
| <i>Weidong Huang, Seok-Hee Hong and Peter Eades</i> | |
| Predicting Graph Reading Performance: A Cognitive Approach | 207 |
| <i>Weidong Huang, Seok-Hee Hong and Peter Eades</i> | |
| Visual Narratives: The Essential Role of Imagination in the Visualization Process - Short Paper | 217 |
| <i>Mark Baskinger and Ki-Chol Nam</i> | |
| Pattern Puzzle: A Metaphor for Visualizing Software Complexity Measures - Short Paper | 221 |
| <i>Adam Ghandar, A.S.M Sajeev and Xiaodi Huang</i> | |
| Author Index | 225 |

Preface

Information visualization is an indispensable tool in the information and knowledge driven society that has already started. It helps to tackle the increasing size and complexity of data in a wide range of applications. Examples of application domains include capital markets, data mining, software development, communication, logistics, social networks, knowledge management and bio-informatics. The discipline includes elements of many fields, including human-computer interaction, computer graphics, cognitive psychology, statistics, graphic design, algorithms, virtual and augmented reality, and software engineering.

The Asia-Pacific Information Visualization Symposium provides an opportunity for the community of researchers in the Asia-Pacific region to share the progress that has been made in information visualization with colleagues.

The papers in this proceedings were selected for presentation at the Second Asia-Pacific Symposium on Information Visualization (APVIS 2006), held on February 1-3, 2006 in Tokyo, Japan. The conference was supported by the 21st-Century COE Program *Technology Creation Based on Knowledge Science* (JAIST) and the Grant-in-Aid for Scientific Research on Priority Area *Informatics* of MEXT, Japan, and cooperated with ACM Japan Chapter, CRPIT, Australian Computer Society, Fujitsu Laboratories Limited, Japan Society for the Science of Design, and National Institute of Informatics, Japan.

In response to the call for papers, we received 37 submissions from Australia, Brazil, China, Germany, Greece, Japan, Singapore, Korea, Taiwan, and United States. Each submitted paper was fully refereed by at least three program committee members and 19 long papers and 11 short papers were selected for publication. We also received two papers from keynote speakers.

We thank all program committee members for their excellent work, especially given the time constraints. We also thank keynote speakers and all who submitted papers for consideration; they all contributed to the high quality of the conference.

Finally, we thank the members of the organizing committee. It is their work that made the conference possible and enjoyable.

Kozo Sugiyama

Japan Advanced Institute of Science and Technology
APVIS2006 Symposium Chair

Jiro Tanaka

University of Tsukuba
APVIS2006 Program Co-chair

Kazuo Misue

University of Tsukuba
APVIS2006 Program Co-chair

February, 2006

Programme Committee

Chairs

Jiro Tanaka, University of Tsukuba, Japan
Kazuo Misue, University of Tsukuba, Japan

Members

James Abello, ASK.com and DIMACS, Rutgers University, USA
Neville Churcher, University of Canterbury, New Zealand
Guozhong Dai, Chinese Academy of Sciences, China
Peter Eades, National ICT Australia, University of Sydney, Australia
Seokhee Hong, National ICT Australia, University of Sydney, Australia
Hiroshi Hosobe, National Institute of Informatics, Japan
Takayuki Itoh, Ochanomizu University, Japan
Hideki Koike, University of Electro-Communications, Japan
Wei Lai, Swinburne University of Technology, Australia
Kwan-Liu Ma, University of California Davis, USA
Kazuyuki Miura, Fukushima University, Japan
Edmond Cyril Prakash, Nanyang Technological University, Singapore
Yeong-Gil Shin, Seoul National University, Korea
Kozo Sugiyama, Japan Advanced Institute of Science and Technology, Japan
Vincent Tam, University of Hong Kong, Hong Kong
Bruce Thomas, University of South Australia, Australia
Rudi Vernik, Defence Science and Technology Organisation, Australia
Hui Wang, Chinese Academy of Sciences, China
Pak Chung Wong, Pacific Northwest National Laboratory, USA
Hsu-Chun Yen, National Taiwan University, Taiwan

Organising Committee

Chair

Kozo Sugiyama, Japan Advanced Institute of Science and Technology, Japan

Members

Hiroshi Hosobe, National Institute of Informatics, Japan

Takayuki Itoh, Ochanomizu University, Japan

Hideki Koike, University of Electro-Communications, Japan

Kazuo Misue, University of Tsukuba, Japan

Jiro Tanaka, University of Tsukuba, Japan

Sponsors

Host Organization

APVIS2006 Organizing Committee

Co-host Organization

21st-Century COE Program "Technology Creation Based on Knowledge Science" (JAIST)
Grant-in-Aid for Scientific Research on Priority Area "Informatics" (Area #006) of MEXT, Japan

In Cooperation with

ACM Japan Chapter
CRPIT, Australian Computer Society
Fujitsu Laboratories Limited
Japan Society for the Science of Design
National Institute of Informatics, Japan

KEYNOTE PAPERS

Cyber Security Through Visualization

Kwan-Liu Ma

Department of Computer Science
University of California at Davis
Email: ma@cs.ucdavis.edu

Networked computers are subject to attack, misuse, and abuse. Organizations and individuals are making every effort to build and maintain trustworthy computing systems. The main strategy is to closely monitor and inspect network activities by collecting and analyzing data about the network traffic and the trails of system usage. The analysis usually requires large amounts of finely detailed, high-dimensional data to enable analysts to uncover hidden threats and make calculated predictions in a timely fashion. The traditional, signature-based and statistical methods are limited in their capability to cope with the large, evolving data and the dynamic nature of the Internet. Visualization proves effective to aid in understanding large, high-dimensional data commonly found in many demanding applications such as large-scale scientific simulations and biomedicine. There is thus a growing interest in the development of visualization methods as alternative or complementary solutions to the pressing cyber security problems (Brodley, Chan, Lippmann & Yurcik 2004, Ma, North & Yurcik 2005). The challenge is to develop new visual representations, layout methods, user interfaces, and interaction techniques that can effectively facilitate visual interrogation and communication of the vast amounts of cyber security information.

Visual data analysis is inherently an iterative process, where each iteration provides more insight into the data being shown. A typical example of this process occurs with any type of overview plus detail visualization. Patterns in the overview tend to direct what the user chooses to view in more detail, and the detailed view can provide insight on regions of the overview. This drill-down process, starting at a high semantic level and progressing to more detailed views, creates a feedback loop as shown in Figure 1, which can lend itself well to visualizing the relationships between large number of objects, such as port and network scans. In most cases, different visual representations are needed for constructing these different views. In particular, each specific region of interest may be defined in a space of arbitrary dimensions. The challenge is thus to seek the best space and visual representation in that space for each type of analysis task. I show with three different tasks how visualization can assist in the analysis of computer network activities for detecting anomalies using the drill-down process.

Copyright ©2006, Australian Computer Society, Inc. This paper appeared at Asia Pacific Symposium on Information Visualization (APVIS 2006), Tokyo, Japan. Conferences in Research and Practice in Information Technology, Vol. 60. K. Misue, K. Sugiyama and J. Tanaka, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

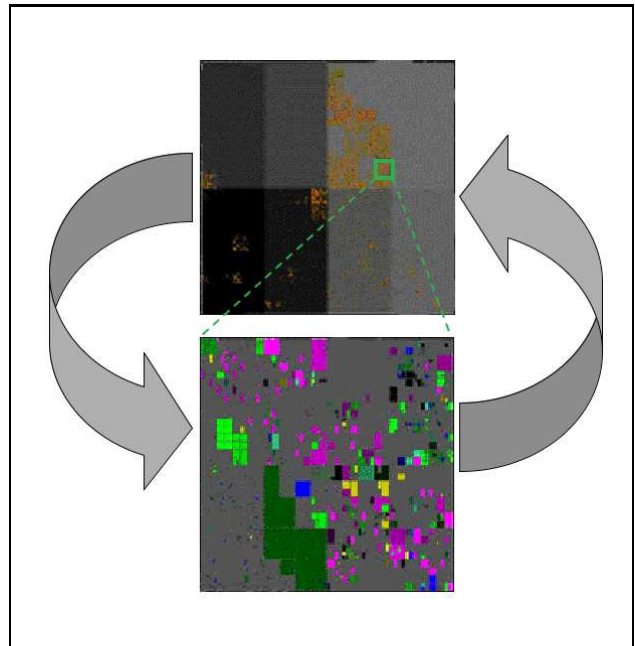


Figure 1: The process of visual data analysis is inherently iterative to see both summary and details in context.

Analysis of Internet Routing Data

The Internet can be considered as a set of sub-networks, each of which represents an organization's network. The problem of packet routing can simplify to routing data between these larger entities, referred to Autonomous Systems, according to the Border Gateway Protocol (BGP). For data packets to arrive at the correct destination, these Autonomous Systems exchange network reachability information in the form of BGP path announcements. Studying and understanding the dynamics of BGP routing changes is thus crucial to ensuring robust network performance. A drill-down process of analysis (Teoh, Jankun-Kelly, Ma & Wu 2004) can start by looking at the aggregate information about routing changes over a complete period of time, followed by examining routing update messages over selected period of time and their corresponding statistical values; next, particular instances of instability can be visualized in detail. Figure 2 shows a two-level aggregate data browser which displays the distribution of the BGP announcements over a 1-year period (bottom), and allows the analyst to select a focused period of several minutes. Figure 3 shows the color coded text visualization of individual BGP path announcements as well as statistical measures of the routing updates during the focused period. This joint visualization al-

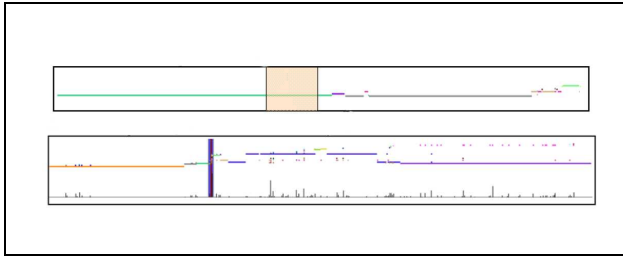


Figure 2: A two-level aggregate data browser can cover a wide range of granularity. Bottom: At the overview level, the analyst first looks at the aggregate information of the entire time period (typically one year) and specifies a period to focus on. Top: At the next level, the focus can be further narrowed down to a period of several minutes.

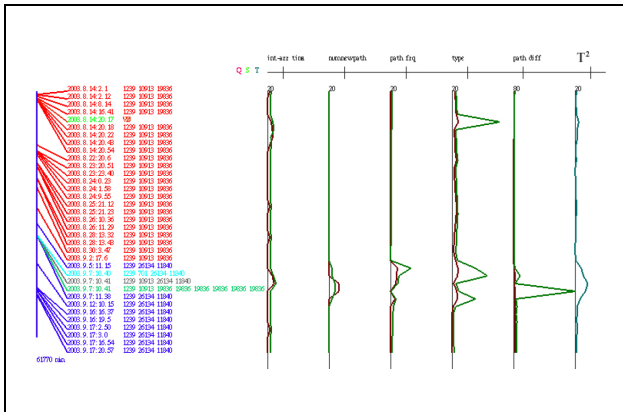


Figure 3: Left: Text visualization of a sequence of Autonomous System (AS) path announcements. Each unique AS path is shown in a different color, which effectively forms visual patterns of the updates such as oscillations, repeats, or slow convergences. Right: Statistical measures corresponding to each announcement can be used to help verify any detected anomaly.

allows for verification of the visual and statistical information for anomaly detection (Teoh, Zhang, Tseng, Ma & Wu 2004). After instability events are identified, it is possible to see the distribution of different events, their severity, duration, and frequency all in one single visualization (Teoh, Ma & Wu 2003), as shown in Figure 4.

Port Data Visualization

Scanning a network is a very common first step in a network intrusion attempt. Crackers frequently scan entire ranges of ports, looking for open ports that can be exploited to gain access to a system. Worms and viruses often target specific ports in an attempt to locate systems that are vulnerable to the mechanisms they use to spread. These attacks are all recorded in security logs, but these logs are time-consuming for administrators to analyze by hand. One way to understand the collected security logs is to produce images of network traffic by choosing axes that correspond to important features of the data, creating a grid based on these axes and then assigning each cell of the grid with a visual property such as color to represent the network activity there.

The drill-down process of analyzing port data begins with an overview that presents a time-ordered view of the entire data set. The goal here is to choose a particular range of time to zoom into. In the following steps, detailed views are created to eventu-

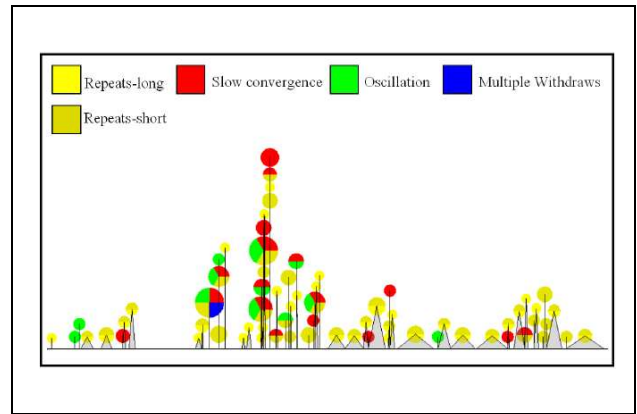


Figure 4: Visualization of instability events for a particular IP prefix. Each event is represented by a circle and a base. The area of the circle is proportion to the number of announcements and each color segment indicates a specific type of instability. The triangular base shows the duration of the event. The position of each circle is placed to avoid occlusion. Nevertheless, tall events suggest that there are many events at that time.

ally reach an atomic unit of network security interest, which may represent a port scan, an intrusive attack, the activities of spyware, the flocking of employees to Web news sites after a major news event, or any other discrete feature that can be identified in the data. Figure 5 shows, from left to right, a 3-tier process for studying TCP port data (Muelder, Ma & Bartoletti 2005a), in which the left most image displays a highly condensed port data over a period of time such as a week or a month. Each row of the visualization represents one unit (generally an hour) of time. Each pixel corresponds to a range of ports and is colored according to the level of activity on the ports during the time unit. Figure 6 shows different levels of enhancement can bring out port scans; furthermore, using different data metrics can reveal different patterns in the data leading to new discoveries (Muelder et al. 2005a). The grid visualization, shown in the middle of Figure 5, depicts the activity during a given time unit. It consists of a dot on the grid for each of the 65,536 ports. The user can select a port to see detailed information about that port and its surrounding ports, as shown in the upper right image in Figure 5. The bottom right picture shows a single port over the entire time range, for each of the metrics of interest including, from top to bottom, session count, destination count, source count, the ratio of source and destination, and country count. Such a visualization is useful for finding relationships between metrics, as well as showing periodic structures in the data such as the change in web traffic throughout the day. Figure 7 presents some distinct patterns of activity.

Scan Characterization

In order to obfuscate an attack, an attacker frequently alters identifying features like source IP addresses. Thus, in order to identify an attacker, some more immutable aspect of the attack must be considered, such as packet arrival timing, which is dependent on several factors that are difficult to alter, such as hardware or operating system limitations or router delays. However, due to the chaotic nature of packet arrival times, one must analyze a large quantity of packets. Network scans provide a good source of such timing information. It is thus beneficial to take network scan

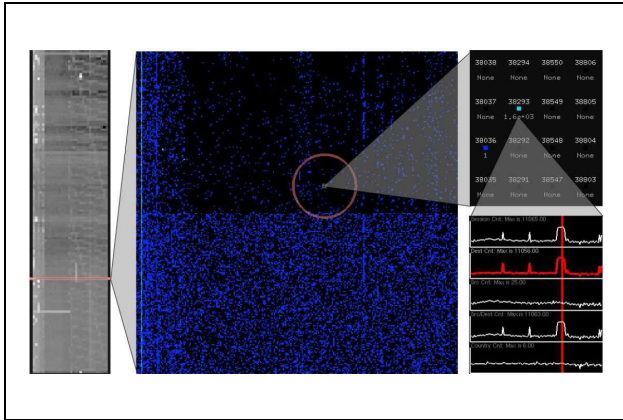


Figure 5: A drill-down process for finding network scans by applying it to a 24 hour long dataset at 10 minute resolution. Starting at the timeline on the left, a spike is found on a high port that crosses several hours. One of these hours is then selected for viewing in the grid based visualization shown in the middle. In it, there is exactly one port with unusually large values in the range of ports that correspond to the spike. The range around this port is zoomed into which reveals in the bottom-right image that an abnormally large number of destinations being connected to by a small number of sources, which means that this is likely a network scan.

timing information and use visualization techniques to characterize the scans.

An underlying premise of this approach to the network scan characterization problem is that humans are creatures of habit (and autonomous software applications even more so.) Having created an environment of attack tools with which they have become familiar, they tend to reuse those tools and support systems in future activities. The particular settings employed would likely remain consistent as well, both out of familiarity and a desire to properly compare recent results with earlier findings. Additionally, other software processes that may be running concurrently in support of analysis or ancillary activities compete with and impact the performance of their network activities in reliably consistent ways, imposing uniquely identifiable characteristics in the sequence and packet arrival times of high packet count interactions.

Individual scans can be shown with a grid-based visualization, where the axes are the third and fourth octets of the destination IP addresses, and the color is based on a metric derived from timing information. Metrics range from simple metrics such as arrival time of the first probe or number of probes for each address to complex metrics such as deviation from a linear expected arrival time for each destination. Patterns that are nearly identical in one metric can be distinct in others. Figure 8 shows such scan visualizations.

The relationships between network scans may be understood by statistically comparing pairs of scans and it is also possible to get a quantitative measure on how well they match. However, because the scans are too chaotic to compare directly, frequency analysis, such as Fourier transforms or wavelet transforms, can be used to convert the scan patterns into scalograms, which can then be systematically compared (Muelder, Ma & Bartoletti 2005b). Although network scan patterns can exhibit a periodic or quasi-periodic structure, they often contain gaps, aperiodic aberrations and regions where the relative phase of the periodic structures has shifted, which are things that Fourier analysis has been found to not handle well. Wavelets, on the other hand, are relatively resistant to phase

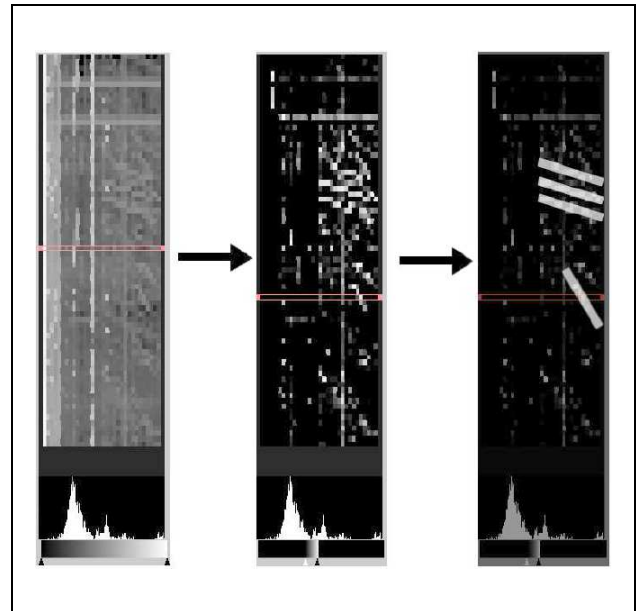


Figure 6: Visualization of the entire time range with enhancement to bring out port scans corresponding to a particular spike using the activity-level histogram and gradient editor shown on the bottom.

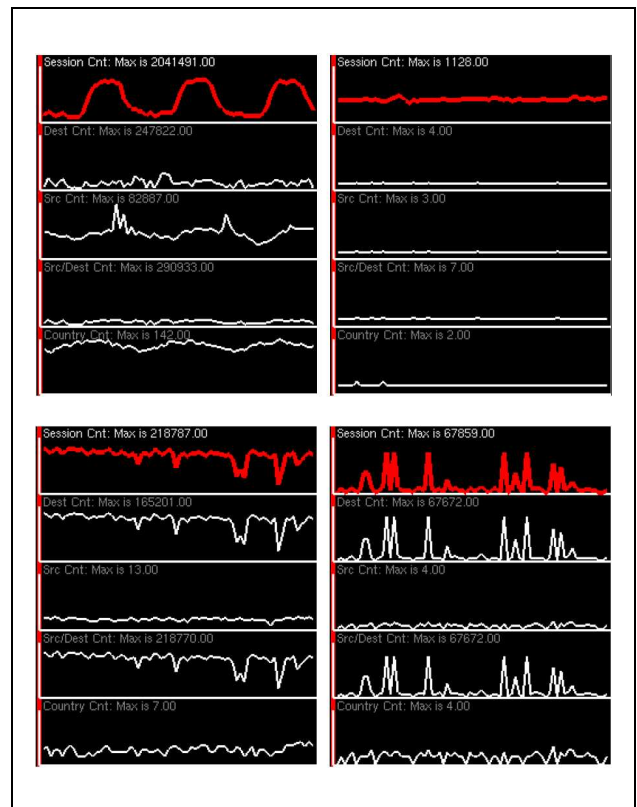


Figure 7: Plots metrics versus time for individual ports. In each example, the session count (the first metric) is highlighted. The other four metrics shown are destination address count, source address count, unique source and destination pair count, and country count. The usage of Port 80 (upper left) is very periodic while Port 46011 (upper right) has a fairly constant level of activity, with a few spikes. Port 27374 (lower left) is more erratic, though, interestingly, its usage drops noticeably as time goes on. Port 34816 (lower right) has one of the most suspicious usage graphs; it is only used a few times, but it is used fairly heavily during those times.

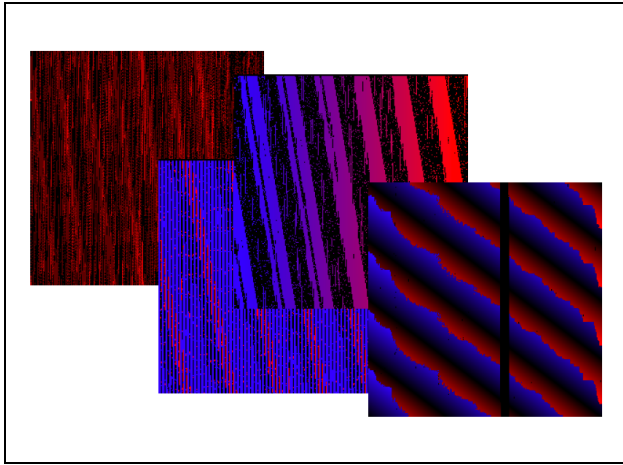


Figure 8: Visualization of individual scans showing patterns that can easily be compared by eye. These images show the arrival time of the first connection attempt to each address, with blue being early in the scan, red being late in the scan, and black being an address that had no connection attempt.

shifts and noise. Figure 9 shows that dissimilar patterns result in different scalograms.

At this point, the scans can be directly visualized individually, but when dealing with large numbers of scans, this is unfeasible. So, once the scans are isolated, in order to automatically compare them, fingerprints are generated to be fed into an overview visualization. This overview of the relationships between the scans and the detailed view of individual pairs of scans for comparison purposes compose an overview plus detail feedback loop. As described before, the overview allows the analyst to drill down into certain areas, by showing them in the detailed view.

The overview can be provided by a graph visualization of the relationship between scans. Each node is a scan, and each edge is their relationship. The edge weights are derived from the wavelet analysis of the scans, and they range from 0, which means the scans are completely different, to 1 which means they are identical. The graph can be displayed with a force directed layout algorithm and edges below a particular threshold may be dropped for clarity. The resulting image then shows clusters of scans that are similar, as shown in Figure 10. The user can start with such an overview graph and then drill down to detailed characteristics of the scans in the same cluster. In this way, a graph of a large number of scans can be rapidly compared and subsequently identified. Furthermore, it is very helpful to allow the analyst to bias the overview in a manner reflecting the cognitive insight gained from looking at the detail view (Muelder et al. 2005b). This enhances the feedback loop by allowing information gained by viewing the details of network scans to be reflected back in the overview.

Concluding Remarks

Visualization leverages human's extraordinary ability to detect patterns in images; nevertheless, by itself visualization does not answer all the questions the analyst has. Visualization is best for guiding a complex data analysis process since visualization is particularly good for showing an overview of the data, which can direct the analyst's attention to the aspects of the data that require further investigations, as demonstrated by the three examples I have given. The ability to show details in context with visualization is also very powerful. To fight against the in-

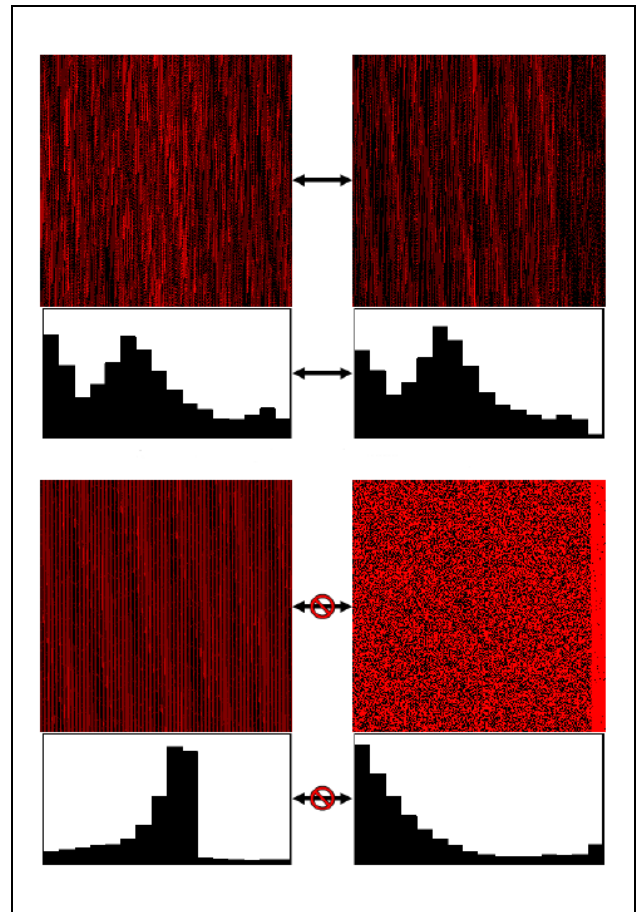


Figure 9: Wavelet scalograms reduce large complex pattern to smaller simpler vectors that can be compared. This example was made using a metric based on the number of visits per unique address, with black to red gradient, where black is no probes and red is the maximum for that scan. Top: Similar scans have similar wavelet scalograms. Bottom: Dissimilar scans have very different scalograms.

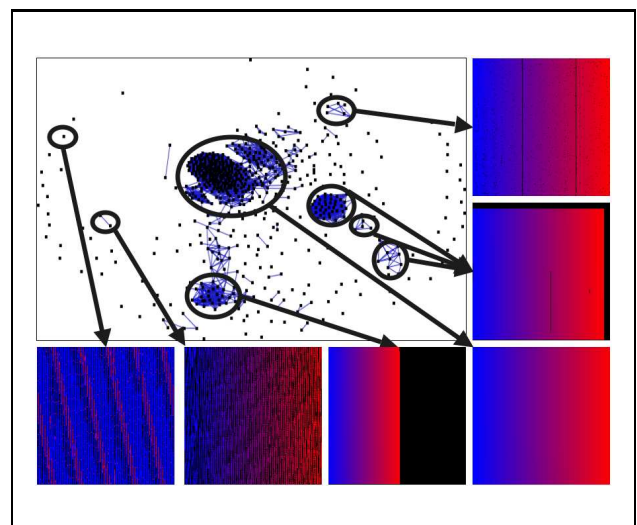


Figure 10: Graph visualization of network scans. Clusters contain scans with a general pattern. A representative example from each selected cluster is shown.

creasingly creative and malicious attacks to network security I believe a promising approach is to add intelligent reasoning with learning capabilities into the visualization directed analysis.

Cyber security is a much broader topic than ensuring the normal operation of a computer network. The task of analyzing network security data represents only a small subset of the greater security problem that we are faced with today. For example, data collected for intelligence information analysis are more heterogeneous, including text, measurements from sensors, imagery, video and audio from diverse sources. What visual representations should we use to study such heterogeneous data in a unified manner? What would be the meaningful linkages among the disparate data to facilitate cross-exploration? For the field of information visualization, this new class of problems presents many challenges and open research questions. We have only addressed a very small fraction of these challenges. Please join me in this research endeavor.

References

- Brodley, C., Chan, P., Lippmann, R. & Yurcik, B., eds (2004), *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security (VizSEC/DMSEC 2004)*, ACM.
- Ma, K.-L., North, S. & Yurcik, B., eds (2005), *Proceedings of the IEEE Workshop on Visualization for Computer Security 2005 (VizSEC 2005)*, IEEE Computer Society.
- Muelder, C., Ma, K.-L. & Bartoletti, T. (2005a), Interactive visualization for network and port scan detection, in 'Proceedings of the Eighth International Symposium on Recent Advances in Intrusion Detection (RAID 2005)'.
- Muelder, C., Ma, K.-L. & Bartoletti, T. (2005b), A visualization methodology for characterization of network scans, in 'Proceedings of the Workshop on Visualization for Computer Security (VizSEC 2005)', pp. 29–38.
- Teoh, S. T., Jankun-Kelly, T. J., Ma, K.-L. & Wu, S. F. (2004), 'Visual data analysis for detecting flaws and intruders in computer network systems', *IEEE Computer Graphics and Applications (special issue on Visual Analytics)* **24**(5), 27–35.
- Teoh, S. T., Ma, K.-L. & Wu, S. F. (2003), Visual exploration process for the analysis of internet routing data, in 'Proceedings of the IEEE Visualization 2003 Conference', pp. 523–530.
- Teoh, S. T., Zhang, K., Tseng, S.-M., Ma, K.-L. & Wu, S. F. (2004), Combining visual and automated data mining for near-real-time anomaly detection and analysis in gbp, in 'Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security (VizSEC/DMSEC 2004)', pp. 35–44.

Japanese Landscape and My Environmental Design

Taketo Kurokawa

Space Design Course, Dept. of Design

Kanazawa College of Arts

5-11-1 Kodatsuno Kanazawa-city, Ishikawa-ken 920-8656 JAPAN

kuro-t@kanazawa-bidai.ac.jp

Abstract

This paper introduces studies on Japanese Landscape and environmental design work by author. Japanese people have venerated nature from the far ancient times and led life in unison with nature. This paper describes about the characteristics of Japanese architecture, the characteristics of Japanese civilization, and influence on "Japonism" from the viewpoint of the author as a design researcher and a designer. It is said the environmental design is the ideal method based on Japanese civilization. The author considers environmental design as a figurative method which is created with the sensitivity brewed in the climate of Japan and which is invented from various interrelations, and he shows his own environmental design to replace the conclusion.

Keywords: Space design, Environmental design, Landscape.

1 Preface

The first text that has a discussion over Japanese scenery and landscape is considered to be Shiga Shigetaka's "Japanese Landscape (1896)". However, it was Law Concerning Special Measures for Preservation of Historic Natural Features in Ancient Cities practiced in 1967 that brought an exponential increase in the number of texts on townscape and its preservation. The momentum of this trend was even strengthened as more local government landscape regulations were practiced. It seems that the momentum of this trend has reached its peak with the enactment and enforcement of the Scenery Law.

In this paper, the author will discuss Japanese original scenery and landscape from the viewpoint of the author as a design researcher and a designer and will also include some discussion on environmental design which is the author's expertise.

2 Japanese original landscape

Japanese people have venerated nature from the far ancient times and led life in unison with nature. They saw the existence of gods in all sorts; a mountain, a river, a plant,

and a tree. This is so-called animism. However, unlike other nations of the arid region from Central Asia, Middle East and North Africa or West Europe where polytheism was gradually converged into monotheism, Japanese civilization was established with polytheism.

In one research, I asked Japanese researchers the original landscape for them, and more than a half of their replies were natural landscape such as a mountain and river, or woods. It was followed by the "townscape" and "village scape" but the details of the townscape showed that more than 40 percent of "townscape" was occupied by the wooden constructions. This shows their familiarity towards the scenes where nature is felt in human works. Many of the details in the whole data on the same level are a cityscape, grove, village, country, river, and street. "Modern scenery" appears with the frequency of only a few percent [1].



Figure 1: "Haseyama Shugetsu" from Eight Views of Kinjo. Individual collection. Edo period.

Since the Edo period, following the example of "Eight Views of the Xiao and the Xiang" in China, local governments began selecting the scenic places called eight views and ten views including "Japanese Eight Views". In Kanazawa city where I live, scenic places such as "Eight Views of Kinjo" and "Kanazawa Eight Views" were also selected. Many are the places connected with woods close to a village such as "Haseyama Shugetsu" (Figure-1) on the hillside of Utatsu mountain spreading over the right bank of Asano river or "The Shower in Kuragatake" of Sai river upstream, or those where nature and human life is harmonized, such as a scene of boats sailing from an inlet like "Sailing at Miyanokoshi." From such characteristics

of Japanese people, Augustin Berque, a French anthropologist, assumes that the highest-level form of Japanese civilization points to nature [2].

3 The characteristics of Japanese architecture

Meanwhile, Augustin Berque also writes in the same book that people of any country feel “no other culture pays as much respect to nature as theirs”. However, Bruno Taut, a German architect, who left for Japan to escape from the Nazis, was overwhelmed when he visited Katsura Rikyu the day after he landed on Maizuru harbor. Its beauty harmonized with nature exceeded his anticipation by far (Figure-2).

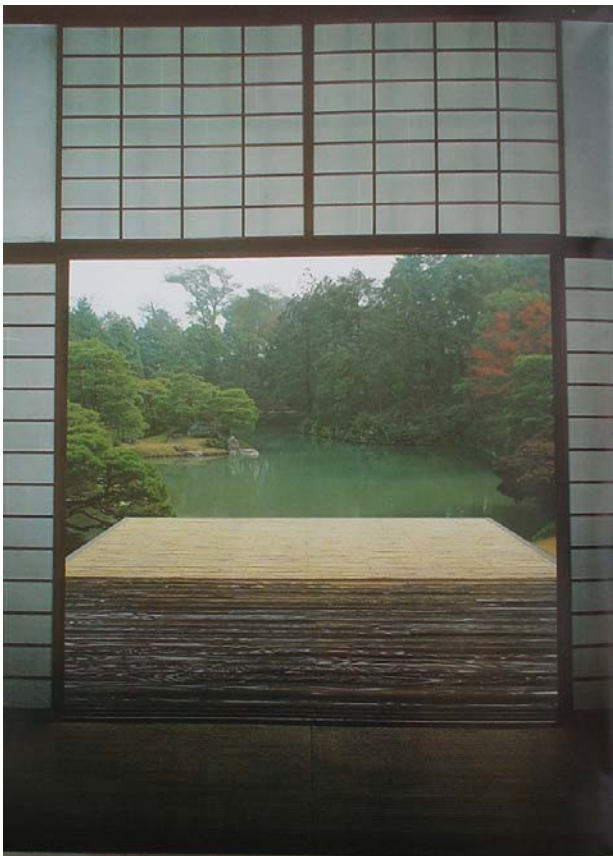


Figure 2: Photo of Katsura-Rikyu

This photograph is a part of the garden viewed from the inside of the residence. The garden seems like woods deep in a mountain and, in fact, it was made such. In Japanese style, the original beauty of nature kept unspoiled while natural settings are completely tampered. Constructions appear themselves in nature only from time to time.

Figure-3 is a construction in unison with nature even more directly, and that is a pit dwelling. There is also a pit dwelling with a common thatched roof. It is described that ancient Yamato Takeru punished earth spiders or the like as they resisted to him in his conquest of the east (Hitachi-no-kuni Hudoki). It is assumed that the earth spiders were possibly those living in such pit dwellings. Although a pit dwelling has no window and is not as open as beam-column structure, which is the characteristic of

Japanese architecture after the classic aristocratic mansion style, it is the same at the point that it is well harmonized with nature.



Figure 3: Photo of a pit dwelling

4 Characteristics of Japanese civilization

A new study field called phisiography has been established recently mainly by the scholars known for studying landscape. The aim of phisiography is to investigate how Japanese land was formed by a method different from the conventional ones. One of the results is the historical investigation of Kesaki legend (a giant kicked the bank of a lake and arable land was made), a legend that various regions of Japan have. The investigation found that this legendary deed was done not by the gods but the people [3]. Japan achieved the top-class population density by the power of the people even before Meiji Restoration which is generally regarded as the turning point of modernization (see Table-1).

Japanese people increased arable land strenuously by public works, produced more food, and developed culture. As a result, the period without war lasted for 350 years in Heian period, and 250 years in Edo period. This became the foundation of Japanese civilization.

In this aspect, Japanese people succeeded in utilizing and controlling nature. Umesao Tadao suggests that the root of advancement of Japanese civilization is that it was not invaded by a foreign enemy because Japan is separated from an arid region in the Eurasian continent. Recently, Kawakatsu Heita has pointed out further that the exchange of texts and goods across the ocean also influenced a lot [5].

From the end of Edo to Meiji was the period when the climax (reaching point) of such Japanese civilization appeared. Although many of the positive comments on Japan in the texts of the visitors from abroad have already been lost, to them, the sky of Edo (Tokyo) which had not received negative influence from the Industrial revolution was very clear, and the suburbs where well-cared-for paddy fields spread were as beautiful as gardens.

| | China | Germany | Great Britain | Japan | Notes |
|------------------------------|------------|-------------|---------------|----------------|---------------------|
| 2500 years ago | | | 1 mil. | 75 thousand | Jomon period |
| 2200–2400 years ago | 20 mil. | | | 0.3 mil. | Yayoi period |
| 2100–2200 years ago | | 3 to 4 mil. | | | |
| Present | 12–13 bil. | 82mil. | 58mil. | 1.28 bil. | |
| Growth rate | 60 times | 20–27 times | 58 times | 430–1700 times | |
| Before Industrial Revolution | | | | | |
| 1750 | | | 5.7 mil. | | |
| 1800 | | 16 mil. | | | |
| 1846 | | | | 32 mil. | End of Edo period |
| 1910 | 4 billions | | | | |
| Growth rate | 20 times | 4–5 times | 5.7 times | 100–400 times | |
| IR to Present | | | | | |
| Growth rate | 3 times | 5.1 times | 9.6 times | 4 times | China is developing |

Table 1: Transition of population in the world

(Reference: “Chimongaku Kotohajime: Nihonjin wa Donoyou ni Kokudo wo Tsukuttaka”)

5 Influence on "Japonism"

Vincent van Gogh, in his letter to his younger brother Theodorus (Gogh's letter), wrote that any impressionist would yearn for Japan and many artists other than Gogh were also influenced by Ukiyoe prints, brought from Japan, and travel journals. Above-mentioned Bruno Taut is also one of them and everybody knows the influence of Japan on the glassworks by Emile Galle with the motif of insects and flowers. The attitude of Japanese people toward nature even created a new trend in art called “Japonism” (Figure-4).

**Figure 4: Japonism painting by Vincent van Gogh**

The author came to know only recently that Georges Henri Riviere, a museologist who advocated ecomuseum, was also a painter and left pictures with a strong influence of Ukiyoe prints. The series, “Thirty Six Views of the Eiffel Tower,” is of course a work painted with “Thirty-Six Views of Mt. Fuji” in mind. The series have seals on each. Even today, French people may be the people most positive about Japanese sensitivity. Jean-Philippe Lenclos, a well-known colorist, is a friend of mine who held a symposium in Kanazawa with me only recently. He also often uses a seal in katakana for his letters.

6 My environmental design (to replace the conclusion)

My expertise is environmental design and I believe that environmental design is the ideal method based on Japanese civilization.

Its first characteristic is that it pays great attention to the interrelation; it regards “region (including place), “history”, and “ecology” as important features. For example, Japanese construction, interior, garden, and surrounding sceneries were realized only with their perfect interrelations among each. This is something the design in the 20th century ignored.

As one of the examples of interrelation, the one between construction and furniture or tools inside the construction will be discussed. In the most perfect Japanese-style constructions, most part of the floor is covered with tatami. Tatami is equipment developed to soften the relation between the construction and the human body. Since it is a tool of rough use in daily life, people invented the devices

to reduce the damage on tatami. One of the examples is a device used in Japanese lacquer woodcrafts called “tatami-zuri,” or “tatami-zure (Figure-5,6,7).”



Figure 5: An armrest (Furniture Museum)



Figure 6: A part of box-shaped small dining table (Furniture Museum)



Figure 7: A stepladder (Nagaoka district, Niigata)

The manner of not stepping on the edges of tatami is also considered to have originated from the custom of not stepping on the edges of tatami to protect from damages to tatami. Such attention is paid even to small tools of daily use; wooden tableware and wooden chopsticks are the outstanding material selection in which the delicate relation with small parts of human body, lips and hands, was considered, and the furniture such as fittings or floor cushions is changed from summer to winter, which is the wisdom to live comfortably in humid summer of Japan and was developed from the relation with the climate of Japan.

The author therefore considers environmental design as a figurative method which is created with the sensitivity brewed in the climate of Japan and which is invented from various interrelations [6]. In the lecture, hereafter, the author will introduce some of his works and explain more in detail.

(This paper was made with corroborative work by Y. Nagai Lab., Japan Advanced Institute of Science and technology)

7 References

1. KUROKAWA Taketo, “Classification of Landscapes by Scenes and Words -Original Landscape of the Japanese-”, No. 45 2001, Kanazawa College of Art bulletin
2. Augustin Berque, “Nihon no Hukei -Seiou no Keikan-Soshite Zoukei no Jidai” (Japanese Landscape and Western Landscape: the Time of Landscape), translated by Katsuhide Shinoda, Kodansha, 1990.
3. UEDA Atsushi and NAKAMURA Yoshio, “Nihonjin wa Donoyouni Kokudo wo Tsukuttaka” (How did the Japanese build their national land?), Gakugei Shuppan-sha, 2005.
4. UMESAO Tadao, “Nihon towa Nanika?” (What is Japan?), NHK Books, 1986.
5. KAWAKATSU Heita and YASUDA Kazunori, “Teki wo Tsukuru Bunmei Wa wo Nasu Bunmei” (Civilization which makes the enemy and civilization which makes harmony), PHP Research institute, 2003.
6. KUROKAWA Taketo, “Environmental Design” as an Ideal Design Methodology of the Japanese Model -New Positioning of Environmental Design Education in Universities of Japan,” The Bulletin of Kanazawa College of Art, 2001.MySQL: SQL Shareware Software, MySQL AB Co. <http://www.mysql.com/>. Accessed 29 Dec 2001.

CONTRIBUTED PAPERS

Polygons Labelling of Minimum Leader Length

Michael A. Bekos¹

Michael Kaufmann²

Katerina Potika¹

Antonios Symvonis¹

¹ National Technical University of Athens, School of Applied Mathematical & Physical Sciences,
15780 Zografou, Athens, Greece

mikebekos@math.ntua.gr, symvonis@math.ntua.gr, epotik@cs.ntua.gr

²University of Tübingen, Institute for Informatics, Sand 13,
72076 Tübingen, Germany

mk@informatik.uni-tuebingen.de

Abstract

We study a variation of the boundary labelling model, with *floating sites*, labels of uniform size placed in fixed positions on the boundary (that encloses all sites) and special type leaders connecting labels to sites. We seek to obtain a labelling of all sites with leaders that are non-overlapping and have minimum total length. We present an $O(n^2 \log^3 n)$ time algorithm for the labelling of polygons.

Keywords: map labelling, boundary labelling, floating sites, polygons.

1 Introduction

Placing extra information, in the form of text labels, next to features of a drawing (map) is an important task in the process of information visualization. Usually, it is desired that the label placement is done so that each label is close to the feature (site) it describes and is not intersecting with any other label. In general it is *NP*-hard (Formann & Wagner 1991) to obtain optimal label placements. An extensive bibliography about map labelling can be found at (Wolff & Strijk 1996). Besides labelling point feature in a map, some deal with labelling lines such that labels do not intersect (Strijk & van Kreveld 1999).

There are cases, i.e. when the labels are very large or the features are too many, where it is impossible to find a label placement so that the labels are close to the feature they describe. To cope with such cases, one direction is to allow the labels to be placed not close to each feature but in the boundary of the rectangle that encloses all features. Each feature is connected to its label by non-intersecting polygonal lines, called *leaders*. We call such a label placement *legal* or *crossing free*. The boundary labelling was first defined in (Bekos, Kaufmann, Symvonis & Wolff 2005). Bler (Bekos & Symvonis 2005) supports the boundary labelling process.

The sites model features of the drawing. Very often in practice, we want to label an area feature (e.g. a region of a map). Any point inside the area feature

can be arbitrarily chosen to represent the area in the input of the boundary labelling problem. However, instead of arbitrarily selecting a point to represent the area feature, we can specify as part of the input a *region* in which the site is allowed to “float” in any legal solution of the boundary labelling problem. To keep things simple, we specify these regions to be *generalized canonical polygons* or *rectangles* or *line segments* internal to the feature area, and assume that the site “slides” along the boundary of the polygon or on the line segment. We call *generalized canonical polygon* or *GC-POLYGON*, a simple closed polygon whose edges are vertical, horizontal or diagonal (at angles which are multiples of 45 degrees with respect to the axes). Figure 1 shows an example where the regions p_i and p_j are represented by GC-POLYGONS.

All labels have the same width and the same height. The labels are placed in distinct places on all four sides of the boundary of an axis parallel rectangle $R = [l_R, r_R] \times [b_R, t_R]$ of height $H = t_R - b_R$ and width $W = r_R - l_R$ which contains all sites p_i in P .

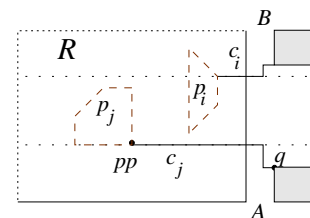


Figure 1: Leader c_j is oriented towards corner A and leader c_i is oriented away of corner A .

Each site is connected with its corresponding label in a simple and elegant way by using polygonal lines, called *leaders*. Labels are placed on the boundary of the enclosing rectangle and are connected to their site in such a way that the labels are non overlapping and the leaders are non crossing. In our approach we have leaders that consist of a single straight line segment or a sequence of rectilinear segments. When a leader is rectilinear, it consists of a sequence of axis-parallel segments either parallel (p) or orthogonal (o) to the side of R containing the label it leads to. The *type* of a leader is defined by an alternating string over the alphabet $\{p, o\}$. We use leaders of type-*opo* and *po*, see Figure 2. Furthermore, we assume that each type-*opo* leader has the parallel p -segment outside the bounding rectangle R , routed in the so-called *track routing area*. We consider type- o leaders to be of type-*opo* and of type-*po* as well.

Each leader that connects a site to a label, touches

Copyright ©2006, Australian Computer Society, Inc. This paper appeared at Asia-Pacific Symposium on Information Visualization (APVIS 2006), Tokyo, Japan, February 2006. Conferences in Research and Practice in Information Technology, Vol. 60. K. Misue, K. Sugiyama and J. Tanaka, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

The project is co - funded by the European Social Fund (75%) and National Resources (25%) - Operational Program for Educational and Vocational Training II (EPEAEK II) and particularly the Program PYTHAGORAS.

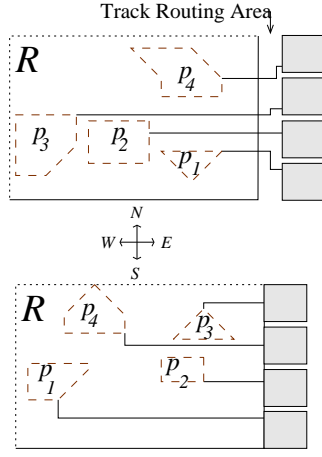


Figure 2: Type-*opo* (top) and type-*po* (bottom) leaders.

the label on a point on its side that faces the enclosing rectangle. The point where the leader touches the label is called *port*. We can assume *fixed ports* where the leader is only allowed to use a fixed set of ports on the label side (a typical case is where the leader uses the middle point of the label side) or *sliding ports* where the leader can touch any point of the label's side. In the labelling presented in Figure 1, the label port is point q . Let us denote by c_j the leader of site p_j . The labellings in Figures 1-2 use sliding ports.

We want to obtain labellings that optimize some criterion. Keeping in mind that we want to obtain simple and easy to visualize labellings, the following criterion of *minimizing the total leader length* can be adopted from the areas of VLSI and graph drawing. The length of a leader c_i is the Manhattan (or L_1) distance of site p_i to its label.

A GC-POLYGON p is a set of k corners indexed $\{1, \dots, k\}$ in clockwise direction that define the boundary of p . (see in Figure 1 GC-POLYGON p_i has 4 corners while GC-POLYGON p_j has 5 corners). We extend the notion of general position for points to GC-POLYGONS as follows: We say that GC-POLYGONS p_1, p_2, \dots, p_n are in general position if we can not locate any two corners belonging to GC-POLYGONS p_i and p_j with the same x - or y -coordinate. The point of site p_j that the leader c_j hits, is called *port* of site p_j (see an example in Figure 1, the port of site p_j is point pp).

Generally, the number of sites is n , the number of labels is m and the maximum number of corners of each site is k . Each label is a rectangle with four corners.

Lets give some useful definitions for the type-*opo* labelling. Consider an type-*opo* leader c which originates from port pp of a GC-POLYGON and is connected with a label on the east side (segment AB) of the rectangle at port q (see Figure 1). The line y_{pp} (containing the segment of the leader which is incident to pp and is orthogonal to side AB) divides the plane into two half-planes. We say that leader c is *oriented towards* corner A of the rectangle if port q and corner A are on the same half-plane, otherwise, we say that leader c is *oriented away* of corner A . In the case of type-*o* leader, we consider the leader to be oriented towards corner A (and also towards corner B).

This paper is structured as follows: In Section 2, we formally define the boundary labelling problem. Section 3 studies the problem of minimizing the total leader length when type-*opo* leaders are used and the labels are placed on all four sides of R . In Section 4, we study the problem of minimizing the total leader length when type-*po* leaders are used and the labels

are placed on two opposite sides of R . We conclude in Section 5 with open problems and future work.

2 The Boundary Labelling models

The boundary labelling model is a 7-tuple (*Side*, *LabelSize*, *LabelPort*, *LabelPos*, *Leader*, *Site*, *Objective*), where:

Side: Sides of the enclosing rectangle next to which we place labels. We use any sequence of N , E , W and S (for North/East/West/South). In the case of multiple stacks, we use $N_i E_j W_k S_l$ when the labels are attached to the North, East, West and South side of R and use i, j, k, l number of stacks, respectively. If no labels are placed next to a side we omit the letter corresponding to that side, and if only one stack is used we omit the index 1.

LabelSize: UnifSize (all labels have the same size), MaxSize (all labels are Uniform of Maximum Size) or NonUnifSize (each label l_i is associated with a height h_i and a width w_i)

LabelPort: FixedPorts (points where a leader can touch a label are predefined) or SlidPorts (points can slide along the label's edge)

LabelPos: FixedPos (labels have either to be aligned with a predefined fixed set of points on the boundary of the rectangle) or SlidPos (labels can slide along the rectangle's sides)

Leader: Type of the leader (*opo*, *po* or *o*)

Site: Type of the sites. Each site is a 1-point, line, rectangle, a polygon etc.

Objective: LEGAL (just find a legal label placement), *TLLM* (find a legal label placement, such that the total leader length is minimum), *TBM* (find a legal label placement, such that the total number of bends is minimum or, equivalently, the number of type-*o* leaders is maximum), *LSM* (find the maximum label size for which a legal label placement is possible), etc.

2.1 Previous Work and Our Results

All the known results on boundary labelling are given in Table 1. Most of the results were presented in (Bekos, Kaufmann, Symvonis & Wolff 2005) where the boundary labelling problem was defined. A variety of models based on the type of leader, the location of the label and the size of the label are studied for legal label placement and leader bend and leader length minimization.

In Table 2 we present the results of this paper.

3 Four-side Labelling of gc-polygons with type-opo Leaders

We consider boundary labelling with “floating” sites, such as GC-POLYGONS, rectangles and lines. According to the notation of Section 2, first we examine the Boundary Labelling(*NEWS*, UnifSize, SlidPort, FixedPos, *opo*, GC-POLYGON, *TLLM*) problem. We assume that we have fixed labels of uniform size, placed on all four sides of rectangle R , sliding ports and type-*opo* leaders. We present a polynomial time algorithm, that returns a legal labelling of minimum total leader length.

Let $P = \{p_1, p_2, \dots, p_n\}$ be the set of GC-POLYGONS and $L = \{l_1, l_2, \dots, l_m\}$ be the set of labels. Since the labels have uniform size, each site p_i can be

| Model | Time complexity |
|---|----------------------------------|
| In (Bekos, Kaufmann, Symvonis & Wolff 2005) | |
| (E, NonUnifSize, SlidPort, SlidPos, <i>opo</i> , 1-point, LEGAL) | $O(n \log n)$ |
| (E, NonUnifSize, SlidPort, SlidPos, <i>opo</i> , 1-point, TBM) | $O(n^2)$ |
| (NESW, UnifSize, SlidPort, FixedPos, <i>opo</i> , 1-point, LEGAL) | $O(n \log n)$ |
| (E, UnifSize, SlidPort, SlidPos, <i>po</i> , 1-point, LEGAL) | $O(n^2)$ |
| (EW, MaxSize, SlidPort, FixedPos, <i>opo/po</i> , 1-point, TLLM) | $O(n^2)$ |
| (EW, NonUnifSize, SlidPort, SlidPos, <i>opo</i> , 1-point, TLLM) | $O(nH^2)$ |
| (E, UnifSize, SlidPort, SlidPos, <i>s</i> , 1-point, LEGAL) | $O(n \log n)$ |
| (E/NEWS, UnifSize, SlidPort, FixedPos, <i>s</i> , 1-point, TLLM) | $O(n^{2+\delta})$, $\delta > 0$ |
| In (Bekos, Kaufmann, Potika & Symvonis 2005) | |
| (NESW, UnifSize, SlidPort, FixedPos, <i>opo</i> , 1-point, TLLM) | $O(n^2 \log^3 n)$ |

Table 1: Known results on boundary labelling. H is the height of the enclosing rectangle.

| Model | Time complexity |
|---|-------------------|
| (NEWS, UnifSize, SlidPort, FixedPos, <i>opo</i> , GC-POLYGON TLLM) | $O(n^2 \log^3 n)$ |
| (NEWS, UnifSize, SlidPort, FixedPos, <i>opo</i> , Rectangle/Line, TLLM) | $O(n^2 \log^3 n)$ |
| (EW, UnifSize, SlidPort, FixedPos, <i>po</i> , GC-POLYGON TLLM) | $O(n^2 \log^3 n)$ |
| (EW, UnifSize, SlidPort, FixedPos, <i>po</i> , Rectangle/Line, TLLM) | $O(n^2 \log^3 n)$ |

Table 2: The results presented in this paper.

connected to any label l_j . We seek to connect each site p_i to a label l_j and to specify two points one on the periphery of p_i (port of site p_i) and one on the periphery of label l_j (port of label l_j).

We propose Algorithm 1 for this problem.

Algorithm 1: 4SIDE-AREA-OPO

input : A set of n GC-POLYGONS p_i in the plane and a set of m uniform sized labels l_j .

output: A crossing free four-side type-*opo* labelling of minimum total leader length.

Step A. Shortest Leader Computation:

Construct a complete weighted bipartite graph $G = (P \cup L, E, w)$ between all sites $p \in P$ and all labels $l \in L$. The weight of an edge $(p_i, l_j) \in E$ is the Manhattan length of the *shortest* (under the Manhattan metric) leader, say d_{ij} , which connects p_i with l_j .

Step B.

Proceed by applying to graph G , Vaidya's algorithm (Vaidya 1989) for minimum-cost bipartite matching under the Manhattan metric. It computes a matching between sites and labels that minimizes the total Manhattan distance of the matched pairs.

Step C. Obtain a labelling M as follows:

If an edge $(p_i, l_j) \in E$ is selected in the matching **then** connect site p_i to label l_j with a leader of length d_{ij} .

Step D. Crossing Free Procedure:

Eliminate all crossings of leaders and obtain a crossing free labelling M' .

3.1 Shortest Leader Computation

We propose Algorithm 2 for computing the minimum Manhattan distance between every site and every label (Step A of Algorithm 1).

Theorem 1 *Algorithm 2 computes the minimum distance under the Manhattan metric between any label and any polygon, when the labels are placed in fixed positions on all four sides of rectangle R . Moreover this algorithm runs in $O(n(k' + m) \log k')$ time, where m is the number of labels, n is the number of GC-POLYGONS, $k' = O(k + m)$ and k is the maximum number of corners that a site of type GC-POLYGON can have.*

Proof: The number of points in each set p_i^e is $O(m + k)$ and each set p_i^e is computed in $O(n(m + k))$ time. Note that set p_i^e contains candidate site ports. In Step 1 we construct the Voronoi diagram under the Manhattan distance of the set $p_i \cup p_i^e$ (k' points total), where $k' = O(k + m)$. The construction of the Voronoi diagram can be done in $O(k' \log k')$ time (Lee 1980).

Algorithm 2: MINIMUM MANHATTAN DISTANCE BETWEEN ANY GC-POLYGON AND ANY LABEL.

input : A set of m labels placed in fixed positions on all four sides of rectangle R and a set of n GC-POLYGON sites in the plane, with their corners indexed clockwise.

output: The minimum Manhattan distance between any GC-POLYGON and any label.

Step A.

for each site p_i ($1 \leq i \leq n$) **do**
for each label site l_j ($1 \leq j \leq m$) **do**
find the crossing points of each edge of p_i with the vertical (horizontal) lines of each corner of label l_j . Add these points to p_i^e .

Step B.

for each site p_i ($1 \leq i \leq n$) **do**
1. Construct the Voronoi diagram, under the Manhattan distance, H_i for $p_i \cup p_i^e$.
2. **for** each label l_j ($1 \leq j \leq m$) **do**
for each corner of l_j find the nearest neighbor in H_i (Voronoi diagram) and compute their Manhattan distance. Set d_{ij} to be the minimum distance and pp_{ij} the port of p_i for this distance.

Finding the nearest neighbor of a point q in the Voronoi diagram H_i costs $O(\log k')$. Therefore, we compute Step B.2 in $O(m \log k')$ time. Totally the running time of Algorithm 2 is $O(n(k' + m) \log k')$. \square

If the polygons are convex, then we can find the minimum Manhattan distance between any label and any convex polygon faster by using Algorithm 3.

Algorithm 3: MINIMUM MANHATTAN DISTANCE BETWEEN ANY CONVEX POLYGON AND ANY LABEL.

input : A set of m labels placed in fixed positions on all four sides of rectangle R and a set of n convex GC-POLYGONS sites in the plane, with their corners indexed clockwise.

output: The minimum Manhattan distance between any convex GC-POLYGON and any label.

for each site p_i ($1 \leq i \leq n$) **do**
 for each side (West | North | East | South) **do**
 1. take the south | west | north | east - most label of that side, say l_j
 2. compute the minimum distance of l_j to all corners $1, \dots, k_i$ and edges of p_i . Keep the minimum distance in d_{ij} , and the point pp_{ij} of p_i for which this minimum was achieved.
 3. **while** not all labels of the West | North | East | South side have been examined **do**
 i) $pp := pp_{ij}$; take the next label in clockwise direction say $l_{j'}$
 ii) compute the minimum distance of label $l_{j'}$ to pp , the corners that are between pp and the north | east | south | west - most corner of p_i in clockwise direction, and the edges that have these corners as endpoint. keep the minimum distance in $d_{ij'}$, and the point $pp_{ij'}$ of p_i for which this minimum was achieved.

Theorem 2 *Algorithm 3 computes the minimum distance under the Manhattan distance between any label and any convex GC-POLYGON when the labels are placed in fixed positions in all four sides of rectangle R . Moreover this algorithm runs in $O(n(m + k))$ time, where m is the number of labels, n is the number of sites and k is the maximum number of corners that a site of type GC-POLYGON can have.*

Proof: In each side we compute the minimum distance for the first pair of label site in $O(k)$ time (Step (2)). Recall that our sites are convex and therefore we can determine the minimum Manhattan distance without checking all corners and edges of a site to a label, just by finding the first point of the convex site where the distance starts again to increase. In the computation of the minimum distance between site p_i and label $l_{j'}$ (Step 3.(ii) of Algorithm 3), we need to examine only the part of the site that lies in between the port of the previous label and the north (east | south | west) most corner, because label $l_{j'}$ lies between the previous label and the north | east | south | west -side of rectangle R in clockwise direction. This step requires $O(m + k)$ time. Totally the required time is $O(n(m + k))$. \square

3.2 Crossing Free Procedure

After Step (C) of Algorithm 1 some leaders may cross, thus we have not yet a legal label placement.

Lemma 1 *Labelling M of Algorithm 1 is of minimum total leader lengths with some crossings. Let c_i and c_j be two leaders that cross each other. Then the following holds (i) the labels of these leaders are on adjacent sides of the rectangle R and the sides are incident to a corner A and (ii) leaders c_i and c_j are oriented towards corner A of the rectangle R and can be rerouted so that they do not cross each other with unchanged total leader length.*

Proof:

Prove of (i):

We show that it is impossible to have the labels placed at the same or opposite sides of R . Suppose that the labels lie on the same side, say the east side, and the leaders intersect or overlap.

If leaders c_i and c_j overlap, then we can slide on of the two leaders, by choosing new site port and probably new label port. Then either the total leader length is reduced (see an example in Figure 3), which is a contradiction, or it remains the same. Note that in the latter case, in a GC-POLYGON we can use a new leader with site port a point that is next to the site port of the old leader, so that the new leader length is equal to the old leader length.

If the leaders c_i and c_j intersect, then the intersection takes place outside rectangle R (in the track routing area). This implies that, along the east side, the order of the sites is the reverse of the order of their associated labels. However, by swapping the labels to which each site is connected, we can reduce the total leader length (and also eliminate the crossing), a contradiction since we assumed that the total leader length of the labelling is minimum (see an example in Figure 4).

Consider now the case where the labels lie on opposite sides of rectangle R . Then, since the leaders intersect each other, the segments of the leaders which are inside the rectangle (and incident to the sites) have to overlap. Again, if the leaders overlap then by swapping the labels to which each site is connected, we can reduce the total leader length (and also eliminate the overlapping), a contradiction since we assumed that the total leader length of the labelling is minimum (see an example in Figure 5).

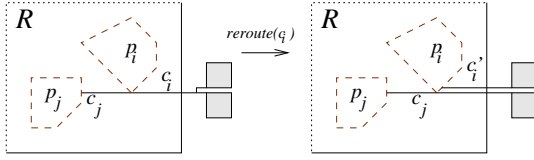
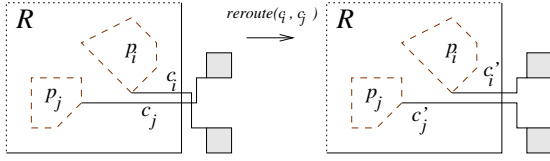
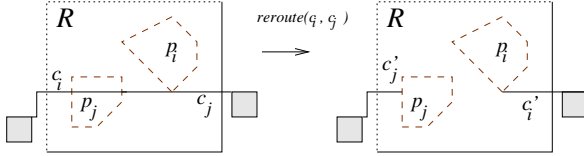
We showed that leaders which have their associated labels lying on the same or on opposite sides of rectangle R can not cross and therefore, if we find two crossing leaders, their associated labels must lie on adjacent sides of the rectangle R .

Prove of (ii):

Lets say that leaders c_i and c_j are oriented towards corner A . First we show that in a labelling of minimum total leader length, it is impossible to have one (Case (a)) or both leaders oriented away of corner A (Case (b)). We proceed to consider these two cases.

Case a: Exactly one leader, say c_j , is oriented away of corner A . This case is described in the left of Figure 6. Rerouting the leaders as described in Figure 6 results in a reduction of the total leader length, a contradiction since we assumed that the total leader length of the labelling is minimum.

Case b: Both leaders c_i and c_j are oriented away of corner A (see left of Figure 7). When both leaders are oriented away of corner A , rerouting results in higher reduction of the total leader length, compared to Case (a) where only one leader was oriented away of corner A . The rerouting of the leaders is shown in Figure 7.

Figure 3: Leaders c_i and c_j overlapFigure 4: Leaders c_i and c_j intersectFigure 5: Leaders c_i and c_j overlap

Having eliminated the cases (a) and (b) where one or both crossing leaders are oriented away of corner A, the only case left is the one where both leaders are oriented towards corner A (see an example in Figure 8).

Case c: Leaders c_i and c_j , that are oriented towards the same corner, say A, can be rerouted (see Figure 8) so that they do not cross each other and the sum of their leader lengths remains unchanged. Partition the first segment of each leader c_i and c_j into two sub-segments in their crossing point. Then obtain the new leaders c'_i and c'_j by a sliding the (sub)segments of leaders c_i and c_j , leaving their sum unchanged. To complete the proof of the lemma, we note that whenever we perform a rerouting, we never change the position of a label or site port. So, since the used port would also be available in the case where the fixed-port model is used, the lemma applies to fixed (label) ports, as stated.

If we had a reduction of the total leader length, by taking as site or label ports other points, this would be a contradiction since we assumed that the total leader length of the labelling is minimum.

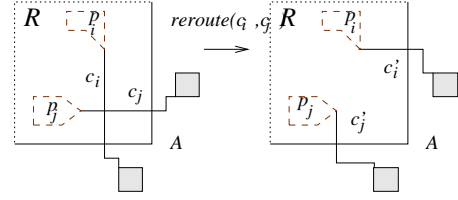
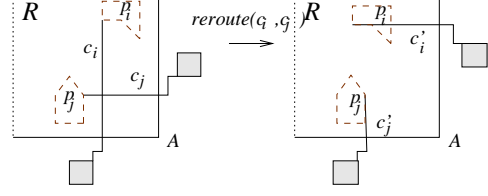
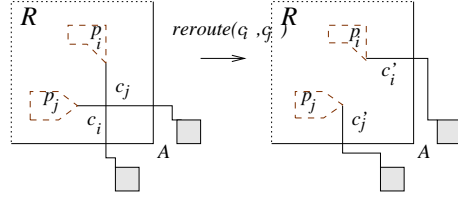
□

We proceed by showing that given a labelling of minimum total leader length which may contain crossings, we can efficiently construct a crossing free labelling of identical total leader length, by first identifying such a crossing and then eliminating the crossing (with the help of Lemma 1.ii).

Theorem 3 *Step D of Algorithm 1 produces a crossing free labelling M' of minimum total leader length in $O(n \log n)$ time.*

Proof: We can resolve all crossings (as in (Bekos, Kaufmann, Potika & Symvonis 2005)) in $O(n \log n)$ additional time and keep the total leader length unchanged. We show how to eliminate all crossings in labelling M by rerouting the crossing leaders. Our method performs two passes over the sites, one in the west-to-east and one in the east-to-west direction.

Consider first the west-to-east pass. In the west-to-east pass of labelling M , we consider all sites with labels on the east side of the rectangle. We examine the sites from west-to-east and we are interested only on those who have crossing leaders. Let p_i be the

Figure 6: Case (a): Leader c_j is oriented away of corner A and leader c_i is oriented towards corner A.Figure 7: Case (b): Both leaders c_i and c_j are oriented away of corner A.Figure 8: Case (c): Both leaders c_i and c_j are oriented towards corner A.

west-most such site and let c_i be the leader that connects it with its corresponding label on the east side of the rectangle (see Figure 9). Lemma 1(i) implies that leader c_i intersects only with leaders that are connected with labels on the north and south sides of rectangle R . Without loss of generality, assume that c_i is oriented towards the east-south corner of the rectangle, say A. Then all leaders that intersect c_i have their labels on the south side of R and are also oriented towards A (by Lemma 1(ii)). Let c_k be the west-most leader that intersects c_i , and let p_k be its incident site. According to Lemma 1(ii), we can reroute leaders c_i and c_k so that the total leader length remains unchanged (Figure 10). The total number of crossings is reduced and the next west-most site with intersecting leader and connected to a label on the east side of the rectangle is located to the east of site p_i . Finally, all west-to-east crossings are eliminated. It is also impossible to introduce a east-to-west crossing when the west-to-east pass is executed, as an example see leaders c'_i and c in Figure 11. Both leaders c'_i and c must be oriented towards corner B (by Lemma 1(i)), a contradiction since leader c'_i is oriented away of corner B (and towards corner A).

After the west-to-east and the east-to-west pass, we obtain a labelling M' without any crossings and of total leader length equal to that of M , that is, minimum.

By employing a dynamic priority search tree based on half-balanced trees [7, pp. 209] we can answer question of the form: 'given a point p return the segment that intersects line y_p and is the nearest in the east side of p ', insert and delete operations in $O(\log n)$ time. Thus, identifying the (at most n) pairs of leaders to be rerouted during the west-to-east pass takes only $O(n \log n)$ time, resulting to a total time complexity of $O(n \log n)$ for the production of the crossing free boundary labelling M' .

□

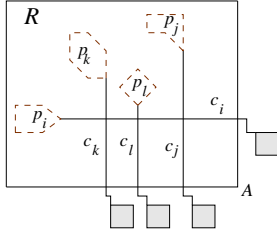


Figure 9: A west-to-east pass. First the crossing of leaders c_i and c_k must be eliminated.

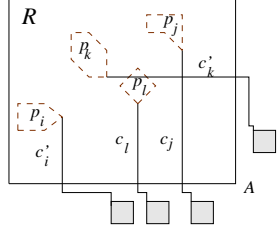


Figure 10: Rerouting used to eliminate crossings in an type *opo*-labelling.

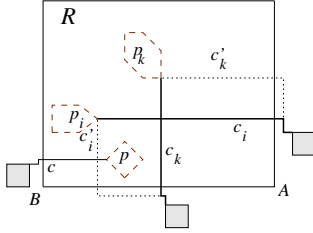


Figure 11: We can not introduce any east-to-west crossing during the west-to-east pass (the new leaders are with dotted lines).

Theorem 4 Algorithm 1 solves problem *Boundary Labelling(NEWS, UnifSize, SlidPort, FixedPos, opo, GC-POLYGON, TLLM)* in $O(n^2 \log^3 n)$ time.

Proof: Let L be the set of the n labels (we assume that the number of labels is equal to the number of sites). We assume that all labels are around the boundary of the rectangle. We construct a complete bipartite graph between all sites $p_i \in P$ and all labels $l_j \in L$, with edge weights to be the Manhattan length d_{ij} of the corresponding leaders computed with the help of Algorithm 2 (or Algorithm 3). This step costs $O(n(k+n) \log(k+n))$ time, where k is the maximum number of corners that a site of type GC-POLYGON can have (Theorem 1). Or $O(n(n+k))$ time (Theorem 2).

In Step B we proceed by applying the Vaidya's algorithm (Vaidya 1989) for minimum-cost bipartite matching for points in the plane under the Manhattan metric. It runs in $O(n^2 \log^3 n)$ time and finds a matching between sites and labels that minimizes the total Manhattan distance of the matched pairs.

The leaders in the produced solution might cross. However, in Step D based on Theorem 3 we can obtain a crossing free solution in $O(n \log n)$ additional time. \square

Since a rectangle is a GC-POLYGON with only four corners and a line-segment of two corners, we have:

Corollary 2 Problem *Boundary Labelling(NEWS, UnifSize, SlidPort, FixedPos, opo, rectangle/line, TLLM)* can be solved in $O(n^2 \log^3 n)$ time.

Remark Notice that Algorithm 1 works for any kind of polygon. We choose the special kind of GC-

POLYGONS because these polygons offer better visualization and because one can easily find an alternative leader, in the case of overlapping leaders.

3.3 Sample Labelling of type-*opo* Leaders

Figures 12 and 13 depict the regions of Germany and a boundary labelling on opposite sides (east and west) of R with type-*opo* leaders. Figure 12 is produced by the algorithm for boundary labelling points and provides an optimal solution. The labelling of Figure 12 is visually improved in Figure 13 by replacing the points with rectangles within each region. The labelling of Figure 13 is optimal, with the use of less total leader length (37% less pixels) than Figure 12. Note that we achieved to reduce the number of leader bends to 5 (in Figure 13) from 8 (in Figure 12), just by the use of rectangle sites instead of points.

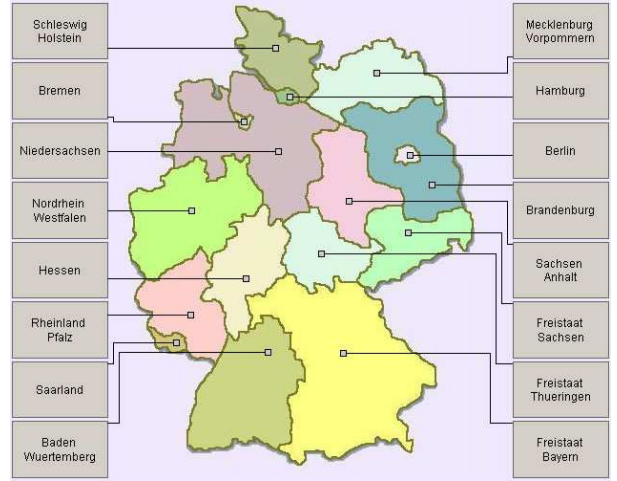


Figure 12: A regional map of Germany; a point is the representative of each region.

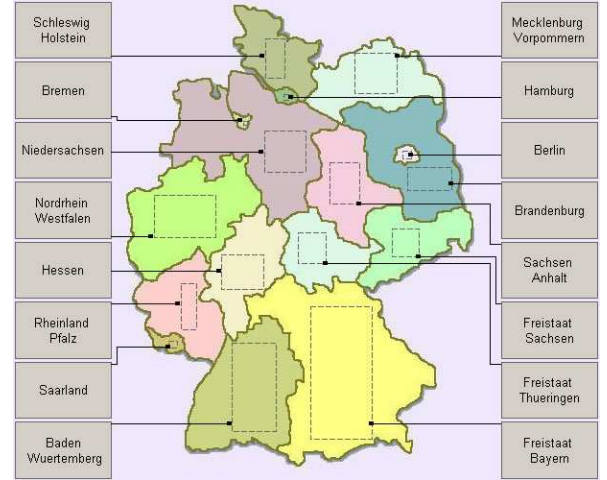


Figure 13: A visually improved map; a rectangle is the representative of each region.

4 Two-side Labelling of gc-polygon sites with type-*po* Leaders

We adopt the same scenario as in Section 3, assuming type-*po* leaders and 2-side labelling. According to the notation of Section 2, we examine the Boundary Labelling(EW, UnifSize, SlidPort, FixedPos, *po*, GC-POLYGON, TLLM) Problem. We suppose that we have fixed labels of uniform size placed on two opposite sides of rectangle R , sliding ports and type-*po*

leaders. Again, our objective is to minimize the total leader length.

To deal with this problem, we use the (matching based) Algorithm 1 for the case of type-*opo* leaders to get a label placement of minimum total leader length. As already mentioned, this can be done in $O(n^2 \log^3 n)$ time (Theorem 4). Instead of placing type-*opo* leaders we use type-*po* leaders. Note that connecting a site to its label with a type-*opo* or a type-*po* leader requires the same leader length under the Manhattan metric, assuming that we keep the same ports. Therefore, the returned solution remains optimal, but might contain crossings.

Crossings of leaders oriented towards the same side can occur, however, they can be resolved following a similar strategy as in (Bekos, Kaufmann, Symvonis & Wolff 2005), without changing the total leader length. This can be done in $O(n^2)$ additional time. Crossings of leaders to opposite sides cannot occur, since swapping the labels of the sites that have crossing leaders, would result in a solution with smaller total leader length. This is a contradiction, because we assumed that the original solution minimizes the total leader length.

The same strategy can be applied when labels are placed in only one side of rectangle R or when the sites are line segments. However, this strategy does not result in a crossing free solution, in the case where labels are placed on all four sides of rectangle R , since a crossing between two leaders that are oriented towards two adjacent sides of the enclosing rectangle can not always be eliminated. We conclude:

Corollary 3 *Problem Boundary Labelling(EW, UnifSize, SlidPort, FixedPos, po, rectangle/line, TLLM) can be solved in $O(n^2 \log^3 n)$ time.*

4.1 Sample Labelling of type-*po* Leaders

Figures 14 and 15 depict labelling of the regions of France with type-*po* leaders. Restricting the sites to by rectangles leads to labelling with reduced total leader length than the use of points (10.73% less pixels). The order of the labels is the same as the order of the correspondent sites in Figure 15, in contrast to Figure 14. Note also that we achieved to reduce the number of leader bends to 1 (in Figure 15) from 14 (in Figure 14).

5 Open Problems and Future Work

We have presented results for boundary labelling of GC-POLYGONS, rectangle or lines instead of points with the objective of minimizing the total leader length. It is interesting to see the visual result of the labelling if we choose another objective as the one of minimizing the number of bends of the leaders.

Another future research is to find efficient ways to determine for each region of a map a representative GC-POLYGON site that has less than k corners.

References

- Bekos, M. A., Kaufmann, M., Potika, K. & Symvonis, A. (2005), Boundary labelling of optimal total leader length., in P. Bozanis & E. N. Houstis, eds, 'Panhellenic Conference on Informatics', Vol. 3746 of *Lecture Notes in Computer Science*, Springer, pp. 80–89.
- Bekos, M. A., Kaufmann, M., Symvonis, A. & Wolff, A. (2005), Boundary labeling: Models and efficient algorithms for rectangular maps, in J. Pach, ed., 'Proc. 12th Int. Symposium on

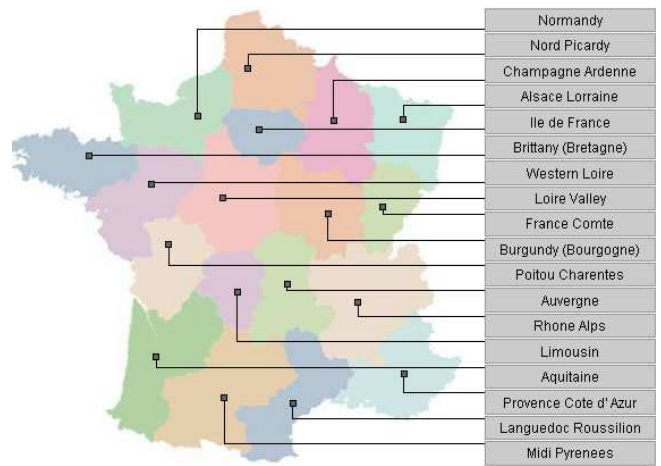


Figure 14: A regional map of France; a point is the representative of each region.

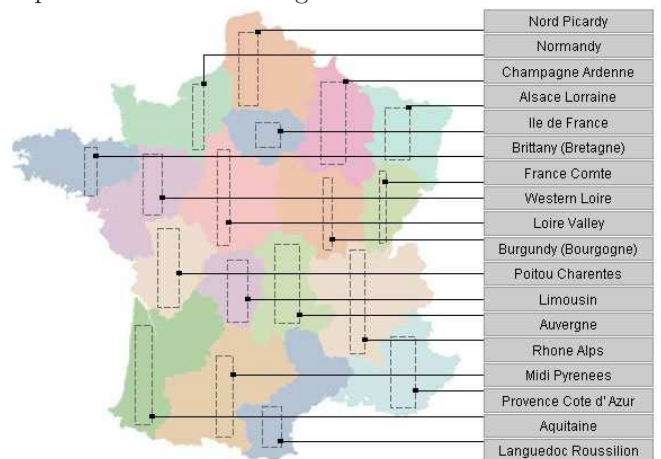


Figure 15: A visually improved map; a rectangle is the representative of each region.

Graph Drawing (GD'04)', Lecture Notes in Computer Science, pp. 49–59.

Bekos, M. A. & Symvonis, A. (2005), Bler: A boundary labeller for technical drawings., in 'Graph Drawing, LNCS'.

Formann, M. & Wagner, F. (1991), A packing problem with applications to lettering of maps, in 'Proc. 7th Annual ACM Symposium on Computational Geometry (SoCG'91)', pp. 281–288.

Lee, D. T. (1980), 'Two-dimensional voronoi diagrams in the lp-metric', *J. ACM* **27**(4), 604–618.

Strijk, T. & van Kreveld, M. (1999), 'Labeling a rectilinear map more efficiently'.
*citeseer.ifi.unizh.ch/article/strijk99labeling.html

Vaidya, P. M. (1989), 'Geometry helps in matching', *SIAM Journal on Computing* **18**, 1201–1225.

Wolff, A. & Strijk, T. (1996), 'The Map-Labeling Bibliography', <http://i11www.ira.uka.de/map-labeling/bibliography/>.
*<http://i11www.ira.uka.de/map-labeling/bibliography/>

Visualization of multi-dimensional data of bioactive chemicals using a hierarchical data visualization technique "HeiankyoView"

Takayuki ITOH* Fumiyoshi YAMASHITA**

*Department of Information Sciences, Ochanomizu University

**Graduate School of Pharmaceutical Sciences, Kyoto University

* itot@computer.org

** yama@pharm.kyoto-u.ac.jp

Abstract

In the age of combinatorial chemistry and high throughput screening, large-scale data of bioactive chemicals oriented to drug development are being accumulated. Due to the difficulties inherent in understanding such large quantities of data, information visualization techniques are increasingly attractive. Authors apply "HeiankyoView", which is the technique for the representation of large-scale hierarchical data, for the visualization of multi-dimensional data of bioactive chemicals. In the present study, we investigated applicability of the visualization technique to the structure-activity relationship (SAR) analyses. The study first classifies chemicals according to similarity in their biological actions through self-organizing map analysis. It then applies a recursive partitioning method to find the relationship between biologically based categories and chemical structure, and finally it stores the drugs as hierarchical data. HeiankyoView is suitable for the visualization of such hierarchical data. This paper first describes the algorithmic overview of HeiankyoView, and then provides some example of visualization of multi-dimensional data of bioactive chemicals.

Keywords: Visualization, Hierarchical data, Rectangle packing, Combinational chemistry.

1 Introduction

Drug discovery and development is costly, time consuming, and high risk activity. The process starts with the discovery of a chemical or class of chemicals with particular biological activity. Lead compounds must then be identified, optimized, and only then tested in preclinical animal studies for efficacy, toxicity, etc. Those bioactive chemicals still considered viable after such rigorous scrutiny are then brought to human subjects for clinical evaluation of a variety of aspects of the chemical, including safety, effectiveness, and dosage determination.

Recent evolution of combinatorial chemistry and high-throughput screening technologies has brought exhaustive research of exploration of drugs. Large populations of chemical compounds are synthesized

combinatorially, using sets of chemical "building blocks", and then subjected to automated biological assay systems. In vitro techniques of evaluating the metabolism and toxicity of drug candidates, which are major reasons of withdrawal of preclinical and clinical drug development, is also developing. These innovations have enabled collection of multi-dimensional and large-scale chemical compound data for drug discovery. Information technologies are potential to assist the selection and optimization of lead compounds and the selection of drug candidates; however, it is a very complicated problem to optimize the multi-dimensional large-scale data and design high-quality compound libraries. Information visualization techniques should be useful to discover characteristics from such multi-dimensional and large-scale data.

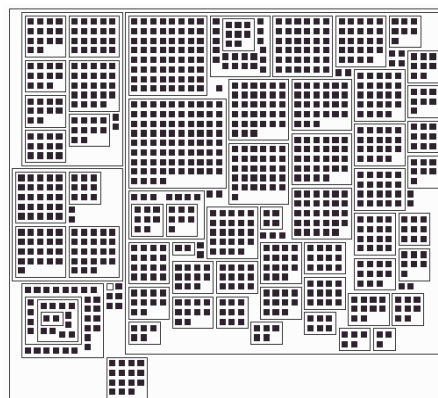


Figure 1. Example of hierarchical data visualization by HeiankyoView.

This paper presents the visualization of multi-dimensional data of bioactive chemicals by applying the hierarchical data visualization technique "HeiankyoView". As shown in Figure 1, HeiankyoView visualizes the hierarchical data by mapping leaf-nodes as painted square icons, and non-leaf-nodes as rectangular borders. The technique targets to represent all leaf-nodes of large-scale hierarchical data in one display space without any focus-and-context techniques. One of authors has already applied the technique to visualize various data, including access trend of Web sites [Yam02], jobs in distributed computing environments [Yam03], distribution of network intrusion detection data [Ito05], and so on. These applications proof that HeiankyoView is useful not only for the overview of large-scale hierarchical data, but also discovery of interesting but minor local characteristics, and exploration of detailed local information.

Cytochrome P450 (CYP) enzymes are a super-family of drug metabolizing enzymes that extensively affect the elimination of drugs from the body. Competitive interaction of simultaneously administered drugs with CYPs [Gue97] or genetic polymorphism of CYPs [Ing04] increase the drug level in the body unexpectedly, and sometimes cause lethal adverse reactions. Therefore, to characterize the interaction of investigational compounds with CYPs is an important issue in their safety evaluations. In the present study, authors experiment the visualization for the structure-activity relationship (SAR) analyses of CYP-related metabolism. Susceptibility of 161 drugs to major five CYP isoforms (i.e., 1A2, 2C9, 2C19, 2D6, and 3A4) was analyzed. Of more than 40 CYPs encoded in the human genome, only the five CYP isoforms account for 95% of hepatic drug metabolism [Ren97]. By understanding what molecular structural attributes relate to substrate specificity of each CYP isoform, design of molecules or their libraries is more effective. The drugs were classified into 6 groups according to their metabolic susceptibility to the CYP isozymes through self-organizing map analysis. Next, a recursive partitioning method was applied to find the relationship between metabolic susceptibility profile and chemical structure, and it finally stores the drugs as hierarchical data. Heiankyoview is useful for the visualization of such large-scale hierarchical data.

The reminder of this paper is as follows. Section 2 summarizes requirements for the visualization of multi-dimensional data of bioactive chemicals, and introduces existing hierarchical data visualization techniques. Section 3 introduces the algorithm of Heiankyoview. Section 4 describes about experiments of visualization of combinational chemistry data and some results. Section 5 summarizes this study and discusses about the future works.

2 Required visualization techniques

This section first summarizes requirements for the visualization of multi-dimensional data of bioactive chemicals, and then introduces well-known information visualization techniques related to Heiankyoview.

2.1 Requirements

Heiankyoview is the hierarchical data visualization that satisfies the following requirements, which are definitely preferable for the applications described in Section 1 [Yam02][Yam03][Ito05], and also for the visualization of large-scale multi-dimensional data of bioactive chemicals.

[Requirement 1: No overlaps between leaf-nodes] Some visualization methods may cause overlap of leaf-nodes in defocused regions. Heiankyoview does not overlap them, so it provides a uniform overview of the data. It is also useful for detail-on-demand user interface, because it lets every leaf-node as clickable metaphor.

[Requirement 2: Efficient use of display spaces] It is often useful if visualization techniques pack all data items in a limited display space to provide a good overview. Traditional orthogonal tree-based systems, such as well-known file system viewers, have a bottleneck in that

they may need a large display space if there are many nodes under one non-leaf node, or if there is a deep hierarchy.

[Requirement 3: Aspect ratio of subspaces] When visualization techniques subdivide a display space to represent the parts of the given data, squarish subspaces are usually preferable over thin subspaces so that users can visually recognize the parts. Therefore, aspect ratios of subspaces should be considered.

[Requirement 4: Flexible placement of arbitrarily shaped nodes] When data items are represented as rectangular icons, we often assume that the aspect ratios and sizes of all icons should be entirely unified. It is especially preferable if applications require representing all leaf-nodes equally. On the other hand, we also assume that the aspect ratios and sizes of all the icons should be specifiable by users, and they may even be varied. This makes it possible to visually emphasize important data items.

2.2 Related techniques

This section introduces well-know related information visualization techniques.

2.2.1 Space-filling hierarchical data visualization techniques

The technique subdivides display spaces to represent each portion of hierarchical data. TreeMaps [Joh91] recursively subdivides the display spaces into rectangular regions to form nested bar charts. Another technique subdivides the display space into sectors to form nested pie charts [Chu98]. Heiankyoview is somewhat analogous to TreeMaps because both techniques subdivide display spaces into rectangular area.

Variation improved TreeMaps have been recently proposed. Squarified Treemap [Bru00] subdivides display spaces into rectangles as much as square. Ordered Treemap [Shn01] subdivides display spaces and assigns the subregions in the predefined order of leaf-nodes. Quantum Treemap [Bed02] applies modified Squarified or Ordered Treemap so that it can represent leaf-nodes that are equally shaped and sized. Target of the Quantum Treemap is very similar to Heiankyoview, and actually Quantum Treemap satisfies requirements described in Section 2.1. Experiments described in [Ito04] discusses trade-offs between Quantum Treemap and Heiankyoview.

2.2.2 Other hierarchical data visualization techniques

Tree visualization techniques are other well-known hierarchical data visualization techniques. Hyperbolic Tree [Lam96], Cone Tree [Car95], and Fractal Views [Koi95] provides navigation and exploration capabilities for large-scale hierarchical data. Space-filling techniques have advantages against tree visualization techniques in the view of requirements 1 and 2.

Heiankyoview represents hierarchical data as two-dimensional nested metaphor. Three-dimensional nested metaphor has been applied by Information Cube [Rek93] and H-BLOB [Spr00]. These techniques require

capability and skill of 3D graphics, and adjustment of semi-transparency may be difficult for large-scale data visualization.

3 Hierarchical data visualization technique “HeiankyoView”

This section describes the hierarchical data visualization technique “HeiankyoView”.

3.1 Overview

The technique represents leaf-nodes of hierarchical data as square icons, and non-leaf-nodes as nested rectangular borders. Figure 2 denotes the processing order of display layout of the nodes. The technique first places the leaf-nodes in the lowest-level of the hierarchy onto a display space, and represents a non-leaf-node by enclosing the leaf-nodes. The technique then places leaf-nodes or non-leaf-nodes in a higher level, and again encloses them by another rectangle. The technique represents whole the hierarchical data by repeating the similar process until it arrives at the highest level of the hierarchy.

Assuming the square icons represents leaf-nodes as rectangles, the visual items of the hierarchical data can be treated as a collection of rectangles. In other words, the technique solves the rectangle packing problem satisfying the following conditions, starting the lowest level, toward the top of the hierarchy:

[Condition 1] Rectangles never overlap each other.

[Condition 2] Area of rectangular region enclosing the placed rectangles is to be minimized.

[Condition 3] Aspect ratio of rectangular region enclosing the placed rectangles is to be optimized.

[Condition 4] If ideal positions of the rectangles are given, the rectangles are to be placed where enough close to the ideal positions. (This condition is ignored in this paper, but often the condition is quite important.)

The technique places the rectangles onto the display space one-by-one, selecting each position of the rectangles from multiple candidate positions. Original rectangle placement algorithm [Ito04] calculates the candidates by referring Delaunay triangular mesh connecting the centers of previously placed rectangles, but the further improved rectangle placement algorithm has been briefly reported [Ito03]. Authors measured the rectangle layout results and computation time of the proposed algorithm [Ito03] and proofed that it improved the results from the previous algorithm [Ito04]. This section describes the detail of the improved rectangle placement algorithm.

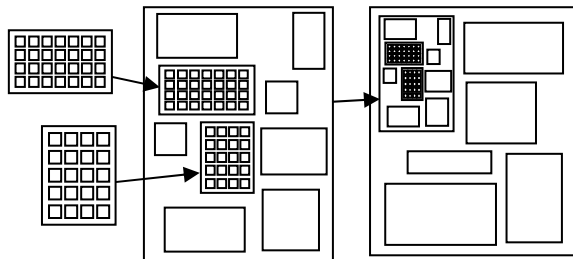


Figure 2. Processing flow of HeiankyoView.

3.2 Rectangle Placement algorithm

The visualization technique places rectangles in one level of the hierarchical data by the following processing order:

1. Select a rectangle R.
2. Calculate multiple candidate positions for R.
3. Repeat the following process for each candidate position, by the trial placement of R there:

(a) If the candidate does not satisfy [condition 1], then select another candidate and repeat the similar process.

(b) Otherwise, calculate the penalty value for other conditions, and record the candidate if the penalty value is smaller than any of penalty values of previously processed candidates.

4. Place R at the recorded candidate, and update the data structure for the placement of rectangles.

3.2.1 Definition of rectangles

The technique assumes that rectangles are placed onto a 2D display space represented as x-y orthogonal coordinates, and all edges of rectangles are parallel to x- or y- axis. This section describes parameters of a rectangle as follows;

w : width of a rectangle

h : height of a rectangle

x : x coordinate value of the center of a rectangle

y : y coordinate value of the center of a rectangle

u : x coordinate value of the ideal position of a rectangle

v : y coordinate value of the ideal position of a rectangle

Here the input and output of the rectangle placement algorithm is defined as follows:

Input: w and h .

Optional input: u and v .

Output: x and y .

Here we assume that w and h values of leaf-nodes are given by users, but w and h values of non-leaf-nodes are calculated when child-level of the non-leaf-node is processed. u and v are not used in this paper.

3.2.2 Definition of grid-like subdivision of display spaces

As shown in Figure 3, the proposed algorithm subdivides a display space by using extensions of edges of previously placed rectangles. The technique manages of occupancy of display space using the grid-like subdivision. Painted area in Figure 3(right) has been already occupied by a rectangle, and white area has not been yet occupied by a rectangle.

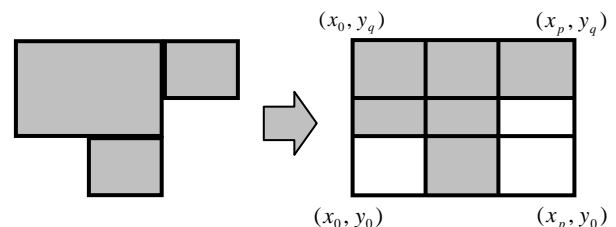


Figure 3. (Left) Already placed rectangles. (Right) Grid-like subdivision of display space by extension lines of edges of the placed rectangles.

Let us assume that the display space is divided into p along x-axis, and q along y-axis. Here the paper represents

the subdivision results as follows:

- x-coordinate values of $(p+1)$ extension lines parallel to y-axis as x_0 to x_p .
- y-coordinate values of $(q+1)$ extension lines parallel to x-axis as y_0 to y_q .

The paper also denotes a subdivided grid area, enclosed by extension lines $x=x_s$, $x=x_{s+1}$, $y=y_t$, and $y=y_{t+1}$, as $[s,t]$. Each grid area has a boolean value that denotes if it has been already occupied by a rectangle or not.

3.2.3 Calculation of candidate positions

Assume that $i-1$ rectangles has been placed onto a display space, and the display space is divided into p by q grid subspaces. Here the technique attempts to place the i -th rectangle by calculating candidate positions inside the subspaces, and decide the best position to place the rectangle.

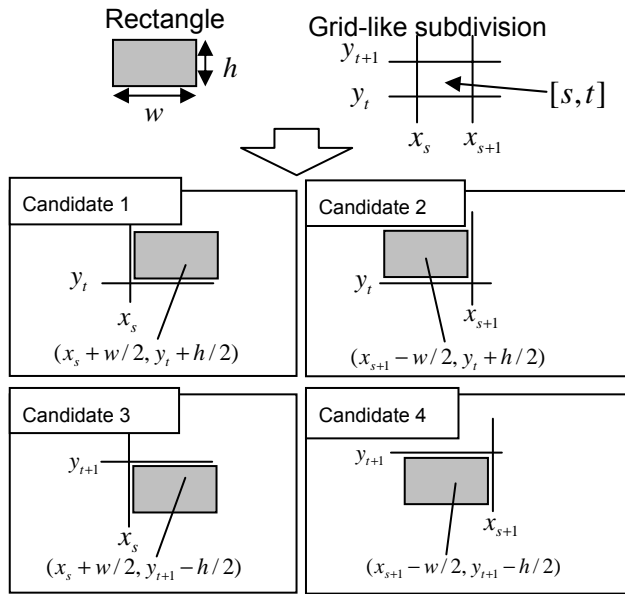


Figure 4. Four candidate positions for placing a rectangle.

As defined in Section 3.2.2, suppose x-coordinate values of the corners of the subspace $[s,t]$ as x_s and x_{s+1} , and y-coordinate values of the corners of the subspace $[s,t]$ as y_t and y_{t+1} . The technique first checks if the subspace $[s,t]$ has been already occupied by a previously placed rectangle or not. If the subspace has not been occupied yet, the technique attempts to place the center of the i -th rectangle at the following four candidate positions:

Candidate 1: $((x_s + w/2), (y_t + h/2))$

Candidate 2: $((x_{s+1} - w/2), (y_t + h/2))$

Candidate 3: $((x_s + w/2), (y_{t+1} - h/2))$

Candidate 4: $((x_{s+1} - w/2), (y_{t+1} - h/2))$

As shown in Figure 4, the candidates are the positions that a corner of the i -th rectangle locates at a corner of the subspace.

3.2.4 Penalty values for candidate positions

Placing the i -th rectangle at each of the candidate positions, the technique first checks if the rectangle overlaps with any of previously placed rectangles. Supposing a candidate position as (x, y) , the technique specifies the integer values j, k, l, m . As shown in Figure 5(left), these

values satisfy the following inequation:

$$\begin{aligned} x_j & x-w/2 & x_{j+1} \\ x_k & x+w/2 & x_{k+1} \\ y_l & y-h/2 & y_{l+1} \\ y_m & y+h/2 & y_{m+1} \end{aligned}$$

The technique then checks the boolean values of the subspaces $[j,l]$ to $[k,m]$, to check if any of the subspaces has been already occupied by a rectangle or not. If at least one subspace has been occupied, the technique gives up the candidate and deals with the next candidate.

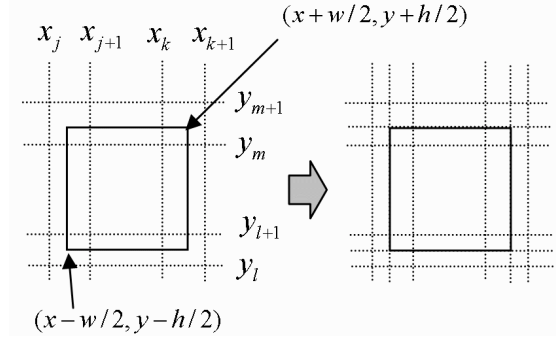


Figure 5. (Left) Integer values j, k, l, m and coordinates of vertices of a rectangle. (Right) Subdivision of grid.

The technique then calculates the penalty value for the candidate position using the equation (1), and decides to place the rectangle where the penalty value is minimum:

$$aD + bS + cA + dT \dots (1)$$

where,

a, b, c, d : User-defined constant value,

D : Distance between the ideal position (u, v) of the rectangle and the position of the candidate (x, y) . D is ignored (treated as zero) in this paper.

S : Extension of area of the display space after the placement of the rectangle.

A : Aspect ratio of the display space.

T : Constant value lead by the below procedure.

T is calculated by the following procedure. Supposing the position of the corner of the rectangle lapping over a corner of a subspace as (x_s, y_t) , the technique specifies the priority of the candidate position as follows:

Priority 1: If all of the three (or less) subspaces touching (x_s, y_t) have been already occupied, the candidates are categorized as priority 1. See Figure 6(upper-left).

Priority 2: If one of the three (or less) subspaces touching (x_s, y_t) have been already occupied, the candidates are categorized as priority 2. See Figure 6(upper-right).

Priority 3: If two of the three (or less) subspaces touching (x_s, y_t) have been already occupied, the candidates are categorized as priority 3. See Figure 6(mid-left).

Priority 4: If none of the three (or less) subspaces touching (x_s, y_t) have been already occupied, the candidates are categorized as priority 4. See Figure 6(mid-right).

The technique additionally calculates candidate positions outside the display space, if there is no candidate position adequate to place the rectangle. The outside

candidates are treated as lowest priority candidates as follows:

Priority 5: Outside candidates are categorized as priority 5. See Figure 6(lower).

The technique defines constant values t_k ($k=1..5$), which satisfies the following inequation:

$$t_1 < t_2 < t_3 < t_4 < t_5$$

The above values are used as $T=t_k$ in the equation (1).

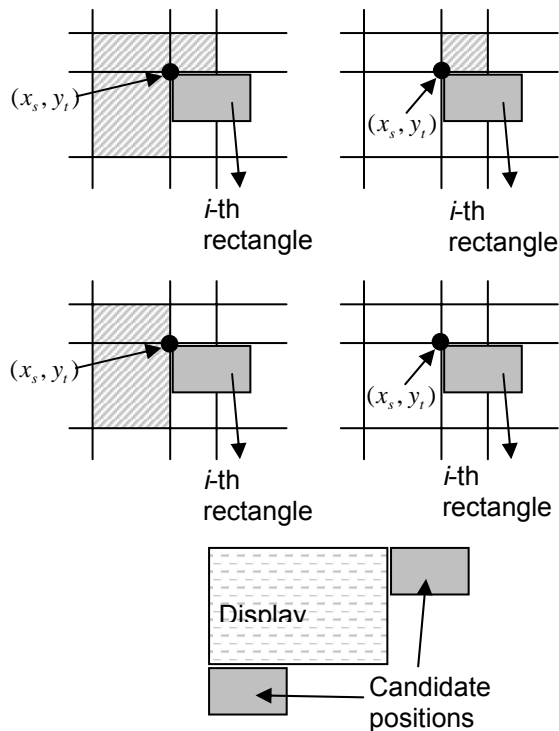


Figure 6. Five stages of priority of candidate

3.2.5 Decision of position of rectangles

The technique repeats the process described in Section 3.2.3 for all subspaces, and Section 3.2.4 for all candidate positions. Finally the technique decides the position of the i -th rectangle where the penalty value is minimum. As shown in Figure 5(right), the technique then divides the subspaces by the extension lines of the i -th rectangle, and repeats the similar process for $(i+1)$ -th rectangle.

4 Experiments of visualization of multi-dimensional data of bioactive chemicals

This section described the experiments of visualization of multi-dimensional data of bioactive chemicals using HeiankyoView. In this experiments, authors used the metabolism data of 161 drugs against 5 CYPs (CYP1A2, CYP2C9, CYP2C19, CYP2D6, and CYP3A4).

4.1 Grouping of CYP isozymes according to their metabolic susceptibility

First stage of the experiment classifies the 161 drugs into 6 groups according to their metabolic susceptibility to the CYP isozymes. Here authors represented the characteristics of the drugs as five-dimensional vectors led from their metabolic susceptibility with each of 5 CYPs,

and classified the drugs using self organizing map (SOM). Figure 7 shows an example of SOM, where the darkness of each node denotes the value of elements of the vectors. The SOM denotes that many drugs are metabolized by CYP3A4. It also denotes that substrate specificity of CYP1A2, CYP2C9, and CYP2C19 does not overlap with one another.

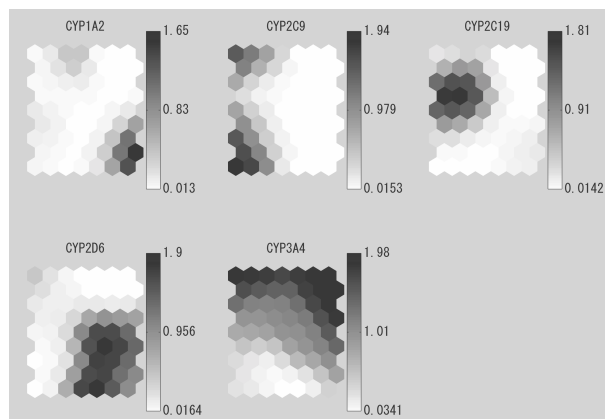


Figure 7. Example of SOM classified according to the metabolic susceptibility of 161 drugs.

Next authors divided the SOM nodes into 6 groups according to their Davies-Bouldin Index (DB) [Dav79]. They then divided the 161 drugs into 6 groups by mapping the drugs to their best-matched winning node. The group information was subjected to the following recursive partitioning analysis.

4.2 Recursive partitioning

Authors divided drugs into two subsets to maximally increase the internal homogeneity of groups within the subsets. Here, the internal homogeneity was defined as information entropy: *i.e.*, $\sum \{-P(s_i) \log_2 P(s_i)\}$ where $P(s_i) = n_i/N$; n_i and N are numbers of category i data and the total in (sub)data set, respectively. On the other hand, predictor variables for partitioning were molecular constitutional descriptors derived from chemical structure by using Dragon 5.2 (Talet srl, Italy) [Tod00]. Recursively repeating the division, they constructed a binary classification tree. The binary tree aimed at classification of the drugs varying in metabolic susceptibility based on their chemical structure.

4.3 Visualization by HeiankyoView

Authors visualized the hierarchical data of the 161 drugs using HeiankyoView. Figure 8 shows the example of visualization results, where icons denote the drugs, darkness of the icons denotes the metabolic susceptibility, and rectangular borders represent the group of drugs according to their chemical structures.

The visualization result denotes that the most of drugs metabolized by CYP1A2 or CYP2C19 are found in the right side of the display, but drugs metabolized by CYP2D6 or CYP3A4 are found in both left and right sides of the display. In the recursive partitioning analysis, the primary concern raised for classification was whether sum

of atomic Sanderson's electronegativity (Se) be less than 44.89 or not. Taking together, it was found that drugs metabolized by CYP1A2 and CYP2C19 mostly possess the Se less than 44.89. Such visualization result proves that the proposed visualization technique can contribute to analyze the relationship between metabolic susceptibility profile and chemical structure.

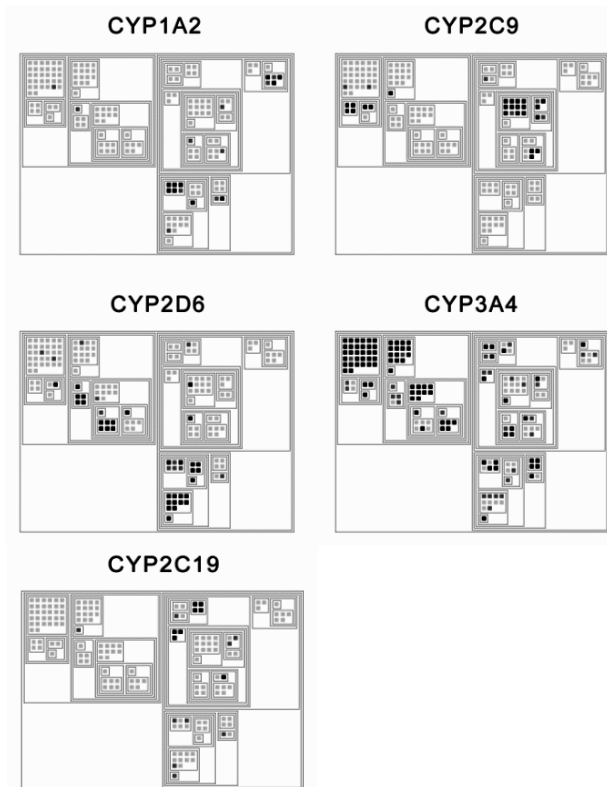


Figure 8. Visualization of hierarchical data of the 161 drugs using HeiankyoView. Dark leaf-nodes denote the drugs whose metabolic susceptibilities are high.

4.4 Discussion

This study was the first experiments for authors to deal with binary trees as hierarchical data. Visually comparing between Figures 1 and 8, HeiankyoView does not provide good display layout for binary trees as much as those of non-binary trees. Authors are currently studying some algorithmic improvement, especially for reducing waste spaces of layout results with binary trees.

Authors are implementing the metaphor of bioactive chemicals as clickable icons, so that users can obtain detailed information of the chemicals on demand. [Requirement 1] listed in the Section 2.1 is very important for developing clickable visualization techniques. HeiankyoView is advance in this point, as well as some of space-filling hierarchical data visualization technique such as Treemap. HeiankyoView is somewhat similar to Quantum Treemap [Bed02] because both techniques attempt to represent metaphors as equally- shaped and sized icons. Comparison between authors' previous technique and Quantum Treemap is provided in [Ito04], and authors are currently evaluating HeiankyoView and Quantum Treemap to provide the comparison.

5 Conclusion

This paper introduced an application of "HeiankyoView", a large-scale hierarchical data visualization technique, to the visualization of multi-dimensional data of bioactive chemicals. In the application authors constructed the hierarchical data of 161 drugs according to the chemical structure, and represented their metabolic susceptibilities as darkness of icons denoting the drugs. They proved that the visualization technique can contribute to analyze the relationship between metabolic susceptibility profile and chemical structure.

Authors are improving the rectangle packing algorithm of HeiankyoView according to the issue discussed in the Section 4.4, and enhancing the implementation of the presented visualization technique according to the following issues:

- All-in-one representation of multi-dimensional values in one rectangular space.
- Filtering of non-interesting compounds.
- Sophisticated indication of detailed information.

In addition to improvements of visualization techniques themselves, it is also important to combine other computing techniques with the visualization techniques. For example, it should be useful if multi-dimensional data of virtual compounds are visualized with those of real compounds at the same time. The virtual compounds can be developed on a computer, and their multi-dimensional data can be predicted from the data of structurally similar real compounds. Such simulation and visualization can contribute to shorten the development cycle and expand the possibility of new compounds.

6 References

- [Bed02] Bederson B., Schneiderman B., Ordered and Quantum Treemaps: Making Effective Use of 2D Space to Display Hierarchies, ACM Transactions on Graphics, Vol. 21, No. 4, pp. 833-854 (2002).
- [Bon01] Bonabry, P., Sievering, J., Leemann, T., Dayer, P. Quantitative drug interactions prediction system (Q-DIPS): a dynamic computer-based method to assist in the choice of clinically relevant in vivo studies. Clinical Pharmacokinetics, Vol. 40, pp. 631-640 (2001).
- [Bru00] Bruls D.M., Huizing K., Wijk J. J., Squarified Treemaps, Proceedings of Data Visualization 2000, pp. 33-42 (2000).
- [Car95] Carriere J., et al., Research Paper: Interacting with Huge Hierarchies beyond Cone Trees, IEEE Information Visualization 95, pp.74-81 (1995).
- [Chu98] Chuah M., Dynamic Aggregation with Circular Visual Designs, IEEE Information Visualization '98, pp.35-43 (1998).
- [Dav79] Davies D. L., and Bouldin D. W., A Cluster Separation Measure, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 1, pp. 22-27 (1979).
- [Gue97] Guengerich F. P., Role of cytochrome P450 enzymes in drug-drug interactions, Advances in Pharmacology, Vol. 43, pp. 7-35 (1997).

[Ing04] Ingelman-Sundberg M., Pharmacogenetics of cytochrome P450 and its applications in drug therapy: the past, present and future. Trends in Pharmacological Sciences, Vol. 25, pp. 193-200 (2004).

[Ito03] Itoh T., Koyamada K., HeiankyoView: Orthogonal Representation of Large-scale Hierarchical Data, International Symposium on Towards Peta-Bit Ultra Networks (PBit 2003), pp. 125-130, 2003.

[Ito04] Itoh T., Yamaguchi Y., Ikehata Y., and Kajinaga Y., Hierarchical Data Visualization Using a Fast Rectangle-Packing Algorithm, IEEE Transactions on Visualization and Computer Graphics, Vol. 10, No. 3, pp. 302-313 (2004).

[Ito05] Itoh T., Takakura H., Sawada A., Koyamada K., Interactive Poster: Visualization of Network Intrusion Detection Data Using a Hierarchical Data Visualization Technique, IEEE Information Visualization, 2005.

[Joh91] Johnson B., et al., Tree-Maps: A Space Filling Approach to the Visualization of Hierarchical Information Space, IEEE Visualization '91, pp. 275-282 (1991).

[Koi95] Koike H., Fractal Views: A Fractal-Based Method for Controlling Information Display, ACM Transactions on Information Systems, Vol. 13, No. 3, pp.305-323 (1995).

[Lam96] Lamping J., Rao R., The Hyperbolic Browser: A Focus+context Technique for Visualizing Large Hierarchies, Journal of Visual Languages and Computing, Vol. 7, No. 1, pp. 33-55 (1996).

[Ren97] Rendic, S., Di Carlo, F. J., Human cytochrome P450 enzymes: a status report summarizing their reactions, substrates, inducers, and inhibitors. Drug Metabolism Reviews, Vol. 29, pp. 413-580 (1997).

[Rek93] Rekimoto J., The Information Cube: Using Transparency in 3D Information Visualization, Third Annual Workshop on Information Technologies & Systems, pp.125-132 (1993).

[Shn01] Shneiderman B., et al., Ordered Treemap Layouts, IEEE Information Visualization Symposium 2001, pp. 73-78 (2001).

[Spr00] Sprenger T. C., et al, H-BLOB: A Hierarchical Visual Clustering Method Using Implicit Surfaces, IEEE Visualization 2000, pp. 61-68 (2000).

[Tod00] Todeschini R., and Consonni V., *Handbook of Molecular Descriptors*, Wiley-VCH, Weinheim and New York, 2000.

[Yam02] Yamaguchi Y., Itoh T., Ikehata Y., Kajinaga Y., Interactive Poster: Web Site Visualization Using a Hierarchical Rectangle Packing Technique, IEEE Information Visualization Symposium, 2002

[Yam03] Yamaguchi Y., Itoh T., Visualization of Distributed Processes Using "Data Jewelry Box" Algorithm, CG International 2003, pp. 162-169, 2003.

Mesh Simplification using Ellipsoidal Schema for Isotropic Quantization of Face-Normal Vectors

Ganesan Subramaniam and Kenneth Ong

Department of Electronics and Computer Engineering
National University of Singapore
Block E4, Level 5, Room 48, 4 Engineering Drive 3, Singapore 117576

engp1661@nus.edu.sg, eleongk@nus.edu.sg

Abstract

In this paper, we present a method for simplification of arbitrary 3D meshes that is based on Isotropic Quantization of face-normal vectors. There are three stages. Firstly, a codebook that contains the unique face-normal vectors of the 3D mesh is generated using our Ellipsoidal Schema. Secondly, the polygons of the mesh are grouped into patches: based on the codebook vectors and the locality information of the polygons. Polygons that have isotropic and geographical similarities are grouped together. And the resulting patch is approximately a flat plane with its corresponding codebook vector as its normal. In the last stage, our mesh simplification technique re-triangulates the patch, in which the algorithm only considers the vertices on the borders of the patch. We demonstrate that our technique yields better results when applied with multiple iterations than when using a single iteration. Thus using patch-wise quantization, our technique is able to simplify 3D meshes.

Keywords: Mesh Simplification, Ellipsoidal Schema, Isotropic Quantization, Codebook

1 Introduction

The performance of graphics subsystems has improved enormously in the last few years. However the complexity of 3D scenes and graphics applications has also increased greatly. A very critical aspect in interactive 3D graphics is the complexity, in terms of number of triangles, of the meshes to be processed and rendered. A large number of meshes can be easily produced in many applications: by fitting isosurfaces on large data-sets, that is by converting surfaces into triangulated meshes, by 3D scanning real-world objects. All 3D scanners perform a regular sampling of the surface of an object, returning triangle meshes of complexity directly proportional to the scanner's sampling resolution and the object surface area. The resulting meshes typically have a few million triangles.

These complex meshes introduce severe overheads in transmission, rendering, processing and storage. Thus the visualization of such models cannot be realized in real-time without mesh simplification techniques to reduce the

number of triangles that are rendered while maintaining the quality of the mesh. For digital geometry processing, most scanned models must undergo complete re-meshing before processing. Geometry-processing algorithms such as smoothing and compression can benefit from parameterization-based re-meshing techniques, combined with uniform or curvature adapted sampling. Here the key is to parameterize the original mesh and minimize distortion due to the flattening process. However such techniques suffer from various drawbacks such as surface cutting, overlapping, increased computational load (due to expansive mesh optimization algorithms) and topological errors (due to inaccuracy as a tradeoff for complexity).

Our main contribution to the domain of mesh simplification is the formulation of a dynamic codebook schema for isotropic surface sampling. We present in this paper, the Ellipsoidal Schema (see Section 2.1.1 and 2.1.2) and its variants that is dependent on our geometric model called the GAEA¹. The GAEA is used in the creation of the Ellipsoidal Schema codebook that is based on the face-normal vectors of the GAEA face-normals. It also ensures distinct separation of the regions formed by the codebook vectors (also known as code-vectors). Due to the inherent properties of a GAEA, it is possible to create both uniformly and non-uniformly distributed set of code-vectors. These code-vectors are later used by the Polygon Grouping stage as shown in Figure 1 (see next page).

The generated codebook is applied to the 3D mesh in the Polygon Grouping stage (see Section 2.2). Here the polygons of the 3D mesh are grouped into non-overlapping regions based on orientation and locality of the polygons. In the final stage of our mesh simplification process, we re-mesh the isosurfaces created in the Polygon Grouping stage. Here we employed Edge Collapse algorithm (see Section 2.3) to flatten out the polygons. To test the effectiveness of our mesh simplification technique, we applied our mesh simplification technique several models of varying sizes (see Section 3) and show our findings when the simplification technique is iterated on the same model.

2 Mesh Simplification Technique

Our mesh simplification technique focuses on the generation of a codebook for quantizing a 3D mesh into patches (groups of polygons). There are three steps involved for the creation of the simplified mesh (see Figure 1). They are Codebook Generation (see Section 2.1), Polygon Grouping (see section 2.2), and Patch Re-meshing (see section 2.3). Each of these processes is discussed in greater detail in the following subsections.

Copyright (c) 2005, Australian Computer Society, Inc. This paper appeared at Asia Pacific Symposium on Information Visualization 2006, National Center of Sciences, Tokyo, Japan. Conferences in Research and Practice in Information Technology, Vol. 60. Editors, K. Misue, K. Sugiyama and J. Tanaka. Reproduction for academic, not-for profit purposes permitted provided this text is included

¹ GAEA is the name that we have given for the new geometric model that we are introducing and it is not an acronym.

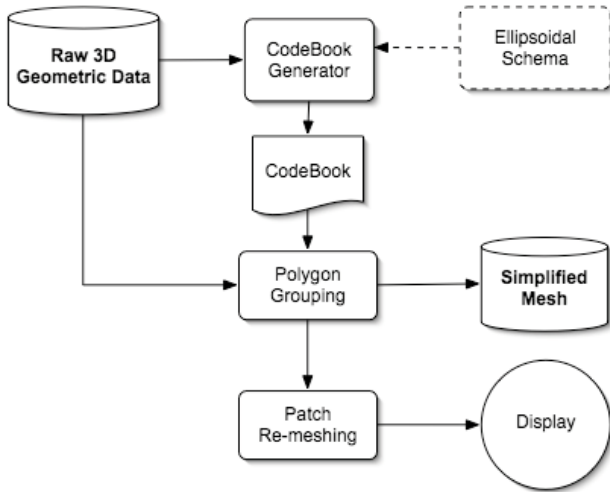


Figure 1: Shows the overall process of our mesh simplification technique.

2.1 Codebook Design

The Polygon Grouping process requires a codebook to quantize the face-normal vectors of a 3D mesh. However designing a codebook can sometimes requires an exhaustive search for the best possible code-vectors in space; and the search increases exponentially as the number of code-vectors increases. Therefore we resort to a suboptimal codebook design schema.

We introduce the Ellipsoidal Schema that uses a pre-defined general model called GAEA- n (see Section 2.1.1 for the explanation of $-n$) whose code-vectors are uniformly distributed about the origin. This schema can be extended to create a non-uniformly distributed codebook, GAEA- xyz (see Section 2.1.2 for the explanation of $-xyz$) using the pre-defined general model.

2.1.1 Uniform Ellipsoidal Schema

To produce our Uniform Ellipsoidal Schema codebook, we introduce a new geometric model called GAEA- n . A GAEA- n is basically a sphere, which is symmetrical about $n/2$ different directions (much like a cube) and has n number of subdivisions along each axis. In fact, we can create a GAEA- n by expanding all the vertices of a cube to fit a sphere's profile (see Figure 2).

A GAEA- n model provides a uniformly² distributed set of code-vectors that shall be used in the quantization of the orientation angles of the face-normal vectors of a 3D mesh. The n in GAEA- n refers to the number of subdivisions along the each axis where $n > 0$. For example, GAEA-3 means that this object is created from a cube that has 3 subdivisions along each axis. A GAEA- n object has two notable properties. One of which is that a GAEA-1 is actually a cube with 1 subdivision. This means that a GAEA- n object shall have at least 6 unique face-normals and each one lies along one of the three axes (i.e. X, Y, and Z axes).

The second property of the GAEA- n is that all its face-normals are unique and are Since our goal is to quantize the face-normal vectors of a 3D mesh, we would need to choose the right number of subdivisions for the GAEA- n to create a comprehensive codebook. To aid in this decision, it is important to note that the number of code-

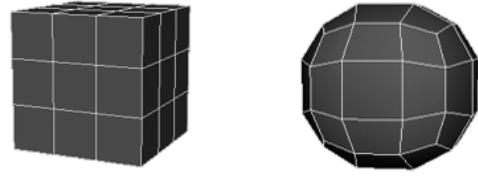


Figure 2: Shows that a cube with 3 subdivisions can be transformed into a GAEA-3 whose face-normals form the vectors for the codebook of the Ellipsoidal Schema.

vectors obtained through this schema is directly related to the number of subdivisions based on the following equation:

$$V = 6 * n^2$$

The equation above shows that the codebook size has a quadratic relationship with the number of subdivisions. In other words, a small increase in the number of subdivisions shall yield a large number of code-vectors. For example, GAEA-3 will yield 54 code-vectors while GAEA-4 will yield 96 code-vectors. As the codebook size increases, the resolution (refers to the number of polygons) of the simplified mesh will also increase with the number of quadrilaterals in the simplified mesh.

2.1.2 Non-Uniform Ellipsoidal Schema

GAEA- n provides a general solution to quantizing orientation angles of a mesh and thus, can be applied to all meshes. In some cases however, a uniformly distributed codebook may not be adequate enough to describe the 3D mesh with the desired accuracy. This might become apparent in cases where only certain face-normal vectors are predominant in the mesh. A uniformly distributed set of code-vectors tends to over generalize this mesh, resulting in the loss of key details in the final simplified mesh.

To address this problem, we extend GAEA- n such that certain ranges of face-normal vectors are sampled at a higher frequency than others. This property can be achieved by varying the number of subdivisions in each axis independently. To cater to this modification, we introduce a new model called GAEA- xyz .

The xyz in GAEA- xyz denotes the number of subdivisions along the respective axes. Since each axis is now controlled separately, the formula for finding the number of code-vectors of a GAEA- xyz object shall be given as follows:

$$V = (x * y + y * z + z * x) * 2$$

By varying the subdivisions along each axis, GAEA- xyz creates a codebook for the 3D mesh as shown in Figure 3. Unlike GAEA- n 's codebook, this codebook is able to relax quantization on areas that have fewer features and emphasizes quantization on areas that have more features. The values for x , y , and z can be found by determining the bounding box dimensions of the 3D mesh. Then, for a desired codebook size, one can determine the values of the subdivisions for a GAEA- xyz based on a pre-determined filter.

² Uniform distribution here means that the angles between the face-normals of the GAEA- n are equal.

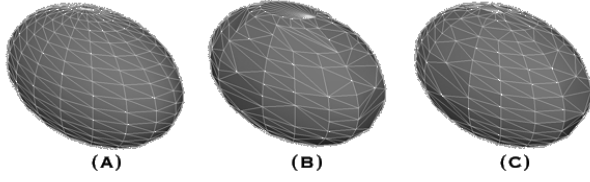


Figure 3: Shows (A) original mesh (B) after applying GAEA-3 codebook (C) after applying GAEA-5-4-3.

Having determined the initial codebook for the 3D mesh, we can now proceed to the Polygon Grouping stage (see Section 2.2). We refine the codebook to accommodate the locality criteria for our mesh simplification technique. We accomplish this by first grouping the polygons based on the orientation angles of the code-vectors. In other words, all polygons that face a certain (approximate) angle shall be grouped together. This angle is provided by the vectors of the codebook. Then using the adjacent polygon information from the 3D mesh, we break down the groups into smaller sets of connected polygons. Finally the averages of the face-normal vectors of each set of connected polygons shall form the code-vectors for the new codebook.

2.2 Polygon Grouping

Polygon Grouping is a process that maps a 3D mesh's face-normal vectors (in the vector space R^k) into a codebook, $Y = \{y_i: i = 1, 2 \dots N\}$ where y_i is a finite set of code-vector. Associated with each code vector, y_i , is a nearest neighbor region called *Voronoi* region, and is defined as:

$$V_i = \{x \in R^k : \|x - y_i\| \leq \|x - y_j\|, \text{ for all } j \neq i\}$$

One of the key features of a Voronoi partitioning is that a set of *Voronoi* regions partition the entire space R^k such that:

$$\bigcup_{i=1}^N V_i = R^k \quad \text{and} \quad \bigcap_{i=1}^N V_i = \emptyset \quad \text{for all } i \neq j$$

The representative code-vector is determined to be the closest in Euclidean distance from the input vector. The Euclidean distance is defined by:

$$d(x, y_i) = \sqrt{\sum_{j=1}^k (x_j - y_{ij})^2}$$

Euclidean distance accounts for the locality of the polygons in a 3D mesh. However for the purposes of Polygon Grouping, we have decided that orientation is also important. In other words, the polygons that are close together in Euclidean space and are facing in the same direction shall be clustered together to form a group. To accommodate the latter criterion, the orientation-based grouping, we redefine our *Voronoi* region as follows:

$$V_i = \begin{cases} x \in R^k : \|x - y_i\| \leq \|x - y_j\|, & \text{for all } j \neq i \\ \cos\left(\frac{x \cdot y_i}{\|x\| \cdot \|y_i\|}\right) \leq \cos^{-1}\left(\frac{x \cdot y_j}{\|x\| \cdot \|y_j\|}\right), & \text{for all } j \neq i \end{cases}$$

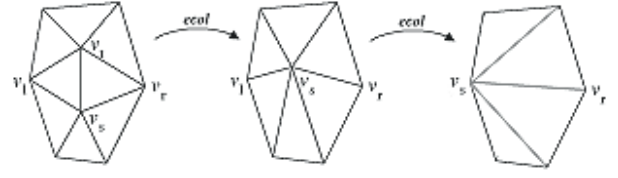


Figure 4: Illustration of the flattening process of a patch using a series of edge collapse transformation.

Thus for Polygon Grouping, the representative code-vector is determined to be the closest in Euclidean distance from the input vector and closest in orientation to the input vector. During the Polygon Grouping process, an input face-normal vector is matched against the code-vectors generated from an Ellipsoidal Schema and the index to the code-vector that offers the least distortion is output. In this case the lowest distortion is found by evaluating the Euclidean distance and the orientation angle between the input vector and the code-vector in the codebook.

2.3 Patch Re-meshing

After grouping polygons into patches, we are now left with the task of flattening the patches. Since the patches consist of polygons that are isotropic, each patch can be treated as a flat 2D plane. Thus removing the vertices in the interior of the patch should not affect the fidelity of the simplified mesh to the original mesh too greatly. The remaining boundary vertices shall be triangulated to form the new patch. Although any polygon triangulation algorithms can be used, we have implemented edge-collapsing technique to flatten a patch.

As shown in Figure 4, an edge collapse transformation $ecol(\{v_s, v_t\})$ unifies 2 adjacent vertices v_s and v_t into a single vertex. The vertex v_t and the two adjacent faces $\{v_s v_t v_l\}$ and $\{v_t v_s v_r\}$ vanish in the process. A position s is specified for the new unified vertex.

To begin an edge-collapse operation, we first have to select an edge based on a cost-function. Our cost function shall be the shortest edge with an *interior* vertex. An interior vertex is one that does not lie on the boundary of a patch. Since our goal is to flatten a patch, the interior vertex shall collapse into a boundary vertex. Thus for a set of β boundary vertices of a patch, the cost function for the selection of an edge can be given as follows:

$$C = \{\min [|v_s - v_i|], \text{ for } i \in \beta\}$$

Using this equation, we can then proceed to flatten a patch and thus complete our simplification process. Note however that other polygon triangulation techniques can be used to achieve our goal. However we have chosen this technique due to its simplicity and ease of implementation.

3 Results and Discussion

The mesh simplification technique described in this paper has been implemented as an interactive software. The user shall be able to control the simplification via the selection of parameters for our codebook generation schema. Models in PLY formatted files can be loaded into the program directly and the Ellipsoidal schema can be applied onto them. Figure 5 shows the result of the applying mesh simplification using our software.

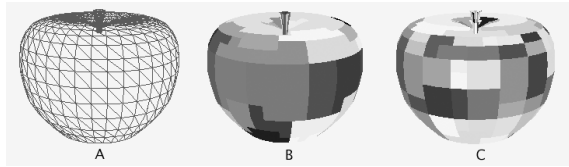


Figure 5: Shows (A) original mesh (B) after applying GAEA-3 codebook (C) after applying GAEA-5.

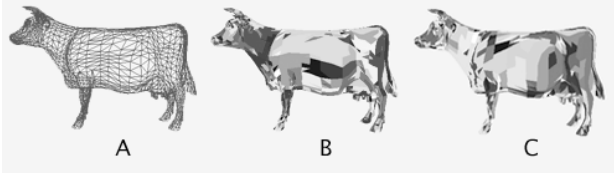


Figure 6: Shows (A) original mesh (B) after applying GAEA-3 codebook (C) after applying GAEA-5 to GAEA-1 iteratively.

We have run our technique on a variety of models of arbitrary complexity. Figure 5 illustrates simplification of a polygonal model with 1704 polygons. After applying our simplification technique, the new model has 1548 polygons by using GAEA-3 codebook and 1672 polygons by using GAEA-5 codebook. The following table shows all the results for the polygonal models on which we have used our technique.

From the Table 1 above, we can deduce that while a smaller order GAEA produces a simpler resulting mesh, it also suffers in terms of quality. However in another observation of the results in the table above, it seems that the algorithm is only able to produce minimal reduction in the number of polygons. To achieve higher rates of polygon reduction, we have realized that a model needs to be passed through our algorithm for several iterations. In each iteration, if the number of polygon does not reduce, we would reduce the GAEA order till GAEA-1 is reached. The results of this experiment is shown Figure 6 where the cow model was put through this test. We started with GAEA-5 and ended the algorithm after GAEA-1.

We have also discovered that instead of using uniform codebooks during the iterations, using non-uniform codebooks yielded more polygon reductions. This is due to the properties of the non-uniform codebooks to “better” fit the model. Based on the shape of the model we have selected appropriate values for the dimensions of GAEA-xyz. During the iterations, we reduce the every dimension of GAEA till they are all 1s (i.e. till GAEA-1-1-1 is reached). The results are as follows:

4 Conclusion

Our Mesh Simplification technique involves three steps as illustrated in Figure 1. The first step creates a codebook based on either of one of our polygon quantization schemas: Uniform Ellipsoidal Schema and Non-Uniform Ellipsoidal Schema. In the second step of our mesh simplification process, we group polygons based on the resulting codebook from step 1. The grouped polygons shall be isotropic and have similarities in locality. This thus allows us to flatten each group of polygons without too much loss

| | apple.ply | cow.ply | bunny.ply | dragon.ply |
|---------------------|-----------|---------|-----------|------------|
| Original | 1704 | 5804 | 69451 | 871414 |
| Using GAEA-1 | 1216 | 3074 | 17993 | 265776 |
| Using GAEA-3 | 1548 | 5274 | 37660 | 435814 |
| Using GAEA-5 | 1672 | 5614 | 48229 | 550340 |

Table 1: Shows the polygon counts after applying the mesh simplification technique on several models.

| | cow.ply | bunny.ply | dragon.ply |
|------------------------|---------|-----------|------------|
| Original | 5804 | 69451 | 871414 |
| Initial GAEA | 5-3-1 | 5-5-3 | 5-5-1 |
| Simplified Mesh | 2755 | 10857 | 130611 |
| Timing | 27 secs | 65 secs | 935 secs |

Table 2: Shows the results when iterative simplification and non-uniform codebooks were used.

of fidelity. As observed in step 3, different codebooks provide different results. Thus the choice of the quantization schemas can affect the size of the final simplified mesh.

5 References

- Michael Garland and Paul S. Heckbert (1997): Surface simplification using quadric error metrics, SIGGRAPH, pages 209–216.
<http://www.cs.cmu.edu/~garland/quadrics>.
- Cohen, J., M. Olano, and D. Manocha (1998): Appearance-Preserving Simplification, SIGGRAPH.
- Hugues Hoppe (1996): Progressive meshes, SIGGRAPH, pages 99–108, <http://research.microsoft.com/hoppe/>
- W. Sweldens and P. Schröder (2001): Digital Geometry Processing, Course Notes. ACM SIGGRAPH.
- X. Gu, S. Gortler, and H. Hoppe (2002): Geometry images, SIGGRAPH, pages 355–361.
- P. Alliez, E. Colin de Verdière, O. Devillers, and M. Isenburg (2003): Isotropic surface remeshing, Shape Modeling International.
- P. Alliez, M. Meyer, and M. Desbrun (2002): Interactive geometry remeshing. ACM Transactions on Graphics, 21(3):347–354. SIGGRAPH conference proceedings.
- M. Desbrun, M. Meyer, and P. Alliez (2002): Intrinsic parameterizations of surface meshes, Eurographics, pages 209–218.
- B. Lévy, S. Petitjean, N. Ray, and J. Maillot (2002): Least squares conformal maps for automatic texture atlas generation, SIGGRAPH, pages 362–371.
- Pragyana Mishra, Omead Amidi and Takeo Kanade (2004): EigenFairing: 3D Model Fairing using Image Coherence, British Machine Vision Conference, Vol. 1, pp. 17-26
- Markus Hadwiger (1998): Mesh Simplification and Multi-resolution Data Structures,
<http://www.cg.tuwien.ac.at/studentwork/VisFoSe98/msh/>

Visualization of a Closed Three-Dimensional Surface using Portal-based Rendering

Michael Bui

Nick Lowe

Masahiro Takatsuka

ViSLAB, The School of IT, The University of Sydney
Email: mbui|nickl|masa@vislab.usyd.edu.au

Abstract

The complexity and size of data is rapidly increasing in modern science, business and engineering. This has resulted in increasing demands for more sophisticated data analysis methods. Multidimensional scaling has been used to visualize large high-dimensional datasets in the form of a map. Such maps are very intuitive for us, as we are familiar with reading geographical maps. However, they typically result in a flat space (world), which presents undefined discontinuous edges at the end of the world.

In order to provide a continuous space for visualizing high-dimensional data, a closed three-dimensional (3D) surface, such as surfaces of a sphere and a torus, has been used as a target mapping space for multidimensional scaling. This paper proposes an application of a portal-based rendering technique to visualize such closed 3D surfaces. This 3D rendering technique allows users to see and navigate through the surface in a natural manner. This eliminates any discontinuous boundaries usually introduced by a process of projection from 3D to two-dimensional space. Furthermore, as our initial investigations show, it is a general technique that is applicable to any closed 3D surface.

Keywords: Visualization, Portal-based Rendering, Closed 3D Surface, Geodesic Dome, Self-Organizing Map.

1 Introduction

Many multidimensional scaling techniques are used to map the high-dimensional data onto two or three dimensional Euclidean space in order to allow user to visually inspect massive and complex datasets. A typical multidimensional scaling method, however, produces a discontinuous and distorted 2D map-like representation which has boundaries, whose connectivities are not well defined. To solve this, a few techniques to map multidimensional data to a closed 3D surface, such as a sphere (Wu & Takatsuka 2005) and a torus (Li 2004), have been developed. Such methods typically use a projection method to visualize the 3D surface on a 2D plane. The other approach is to remain in the 3D space and to provide intuitive and effective visualization/navigation methods.

In this paper, we propose a visualization technique that renders a closed 3D surface as a flat surface using a portal-based rendering technique. The resulting

3D display of the surface would give a similar environment as that seen in first person 3D video games. This new approach allows us to render the flat 3D scene without re-projecting the 3D surface onto a 2D rectangular surface. Moreover, it significantly increases the amount of visible data compared to conventional methods and can be applied to any closed surface.

2 Background

2.1 Spatialization

Information Visualization methods, which produce a map-like representation (Skupin & Fabrikant 2003) are of key interest to our research. The primary reason for this is familiarity, where most people at one stage or another will interact with a map, for example, when travelling. Therefore users are much more likely to be able to interpret maps or environments with little to no training. Furthermore, the 3D terrain-like rendering of data space will allow us to use our 3D navigational skill to explore the data space.

Multidimensional scaling is a set of data analysis methods that visualize a data structure of a high-dimensional data space as a two or 3D geometrical picture. It was originally used to study psychometrics; understanding how people judge similarities in a set of objects (Torgerson 1952).

The Self-Organizing Map (SOM) (Kohonen 1995), is a popular artificial neural network algorithm that has been used for a diverse range of tasks, such as unsupervised clustering, topological mapping, and of course, visualization. Recent attempts by cartographers to use multidimensional scaling, especially SOMs, to produce map-like representation has resulted in the growing popularity of such visualizations in order to assist visual inspection of complex data.

2.2 Mapping onto toroidal surfaces

Although typical multidimensional scaling techniques that map data onto a 2D rectangular space such as the SOM can be very useful for visualization, these techniques generate discontinuous maps where the connectivity of the boundaries may not be well defined. Furthermore, the SOM has a problem known as the "border effect", where the map appears to be less well ordered near the borders of the map. This led to the development of techniques that map high dimensional data onto a closed surface. For instance, Li's work (Li 2004) used a surface of a torus to map an n-dimensional feature space onto a 2D space (the relational perspective map). In order to provide an intuitive visualization, this surface was opened up and flattened to produce a 2D rectangular map-like representation.

For the SOM however, one proposed solution was to implement the SOM on a spherical lat-

tice. After investigating various results from different researchers, Wu and Takatsuka concluded that an icosahedron-based geodesic dome would be most suitable for implementing a spherical SOM(Wu & Takatsuka 2005). This eliminated the border effect while being able to perform fast neighbourhood searching on the 2D data structure that used only $O(n)$ space. Additionally, extreme points can be defined to create a sense of direction that would help in the comparison of visualizations of different datasets. This technique would further aid users in the building of their own mental maps and as experiments showed, the GeoSOM had more uniform error distribution and potentially better performance in dealing with large datasets.

The major disadvantage these approaches have though, is that in order to get a full view of the entire data, the user would need to rotate a 3D surface or to introduce artificial boundaries to re-project the 3D surface onto a 2D surface. However, conventional techniques applied to this domain introduce distortions while also requiring cut points to be defined, which varies from surface to another, in order to open up the 3D surface.

2.3 Portal-based rendering

The concept of portal-based rendering was first proposed by Jones for hidden-line removal(Jones 1971). The fundamental idea behind portal-based rendering is that a cell can only be visible if the "user" is inside it or they can see it through a doorway, that is, a portal. As a result of scene decomposition, a Cell and Portal Graph (CPG) is created. This graph describes the adjacency information between cells by representing cells as nodes and portals as edges. Traversal of the CPG, from the cell containing the eye point and following only edges that represent visible portals, produces a visible set of cells.

Recently, Lowe and Datta proposed a general paradigm for portal-based rendering, where the portals were nonconvex and nonplanar(Lowe & Datta 2005). This provides a more flexible framework for dynamic scene composition and more applications where portal-based rendering can be used. In contrast to previous works, their paradigm simply states that cells contain data, and portals can be used to connect them. There are no binding geometric constraints. Doing so, allows meaningful, dynamic and visual links can be created between related data in a visualization environment, such as the the GeoSOM.

3 Portal-based Rendering of a Closed 3D Surface

3.1 Cell and Portal Configuration

Our proposed technique relies on having the multi-dimensional data mapped onto a closed 3D surface. Furthermore, this surface must be triangulated. Each triangle on a surface represents a cell, while portals are constructed at edge of every triangle. Thus for an edge that is shared between two triangles, there are actually two portals on that edge, both leading to opposite directions. This approach to decomposition allows our technique to be applicable to any closed surface while being distance preserving. Figure 1 shows an example of the decomposition process for a triangular pyramid. This closed surface has four cell geometries (triangles A, B, C and D). Edges connecting these cells defines twelve portals. In a Cell Portal Graph (CPG), each portal that connects two adjacent cells is represented as an edge as shown in figure 1(b). In order to generate a seemingly flat surface, we

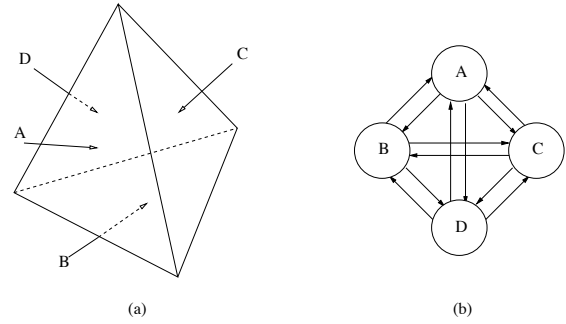


Figure 1: This figures illustrates the decomposition of (a)a triangular pyramid to (b)its corresponding Cell Portal Graph (CPG).

use transformative portals by modifying the portal's transformation based on the adjacency information between cells. For instance, as shown in figure 2, two portals P_{ij} and P_{ji} can be defined for two adjacent triangle cells C_i and C_j . The portal has a rectangle shape and the bottom edge is aligned to one edge of the current cell. The two vertical edges of the portal are defined as edges parallel to the surface normal N_i of the cell C_i and located at the two end points of the bottom edge (see figure 2). Transformation matrices M_{ij} and M_{ji} enables the smooth alignment of these two portals. These transformation matrices are computed from the angle between the two surface normals N_i and N_j using the shared edge as the axis of rotation, while the direction of rotation depends on the existence of concavity/convexity between the two triangles. With portal rendering, the cell C_j viewed from the cell C_i through the portal P_{ij} appears to be on the same flat surface. As a result of this portal configuration process, each triangle cell has three portals. Moreover, for each of these portals, there is another corresponding portal which would face towards the opposite direction, that is, there would be a portal leading the user from the destination cell to the current cell. We also define a plane using three points of the rectangular portal for collision detection, that is, portal traversal.

3.2 Portal-based rendering

Once portals are defined along with transformation matrix between cells, the next and last step of the portal-based rendering is the rendering algorithm. Figure 3 depicts the abstract form of the entire algorithm. Much of it remains the same as Lowe and Datta's portal-based rendering algorithm except in this case a transformation is applied to each adjacent cell to generate a flat surface. This is done by transforming the coordinate system using the transformation matrix obtained during the cell and portal configuration stage. This is done before rendering the adjacent whereby it will be aligned based on the angle between the cells, using the edge as the axis of rotation to ensure both cells are on the same plane. Such transformations are recursively applied to ensure a flat surface will occur. The recursion ends once the maximum recursion depth has been reached or when there are exist portals that are not visible. However for portals that are visible, its visible pixels are marked and unmarked using the stencil buffer along with the recursion depth to ensure that visible portions of a cell are only rendered there.

Note that the camera is computed such that the user views and navigates the scene in a first person manner. Consequently, only portals facing the eye point are rendered with back face culling. This elimi-

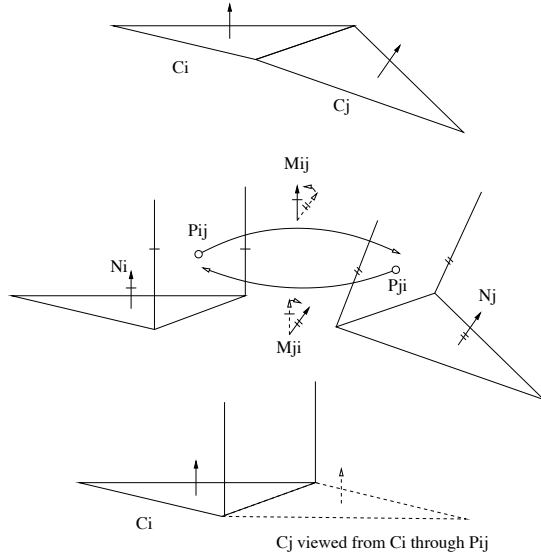


Figure 2: A definition of a portal for a triangulated cell. For two adjacent triangle cells C_i and C_j , two portals P_{ij} and P_{ji} are defined. Transformation matrices M_{ij} and M_{ji} allow smooth rendering between these two portals

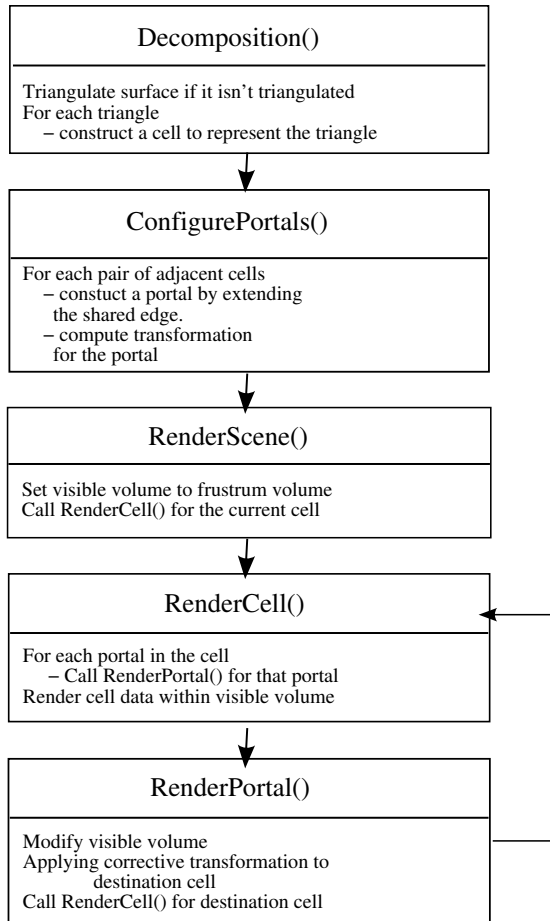


Figure 3: This figure illustrates the abstract form of the entire algorithm to render a closed surface as though it is flat.

nates problems that would have occurred with visibility determination as a result of having two "opposite" portals on a shared edge between two triangles.

4 Experimental results and discussion

In experiments, our aim is to verify the correctness of our approach and algorithm. Our main concern is whether or not the cells are being placed correctly such that our visualization correctly reflects the adjacency information and flattens the closed surface. Additionally, an investigation is needed to verify that the algorithm is applicable to any closed surface.

Our prototype was developed in C++ using OpenGL 1.5.3 implemented by NVIDIA and the ARB_occlusion_query extensions. In order to provide a natural navigation interface, a first person camera is used to navigate the scene. Details for how to implement this will be omitted as this is a textbook material.

4.1 A spherical surface

The first test involves portal-based rendering of a surface of a geodesic dome. It is the same geodesic dome to visualize high-dimensional feature surface on the surface of a sphere in Wu and Takatsuka's work (Wu & Takatsuka 2005). It is the sixth frequency icosahedron-based geodesic dome. Wu and Takatsuka used this geometry to create a spherical Self-Organizing Map. The SOM with 362 neurons were placed on the spherical surface during the training process representing the cluster and topological structure of the high-dimensional dataset. The illiteracy dataset for the year 2000 (*Public Reports: Illiteracy rate and Illiterate Population* n.d.) was used. Note that there is an approximate direction assigned to the dataset. This is particularly interesting as our technique allows users to navigate the data according to this direction which is especially helpful when surfaces are much more complex.

In figure 4, there are two different scenes. A 3D view of the GeodesicSOM is shown in the upper right corner with the location of the user's viewpoint being indicated by the white triangle. White flags were placed at each cell in order to demonstrate the visibility of the surface. Since the scene in the top half of the figure does not use transformative portal-based rendering, a user would not be able to see the surface over the horizon. When the transformative portal rendering is enabled, the rendering provides the same effect as if the spherical geometry is opened up and flattened. This results in provision of the much wider visibility and the ground coverage as shown in the bottom half of the figure.

4.2 A toroidal surface

In the previous experiment, the geodesic dome was entirely convex. To see if our technique is applicable to any closed surface, it would be best if an experiment is conducted on a partially convex and concave surface. Thus the torus was used in this example. Furthermore, visualization techniques that use the torus as a mapping space already exist.

Due to difficulties in finding a dataset that has been mapped onto a torus, we constructed a torus using the Anim8or program (Matt Craighead 2005). In this case, the vertices (even the duplicate vertices) are randomly coloured. The results can be seen in figure 5.

4.3 Visual Artefacts

Our work is still at an early stage and uses simplistic collision detection code. Consequently, portal traversal is being calculated incorrectly. This results in part of the scene being missing when the user is very close

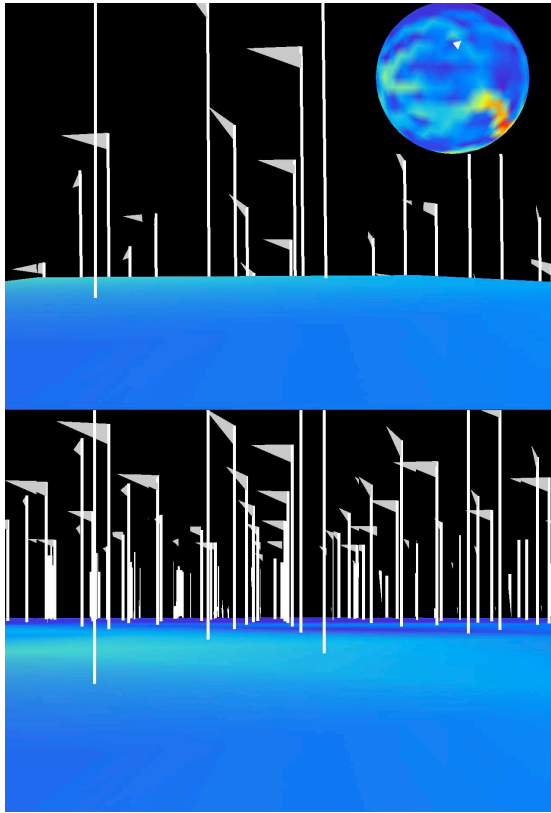


Figure 4: This figure shows the different 3D views of GeodesicSOM. The upper right image contains the original GeodesicSOM with the current cell coloured in white. A first person view from the current location in the GeodesicSOM is shown in both halves of the figure. Here the flags are rendered as a measure of visibility, and transformative portal rendering is not used in the top half while this is enabled for the scene in the bottom half of the figure.

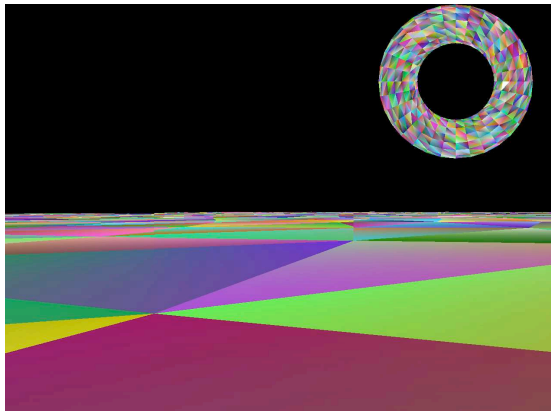


Figure 5: A first person view from the current location on the torus. Transformative portals are enabled which has the capability of flatten a torus where concavity exists between adjacent triangles, not just convexity which is the case for the geodesic dome.

to a portal that is either to the right or left of them. However, this is purely an implementation issue on our part and is not one of the main focuses of our technique.

Another problem that exists are visible portal borders. These exist because when a corrective transformation is applied by the portals, a volume of space is

lost. In practice, these artefacts are barely noticeable, and further subdivision of the closed surface(which can theoretically be applied until the artefacts are subpixels) can alleviate this.

5 Conclusion

Multidimensional scaling techniques that result in a map are intuitive for users to interpret. However undefined discontinuous edges exist at the end of the world. This has led to the development of visualization techniques that use a closed 3D surface such as a sphere and torus. Taking the user into consideration, it is desirable to have an intuitive navigation technique to accommodate the visualization.

We presented an alternative to other conventional techniques for generating a flat surface map that can be navigated through in a natural manner. Although we are using portal-based rendering, our focus here is on "portal-based construction". That is, in contrast to previous work, our objective is to use portals as a tool for novel scene construction, rather than visibility determination.

Furthermore, the use of portal-based rendering has not been applied to this domain before. As our technique decouples the process of generating a closed surface and visualization, this technique can be used on *any* closed surface. This enabling technology clearly shows promise as it can be applied to any surface, improves visibility(of a region of interest), has no distortion along the surface, and provides a consistent way to interpret data. Moreover, this allows various mappings onto more complex 3D surfaces to be used, despite the fact that such closed surfaces may have suited certain datasets more but would have led to difficulties in rendering and interpretation. With our technique, consistent comparisons between any mapping is now possible and data is no longer required to have the same mapping to be compared.

However, as this is initial development of enabling technology, further experimentation is required including the use of more real-world data sets and investigation of complementary techniques such as multi-perspective imaging that may lead to more complete visualizations. Furthermore, our technique is characteristically different than conventional techniques which may open up new avenues for future research on a whole new set of methods.

References

- Jones, C. B. (1971), 'A new approach to the hidden line problem', *The Computer Journal* **14**(3), 232–237.
- Kohonen, T. (1995), *Self-Organizing Maps*, Springer-Verlag, Berlin Heidelberg.
- Li, J. X. (2004), 'Visualization of high-dimensional data with relational perspective map', *Information Visualization* **3**, 49–59.
- Lowe, N. & Datta, A. (2005), 'A new technique for rendering complex portals', *IEEE Transactions on Visualization and Computer Graphics* **11**(1), 81–90.
- Matt Craighead, D. G. (2005), 'Anim8or'. Accessed 31th Oct 2005.
- Public Reports: Illiteracy rate and Illiterate Population* (n.d.). The UNESCO Institute for Statistics.
- Skupin, A. & Fabrikant, S. (2003), 'Spatialization methods: A cartographic research agenda for non-geographic information visualization', *cartography and Geographic Information Science* **30**(2), 95–119.
- Torgerson, W. S. (1952), 'Multidimensional scaling: I. theory and method', *Psychometrika* **17**, 401–419.
- Wu, Y. & Takatsuka, M. (2005), Geodesic self-organizing map, in 'Proceedings of Conference on Visualization and Data Analysis 2005', IS&T / SPIE.

Coordinated Perspectives and Enhanced Force-Directed Layout for the Analysis of Network Motifs

Christian Klukas

Falk Schreiber

Henning Schwöbbermeyer

Leibniz Institute of Plant Genetics and Crop Plant Research Gatersleben
Corrensstraße 3, 06466 Gatersleben, Germany,
Email: {klukas,schreibe,schwoebb}@ipk-gatersleben.de

Abstract

The analysis of network motifs, patterns of local interconnections with potential functional properties, has applications in many fields of science. Network motif analysis is particularly important for the exploration of biological networks. The distribution and multiple occurrences of motifs in a network, complicated by different concepts for the determination of motif frequency, create various difficulties in understanding and interpretation. We present a system that facilitates the analysis of network motifs by presenting coordinated perspectives simultaneously. A list of motifs supported by the network, visual representations of motifs of interest, a motif fingerprint and a visualisation of motif matches in the network are interwoven in a single interface. A motif and cluster-preserving force-directed layout algorithm supports the visual analysis of network motifs. This coordination of perspectives significantly enhances the explorative power of network motif analysis.

Keywords: network analysis, network motifs, coordinated perspectives, multiple views, information visualisation, graph drawing, force-directed layout

1 Introduction

The analysis of local structural properties of networks, patterns of local interconnections, is of interest to many research areas. These patterns, called *network motifs*, can be seen as the basic building blocks of complex networks (Milo, Shen-Orr, Itzkovitz, Kashtan, Chklovskii & Alon 2002). There are different definitions of a network motif. Some authors use this term to represent a set of related networks (Shen-Orr, Milo, Mangan & Alon 2002), for others it is simply one single small network (Wuchty, Oltvai & Barabási 2003). Motifs have been described as patterns of local interconnections which occur in networks at numbers significantly higher than those in randomised networks (Milo et al. 2002, Shen-Orr et al. 2002), or as patterns with functional properties (Schreiber & Schwöbbermeyer 2004). Nowadays, however, the term motif is widely used for a single

small connected network without consideration of statistical significance or functional properties (Sporns & Kötter 2004, Wuchty et al. 2003) and we adopt this notion in this work.

Interesting motifs have been shown to be present in complex networks of various fields. Network motif analysis is particularly important for the functional analysis of biological networks (Shen-Orr et al. 2002, Wuchty et al. 2003, Conant & Wagner 2003), where it helps to uncover important properties of these networks and assists scientists in understanding processes in organisms. The *feed-forward loop* motif has gained the most attention and has been studied theoretically and experimentally (Shen-Orr et al. 2002, Mangan & Alon 2003, Mangan, Zaslaver & Alon 2003). This motif has been shown to perform information processing tasks in cells (Mangan & Alon 2003, Mangan et al. 2003). Further examples for network motifs are the *single-input* motif which coordinates specific biological functions, such as the regulation of enzymes (Lee, Rinaldi, Robert, Odom, Bar-Joseph, Gerber, Hannett, Harbison, Thompson, Simon, Zeitlinger, Jennings, Murray, Gordon, Ren, Wyrick, Tagne, Volkert, Fraenkel, Gifford & Young 2002, Shen-Orr et al. 2002), the *multi-input* motif which coordinates pathways under several different conditions (Lee et al. 2002), and, as a special case of a multi-input motif the *b-fan* motif with exactly two regulators and two regulated elements (Milo et al. 2002, Dobrin, Beg, Barabási & Oltvai 2004). Figure 1 shows the structure of these motifs.

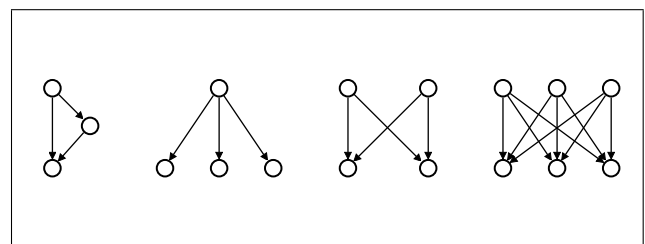


Figure 1: Structure of well-known network motifs (from left to right): feed-forward loop motif, single-input motif, b-fan motif and multi-input motif.

Copyright ©2006, Australian Computer Society, Inc. This paper appeared at Asia-Pacific Symposium on Information Visualization (APVIS 2006), Tokyo, Japan, February 2006. Conferences in Research and Practice in Information Technology, Vol. 60. K. Misue, K. Sugiyama and J. Tanaka, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

We would like to thank Franz J. Brandenburg (University of Passau, Germany) for the excellent cooperation and for granting usage of Gravisto. This work was supported by the German Ministry of Education and Research (BMBF) under grant 0312706A.

To our knowledge only a few systems exist that provide methods for the numerical and statistical analysis of motifs in networks. The *Mfinder 1.1* is a software tool for network motif detection in directed and undirected networks (Kashtan, Itzkovitz, Milo & Alon 2002). It computes the number of occurrences of a motif of restricted size in the target network and a uniqueness value, which is a lower bound of the number of times a subgraph appears in the network with completely disjoint groups of vertices. Furthermore, the statistical significance is determined on the basis of the number of occurrences of the motif in randomised networks. *Pajek* is a program for the

analysis and visualisation of large networks (Batagelj & Mrvar 2004) and is widely used in social network analysis. It contains an algorithm for calculating the frequencies of motifs. Sporns and Kötter used a MATLAB module for the computation of network motifs (Sporns & Kötter 2004).

The distribution and multiple occurrences of motifs in a network, complicated by different concepts for the determination of motif frequency, create various difficulties in understanding and interpretation and there are no systems supporting motif exploration. Here we discuss different motif-related information shown in multiple coordinated views and present a motif- and cluster-preserving layout. These methods are implemented in a system called MAVisto (Motif Analysis and Visualisation tool) which supports finding motifs of any size and exploring motif distribution and motif matches in an intuitive graphical way.

This paper is organised as follows: in Section 2 we define the graph model on which we operate, introduce concepts for the determination of motif frequency, an algorithm to find the matches in the target graph and a method for calculating the statistical significance of motifs. Section 3 presents the visualisation methods, i.e. the different perspectives used to display motif-related information such as motif table, motif view, motif fingerprint and motif matches and discusses the interaction between these perspectives. Section 4 presents enhancements of the well-known force-directed layout method which calculates motif- and cluster-preserving layouts. The MAVisto system is described in Section 5. Finally, Section 6 contains a general discussion and an example.

2 Definitions and Descriptions

2.1 Graph, Motif, Match

Graphs can be directed or undirected. In the remainder we consider directed graphs, however the presented method works also for undirected graphs. A directed graph $G = (V, E)$ consists of a finite set V of vertices and a finite set of edges $E \subseteq V \times V$ of edges where each edge $e = (u, v) \in E$ connects two vertices u and v . We consider loop free graphs, i.e. no edge connects a vertex with itself. Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are *isomorphic*, if (1) there exists a bijective mapping between the vertices in V_1 and V_2 , (2) there is an edge between two vertices of one graph if and only if there is an edge between the two corresponding vertices in the other graph. A graph $G' = (V', E')$ is a subgraph of a graph $G = (V, E)$ if $V' \subseteq V$, $E' \subseteq E \cap (V' \times V')$.

A *motif* as used here is a small connected graph. A *match* of a motif within a target graph G is a graph G' , which is (1) isomorphic to the motif and (2) a subgraph of G . See Figure 2 for a graph, a motif and a match of the motif.

2.2 Concepts for Determination of Motif Frequency

The frequency of a motif in a particular network is the number of different matches of this motif. The concepts for the determination of the frequency of a motif are based on different restrictions on sharing of network elements (vertices or edges) for the matches counted for the frequency. These concepts have different properties and are used to analyse different aspects of the motifs.

There are three reasonable frequency concepts (Schreiber & Schwöbbermeyer 2004). Concept \mathcal{F}_1 has no restrictions and counts every match of a motif. This concept gives a complete overview of all

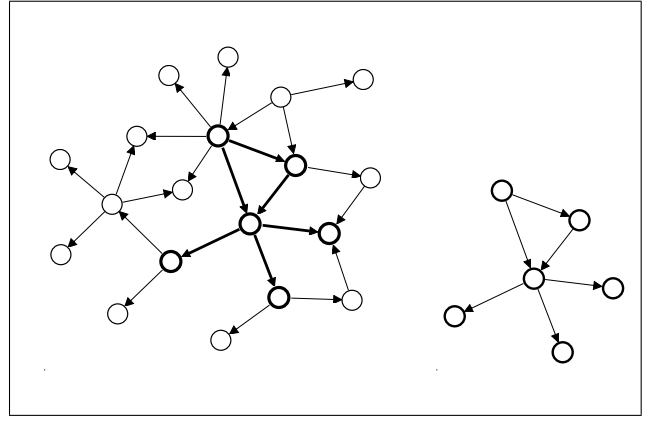


Figure 2: A graph (left) with a match of the motif shown right highlighted with thick lines.

possible occurrences of the motif even if elements of the target graph have to be used several times. Frequency concepts \mathcal{F}_2 and \mathcal{F}_3 restrict the reuse of graph elements shared by different matches counted for the frequency of a motif. Concept \mathcal{F}_2 is a common concept (Kuramochi & Karypis 2004, Vanetik, Gudes & Shimony 2002) that excludes sharing of edges and only allows edge disjoint matches. In concept \mathcal{F}_3 no sharing of graph elements is allowed, therefore all matches have to be vertex and edge disjoint. The advantage of concept \mathcal{F}_3 is that the matches of one motif can be seen as non-overlapping clusters. This clustering of the target graph allows specific analysis and navigation methods such as motif-preserving layout of the network.

The restrictions on the reuse of graph elements for concepts \mathcal{F}_2 and \mathcal{F}_3 have consequences for the determination of motif frequency in case of overlapping matches, as not all matches can be counted for the frequency. To determine the maximum number of different matches of a motif the maximum set of non-overlapping matches has to be calculated. This is known as the maximum independent set problem. Since this problem is \mathcal{NP} -complete (Garey & Johnson 1979), usually a heuristic is used to compute a lower bound for the frequency.

2.3 Motif Search Algorithm

We use a previously presented search algorithm (Schreiber & Schwöbbermeyer 2004) to identify all non-isomorphic motifs of a particular size supported by a target network. It furthermore detects all occurrences of these motifs. The algorithm uses a method that builds a tree of the motifs found in the target graph and traverses this tree, starting with the smallest motif. For the construction of the motif tree each motif m is assigned to one defined parent motif m' , from which m can be derived by extension of m' with one edge. Since more than one possibility exists for the generation of a motif through extension of a smaller motif, this assignment assures that each motif is discovered only once in the search process.

The traversal of the motif tree starts with the smallest motif consisting of two vertices and one connecting edge, whose occurrences in the network can be easily determined. The occurrences of a motif are iteratively extended by one edge using edges adjacent to them within the target network. The newly discovered motifs are kept for further extension. This procedure is repeated until the requested size for motifs is reached or there are no valid extensions at this branch of the traversal tree. A canonical label is calculated

to identify isomorphic motifs. Since the generation of new motifs by extension of already identified motifs is based on the topology of the target network, only motifs supported by the target graph are generated. This is important since the number of non-isomorphic motifs grows exponentially with increasing size of the motif (Harary & Palmer 1973), but in practice only a small fraction occurs in the network.

2.4 Statistical significance calculation

The statistical over-representation of motifs has been seen as a first indication for potential functional units and design principles that are present in a particular network (Shen-Orr et al. 2002, Milo et al. 2002). Z-score and P-value have been introduced for the analysis of statistical significance of motifs (Milo et al. 2002, Maslov, Sneppen & Alon 2003). The Z-score is defined as the difference of the frequency \mathcal{F}_1 of a motif in the target network and its mean frequency $\overline{\mathcal{F}_{1,r}}$ in a sufficiently large set of randomised networks, divided by the standard deviation σ_r of the frequency values for the randomised networks, see the following Equation.

$$\text{Z-score}(m) = \frac{\mathcal{F}_1(m) - \overline{\mathcal{F}_{1,r}(m)}}{\sigma_r(m)} \quad (1)$$

The randomised versions of the original network are generated using a random local rewiring algorithm which preserves the degrees of the vertices. In one step (v_1, v_2) and (v_3, v_4) are rewired in such a way that v_1 becomes connected to v_4 and v_3 to v_2 , provided that no such edge already exists in the network (Maslov & Sneppen 2002, Maslov et al. 2003). This rewiring step is repeated many times (e.g., $200 \times |E|$) to generate a properly randomised network.

The P-value of a motif m is defined as the probability P that the frequency of m in a randomised network is equal or larger to the frequency of m in the target network (Milo et al. 2002, Maslov et al. 2003).

3 The Perspectives of MAVisto

3.1 Overview

Coordinated perspectives are used for the comprehensive analysis of network motifs, see Figure 3. In Section 3.2 these perspectives are described and in Section 3.3 the interaction between them is discussed.

3.2 Perspectives

1. Motif Table

All resulting motifs of a search are listed in the motif table along with information about them, see Figure 4:

- Label**
For identification of motifs a unique label is used (canonical label).
- Size**
The size of each motif is given by the number of vertices and edges.
- Structure**
Information about the structure is given by the number of branches (buug) and cycles (cuug) in the underlying undirected graph. A branch is a vertex with a degree of 3 or higher. A cycle is a walk in which the initial vertex of the walk is also the terminal vertex of the walk.

(d) Frequency

The frequency values of each motif are listed for the three different frequency concepts \mathcal{F}_1 , \mathcal{F}_2 and \mathcal{F}_3 .

(e) Statistics

Information about the statistical significance is given by the Z-score and the P-value. In addition, mean value and standard deviation of motif frequency in randomised versions of the network are presented.

| Label | vertices | edges | buug | cuug | f1 | f2 | f3 | p-vs. r | z-score | mean | std |
|-------|----------|-------|------|------|----|----|----|---------|---------|-------|-------|
| PNOPF | 4 | 5 | 0 | 0 | 1 | 1 | 1 | 0.05 | 0.705 | 0.215 | 1.114 |
| VSVFX | 4 | 4 | 0 | 0 | 14 | 2 | 1 | 0.051 | 2.124 | 3.801 | 4.803 |
| VOPF | 4 | 5 | 1 | 1 | 2 | 1 | 1 | 0.061 | 2.481 | 0.193 | 0.728 |
| PUBTX | 4 | 4 | 0 | 1 | 9 | 3 | 2 | 0.062 | 1.92 | 4.387 | 2.403 |
| PNMF | 4 | 5 | 1 | 1 | 1 | 1 | 1 | 0.064 | 0.722 | 0.245 | 1.046 |
| PNRM9 | 4 | 4 | 0 | 1 | 11 | 5 | 4 | 0.066 | 1.802 | 5.98 | 2.786 |
| PNRPF | 4 | 5 | 1 | 1 | 1 | 1 | 1 | 0.066 | 0.525 | 0.305 | 1.325 |
| PNRPF | 4 | 5 | 1 | 1 | 1 | 1 | 1 | 0.071 | 1.185 | 0.164 | 0.705 |
| QSGCF | 4 | 5 | 1 | 1 | 3 | 1 | 1 | 0.073 | 1.947 | 0.398 | 1.409 |
| VSPCF | 4 | 5 | 1 | 1 | 2 | 1 | 1 | 0.073 | 0.909 | 0.489 | 1.868 |
| PNMHF | 4 | 5 | 1 | 1 | 1 | 1 | 1 | 0.081 | 0.641 | 0.267 | 1.144 |
| QASOX | 4 | 4 | 0 | 0 | 10 | 1 | 1 | 0.093 | 1.566 | 3.045 | 4.44 |
| PNODF | 4 | 5 | 1 | 1 | 1 | 1 | 1 | 0.096 | 0.25 | 0.52 | 1.92 |

Figure 4: Motif Table showing label, size, structure, frequency and statistics for motifs. The selected motif (blue) is shown in the motif view (Figure 5) and is marked in the motif fingerprint (Figure 6).

2. Motif View

The motif view provides a visual representation of motifs. Selected motifs are displayed with a layout computed by a force-directed layout method (Fruchterman & Reingold 1991), see Figure 5.

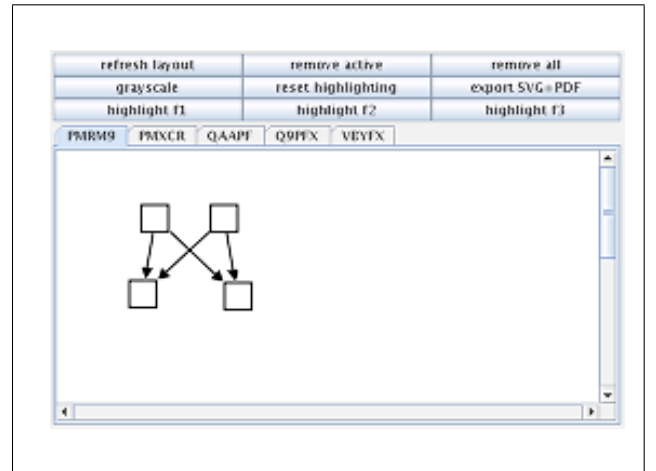


Figure 5: Motif View showing the structure of motifs. The displayed (active) motif is highlighted in the motif table (Figure 4) and marked in the motif fingerprint (Figure 6).

3. Motif Fingerprint

The motif fingerprint shows a motif frequency spectrum for the target network. The fingerprint comprises all motifs of a particular size. In the motif fingerprint, as shown in Figure 6, the motif frequency values of the motif table are plotted. The order of the motifs is defined by the alphanumeric order of the motif label, taking both the supported motifs and the motifs with frequency

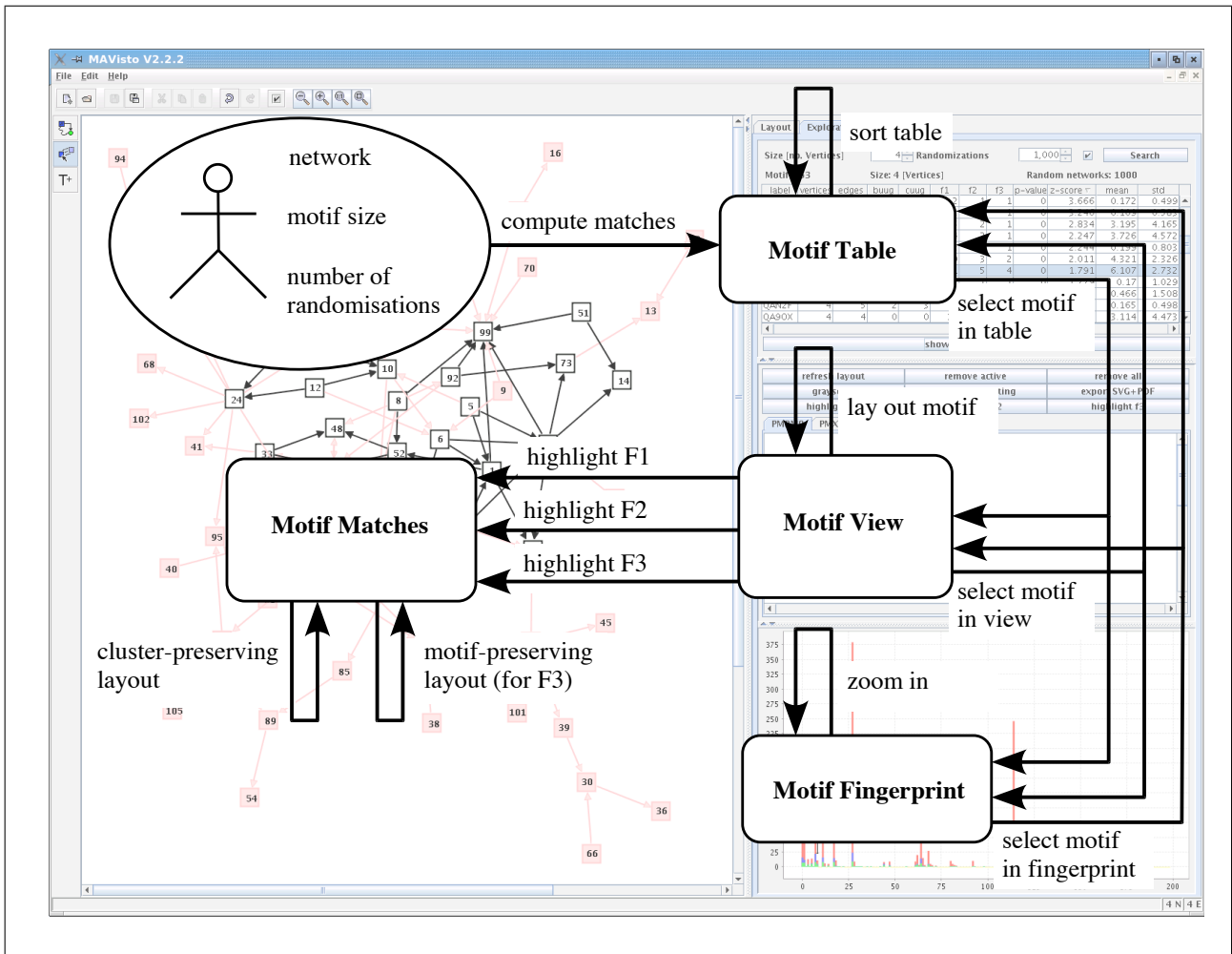


Figure 3: Use case diagram showing the different perspectives and possibilities for user interaction. For a screenshot of MAVisto see also Figure 17.

zero into account. This guarantees the comparability of motif fingerprints of different networks.

The values for the frequency concepts \mathcal{F}_1 , \mathcal{F}_2 and \mathcal{F}_3 are plotted as overlaid columns. The value for \mathcal{F}_1 is the red column in the background, which is covered at least partially by the blue column for \mathcal{F}_2 . The green column for the value for \mathcal{F}_3 is in front of the red and the blue column. This covering does not lead to a loss of information, since the frequency of a motif decreases monotonically for concepts \mathcal{F}_1 to \mathcal{F}_3 . For motifs not supported by the analysed network, an additional column is shown using the frequency value -1 as a placeholder (yellow).

4. Motif Matches

For the visual exploration of the occurrences of a motif within the network highlighting of the matches is provided, see Figures 7 - 9.

For frequency concept \mathcal{F}_1 , which does not restrict the sharing of network elements, the covering of network elements by motif matches is displayed. Frequency concept \mathcal{F}_2 allows sharing of vertices for different matches, but prohibits sharing of edges. Therefore, only the edges of each match are highlighted uniquely. For frequency concept \mathcal{F}_3 , where network elements are not shared by different matches, the elements of each match are uniquely highlighted.

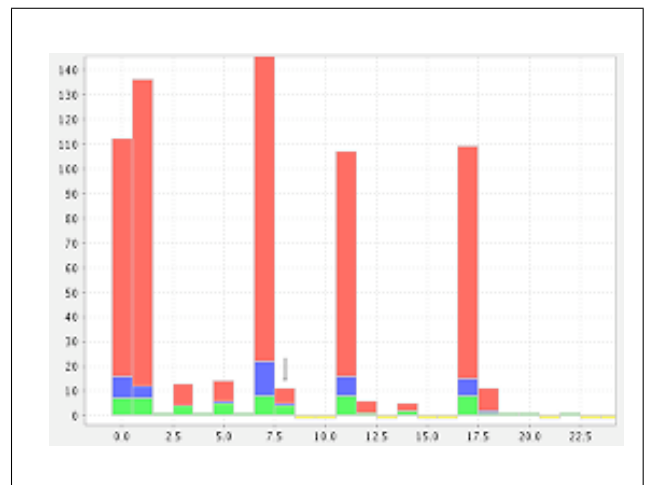


Figure 6: The Motif Fingerprint shows a motif frequency spectrum for the analysed network (here a zoom-in is shown). In the motif fingerprint the active motif is marked by an arrow on top of the column corresponding to the active motif of the motif view (Figure 5) and the motif table (Figure 4).

3.3 Interaction

1. Motif Table

The motif table is the starting point of the exploration process. After the search has been car-

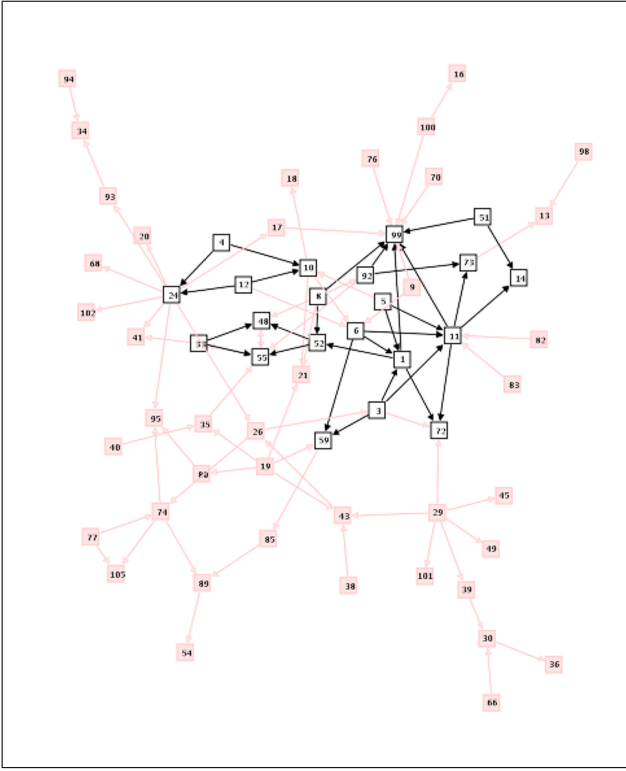


Figure 7: Network with highlighted motif matches of frequency concept \mathcal{F}_1 . The corresponding motif is shown in the motif view, see Figure 5. Network elements covered by matches are shown in black, elements not involved in any match are coloured light red.

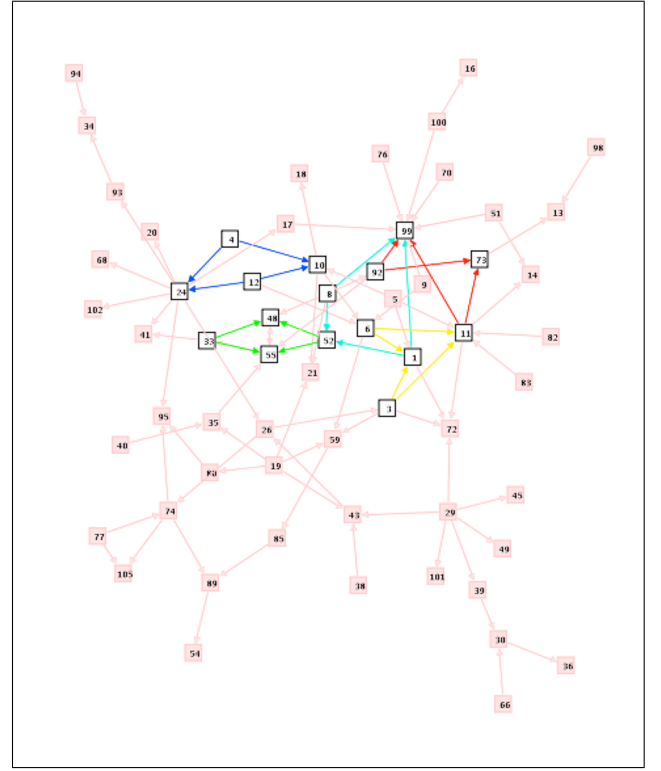


Figure 8: Network with highlighted motif matches of frequency concept \mathcal{F}_2 . The corresponding motif is shown in the motif view, see Figure 5. Edges of each motif are uniquely coloured. Vertices included in motif matches are displayed black and the remaining elements not involved in any match are coloured light red.

ried out it lists all discovered motifs of the given size along with additional information. The motif table supports sorting of the entries according to each column (alpha numerical for the label column and numerical sorting for all other columns). Furthermore, multilevel sorting is supported by keeping one (or more columns) sorted and sorting identical values by another column. Motifs can be selected for further analysis to be displayed in the motif view. The table supports the selection of one or multiple motifs to be displayed in the motif view. Additionally, the selection of a motif in the motif view or in the motif fingerprint leads to the selection of the corresponding row of the motif in the table, if contained in the table.

2. Motif View

In the motif view motifs are displayed as small networks allowing for visual analysis of its structure, showing one motif at a time (the active motif). The layout of a motif can be changed using a force-directed layout algorithm (Fruchterman & Reingold 1991) or can be manually adjusted. The user can switch between different motifs displayed in the motif view. The active motif is always marked in the motif fingerprint and is selected in the motif table. The matches of the active motif can be highlighted within the analysed network for the three frequency concepts. The user can select either a colour spectrum or a gray scale spectrum for highlighting of matches with disjunct network elements (i.e., edges for \mathcal{F}_2 , vertices and edges for \mathcal{F}_3).

3. Motif Fingerprint

The motif fingerprint allows a visual exploration of the motif frequency spectrum. The fingerprint

supports zooming into a particular region of interest. The label of the corresponding motif of a column is displayed as a tooltip text by moving the mouse cursor over this column. By clicking on a column, this column is marked by an arrow, the motif is displayed in the motif view and the corresponding row is selected in the motif table (if present in the table). Furthermore, motifs of the fingerprint not supported by the analysed network can also be selected to be displayed in the motif view.

4. Motif Matches

To obtain an overview of the network of interest it can be laid out using a force-directed placement algorithm (Fruchterman & Reingold 1991). The implementation of the algorithm offers the possibility to interactively change the parameters affecting the layout. To explore the distribution of the different motif occurrences the matches of the motifs are highlighted within the target network. For concept \mathcal{F}_1 the matches can overlap and the edges and vertices cannot be uniquely assigned to one match. Therefore, the individual matches are not marked, but the coverage of the network elements by matches is shown. The matches for frequency concept \mathcal{F}_2 and \mathcal{F}_3 can be uniquely highlighted by marking the edges (\mathcal{F}_2), or both vertices and edges (\mathcal{F}_3), of a match individually.

4 Enhanced Force-Directed Layout

Motifs structure a network. For example, motif matches may be relatively isolated parts of the network with only few connections to other matches or

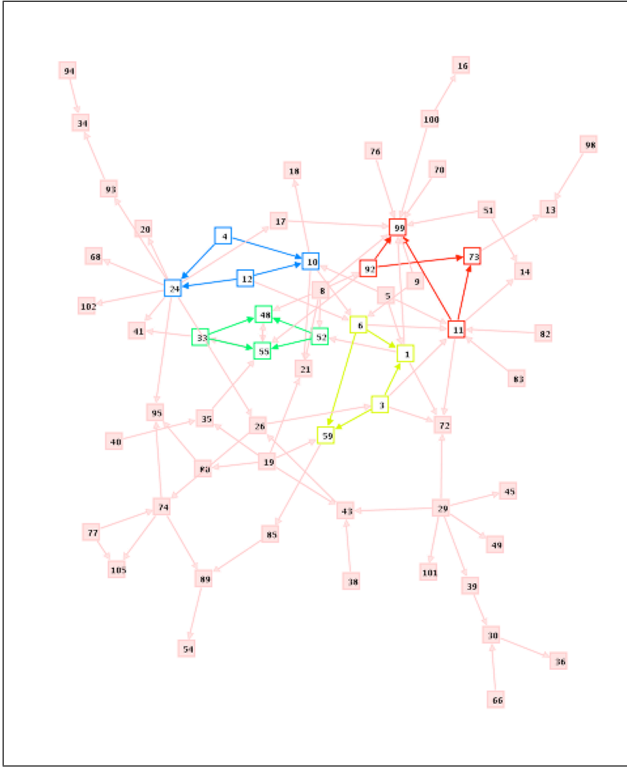


Figure 9: Network with highlighted motif matches of frequency concept \mathcal{F}_3 . The corresponding motif is shown in the motif view, see Figure 5. Each match is highlighted by colouring the vertices and edges, the remaining vertices and edges are coloured light red.

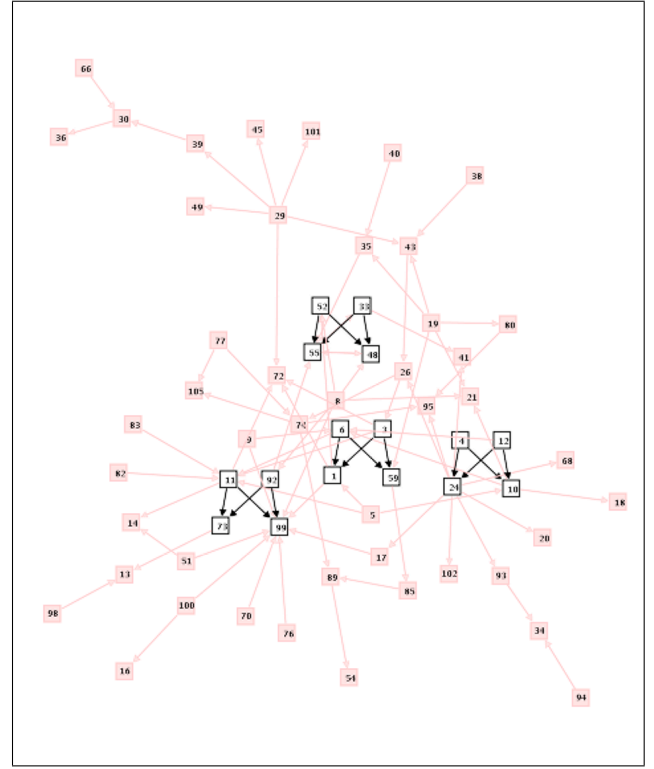


Figure 10: Here the same matches as in Figure 9 are shown, but the network is laid out by preserving the layout of the motif for each match (motif-preserving layout).

the remaining network, or may be heavily connected to them. Furthermore, all matches of a specific motif separate the network into two parts: network elements covered by motif matches and uncovered network elements. In this section we discuss enhancements of the layout of a network to improve the visual analysis of network motifs based on a flat (non-hierarchical) clustering, and extensions of the motif-preserving force-directed layout algorithm presented in (Klukas, Koschützki & Schreiber 2005).

A force-directed layout method (Eades 1984, Fruchterman & Reingold 1991) works by computing a 'force' vector \vec{f} for each vertex u . For each edge from vertex u to a vertex v an attractive force is computed such that vertex u gets an optimal distance to v . For all vertices v not connected to u repulsive forces are computed. The vector \vec{f} of the vertex u is the sum of the attractive and repulsive forces. Each vertex is then moved by a small amount in the direction of the corresponding force. This is repeated until the sum of all force vectors falls below some threshold or a time limit is reached.

A motif-preserving layout for disjoint motif matches (e.g. based on frequency concept \mathcal{F}_3) has been described previously (Klukas et al. 2005). Every match is uniformly laid out corresponding to a given layout, e.g. the layout in the Motif View. The algorithm works as follows: An initial placement of the vertices is computed which assigns random positions to all vertices that do not belong to a motif match. Then all vertices of the matches are placed with the given layout. A random move vector is calculated and applied to all vertices that belong to a certain match such that at the end of the initial layout every match is placed on a different position without overlapping another match. The modified force-directed layout loop calculates the usual force vector for each vertex

not belonging to a motif match. For all vertices belonging to a particular match a uniform force vector is calculated which is the average of the forces for all vertices of the match. All vertices of the match are moved at once, therefore the relative layout of the vertices within the match does not change. Figure 10 shows a result of the motif-preserving layout algorithm. In addition to the existing algorithm we found that the introduction of higher repulsive forces between vertices of a motif match and other vertices improves the overall layout and better separates the matches from the network, see Figures 11 and 12.

To further improve the visual analysis of motifs in networks we introduce a new cluster-preserving layout. A cluster-graph $G_C = (V_C, E_C)$ to a given graph $G = (V, E)$ is a graph where each node¹ $n \in V_C$ represents a subgraph G_n of G and each link $l \in E_C, l = (n_i, n_j)$ represents the set of edges of G which connects vertices of the subgraph G_i with the subgraph G_j . We consider simple cluster-graphs where each vertex of G belongs to one node of G_C and each edge of G belongs to either one node or one link of G_C . Figures 13 and 14 show a graph G and a cluster-graph G_C , respectively. Here the size of a node is proportional to the number of vertices in the corresponding subgraph and the width of a link between two nodes is proportional to the number of edges between the corresponding subgraphs.

The idea is to use a layout of the cluster-graph G_C to obtain a better layout of the graph G which helps in a visual separation of the clusters. Therefore another initial step is added: First the cluster-graph has to be laid out. Any graph drawing algorithm can be used depending on the structure of the cluster-graph, see for example (Battista, Eades, Tamassia & Tollis 1999). Then the layout of the graph itself is

¹Vertices and edges of the cluster-graph are called nodes and links, respectively.

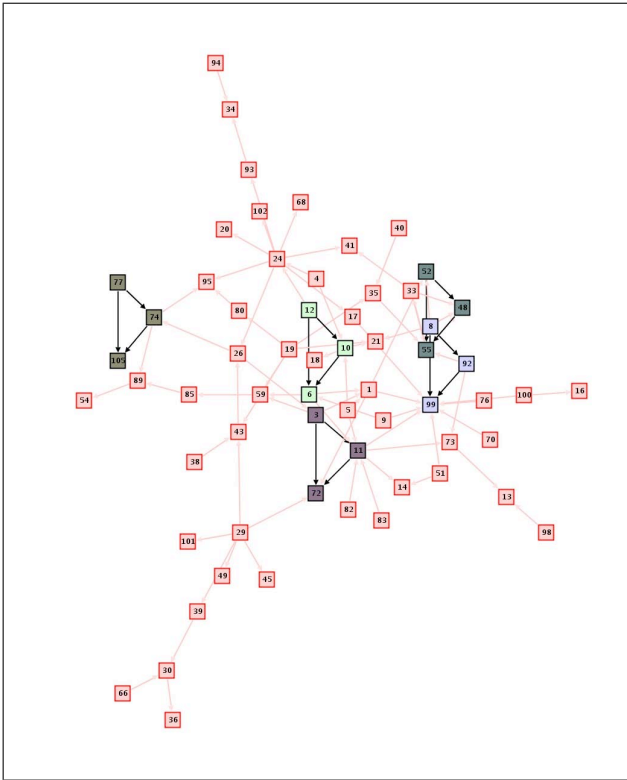


Figure 11: Matches of the feed-forward loop motif laid out by the motif-preserving layout.

computed. Additional gravity forces are introduced in the force-directed layout method. Beside attractive and repulsive forces the gravity force is also added to the force vector \vec{f} of each vertex. The coordinates of the nodes of the cluster-graph are the gravity centres for the gravity forces of the corresponding vertices of the graph. Depending on the strength of this force the clusters are more or less preserved: a force close to zero results in a typical force-directed layout whereas a high force clusters vertices around the corresponding gravity centres, see Figures 15 and 16. This method gives an application or user a great flexibility for the layout of the cluster-graph. An alternative method which does not use an additional layout of the cluster-graph but uses additional vertices without fixed coordinates as gravity centres is described by Eades and Huang (Eades & Huang 2000).

Cluster-preserving layouts can be used for two analysis tasks: (1) to get a better understanding of the interconnections between motif matches and (2) to visually study the coverage of the network by matches.

(1) To better understand the function of a motif match and also the functional role of the motif it is important to help in analysing if motif matches are relatively separate parts of the network with only few connections to other matches or the remaining network, or if such matches have many connections. Figure 11 shows a typical example where the connections of motif matches are hard to analyse. However, using the cluster-preserving layout it is possible to place all matches apart from the remaining network and emphasise the connections of a match to other vertices, see Figure 13.

(2) Individual motif matches aggregate into a sub-network of overlapping matches also called a *network theme*. Network themes can be tied to specific biological phenomena and may represent more fundamental network design principles (Zhang, King, Wong, Goldberg, Tong, Lesage, Andrews, Bussey, Boone &

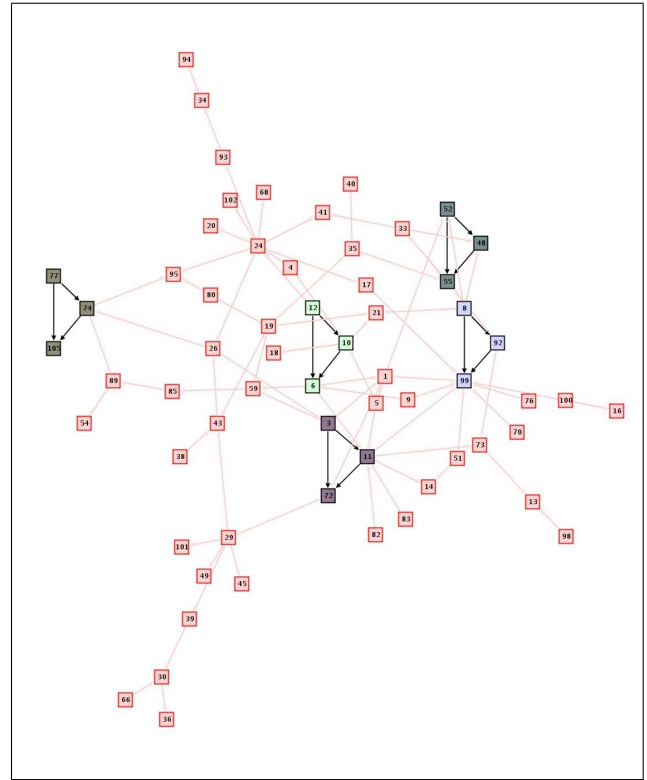


Figure 12: The same motif matches as in Figure 11 are shown, however, increased repulsive forces between matches and the remaining vertices separate the motif matches clearly from the remaining network and enhance the readability of the diagram.

Rot 2005). Examples of important network themes have been shown in an integrated *S. cerevisiae* interaction network (Zhang et al. 2005) and (as combination of several motifs) in the *E. coli* transcriptional regulatory network (Dobrin et al. 2004). Representing networks in terms of network themes provides a useful simplification of complex biological relationships (Zhang et al. 2005). By clustering vertices into two clusters (vertices covered by a match and uncovered vertices) and placing these two clusters in different areas, the network theme can be easily shown, see Figure 15.

5 The MAVisto System

Most of the described features are implemented in MAVisto, the Motif Analysis and Visualisation tool (Schreiber & Schwöbbermeyer 2005). MAVisto is written in Java and is freely available as a Java Webstart application at <http://mavisto.ipk-gatersleben.de/>. It is based on Gravisto (Bachmaier, Brandenburg, Forster, Raitner & Holleis 2005), an editor for graphs and a toolkit for implementing graph visualisation algorithms and supports the Pajek-net (Batagelj & Mrvar 2004) and the GML-file (Himsolt 2000) formats. Furthermore, it offers graph editor functionality for network manipulation and for creation of a network from scratch. Networks are interpreted as either directed or undirected using this property for all edges. The improved cluster-preserving layout is currently under evaluation and will be included in the next release of MAVisto.

The running time of the motif search algorithm depends on both the size of the analysed network and the size of the motifs. For searching motifs of size three to five the algorithm has been optimised. Here,

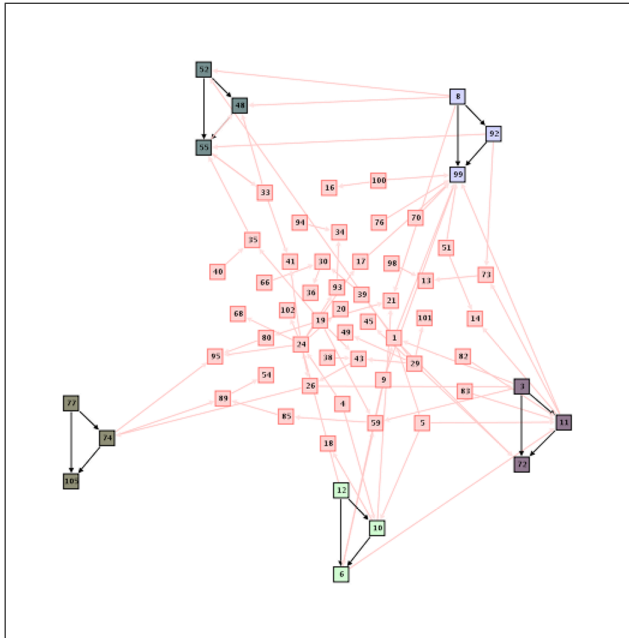


Figure 13: Cluster layout for frequency concept \mathcal{F}_3 . The layout of the corresponding cluster-graph is shown in Figure 14.

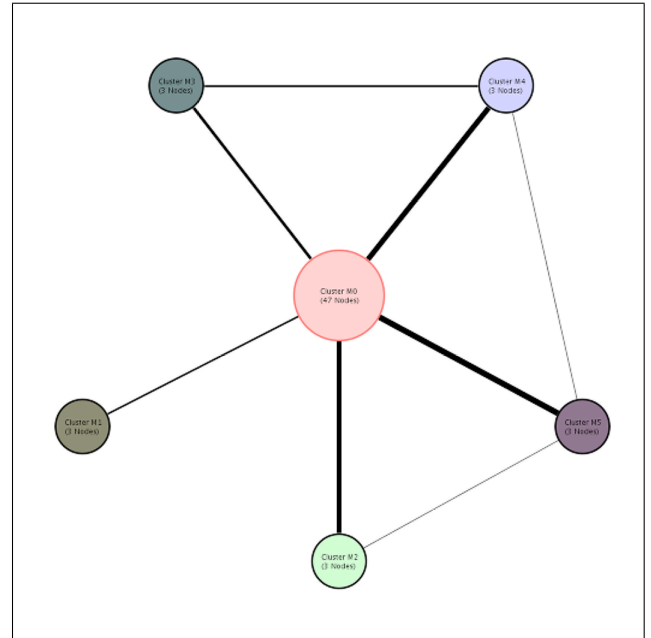


Figure 14: The layout of the cluster-graph corresponding to the graph in Figure 13.

the determination of the canonical label is accelerated by the use of a lookup-table and analysis of motifs can be reasonably applied for sparse networks with up to several hundred vertices. The other important factor affecting the running time of the search process for all motifs is the number of randomisation of the network. To obtain significant results at least a thousand networks have to be considered (Milo et al. 2002, Maslov et al. 2003). However, the running time of the algorithms for visualisation and interactive navigation between the different perspectives is much faster for these networks and motif sizes as these algorithms are independent of the search. For example sorting the motif table takes $O(n \log n)$ and computing the motif fingerprint $O(n)$ if n is the number of motifs.

6 Discussion

We have presented coordinated perspectives for the analysis of network motifs and a cluster-preserving force-directed layout algorithm. Most parts of this methodology has been implemented in MAVisto. The development of the presented methods has been carried out in close collaboration with users from biology and the tool is used to analyse different biological networks such as protein-protein interaction, gene regulatory and metabolic networks. Here we discuss the usage of the system on a practical example.

Network motif analysis with MAVisto starts by loading the network of interest. Here we use a network with information about transcriptional regulation of genes of *E. coli*. To get an overview of the network structure, the network can be automatically laid out or manually adjusted. After the layout is calculated for our *E. coli* network, different unconnected components appear: one big connected component which comprises most of the network elements and some smaller components. Moreover, one frequently occurring network topology is noticeable that consists of one central vertex pointing to numerous other vertices.

For the search for network motifs, parameters such as the size of motifs to look for and the number of randomised networks to consider for the calculation

of the statistical significance have to be adjusted. For this network we start the search for motifs with four vertices and use 1000 randomised networks for statistics calculation. Once the search has finished, the motifs supported by the network along with information about them are listed in the Motif Table and the frequency distribution is shown in the Motif Fingerprint. 18 motifs with four vertices out of 199 theoretically possible motifs have been found in the *E. coli* network covering a broad range of frequency, that now can be interactively explored. The motifs in the table can be sorted for example by their statistical significance, by their frequency or by structural properties such as number edges. Motifs found to be interesting can be analysed by displaying their structure in the Motif View. In the *E. coli* network four motifs were found having a Z-score above 10. Displaying these motifs in the Motif View reveals that one is the b-fan motif and the other three motifs are extensions of the feed-forward loop motif. Motif matches of selected motifs can be highlighted in the studied network for the three frequency concepts to analyse their overall distribution and which network elements are involved. In our network the well-known b-fan motif has the highest frequency of the four most significant motifs for all three frequency concepts. Visualising the covering of network elements by matches of the b-fan motif reveals that one big motif theme is build by overlapping matches, see Figure 17.

Now other motifs found to be interesting can be explored by going back to the Motif Table and Motif View. The different views of MAVisto are coordinated as the selection of a motif in one view on the right side also highlights this motif in the other two views. The Motif Table is the starting point of the exploration process where the search process is started and where the results are presented. The Motif View is the starting point for the visual exploration of motif matches. This way, the user first can inspect the structure and significance of the motifs and then start to explore the matches within the network.

MAVisto has so far been used to study networks in the size of dozens to several hundred vertices and motif sizes of three to five vertices. The motif search may always take a long time especially if several hundred randomised networks are compared for the statistical

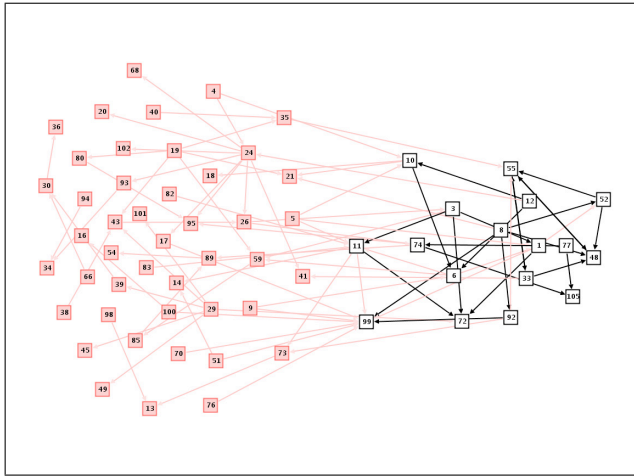


Figure 15: Cluster layout for frequency concept \mathcal{F}_1 . Strong cluster-forces separate both clusters in the drawing. The right side shows the network theme, the subnetwork covered by motif matches.

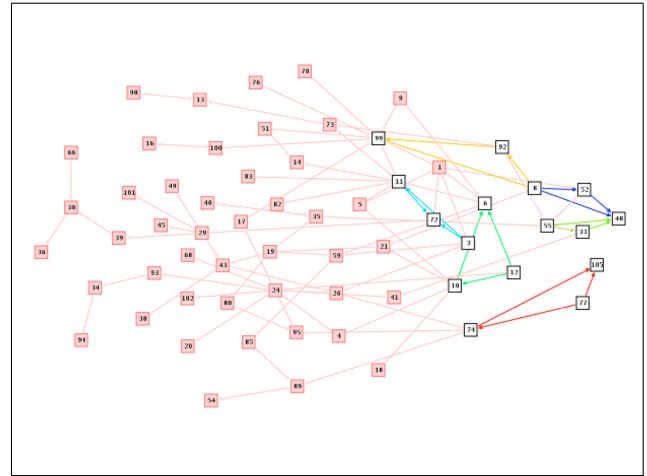


Figure 16: Cluster layout for frequency concept \mathcal{F}_2 . In contrast to Figure 15 lower cluster-forces have been used.

analysis. However, the interaction between the different perspectives as well as the layout for this network size is fast. An important question is the scalability of our approach to analyse larger networks with thousands of vertices. Two main problems have to be considered: finding all motif matches and presenting the results in a readable way. To address the first question we are working on a parallel version of the search algorithm. The second problem is a general problem in information visualisation and graph drawing: how to communicate vast amounts of information to the user. In MAVisto several methods have been implemented to provide overview, zoom and filter as well as details on demand. However, for large networks new abstraction techniques are necessary such as folding and unfolding of matches and sampling to approximate information from small parts of the network. Currently we are focusing on biological networks as our users are from the life science community. However, network motif analysis has applications in many fields of science from biology to electrical engineering to sociology and we are interested using our system in these fields.

References

- Bachmaier, C., Brandenburg, F. J., Forster, M., Raitner, M. & Holleis, P. (2005), Gravisto: Graph visualization toolkit, in 'Proceedings of the International Symposium on Graph Drawing (GD 2004)', Vol. 3383 of *Lecture Notes in Computer Science*, Springer, pp. 502–503.
- Batagelj, V. & Mrvar, A. (2004), Pajek - Analysis and Visualization of Large Networks, in M. Jünger & P. Mutzel, eds, 'Graph Drawing Software', Springer, pp. 77–103.
- Battista, G. D., Eades, P., Tamassia, R. & Tollis, I. G. (1999), *Graph Drawing: Algorithms for the Visualization of Graphs*, Prentice-Hall.
- Conant, G. C. & Wagner, A. (2003), 'Convergent evolution of gene circuits', *Nature Genetics* **34**(3), 264–266.
- Dobrin, R., Beg, Q. K., Barabási, A.-L. & Oltvai, Z. N. (2004), 'Aggregation of topological motifs in the Escherichia coli transcriptional regulatory network', *BMC Bioinformatics* **5**(1), 10.
- Eades, P. (1984), 'A heuristic for graph drawing', *Congressus Numerantium* **42**, 149–160.
- Eades, P. & Huang, M. L. (2000), 'Navigating clustered graphs using force-directed methods.', *Journal of Graph Algorithms Applications* **4**(3), 157–181.
- Fruchterman, T. & Reingold, E. (1991), 'Graph drawing by force-directed placement', *Software - Practice and Experience* **21**(11), 1129–1164.
- Garey, M. R. & Johnson, D. S. (1979), *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, New York.
- Harary, F. & Palmer, E. M. (1973), *Graphical Enumeration*, Academic Press, New York.
- Himsolt, M. (2000), 'Graphlet: design and implementation of a graph editor', *Software - Practice and Experience* **30**(11), 1303–1324.
- Kashtan, N., Itzkovitz, S., Milo, R. & Alon, U. (2002), Mfinder tool guide, Technical report, Department of Molecular Cell Biology and Computer Science & Applied Mathematics, Weizman Institute of Science.
- Klukas, C., Koschützki, D. & Schreiber, F. (2005), 'Graph pattern analysis with PatternGravisto', *Journal of Graph Algorithms and Applications* **9**(1), 19–29.
- Kuramochi, M. & Karypis, G. (2004), Finding frequent patterns in a large sparse graph, in 'SIAM International Conference on Data Mining (SDM-04)'.
- Lee, T. I., Rinaldi, N. J., Robert, F., Odom, D. T., Bar-Joseph, Z., Gerber, G. K., Hannett, N. M., Harbison, C. T., Thompson, C. M., Simon, I., Zeitlinger, J., Jennings, E. G., Murray, H. L., Gordon, D. B., Ren, B., Wyrick, J. J., Tagne, J. B., Volkert, T. L., Fraenkel, E., Gifford, D. K. & Young, R. A. (2002), 'Transcriptional regulatory networks in Saccharomyces cerevisiae', *Science* **298**(5594), 799–804.
- Mangan, S. & Alon, U. (2003), 'Structure and function of the feed-forward loop network motif', *Proceedings of the National Academy of Sciences* **100**(21), 11980–11985.

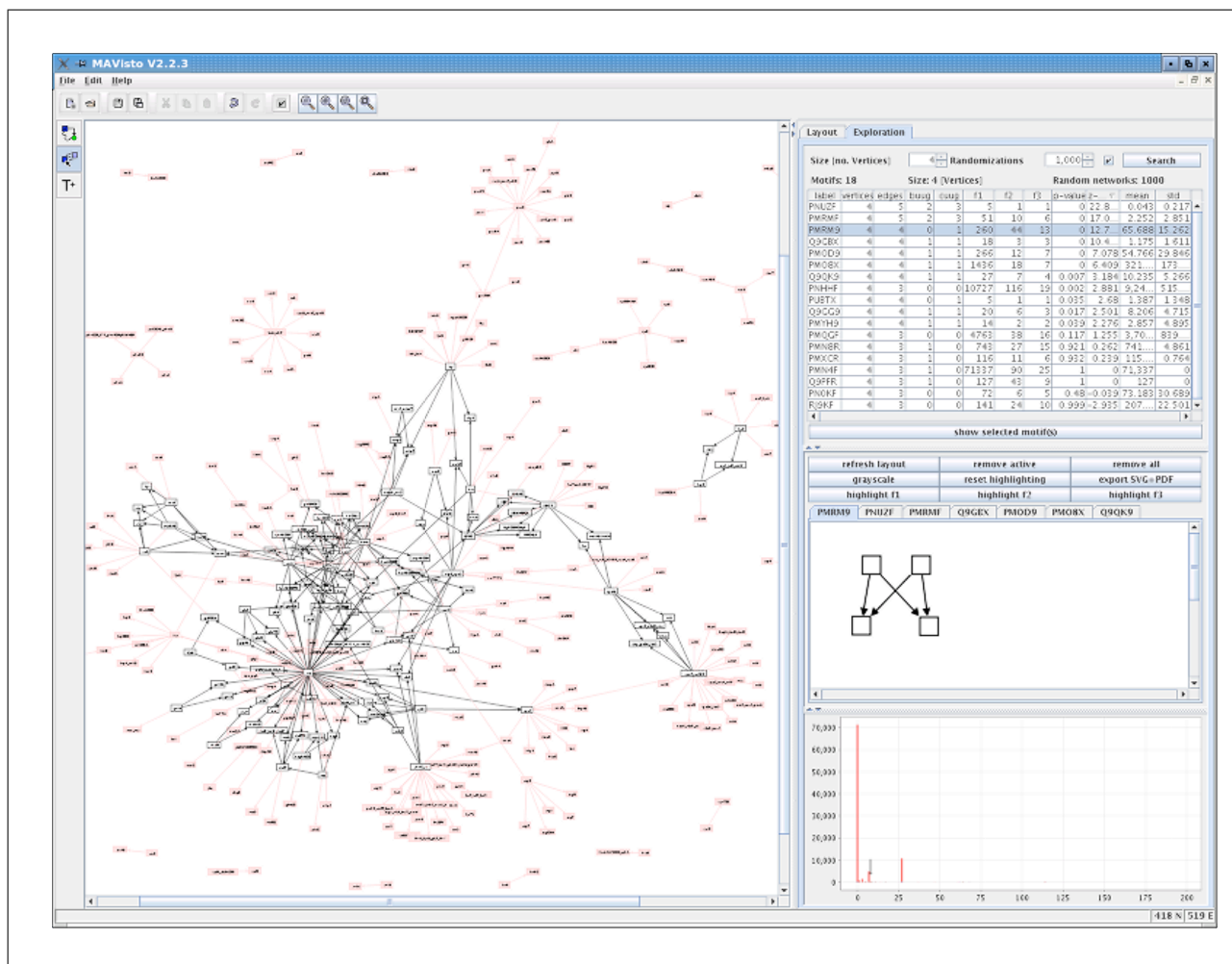


Figure 17: Screenshot of the system showing a step of the analysis of the *E. coli* transcriptional regulatory network.

- Mangan, S., Zaslaver, A. & Alon, U. (2003), 'The coherent feedforward loop serves as a sign-sensitive delay element in transcription networks', *Journal of Molecular Biology* **334**(2), 197–204.
- Maslov, S. & Sneppen, K. (2002), 'Specificity and stability in topology of protein networks', *Science* **296**, 910–913.
- Maslov, S., Sneppen, K. & Alon, U. (2003), Correlation profiles and motifs in complex networks, in S. Bornholdt & H. G. Schuster, eds, 'Handbook of Graphs and Networks: From the Genome to the Internet', Wiley-VCH, Berlin, pp. 168–198.
- Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D. & Alon, U. (2002), 'Network motifs: Simple building blocks of complex networks', *Science* **298**(5594), 824–827.
- Schreiber, F. & Schwöbbermeyer, H. (2004), Towards motif detection in networks: Frequency concepts and flexible search, in 'Proceedings of the International Workshop on Network Tools and Applications in Biology (NETTAB'04)', pp. 91–102.
- Schreiber, F. & Schwöbbermeyer, H. (2005), 'MAVisto: a tool for the exploration of network motifs', *Bioinformatics* **21**(17), 3572–3574.
- Shen-Orr, S., Milo, R., Mangan, S. & Alon, U. (2002), 'Network motifs in the transcriptional regulation network of *Escherichia coli*', *Nature Genetics* **31**(1), 64–68.
- Sporns, O. & Kötter, R. (2004), 'Motifs in brain networks', *PLoS Biology* **2**(11), e369.
- Vanetik, N., Gudes, E. & Shimony, E. (2002), Computing frequent graph patterns from semistructured data, in 'Proceedings of the IEEE International Conference on Data Mining (ICDM)', pp. 458–465.
- Wuchty, S., Oltvai, Z. N. & Barabási, A.-L. (2003), 'Evolutionary conservation of motif constituents in the yeast protein interaction network', *Nature Genetics* **35**(1), 176–179.
- Zhang, L. V., King, O. D., Wong, S. L., Goldberg, D. S., Tong, A. H., Lesage, G., Andrews, B., Bussey, H., Boone, C. & Rot, F. P. (2005), 'Motifs, themes and thematic maps of an integrated *saccharomyces cerevisiae* interaction network', *Journal of Biology* **4**(2), Epub.

Generation of Relevance Maps and Navigation in A Digital Book

Katsuhiro Ikeda¹, Kozo Sugiyama¹, Isamu Watanabe², Kazuo Misue³

¹ Japan Advanced Institute of Science and Technology, Asahidai 1-1, Nomi, Ishikawa, 923-1292 Japan

² Fujitsu Laboratories Ltd, Kamiodanaka 4-1-1, Nakahara, Kawasaki, Kanagawa, 211-8588 Japan

³ University of Tsukuba, Ten-nodai 1-1-1, Tsukuba, Ibaraki, 305-8573 Japan

{Katsu-i, sugi}@jaist.ac.jp, watanabe.isamu@jp.fujitsu.com, misue@cs.tsukuba.ac.jp

Abstract

This paper describes how to design a digital book for a new science called 'Knowledge Science'. We prepare several types of navigation facilities for browsing the book. Specially, we generate relevance maps by mining useful information from large volume of textual data and visualizing it. We evaluate the quality of the generated relevance maps through questionnaires to the authors. Also, we evaluate the usability of prepared navigation facilities through questionnaires to the users.

Keywords: digital book, relevance map, navigation

1 Introduction

Human society is becoming increasingly complex. If science remains segmented into specialized disciplines, we cannot deal effectively with multifaceted problems that we now face. We need a new integrative science that is founded on the deep understanding of humanity and society.

In view of this need, the School of *Knowledge Science*, Japan Advanced Institute of Science and Technology has embarked upon a new initiative that aims to discover both theoretical and practical principles of knowledge management. The School has enlisted not only natural scientists and engineers but also social scientists and humanities scholars; the number of stuffs of the faculty has become more than thirty at the moment. The School provides master and doctoral programs to educate knowledge engineers, coordinators, and scientists who are expected to become pioneers of the knowledge society (JAIST 2005).

Recently, to advertise the knowledge science and increase its social recognition, the School has published a book 'Knowledge Science' (Sugiyama, Nagata and Shimojima 2002), where we have edited it with a special structural and visual form to well organize complex, non-integrated information relating to the knowledge science. This publishing has been very successful: we have sold more

than 10,000 copies. This book also has been translated into Korean (Sugiyama, Nagata and Shimojima 2005). (See Figure. 1.)



(a) Japanese edition



(b) Korean edition



(c) Web book

Figure.1: Books 'Knowledge Science'

After publishing the book, to open it to the public, we have developed a Web Book 'Knowledge Science' using technologies of hypertext, *relevance maps* (i.e. text mining and graph visualization), and navigation. We have calculated and evaluated relevance maps with questionnaires to the authors (i.e. members of the faculty). Also, we have evaluated the usability of navigation facilities (including relevance maps) prepared for the Web Book.

In this paper, we describe the design of book 'Knowledge Science' and the development and evaluation of Web book 'Knowledge Science'.

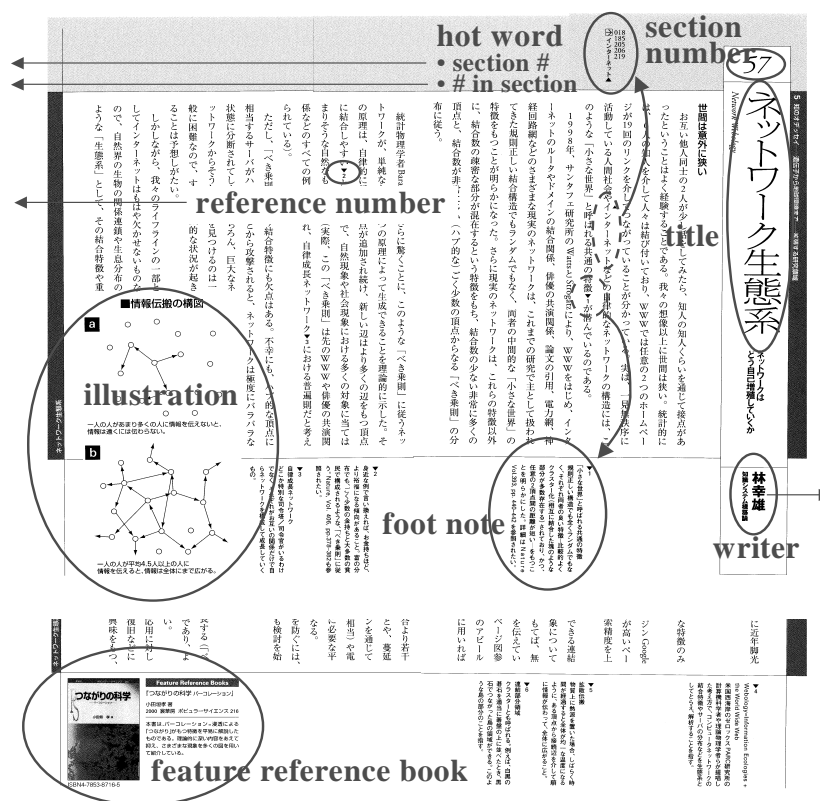


Figure.2: Design of book 'Knowledge Science'

2 How to Design a Book 'Knowledge Science'

Requirements for designing a book 'Knowledge Science' are enumerated as follows. It is desirable that every member of the faculty joins to write the book. Though members' research domains are diversified, core concepts and mutual relationships in complex non-integrated materials should be clarified. Descriptions and terminologies should be plain enough for target readers such as undergraduate and graduate students, and businesspersons. Information such as author introductions and related books is important for the reader to know the context and to learn further.

The editing process we took is as follows. Due to the insufficiency of development of the knowledge science, it is difficult to make its systematic descriptions at the moment. We have decided to design the book as collections of keywords (important terminologies) for the knowledge science. Every member has proposed several keywords, and then editors have selected 64 keywords among them and divided them into five groups. The design of structural and visual form for this book is devised collaborating with stuffs of the Editorial Engineering Laboratories Co. so that it can satisfy the above requirements (see Figure.2). The explanation of each keyword (*title*) forms a *section* with two or four pages. A group of keywords forms a *chapter*. This book has a hypertext-like structure where *hot words* (or core terms) specified by the author are linked to other sections or parts

of other sections. Important terminologies are explained in *footnotes*. A *feature reference book* and *references* are convenient for the reader to learn in more detail.

3 Developing a Digital Book on the Web

Based on the success of publishing the book, we have decided to develop a digital book that is open to the public on the Web (Sugiyama, Nagata and Shimojima 2005). A book on the Web seems the best way for increasing social recognition of the knowledge science. Also, a digital form seems the best way to describe this developing science because we need to revise and elaborate the book readily and repeatedly. Moreover, we can make hyper-links to the outer world, which must be important specifically for the knowledge science.

The digital version basically inherits the structure of the previous version. In each page of the digital book the user can make *Keyword search*, and choose one of hyper-links to go to other pages such as *Contents*, *Texts* (chapters, sections), *Archives* (feature reference books, authors' introduction etc.), *Indices* (terms, authors, references, publishers etc.), *Site maps* (tables for sections, feature books, figures etc.), or *Access maps* (i.e. relevance maps). Through these facilities, it is expected that the user can browse the book flexibly and get information quickly.

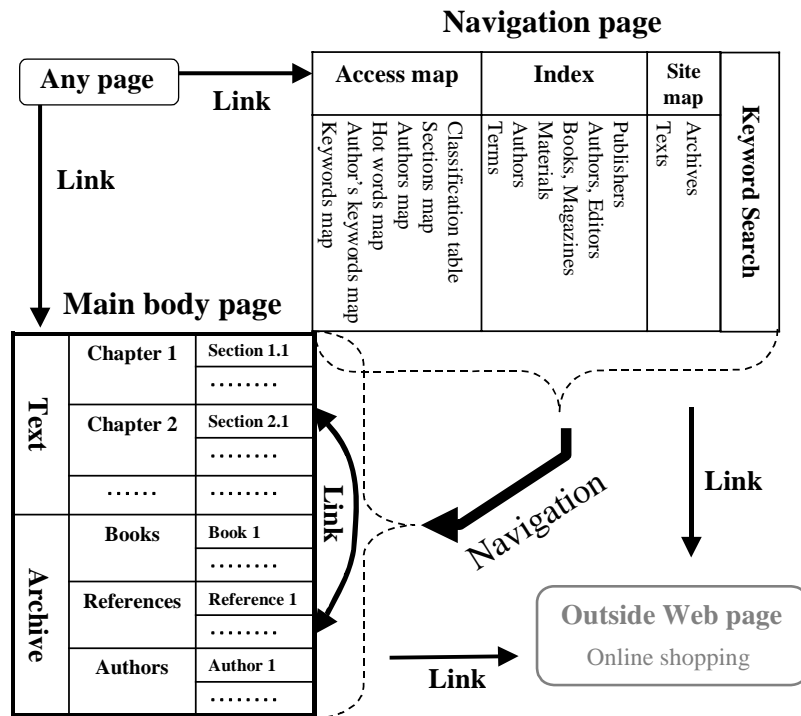


Figure 3: Hyper-link structure of the digital book

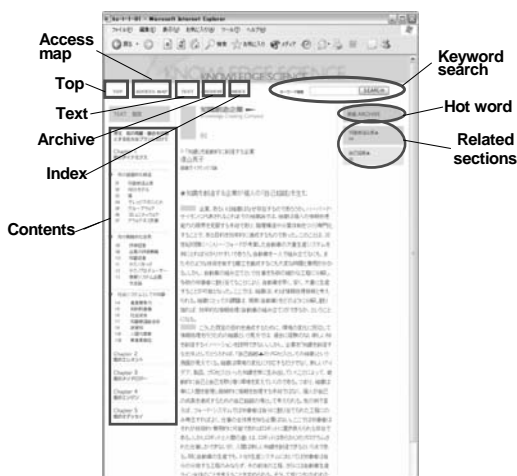


Figure 4: An example of Section page

Different types of access maps have been generated using techniques for text mining and graph drawing. Through the maps, it is expected that we can easily grasp the images of relational structures of the whole book, which cannot be captured from the individual sections and by browsing link to link. Consequently, the user can be navigated to local places while seeing globally. How to generate access maps will be explained in the next section. Figure 3 illustrates a hyper-link structure of the digital book: This consists of main body pages (chapters and sections) and navigation pages (access maps, indexes, site maps). The reader can go to one of pages of Text, Archive, Access map, Index, and Site map from any page. Figure 4 shows an example of Section page.

4 Generation of Relevance Maps

A text-mining and visualization tool called ACCENT (Watanabe and Misue 1999, Misue and Watanabe 1999, Watanabe 2005) has been developed in the Fujitsu Laboratories Ltd. that facilitates us to mine useful information from large volume of textual data and visualize it. The tool statistically calculates relevance among text segments and words in documents using co-occurrence of words, and then visualizes them as a relevance map. Through the map, we can easily grasp traits and trends in the whole documents, which cannot be captured from the individual documents. We made relevance analyses of textual data appeared in the digital version of 'Knowledge Science' by using ACCENT.

4.1 Calculation of Relevance among Documents

The textual data is divided into specific *units*: in this paper units correspond to sections. Then, words are extracted from each unit, and the frequency of occurrence of each word is obtained. A frequency table obtained in this manner is referred to as a *frequency dictionary*. Table 1 shows the frequency of words (top 100) occurred in 'Knowledge Science'.

Based on the frequency dictionary, the *importance* $S(t, w)$ of word w in a text unit t is calculated by

$$S(t, w) = p(t, w) \log \frac{p(t, w)}{q(w)} \quad (1)$$

where $p(t, w)$ is the ratio of the frequency of word w to the frequency of all words in unit text t ; and $q(w)$ is the ratio of the frequency of word w to the frequency of all words in the all unit texts T (Watanabe and Misue 1999). (Instead, we can use TFIDF (Tokunaga 1999).) Usually, $S(t, w)$ is normalized so that its square sum is 1, and expressed as $S'(t, w)$. The *importance* $S'(T, w)$ of word w in T is defined by

$$S'(T, w) = \sum_{t \in T} S'(t, w) \quad (2)$$

Several types of *relevance matrices* R_{wt} , R_{ww} , R_{tt} between a word and a text unit, between words, and between unit texts, respectively, are calculated by

$$\begin{aligned} R_{wt} &= R_{tw} = S'(t, w) \\ R_{ww}(w_i, w_j) &= \sum_{t \in T} S'(t, w_i) S'(t, w_j), \quad w_i, w_j \in W \\ R_{tt}(t_i, t_j) &= \sum_{w \in W} S'(t_i, w) S'(t_j, w), \quad t_i, t_j \in T \end{aligned} \quad (3)$$

where W and T are sets of words and text units, respectively, specified in advance.

| | | | |
|---------|-----------|-----------|---------|
| 57 知識 | 27 変化 | 22 知 | 18 経験 |
| 52 研究 | 27 利用 | 22 知識創造 | 18 場 |
| 46 情報 | 26 活動 | 22 特定 | 18 発見 |
| 41 意味 | 26 間 | 22 表現 | 18 部分 |
| 41 技術 | 26 構築 | 22 目的 | 18 理論 |
| 41 必要 | 26 対象 | 21 レベル | 17 基本 |
| 40 重要 | 26 点 | 21 過程 | 17 機能 |
| 40 創造 | 25 概念 | 21 時間 | 17 空間 |
| 38 システム | 25 企業 | 21 手法 | 17 研究開発 |
| 38 可能 | 25 自分 | 21 知識科学 | 17 言葉 |
| 37 科学 | 25 実現 | 21 例 | 17 従来 |
| 35 問題 | 25 発展 | 20 ネットワーク | 17 生産 |
| 32 社会 | 24 可能性 | 20 課題 | 17 要素 |
| 32 組織 | 24 結果 | 20 形式 | 17 領域 |
| 31 環境 | 24 現在 | 20 最近 | 17 力 |
| 30 関係 | 24 支援 | 20 知的 | 16 科学技術 |
| 30 人間 | 24 状況 | 20 複雑 | 16 科学的 |
| 30 図 | 24 世界 | 19 活用 | 16 学 |
| 30 存在 | 23 データ | 19 形 | 16 期待 |
| 30 多く | 23 共有 | 19 処理 | 16 決定 |
| 29 モデル | 23 全体 | 19 定義 | 16 結合 |
| 29 考え | 22 コンピュータ | 19 特徴 | 16 現象 |
| 29 分野 | 22 関連 | 19 認識 | 16 効果 |
| 27 プロセス | 22 個人 | 19 方法 | 16 行為 |
| 27 開発 | 22 構造 | 18 解決 | 16 人 |

Table 1: The frequency of words (top 100) occurred in ‘Knowledge Science’

4.2 Visualization

An undirected graph corresponding to a relevance matrix usually has edges between almost all pairs of vertices, or generated graphs are usually too dense. Therefore, we need to reduce the number of edges for good visualization of core structure (Chen 1999). Our strategy for thinning out edges is to remove an edge one by one in the increasing order of their relevance values while maintaining the connectedness of the graph. If removing a candidate edge violates the connectedness, the edge is not removed. How many edges should be removed is given as a parameter. If a resulting graph becomes a tree (minimum spanning tree (Even 1979)), this procedure finishes.

We have developed a method to visualize several types of relevance among words and text units (i.e. *relevance maps*) by extending techniques in graph drawing (Sugiyama 2002).

The spring embedder (Eades 1984) is expressed by the following two types of forces between vertices:

$$\begin{aligned} f_s &= c_s \cdot \log(d / d_0) \\ f_r &= c_r / d^2 \end{aligned} \quad (4)$$

where d is the standard distance between vertices, d_0 is the standard strength of springs, and c_s and c_r are parameters. In our model, relevance r is related to the distance d_0 and the spring strength c_s by

$$\begin{aligned} d_0 &\propto 1/r \\ c_s &\propto r. \end{aligned} \quad (5)$$

The first extension can be effective for placement so that a distance between vertices of which relevance is large becomes short, and the second extension can be effective for absorbing global strains with local places where relevance is small.

5 Evaluation of Relevance Maps

We generated five types of relevance maps:

- Sections map*: Relevance map among all section titles
- Keywords map*: Relevance map among *important* words appeared in the whole book
- Author’s keywords map*: Relevance map among important words used by the respective author
- Hot words map*: Relevance map among all hot words appeared in the whole book
- Authors map*: Relevance map among all authors

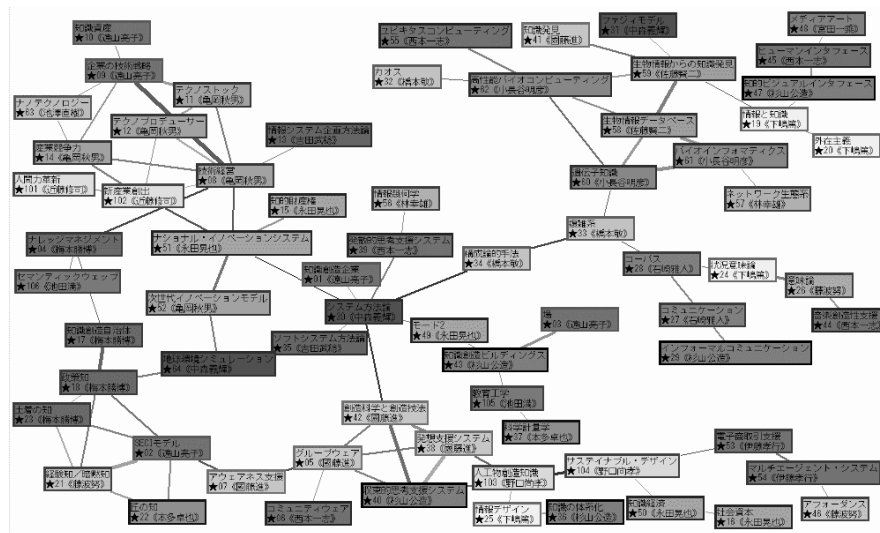
For example, Figures 4(a) to 4(d) show Sections map, Keywords map, Hot words map, and Authors map, respectively (Ikeda 2004).

It is very important to evaluate the quality of these maps in terms of adequateness and comprehensiveness to express traits and trends in the book. For the evaluation of these maps, we have delivered questionnaire to all the authors (twenty-three) to give rating values (lowest: 1, low: 2, middle: 3, high: 4, highest: 5) and asked free description of their impressions. All authors are colleagues in the same school and know each other’s research domain(s). Therefore, authors are the best evaluators for the quality of generated relevance maps.

Table 2 shows values of mean and SD of rating for eight questions (Ikeda 2004). Rating is considerably high

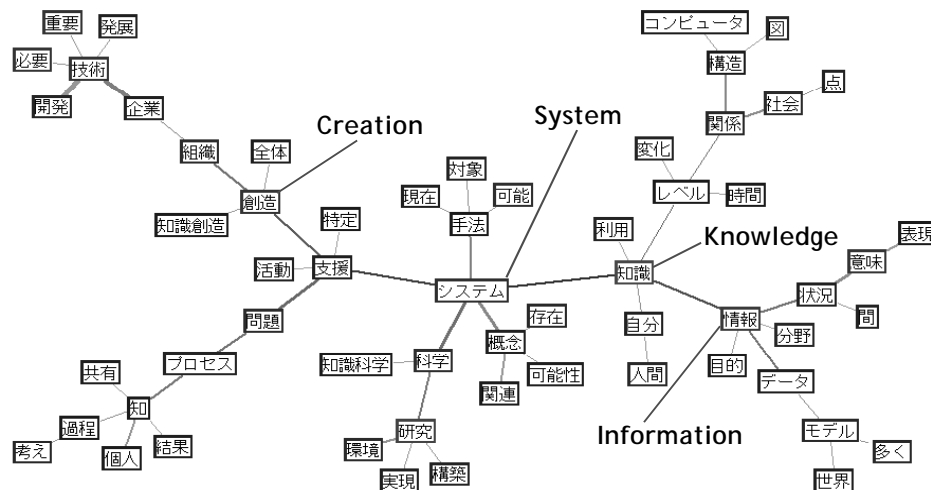
(3.0~3.66) in average in all questions. This means that the quality of generated maps is considerably reliable. However, many persons have pointed out that most maps are minimum spanning trees and so the reduction of edges is too much to see relationships between non-connected vertices. Rating in the authors map is high (questions 1 and 2). This is because authors know their domains each other

and are interested in how their own position in the networks, and who are the key persons. Rating in the hot words is low (question 7) of which reason might be that most hot words are specified by other authors. Rating in the other author's map is low. This might be that each author feels the incongruity because others' keywords are different from own keywords.



(a) Sections map (70 sections)

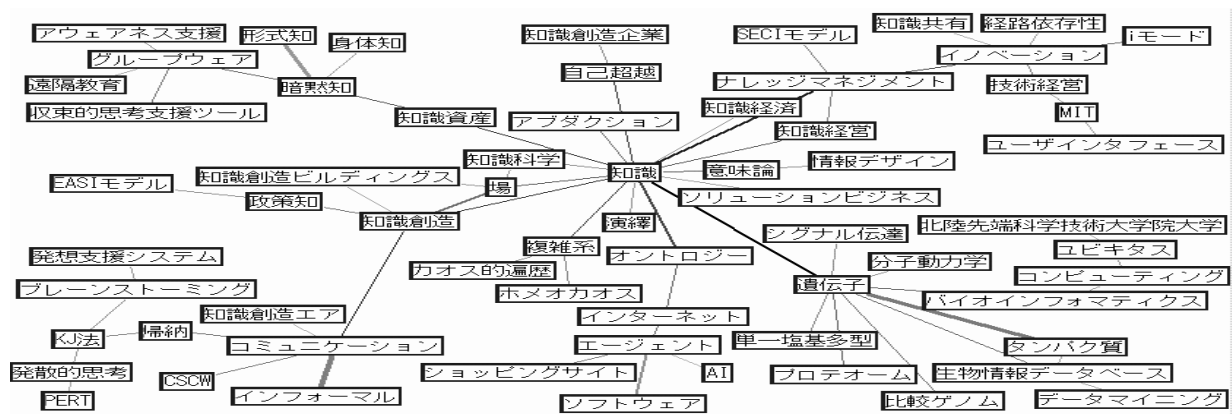
The relevance among sections is visualized. Each vertex label contains section title and author's name. Each author is distinguished with different colours. The width of edge means the degree of relevance.



(b) Keywords map (top 60 words)

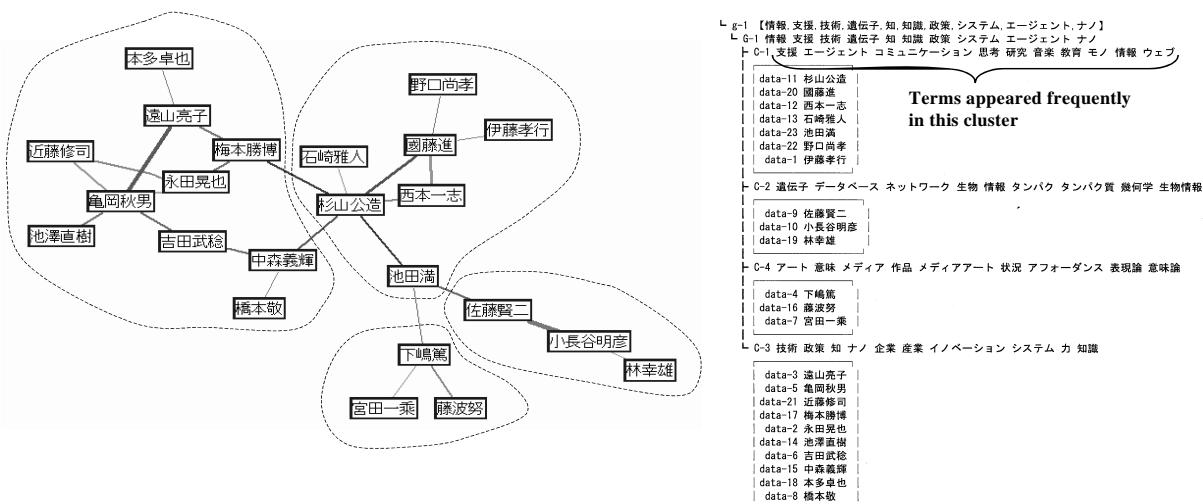
The frequency is distinguished with colours. The width of edge means the degree of relevance. It can be seen that important keywords in the knowledge science such as 'knowledge', 'information', 'system', and 'creation' are situated in the position of hub. Also, the unit of word is relatively small comparing with hot words in Figure 4(c).

Figure 4.: Generated relevance maps (continued)



(c) Hot words map (64 words)

The width of edge means the degree of relevance. It seems that the relevance among technical terms are well represented, for example term 'knowledge' is situated at the position of the most important hub. The authors specify these terms. It is very difficult to mine technical terms appropriately.



(d) Authors map (23 authors) and clustering tree

Relevance map (left) and cluster tree (right) among authors are represented. ACCENT can show clustering trees and terms that appear frequently in each cluster. In the relevance map clusters are well represented. The width of edge the degree of relevance.

Figure 4: Generated relevance maps

| Question | Rate | |
|--|------|------|
| | Mean | SD |
| 1. <i>Your relating part in authors map</i> is agreeable? | 3.66 | 0.94 |
| 2. <i>Authors map</i> is reasonably interesting on the whole? | 3.55 | 0.68 |
| 3. <i>Sections map</i> is reasonably interesting on the whole? | 3.33 | 1.15 |
| 4. <i>Keywords map</i> is reasonably interesting on the whole? | 3.30 | 0.90 |
| 5. <i>Your keywords map</i> is agreeable? | 3.22 | 0.91 |
| 6. <i>Your relating part in sections map</i> is agreeable? | 3.11 | 0.87 |
| 7. <i>Hot words map</i> is reasonably interesting on the whole? | 3.00 | 0.94 |
| 8. <i>Other author's keywords map</i> is reasonably interesting? | 3.00 | 0.86 |

Table 2: Results of questionnaire to evaluate the quality of generated maps

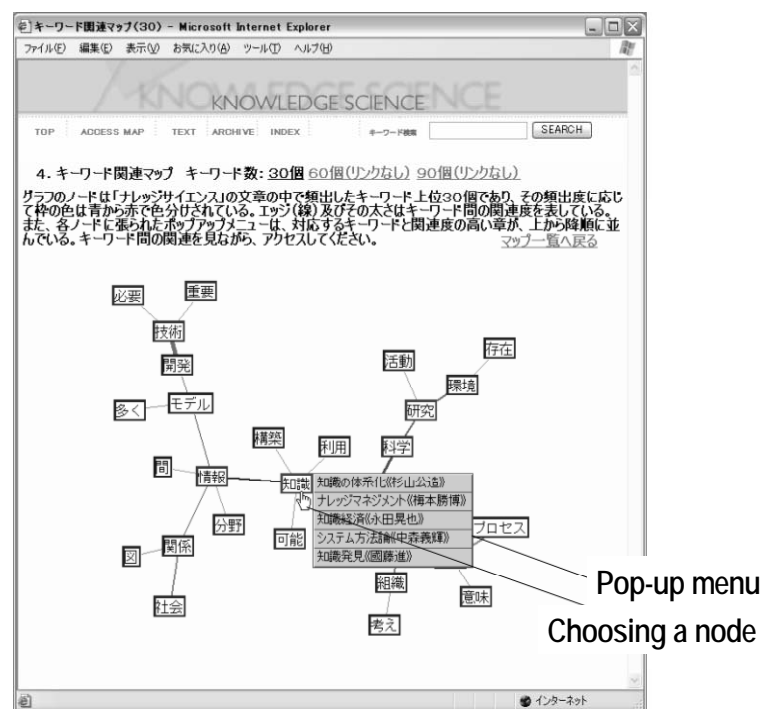


Figure 5: An access map for keywords

6 Evaluation of Navigation Facilities

For navigation facilities, Access map, Index, and Site map are prepared as shown in the Navigation page box in Figure 3. We have used the obtained relevance maps as access maps for navigation where pop-up menus provide links to related sections, authors etc. We can choose an adequate link among them. Figures 5 and 6 show visual interfaces for various types of navigation facilities. Figure 5 shows an interface for an access map (*Keywords map*) and pop-up menu. Other access maps also are used in a way similar to the keywords map.

Figures 6(a) to 6(d) shows other types of navigation facilities (Index, Site map): *Classification matrix*, *Sections table*, *Authors list*, and *Terms Index*, respectively. In the *Classification matrix*, sections are classified according to two dimensions for objectives (Community, Organization, Individual, Lives, and Nature), and objects (Knowledge, Knowledge creation, Knowledge communication/practice/assets, Knowledge creation support, Knowledge representation, Knowledge search/discovery, and Knowledge systematisation).

The *Sections table* provides links not only to every section but also archives and figures. In the *Authors list*, the user can choose a link to the page introducing to a selected

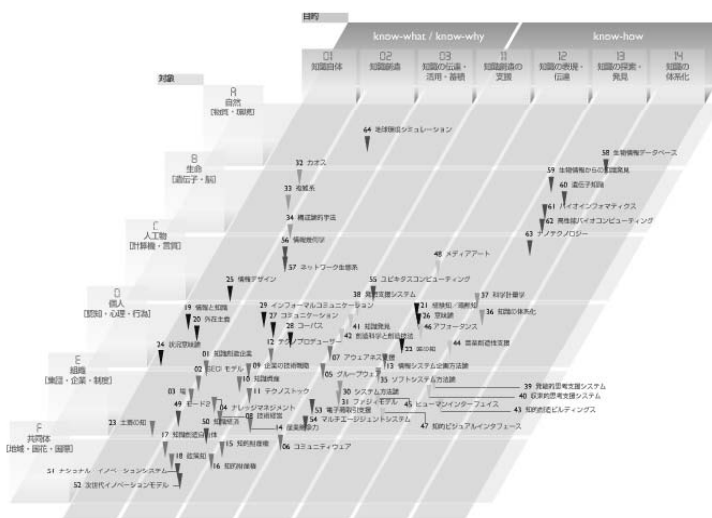
author. With the *Terms index* the user can search terms by selecting an alphabet or hiragana as a searching key.

For the usability evaluation of navigation facilities in the developed digital book, we have carried out questionnaires to the users (twenty graduated students in JAIST) to give rating a degree (five levels) and free descriptions of impression for each navigation facility after reading the digital book by using various navigation facilities.

Table 3 shows results of the questionnaires (Ikeda 2004). Among the access maps, rating for the usability of *Hot words map* is the highest (4.0) while rating for the quality of *Hot words map* by the authors is the lowest (3.0) as seen in Table 3. Ratings for the usability of access maps are relatively lower than those for indexes and site maps. This is contrary to our expectation. Specially, rating for

Sections map is low (2.4) and rating for *Sections table* is high (4.3) though both seem to be similar. The authors can read meanings of *Sections map* representing relationships among sections employing their professional knowledge. Therefore, *Sections map* might be more interesting than *Sections table* for the authors. However, the ordinary users cannot do this, instead, they might need some given framework that is considered to express editors' intension to read the book.

Table 3 shows results of the questionnaires (Ikeda 2004). Among the access maps, rating for the usability of *Hot words map* is the highest (4.0) while rating for the quality of *Hot words map* by the authors is the lowest (3.0) as seen in Table 3. Ratings for the usability of access maps are



(a) Classification matrix

| TEXT | ARCHIVE | INDEX |
|--------------------------|--------------------|-------|
| 序文 知の発展・融合を目指す巨大なプランにむけて | | |
| Chapter 1 知のダイナミクス | Chapter 1 知のダイナミクス | |
| 今知の発展的な動向 | 今知の発展的な動向 | |
| 01-知の発展的な動向 | 01-知の発展的な動向 | |
| 02-知の発展的な動向 | 02-知の発展的な動向 | |
| 03-知の発展的な動向 | 03-知の発展的な動向 | |
| 04-知の発展的な動向 | 04-知の発展的な動向 | |
| 05-知の発展的な動向 | 05-知の発展的な動向 | |
| 06-知の発展的な動向 | 06-知の発展的な動向 | |
| 07-知の発展的な動向 | 07-知の発展的な動向 | |
| 08-知の発展的な動向 | 08-知の発展的な動向 | |
| 09-知の発展的な動向 | 09-知の発展的な動向 | |
| 10-知の発展的な動向 | 10-知の発展的な動向 | |
| 11-知の発展的な動向 | 11-知の発展的な動向 | |

(b) Sections table

| 作者名 | 氏名 | プロフィール |
|------|------------------|--------|
| 山本浩一 | Yamamoto Hiroshi | プロフィール |
| 山本浩一 | Yamamoto Hiroshi | プロフィール |
| 山本浩一 | Yamamoto Hiroshi | プロフィール |
| 山本浩一 | Yamamoto Hiroshi | プロフィール |
| 山本浩一 | Yamamoto Hiroshi | プロフィール |
| 山本浩一 | Yamamoto Hiroshi | プロフィール |
| 山本浩一 | Yamamoto Hiroshi | プロフィール |
| 山本浩一 | Yamamoto Hiroshi | プロフィール |
| 山本浩一 | Yamamoto Hiroshi | プロフィール |
| 山本浩一 | Yamamoto Hiroshi | プロフィール |

(c) Authors list

| キーワード | 検索結果 |
|----------|-----------|
| 知の発展 | 12 |
| 知の融合 | 17 |
| 知のダイナミクス | 30 |
| 知の発展的な動向 | 1-100 |
| 知の発展的な動向 | 2-100, 25 |
| 知の発展的な動向 | 18, 20 |
| 知の発展的な動向 | 5-100 |

(d) Terms index

Figure 6: Various navigation facilities

| | Question | Rating | |
|-----------------|--|--------|------|
| | | Mean | SD |
| Access map | 1. <i>Hot words map</i> is useful for reading the map? | 4.0 | 0.97 |
| | 2. <i>Authors map</i> is useful for reading the book? | 3.4 | 1.27 |
| | 3. <i>Sections map</i> is useful for reading the book? | 2.4 | 0.94 |
| | 4. <i>Keywords map</i> is useful to reading the book? | 2.2 | 1.1 |
| | 5. <i>Classification matrix</i> is effective for reading the book? | 2.9 | 0.91 |
| Index, Site map | 6. <i>Sections table</i> is useful for reading the book? | 4.3 | 0.85 |
| | 7. <i>Authors list</i> is useful for reading the book? | 3.7 | 0.90 |
| | 8. <i>Contents</i> are useful for reading the book? | 3.4 | 0.83 |
| | 9. <i>Indexes</i> are useful for reading the book? | 3.4 | 0.94 |

Table 3: Results of questionnaire to evaluate the usefulness of access

relatively lower than those for indexes and site maps. This is contrary to our expectation. Specially, rating for *Sections map* is low (2.4) and rating for *Sections table* is high (4.3) though both seem to be similar. The authors can read meanings of *Sections map* representing relationships among sections employing their professional knowledge. Therefore, *Sections map* might be more interesting than *Sections table* for the authors. However, the ordinary users cannot do this, instead, they might need some given framework that is considered to express editors' intension to read the book.

7 Related Work and Our Approach

A hierarchically organized hypertext is a dominant model for publishing electronic books. Useful authoring tools are providing support for creating hierarchical hypertexts (WBT Systems 2005, WebCT 2005 etc.). There have existed many works on concept-based and map-based navigation in Web-based electronic books. They are classified into vertical and horizontal navigation (Brusilovsky and Rizzo 2002) where automatic linking techniques such as word-level similarity (Rizzo et al. 2000), ontology (Crampes and Ranwez 2000), self organizing map (Rizzo et al. 1999) and so on.

Our approach utilizes word-level similarity techniques for automatic and byhand linking, and combines concept-based network navigation and map-based horizontal navigation. Strengths of this paper are that resulting relevance maps are verified through evaluation by authors and the usability of various navigation facilities are evaluated by users.

8 Concluding Remarks

We have published a book of which title is 'Knowledge Science' where we have adopted a specific structural and

visual form (i.e. hypertext-like form) to edit the book. This is because the knowledge science is a new integrative science and therefore we need to well exhibit relationships among complex and non-integrated materials provided by the authors who are staffs of the School of Knowledge Science, JAIST. This attempt has been very successful. We have been able to sell more than 10,000 copies of it. Moreover, Korean edition of the book has been published.

Next, we have aimed to develop a digital book (hypertext) that is open to the society on the Web. In this version we have intended to well navigate the reader in reading this hypertext because two problems are often pointed out in browsing hypertexts: lost the position and cognitive overload.

To overcome the problems, we have prepared various types navigation facilities that can afford global views of the whole documents in keywords, authors, sections and so on. Specially, we have made efforts to generate several types of relevance maps that can be used as access maps for navigation. We have used software called ACCENT that mine useful information from large volume of textual data and visualize it. Graphs corresponding to generated relevance are usually too dense. We have applied the reduction of edges of graphs before visualization.

We have evaluated the quality of generated relevance maps through questionnaires to the authors. As the results, we have concluded that the quality of the relevance maps is not bad, rather good. This might mean that specialists can *read* interesting information from the generated relevance maps. It should be noted that many authors have pointed out the lack of relationships among vertices, which is caused by reducing edges too much.

Then, we have evaluated the usability of navigation facilities through questionnaires to the users. Our expectation have been that access maps are effective to

navigate the users well to read the book. However, the results of the evaluation have revealed that the usability of indexes and site maps are relatively better than the usability of access maps in spite of the good results of evaluation about the quality of access maps.

As conclusions, we remark the followings:

- (1) We need to study more about how many edges should be reduced to well visualize a perspective of given data.
- (2) Generated access maps are sufficiently interesting for authors but not useful for the users. It seems that the user rather like to be given a framework a priori as an authorized view. This gap provides a very important hint for navigation. In the first phase, indexes and site maps might be good for the reader and then gradually the effectiveness of access maps might become important.
- (3) For the future study, dynamic (or interactive) generation of access maps that are specified by keywords the user has given are desired.

9 References

- Brusilovsky, P. and Rizzo, R. (2002): Map-based horizontal navigation in educational hypertext. *Proc. Hypertext 2002*, College Park, Maryland, USA, ACM Press.
- Chen, C. (1999): *Information visualization and virtual environments*. London, Springer.
- Crampex, M. and Ranwex, S.: Ontology-supported and ontology-driven conceptual navigation on the World Wide Web. *Proc. Hypertext 2000*, San Antonio, TX, ACM Press, 191-199.
- Eades, P. (1984): A heuristics for graph drawing. *Congressus Numerantium* **42**: 146-160.
- Even, S. (1979): *Graph Algorithms*. Rockville, Computer Science Press.
- Ikeda, K. (2004): Analyses of keyword spaces in a digital book and its navigation based on access maps. Master thesis. Japan Advanced Institute of Science and Technology, Ishikawa.
- Misue, K. and Watanabe, I. (1999): Visualization of keyword association for text mining. *IPSJ SIGNotes of Fundamental Infology* **99**(57): 65-72.
- JAIST: Welcome to the School of Knowledge Science. <http://www.jaist.ac.jp/ks/index-e.html>. Accessed 24 Sept 2005.
- Rizzo, R., Allegra, M., and Fulantelli, G.: Hypertext-like structures through a SOM network. *Proc. Hypertext 1999*, Darmstadt, Germany, ACM Press.
- Rizzo, R., Fulantelli, G. and Allegra, M.: Browsing a document collection as as hypertext. *Proc. WebNet 2000*, San Antonio, TX, AACE, 454-458.
- Sugiyama, K. (2002): *Graph drawing and applications for software and knowledge engineers*. Singapore, World Scientific.
- Sugiyama, K., Nagata, M. and Shimojima, A. (eds) (2002): *Knowledge Science*. Tokyo, Kinokuniya Publishing. (in Japanese)
- Sugiyama, K., Nagata, M. and Shimojima, A. (eds): Digital version of 'Knowledge Science'. <http://www.kousakusha.com/ks/index.html>. Accessed 26 Sept 2005.
- Sugiyama, K., Nagata, M. and Shimojima, A. (eds) (2005): *Knowledge Science*, Seoul, BADA Publishing. (in Korean)
- Tokunaga, T. (1999): *Information retrieval and language processing*. Tokyo, Tokyo University Press. (in Japanese)
- Watanabe, I. (2005): Visual text mining technology for patent mining. *FIJITSU* **56**(4): 371-377.
- Watanabe, I. and Misue, K. (1999): Text mining based on keyword association. *IPSJ SIGNotes of Fundamental Infology* **99**(57): 57-64.
- WBT Systems: E-learning from WBT Systems, <http://www.wbtsystems.com/>. Accessed 23 Nov. 2005.
- WebCT: Learning without limits, <http://www.webct.com/>. Accessed 23 Nov. 2005.

Knowledge Visualization in Hepatitis Study

DucDung Nguyen, TuBao Ho, and Saori Kawasaki

School of Knowledge Science
Japan Advanced Institute of Science and Technology
1-1 Asahidai, Nomi city, Ishikawa 923-1292, Japan
{ducdung, bao}@jaist.ac.jp

Abstract

Evidence-based medicine (EBM) is a shift in which medicine from being based on individual experience of doctors to evidence with clear background. In this shift, data mining can play a significant role as its ability of uncovering medical evidence from large volumes of medical data. Recognizing the crucial role of visualization in discovering such evidences, this work presents some developed tools integrated in our data mining system D2MS for appropriately visualizing knowledge, and their usage in hepatitis study. We emphasize on our two rule visualizers, one for individual rule and the other for rule in its relations with the others.

Keywords: Data and knowledge visualization, hepatitis study, model selection.

1 Introduction

Knowledge discovery in databases (KDD) or *data mining* in short—the rapidly growing interdisciplinary field of computing that evolves from its roots in database management, statistics, machine learning, and others—aims at finding useful knowledge (patterns/models) from large databases. Each of five steps in the KDD process (i.e., (1) understanding the application domain, (2) data preprocessing, (3) data mining, (4) evaluation of discovered knowledge, and (5) applying discovered knowledge) requires interaction and many decisions of the user. In other words, the KDD process can be alternatively viewed as a process of model selection, i.e., that of choosing by the user the most interesting discovered patterns/models or choosing algorithms and their settings to obtain interesting patterns/models in an application. In doing such a task, visualization has an indispensable role because it helps to understand complicated patterns/models in addition to using performance metrics [1], [5].

Medicine is a traditional application domain of artificial intelligence. In the new medicine trend of shifting from being based on individual experience of doctors to being based on evidence with clear background (evidence-based medicine, or EBM), KDD has been much expected to contribute to EBM [4]. However, so far KDD is just at a doorstep for medical applications and still too difficult to

use by physicians in their data analysis [3]. Thus, there is a practical need of developing user-friendly KDD tools to support analyzing medical data. In the last four years we have been involved in the projects on mining hepatitis data [9] for which we have developed some new visualization tools, as well as used them in supporting the discovery process for medical knowledge [8].

The role of visualization in medical data analysis has been addressed in several works [4], however, only a few techniques were particularly developed for medical applications. The purpose of this paper is to introduce some of these visualization tools and demonstrate how they can support medical data mining. These tools include visualizers for multidimensional databases, discovered rules, hierarchical structures as well a synergistic visualization of data and knowledge. These tools are integrated in our knowledge discovery system D2MS (Data Mining with Model Selection) [7], [8].

2 The Visual Data Mining System D2MS

D2MS (Data Mining with Model Selection) is the visual data mining system that we have developed [8] in the framework of our active mining project. Figure 1 shows a conceptual architecture of D2MS where Data Mining component currently includes a decision tree learning (CABRO [12]), and a rule learning (LUPC [7]) subsystems.

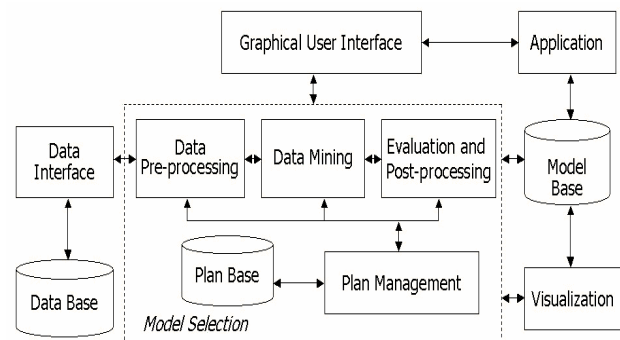


Figure 1: Conceptual architecture of the system D2MS

2.1 D2MS: User-centered system

The *interestingness* of discovered patterns/models is commonly characterized by several criteria: *evidence* indicates the significance of a finding measured by a statistical criterion; *redundancy* amounts to the similarity of a finding with respect to other findings and measures to what degree a finding follows from another one; *usefulness* relates a finding to the goal of the users; *novelty* in-

cludes the deviation from prior knowledge of the user or system; *simplicity* refers to the syntactical complexity of the presentation of a finding, and *generality* is determined by the fraction of the population a finding refers to. The interestingness can be seen as a function of the above criteria, and strongly depends on the user as well his/her domain knowledge.

The key idea of our solution to model selection in D2MS is to support an effective participation of the user in this process. Concretely, D2MS first supports the user in doing trials on combinations of algorithms and their parameter settings in order to produce competing models, and then it supports the user in evaluating them quantitatively and qualitatively by providing both performance metrics values as well as visualization of these models (Figure 2).

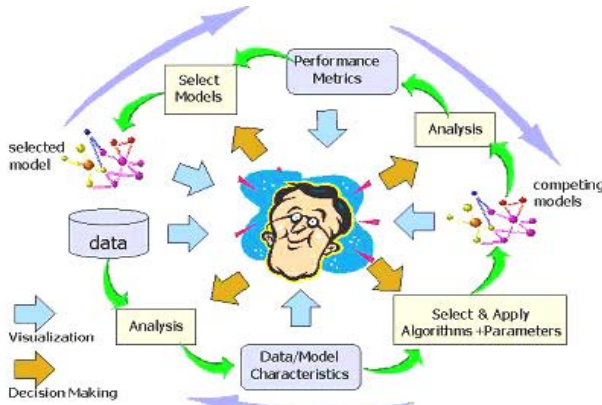


Figure 2: The idea of user-centered system in D2MS and the visualization support

We have chosen the parallel coordinates technique for visualizing 2D tabular datasets defined by n rows and p columns. D2MS improves parallel coordinates in several ways to adapt.

2.2 Knowledge Visualization

2.3.2 Rule visualizer-1

This visualizer aims to view individually a rule. A rule is a pattern related to several attribute-values and a subset of instances. The importance in visualizing a rule is how this local structure is viewed in its relation to the whole dataset, and how the view supports the user's evaluation on the rule interestingness. D2MS's rule visualizer allows the user to visualize rules in the form *antecedent* \rightarrow *consequent* where *antecedent* is a conjunction of attribute-value pairs, *consequent* is a conjunction of attribute-value pairs in case of association rules, and is a value of the class attribute in case of prediction rules. A rule is simply displayed by a subset of parallel coordinates included in *antecedent* and *consequent*. The D2MS's rule visualizer has the following functions:

Each rule is displayed by a polyline that goes through the axes containing attribute-values occurred on the antecedent part of the rule leading to the consequent part of the rule that are displayed with different colour. In the case of prediction rules, the ratio associated with each class in the class attribute corresponds to the number of instances of

the class covered by the rule over the total number of instances in the class, giving a view on the rule quality.

2.3.3. Rule visualizer-2

This aims to view a rule in its relations with others. We developed a graph-based rule visualization technique to support user in finding out interesting patterns.

There have been numerous rule visualization techniques using two dimension matrices (), grid view (), tree view (), and interactive mosaic plots. The common point of the technique is that they stress on visualizing individual rule, or the relation between the left-hand side (antecedent) and the right-hand side (consequent) of a rule. It is still very difficult to user to see which rule is potential new.

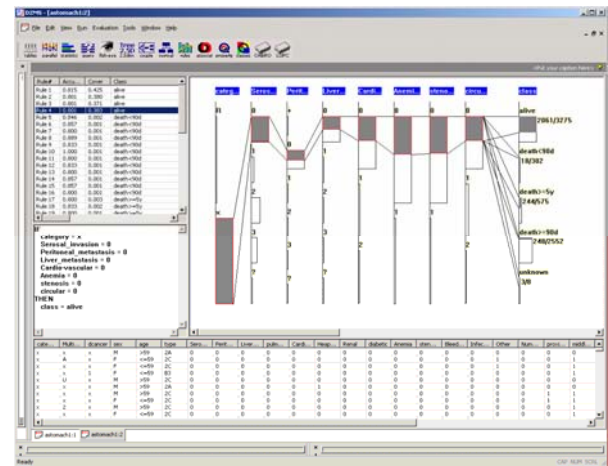


Figure 3: View an individual rule in D2MS: top-left window shows the list of discovered rules, the middle-left and the top-right windows show a rule under inspection, and bottom window displays the instances covered by that rule.

Our approach firstly starts from the question that which could be a good indication of a new rule. According to [13], a rule $A \rightarrow B$ is potentially new with respect to a given rule $X \rightarrow Y$ if

- $B \text{ AND } Y \models \text{FALSE}$. (B and Y logically contradict each other)
- $A \text{ AND } X$ holds on a statistically large subset of tuples in dataset D .
- The rule $A \text{ AND } X \rightarrow B$ holds (so the rule $A \text{ AND } X \rightarrow Y$ holds)

We also view that a rule $A \rightarrow B$ is an unexpected conclusion rule if A and X are similar but B and Y are very different. It is clear that visualizing individual rule could not help to reveal the about relations, but several rules must be viewed together. Our method is to construct a graph based on all rules discovered by data mining algorithms, then visualize the graph and focus on nodes that reflect above relation between precedent and consequence parts of a rule.

In our implementation, each condition (or an attribute-value pair) is represented as a node, and there will be an edge between a conclusion attribute-value pair with any attribute-value pair in the condition part. When focusing on one node, a conclusion or a condition, the magnetic-spring algorithm is applied to visualize a local part of

the whole graph. Because we concern a rule and its related rules and conditions, the local part of the graph includes all nodes with the distance less than or equal 2 from focusing node. Figure 4 shows an example of viewing a rule.

2.3.4 Viewing rules and data

The subset of instances covered by a rule is visualized together with the rule by parallel coordinates or by summaries on parallel coordinates. From this subset of instances, the user can see the set of rules each of them cover some of these instances, or the user can smoothly change the values of an attribute in the rule to see other related possible rules. These possible operations facilitate the user in evaluating the quality of this rule: a rule is good if instances covered by it are not recognized by other rules, and vice-versa (see later in 3.3). The rules for a class can be displayed together, and instances of the class as well of other classes covered by these rule are displayed.

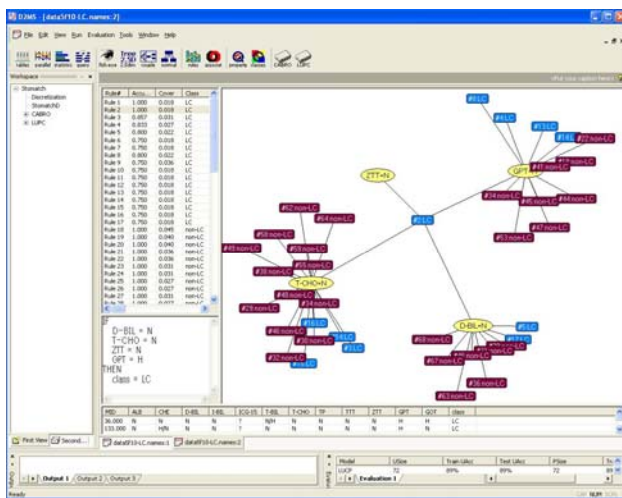


Figure 4: A rule viewed in its relations with others.

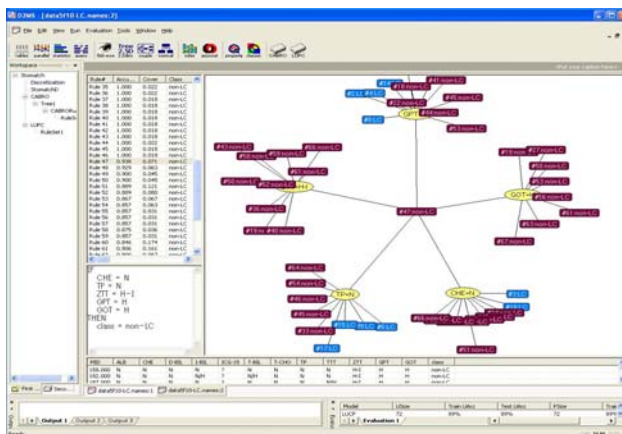


Figure 5: Visualization of rule No.2 for class LC

3 Visualization in mining hepatitis data

3.1 The data mining projects in hepatitis study

The hepatitis database is composed of 6 relational tables of time-series data on 983 laboratory examinations of 771 patients of hepatitis B and C. The data are broadly split into two categories. The first includes administrative in-

formation such as patient's information (age and date of birth), pathological classification of the disease, date of biopsy, result of biopsy, and duration of interferon therapy. The second includes temporal records of blood examination and urinalysis that can be further split into two sub-categories, in-hospital and out-hospital examination data. In-hospital examination data contain the results of 230 examinations that were performed using the hospital's equipment. Out-hospital examination data contain the results of 753 examinations, including comments of staffs, performed using special equipment on the other facilities. Consequently, the temporal data contain the results of 983 types of examinations. The database is given recently to challenge the data mining research community. Among six problems posed by physicians, we focus on the following problems:

- P1. Discover the differences in temporal patterns between hepatitis B and C.
- P2. Evaluate whether laboratory examinations can be used to predict the stage of liver fibrosis. A variant of this problem is prediction of LC (liver cirrhosis, related fibrosis stages F3 and F4) and non-LC (related to fibrosis stages F0, F1, and F2).
- P3. Evaluate whether the interferon therapy is effective or not.

Our mining methods of temporal abstractions to this database were described in detail in [9].

3.2 Visualization support in finding interesting rules in hepatitis study

We illustrate the use of rule visualization in support of interpreting and understanding discovered rules in hepatitis study, in particular the LC vs. non-LC problem.

Figure 6 shows a typical screenshot of rule visualizer2 in investigating this problem. A total of 22 rules were found for LC and 59 rules for non-LC with program LUPC (75% minimum confidence and 4 as minimum support count). Assume that we want to assess its interestingness, says, rule #47 of non-LC whose precedent is the conjunction of "CHE = N and TP = N and ZTT = H-I, GPT = H and GOT = H" as shown in Figure 6.

Rules of each class are displayed with a color, says, red for LC class and dark green for non-LC class, while attribute-value pairs are displayed with yellow color. The rule #47 non-LC is shown at the center in connecting with its four attribute-value pairs in the precedent part (Figure 6). Each attribute-value pair then linked to all rules where it appears in the precedent parts. We can easily observe that conditions "GPT=H", "TP=N", and "CHE=N" occurred in both LC and non-LC classes while the other two are only in non-LC. By a double click on a attribute-value node in Figure 10, says, "GPT=H" we can see in another screen (Figure 11) the links from this node to all the rules that contains "GPT=H" in the precedent part. In fact, we can switch between these two modes in rule visualizer-2 to observe each rule or attribute-value pair and its related conditions or rules, i.e., its neighborhood information. Many views of rule interestingness can be assessed with the visualizers. For example, conditions (a), (b) and (c) of

view in [13] addressed in 2.3.3 can be assessed by rule visualizer-1 and rule visualizer-2. In fact, condition (a) can be applied to any two rules each describes one of two classes LC or non-LC; condition (b) can be accessed by the checking the sets of instances covered by the two rules shown in the window under the graph in Figure 10 (or by clicking the mode that shows the interaction of “X AND A”); (c) can be easily checked by using rule visualizer-1 for the rule “ $A \rightarrow B$ ” as seen in Figure 6, then adding conditions X to A to check “X AND $A \rightarrow B$ ”.

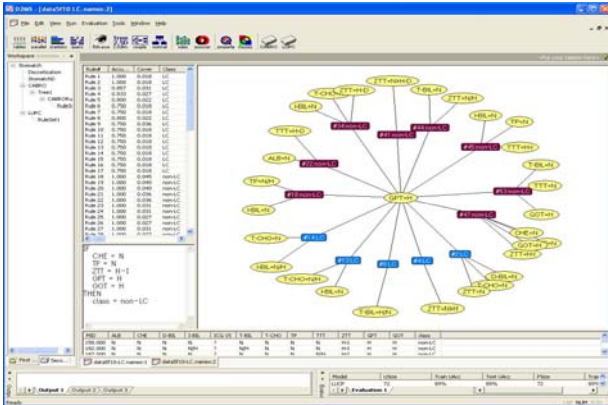


Figure 6: Rules that contain “GPT=H” in the precedent part, obtained by a double click on node “GPT=H” in Figure 10.

Various rules found by D2MS with its visualization tools from stomach cancer data and hepatitis data have been encouragingly evaluated by physicians and they really interested in the tools [7], [8], [9]. A number of rules found recently were judged to be potentially new and useful, for example:

R#10: NonLC: “GPT in very high state with peaks” AFTER “TTT in high state with peaks” AND “GOT in very high state with peaks” ENDS “GPT in very high with peaks” AND “GOT in very high state with peaks” AFTER “TTT in high state with peaks” (support count = 10, conf. = .80).

R#8: LC: “GPT in very high state with peaks” AFTER “TTT in very high state with peaks” AND “GPT in very high state with peaks” BEFORE “TTT in high state with peaks” AND “GOT in very high state with peaks” AFTER “TTT in high state with peaks”, (support count = 8, conf. = .80)

R#42: NonLC: “IDH changed from normal to low state” BEFORE “TTT in normal state with peaks” AND “GOT in very high state with peaks” BEFORE “ZTT in high state with peaks” AND “GOT in very high state with peaks” AFTER “GPT in very high state with peaks” (support count = 6, conf. = .86).

4. Conclusion

We have shortly presented the visual data mining system D2MS. We emphasized the central role of the user's participation in the knowledge discovery process and have developed data and knowledge visualizers in D2MS, in

particular rule visualizer-1 (for individual rules) and rule visualizer-2 (for rules and their neighbourhood information) to support such participation. The visualizers of D2MS have been used in mining stomach cancer and hepatitis databases and shown their usefulness.

References

1. Ankerst, M., Grinstein, G., Keim, D. (2002): Visual Data Mining: Background, Techniques, and Drug Discovery Applications, *Tutorial Notes of ACM SIGKDD '02*, 277-245.
2. Card, S. K., Mackinlay, J. D., Shneiderman, B. (1999): *Readings in Information Visualization*, Morgan Kaufmann.
3. Chittaro, L. (2001): Information Visualization and Its Application to Medicine, *Artificial Intelligence in Medicine* **22**, 81-88.
4. Cios, K.J. (2001): *Medical data mining and knowledge discover* (Ed.) Physica-Verlag.
5. Fayyad, U.M., Grinstein. G.G., and Wierse, A. (2002): *Information Visualization in Data Mining and Knowledge Discovery*, Morgan Kaufmann.
6. Han, J. and Cercone, N. (2000): RuleViz: A Model for Visualizing Knowledge Discovery Process, *Sixth Inter. Conf. on Knowledge Discovery and Data Mining ACM SIGKDD '00*, 244-253.
7. Ho, T.B., Nguyen, T.D., Nguyen, D.D., Kawasaki, S. (2001): Visualization Support for User-Centered Model Selection in Knowledge Discovery and Data Mining, *International Journal of Artificial Intelligence Tools*, World Scientific, **10**(4), 691-713.
8. Ho, T.B., Nguyen, T.D., Nguyen, D.D. (2002): Visualization Support for a User-Centered KDD Process, *ACM International Conference on Knowledge Discovery and Data Mining ACM SIGKDD '02*, 519-524.
9. Ho, T.B., Nguyen, T.D., Kawasaki, S., Le, S.Q., Nguyen, D.D., Yokoi, H., Takabayashi, K. (2003): Mining Hepatitis Data with Temporal Abstraction, *ACM International Conference on Knowledge Discovery and Data Mining ACM SIGKDD '03*, 24-27.
10. Kumar, H. P., Plaisant, C., Shneiderman, B. (1997): Browsing Hierarchical Data with Multi-Level Dynamic Queries and Pruning", *Inter. Journal of Human-Computer Studies*, **46**(1), 103-124.
11. Lee, H.Y., Ong, H.L., Quek, L.H. (1995): Exploiting Visualization in Knowledge Discovery, *ACM International Conference on Knowledge Discovery and Data Mining ACM SIGKDD '95*, 198-203.
12. Nguyen, T.D. and Ho, T.B. (1999): An Interactive Graphic System for Decision Tree Induction, *Journal of Japanese Society for Artificial Intelligence*, **14**(1), 131-138.
13. Padmanabhan, B. and Tuzhilin, A. (1999): Unexpectedness as a Measure of Interestingness in Knowledge Discovery, *Decision Support Systems*, **27**(3), 303-318.

Integrated Visualization for Geometry PIG Data

**Bok Dong Kim¹, Sang Ok Koo¹, Hyok Don Kwon¹, Seong Dae Jung¹,
Soon Ki Jung¹, Minhoo Lee², YongWoo Rho³ and SungJa Koo³**

¹Department of Computer Engineering

²School of Electrical Engineering and Computer Science

Kyungpook National University, 702-701 Daegu, South Korea

¹{comthief, sokoo, hdkwon, tinywolf}@vr.knu.ac.kr, {¹skjung, ²mholee}@knu.ac.kr

³R&D Division, Korea Gas Corporation, 426-790 Ansan, South Korea

{ywrho, sjkoo}@kogas.re.kr

Abstract

The geometry PIG is a system that can inspect the inside of a pipeline. The amount of pigging data is usually considerable because the system records multi-channel sensor values for a long distance inspection. Therefore, the efficient visualization of geometry PIG data is essential for the operator to analyze the amount of sensor data and to assess the pipeline features and conditions within a few hours.

In this paper, we develop a geometry PIG software, which is a powerful and highly user friendly diagnostic tool. It allows the user to visualize the sensor data with various views such as fisheye, histogram and wave views to distinguish particular type of features. Furthermore, an integrated GUI with synchronized multiple views is very effective managing and navigating the multi-channel data. As a result, our system contributes to the high precision of the analysis results as well as in reducing the time to take to analyse the geometry PIG data.

Keywords: geometry PIG, multi-channel data, large data visualization, features visualization, navigation.

1 Introduction

The geometry PIG (Pipeline Inspection Gauge) is designed to provide information such as dents, ovalities, bend radius and angle by making measurement of the inside surface of pipeline (Kim et al. 2003). It is widely used during the commissioning process of new systems and during the regular maintenance of operating pipelines. The geometry PIG must be equipped with a calliper system, odometers, IMU (Inertial Measurement Unit), tracking transmitter and large size of data acquisition and storage system. The geometry PIG rapidly travels through a natural gas pipeline which can be composed of many valves, welds, bends and tees. In passing through the pipeline, the geometry PIG records displacements of the finger, which consists of bit array. The data that is acquired by pigging is stored in a database after pre-processing such as low-pass filter, binary interpolation and so on.

In order to obtain accurate geometric information of a pipeline, it is necessary for the operator to closely examine the multi-channel sensor data for the entire inspection region. Although the operator has professional knowledge about signal patterns, it is difficult and very time consuming to find abnormal signal regions in a vast amount of data and to assess pipeline features. Therefore, the efficient visualization of finger data is an important issue for the operator when reporting on pipeline condition within few hours.

In this paper, we present geometry PIG analysis software, which offers multiple views for visualizing finger data. Finger data is composed of a significant amount of data pertaining to multi-channel sensor values. In the next section, we introduce some visualization techniques and previous research regarding multi-dimensional and large data visualization within various domains.

2 Related Works

Visualization techniques are needed to visualize the many data types. There are several data types of the visualizations. Those types are one, two, three, multi-dimensional, network, tree and so on (Shneiderman 1996). The One-dimensional data type, which is a linear data type, includes textual documents, program sources and temporal data. The two-dimensional data type, which is planar or map data, is used for geographical maps, floor plans and newspaper layouts. The three-dimensional data type, such as the human body, and the multi-dimensional data type are used in many fields (Daassi et al. 2004, Novotny 2003). Each data type is combined with appropriate visualization techniques in order to analyse and extract information. For example, one-dimensional data type such as temporal data has been used in ThemeRiver (Havre et al. 2002).

Our data type is multi-channel one-dimensional data because each signal is closely related to other signals in the geometry PIG. We have to visualize the drift of data stream and make the operator distinguish between feature signals and normal signals. Unfortunately, the operator can not analyse accurately using only one type of the visualization since variations of finger data are sometimes very ambiguous to estimate. For this reason, we developed a visualization system which offers multiple views using various visualization techniques.

This paper is organized as follows: In Section 3, we show the geometry PIG system and its dataset. In Sections 4, 5

and 6 describe the visualization methods for each view and the integrated GUI. In Section 4, we present traditional visualization techniques for geometry pig data. In Section 5, we suggest three views to distinguish a particular feature. In Section 6, we show the integrated GUI with synchronized multiple views. Finally, we draw some conclusions from this work.

3 System Overview

We have to understand the characteristics of the geometry PIG data in order to apply them to visualization techniques. In this section, we present the structure of a hardware system and their components of the geometry PIG. After describing the data acquisition principles, we consider the type and size of the datasets that are manipulated for visualization. Finally, we introduce the architecture of an analysis system. The analysis system has a database that can manage the geometry PIG data and visualization modules. The geometry PIG, which has a flash hard disk, travels and captures an amount of data every second. The geometry PIG, which was developed for this study, is composed of driving cups made of urethane, calipers, odometers, IMU, a tracking transmitter and a storage system, so it can acquire an amount of data, as shown in Fig. 1(a). The geometry PIG passes through the pipelines. Fig. 1(b) shows that the geometry PIG meets a dent feature. Then, we can see that the finger has moved at the bottom of the pipe.

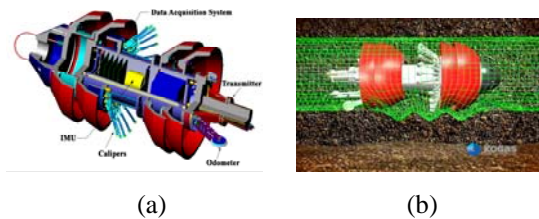


Figure 1. Structure of Geometry PIG. The geometry PIG has 24-fingers : (a) Structure of geometry PIG, (b) Geometry PIG travels through the pipelines.

There are several types of sensor in a geometry PIG: IMU (gyroscope, acceleration), fingers, pressure, temperature and odometer. As you can see in Table 1, each sensor has different sample rates and the rates are so fast. The software is composed of the visualization module, feature editing interface and data access module. Both of two former modules run based on data access module. The data access module offers API that supports to access easily and fast the database.

| Sensor Type | Quantity of Sample Data | Sample Rates |
|-------------|-------------------------|--------------|
| Finger | 48 byte | 400-800 Hz |
| Odometer | 6 byte | 100 Hz |
| IMU | 12 byte | 100 - 400 Hz |
| Pressure | 4 byte | 1 - 10 Hz |
| Temperature | 2 byte | 1 - 10 Hz |

Table 1. Quantity of data per second and sample rates.

4 Visualization of Geometry PIG Data

In our analysis software, several types of visualization techniques are implemented. First, four types of visual strategies are well-known, general purposed visualization techniques: line graph, pseudo color image, three-dimensional surface and polar view (Saraiya et al. 2004, Shimabukuro 2004, Novotny 2004). In this section, we describe how to use those traditional visualization techniques and show its results.

4.1 Visualization Types of Signal Data

Finger signal data are used to analyze information such as bend, dent or weld features. In our system, 24 finger data is used for visualization.

4.1.1 Line Graph Visualization

Line graph visualization displays a two dimensional graph; horizontal axis is used for the distance, and vertical axis is used for the finger value. Fig. 2(a) shows the line graph visualization. Every finger takes a vertical axis and overlaps each other. Therefore, in order to identify fingers, they have different colors. Moreover, the color of a finger or the thickness of a line can be changed by the user, and the range of the vertical axis and horizontal axis can be set arbitrarily. Users can read these attributes on the view. Line graph visualization is mostly used to detect certain features. The operator can quickly identify different values from the drift of the data and their numbers because all of the fingers have the same vertical axis and they have a variety of colors each other.

4.1.2 Pseudo Color Image Visualization

The visualization technique using a pseudo color image which takes the horizontal axis as the distance, divides the vertical axis to the number of fingers. This technique maps the finger values into the min/max values that are set by the user as the rectangular shape and present as an image. Brighter colors mean that the finger value is larger. Darker colors mean lower values. The position of the finger located on the vertical axis takes rolling option and user can set the position of the finger as the user defined the key and the mouse. Since the number of fingers is 24, aliasing occurs between each finger. Therefore, in order to make softer color, we applied the interpolation between the fingers. Fig. 2(b) presents some bend features using this visualization technique.

4.1.3 3D Surface Visualization

Three-dimensional visualization supports translation, rotation and zooming. Plus, the information is represented for a user to recognize the current status on the screen. We also use this skill. Fig. 2(c) represents some features of the finger data. In 3D surface visualization, the horizontal axis is the distance and vertical axis is divided into the number of fingers with pseudo color image visualization. In order to make 3D surface, we apply the z-axis whose magnitude equals finger value. In addition, the user defined texture is used. This visualization is as good as the pseudo color

image visualization, but it is difficult to edit features and to identify values which are overlapped data.

4.1.4 Polar View Visualization

In previous research, circle-type visualization such as star representation, which is used to describe monthly data, was applied (Daassi et al. 2004). We use similar visualization to represent PIG data called polar view visualization. The idea of our visualization technique is to show a section of a pipeline. The finger values are mapped to a circle, whereby the user is able to know directly the values. This visualization, however, has a weakness in that it represents data against another technique because it is difficult for the user to understand the drift of overall data; we can not see many data in the screen using this technique. As shown Fig. 2(d), the thick (blue) line is the values of now distance that is pointed in another view because the polar view is synchronized the mouse point from another view. The finger values are set from the center of the circle. It is also available to estimate leaning toward right-bottom.

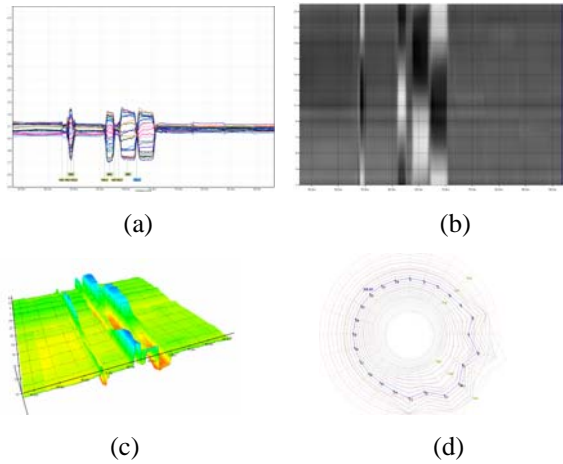


Figure 2. Visualization types of signal data : (a) Line graph visualization, (b) Pseudo color image visualization, (c) Three-dimensional surface visualization, (d) Polar view visualization.

4.2 Efficient Visualization for Feature Signal

The operator examines the signal to input the feature. In particular, weld features are very frequently occurring features but the operator must input all of them in order to compute the bend radius with accuracy. In this section, we propose three additional visualization techniques in order to help the operator better navigate and to distinguish features. The fisheye view using cylindrical mapping that transforms data intervals to rearrange interval from regular intervals. The second visualization method offers a histogram view and the last method promotes wave view visualization.

4.2.1 Fisheye View Visualization

There are problems in the visualization of large sets of data in many interactive applications due to the considerable screen space (Gutwin and Fedak 2004). In order to solve the problem, fisheye view strategies are proposed. In our case, the idea of the proposed fisheye view zooms on the center. To distinguish the relation between previous and

next features, most operators tend to locate feature data in the center of a screen.

In previous methods, the values of the finger data are drawn at regular intervals in the horizontal axis. In this section, however, in order to make fisheye view, the regular intervals are substituted with the rearranged intervals. In Fig. 3(a), dots of left figure indicate the regular intervals of the horizontal axis. The right figure shows the rearranged intervals. In order to change the intervals, we used this process. First, we assume that the circle is a unit circle. Then, the variable r means the radius of the circle. New intervals are acquired by projecting to the horizontal axis after a semicircle divides the regular intervals. Below Eq. (1) and (2), variable k is the number of the data, and N is the number of fingers. Our purpose is to obtain x_k which is the location on the screen. Locations are acquired by subtracting the projected positions from the radius of the circle in the Eq. (1).

$$x_k = r - r \cos \theta, \quad (1)$$

$$\theta = \frac{k}{N} \times 180. \quad (2)$$

It is useful for the operator because the fisheye view does not need while the user is analysing the feature. Fig. 3(b) shows applied fisheye view that rearranges location of the horizontal axis. In the example, the user is able to experience wheeling cylinder in view.

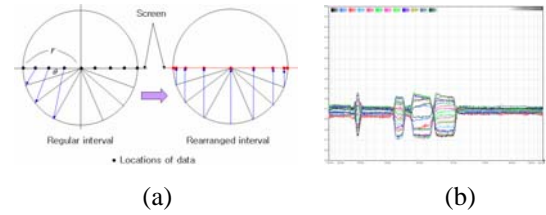


Figure 3. Fisheye View : (a) Regular intervals and new intervals. (b) Visualization using line graph applied fisheye view.

4.2.2 Histogram view

The finger values have the characteristic that sharply changes when the geometry PIG passes through some features. Even though visualization that uses line graph utilizes several colors to distinguish each finger, the user can not recognize some features when the values are same or similar. Therefore, we proposed another view to assist the operator.

There are two types of views. One is the histogram view and the other is wave view. In Eq. (3), variable k means the number of finger data, N is the number of fingers and i is a finger number. In Eq. (3), we can find differences that consist of the absolute value in the location of k th. Then, By accumulating the differences, the histogram view is obtained. The histogram view is shown in Fig. 4.

$$D_k = \sum_{i=0}^N \sqrt{(f_{(k+1,i)} - f_{(k,i)})^2}. \quad (3)$$

In the histogram view, since the visualized data is the accumulated values, we can see clearly the tendency of the values of the fingers. In particular, the peaks of the

histogram indicate accurate feature locations. Therefore, the histogram view is very helpful in determining an accurate location. In our experiment, the histogram view works well in detecting weld features.

4.2.3 Wave view

The wave view solves the problem regarding overlapping finger values in a visualization technique, by using line graph. Also, the wave view helps the operator to identify average of all fingers. The average is important in detecting the thickness of a pipeline. Fig. 4 shows the wave view. In order to visualize the wave view, above all, the average value which acquired from Eq. (4), is drawn by one line. Next, each deviation from the average accumulates in the upper direction until k equals $N/2$. The remaining deviation, when k starts from $N/2$ to N , accumulates in the lower direction. The deviation can be computed by Eq. (5).

$$\mu_k = \frac{1}{N} \sum_{i=0}^N f_{ki} \quad , \quad (4)$$

$$D_i = \sqrt{(\mu_k - f_{ki})^2} \quad . \quad (5)$$

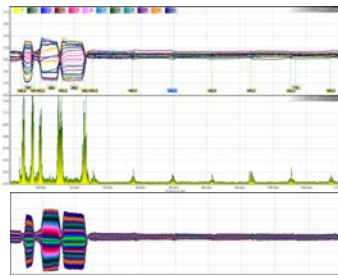


Figure 4. Line graph, Histogram view and Wave view.

5 Integrated Interface

Since the operator has to analyse so a lot of data, efficient interface is essential. Usually, operators want multiple views to analyse the same data space because with only one view, it is difficult to get accurate information of the pipeline and to assess all features. In our system, each view can be viewed independently or all of the views can be shown on the screen simultaneously. Mostly, the analyser detects and records features of a pipeline through the line graph. At the same time, the operator looks over other views to get more precise information. These multiple view interface supports to set the position of every child window in an application. We said that all child windows are synchronized. In moving one child window, the user can identify the same feature through another window at the same time. Fig. 5 shows the user defined setting.

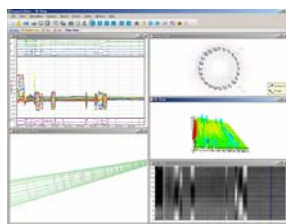


Figure 5. User-defined multiple windows.

6 Conclusion

We developed a geometry PIG software, which is a powerful and highly user friendly diagnostic tool. Our system provides three additional visualization methods such as fisheye view, histogram view and wave. These views help the user to navigate though a huge amount of PIG data and to distinguish particular pipeline features. The fisheye view allows the operator to see more precisely the region in which she or he is interested. The histogram view is very helpful in finding the exact location of the feature. In particular, we can detect weld points very accurately by using the histogram view. The wave view removes the ambiguity in a line graph that whole signal data is blurred due to the interference between signals. Furthermore, in our software, the integrated GUI with synchronized multiple views is very effective in managing and navigating the multi-channel data.

The visualization method and software interface in this paper can be applied to a visualization of data that is similar to the geometry PIG data, i.e. multi-channel large data such as MFL PIG data or various sensor data.

7 References

- Daassi, C., Nigay L. and Fauvet, M.C. (2004): Visualization process of Temporal Data. *Database and Expert Systems Applications, 15th International Conference (DEXA 2004)*, 914-924.
- Gutwin, C. and Fedak, C. (2004): Interacting with Big Interfaces on Small Screens: a Comparison of Fisheye, Zoom, and Panning Techniques. *ACM Proceedings of the 2004 Conference on Graphics Interface*, 145-152.
- Havre, S., Hetzler, E., Whitney, P. and Nowell, L. (2002): ThemeRiver: Visualizing Thematic Changes in Large Document Collection. *IEEE Transaction on Visualization and Computer Graphics*, 8(1): 9-20.
- Kim, D.K., Cho, S.H., Park, S.S., Yoo, H.R., Rho, Y.W. and Kho, Y.T. (2003): Development of the Caliper System for a Geometry PIG Based on Magnetic Field Analysis. *KSME International Journal*, 17(12): 1835-1843.
- Novotny, M. (2004): Visually Effective Information Visualization of Large Data. *Central European Seminar on Computer Graphics (CESCG'04)*.
- Saraiya, P., North, C. and Duca, K. (2004): An Evaluation of Microarray Visualization Tools for Biological Insight. *10th IEEE Symposium on Information Visualization (INFOVIS 2004)*, 1-8.
- Shimabukuro, M.H., Flores, E.F., Oliveira, M.C.F.de and Levkowitz, H. (2004): Coordinated Views to Assist Exploration of Spatio-Temporal Data: A Case Study. *2nd International Conference on Coordinated & Multiple Views in Exploratory Visualization (CMV'04)*, 107-117.
- Shneiderman, B. (1996): The eyes have it: a task by data type taxonomy for information visualizations. *Proceedings of the 1996 IEEE Symposium on Visual Languages*, 336-343.

Increasing the Readability of Graph Drawings with Centrality-Based Scaling

Damian Merrick

Joachim Gudmundsson

School of Information Technologies, University of Sydney 2006, Australia
and National ICT Australia, Australian Technology Park, Eveleigh 1430, Australia.
Emails: dmerrick@it.usyd.edu.au, Joachim.Gudmundsson@nicta.com.au

Abstract

A common problem in visualising some networks is the presence of localised high density areas in an otherwise sparse graph. Applying common graph drawing algorithms on such networks can result in drawings that are not highly readable in the dense areas. Additionally, networks whose layouts are defined geographically often have dense areas that are located within small geographical regions relative to the size of the entire network. In cases where relationships within these dense areas are of interest, it is desirable to be able to distort the graph layout such that the denser areas are enlarged from their original sizes.

In this paper, we propose a technique for enlarging dense areas of a given graph layout, and shrinking sparse areas. This technique is applied to geographical layouts of railway networks and force-directed layouts of non-geographical networks. The results show an increase in readability of dense parts of the networks. In addition, they provide improved starting layouts for schematisation methods which may be used to further increase readability.

Keywords: Graph drawing, centrality, metro maps

1 Introduction

A common problem in visualising some networks is the presence of localised high density areas in an otherwise sparse graph. Consider the image shown in Figure 1. This image shows the geographical layout of the Cityrail railway network in Sydney, Australia. The network consists of a relatively dense area around the inner suburbs of the city, and long paths emanating outward from this central region. Note that the dense central region takes up only a very small portion of the diagram, and as such the details of the network in that area are difficult to see. Also, because of the sparse outer paths in the network, a large percentage of the diagram shows very little or no information.

These properties are common in geographical networks that are based around a city but also cover the outlying regions of that city, as is the case with the Sydney Cityrail network. Similar issues can be found



Figure 1: Geography of the Sydney Cityrail network.

in some cases where graph drawing algorithms have been applied to non-geographical networks. Figure 2 shows one such case, where a graph has been laid out with a spring embedder (Eades 1984). In this drawing, several dense clusters are obvious, with sparse connections between them. However, the intra-cluster edges are not clearly readable, as each of the dense clusters is drawn in only a small area of the diagram.

In both geographical and non-geographical cases, it may be desirable to distort the layout somewhat in order to increase the size of the dense areas with respect to the size of the sparse areas.

One way to do this is to allow the user to interact with the diagram. Zooming, panning and rotating along with tools such as a fish eye lens can be useful in exploring the dense areas of a layout; for a comprehensive review, see the literature (Herman, Merlançon, Marshall 2000). Even when interaction is involved, it is usually desirable to start with the best graph drawing possible. Additionally, there are some places where user interaction is not a possibility, such as the production of drawings and maps in print, or in another non-interactive format.

In this paper, we propose analysing the structure of a network in order to *automatically* determine which parts of the network are the most dense, or

Copyright ©2006, Australian Computer Society, Inc. This paper appeared at Asia-Pacific Symposium on Information Visualization (APVIS 2006), Tokyo, Japan, February 2006. Conferences in Research and Practice in Information Technology, Vol. 60. K. Misue, K. Sugiyama and J. Tanaka, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

National ICT Australia is funded through the Australian Government's Backing Australia's Ability initiative, in part through the Australian Research Council.

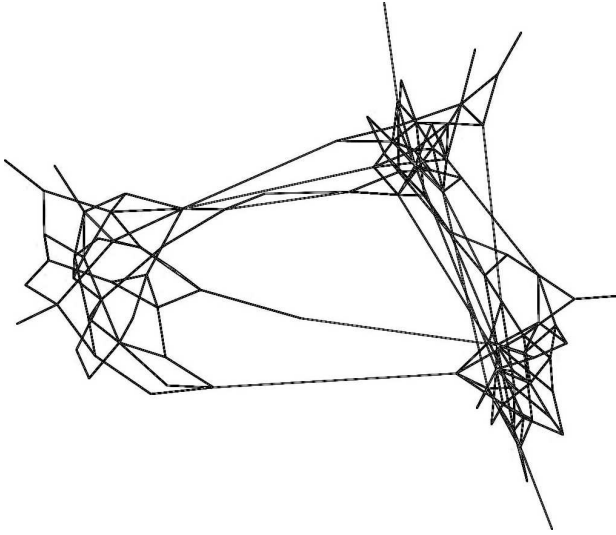


Figure 2: A force-directed graph layout.

in some specified way the most important. To do this, we utilise the concept of node *centrality*, common in the field of network analysis. Once centralities have been calculated, we perform a scaling process to enlarge the parts of the network that are the most important and shrink those that are less important, while still retaining some similarity to the initial layout. Centrality is detailed further in Section 2.2.

The primary motivation for this paper is the automatic visualisation of metro maps. Metro maps have been used to effectively illustrate transportation networks for many decades. The style in which metro maps are drawn is quite distinct, and has been widely accepted as an effective way to communicate information about a metropolitan transportation network.

Most existing metro maps achieve a balance between an easily understandable abstraction of the network and geographical accuracy in portraying the locations of metro stations and train lines. Such maps are generally drawn manually by a graphic designer. Recently, there has been strong interest in developing methods that automatically create metro maps to a similar standard, or at least provide a reasonable starting layout from which a human designer may quickly reach a final, high-quality drawing. However, creating metro map visualisations, particularly automatically, is a challenging problem, as observed by Beck (Garland 1994) and Tufte (1997).

Hong *et al.* (Hong, Merrick, do Nascimento 2005) present some force-directed graph drawing approaches to metro map layout, which are also used to produce metro map layouts of non-geographical networks. Slower but more geographically accurate optimisation-based methods are detailed by Stott and Rodgers (2004), and more recently by Nöllenburg and Wolff (2005).

A problem related to metro map layout is the schematisation of networks. Schematisation usually involves altering the geometry of a network such that each line in the final layout lies in one of a restricted number of orientations. Cabello *et al.* (Cabello, de Berg, van Kreveld 2005) give an $\mathcal{O}(n \log n)$ time algorithm which creates a schematised map of a given network, where intersection points remain in fixed locations and paths between intersection points are drawn using two or three links. Individual paths in a network can be schematised using a restricted orientation path simplification algorithm, such as the one proposed by Neyer (1999). More recent work investigates path simplification in restricted orienta-

tions specifically applied to metro networks (Merrick, Gudmundsson 2005).

Section 2 gives some background on the key concepts used in the rest of this paper. Section 3 details our centrality-based scaling technique. Results of applying the scaling technique on various graphs are presented in Section 4, and Section 5 gives some concluding remarks and directions for future research.

2 Background

As discussed in Section 1, the primary motivation for this paper is metro map visualisation. An additional motivation for centrality-based scaling is to alter graph drawings such that they become more readable. Graph drawing is an extensively studied topic of research, and over the past decades many algorithms for drawing graphs have been developed and improved.

In this paper, we are interested in utilising the output of graph drawing algorithms as input to the centrality-based scaling algorithm of Section 3. A comprehensive review of graph drawing concepts and techniques may be found in the literature (Di Battista, Eades, Tamassia, Tollis 1999).

Some related research has been done in “cluster busting” (Lyons, Meijer, Rappaport 1998), where a point set is non-uniformly scaled in such a way as to spread out clusters in the point set. In this work, the points are scaled with some anchoring to the original layout, but edges in a graph are not considered.

2.1 Time-Distance Mapping

The algorithm presented in Section 3 of this paper utilises the *time-distance mapping* method of Shimizu and Inoue (2003) to perform scaling on a graph layout. Time-distance mapping is a technique for altering the lengths of edges in a geographical railway map, such that the time taken to travel from one station to an adjacent station is represented on the map as the Euclidean distance between the two stations. In addition, they attempt to minimise the change in orientation of each edge from the initial layout to the final layout.

Given a graph $G = (V, E)$ and an initial layout of G , Shimizu and Inoue’s algorithm iteratively computes linear least-squares solutions to the following two functions:

$$\begin{aligned} \min \quad & \sum_{(u,v) \in E} (d'(u,v) \sin \theta_{(u,v)} - (x_v - x_u))^2 \\ \min \quad & \sum_{(u,v) \in E} (d'(u,v) \cos \theta_{(u,v)} - (y_v - y_u))^2, \end{aligned}$$

where $d'(u,v)$ is the desired length of the edge from u to v , $\theta_{(u,v)}$ is the angle from east of the edge from u to v in the original geometry, and x_u, y_u, x_v and y_v are the x - and y -coordinates of the vertices u and v respectively.

At each iteration, the new orientations of edges are checked against those from the last iteration, and the algorithm stops if the total change was smaller than some given threshold.

2.2 Centrality

In our scaling algorithm, we make use of the notion of *centrality* in order to decide on appropriate edge lengths in a layout. Centrality is a concept that is

often used in social network analysis, whereby a combinatorial function is applied to each vertex in a network to quantify that vertex's importance relative to other vertices.

Research on this topic has produced a number of different measures for calculating centrality, each focusing on particular aspects of a network's structure. We utilise three different measures in application to our centrality-based scaling method: *degree centrality*, *betweenness centrality* and the so-called *hubs and authorities* method applied to undirected graphs. Wasserman and Faust (1994) review other centrality measures in detail.

Degree centrality assigns a value to each vertex according to the number of edges to which it is adjacent. This provides a localised measure of density at each vertex. The normalised degree centrality for a vertex v is defined as

$$C_D(v) = \frac{\deg(v)}{|V| - 1},$$

where $\deg(v)$ is the number of edges adjacent to v (the degree of v). Degree centrality for an entire network can be computed in $\mathcal{O}(|V|)$ time.

Betweenness centrality (Freeman 1977, Anthonisse 1971) measures, for each given vertex, the proportion of all paths through the network which pass through that vertex. In social network analysis, betweenness is used to identify vertices that have control of the information flow in the network. That is, a higher betweenness for a particular vertex indicates that the path taken to travel between any two other vertices in the graph is more likely to include the vertex in question. The normalised betweenness centrality for a vertex v is defined as

$$C_B(v) = \frac{1}{(n-1)(n-2)} \sum_{\substack{s,t \in V; \sigma_{s,t} > 0; \\ s \neq v \neq t}} \frac{\sigma_{s,t}(v)}{\sigma_{s,t}},$$

where $\sigma_{s,t}$ is the number of shortest paths from vertex s to vertex t , and $\sigma_{s,t}(v)$ is the number of these paths which pass through vertex v . The algorithm of Brandes (2001) may be used to compute betweenness centralities for an entire network, and requires time $\mathcal{O}(|V||E|)$.

The hubs and authorities method was developed as a way of calculating importance rankings for web pages (Kleinberg 1999). Kleinberg defines *authorities* to be pages that are primary sources of information for a given topic, and *hubs* to be pages such as guides or resource lists for that topic, which refer users to relevant authorities. Authorities generally have the characteristic that although they may have many incoming links, they do not necessarily link to other major authorities, with whom they may be in competition. Conversely, hubs may have many outgoing links, as they refer to a number of relevant authorities on a given topic, but they may not have a large number of incoming links from other pages.

Kleinberg proposes an algorithm for computing both a hub weight and an authority weight for vertices in a directed graph. We may apply the same process to an undirected graph, such as a metro map graph, by replacing each edge with arcs in both directions. In this undirected case, the separate hub and authority weights merge to produce a single importance value for each vertex. The set of importance values C_H for all vertices in the graph may be defined as the solution to the matrix equation

$$A^T A C_H = \lambda C_H,$$

where A is the adjacency matrix of the graph, and λ is the highest real, positive and simple eigenvalue of the matrix A . This calculation is akin to the earlier eigenvector-based centrality measures (Bonacich 1987). The value $C_H(v)$ at a vertex v gives an indication of the density of the graph around v , but in a more global context than degree centrality, as v 's proximity to other vertices of high degree contributes to the value. Kleinberg's algorithm can be used to compute these values for an entire network in time $\mathcal{O}(|V|I)$, where I is the number of iterations taken by the algorithm to converge.

3 Centrality-Based Scaling

In this section, we propose an algorithm to scale the set of vertices of a given graph drawing or geographical network layout non-uniformly, in order to enlarge the most important areas. The problem addressed by the algorithm may be defined as follows.

Input: a graph $G = (V, E)$, a layout L of G and a vertex centrality measure C .

Output: a layout L' of G with important areas enlarged.

We now describe the algorithm.

Centrality-Based Scaling

1. Calculate vertex centrality $C(v)$ for every vertex $v \in V$.
2. Compute desired edge length $d'(u, v)$ for every edge $(u, v) \in E$.
3. Reposition vertices such that the length of each edge $(u, v) \in E$ is close to $d'(u, v)$, and the orientation (u, v) is close to its orientation in L .

Step 1 is performed using the methods described in Section 2.2. Step 2 involves using the calculated centrality values to determine desired edge lengths, and is described in detail in Section 3.1. Finally, the operation of the algorithm in Step 3 is detailed in Section 3.2.

3.1 Mapping Centrality to Edge Length

Once centrality values have been established, we compute desired lengths for every edge in the network. The desired length for an edge between vertices u and v is calculated as

$$d'(u, v) = \alpha \left(1 + \left(\beta \frac{C(u) + C(v)}{2} \right)^\gamma \right),$$

where $C(u)$ and $C(v)$ are the centrality values of vertices u and v respectively, α and β are parameters defining the scale of the output and γ determines the rate at which edge length decreases with respect to centrality.

Section 4 specifies the values of α , β and γ that we used in applying centrality-based scaling to a number of different data sets.

3.2 Scaling Edges

Given a complete set of desired edge lengths for a metro map graph, the task remains to scale the coordinates of the vertices such that the desired edge lengths are obtained. Of course, a trade-off must occur, as we also desire that the orientations of the edges

remain as close as possible to their orientations in the original given geometry.

To perform this task, we adapt the method of Shimizu and Inoue (2003) introduced in Section 2.1. In the original time-distance scaling method, no guarantee of convergence is given, and we found that executing the least-squares minimisation step just once gave satisfactory results. Further iteration tended to worsen the error in edge orientation, while often not reducing the distance error substantially. We therefore decided to use only a single least-squares minimisation step in performing the mapping of centrality values to edge lengths.

After the least-squares squares minimisation step has been completed, the final scaled layout L' is determined by the repositioned coordinates of the vertices.

4 Results

We applied the centrality-based scaling method of Section 3 to the geographical layouts of the Sydney Cityrail and London Underground railway networks, and to force-directed layouts of two non-geographical random graphs. The geographical layout of the Sydney Cityrail and London Underground networks are depicted in Figures 1 and 3 respectively.

The first of the random graphs, henceforth called RG1, is the example from Section 1. Its initial layout can be seen in Figure 2. The initial layout of the second random graph RG2 is shown in Figure 4. RG1 and RG2 were generated using the random clustered graph generator of the GEOMI software package presented by Ahmed *et al.* (2005).

We implemented the centrality-based scaling algorithm as a plug-in to the graph editor jj-Graph (Friedrich 2002), utilising the JUNG library (O'Madadhain, Fisher, Smyth, White, Boey 2005) to calculate node centralities and the JAMA matrix package (Boisvert, Hicklin, Miller, Moler, Pozo, Remington, Webb 2005) to perform the linear least-squares calculations.

Degree, betweenness and hubness centralities were calculated for each example. Desired edge lengths were computed as described in Section 3.1 with $\alpha = 10.0$ and $\beta = 50.0$, which were determined through extensive experimentation to be appropriate values for our particular implementation. For each example under each centrality measure, γ was set to 0.5, 1.0, and 2.0, and a scaled layout generated each time. Thus square root, linear and quadratic scalings for each centrality were obtained.



Figure 3: Geography of the London Underground network.



Figure 4: Initial layout of RG2.

The results of applying centrality-based scaling on the Sydney Cityrail, London Underground, RG1 and RG2 data sets are shown in Figures 5, 6, 7 and 8 respectively.

It can be seen from these results that in general, degree and hubness centralities significantly expand some of the clusters from the original layouts. This effect tends to increase with the power of the scaling, in some cases to an extreme amount, as in the quadratic hubness result seen in Figure 8(i). The degree centrality scalings of the London network do not expand the centre as much as in the other examples, due to the degree distribution being more uniform.

In Sydney Cityrail, RG1 and RG2, scaling by betweenness appears to have the opposite effect to degree and hubness. In the random graph examples, there are relatively few nodes linking two or more clusters, and in the Cityrail example, there are very long paths crossing from one side of the network, through the centre, to the other side. Thus, the nodes in these sparser areas have high betweenness values, and their coincident edges are extended much more than the edges within the dense clusters. The result of this is a layout which emphasises the clustering of the graph by pushing each cluster away from every other cluster, rather than expanding the inside of the clusters. Again, the effect is more pronounced with a higher value of γ .

In the London network, betweenness actually has the effect of expanding the inner city area, as in this graph the central area is more lattice-like than in the other examples. This means that most paths through the network will pass through the nodes in the central area, giving those nodes high betweenness values and thus expanding the area.

It is clear from the results that both the centrality measure and the parameters chosen for use in the scaling process have a significant effect on the output. The most appropriate values may depend heavily on the input graph itself. It is expected that for a given graph, these parameters will be tweaked experimentally by a user to obtain the best result possible, in a similar manner to the common practice of fine-tuning of individual forces for force-directed layout algorithms.

After obtaining the above results, we applied a path simplification technique (Merrick and

Gudmundsson 2005) to schematise one of the scaled results from each of the Sydney Cityrail and London Underground networks. We applied the same schematisation technique to the original, unscaled layouts of these networks, and the results can be seen in Figures 9 and 10. Figure 9(b) used the quadratic hubness scaling of Cityrail as input, and Figure 10(b) used the linear betweenness scaling of London.

From the scaled and schematised results, it is clear that the detail of the central areas of the railway networks is much more visible. The schematisation process that we used creates simplified paths within a fixed distance threshold of the original paths, so the scaled versions of the schematised layouts are less confusing. There is more space between pairs of nearby edges, and hence the schematisation process creates a lower number overlapping edges in simplified paths.

5 Conclusion

We have proposed an algorithm for centrality-based scaling, and illustrated its use on several geographical and non-geographical networks. The effect of three different centrality measures on the scaling process was shown, as well as the variations obtainable through changes in the mapping of centrality to desired edge length. Finally, the algorithm was applied in combination with a schematisation approach to create clearer automatic layouts of metro maps.

Further work may be worthwhile in exploring the use of other centrality measures, and in performing a comprehensive sensitivity analysis of changes in the parameters involved in the scaling process. In addition, only a single method for adjusting the lengths of edges in a graph layout was tried; an investigation into other methods may result in a better minimisation of angle and length error.

Formal evaluations of the algorithms, both numerically and user-based, may be beneficial in providing more concrete conclusions on the benefits and disadvantages of centrality-based scaling. Toward this goal, quantitative aesthetic criteria specific to the problem would be needed. Finally, it is not yet clear whether the process of choosing the parameters α , β and γ could be effectively automated.

6 Acknowledgements

The authors would like to acknowledge the help of Martin Nöllenburg and Alexander Wolff in obtaining the geographical data for the London Underground network.

References

- Ahmed, A., Dwyer, T., Forster, M., Fu, X., Ho, J., Hong, S.-H., Koschützki, D., Murray, C., Nikolov, N. S., Taib, R., Tarassov, A. & Xu, K. (2005), GEOMI: GEOMetry for Maximum Insight, to appear in Proc. of the 13th International Symposium on Graph Drawing.
- Anthonisse, J. M. (2005), The rush in a directed graph, Technical Report BN 9/71, Stichting Mathematisch Centrum, Amsterdam.
- Boisvert, R. F., Hicklin, J., Miller, B., Moler, C., Pozo, R., Remington, K. & Webb, P. (2005), JAMA: Java Matrix Package, Accessed on 7th October 2005 at <http://math.nist.gov/javanumerics/jama/>.
- Bonacich, P. (1987), Power and centrality: a family of measures, *American Journal of Sociology*, 92:1170–1182.
- Brandes, U. (2001), A faster algorithm for betweenness centrality, *Journal of Mathematical Sociology*, 25(2):163–177.
- Cabello, S., de Berg, M. & van Kreveld, M. (2005), Schematization of networks, *Computational Geometry and Applications*, 30:223–238.
- Di Battista, G., Eades, P., Tamassia, R. & Tollis, I. G. (1999), *Graph Drawing: Algorithms for the Visualization of Graphs*, Prentice-Hall.
- Eades, P. (1984), A heuristic for graph drawing, *Congressus Numerantium*, 42:149–160.
- Freeman, L. (1977), A set of measures of centrality based on betweenness, *Sociometry*, 40:35–41.
- Friedrich, C. (2002), Animation in relational information visualisation, PhD Thesis, University of Sydney, Sydney, Australia, 2002.
- Garland, K. (1994), *Mr. Beck's Underground Map*, Capital Transport Publishing, England.
- Herman, I., Melançon, G. & Marshall, M. S. (2000), Graph Visualization and Navigation in Information Visualization: A Survey, *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24–43.
- Hong, S.-H., Merrick, D. & do Nascimento, H. A. D. (2005), The metro map layout problem, In Proc. of the 12th International Symposium on Graph Drawing, p. 482–491.
- Kleinberg, J. M. (1999), Authoritative sources in a hyperlinked environment, *Journal of the ACM*, 46:604–632.
- Lyons, K. A., Meijer, H. & Rappaport, D. (1998), Algorithms for cluster busting in anchored graph drawing, *Journal of Graph Algorithms and Applications*, 2(1):1–24.
- Merrick, D. & Gudmundsson, J. (2005), C-Directed Path Simplification for Metro Map Layout, Manuscript.
- Neyer, G. (1999), Line simplification with restricted orientations, In Proc. of the 6th International Workshop on Algorithms and Data Structures, p. 13–24.
- Nöllenburg, M. & Wolff, A. (2005), A mixed-integer program for drawing high-quality metro maps. To appear in Proc. of the 13th International Symposium on Graph Drawing.
- O'Madadhain, J., Fisher, D., Smyth, P., White, S. & Boey, Y.-B. (2005), Analysis and visualization of network data using JUNG, To appear in *Journal of Statistical Software*. Accessed on 7th October 2005 at <http://jung.sourceforge.net/>.
- Shimizu E. & Inoue, R. (2003), Time-distance mapping: visualization of transportation level of service, In Proc. of Symposium on Environmental Issues Related to Infrastructure Development, p. 221–230.
- Stott J. & Rodgers, P. (2004), Metro map layout using multicriteria optimization, In Proc. of the 8th International Conference on Information Visualization, p. 355–362.
- Tufte, E. R. (1997), *Visual Explanations*, Graphics Press, Cheshire.
- Wasserman, S. & Faust, K. (1994), *Social Network Analysis: Methods and Applications*, Cambridge University Press.

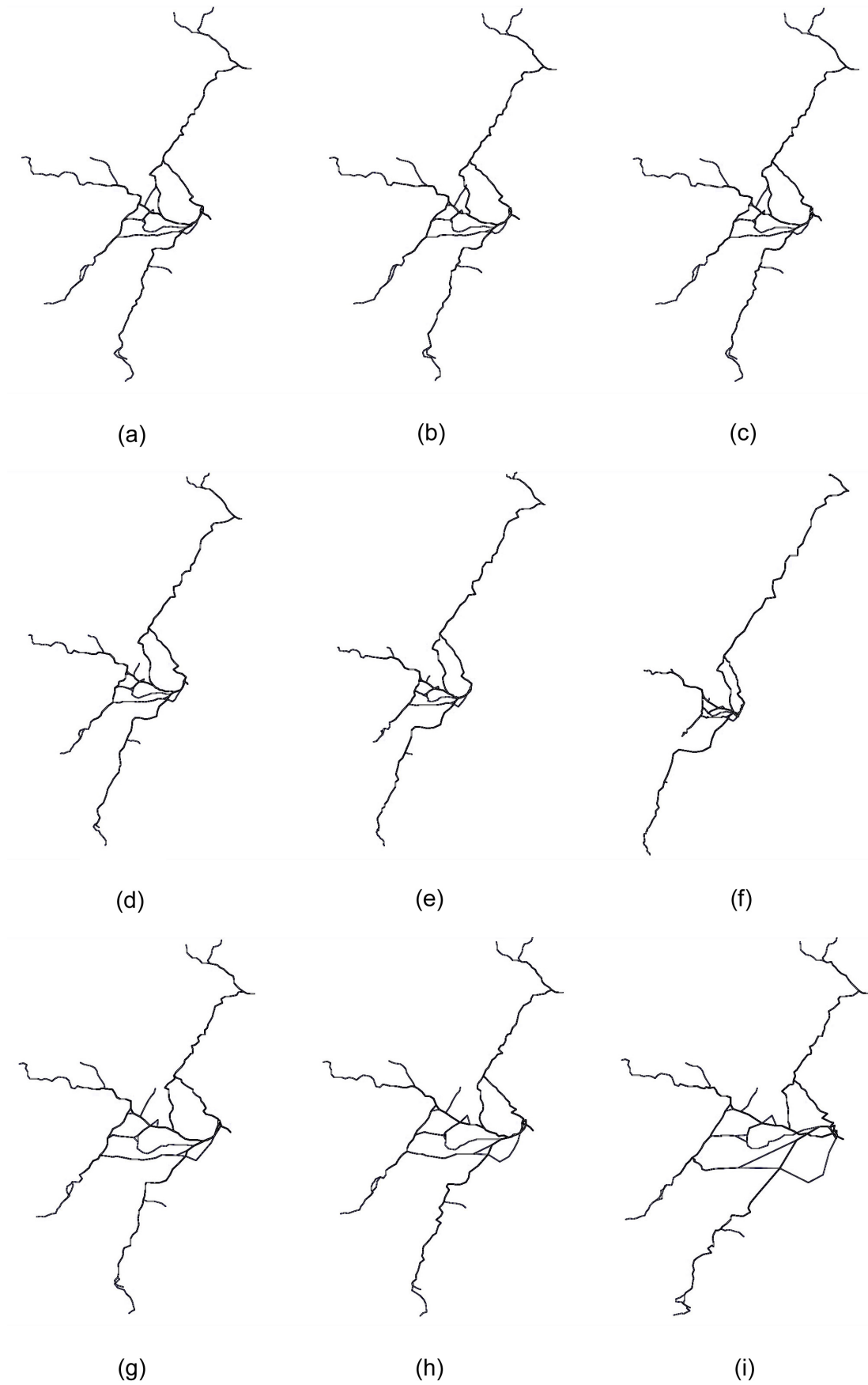


Figure 5: Scaling results for Sydney Cityrail: (a) square root degree, (b) linear degree, (c) quadratic degree, (d) square root betweenness, (e) linear betweenness, (f) quadratic betweenness, (g) square root hubness, (h) linear hubness, (i) quadratic hubness.

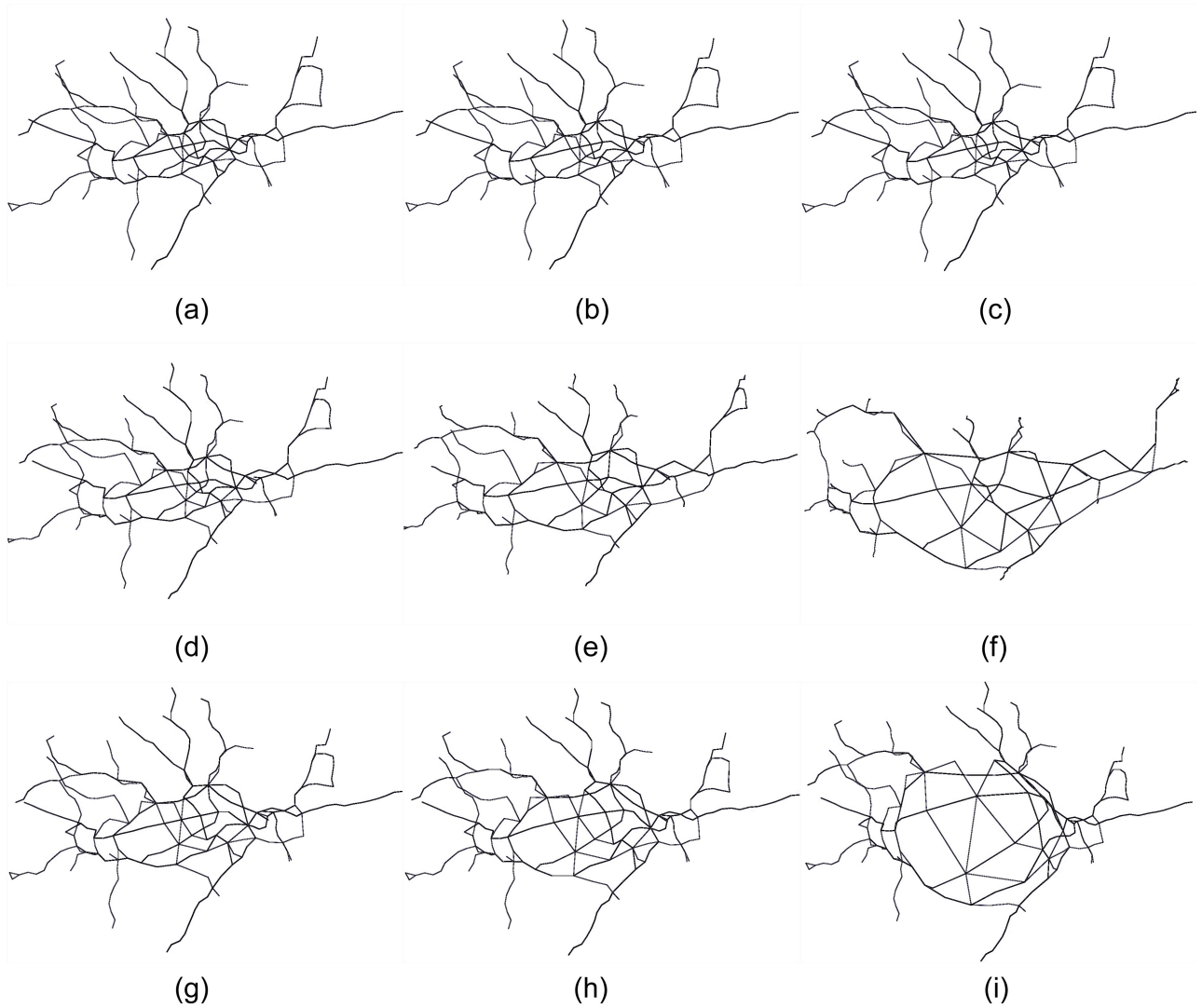


Figure 6: Scaling results for London Underground: (a) square root degree, (b) linear degree, (c) quadratic degree, (d) square root betweenness, (e) linear betweenness, (f) quadratic betweenness, (g) square root hubness, (h) linear hubness, (i) quadratic hubness.

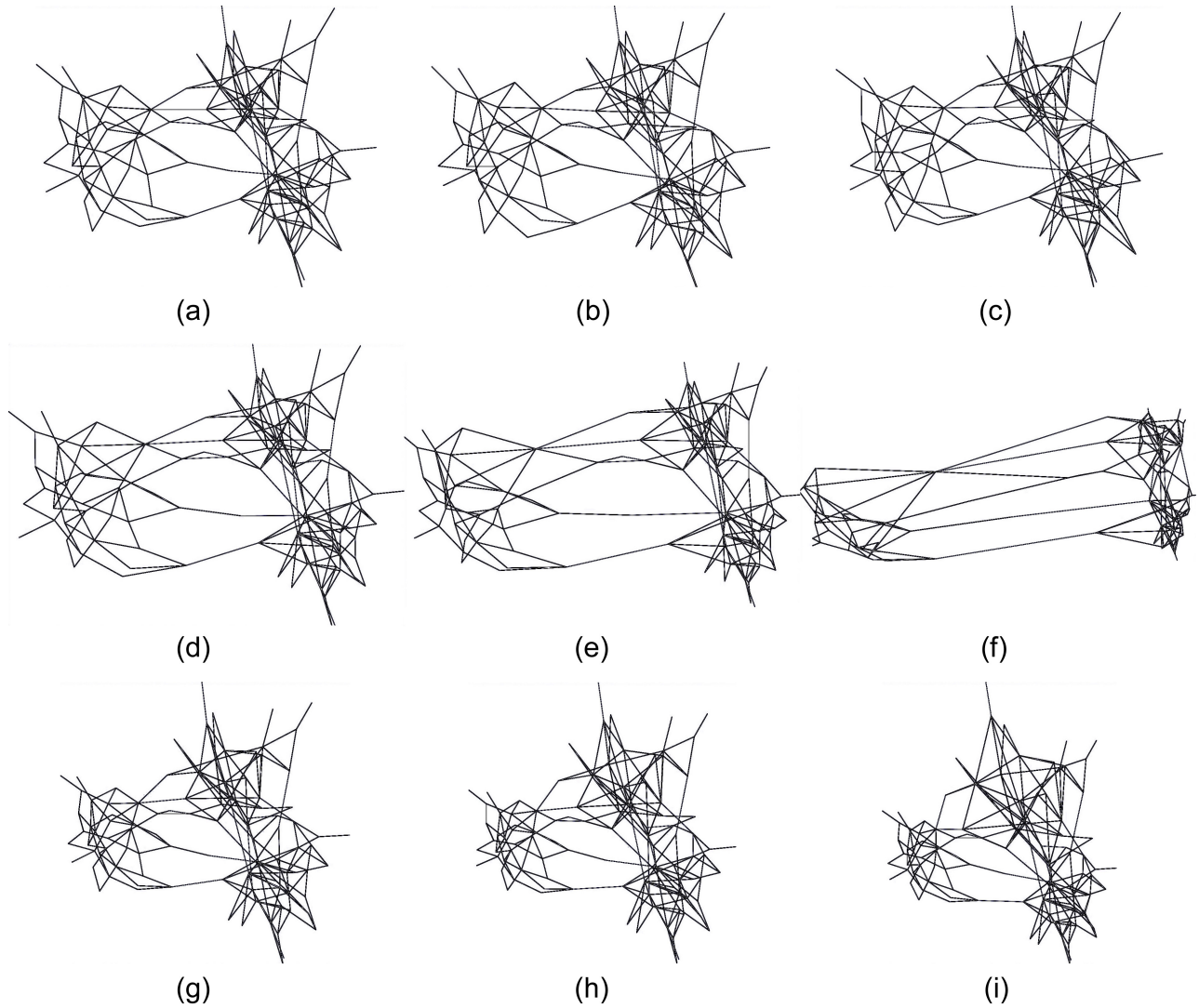


Figure 7: Scaling results for RG1: (a) square root degree, (b) linear degree, (c) quadratic degree, (d) square root betweenness, (e) linear betweenness, (f) quadratic betweenness, (g) square root hubness, (h) linear hubness, (i) quadratic hubness.

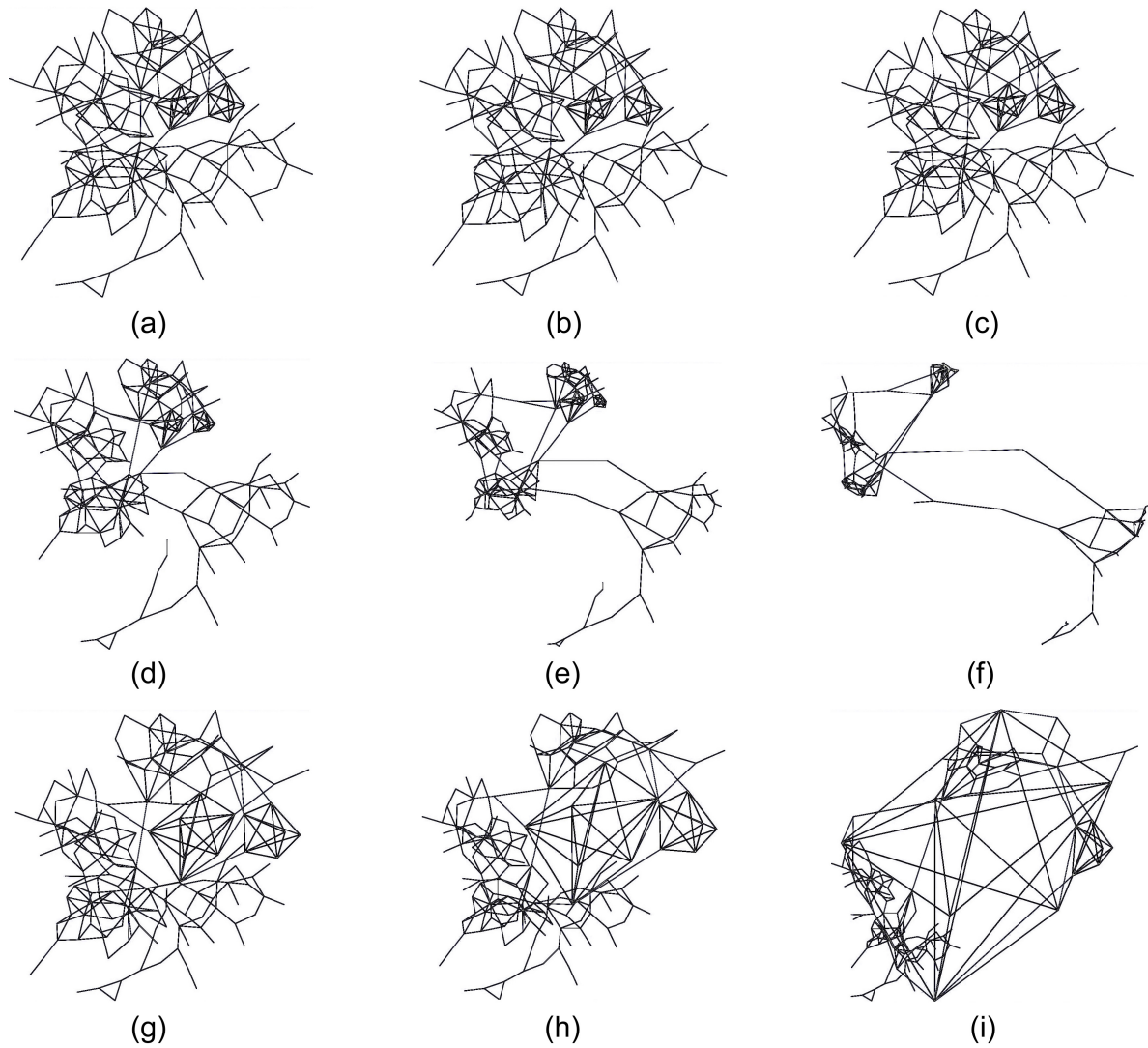


Figure 8: Scaling results for RG2: (a) square root degree, (b) linear degree, (c) quadratic degree, (d) square root betweenness, (e) linear betweenness, (f) quadratic betweenness, (g) square root hubness, (h) linear hubness, (i) quadratic hubness.

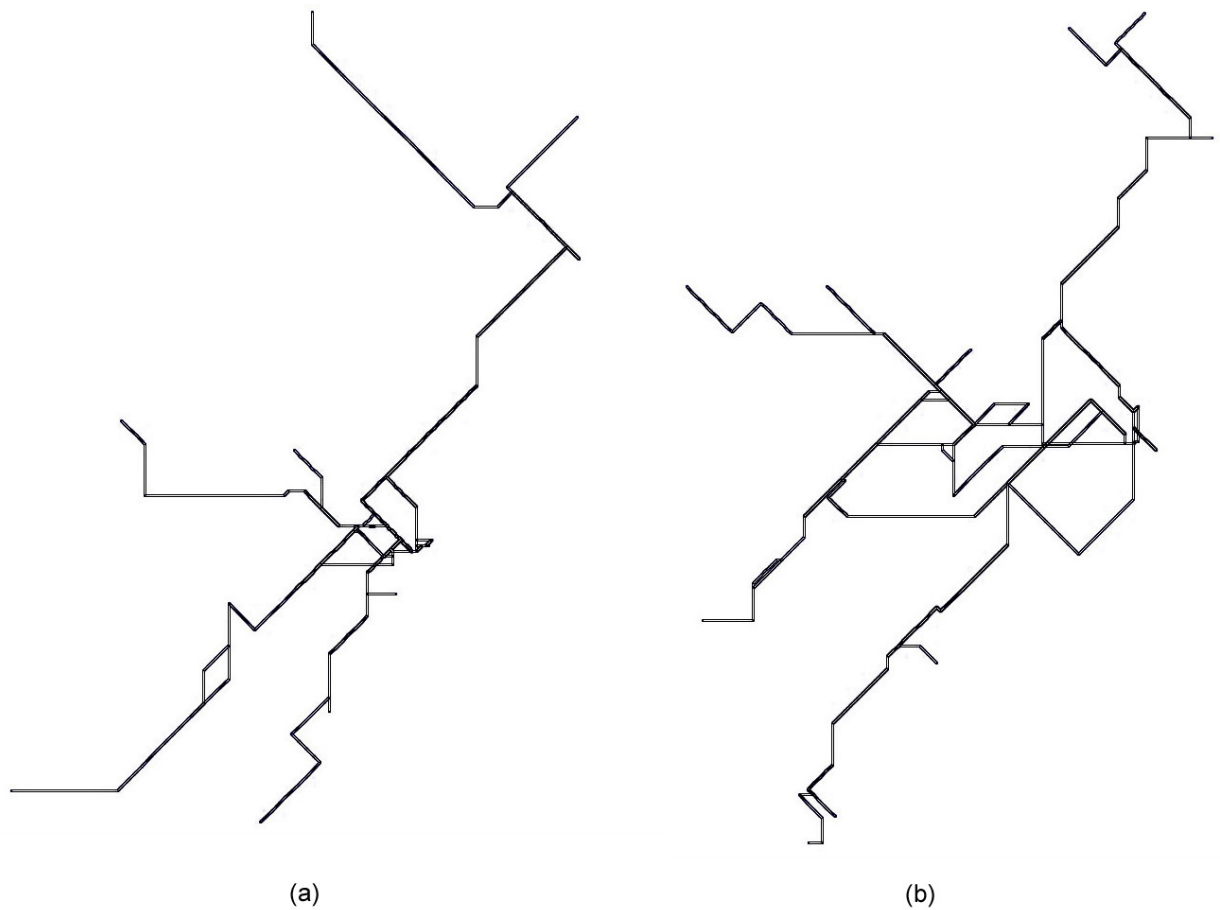


Figure 9: Schematised versions of the Sydney Cityrail network: (a) schematised original geography, (b) schematised quadratic hubness scaling.

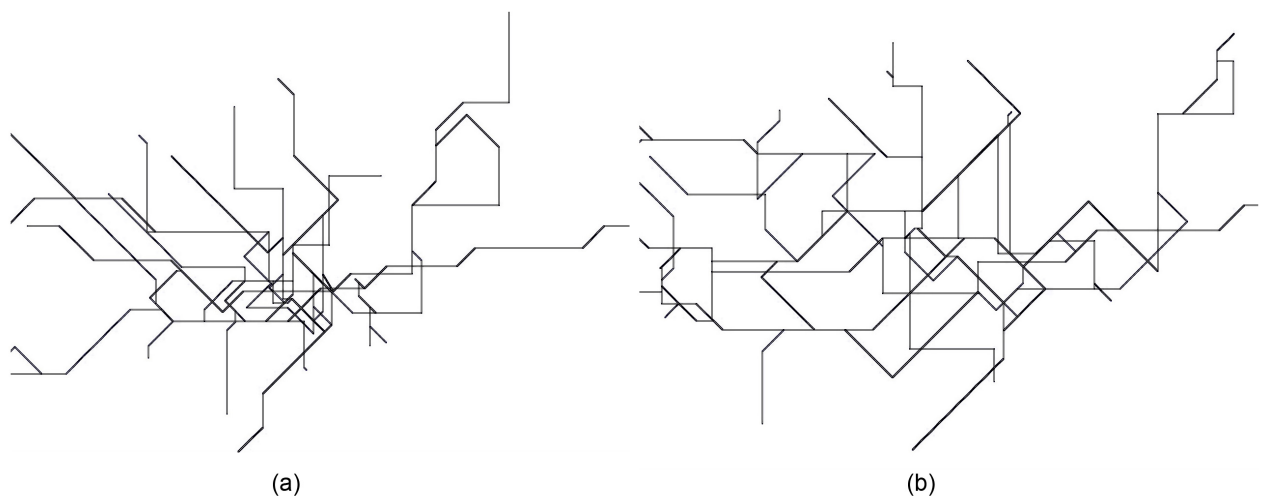


Figure 10: Schematised versions of the London Underground network: (a) schematised original geography, (b) schematised linear betweenness scaling.

Visualizing Multivariate Network on the Surface of a Sphere

Yingxin Wu¹

Masahiro Takatsuka²

School of Informatics Technologies

The University of Sydney,

NSW 2006 Australia

Email: ¹chwu@it.usyd.edu.au ²masa@vislab.net

Abstract

A multivariate network is a graph whose nodes contain multi-dimensional attributes. We propose a method to visualize such a network using spherical Self-Organizing Map (SOM) and circular layout. The spherical SOM produced an initial graph layout by grouping nodes with similar attributes to adjacent areas on the sphere. The circular layout algorithm fine tunes the position of each node to avoid node/node-edge overlap and minimize the number of edge crossings. We also implement an interface to project the spherical image onto a 2D plane such that the user can have a global view of the data set. The user is allowed to select any point of interest to be the center of the 2D projection.

Keywords: Multivariate Network, GeoSOM, Graph Drawing, Circular Layout, Information Visualization

1 Introduction

Two distinctive types of data are of particular interest to the information visualization community. One is multi-dimensional data and the other is relational data. Techniques for visualizing these two types of data are very different. For multi-dimensional data, techniques focus mainly on constructing low-dimensional representations (typically 2D or 3D) that preserve certain property of the original data — either preserving distance or topology. The most popular methods include Principle Component Analysis (Hotelling 1933), Sammon's mapping (Sammon 1969) and Self-Organizing Maps (Kohonen 2001). For relational data, researchers emphasize the readability of the layouts — the position of the nodes or bending of the edges are optimized according to certain aesthetic criteria. These criteria reflect the preference of human perception, such as symmetries and few edge crossings. Force directed approaches such as spring embedder (Eades 1984), Kamada and Kawai's algorithm (Kamada & Kawai 1989) are the most common approaches to layout general graphs.

Advances have been made in each of these two fields. However, algorithms rarely take both data types into consideration. In the real world, quite a lot of data contain both relational information and multivariate attributes. For example, in a world trade network, countries are represented as nodes and their trading as edges. Furthermore, each country has attributes like gross domestic product (GDP), GDP

growth, population and population growth. Such networks are generally called multivariate networks. Using existing visualization method, it is difficult to reveal the two aspects of the networks at the same time. Hence, answers to either of these questions won't be straightforward:

- Do major trading partners have similar attributes?
- Are countries with similar attributes major trading partners?

To address this deficiency, separate windows have been used to display these information. For example, the GGobi system draws a telecommunication network in one window while the attributes of each node are display in another window (Swayne, Buja & Lang 2003). Selecting some data in one view causes the corresponding data in the other view to change color. The disadvantage of this method is that the user needs to compare two pictures to comprehend the data. We are introducing a method to visualize all the information in one picture. The basic idea is first to use SOM to find an initial layout of the graph. We alter the training process of SOM to consider both the multi-dimensional attributes and graph relations. We then use a graph drawing algorithm to produce a better layout.

The reminder of this paper is organized as follows. Section 2 briefly introduces the self-organizing map and its advantages over other dimensional reduction methods. Section 3 discusses why we use *spherical* self-organizing map and layout the graph on the sphere. Section 4 presents the details of our method and section 5 describes the mechanism for projecting spherical images onto a 2D plane using a cartographic approach. Conclusions and future work are presented in section 6.

2 Self-Organizing Maps

Self-Organizing Map (Kohonen 2001) is an widely used artificial neural network. The neurons are usually organized as 2D rectangular/hexagonal grid. Each neuron represent a set of inputs with a weight vector \vec{w}_i . Before training, the components of each weight vector are initialized as random numbers. Training of a SOM is based on competitive unsupervised learning: when an input \vec{x} arrives, the neuron c whose weight vector is nearest to the input win the competition:

$$c = \arg \min_i \|\vec{x} - \vec{w}_i\|^2 \quad (1)$$

Then the winning neuron together with its neighbors update their weight vectors to represent the data better. The update amount is changing with time t :

$$\vec{w}_i(t+1) := \vec{w}_i(t) + \alpha(t) * h_{ci}(t) * (\vec{x}(t) - \vec{w}_i(t)), \quad (2)$$

Copyright ©2006, Australian Computer Society, Inc. This paper appeared at Asia-Pacific Symposium on Information Visualization (APVIS 2006), Tokyo, Japan, February 2006. Conferences in Research and Practice in Information Technology, Vol. 60. K. Misue, K. Sugiyama and J. Tanaka, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

Here $\alpha(t)$ is the learning rate which decreases with time; $h_{ci}(t)$ is a decreasing function of the distance between the winning neuron c and its neighbors i on the grid. All the neighbors within a radius r of the winning neuron are updated. Usually, the update radius is large at the beginning of the training and shrinking with time. Due to the training process, SOM behaves like a “elastic net” unfolding into the high-dimensional data space. Neighboring neurons tend to model similar regions of the data space, which is SOM’s topology preserving characteristic.

In SOM, point density $p(m)$ means how many neurons SOM uses to represent a certain area in the high-dimensional space. Ritter(Ritter 1991) proved that the asymptotic point density for the one-dimensional SOM with one-dimensional input is:

$$p(m) \propto p(x)^{\frac{2}{3} - \frac{1}{3\sigma^2 + 3(\sigma+1)^2}} \quad (3)$$

where σ is the update radius. This result shows that the SOM roughly follows the density of the training data: the higher the density, the bigger number of neurons are allocated to represent that region in high-dimensional space. Because each neuron occupies the same amount of area on the 2D grid, the total area allocated to represent a cluster on the SOM is somehow proportional to the density of the cluster.

This characteristic makes SOM more suitable for our task than other linear/nonlinear projection methods: for graph drawing, dense areas require more resolution to make the graph readable. Approaches like Principle Component Analysis(PCA) try to maintain the pairwise distance between the data as faithfully as possible. So the area allocated to a cluster on the 2D data map is proportional to the diameter of the cluster in high-dimensional space. This phenomenon is shown in Figure 1. It shows the 2D projections of the same data set using PCA and SOM (Vesanto 2002). Different colors represent different data clusters. Although the yellow and orange cluster have the same amount of data as the pink cluster, they are projected to a very tiny area on the data map produced by PCA, while the area of these three cluster are approximately the same on SOM.

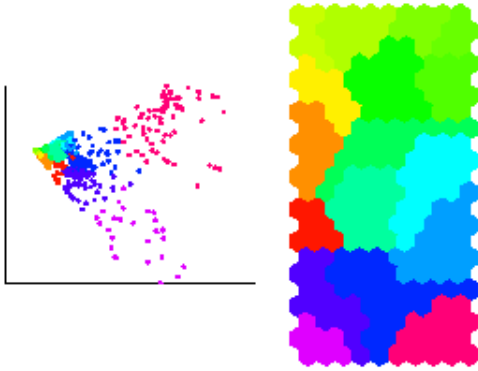


Figure 1: Left: Projection using Principle Component Analysis; Right: Projection using Self-Organizing Map.

3 Spherical Self-Organizing Maps and Spherical Graph Drawing

3.1 Spherical SOM Has Lower Quantization Error

The quality of a SOM can be measured by quantization error, which is the average distance between an

input x_i and its best matching neuron w_{bi} :

$$E_q = \frac{1}{N} \times \sum_{i=1}^N \|\vec{x}_i - \vec{w}_{bi}\| \quad (4)$$

where N is the number of inputs. Generally speaking, lower quantization error indicates better quality.

The 2D SOM has a notorious “border effect”: neurons at the boundaries of the 2D grid have fewer neighbors than those inside the map, so they have less chance to be updated. This not only lowers the SOM’s accuracy, but also makes it harder to converge. One solution is to implement the SOM on a spherical grid, thus removing all the boundaries. In (Wu & Takatsuka 2005) we developed the GeoSOM, which used icosahedron-based geodesic dome as the lattice (see Figure 2). Several experiments were carried out to compared the GeoSOM and the 2D SOM. The GeoSOM can reduce about 25% of the quantization error while running at approximately the same speed as the 2D SOM.

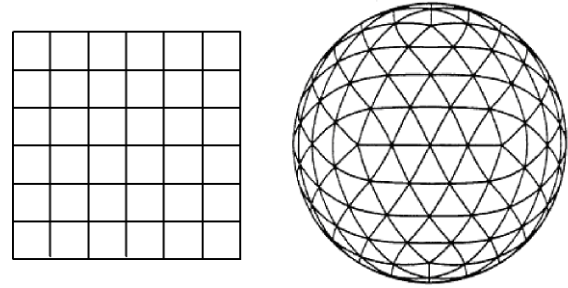


Figure 2: A rectangular grid and a four frequency icosahedron.

3.2 Spherical SOM Shows More Information

2D SOMs have boundaries, so cannot easily show the relationships between data mapped to the borders. An example is shown in Figure 3. We generated an artificial data set which containing 7 clusters in three-dimensional space. Each cluster contains 500 normally distributed points. Figures 3(b) and 3(c) are, respectively, the corresponding GeoSOM and 2D SOM. As can be seen, clusters 5 and 4 are close to each other in the original space. However, on the 2D SOM, they are mapped to opposite borders, which makes them appear to be far apart. The GeoSOM clearly shows that the two clusters are adjacent.

3.3 Graph Drawing on the Sphere

Compared to a 2D plane, a spherical surface provides a natural fisheye effect which enlarges the focus point and shows other portions of the image with successively less detail. The fisheye effect is a popular technique for visualizing large graphs(Herman, Melançon & Marshall 2000). As pointed out in (Meyer 1998), real spherical 3D-layout that allows interactive rotation opens a novel interaction for graph navigation. In (Kobourov & Wamplery 2004), the 2D spring-embedder was extended to layout graphs on the sphere. In this paper, we use a spherical circular layout to fine tune the nodes’ positions. Details are shown in the next section.

4 Details of Our Approach

A relative small data set is used to demonstrate our approach — the import and export of manufactured metal between 32 countries in the year

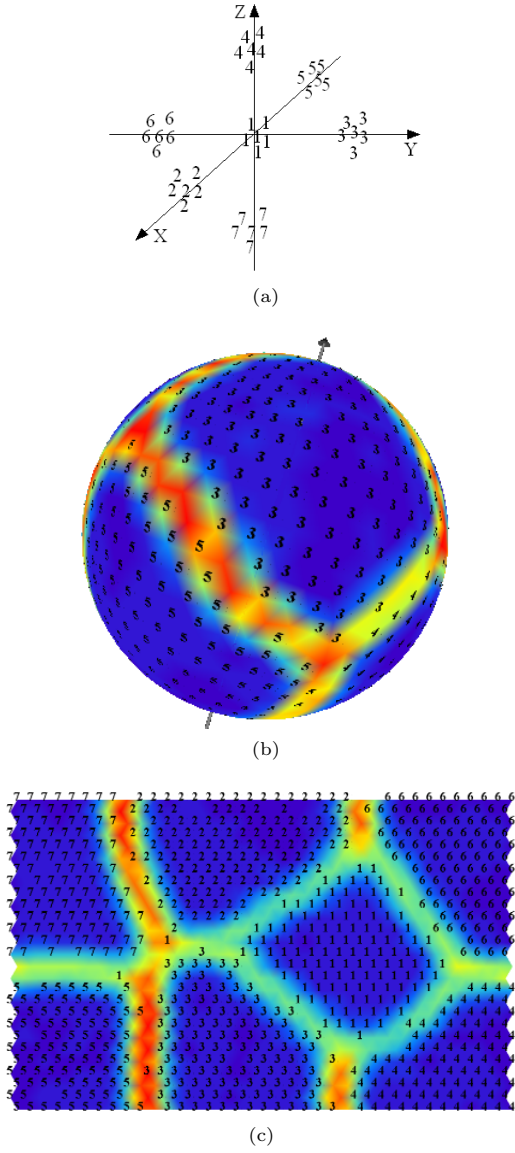


Figure 3: (a)The data set contains 7 clusters; (b) The GeoSOM; (c)The 2D SOM.

1994. It is extracted from the Pajek (Nooy, Mrvar & Batagelj 2005) database. Each country has four attributes: GDP, GDP growth, population and population growth. There are 62 edges in total. The edges are directed, pointing from exporting countries to importing countries.

There are two stages in our approach to visualize the multivariate network. The first stage is dimensional reduction using GeoSOM. In this stage, nodes close in high-dimensional space are mapped to similar areas on the sphere. Several nodes might be mapped to the same neuron (the same point on the sphere). So, in the second stage, they are arranged on a circle surrounding the neuron. For each node, our algorithm try to find a position on the circle that minimizes the number of edge crossings and avoids node-node and node-edge overlap.

4.1 The First Stage — Training of the GeoSOM

4.1.1 Linear Initialization

The GeoSOM use a two frequency icosahedron as the lattice which contain 42 neurons. Instead of random initialization, we first calculate the covariance ma-

trix of the data set and then use its two principle eigenvectors to initialize the neurons. This is called linear initialization. It “has the effect of stretching the SOM to the same orientation as the data having most significant amounts of energy” (Hollmén 1996). So the training result are more stable and the SOM converges faster than random initialization.

The original linear initialization method is designed for the rectangular array: the two eigenvectors are arranged along the X and Y axis of the grid. Some modifications are needed in our case. The eigenvector with the largest eigenvalue is arranged along the parallels while the next eigenvector is arranged along the meridian. So the weight vectors are initialized as:

$$\vec{w}_i = \vec{m} + \alpha_1 \theta R \cos \varphi \vec{e}_1 + \alpha_2 \varphi R \vec{e}_2 \quad (5)$$

Here, \vec{m} is the mean of the data set; α_1 and α_2 are two constants controlling the amount of change along the eigenvectors \vec{e}_1 and \vec{e}_2 ; R is the radius of the sphere which is one in our program; θ and φ are the longitude and latitude of the neuron (see Figure 4).

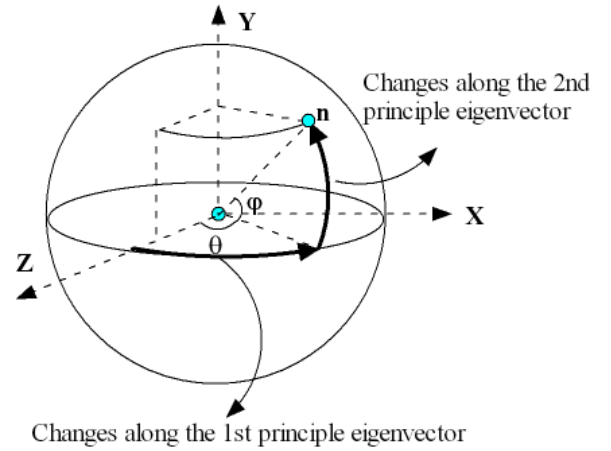


Figure 4: Linear initialization of the GeoSOM.

4.1.2 Batch Mode Training

The training rule introduced in section 2 is called incremental training. The number of updating steps typically needs to be at least 500 times the number of neurons for the SOM to converge. There is a batch version of the SOM which runs much faster than the incremental version and the training results are approximately the same. In each updating step, every neuron collect a list of input data which are closest to its weight vector. Then the neuron's weight vector is set to be the mean over the data mapped to it and its neighborhood:

$$\vec{w}_j = \frac{\sum_{i=1}^N h_{b_{ij}}(t) \vec{x}_i}{\sum_{i=1}^N h_{b_{ij}}(t)} \quad (6)$$

where b_i is the best matching neuron of input \vec{x}_i and N is the total number of inputs mapped to the neuron j and its neighbors; in our program, $h_{b_{ij}}$ is chosen to be a Gaussian function:

$$h_{b_{ij}}(t) = \exp\left(-\frac{\|\vec{r}_i - \vec{r}_j\|}{2\sigma(t)^2}\right) \quad (7)$$

where $\sigma(t)$ is the update radius decreasing linearly with time, and \vec{r}_i and \vec{r}_j are the position vectors of two neurons.

From a graph drawing point of view, nodes which are connected should be close to each other. So we

modified the batch training rule by adding a constant β :

$$\vec{w}_j = \frac{\sum_{i=1}^N \beta h_{b_{ij}}(t) \vec{x}_i}{\sum_{i=1}^N h_{b_{ij}}(t)} \quad (8)$$

$\beta = 1$ if there are edges connecting the data mapped to neuron i and neuron j ; $\beta = 0.85$ if the data are not connected. This has the effect of dragging together the data which are connected. Figure 5 shows the GeoSOM trained for 30 epochs with the world trade data set. In this visualization, the average distances between each neuron and its direct neighbors' weight vectors were calculated and coded with colors. The colors are linearly changing from blue, cyan, yellow to orange. Blue denotes the smallest variance between the weight vectors, and orange the largest. So data mapped to the same blue area are close in high-dimensional space and can be considered to lie in the same cluster. Each edge is drawn as a geodesic arc. A geodesic arc is the shorter part of the great circle that goes through the two nodes. It is the shortest path connecting any two points on the sphere.

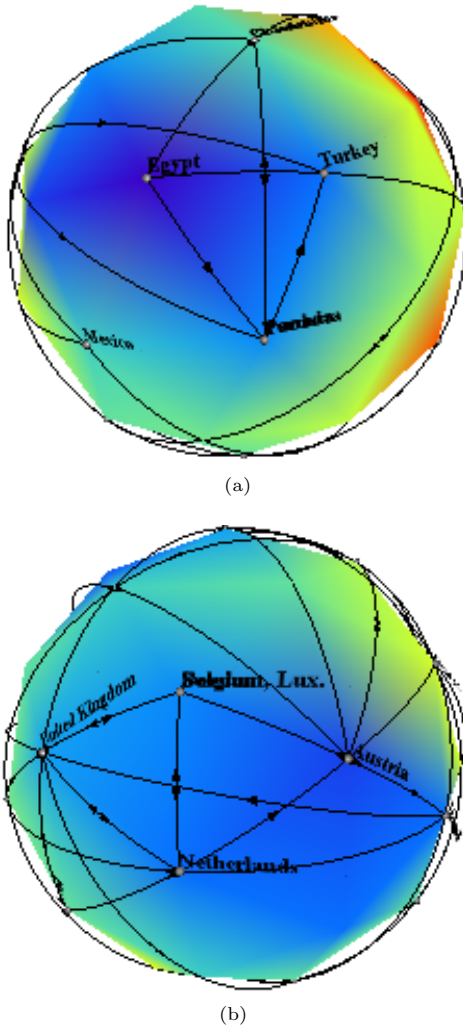


Figure 5: Two views of the GeoSOM trained with the world traded data set for 30 epochs.

4.2 The Second Stage — Circular Layout on The Sphere

As can be seen from Figure 5, some neurons have more than one inputs mapped to them. For example, Panama, Tunisia and Peru are projected to the same neuron. These graph nodes should be separated, but

should not be placed too far away from their best matching neuron or the topology of the data map would be destroyed. Also, moving these nodes too far apart would prevent the use of colors to indicate how adjacent data are related and hence show the cluster boundaries.

We arrange the nodes on a circle surrounding the neuron. The radius of the circle is set to be one-third of the average distance between two neurons on the sphere. On each circle, we generate several equally separated positions. The number of positions is set to be three times the number of nodes the neuron collects. Each node can be placed on one of these positions or remains at the position of its best matching neuron (see Figure 6).

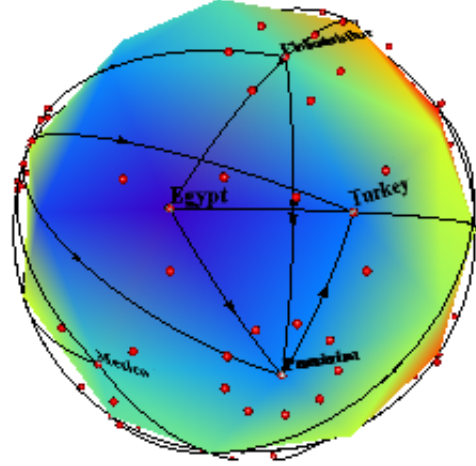


Figure 6: The small red spheres shows the positions where nodes at each neurons can be placed.

The algorithm try to minimize the number of edge crossings and avoid node-node/node-edge overlap when choosing the positions for the nodes. The node-node overlap is solved by not allowing any two nodes to be placed on the same position. The crossing minimization problem is NP complete (Gary & Johnson 1983). Currently, we tackle the problem by greedy search: place a node at each of the positions; calculate a cost function which sums up the number of edge crossings and the number of node-edge overlaps; place the node at a position with the lowest cost. This process is run for several iterations until the nodes' positions are relatively stable.

4.2.1 Detect Edge Crossings on the Sphere

On the 2D plane it is easy to detect whether two line segments intersect because they can be expressed as linear equations. Two steps are needed: calculate the intersection point of two straight lines and then test whether the point is inside both of the line segments.

Calculating the intersection point of two edges on the sphere is more complicated. Remember that an edge is part of the great circle which is the intersection between a sphere and a plane that goes through the center of the sphere. In our program, the center of the sphere is at the origin. Using Euclidean coordinates, the equation for a great circle is:

$$\begin{cases} x^2 + y^2 + z^2 = R^2 \\ ax + by + cz = 0 \end{cases} \quad (9)$$

Using spherical coordinates, the equations for a great circle are second ordered partial differential

Algorithm 1 Circular layout on the sphere

```

1: for each neuron do
2:   Calculate the equal distance positions;
3: end for
4:  $m = 1$ ;
5: repeat
6:   for each neuron do
7:     for each nodes  $n$  the neuron collects do
8:        $cost = \infty$ ;
9:        $pos = null$ 
10:      for each unoccupied position  $p$  do
11:        Find the number of edge crossings  $c_1$ ;
12:        Find the number of node-edge overlap
13:         $c_2$ ;
14:         $c = c_1 + c_2$ ;
15:        if  $c < cost$  then
16:           $cost = c$ 
17:           $pos = p$ ;
18:        end if
19:      end for
20:       $n.pos = pos$ ;
21:      Mark  $pos$  occupied;
22:    end for
23:     $m = m + 1$ ;
24: until  $m > \text{predefined running times}$ 

```

equations(Kreyszig 1968):

$$\begin{cases} \frac{d^2\theta}{dt^2} + \frac{2\cos\phi}{\sin\phi} \frac{d\theta}{dt} \frac{d\phi}{dt} = 0 \\ \frac{d^2\phi}{dt^2} - \sin\phi \cos\phi \left(\frac{d\theta}{dt}\right)^2 = 0 \end{cases} \quad (10)$$

Although the above equations can be solved by numerical methods, the computation cost is too expensive for our purpose.

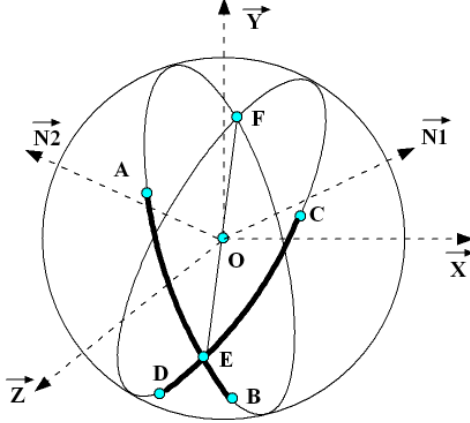


Figure 7: Calculate the intersection point of two edges on the sphere.

Fortunately, the intersection point of two great circles can be obtained geometrically as shown in Figure 7. Given two edges AB and CD, we first calculate the normals, \vec{N}_1 of plane AOB and \vec{N}_2 of the plane COD:

$$\vec{N}_1 = \vec{OA} \times \vec{OB}, \quad (11)$$

$$\vec{N}_2 = \vec{OD} \times \vec{OC} \quad (12)$$

Planes AOB and COD intersect at a straight line which in turn intersects the sphere at two points, E and F. They are the intersection points of the two great circles going through, respectively, AB and CD. The position vector of E and F can be calculated as:

$$\vec{OE} = \frac{\vec{N}_1 \times \vec{N}_2}{\|\vec{N}_1 \times \vec{N}_2\|}, \quad (13)$$

$$\vec{OF} = -\vec{OE} \quad (14)$$

The next step is to check whether E or F are contained in both of the two edges. Take point E for example. We calculate the middle point M of edge AB. If the geodesic arc length EM is less than half of the length of AB, then E is inside the edge (see Figure 8). Whether E is inside edge CD can be checked in a similar way.

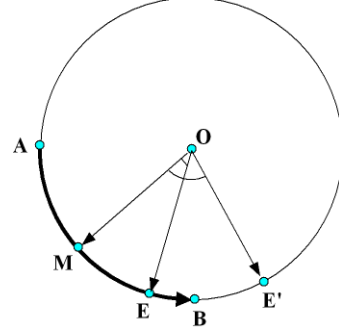


Figure 8: Check whether the intersection point E is inside edge AB.

4.2.2 Node-Edge Distance on the Sphere

A graph node should not be too close to edges which are not incident to the node; otherwise, the user could be confused about which are the edges' end points. In our program, the cost function will increase if the distance between a node and an edge is smaller than a threshold.

Given an edge AB and a node C on the sphere, we first need to find out a point P on the great circle G1 defined by AB such that the geodesic arc length of CP is the shortest. The point P can be obtained as follows (see Figure 9). First calculate the normal of plane AOB using equation (11). It intersects the sphere at point N. The great circle G2 defined by C and N is perpendicular to G1. In fact, P is one of the intersection points of G1 and G2. The coordinates of P can be calculated using the method introduced in the previous subsection.

If the geodesic arc length of CP is bigger than the threshold, then C is not too close to AB; otherwise, we need to check whether P is inside edge AB. If yes, then C is too close to AB and the cost function needs to be increased.

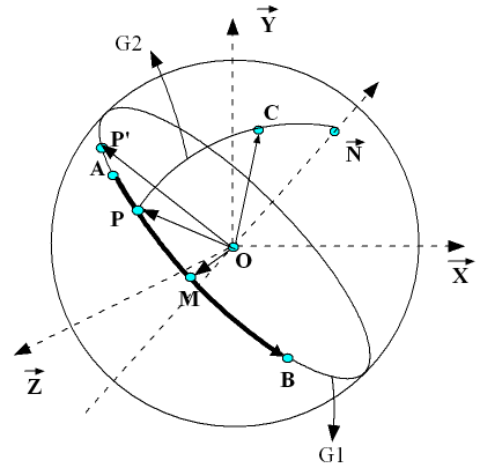


Figure 9: Distance between a node and an edge on the sphere

4.3 The Results

The program is run on a PC with a 3GHZ CPU and 1G of memory. For this small data set, SOM training took less than one second while the circular layout took 13 seconds. The final layout is shown in Figure 10. Compared to a normal graph layout, more trading patterns can be observed. Countries like Panama, Tunisia and Peru are close in the attributes space but there is no trading between them. On the other hand the European countries such as United Kingdom, Belgium and Netherland are major trading partners and also have very similar node attributes.

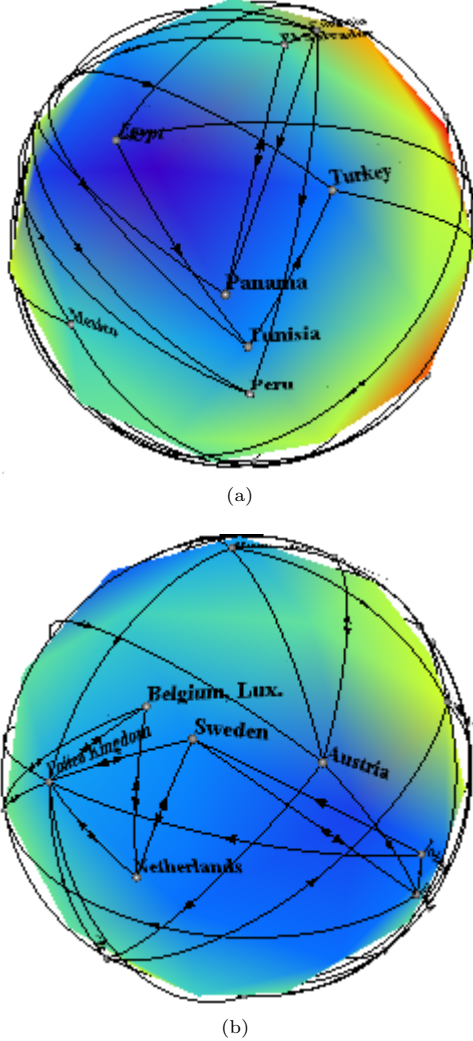


Figure 10: Final layout of the world trade network.

5 Two Dimensional Projection of the Spherical Image

The spherical image is not convenient for users to have a global view of the entire data set. Although they can rotate to see every part of the image, they have to remember the other half that a flat screen cannot simultaneously show. Hence, we implement an interface to project the spherical image onto a 2D plane. Currently, the Wagner III pseudocylindrical projection is used. This projection is neither equal-area or conformal. However, it has less extreme distortions and therefore can give a more balanced representation of the sphere's main features (Canters & Deleir 1989).

Figure 11 illustrates how the interface works. The user needs to select two points on the sphere. The first

point A will be the center of the 2D projection. The second point B together with A defines a plane going through the center of the sphere. Suppose \vec{OC} is the plane normal, then the sphere's central axis can be obtained by the cross product of the plane normals \vec{OC} and \vec{OA} . This axis intersect the sphere at two points, N and S, which are made the "north" and "south" poles of the sphere. The geodesic arc going through these two points and opposite to A becomes the split line to open up the sphere.

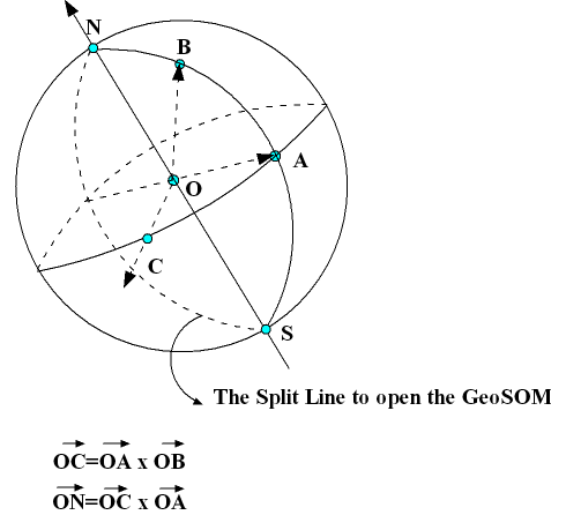


Figure 11: Select two point on the sphere. The first point A will be the center of the projection. The second point B and A defined the central axis and the split line to open the sphere.

The Wagner III pseudocylindrical projection is not conformal. It means the angle between two edges will be changed after projection. This will affect the node-edge distance on the 2D image and may lead to new node-edge overlap. In order to preserve the original distance, for each edge, we transform the whole geodesic arc into 2D. So the edges on the 2D images are curves instead of straight lines. Figure 12 shows the 2D projection using Mexico as the image center. Edges cross the split line will be cut into two segments and drawn separately.

6 Conclusion And Future Work

We introduced a method to visualize the multivariate network. A spherical SOM (GeoSOM) was used to visualize the high-dimensional information contained in the graph nodes and produce the initial layout. We then implement a circular layout algorithm to fine-tune the nodes' positions, which tried to minimize the number of edge crossings and avoid node-node/node-edge overlaps.

However, our approach is only an initial attempt. There are many aspects that need to be improved — in particular, the circular layout. Basically, the problem is graph drawing with the geometric restriction that nodes cannot be placed too far away from the position of their best matching neurons. Currently, we restrict the nodes to lie on a circle. This maybe too restrictive for a good graph layout. Also, the greedy searching approach to find the right position for each node is time consuming. At each allowable position, we need to compute the crossing numbers and calculate the distance between each node to every other edge. The computational complexity is $O(nm + m^2)$, where n is the number of nodes and m is the number

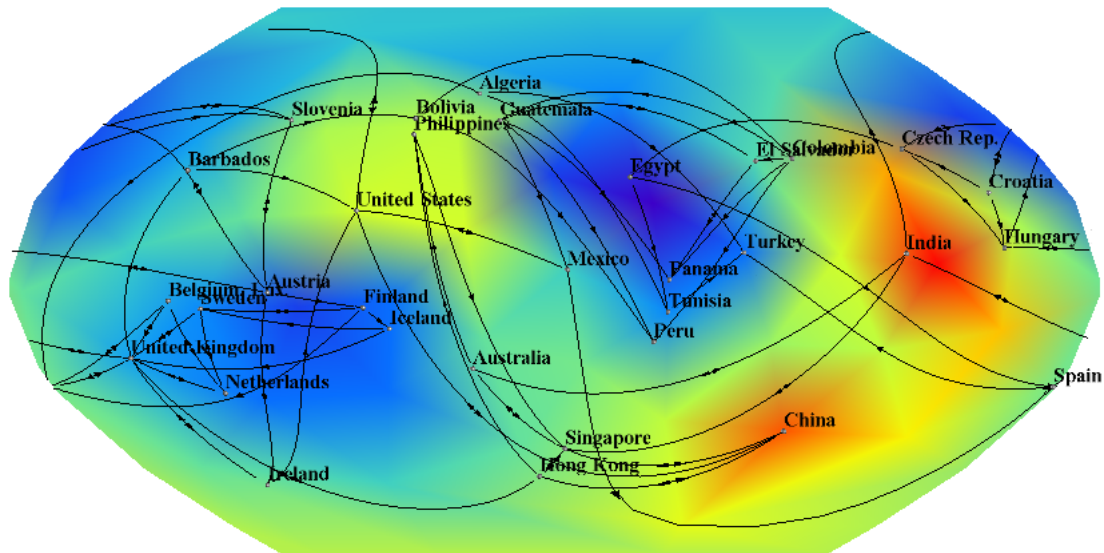


Figure 12: Project the spherical image on to 2D plane such that user can have a global view of the data set. Mexico is chosen to be the center of projection.

of edges. This would be too slow for large networks. We are looking into other more efficient combinatorial methods. Finally, for large networks, it is impossible to show all the nodes and edges; some abstraction methods, such as level-of-detail zooming, are needed.

References

- CanTERS, F. & DeCleir, H. (1989), *The World In Perspective: A Directory of World Map Projections*, John Wiley & Sons Ltd., England.
- Eades, P. (1984), 'A heuristic for graph drawing', *Congressus Numerantium* **42**, 149 – 160.
- GARY, M. R. & JOHNSON, D. S. (1983), 'Crossing number is NP-complete', *SIAM J. Algebraic and Discrete Methods* **4**, 312 – 316.
- HERMAN, I., MELANÇON, G. & MARSHALL, M. S. (2000), 'Graph visualization and navigation in information visualization: A survey', *IEEE Transactions on Visualization and Computer Graphics* **6**, 24 – 43.
- HOLLMÉN, J. (1996), Process modeling using the self-organizing map, Master's thesis, Department of Computer Science, Helsinki University of Technology.
- HOTELLING, H. (1933), 'Analysis of a complex of statistical variables into principal', *Journal of Educational Psychology* **24**, 417 – 441, 498 – 520.
- KAMADA, T. & KAWAI, S. (1989), 'An algorithm for drawing general undirected graphs', *Information Processing Letters* **31**, 7 – 15.
- KOBOUROV, S. G. & WAMPLERY, K. (2004), Non-euclidean spring embedders, in T. Munzer & S. North, eds, 'Proceeding of The IEEE Symposium On Information Visualization 2004 (Infovis2004)'.
- KOHONEN, T. (2001), *Self-Organizing Maps*, Springer Series in Information Sciences, 3rd edn, Springer-Verlag, Germany.
- KREYSZIG, E. (1968), *Introduction to Differential Geometry and Riemannian Geometry*, University of Toronto Press.
- MEYER, B. (1998), Self-organizing graphs - a neural network perspective of graph layout, in S. Whitesides, ed., 'Proceedings of the 6th International Symposium on Graph Drawing', Springer-Verlag, London, UK, pp. 246 – 262.
- NOOY, W., MRVAR, A. & BATAGELJ, V. (2005), *Exploratory Social Network Analysis with Pajek*, CAMBRIDGE UNIVERSITY PRESS, 40 West 20th Street, New York, NY 10011-4211, USA.
- RITTER, H. (1991), 'Asymptotic level density for a class of vector quantization processes', *IEEE Transactions on Neural Networks* **2**, 173 – 175.
- SAMMON, J. W. (1969), 'A nonlinear mapping for data structure analysis', *IEEE Transactions on Computers* **18**, 401 – 409.
- SWAYNE, D. F., BUJA, A. & LANG, D. T. (2003), Exploratory visual analysis of graphs in GGobi, in 'Third Annual Workshop on Distributed Statistical Computing (DSC 2003)', Vienna.
- VESANTO, J. (2002), Data Exploration Process Based on the Self-Organizing Map, PhD thesis, Department of Computer Science and Engineering, Helsinki University of Technology.
- WU, Y. & TAKATSUKA, M. (2005), The geodesic self-organizing map and its error analysis, in V. Estivill-Castro, ed., 'Proceedings of the 28th Australasian Computer Science Conference (ACSC 2005)', Australian Computer Society, Inc, pp. 343 – 351.

An Experimental Study on Algorithms for Drawing Binary Trees

Adrian Rusu¹ Radu Jianu² Confesor Santiago³ Christopher Clement¹

¹Department of Computer Science, Rowan University, Glassboro, NJ 08028, USA

²Department of Computer Science, Brown University, Providence, RI 02912, USA

³Department of Electrical and Computer Engineering, Rowan University, Glassboro, NJ 08028, USA
rusu@rowan.edu, jr@cs.brown.edu, {santia09, clemen55}@students.rowan.edu

Abstract

In this paper we present the results of a comprehensive experimental study on the four most representative algorithms for drawing binary trees without distorting or occluding the information, one for each of the following *distinct* tree-drawing approaches: **Separation-Based approach** (Garg & Rusu 2003), **Path-Based approach** (Chan, Goodrich, Rao Kosaraju & Tamassia 2002), **Level-Based approach** (Reingold & Tilford 1981), and **Ringed Circular Layout approach** (Teoh & Ma 2002).

Our study is conducted on randomly-generated, unbalanced, and AVL binary trees with up to 50,000 nodes, on Fibonacci trees with up to 46,367 nodes, on complete trees with up to 65,535 nodes, and on real-life molecular combinatory binary trees with up to 50,005 nodes. We compare the performance of the drawing algorithms with respect to five quality measures, namely **Area**, **Aspect Ratio**, **Uniform Edge Length**, **Angular Resolution**, and **Farthest Leaf**. None of the algorithms have been found to be the best in all categories.

Keywords: binary trees, experimental study, drawing algorithm, planar, straight-line

1 Introduction

Trees are ubiquitous data-structures, arising in a variety of applications such as Software Engineering, Business Administration, Knowledge Representation, and Web-site Design and Visualization.

Visualizing a tree can enhance a user's ability in understanding its structure. Hence, a lot of research has been done on visualizing trees, which has produced a plethora of tree-drawing algorithms (see for example, (Chan, Goodrich, Rao Kosaraju & Tamassia 2002, Garg & Rusu 2003, Reingold & Tilford 1981, Teoh & Ma 2002, Valiant 1981)).

An experimental evaluation of the practical performance of tree-drawing algorithms can help a practitioner in choosing the algorithm most appropriate for her application. However, we are not aware of any experimental study done to compare the practical performance of tree-drawing algorithms. As a first step, in this paper, we present an experimental study of some well-known algorithms for drawing binary trees. These algorithms represent the *distinct*

approaches that have been used to draw binary trees without distorting or occluding the information.

A *binary tree* is one where each node has at most two children. In contrast to graphs, every tree accepts a *planar drawing*, i.e. without any crossings. A *straight-line drawing* has each edge drawn as a single line-segment. *Grid-based* algorithms place all the nodes of a drawing at integer coordinates. A drawing of a tree T has the *subtree separation* property if, for any two node-disjoint subtrees of T , the enclosing rectangles of the drawings of the two subtrees do not overlap with each other. All algorithms in our experimental study produce planar straight-line grid drawings and exhibit the subtree separation property.

2 The Algorithms Under Evaluation

We have compared four different algorithms for producing planar straight-line grid drawings of binary trees. The four algorithms can be classified into four categories on the basis of their approach to constructing drawings.

Separation-Based: In the *Separation-Based Approach*, a divide-and-conquer strategy is used to recursively construct a drawing of the tree, by performing the following actions at each recursive step: (1) *Find a Separator Edge:* A *separator edge* of a binary tree T is an edge, which, if removed, divides T into at most five smaller, partial binary trees. Every binary tree contains such an edge (Valiant 1981). (2) *Divide Tree:* Divide T into several partial binary trees by removing at most two nodes and their incident edges from it (including the separator edge). (3) *Assign Aspect Ratios:* Pre-assign a desirable aspect ratio to each partial binary tree. (4) *Draw Partial Trees:* Recursively construct a drawing of each partial binary tree using its pre-assigned aspect ratio. (5) *Compose Drawings:* Arrange the drawings of the partial binary trees and draw the nodes and edges that were removed from T to divide it such that the drawing of T thus obtained is a planar straight-line grid drawing. We have chosen to evaluate the $O(n)$ -area bottom-up algorithm of (Garg & Rusu 2003) (we call it *Separation*).

Path-Based: The *Path-Based Approach* uses a recursive winding paradigm as follows: first lay down a small chain of nodes from left to right until near a distinguished node v , and then recursively lay out the subtrees rooted at the children of v in the opposite direction. For our study, we have implemented the $O(n \log \log n)$ -area algorithm described in (Chan, Goodrich, Rao Kosaraju & Tamassia 2002) (we call it *Path*).

Level-Based: The *Level-Based Approach* is characterized by the fact that in the drawings produced, the nodes at the same distance from the root are horizontally aligned. For our study, we have implemented the recursive algorithm described in (Reingold &

Copyright ©2006, Australian Computer Society, Inc. This paper appeared at Asia-Pacific Symposium on Information Visualization (APVIS 2006), Tokyo, Japan, February 2006. Conferences in Research and Practice in Information Technology, Vol. 60. K. Misue, K. Sugiyama and J. Tanaka, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

Tilford 1981) (we call it *Level*). This algorithm uses the following steps: draw the subtree rooted at the left child, draw the subtree rooted at the right child, place the drawings of the subtrees at horizontal distance 2, and place the root one level above and halfway between the children. If there is only one child, place the root at horizontal distance 1 from the child.

Ringed Circular Layout: The algorithms based on the *Ringed Circular Layout Approach* place a node and all its children in a circle. For our study, we have implemented the algorithm described in (Teoh & Ma 2002) (we call it *Rings*). Note that this algorithm was designed for general trees. In this study, we have implemented and studied its performance for the particular case of binary trees. In this algorithm, equal-sized circles corresponding to children are placed in concentric rings inside of the parent circle, around its center, thus trying to minimize the space wasted inside of the interior of the parent circle.

Figure 1 shows drawings of the Fibonacci tree with 88 nodes constructed by the algorithms of our study.

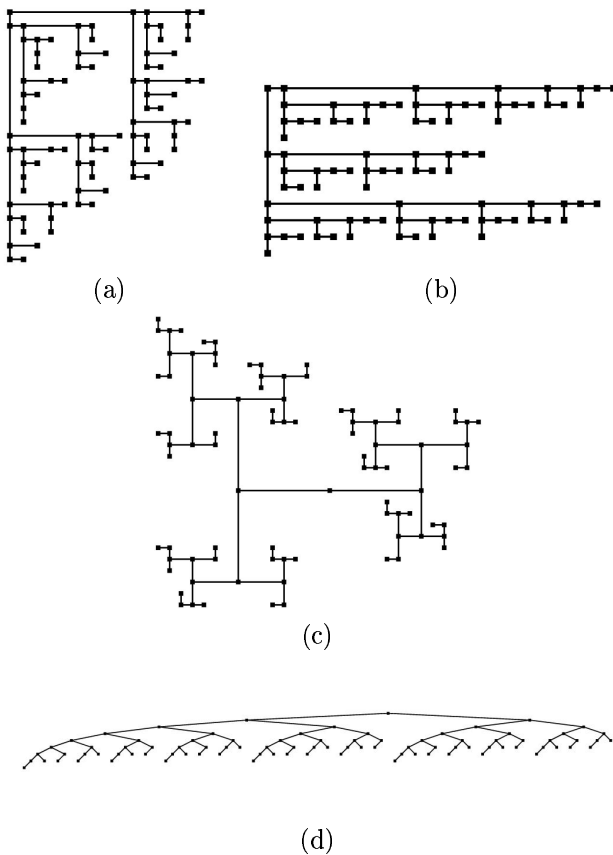


Figure 1: Drawings of the Fibonacci tree with 88 nodes, generated by the algorithms in our study: (a) Separation, (b) Path, (c) Rings, and (d) Level.

3 Experimental Setting

Our experimental setting consists of (i) a large suite of randomly-generated, unbalanced, complete, AVL, Fibonacci, and molecular combinatory binary trees of various sizes; (ii) five quality measures: area, aspect ratio, uniform edge length, angular resolution, and farthest leaf.

3.1 Test Suite

Our test suite consists of five binary trees for each of the following types: *random*, *complete*, *AVL*, *Fi-*

bonacci, and *unbalanced*. We consider a binary tree T_n with n nodes as *unbalanced* if its height is greater than $n/\log n$. A binary tree T_n with n nodes is *unbalanced-to-the-left* (*unbalanced-to-the-right*) if it is unbalanced, and, in addition, the number of left (right) children in T_n is greater than its number of right (left) children. We have generated random, unbalanced, and AVL binary trees with up to 50,000 nodes, complete trees with up to 65,535 nodes, and Fibonacci trees with up to 46,367 nodes. In addition, our test suite includes binary trees from molecular combinatory programs. The data was obtained from the study in (MacLennan 2003) by Dr. Bruce MacLennan at the University of Tennessee. For our experimental analysis, he has generated molecular combinatory binary trees with up to 50,005 nodes.

3.2 Quality Measures

The following four well-known quality measures have been considered:

- **Area:** the number of grid points contained within the enclosing rectangle.
- **Aspect Ratio:** the ratio of the smaller and the longer sides of the enclosing rectangle.
- **Uniform Edge Length:** the variance of the edge lengths in the drawing.
- **Angular Resolution:** the smallest angle formed by two edges incident on the same node.

It is widely accepted (Di Battista, Eades, Tamassia, & Tollis 1999) that small values of the area and uniform edge length are related to the perceived aesthetic appeal and visual effectiveness of the drawing. In addition, an aspect ratio is considered *optimal* if it is equal to 1. High angular resolution is desirable in visualization applications and in the design of optical communication networks. For binary trees, the degree of a node is at most 3, hence a trivial upper bound on the angular resolution is 120° .

We have also considered a new quality measure, specially designed for trees: **Farthest Leaf:** the largest Euclidean distance between the root and a leaf in the drawing. Farthest leaf helps determine whether the algorithm places leaves far from the root. It is important to minimize the distance between the root and the leaves of the tree, especially in the case when the user wants to visually analyze binary search trees.

4 Experimental Analysis

Let T_n be a binary tree with n nodes that is provided as input to the algorithms being evaluated.

Two of the algorithms chosen in this study, namely *Separation* and *Path*, allow user-controlled aspect ratio. The other two algorithms, namely *Level* and *Rings*, generate unique drawings for each value of n . In order to find the parameters for which *Separation* and *Path* perform the best on each of the aesthetics considered in our study, we used the studies in (Garg & Rusu 2003) and (Rusu, Clement, & Jianu 2005), respectively.

Since, for any tree, when the desirable aspect ratio is set to 1, we can always find the actual aspect ratio of the drawing produced by *Separation* close to 1 (Garg & Rusu 2003), we decided not to evaluate the performance of *Separation* for this quality measure.

Since both *Path* and *Rings* produce *orthogonal drawings*, the angles between the edges connecting the nodes will always be either 90° or 180° . For this reason, we do not consider *Path* and *Rings* in our analysis of quality measure angular resolution.

Very interestingly, the performances for all algorithms on the real-life molecular combinatory binary trees resemble very closely those on unbalanced binary trees. Hence, we make many references to their similarities.

In the case of unbalanced and molecular combinatory binary trees, *Rings* quickly becomes prohibitive to use. For this reason, we have decided to not consider *Rings* in our comparisons for unbalanced and molecular combinatory binary trees.

Due to limited space, we only provide a textual analysis of the algorithms under evaluation. For the same reason, the analysis on unbalanced binary trees is not included in this paper. The complete display of performances for all categories of binary trees (which includes 84 charts), as well as the analysis on unbalanced binary trees and the performance of the algorithms on several other aesthetics (such as **Total Edge Length, Average Edge Length, Maximum Edge Length, Size, Closest Leaf, and Minimum Angle Size**), is provided in a technical report (Rusu, Jianu, Santiago, & Clement 2005).

The analysis of the performance of the four algorithms is summarized below:

Area:

- *Complete trees*: Order of performance: *Rings, Separation, Path, Level*. While the difference in the areas produced by *Rings* and *Separation* grows slowly, the difference in the areas produced by *Separation* and *Path* grows much faster. The same behavior is exhibited by *Level* and *Path*. For $n = 65,535$, *Level* produces a drawing having an area almost four times more than the drawing produced by *Path*.
- *Randomly-generated binary trees*: Order of performance: *Separation, Path, Level, Rings*. The performances of all the algorithms are worse than their respective performances on complete trees. In comparison to its behavior on complete trees, where it was the best, *Rings* exhibits the most dramatic change: its behavior is now the worst of all four algorithms. The area produced by *Level* grows rapidly in comparison to the area produced by *Path*, being already three times more for $n = 50,000$. *Rings* and *Separation* exhibit similar behavior, with *Rings* being slightly better. The differences in the areas produced grow slowly.
- *Fibonacci trees*: Order of performance: *Separation, Path, Level, Rings*. *Rings* quickly becomes prohibitive, with area ten times more than the area of *Separation*, for 10,000 nodes. The difference between the areas produced by *Separation* and *Path* grows slowly, while the difference between the areas produced by *Path* and *Level* grows much faster.
- *Molecular combinatory binary trees*: Order of performance: *Path, Separation, Level, Rings*. Even though *Path* is the best performing algorithm on both unbalanced and molecular combinatory binary trees, its behavior on molecular combinatory binary trees is much better: for $n = 50,000$, the area of molecular combinatory binary trees is 78,360, as opposed to unbalanced-to-the-left binary trees, with an area of 258,355. *Level* rapidly becomes prohibitive to use, producing an area over 1,000,000 for $n = 5,989$. This is the best case for *Path* and the worst case for *Separation*.

Uniform Edge Length:

- *Complete trees*: Order of performance: *Rings, Separation, Path, Level*. *Rings* and *Separation* produce very low, almost constant values. *Path* exhibits a non-linear growing behavior. *Level* quickly becomes prohibitive to use.
- *Randomly-generated binary trees*: Order of performance: *Rings, Separation, Path, Level*. The performances of *Rings* and *Separation* are very similar to those on complete binary trees. *Path* grows slowly and linearly. *Level* quickly becomes prohibitive to use.
- *AVL trees*: Order of performance: *Rings, Separation, Path, Level*. The performances of *Rings* and *Separation* are very good: they almost mimic their performances on complete binary trees. The results for *Path* do not have a consistent rate of change and are about 50 times greater than *Rings*.
- *Fibonacci trees*: Order of performance: *Separation, Path, Rings, Level*. *Separation* outperforms all other algorithms, by maintaining a nearly constant value. The difference between the performance of *Path* and *Rings* stays at an approximately constant 30 as n increases.
- *Molecular combinatory binary trees*: Order of Performance: *Separation, Path, Level, Rings*. *Separation* has a slowly growing behavior, similar to unbalanced-to-the-left binary trees. Interestingly, *Level* performs very poorly, which is different from unbalanced binary trees. *Path* quickly becomes prohibitive with $n = 9,973$ resulting in a uniform edge length of 60.

Farthest Leaf:

- *Complete trees*: Order of performance: *Rings, Separation, Path, Level*. While the performances of *Path, Rings* and *Separation* are very good, with a very slow growth rate, the performance of *Level* is unsatisfactory. For example, for $n = 8,191$, farthest leaf for *Rings* is 89.1, for *Separation* is 219.1, for *Path* is 355.7, and for *Level* is 4,095. For $n = 65,535$, farthest leaf for *Rings* is 284.8, for *Separation* is 626, for *Path* is 811.3, and for *Level* is 32,767.
- *Randomly-generated binary trees*: Order of performance: *Separation, Rings, Path, Level*. Surprisingly, both *Separation* and *Rings* perform just slightly worse on randomly-generated binary trees compared to complete trees. Performances of *Path* and *Rings* are almost identical. Again, the distance to the farthest leaf grows much faster for *Level* than for *Path, Separation* and *Rings*.
- *AVL trees*: Order of performance: *Rings, Separation, Path, Level*. The performances of the algorithms on this measure are almost identical to their respective performances on complete trees.
- *Fibonacci trees*: Order of performance: *Separation, Path, Rings, Level*. For *Separation* and *Path* the same pattern remains. On the other hand, *Rings* still remains better than *Level*, but the rate of growth exhibited by *Rings* has increased. *Level* exhibits the same unsatisfactory behavior.
- *Molecular combinatory binary trees*: Order of performance: *Separation, Path, Level, Rings*. As the case in all categories of binary trees, *Level* rapidly becomes prohibitive. *Separation* produces satisfactory results, placing its farthest leaf

at a distance of 764.6 for $n = 50,005$. The behavior of *Path* is approximately three times greater than *Separation*.

Aspect Ratio:

- *Complete trees*: Order of performance: *Separation*, *Path*, *Rings*, *Level*. Quite interestingly, the behaviors of *Path* and *Rings* are very similar. Neither algorithm always produces drawings with aspect ratios close to optimal. For example, if $n = 2^{14} - 1$, the best aspect ratio *Path* produces is around 0.5. *Rings* produces optimal aspect ratios when $n = 2^i - 1$, with i an odd number, and aspect ratios close to 0.5, with i an even number. The aspect ratios of the drawings produced by *Level* are very low (the highest value is close to 0.06), decreasing rapidly as n increases.
- *Randomly-generated binary trees*: Order of performance: *Separation*, *Path*, *Rings*, *Level*. *Level* produces drawings with better aspect ratios for trees with a smaller number of nodes (the highest value is close to 0.1). Still, its behavior is unsatisfactory, as the value of aspect ratio decreases rapidly as n increases. *Path* and *Rings* have uneven behaviors. Most of their aspect ratios are over 0.8, and none is under 0.5.
- *AVL trees*: Order of performance: *Separation*, *Rings*, *Path*, *Level*. *Rings* exhibits a very interesting pattern: its aspect ratios are either 0.5 or optimal. The performances of *Path* and *Level* decrease dramatically, with *Level* quickly producing very small aspect ratios, and *Path* producing aspect ratios less than 0.01 for 50,000 nodes.
- *Fibonacci trees*: Order of performance: *Separation*, *Rings*, *Path*, *Level*. Interestingly, *Rings* exhibits exactly the same behavior as in the case of complete binary trees: alternating optimal aspect ratios with aspect ratios close to 0.5. The behavior of *Level* is only significant for trees with a small number of nodes.
- *Molecular combinatory binary trees*: Order of performance: *Separation*, *Path*, *Level*, *Rings*. *Path* produces close to optimal values until $n = 5,989$. After this point, the values plummet, decreasing to 0.1 for $n = 50,005$. Very interestingly, *Level* always produces values close to 0.1. In our analysis, it was discovered that *Level* always produces drawings of width equal to n . Hence, for molecular combinatory binary trees, the height of the drawings produced by *Level* is almost always one-tenth of the width.

Angular Resolution: Order of performance: *Path*, *Rings*, *Separation*, *Level*.

Path and *Rings* always produce orthogonal drawings, hence their angular resolution is always 90° . *Separation* produces orthogonal drawings for complete, AVL, and Fibonacci trees. *Level* produces unsatisfactory results, with angular resolution less than 10° for all types of trees. *Separation* exhibits good behavior on randomly-generated and molecular combinatory binary trees with small number of nodes. For unbalanced and molecular combinatory binary trees with large number of nodes, *Separation* produces unsatisfactory results.

5 Conclusion

The performance of a drawing algorithm on a tree-type is not a good predictor of the performance of the same algorithm on other tree-types: some of the

algorithms perform best on one tree-type, and worst on other tree-types. Overall, if user-controlled aspect ratio is important, *Separation* is the algorithm which achieves best results in the majority of the aesthetics. If user-controlled aspect ratio is important, *Rings* should be the choice for AVL and complete trees, and *Path* should be the choice for unbalanced-to-the-right and molecular combinatory binary trees. If visualization of leaf nodes is important, then *Separation* should be the choice for randomly-generated, Fibonacci, and molecular combinatory binary trees, *Rings* should be the choice for complete and AVL trees, and *Path* should be the choice for unbalanced binary trees. If angular resolution is important, *Rings*, *Path*, and *Separation* are all good choices, only *Separation* is not recommended for unbalanced binary trees. *Level*, the algorithm used very often in practice, scores worse in comparison to the other algorithms for almost all tree-types and aesthetics considered in our study.

Acknowledgment

We are grateful to Dr. Bruce MacLennan from the Department of Computer Science at the University of Tennessee for providing to us the real-life molecular combinatory binary trees.

References

- T. Chan, M. Goodrich, S. Rao Kosaraju, & R. Tamassia. (2002), Optimizing area and aspect ratio in straight-line orthogonal tree drawings, in *Computational Geometry: Theory and Applications*, 23:153-162.
- G. Di Battista, P. Eades, R. Tamassia, & I. G. Tollis. (1999), *Graph Drawing*, Prentice Hall, Upper Saddle River, NJ.
- A. Garg & A. Rusu. (2003), A more practical algorithm for drawing binary trees in linear area with arbitrary aspect ratio, in *Proceedings 11th International Symposium on Graph Drawing*, volume 2912 of *Lecture Notes Comput. Sci.*, pages 159-165, Springer.
- B. MacLennan. (2003), Molecular Combinatory Computing for Nanostructure Synthesis and Control, in *Proceedings 3rd IEEE Conference on Nanotechnology*, IEEE Press, pp. 179-182 vol. 2.
- E. Reingold & J. Tilford. (1981), Tidier drawings of trees, in *IEEE Transactions on Software Engineering*, 7(2):223-228.
- A. Rusu, R. Jianu, C. Santiago, & C. Clement. (2005), Planar Straight-Line Grid Drawings of Binary Trees: An Experimental Study, *Technical Report 2005-01*, Department of Computer Science, Rowan University, Glassboro, NJ.
- A. Rusu, C. Clement, & R. Jianu. (2005), Performance Analysis of a Path-Based Algorithm for Drawing Binary Trees, in *Proc. 5th International Conference on Artificial Intelligence and Digital Communications*, Research Notes in Computer Science, pages 84-102.
- S. T. Teoh, K. L. Ma. (2002), RINGS: A Technique for Visualizing Large Hierarchies, in *Proceedings 10th International Symposium on Graph Drawing*, volume 2528 of *Lecture Notes Comput. Sci.*, pages 268-275, Springer.
- L. Valiant. (1981), Universality considerations in VLSI circuits, in *IEEE Trans. Comput.*, C-30(2):135-140.

Method for Drawing Intersecting Clustered Graphs and Its Application to Web Ontology Language

Hiroki Omote and Kozo Sugiyama

Japan Advanced Institute of Science and Technology
1-1, Asahidai, Nomi, Ishikawa 923-1292 Japan

{h_omote, sugi}@jaist.ac.jp

Abstract

We introduce a class of graphs called an *intersecting clustered graph* that is a clustered graph with intersections among clusters. We propose a novel method for nicely drawing the graph based on the *electric spring model* that is force-directed approach. We evaluate the performance of this method through experiments. Then we apply the proposed method to the area of Web Ontology Language.

Keywords: Graph Drawing, Force-directed, Intersecting Clustered Graph, OWL

1 Introduction

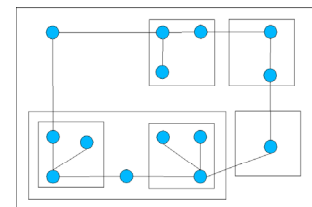
The classical methods for graph drawing so far has been investigated to nicely draw rather simple classes of graphs such as a tree, planar graph, directed graph, undirected graph etc. Recently researches in this area tend to draw more complicated graphs such as *compound graphs* (Sugiyama and Misue 1989) and *clustered graphs* (Eades, Feng and Lin 1996) that consist of inclusions as well as adjacencies among vertices. These graphs can express more complicated information and knowledge structures.

In this paper, we develop a method to draw graphs (called *intersecting clustered graphs*) where intersections among clusters are allowed while it is inhibited in clustered graphs. Figures 1 (a) and 1 (b) show examples of a clustered graph and an intersecting clustered graph, respectively.

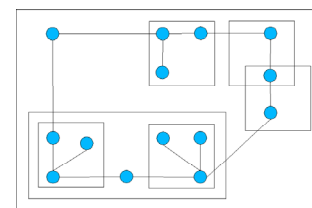
There exist few researches to draw intersecting clustered graphs because of difficulties of the problem: forming intersection areas between clusters and placing shared vertices within the areas, etc.

Our method is based on simulation of a virtual physical system, which is a well known technique called force-directed placement. As one of the early works, Eades (1984) presented a method based upon the *spring model*. We extend the spring model, develop the *electric force model* newly, and combine these two into the *electric spring model*.

Intersections can express elements with some attributes and clusters can express grouping similar elements. Intersecting clustered graphs can be used as tools to foster human thought in a variety of fields. Examples include the *KJ method* (Kawakita 1986) in creativity science, the *Higraph* (Harel 1988) in software engineering, and the *conceptual graph* (Sowa 1984) in knowledge engineering.



(a) Clustered graph



(b) Intersecting clustered graph

Figure 1: Examples of both graphs

In the subsequent sections, we describe a drawing algorithm that can express intersections by using the combination of modified spring embedder and virtual electric forces. Our drawing objects consist of both inclusions and adjacencies among clusters and vertices. Inclusions are drawn like a Venn diagram in set theory. We make experiments to evaluate the performance of our method. Furthermore, we apply our method to Web Ontology Language (OWL).

2 Graph Drawing Method

2.1 Intersecting Clustered Graph

According to Eades, Feng and Lin (1996), a clustered graph $C=(G, T)$ consists of an undirected graph G and a rooted tree T such that the leaves of T are exactly the vertices of G . Each node v of T represents a cluster $\Gamma(v)$ of the vertices of G that are leaves of the sub tree rooted at v . Note that tree T describes an inclusion relation between clusters.

We introduce a new class of graphs. An *intersecting clustered graph* $C_I = (G, D_I)$ consists of an undirected graph G and a digraph D_I that is an extension of a *rooted tree* T . D_I is a rooted tree except to allow sharing of leaves among clusters. Figures 2(a) and 2(b) show examples of T and D_I , respectively.

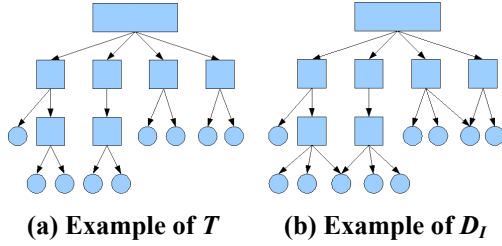


Figure 2: Structure of T_I and D_I

2.2 Aesthetics

Aesthetics consists of drawing conventions and rules. Drawing conventions are conditions that should be satisfied and drawing rules are those that are satisfied as much as possible. We adopt the following drawing conventions (listed in the order of priority):

- 1) A vertex of G is drawn as an ellipse or a rectangle.
- 2) An edge of G is drawn as a straight line.
- 3) A *cluster* is drawn as an ellipse or a rectangle.
- 4) An intersection is drawn as of an area overlapped (*intersection area*).

We adopt the following drawing rules (listed in the order of priority):

- 1) Vertices belonging to a cluster are drawn within the area of the cluster.
- 2) Vertices belonging to more than one cluster are drawn within an intersection area.
- 3) Vertices not belonging to any cluster are drawn outside of cluster areas.
- 4) Overlapping of clusters is reduced as much as possible.
- 5) Neighbours are drawn as close as possible.
- 6) Overlapping of vertices is reduced as much as possible.
- 7) Crossings of edges are reduced as much as possible.

2.3 Models

Our algorithm basically depends on spring forces and electric forces. The spring embedder uses one elastic force and one electric force. However, we use three types of spring forces and two types of electric forces because we need to control positions of clusters and vertices according to their conditions accurately and separately. In the case of spring forces, there exists the concept of an ideal (given) distance between vertices and clusters. The forces control the distances to be close to the *ideal distance*. But in the case of electric forces, there does not exist the concept of the ideal distance. In our method, both forces are utilized properly considering these characteristics.

2.3.1 Extended spring model

The first type of spring force (Type A) is used for controlling distances between neighbours. For this purpose, the following four types of springs with different standard lengths are used according to conditions (see Figure 3):

- Spring 1: springs between adjacent vertices within the same cluster. The length of springs is adjusted to the size of the cluster area.
- Spring 2: springs between adjacent vertices included in different clusters. The length of springs is specified longer than springs of Spring 1.
- Spring 3: springs between adjacent vertices included in different intersecting clusters. The length of springs is adjusted between the lengths of Spring 1 and Spring 2.
- Spring 4: springs between adjacent vertices in the same intersection. The length of springs is specified as the shortest length.

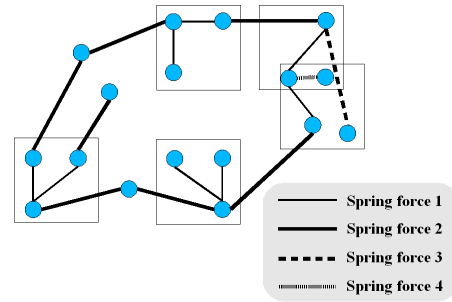


Figure 3: Springs of Type A

The second type of spring force (Type B) makes vertices included within the cluster area. For this purpose we introduce a dummy vertex in the centre of a cluster that connects to each vertex included in the cluster (see Figure 4). The length of springs between the dummy vertex and included vertices is specified according to the size of the cluster area.

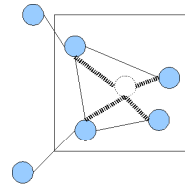


Figure 4: Springs of Type B

The third type of spring forces (Type C) is exerted between clusters that intersect each other (see Figure 5).

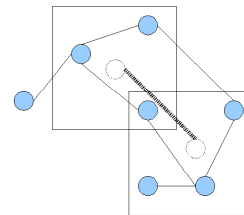


Figure 5: Springs of Type C

2.3.2 Electric model

We also use an electric model. Coulomb forces are induced by electric charges of vertices. When charges are expressed as q and q' and a distance between them as r , an empirical equation called the Coulomb's law is:

$$f = k \frac{qq'}{r^2} \quad (1)$$

where k is a constant and q and q' can be positive and negative. In our model, this force does not need to reflect the physical reality, rather can be used as a virtual physical law. This type of forces is used to place shared vertices within an intersection area (Type D) and to avoid the overlapping among clusters and vertices (Type E). In our model, four types of electric forces are used:

- Electric force 1 is attractive and is exerted between dummy vertices placed on clusters that intersect each other and vertices that should be placed within the intersection area. This force is effective to place vertices within an intersection area.
- Electric force 2 is repulsive and is exerted between every pair of non-adjacent vertices.
- Electric force 3 is repulsive and is exerted between clusters to avoid overlapping of them.
- Electric force 4 is repulsive and is exerted between clusters and free vertices (vertices that are not included in any cluster).

Figure 6 shows four types of electric forces.

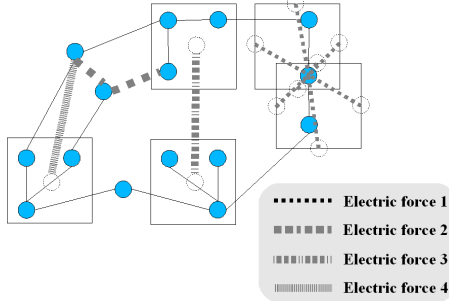


Figure 6: Four types of electric forces

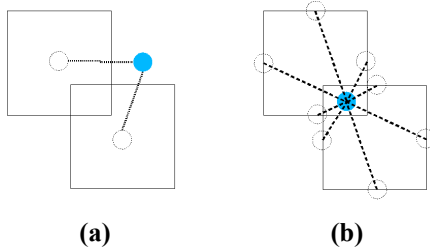


Figure 7: Placement of a shared vertex

Figure 7 shows the effectiveness of Electric force 1. When we try to place shared vertices within an intersection area by spring forces, it is difficult to do so because ideal distances prevent for the vertices to move within the intersection area (see Figure 7(a)). On the contrary, by Electric force 1, it is easy for the vertices to move within the area (see Figure 7(b)).

2.3.3 Electric spring model

The electric spring model is formed combining the spring model and the electric model.

2.4 Algorithm

We consider five types of forces A to E. Let objects be vertices or clusters (or their mixtures). When locations of objects u and v are denoted by $p_u=(x_u, y_u)$ and $p_v=(x_v, y_v)$, and the distance between the objects $d(p_u, p_v)$, force f exerted on objects u and v are given by

- 1) For spring forces,

$$f(u) = k_\alpha \frac{d(p_u, p_v) - l_\alpha}{d(p_u, p_v)} = -f(v) \quad (2)$$

where k_α is a constant and l_α is the ideal distance between objects u and v , and α specifies a force type.

- 2) For electric forces,

$$f(u) = k_\beta \frac{q_u q_v}{d(p_u, p_v)} = -f(v) \quad (3)$$

where q_u, q_v are constants, and β specifies a force type.

[Electric Spring Algorithm]

Input: C_I : Intersecting Clustered Graph, S : Specifications of forces

Output: Drawing of C_I

Method:

- 1) Place vertices and clusters of C_I in random placement;
- 2) Repeat n times;
 - 2-1) Calculate the force exerted on each vertex and cluster area by composing all types of forces;
 - 2-2) Move each vertex and cluster by $\delta \times$ (the force on the vertex and cluster);
- 3) Draw the graph on a screen;

2.5 Experiment

We have performed preliminary experiment to evaluate the performance of our method. In the experiment we have randomly generated twenty sample intersecting clustered graphs $C_I=(G, D_I)$, all of which are connected. We have adopted three criteria for evaluation. Results of the evaluation is summarized in Table 1. In each criterion, error rate is satisfactorily low. And the method can completely form an intersection area.

| Criteria | Graph Type | |
|-----------------------------------|------------|------------|
| | Tree | Undirected |
| Error placement [%] | 3.9 | 8.3 |
| Error overlapping of clusters [%] | 0.9 | 4.1 |
| Error overlapping of vertices [%] | 4.5 | 6.2 |

Table1: Result of evaluation experiment

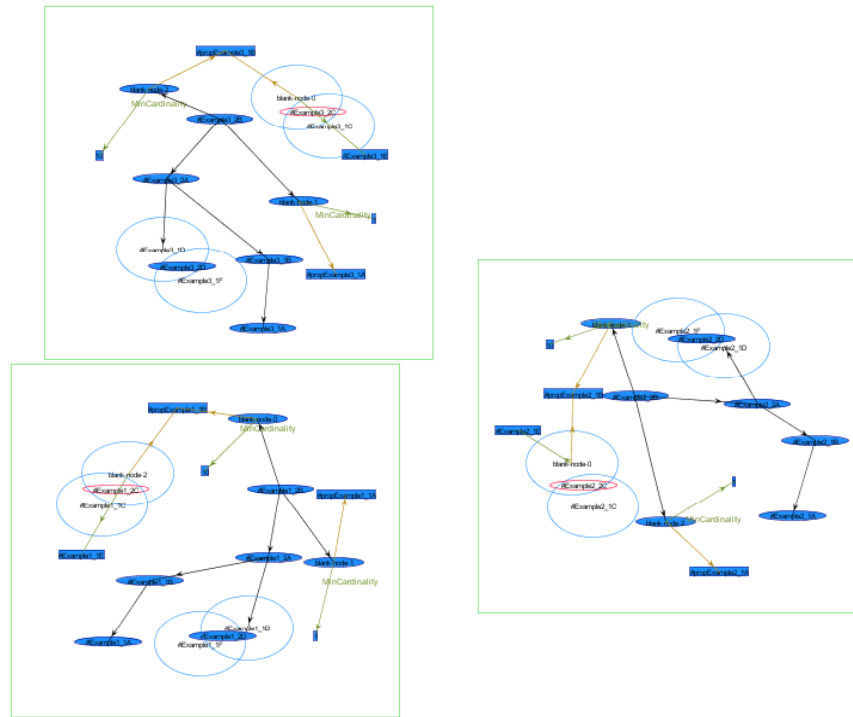


Figure 8: OWL example

3 Application to Web Ontology Language

3.1 How to apply to OWL

The Web Ontology Language is designed for use by applications that need to process the content of information instead of just presenting information to humans (OWL W3C 2005). We have redefined our graph drawing method to conform to the OWL reference. The definition of an intersecting clustered graph has no directed edge but we have improved for the OWL to have a directed edge. A directed edge represents a predicate on OWL. A predicate is a relation among classes and properties. For instance, predicate “subClassOf” has a relation of parent-child, which is represented by an arrow from parent class to child class. A predicate “hasValue” is represented by an arrow from a class to a property. And a predicate “intersectionOf” and “unionOf” also indicates a relation on OWL but these are represented in different ways. A predicate “intersectionOf” that is a conjunction (AND) of classes forms an intersection area. A predicate “unionOf” that is a disjunction (OR). Classes “intersectionOf” and “unionOf” are distinguished by colour. We have developed a modified tool satisfying conventions and rules. The OWL parses by using Jena. Jena is a Java framework for building Semantic Web applications (Jena 2005).

3.2 OWL Drawing Example

Figure 8 shows OWL drawing example. The example has been made out from three OWL files. Each OWL file draws a rectangle.

4 Conclusions

We have defined an Intersecting Clustered Graph and developed a method for drawing it. In developing the

method we have proposed a new model called the electric spring mode. Moreover, we have applied our method to the OWL. In the future, we would like to explore two approaches in expanding our system. The first approach is to improve functions for searching on ontology. We will make it possible to compare various searching results from ontology. And we would like to apply our method to more real world data. The second approach is to improve our visualization. We will make it possible to handle 3D drawing. Therefore we think to be able to use the extended Venn diagram that contains n sets. Finally we will develop tools that can visualize complicated knowledge structure expression for easy use for everyone.

5 References

- Eades, P. (1984): A Heuristic for Graph Drawing, *CONGRESSUS NUMERANTUM* 42, pp.149-160
- Eades, P., Feng, Q. and Lin, X. (1996): Straight-Line Drawing Algorithms for Hierarchical Graphs and Clustered Graphs, LNCS 1027 GD'95 Graph Drawing pp.113-128, Springer-Verlag
- Harel, D. (1972): On visual formalism. *Comm. ACM* 31(5): 514-530.
- Jena: Jena2 A Semantic Web Framework, <http://www.hpl.hp.com/semweb/jena.htm>. Accessed 28 Aug. 2005.
- Kawakita, J. (1986): *The KJ method*. Tokyo, Chuo-koron-sha. (in Japanese)
- Sowa, J. F. (1984): *Conceptual structures*. Addison-Wesley.
- OWL W3C: OWL Web Ontology Language Reference W3C Recommendation 10 February 2004, <http://www.w3.org/TR/owl-ref/>. Accessed 28 Aug. 2005.

BiblioViz: A System for Visualizing Bibliography Information

Zeqian Shen¹, Michael Ogawa¹, Soon Tee Teoh², and Kwan-Liu Ma¹

¹Email: {zqshen, msogawa, klma}@ucdavis.edu,
Computer Science Department, University of California, Davis

²Email: teoh@cs.sjsu.edu
Department of Computer Science, San Jose State University

Abstract

The InfoVis 2004 contest led to the development of several bibliography visualization systems. Even though each of these systems offers some unique views of the bibliography data, there is no single best system offering all the desired views. We have thus studied how to consolidate the desirable functionalities of these systems into a cohesive design. We have also designed a few novel visualization methods. This paper presents our findings and creation: BiblioViz, a bibliography visualization system that gives the maximum number of views of the data using a minimum number of visualization constructs in a unified fashion.

Keywords: bibliography visualization, linked views, graph visualization, text and document visualization, applications of visualization, tables

1 Introduction

In the academic world, the citation databases in each field are very valuable resources to researchers. While text-based searching and browsing is used extensively to look up information in the databases, complex inter-relationships among papers, authors, research areas, and publication venues are best revealed with visual means. The InfoVis 2004 Contest (Plaisant, Fekete, Grinstein 2004) was dedicated to visualizing the bibliography of the field of information visualization. The contest attracted much attention and received many submissions.

The organizers of the InfoVis 2004 Contest selected four First Place winners and eight Second Place winners. Together, these twelve entries presented different visualization representations and systems, with various tradeoffs, strengths and weaknesses. Several other tools for visualizing bibliography information have also appeared over the years. Although these systems provide good visualizations of different aspects of the data, the user is still left with some unanswered questions:

- What are the similarities and differences, and advantages and disadvantages of each system?
- Which of these systems should I use for my needs?

- Is it possible or feasible to integrate the different methods offered by different systems?

We cannot construct a complete bibliography visualization system by simply putting together all the different visual representations presented by the contest entries. The sheer number of choices would lead to information glut and overwhelm the user. Our objective is to analyze the requirements of bibliography visualization and examine the different solutions presented, then to design and create a usable and comprehensive bibliography visualization system, drawing from the experience provided by these existing systems, to create a "best of all worlds" visualization.

We discuss the following issues involved in our analysis, design and construction of our bibliography visualization system, which we call BiblioViz:

1. the methodology and principles in building a usable and comprehensive bibliography visualization system,
2. the goal of bibliography visualization according to what users may want to discover,
3. the various existing systems and their contributions, and
4. our design, based on the principles we outlined and our analysis and integration of existing systems.

We describe the components of BiblioViz, and how we adapt various existing visual representations and visualization components to fit together. We also incorporate some novel visualization methods. We show that BiblioViz is intuitive despite having many different functionalities. Through a case study, we describe how users can utilize the various features of BiblioViz to create custom visualizations that help glean insights in the data.

2 Design Methodology and Principles

We take a systematic approach in our system design by considering the following factors.

2.1 Goal Driven

We consider the goals of the user so that we can tailor our system to help users draw the insights and knowledge they seek. Amar and Stasko (Amar & Stasko 2004) note that "there remains uncertainty about the ability of current systems to adequately support decision making," and therefore propose "a knowledge task-based framework" for designing information visualizations. We approach our design by first asking what information and relationships

the user may be interested in visualizing, and also what insights and knowledge the user may gain out of visualization. Card et al. (Card, Mackinlay & Shneiderman 1999) mention that “the purpose of visualization is insight, not pictures.” Our goal is to provide the best platform for users to “amplify cognition” (Card et al. 1999), and to help them to draw desired insights. The goals we came up with are listed in Section 3.

2.2 Problem Domain Specification

Our system aims to visualize bibliography information. To that end, BiblioViz expects the following data as input:

1. List of Papers (year, publication venue, title, keywords, research areas, primary research area, authors, abstract)
2. List of Publication Venues (name, year)
3. List of Authors (last name, first name)
4. List of Keywords (name, research area)
5. List of Research Areas (name)

BiblioViz currently does not take as input some aspects of publications, such as the month of publication venue, figures, captions and the textual content of the paper. This information can potentially be included and visualized in future versions of BiblioViz.

2.3 Visualization Minimization

In experiments performed by Klein et al. (Klein, Müller, Reiterer, Eibl 2002), they observed that “switching between completely different visualizations confused the users.” We therefore cannot simply throw together all the different contest visualizations into one system; the sheer number of visualizations would surely overwhelm the user. Instead, we will analyze all the contest entries and distill the visualization metaphors down to the essentials needed for a system with a wide range of visualization capabilities. We will also give the user flexibility in choosing some properties of each visualization, such as the use of shapes, colors, etc., without changing the basic metaphor.

2.4 Analysis of Existing Work

We systematically analyze the existing visualization systems according to the goals we outlined for bibliography visualization. Mackinlay et al. (Mackinlay 1986, Mackinlay & Genesereth 1985) use the criteria of *expressiveness* and *effectiveness* to evaluate visual representations. Roth and Mattis (Roth & Mattis 1990) consider an additional criterion: the user’s goal in using the visualization. We use these standards to evaluate various existing visual representations, to derive a set of necessary features for BiblioViz. In analyzing and evaluating existing work, as well as in determining the visual representations in BiblioViz, we consider whether a visualization expresses a relationship in the data, and how effectively it conveys this information. We also consider how useful this information may be to the user. Certainly, these criteria are subjective in their nature, so we use the best of our judgment, and the reader can verify the examples shown in this paper.

2.5 Linking and Extensibility

Although we minimize the number of different representations, there is still more than one representation, and we allow the user to show multiple views simultaneously. We allow linking (Becker & Cleveland 1987) when there are separate windows and different representations so that the user can have a cohesive view of the data.

Since Bibliography data is complex, there are many different ways to visualize the data. Although we have analyzed current systems and defined a set of relationships the user is interested in, there could potentially be additions to the data schema, or inventions of new visualization methods. Our system therefore needs to be able to accommodate these changes. We consider our system to be composed of separate modules: (1) data processing, (2) visual metaphor, (3) auxiliary graphics, and (4) layout algorithm. Each component has a well-defined interface to the others so that changes can be made to one component without affecting others. We currently support a few different visual representations and a few different layout algorithms. From a control panel, the user can choose which to display and use. In the future, if other visual representations and/or layout algorithms are added, we just need to insert new options. Within each visual representation, options are also provided to display additional graphical objects. Again, the current set of auxiliary graphical objects is not exhaustive, so in the future, new graphical objects may be incorporated and added as options.

3 Goals of Bibliography Visualization

In the InfoVis 2004 Contest, three tasks were proposed:

1. “Characterize the research areas and their evolution,”
2. “Where does a particular author fit within the research areas?” and
3. “What are the relationships between two or more of the authors?”

These questions serve as good guidelines for the goals of bibliography visualization. We will keep these goals in mind during the design and evaluation of the system.

4 Analysis of Existing Work

We analyzed the entries of the InfoVis Contest 2004. Some existing bibliography visualization tools such as Chen and Carr’s work (Chen & Carr 1999) and Noel’s work (Noel, Chu & Raghavan 2003) have heavily influenced some of the contest entries, so these works are not listed separately.

We notice that there are many similarities among the different visualizations presented. We can classify them under four broad categories shown below, listed with the visualizations that fall under each category.

- Table: PaperLens (Lee, Czerwinski, Robertson & Bederson 2004), Keyword Burst Table (Ke, Börner, Viswanath 2004), Time Slicer of INSPIRE (Wong, Hetzler, Posse, Whiting, Havre, Cramer, Shah, singhal, Turner & Thomas 2004), Paper Finder (Keim, Panse, Sips, Scheidewind, Barro 2004), One-For-All (Teoh & Ma 2004)
- Network:

- 2D: Citation and Co-citation Network (Ke et al. 2004), Associative Information Visualizer (White, Lin & Buzydlowski 2004), Paper Finder (Keim et al. 2004), Citespace (Chen 2004)
- 3D: WilmaScope (A. Ahmed, T. Dwyer, C. Murray, L. Song, and Y.X. Wu. 2004), van Ham (van Ham 2004), Tulip (Delest, Munzner, Auber, Domenger 2004)
- Node placement without network:
 - 2D: IN-SPIRE, (Wong et al. 2004) Associative Information Visualizer (White et al. 2004), MonkEllipse (Hsu, Farabaugh, McColgin, Stamper 2004)
 - 3D: InfoVisExplorer (Tyman, Gruetzmacher & Stasko 2004)
- Others: InterRing techniques (Yang, Ward & Rundensteiner 2002) used in Paper Finder (Keim et al. 2004), MonkEllipse (Hsu et al. 2004) (arrange papers chronologically, show paper details), Worms in WilmaScope (Ahmed et al. 2004)

Within each category, we find differences and commonalities.

In the Table category, we have found that all the tables assign time to an axis. This indicates a strong consensus that the Table View is very appropriate and effective for representing time-related information. Furthermore, all Table entries, with the exception of One-For-All, use the x-axis to represent time. In the Table View, some systems draw bar graphs (eg. INSPIRE), some draw line graphs (eg. Ke et al.) and others draw nodes (eg. One-For-All and Keim). Many visual methods incorporate color to show additional information (for example, One-For-All uses color to represent selected authors). Size is commonly used to show the importance of the node.

There are also many variations under the Network category. For example, Ke et al. uses color and size to represent number of citations. Keim also uses color. Xia Lin mixes all the attributes such as authors, keywords and documents, Keim uses a spring embedder to lay out the graph, showing co-author relationships, and CiteSpace superimposes citation bars on the network. Node/link color and size are used by many systems to show additional information. For example, one option in Ke et al. is to use size to represent the number of citations, and color to represent publication year. The different methods also vary in their effectiveness; for example, the large number of overlapping lines in WilmaScope may obscure some information.

Some systems such as MonkEllipse and PaperLens also provide Details on demand, displaying the full title, authors and other information of selected papers.

We find that the information conveyed by systems listed in the “Others” category can often be expressed equally effectively by the Table and Network Views. For example, Keim’s use of InterRing to visualize co-authors, papers and year can also be shown in a network of authors, highlighting the appropriate author node, and using the appropriate color-coding (color-code the co-authors of the highlighted author) and text labeling (label the year of publication).

5 BiblioViz

We utilize our classification and analysis of current visualization tools to create a compact but comprehensive bibliography visualization system: BiblioViz. From our study, we discover that only two views—the

Network View and Table View—are sufficient to cover the functionalities of the existing visual representations. The Network View is better able to visually convey relationships between objects, whereas the Table View is better at showing time-related information. We notice that several of the contest entries submitted pictures of both Table and Network Views, and that all the Table Views use time as one of the axes. Because the Network and Table views are very common visual metaphors, and because of our goal of minimizing visual representations (See Section 2), we select these two views to include in BiblioViz.

In addition, we find that displaying the details of a paper on demand, such as when the user clicks on a visual object, is a convenient and useful feature, following Shneiderman’s (Schneiderman 1996) mantra, “Overview first, zoom and filter, then details on demand.” In the Network View, the user can choose to show only the nodes without the links, and hence offer the same visualization as the methods listed in the “node placement without network” class in Section 4.

We then build additional options and visual features to enable BiblioViz to convey deeper and richer information. We describe how we fit the various visualization pieces so that they work cohesively and seamlessly together, and we also describe some novel techniques we introduce in BiblioViz. The InfoVis 2004 Contest dataset, composed of bibliography data in the field of Information Visualization, is used for the rest of the discussion and examples.

5.1 System Overview

BiblioViz consists of five parts: (1) Table View, (2) Network View, (3) Paper Details Panel, (4) Data Menu, and (5) User Control Panel. The user can click in the Table View display or the Network View display to highlight a particular paper, and the details of the paper are shown in the Paper Details Panel. BiblioViz also allows the user to pick data entities (authors, papers, research areas) from the Data Menu to view or to highlight. The User Control Panel allows the user to make dynamic queries into the data, and also to specify visualization parameters for the Table View and Network View. The Views are linked, so interaction with one has an appropriate visual effect on the other.

5.2 Table View

In the Table View, a 2D table is shown. The x-axis depicts time by year. The user can choose to show (1) publication venues, (2) authors, or (3) research areas on the y-axis. Figure 1 shows an author table, while Figure 2 shows a research area table. Each paper is represented as a rectangle and is placed into a cell in the table according to its attribute values in the x and y axes. Within each cell, the rectangles are stacked horizontally. Additionally, a rectangle can be colored by (1) publication venues, (2) authors, or (3) research areas. If a paper has multiple authors or belongs to multiple research areas, its rectangle is divided into multiple sections and colored accordingly. The height of each rectangle indicates its relative importance (how many times it is cited by other papers).

We use a focus+context method resembling DateLens (Bederson, Clamage, Czerwinski, Robertson 2004) for the table navigation. The user may select individual rows or columns to focus upon (see Figure 1). Those table entries will become larger while the surrounding entries become smaller, but still remain visible to provide the context.

One useful feature provided by the Table View is to allow the user to choose different methods of arranging the rows. For example, if research area is

chosen as the y-axis, the user can choose to sort the research areas in terms of their importance, or to arrange them by their relationship with one another (related research areas placed close together via multidimensional scaling). The user can also highlight a certain paper by clicking on it, or by specifying it from a list. When a paper is selected, its border is highlighted in red, and its details are displayed on the side. Rather than draw lines connecting it with its citations, as in One-For-All, the cited papers are highlighted in a similar way in orange border color. This greatly reduces visual clutter while still allowing the user to visualize relationships.

Another feature provided by the Table View is to incorporate the keyword burst visualization offered by Ke et al. We arrange the periods chronologically as usual, but instead of just coloring periods of keyword burst, we show the actual papers in the periods. This provides the user richer visual information, while preserving the perception of the bursts. An example is shown in Figure 2.

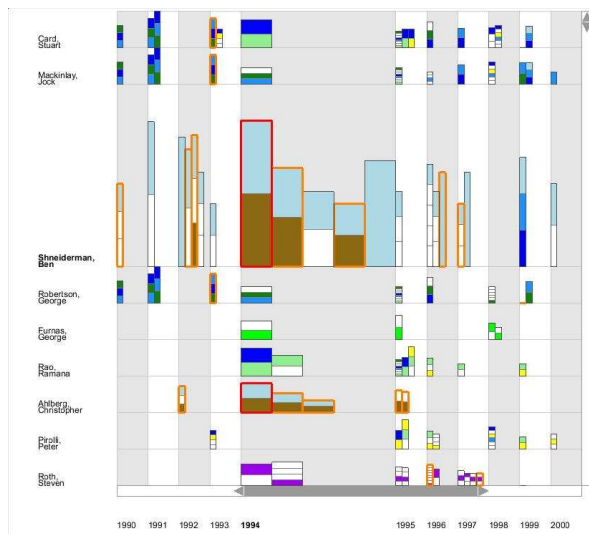


Figure 1: Table of Papers by Year (x-axis) and authors (y-axis). The authors are arranged by their importance. Each rectangle represents a paper. Each segment within a rectangle represents an author of the paper. Selected authors are color-coded.

5.3 Network View

The Network View consists of five components: (1) Filter, (2) Node Placement, (3) Highlight, (4) Rendering, and (5) Auxiliary Graphics.

For the Filter component, the user can choose to draw a network consisting of nodes representing authors, papers, publication venues or research areas. Within each network, the user can choose only to draw nodes representing the top n entities (authors, papers, publication venues or research areas), for example, the user may choose to display the 100 most-cited authors. BiblioViz also provides a menu for the user to select specific entities to display. This menu can be ordered alphabetically or arranged by the importance of the entity.

In the Node Placement component, BiblioViz supports several different layout algorithms. These are self-organizing maps (SOM) (Kohonen 1997), force-directed (Kamada, Kawai 1989), and centroid. SOM is good for arranging the nodes according to research area (see Figure 3). Force-directed is good for reducing edge lengths by placing collaborating authors or paper citation cliques close together (see Figure 4).

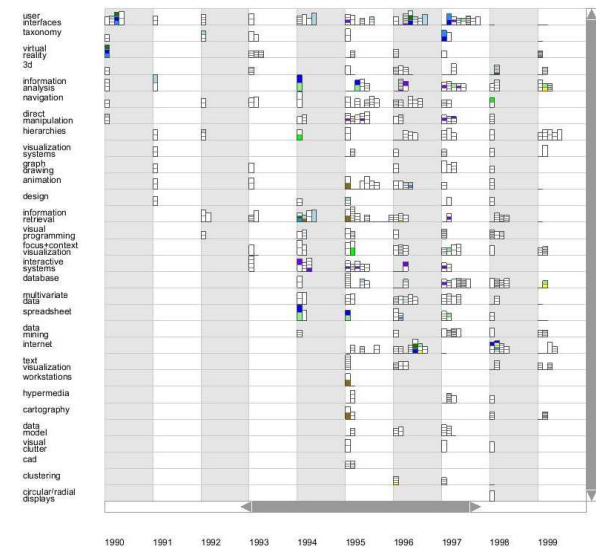


Figure 2: Table of papers by Year (x-axis) and research area (y-axis). The research areas are arranged by their burst in activity.

The centroid layout method is used in the dual-plane rendering mode, discussed below. It places nodes of one type above the centroid of its neighbors of another type.

The Highlight component allows the user to choose certain entities to focus on. This is done by clicking on the node representing the entity (author or paper). The user can also highlight an entity by selecting it from a menu. When an entity is highlighted, it is labeled and its links are drawn with increased thickness. This allows the user to see the highlighted entity clearly, in the context of the global information.

The Rendering component determines how the data is displayed as graphics. We provide two possible views of the network: 2D and 3D. In the Network View, there can be one or more planes. Each plane contains nodes representing one type of entity (author, paper, publication venue, or research area). Planes can be stacked above each other. The 2D view looks at the network plane(s) from the top, so if multiple planes exist, they will be blended into one. The 3D view looks at them from an oblique angle (see Figure 5). Cylinders, spheres, and lighting are used to enhance depth cues in the 3D view. Furthermore, on the same plane, tube arches are used to draw links, rather than straight lines, in order to give a better perception of the end-points of the links (Arcs or curved lines have been used effectively in the past, for example, in Wattenberg's Arc Diagrams (Wattenberg 2002) and InfoVis 2004 Contest entry One-For-All, but 3D arches have not been used in previous bibliography visualization systems.). Colors are mapped to categorical data, and non-highlighted objects are drawn in lighter colors. These graphical methods are used to enhance the *effectiveness* of the visualization. The mapping of color to nominal data, links and proximity for relationships, and size for (quantitative) importance, are all commonly used visual mappings, as they take advantage of the “automatic processing” (Card et al. 1999) ability of the human visual perception system.

The Auxiliary Graphics component helps to convey additional information. We allow the user to add some auxiliary graphical entities to the visualization. For example, the user can choose to show the density map of the SOM nodes similar to INSPIRE, except that in BiblioViz, this can be superimposed on the network display, providing contextual

information. The circular nodes representing papers can also be segmented, so that each segment can be color-coded according to the authors of the paper, or the research areas of the paper, similar to One-For-All. In the 3D Network View, we also allow the user to optionally display a tower on top of each node (similar to the *citation landscape* used in CiteSpace). If the nodes represent authors, each segment in the tower can represent a paper written by the author, and color-coded according to the publication venue.

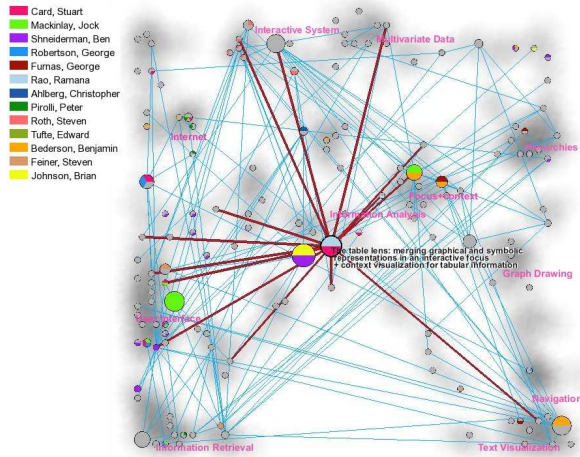


Figure 3: Network View of selected papers. Papers are placed by research areas using SOM.

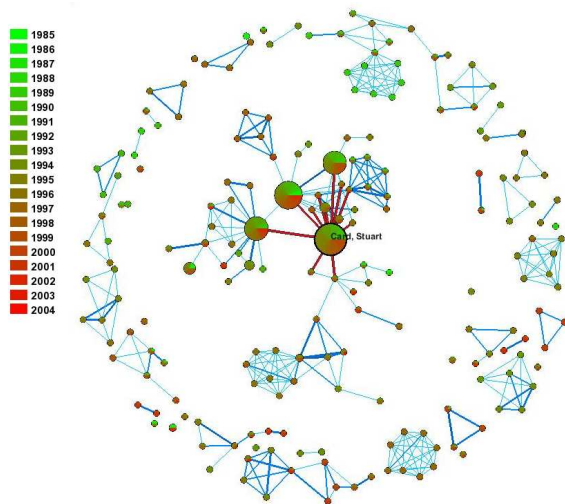


Figure 4: Network View of selected authors. Authors are placed according to their collaborations using force-directed layout methods.

The options in BiblioViz allow the user to balance information quantity against clarity. With the Filtering options, the user can choose how many nodes to show. More nodes show more information, but add to clutter. With the Highlight option, the desired important information can be made more obvious, while keeping a lot of information in the background as context. Similarly, the options to show or to hide auxiliary graphics such as links and towers also have the same trade-offs. With the Filter, Highlight and Auxiliary Graphics options, the user can have a wide variety of choices in making these trade-off decisions according to the knowledge desired.

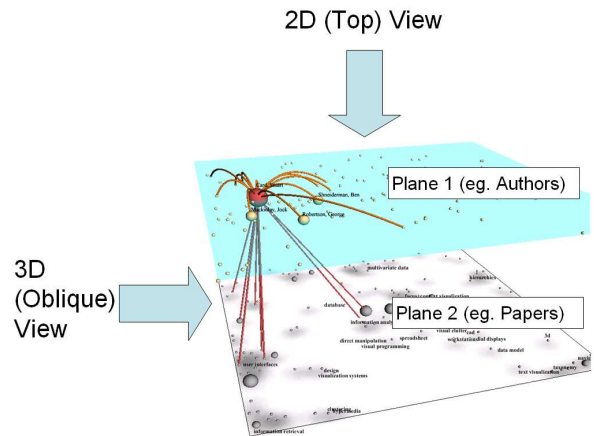


Figure 5: User can choose to view the network from the top (2D view) or from an oblique angle (3D view).

5.4 Linking the Components

The Table View and Network View can be displayed individually or simultaneously, selected by an option in the control panel. In the single view mode, the chosen View occupies the entire center region of the display. In double view mode, the Network View is shown above the Table View (see Figure 6). The two Views are linked, in that interacting with one will have an effect on the other. For example, clicking on a node in the Network View will highlight the node in both Views, as well as display the paper details.

With the combination of these two Views, many visualizations can be generated. For each View, the user can choose to arrange information by author, paper, publication venue, research area, etc. This leads to many possible visualizations per View, and even more visualizations when the two Views are combined.

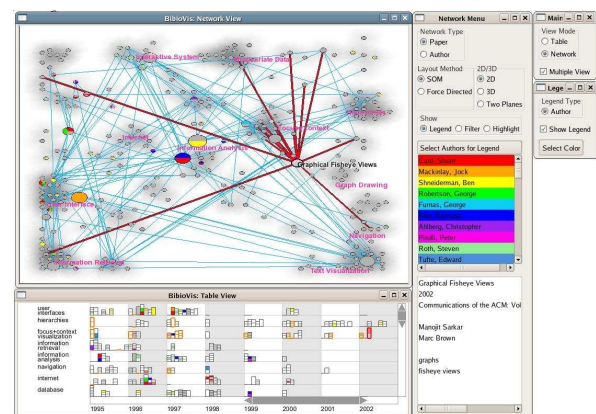


Figure 6: Double view mode layout. The Network View is shown above the Table View. Clicking on a node in one view will highlight the node in both views, as well as display the paper details.

6 Evaluation

In section 3, we listed the tasks of the InfoVis 2004 Contest. We considered these tasks to be the goals of our system. We will now go through and determine if these goals have been met.

Characterize the research areas and their evolution. The Network View is able to show the relationships between research areas well. Using an SOM, the node placement component arranges the nodes

into clusters according to research area. Larger clusters indicate more developed research areas. Clusters which are closer together indicate related areas. The Table View is able to show the evolution of research areas due to its assignment of time on the x-axis. The user can select the keyword burst arrangement and see the progression of research area activity.

Where does a particular author fit within the research areas? For this task, we can use the Network View exclusively. We will display two planes; the bottom will contain nodes representing papers and the top will contain nodes representing authors. The paper nodes will be clustered according to research area by the SOM. Each author node will be placed above the centroid of the papers he or she has published. By clicking on and highlighting a particular author node, the user can easily see which papers he or she has written and in which research areas they belong.

What are the relationships between two or more of the authors? Again, the Network View is the most useful in relationship tasks. We can have a plane containing author nodes, arranged by a force-directed layout. We then select the specific authors we want to investigate either from a list or by clicking a node in the Network View. We can see the other authors who have collaborated with our selection through the edges. We can create another plane containing nodes representing papers, arranged by SOM, placed below the author plane. The papers are clustered according to research area, so now we can see which research areas are common to the authors we have selected.

We have demonstrated how our system can easily accomplish the InfoVis tasks.

7 A Case Study

We now narrate a typical user experience of BiblioViz.

First, the user starts with an overview of the entire dataset in the Table View (Figure 7). The x-axis is year of publication and the y-axis is author. There are too many papers visible, so we use the slider bars to display only the papers of the ten most widely cited authors in the last 20 years (Figure 8). We see that Stuart Card is deemed the most widely cited author in this period, followed by Jock Mackinlay and Ben Shneiderman. Looking at the color of the bars, we see the pattern of dark blue, blue, light blue appearing quite frequently in the works of Card, Mackinlay and Robertson. Indeed, they have collaborated on many papers together, likely due to the fact that they were colleagues at the Xerox Research Center.

We are interested in Ben Shneiderman, so we focus on his row by clicking on his name in the y-axis (Figure 9). The bars in his row become larger so we can more clearly see the collaborations in his papers. We are also interested in the year 1991, so we click on that year in the x-axis. The column corresponding to 1991 similarly becomes wider. We see that Shneiderman published one paper in 1991. We click on it and see from the paper details that it is "Treemaps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures." From the highlighted borders, we can see that almost every other author in the table cites or is cited by this paper.

We then switch to the Network View of authors. The nodes are positioned using a force-directed graph layout algorithm. Since we have already highlighted Ben Shneiderman in the Table View, his node in the Network View is highlighted as well. We can see from the proximity and size of the nodes surrounding Shneiderman, that he has two very important authors closely connected to him. By clicking on the nodes, we see that they are Card and Mackinlay (Figure 10).

We then add another plane to the Network View. It contains paper nodes clustered by SOM into their research areas. The paper plane is placed below the author plane in 3D (Figure 11). Edges extend from Ben Shneiderman's node on the top plane to his papers on the bottom plane. We see that his main research areas are user interfaces, information retrieval and hierarchies. The tower on top of his node shows which publication venues have featured his work. We see from the orange disks that he has published six papers in the Conference on Human Factors in Computing Systems.

From our case study, we have shown that our system can be easily used to explore bibliography information.

8 Future Work

BiblioViz is designed to be extensible. We see several possibilities for extension: adding more (1) data, (2) visual representations, and (3) algorithms. Currently, BiblioViz does not look at the full text and figures/captions of the papers, and these can potentially be added in the future. New visual representations can also be added if they are able to more effectively express otherwise hidden knowledge in the data. Our strategy is to try to embed these additional visual representations into the current Table or Network Views, the way we have embedded citation links, segmented circles, arches and towers. This will reduce users' context-switching and confusion. If this is not possible, we have to add new views apart from the existing Table and Network Views. Other layout algorithms such as Pajek (Batagelj & Mrvar 2003), Galaxies (Wise, Thomas, Pennock, Lantrip, Pottier, Schur & Crow 1995) and BiblioMapper (Song 1998) can also be incorporated as options. This will change the position of the network nodes but will otherwise not alter the look of the current visualization.

The data that we worked with is small relative to other potential bibliography datasets. In order to fully test the robustness of our system, we need a larger dataset, perhaps on the order of several thousands of references.

While we solved the scalability problem in the Table View through the zoomable interface, we still need to address the problem in the Network View. As the number of nodes increases, so does the clutter. We need better methods to draw large graphs.

9 Conclusions

We have introduced a compact, comprehensive and extensible system for visualizing bibliography information. Our design has been driven by a set of goals we defined, and is based on the principle of minimal visual representations, and on our analysis of current visualization techniques.

We first defined a set of goals and inputs for bibliography visualization. We then analyzed and classified current bibliography visualization techniques. From our problem definition and analysis, we designed a system, BiblioViz, combining the features and capabilities of the different available techniques in a seamless, cohesive manner.

We condensed the numerous existing visual representations into only two views: the Table View and the Network View, link them together, and provide details on demand. We found that these two views can effectively express the many relationships contained in bibliography data. We limited the number of different views to prevent users from getting confused by too many different visual metaphors. We also chose the Table and Network Views because they

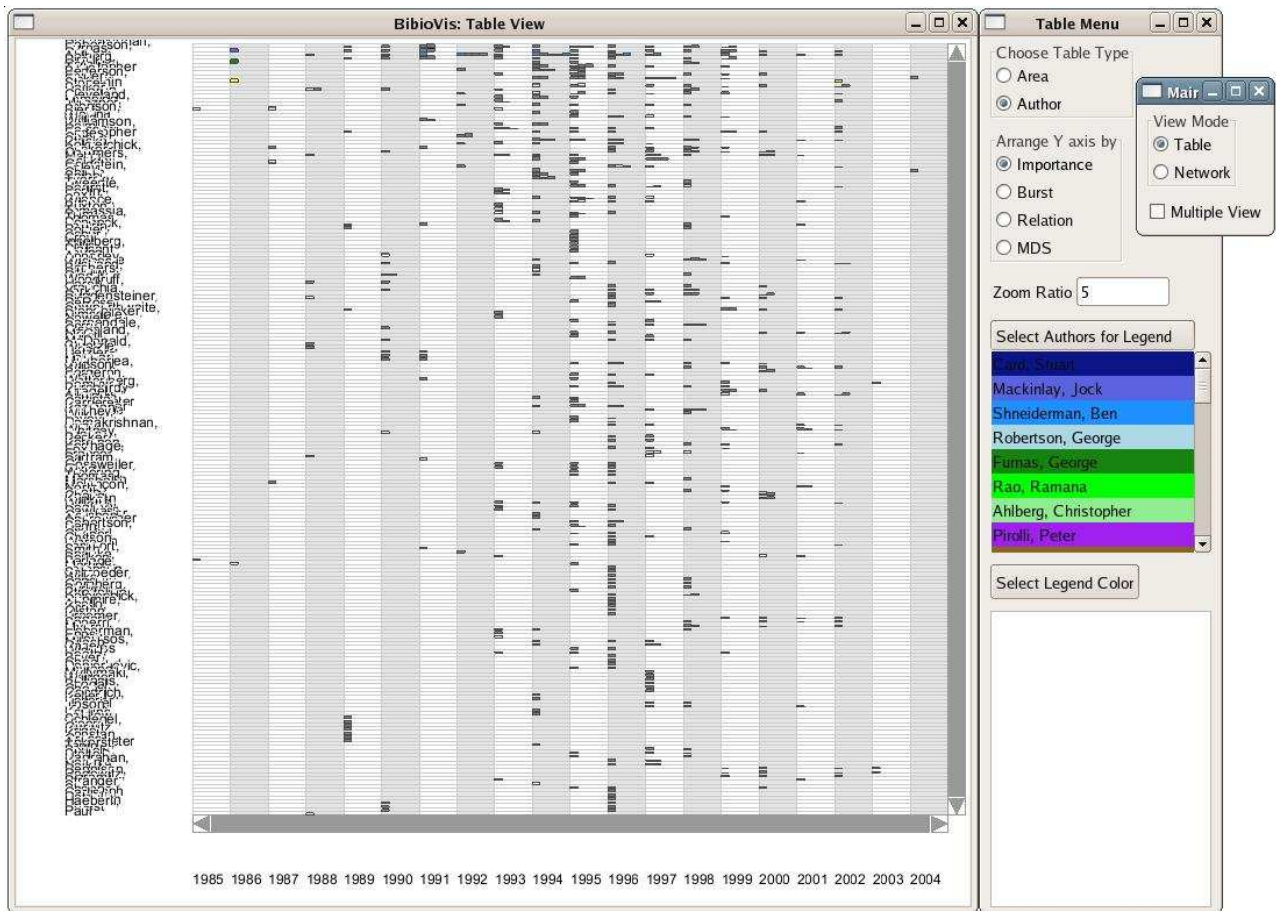


Figure 7: The Table View of authors, showing all authors in the dataset. There are too many authors and papers to discern detailed information, so we allow filtering and focus+context through the user interface.

are commonly used visual metaphors that users are familiar with. Within each View, we provide users with a wider range of options and auxiliary graphics so that users can create different visualizations tailored to explore different parts of the bibliography data schema, while keeping the same view paradigm.

For the Table View, we allow the user to choose the data entity to show on the y-axis, attribute values to display as colors, different methods of arranging the rows, the option to show citation links, and highlight certain data entities. This powerful set of features is not available in any previous package.

For the Network View, we allow 3D or 2D views, providing layout options (force-based, SOM). BibioViz displays nodes representing single or multiple data types (papers, publication venues, authors, research areas) and their inter-relationships, and can super-impose SOM density maps, perform dynamic queries, and highlight individual data entities.

Using various combinations of options, the user has a good control over the type and amount of data to display and what to show in focus and context, and can create custom-made visualizations.

In BibioViz, we introduced novel visual representations, such as multiple parallel network planes, arched links in the 3D Network View, segmented nodes in the Network View, super-imposition of SOM density maps on network displays, and keyword burst visualization with paper-nodes in the Table View. We have demonstrated these functionalities of BibioViz using the InfoVis 2004 Contest data.

Acknowledgment

This work is sponsored in part by NSF PECASE, NSF CCF, and NSF IIS programs.

References

- Ahmed, A., Dwyer, T., Murray, C., Song, L. and Wu, Y. X. (2004), Wilmascope graph visualization, in 'The History of InfoVis (InfoVis 2004 Contest)', IEEE Computer Society, Washington, DC, USA.
- Amar, R. and Stasko, J. (2004), A knowledge task-based framework for design and evaluation of information visualizations, in 'Proceedings of the IEEE Symposium on information Visualization (VIS '04)', Vol. 00, IEEE Computer Society, Washington, DC, USA, pp. 143–150.
- Batagelj, V. and Mrvar, A. (2003), Pajek – Analysis and Visualization of Large Networks, in M. Jnger and P. Mutzel, eds, 'Graph Drawing Software', Springer, Berlin, p. 77–103.
- Becker, R. A. and Cleveland, W. S. (1987), Brushing scatterplots, *Technometrics*, **29**(2), 127–142.
- Bederson, B. B., Clamage, A., Czerwinski, M. P., and Robertson, G. G. (2004), Datelens: A fisheye calendar interface for PDAs. *ACM Transactions on Computer-Human Interaction*, **11**(1), 90–119.
- Card, S. K., Mackinlay, J. D. and Shneiderman, B. (1999), *Readings in information visualization: using vision to think*, Morgan Kaufmann.

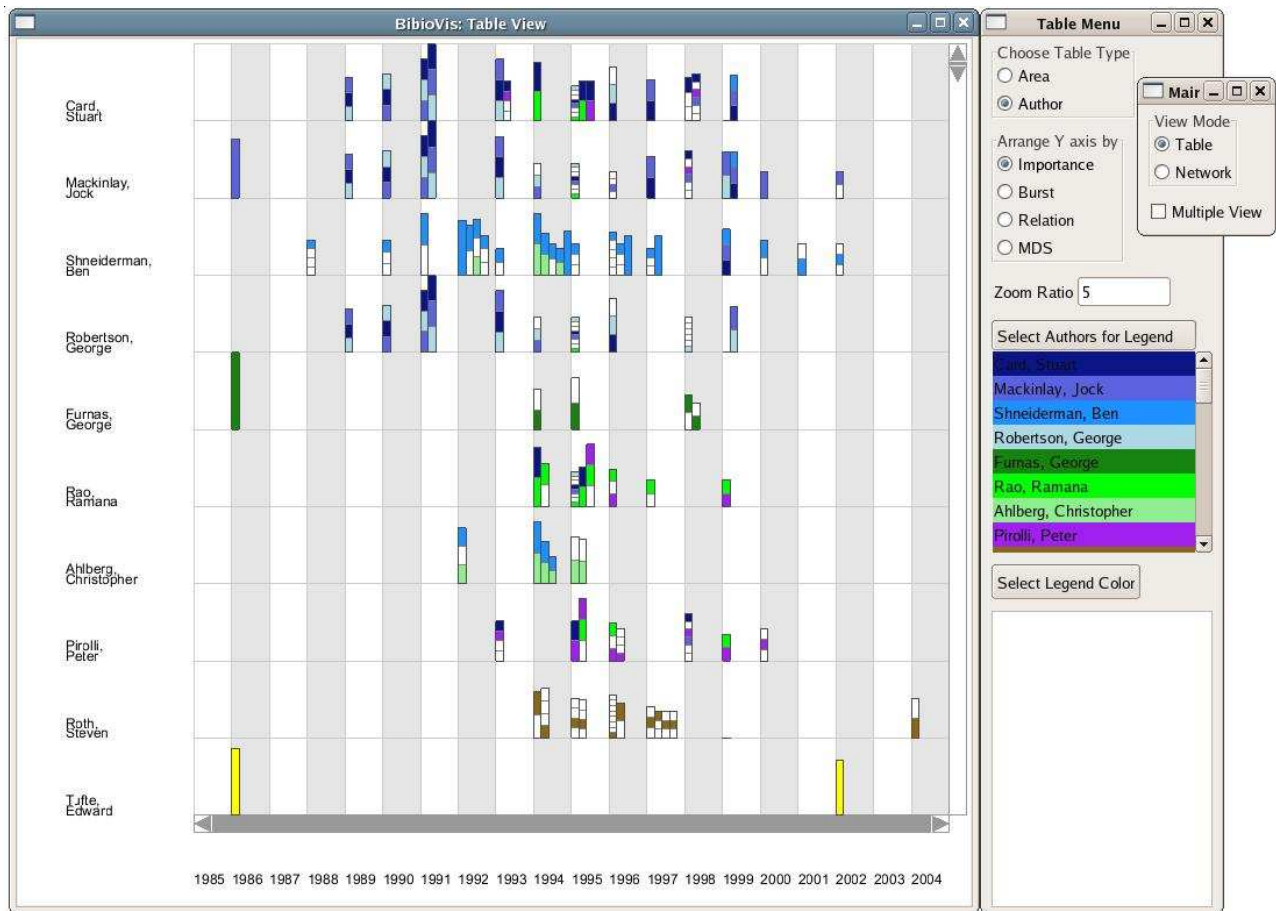


Figure 8: Table View of top 10 most widely cited authors in last 20 years. Stuart Card is deemed the most widely cited author in this period, followed by Jock Mackinlay and Ben Shneiderman.

- Chen, C. (2004), Information visualization research: citation and co-citation highlights, in 'The History of InfoVis (InfoVis 2004 Contest)', IEEE Computer Society, Washington, DC, USA.
- Chen, C. and Carr, L. (1999), Visualizing the evolution of a subject domain: a case study, in 'Proceedings of the Conference on Visualization '99', IEEE Visualization, IEEE Computer Society Press, Los Alamitos, CA, USA, pp. 449–452.
- Delest, M., Munzner, T., Auber, D. and Domenger, J.-P. (2004), Exploring InfoVis publication history with tulip, in 'The History of InfoVis (InfoVis 2004)', IEEE Computer Society, Washington, DC, USA.
- Fekete, J.-D., Grinstein, G. and Plaisant, C., (2004), IEEE InfoVis 2004 contest, The History of InfoVis, Washington, DC, USA.
- Hsu, T.-W., Farabaugh, L. I., McColgin, D. and Stamper, K. (2004), MonkEllipse: Visualizing the history of information visualization, in 'The History of InfoVis (InfoVis 2004 Contest)', IEEE Computer Society, Washington, DC, USA.
- Kamada, T. and Kawai, S. (1989), An algorithm for drawing general undirected graphs, *Information Processing Letters*, **31**(1), 7–15.
- Ke, M., Börner, K. and Viswanath, L. (2004), Major information visualization authors, papers and topics in the ACM library, in 'The History of InfoVis (InfoVis 2004 Contest)', IEEE Computer Society, Washington, DC, USA.
- Keim, D. A., Panse, C., Sips, M., Schneidewind, J. and Barro, H. (2004), Exploring and visualizing the history of infovis, in 'The History of InfoVis (InfoVis 2004 Contest)', IEEE Computer Society, Washington, DC, USA.
- Klein, P., Müller, F., Reiterer, H. and Eibl, M. (2002), Visual information retrieval with the supertable + scatterplot, in 'Proceedings of the 6th International Conference on Information Visualization (IV '02)', pp. 70–75.
- Kohonen, T. (1997), *Self-Organizing Maps*, Springer-Verlag.
- Lee, B., Czerwinski, M., Robertson, G. and Bederson, B. B. (2004), Understanding eight years of infovis conferences using paperlens, in 'The History of InfoVis (InfoVis 2004 Contest)', IEEE Computer Society, Washington, DC, USA.
- Mackinlay, J. (1986), Automating the design of graphical presentations of relational information, *ACM Trans. Graph.*, **5**(2), 110–141.
- Mackinlay, J. and Genesereth, M. R. (1985), Expressiveness and language choice, *Data Knowl. Eng.*, **1**(1), 17–29.
- Noel, S., Chu, C.-H.H. and Raghavan, V. (2003), Co-citation count vs correlation for influence network visualization, *Information Visualization*, **2**(3), 160–170.
- Roth, S. F. and Mattis, J. (1990), Data characterization for intelligent graphics presentation, in

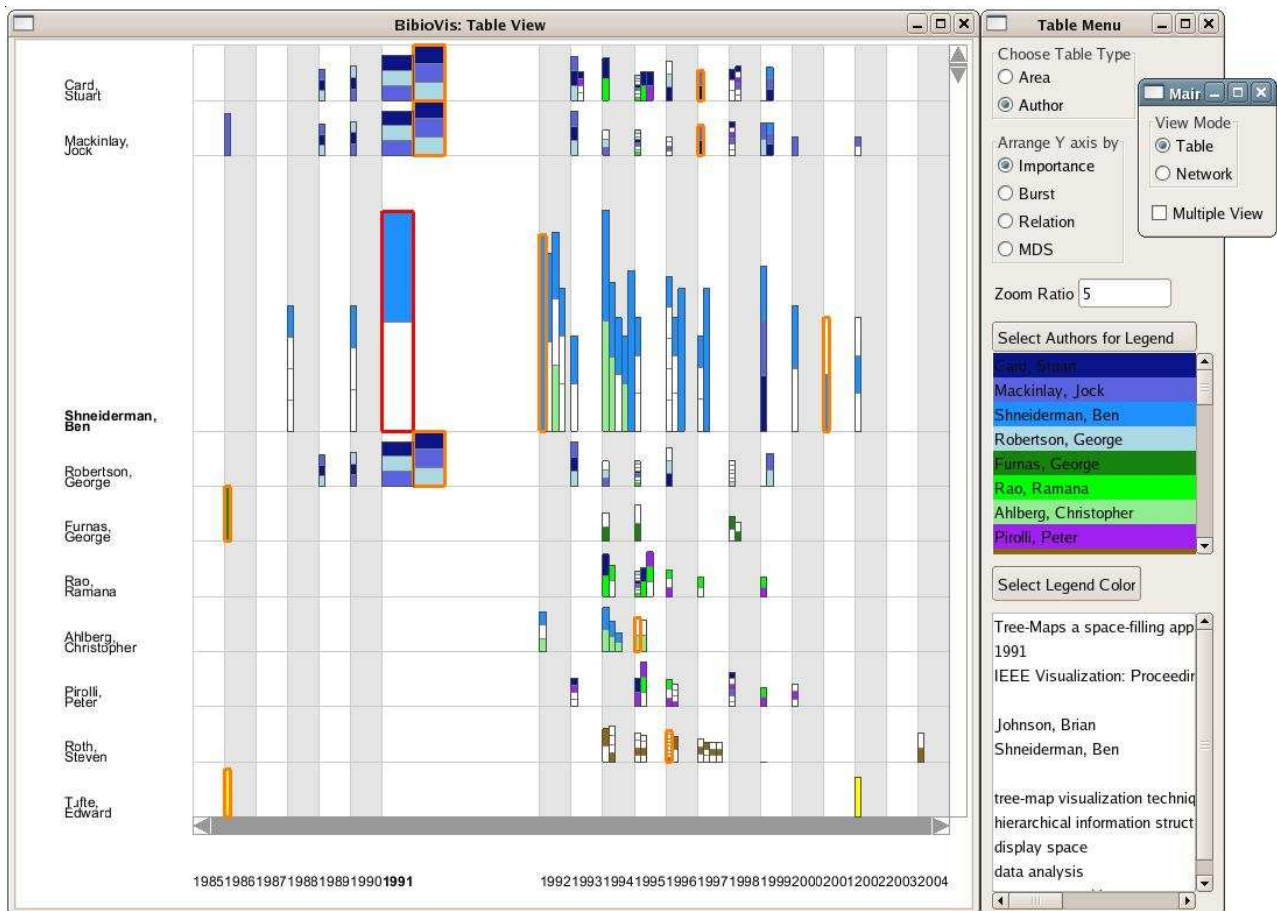


Figure 9: Table View of authors. One of Ben Shneiderman's papers: "Treemaps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures." is highlighted.

'Proceedings of the SIGCHI conference on Human Factors in Computing Systems (CHI '90)', ACM Press, pp. 193–200.

Shneiderman, B. (1996), The eyes have it: a task by data type taxonomy for information visualizations, in 'Proceedings of the 1996 IEEE Symposium on Visual Languages', VL, IEEE Computer Society, Washington, DC, USA, pp. 336–343.

Song, M. (1998), Bibliomapper: a cluster-based information visualization technique, in 'Proceedings of the IEEE Symposium on Information Visualization (Infoviz 98)', IEEE Computer Society, Washington, DC, USA, pp. 130–136.

Teoh, S. T., and Ma, K.-L. (2004), One-for-all: visualization of the information visualization symposia, in 'The History of InfoVis (InfoVis 2004 Contest)', IEEE Computer Society, Washington, DC, USA.

Tyman, J., Gruetzmacher, G. P. and Stasko, J. (2004), InfoVisExplorer, in 'The History of InfoVis (InfoVis 2004 Contest)', IEEE Computer Society, Washington, DC, USA.

van Ham, F. (2004), Case study: visualizing visualization, in 'The History of InfoVis (InfoVis 2004 Contest)', IEEE Computer Society, Washington, DC, USA.

Wattenberg, M. (2002), Arc Diagrams: visualizing structure in strings, in 'Proceedings of the IEEE Symposium on Information Visualization (Infoviz 02)', IEEE Computer Society, Washington, DC, USA, pp. 110–116.

White, H. D., Lin, X., and Buzydlowski, J. (2004), An associative information visualizer, in 'The History of InfoVis (InfoVis 2004 Contest)', IEEE Computer Society, Washington, DC, USA.

Wise, J. A., Thomas, J. J., Pennock, K., Lantrip, D., Pottier, M., Schur, A. and Crow, V. (1995), Visualizing the non-visual: spatial analysis and interaction with information from text documents. in 'Proceedings of the IEEE Symposium on Information Visualization', IEEE Computer Society Press, pp. 51–58.

Wong, P. C., Hetzler, E. G., Posse, C., Whiting, M., Havre, S., Cramer, N., Shah, A. R., Singhal, M., Turner, A. and Thomas, J. (2004), IN-SPIRE infovis 2004 contest entry, in 'The History of InfoVis (InfoVis 2004 Contest)', IEEE Computer Society, Washington, DC, USA.

Yang, J., Ward, M. O., and Rundensteiner, E. A. (2002), InterRing: an interactive tool for visually navigating and manipulating hierarchical structures, in 'Proceedings of the IEEE Symposium on Information Visualization (Infoviz '02)', IEEE Computer Society, Washington, DC, USA, pp. 77–84.

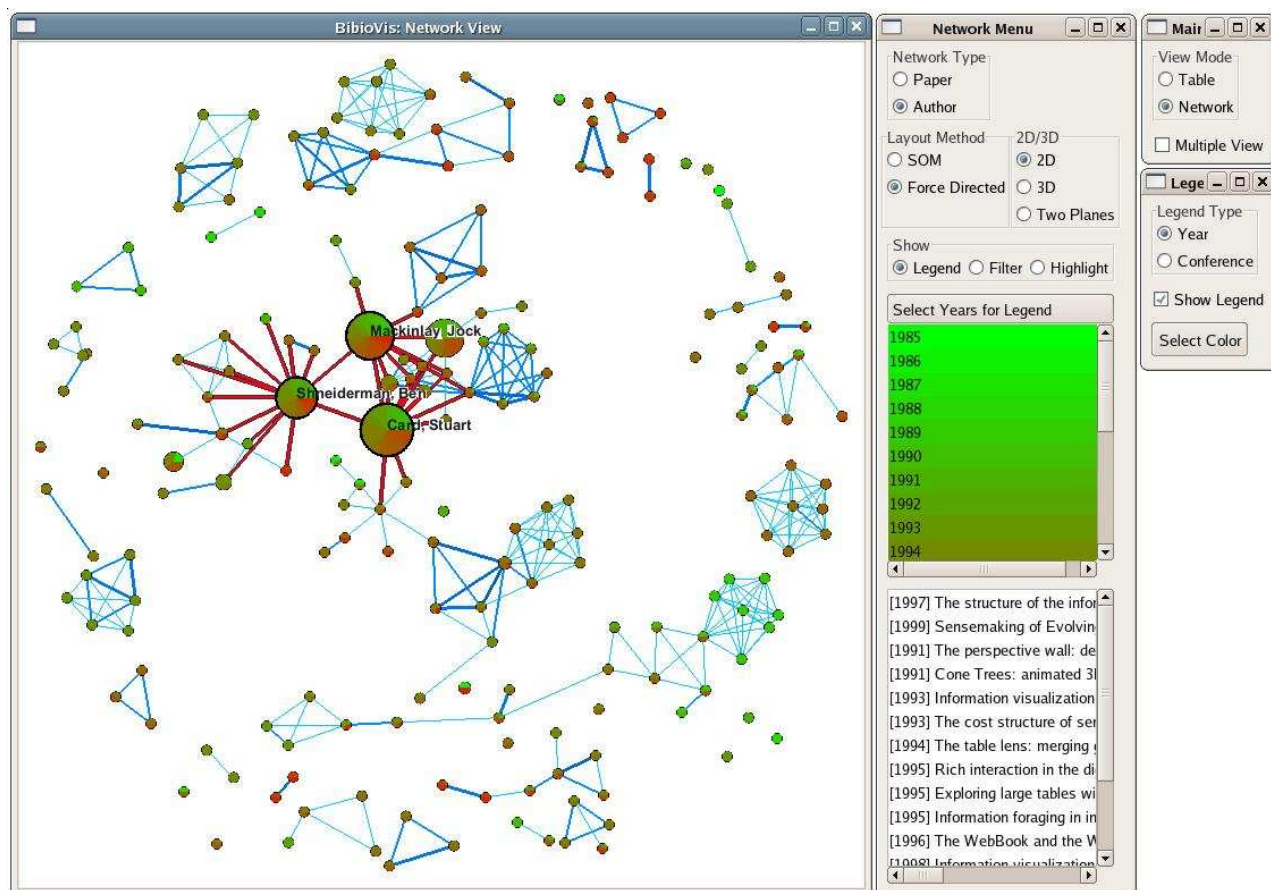


Figure 10: The Network View of authors shows that Card and Mackinlay are closely connected to Shneiderman.

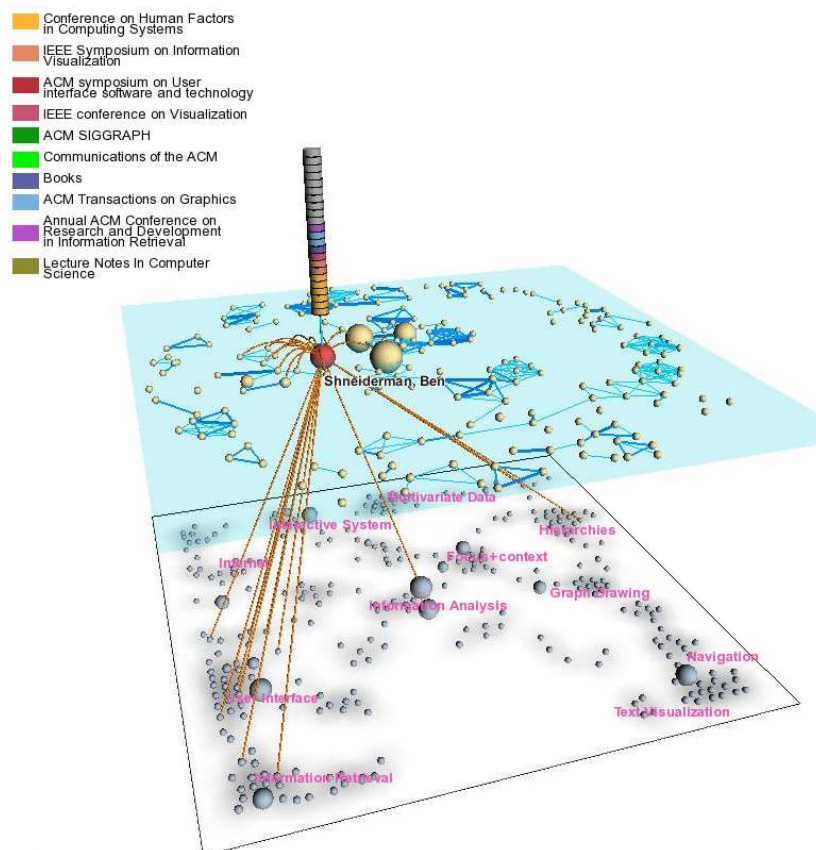


Figure 11: The dual-plane view shows Ben Shneiderman’s main research area is “user interface.”

Information Gathering Support Interface by the Overview Presentation of Web Search Results

Takumi Kobayashi

Kazuo Misue

Buntarou Shizuki

Jiro Tanaka

Graduate School of Systems and Information Engineering
University of Tsukuba,
Tennoudai 1-1-1, Tsukuba, Ibaraki, Japan 305-8571
Email: {takumi, misue, shizuki, jiro}@iplab.cs.tsukuba.ac.jp

Abstract

The Internet consists of several billion documents. Choosing information from such a great number of Web pages is not easy. We do not think that the interfaces of traditional search engines that divide search results into dozens of pages regardless of genre and display the results as a text-based list are necessarily useful. We propose an interface that helps the user to intuitively understand the entire Web search result and to gather information. Our system analyzes and classifies Web search results and presents the classification results to the user on one screen.

Keywords: Web search, Information gathering, Clustering, Visualization, Interface, Hyperbolic tree

1 Introduction

The Internet is used by people all over the world. As it spreads, the volume of information available and the number of Web pages is increasing rapidly. According to Internet Systems Consortium, Inc., the number of hosts connected to the Internet is more than 350 million (Internet Systems Consortium, Inc. 2005).

Having such a huge amount of information, we need to use a search engine to retrieve the information that is relevant to us. When we conduct a Web search, we can use different kinds of retrievals to obtain different kinds of information. We may conduct a Web search that finds one or more Web pages related to a specific topic. One example is a retrieval to obtain information on various topics about Japan. Andrei Broder calls such a Web search an "informational Web search" (Broder 2002). When a user conducts such a search, (s)he must view a great number of Web search results from various viewpoints and choose from them. A search for information about Japan is abstract and includes information on politics, traffic, tourist spots, and so on. Therefore, the user should not gather information from a specific Web page alone but should gather information from a variety of pages.

A conventional keyword search interface presents various problems when we conduct an informational Web search. These problems make the Web search difficult. Gathering information efficiently is difficult when using a conventional Web search interface that presents search results as a text-based list because such a one-dimensional list consists of various genres of Web pages. In addition, the display method, which

divides search results into dozens of pages, does not allow the user to intuitively understand the features of the entire search result. Therefore, the user might overlook profitable information.

We propose a system that classifies Web search results according to the content of the pages. Our system presents the classification results with some labels on one screen.

2 Overview Presentation System

We propose an interface that aids information gathering with the following features to solve the problems of conventional Web search interfaces.

1. The interface arranges similar Web pages in neighborhoods in a two-dimensional space.
2. The interface presents classification results to the user on one screen.

Our system categorizes Web search results into clusters according to the content of the Web pages to arrange similar Web pages in neighborhoods. The concept of feature 1 is shown in Figure 1. The user can gather information more efficiently because the system does not scatter various genres of Web pages but arranges them in neighborhoods. Web pages that the user does not want in the search results are arranged in neighborhoods according to their genres. Thus, the user can avoid looking through distracting scattered Web pages which (s)he does not want.

Our system presents search results in a two-dimensional space. This display method offers the user a retrieval that is more flexible than a one-dimensional presentation interface because the user does not have to sequentially check a one-dimensional list.

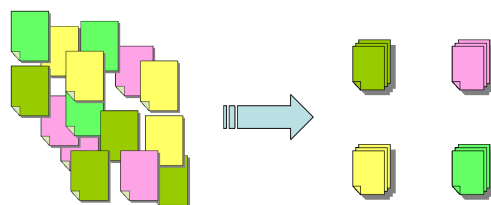


Figure 1: Feature 1

Conventional Web search engines visualize search results as text-based lists. However, one problem with these search engines is that they return too much information. The user cannot intuitively understand the entire search result. In addition, the result list is divided into small portions, but only 5 to 10 results can feasibly be viewed on the screen at any one time without scrolling, which forces the user to click on

the next button to view the next set of results. Our system presents classification results to the user on one screen. Thus, the user can easily understand the entire Web search result in an intuitive manner rather than having an interface that divides the search results into dozens of pages. The user also can obviate the need to frequently click the next button to view multiple search results.

When our system presents search results, it uses labels and the titles of classified Web pages. The labels are composed of several words that represent the features of the cluster. The user can easily find multiple relevant Web pages by referring to the labels.

3 Prototype of the Overview Presentation System

Our overview presentation system consists of two parts: the clustering of the Web search results and the presentation of the clustering result. The composition of our system is shown in Figure 2

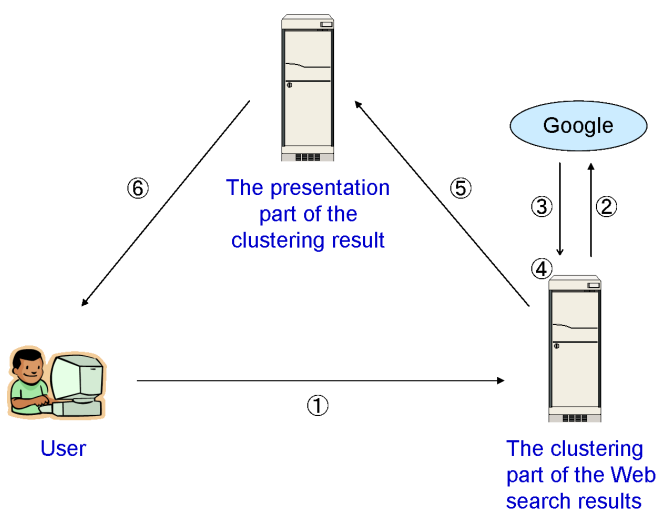


Figure 2: ①Search query, ②GoogleAPI, ③Search results, ④Analyzes Web pages and conducts clustering, ⑤Clustering result, ⑥Visualization

3.1 The Clustering of Web Search Results

To cluster Web search results, multiple Web pages found by a Web search engine are analyzed according to their content and classified into clusters based on the result. When the user enters a search query, our system uses GoogleAPI(Google 2005) to get the relevant URLs. Next, the system generates the HTML files corresponding to the URLs and analyzes them.

To analyze the HTML files, the system extracts words that appear in the HTML files using morphological analysis. The system expresses each HTML file with the vector space method, using the tf-idf algorithm. An HTML file does not consist of simple sentences but has a tree structure of commands called *tags*. We think that our system can better extract the features of Web pages by using the structural information of HTML files.

For example, we think that parts of Web pages modified with *TITLE tags*, *H tags*, and *STRONG tags* contain words that represent the features of Web pages because these parts are highly likely to contain the main points of pages and be words the author of the page wants to emphasize. The words that appear in such parts are given heavier weights by the system than the words that the author does not intend to

emphasize. *META tags* might contain explanations and features of the page not included directly in the page. Our system can extract information from the words on pages with such structures. Web pages that use *FRAME tags* or *IFRAME tags* can be recursively analyzed by obtaining the URLs the tags refer to.

After the system analyzes the HTML tags, it forms clusters based on the document vectors obtained using the analysis. We used hierarchical clustering(Everitt 1993). First, our system considers each Web page as one cluster. Next, the system compares the inner products of vectors that represent the features of each Web page. The two clusters with the biggest inner product value are combined and considered a new cluster. The feature vector of the new cluster is a composite of the feature vectors of the two original clusters. The system repeats this combination process until all the Web pages are part of one cluster. Finally, the system makes a tree diagram in which similar Web pages are arranged in neighborhoods.

When the system combines two clusters, it determines three words that are highly related to both clusters. These words represent a common feature of the two clusters. The system obtains them when it calculates the inner product value of the two clusters. They are important words that compose the label of the two clusters in the presentation of the clustering result.

3.2 The Presentation of the Clustering Result

In general, because Web search results are huge, the tree diagram of the clustering is also huge. Our system uses *Hyperbolic Tree* to fit such huge tree diagrams on one screen and present them to the user. Hyperbolic Tree is a technique proposed by John Lamping for arranging huge tree diagrams on the hyperbolic plane(Lamping, Rao & Pirolli 1995). Hyperbolic Tree can visualize and manipulate large hierarchies. Using Hyperbolic Tree, partial trees near the center of the screen are displayed larger, and those far from the center are displayed smaller. The user can move the focus by moving a partial tree to the center using the mouse. This movement of focus is shown in Figure 3. These features enable more information to fit on one screen than does a usual tree diagram. By moving the focus, the user can focus on relevant portions of the hierarchy while still seeing it embedding in the context of the entire hierarchy.

When the user collects information, (s)he cannot obtain the necessary amount of information by only displaying the tree diagram of the clustering result using Hyperbolic Tree. Thus, our system determines three words that strongly show the relationship between the parents nodes of two clusters to help the user gather information. The relationship between Web pages A, B, and C, and words 1-6 is shown in Figure 4.

In Figure 4, Web pages A and B have a strong relationship with words 1, 2, and 3. Cluster AB, which consists of Web pages A and B, has a strong relationship with words 4, 5, and 6. The user can gather information more efficiently because the system adds these labels that show the features of the clusters.

The clustering technique we used sometimes composes clusters that consist of Web pages that are not so similar. As a result, tree diagrams that look like staircases might be formed. We have improved the display interface to solve this problem. Our system bring Web pages that cluster in a staircase pattern together in one cluster under the label "Others." The system offers an attractive presentation screen by transforming the tree diagram.

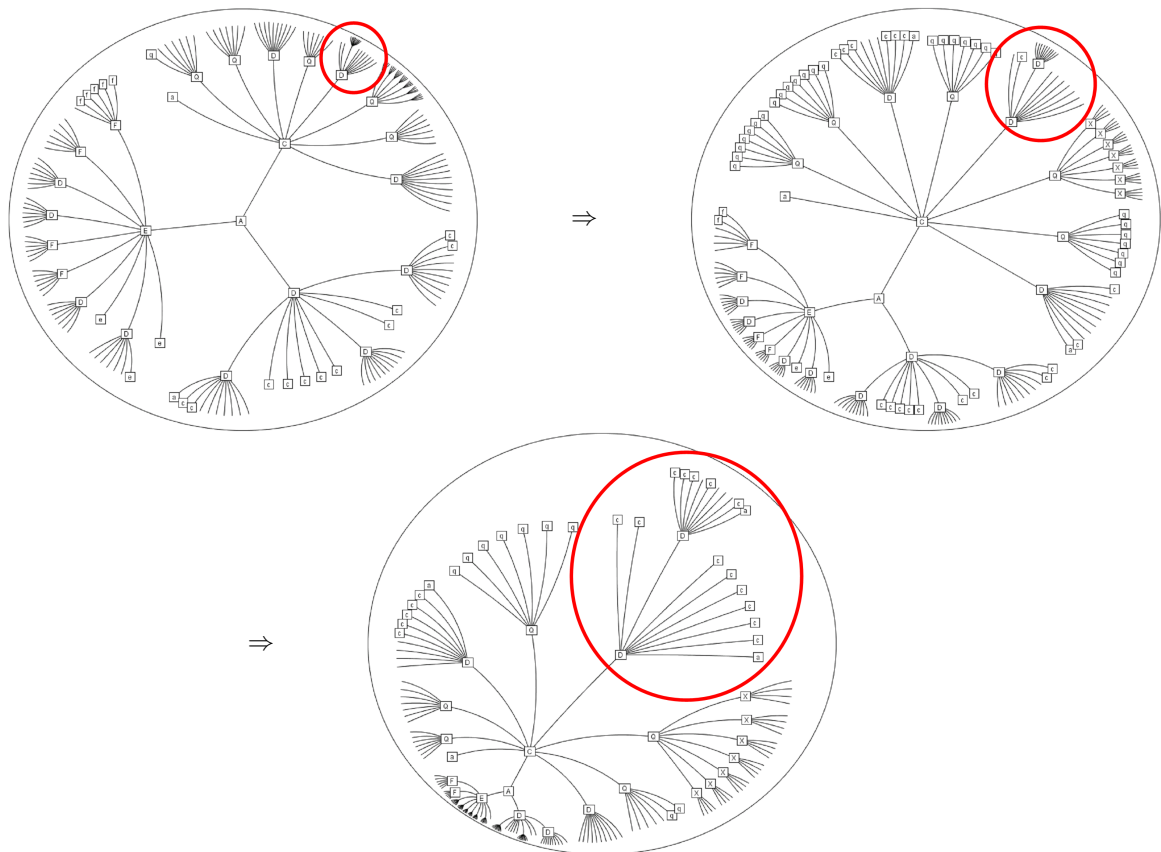


Figure 3: Movement of focus.

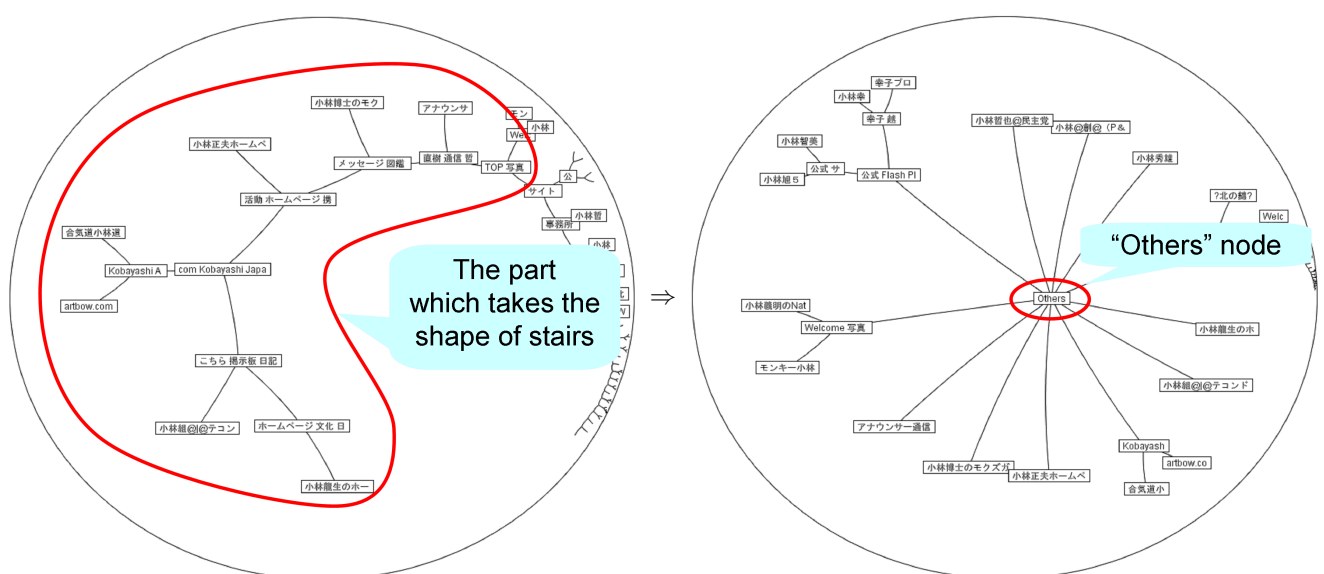


Figure 5: Improved display interface.

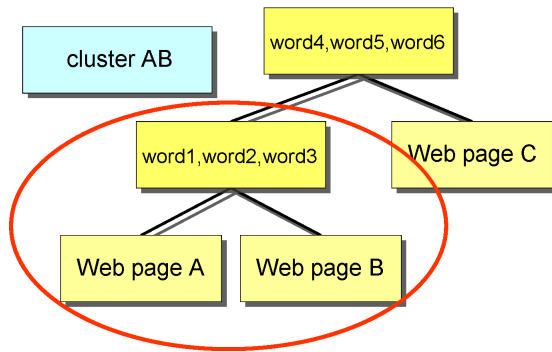


Figure 4: Relationship between clusters and labels.

The left side of Figure 5 shows a cluster in which the Web pages are not so similar. This part has a staircase shape and might confuse the user if displayed as it is. On the right side of Figure 5, our system has brought the Web pages in the staircase pattern together in an "Others" node. In this chart, the useless labels in the chart on the left have been eliminated. Thus, the system increases the number of Web pages that can be displayed at a time. The user can intuitively understand the entire Web search result and gather information efficiently using our improved display interface.

4 Example Application of Prototype System

When the user wants to examine Japanese history from various viewpoints, (s)he can search the Web using the query "Japan, History." A conventional search engine sequentially displays the pages that include the words "Japan" and "History". The result is displayed in a one-dimensional text-based list, and the list order depends on the search engine. Thus, the user must sequentially check each item on the list because various genres of Web pages are mixed together. Many of these pages may be irrelevant to the user. In our example, the user wants to find pages about Japanese history, but many pages about fortune-telling (which the user does not intend) are included in the list of the search results. We think that this is annoying for the user and makes information gathering difficult.

Our system's presentation screen for the search query "Japan, History" is shown in Figure 6. The system formed clusters according to the algorithm and presented the results. In Figure 6, the lower right partial tree consists of fortune-telling pages including the words "Japan" and "History." Thus, the user need not check any further because (s)he can understand at a glance that the Web pages in this partial tree are concerned with fortune-telling (which the user does not intend). The Web pages that could not be clustered by the algorithm are gathered in the lower left partial tree under the label "Others." They can be said to be unique Web pages in the search result.

The partial tree shown in Figure 6 is rather large. The user can gather information by moving the focus with the mouse while viewing the labels and traces of the partial tree. A snapshot of the partial tree of pages related to the word "Geography" and including the words "Japan" and "History" is shown in Figure 7. A snapshot of the partial tree of pages related to the word "Textbook" and including the words "Japan" and "History" is shown in Figure 8. When the tree is observed in detail, the user can see that some Web pages are about the problem of military prostitution and textbooks. The partial trees are of various genres, e.g., books about Japanese history,

educational institutions that deal with Japanese history, and academic societies and theses concerning Japanese history.

When using an interface where search results are presented in a one-dimensional text-based list, the user must check each Web page on the list, including various genres of Web pages. When using our system, only the partial trees composed of similar Web pages must be checked. Thus, when the user needs to gather information from various genres of Web pages, (s)he can efficiently choose relevant information in each partial tree without changing the idea. The user can easily understand the whole search result by referring to the partial trees because the result is displayed on one screen.

5 Enhancing the User Interface

In this section, we introduce our intended enhancements to the user interface. The user will be able to conduct an informational Web search more efficiently by using these enhancements.

5.1 Using Ranking Information

Our system uses GoogleAPI to acquire Web search results. We plan to reflect the GoogleAPI ranking of search results in the display interface. Google uses *Page rank* to calculate the importance of a Web page. *Page rank* automatically calculates the relative importance of Web pages (Page, Brin, Motwani & Winograd 1998) based on a recursive relationship: Web pages that many high-quality Web pages have links to are high-quality. *Page rank* is a brilliant technique for offering the user high-quality Web pages. Our system allows the user to efficiently retrieve high-quality pages by combining the ranking information provided by *Page rank* with the search result presentation method using Hyperbolic Tree.

Our system presents clustering results by using Hyperbolic Tree. Therefore, our system does not sequentially display Web pages in the order of page rank from highest to lowest, like the Google interface does. Instead, our system offers the user the ranking information by shading the nodes. The system colors the nodes of important Web pages with dark colors and those of unimportant Web pages with lighter colors. Thus, the user can easily find the important Web pages in a large search result. In addition, the user can more intuitively understand the whole search result by using the shading information.

5.2 Generating Summaries of Web Pages

In our system, each node of a hyperbolic tree shows the title of a Web page. However, offering only titles does not necessarily help the user gather enough information because (s)he may find the content of the Web pages difficult to guess. We plan to offer the user some additional text that summarizes the content of Web pages. A summary of a Web page can be obtained by using the interface of GoogleAPI. However, because a summary obtained this way is mechanically excerpted text, the user still may have difficulty understanding the content of the Web page. Thus, our system generates comprehensible summaries for the user by using the tag information of Web pages.

For instance, a Meta tag modifies the summary of the Web page that the author of the page gave. H tags modify headings that reflect the content of pages well. Our system can help the user gather information by offering a summary that expresses the content of the Web page remarkably well. The summary is displayed

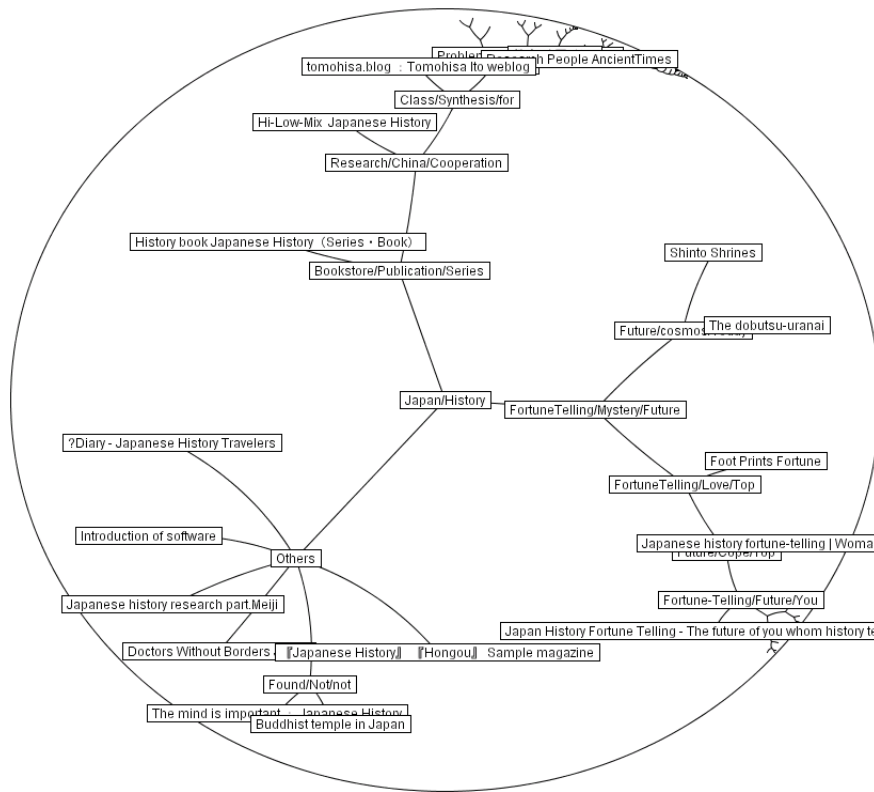


Figure 6: Presentation screen of our system for search query, "Japan, History."

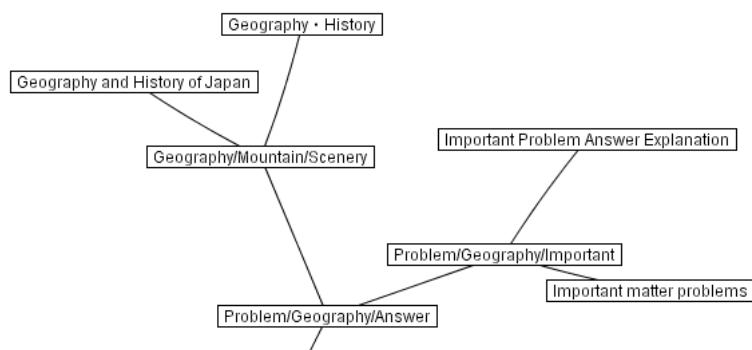


Figure 7: Partial tree related to "Geography."

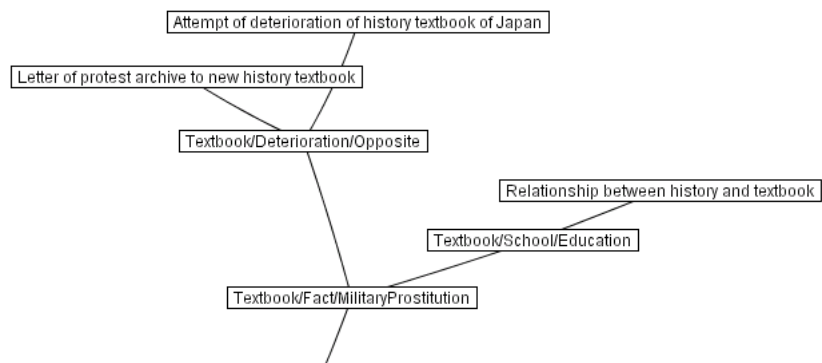


Figure 8: Partial tree related to "Textbook."

in a pop-up window when the user clicks the node that contains the title of the page.

5.3 Classification based on the User's Purpose

Web pages can be classified using standards other than their content. When we search the Web, we browse many types of Web pages. For instance, we might browse the top page of a certain site, a collection of Web links, a page with many characters, a page with many images, a page for shopping, and a Weblog or BBS. This means that Web search results can be classified based on the user's purpose. We considered a classification method and presentation interface that helps the user to intuitively understand many types of Web pages in the search result based on the user's purpose. The system uses colors to appropriately classify each type of Web page. Thus, the system can help the user gather information by combining the colors with the previously mentioned node shading.

6 Discussion and Future Work

Our system generates binary tree diagrams to display search results using our clustering method. When a search result is very large, the grain size of the clusters in the diagram might be too small. In that case, the grain size of the clusters is increased and user's information gathering might become easy by bringing the certain size of partial tree together in one cluster. A common feature of multiple Web pages must be extracted and used as a label.

The execution time of the entire system must be considered for practical use. Web searches take a long time if the system analyzes HTML code and clusters for each Web search. To solve this problem, the system forms clusters by using document vectors with the dimensions compressed. Thus, the computational effort for clustering is reduced, and the calculation time is shortened. In addition, the computational effort when the Web is searched can be reduced if the system collects Web pages and processes them beforehand.

7 Related Work

Periscope is a Web search interface that presents Web search results using a method other than a text-based list (Wiza, Walczak & Cellary 2004). Periscope arranges search results in a three-dimensional space. When Periscope classifies Web pages, it considers the host names, languages, sizes, etc. of the pages. A system that classifies Web pages by host names and arranges them in a two-dimensional space was developed by Roberts, Boukhelifa, and Rodgers (Roberts, Boukhelifa & Rodgers 2002). These interfaces do not classify Web pages based on their content. Thus, these systems do not offer enough information for the user.

Our system takes advantage of the tag information of Web pages and clusters pages based on their content. It presents classification results on one screen with additional information that represents the features of the clusters.

8 Conclusion

We considered the problems of *informational Web searches* using conventional Web search interfaces; these problems are caused by the rapid increase in

the volume of information on the Internet. We developed an interface that solves these problems. Our system can classify Web search results according to the content of Web pages by analyzing HTML tree structures. Our system presents the classification results to the user on one screen. The user can efficiently conduct *informational Web searches* by referring to the labels presented in a hyperbolic tree.

References

- Broder, A. (2002), 'A taxonomy of web search', *SIGIR Forum* **36**(2), 3–10.
- Everitt, S., B. (1993), *Cluster analysis*, 3rd edn, London: E. Arnold.
- Google (2005), 'Google Web APIs', <http://www.google.co.jp/apis/>.
- Internet Systems Consortium, Inc. (2005), 'ISC Internet Domain Survey', <http://www.isc.org/index.pl?ops/ds/>.
- Lamping, J., Rao, R. & Pirolli, P. (1995), A focus+context technique based on hyperbolic geometry for visualizing large hierarchies, in 'CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems', ACM Press/Addison-Wesley Publishing Co., pp. 401–408.
- Page, L., Brin, S., Motwani, R. & Winograd, T. (1998), The pagerank citation ranking: Bringing order to the web, Technical report, Stanford Digital Library Technologies Project.
URL: citeseer.ist.psu.edu/page98pagerank.html
- Roberts, J., Boukhelifa, N. & Rodgers, P. (2002), Multiform Glyph Based Web Search Result Visualization, in 'the Sixth International Conference on Information Visualisation (IV '02)', IEEE, pp. 549–554.
- Wiza, W., Walczak, K. & Cellary, W. (2004), Periscope: a system for adaptive 3D visualization of search results, in 'Web3D '04: Proceedings of the ninth international conference on 3D Web technology', ACM Press, pp. 29–40.

NeL²: Network Drawing Tool for Handling Layered Structured Network Diagram

Nagayoshi Nakazono¹Kazuo Misue²Jiro Tanaka²

¹College of Information Sciences, ²Department of Computer Science
University of Tsukuba,

1-1-1 Tenoudai, Tsukuba, Ibaraki, 305-8573, Japan

Email: ¹zono@iplab.cs.tsukuba.ac.jp, ²{misue, jiro}@cs.tsukuba.ac.jp

Abstract

We propose a “layered structured network diagram,” which consists of layers of time differences. We have implemented a tool called “NeL²” for handling layered structured network diagrams. Layered structured network diagrams have multiple accumulated layers and are not single diagrams. Using this layered structure, time differences can be included in one diagram. In addition, various type of information, such as the overall tendency in a diagram during a specific period of time, can be visualized.

We used layered structured network diagrams to show the co-authorship network of academic literature. Changes in the co-authorship network become visible using the layered structured network diagram. We can read various information such as changes of active research communities and other phenomena. In addition to co-authorship networks, the layered structured network diagram can be applied to the visualization of various data, such as idea processors, changes of Web sites, and others.

Keywords: Visualization, Graph drawing, Time series network, Layered structure, Interactive system

1 Introduction

We propose a network diagram with a layered structure. This represents data that have the form of networks, such as organizational structures, computer networks, or co-authorships of academic literature. We often visualize such data in order to make easy to understand. Most network diagrams which visualized from these data represent the state of a network at a certain point in time (a snapshot). Therefore, several snapshots must be arranged and compared in order to determine “changes (differences)” over time, for example changed of the state of a network from last year until the present, or what is the change of the state of network in the past ten years. In addition, when using tools that show only one network diagram, users must create the difference from snapshots whenever we peruse network changes.

We considered the evolution of a network that continuously changes as a sequence of differences. Our

method expresses the sequence of differences as a layered structure. We developed a tool for handling such diagrams, named “NeL²” (Figure 1). Our expression method does not regard a network diagram as one figure, but instead as an accumulation of several layers. For example, if we want to refer to changes in networks during past year, all we have to observe is only the layer that expresses “the data of last year.” To view the transition over the last ten years, the changes piled up and the layers of the last ten years can be seen. In the same way, piling up all layers enables users to understand the present state of a network. Using a layered structure, it becomes possible to handle differences of network efficiently.

In this paper, we defined layered structured networks mathematically and explained the tool NeL². We also used the co-authorship network of academic literature as an example for a layered structured network diagram. In addition, we hypothesized other kinds of applications.

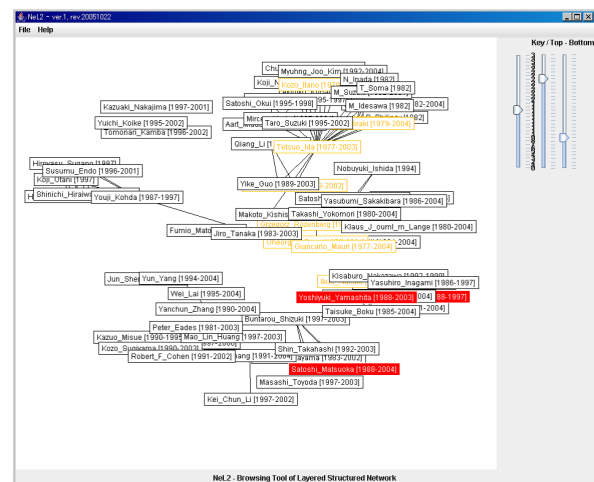


Figure 1: Screenshot of NeL²

2 Related Work

Various approaches have been tried in research related to the visualization of evolving networks. Chen *et al.* proposed a technique that visualizes co-citation networks and evolutionary process, targeting scientific papers (Chen & Carr 1999). Toyoda *et al.* developed the system WebRelievo, which visualizes the evolutionary processes of Web pages using difference views and cluster views (Toyoda & Kitsuregawa 2005). Erten *et al.* proposed a framework that arranges changing graphs in a line, piling it up three-dimensionally, and applied their framework to visualizing the relationship of scientific literature (Erten, Kobourov, Le & Navabi 2003).

In addition, there is a lot of research regarding the visualization style of network. Heer *et al.* developed a toolkit called “prefuse” in order to visualize interactively in various forms (Heer, Card & Landay 2005). Carlis *et al.* (Carlis & Konstan 1998) and Yee *et al.* (Yee, Fisher, Dhamija & Hearst 2001) expanded on former graph layouts and proposed an arrangement technique based on spirals or concentric circles.

This different methods and techniques express changes in networks by keeping snapshots of networks at different points. By contrast, NeL² pays attention to the differences between two points in time, not to states at different times. That is, we try a new approach; we keep the data as a sequence of differences and expresses the state of network at certain points by integrating the differences.

3 Expression of Network Diagram using Layered Structure

We introduce a layered structure to express network diagrams that change over time. Network data is managed separately as several layers, not as a single diagram, and each one depending on the update time. We display the data and color it in a flexible manner. This approach helps users to read large amounts of information from network diagrams.

3.1 Concept of Layered Structure

Our network diagrams which is handled by this research have a concept of layered structure as shown in Figure 2. The structure is similar to layers of transparencies (like OHP sheets). Updates for certain times are saved on individual sheets (layers), and subsequent updates which were done at next period are saved by piling a new layer on existing layers. Changes from each update are saved on each layer. Using the layered structure enables the following operations:

- Network diagrams can be seen by stacking all layers between certain points. It is possible for us to trace the changes of a network; we can trace history in chronological order from past to future when we stack the layers, and we are able to trace history in reverse chronological order from future to past when we remove layer in sequence (Figure 3, 4, 5).
- We can see changes of the network in that point by perusing the optional layer independently.
- We can see changes of the network in a certain period by stacking several layers.

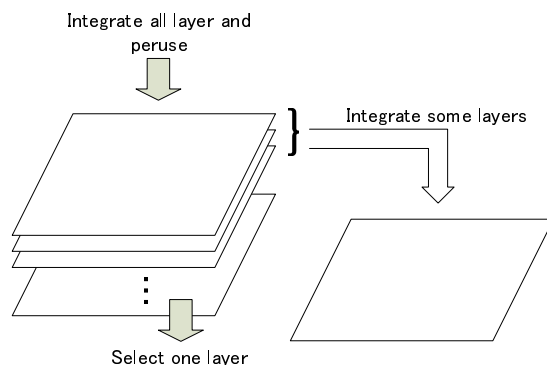


Figure 2: Concept of layered structure

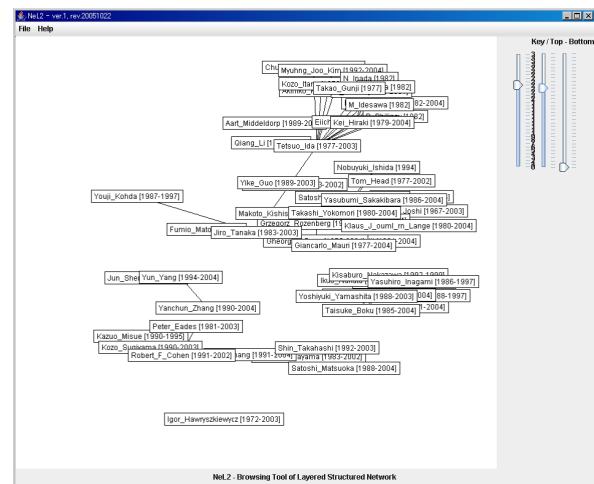


Figure 3: Diagram before difference layer is added

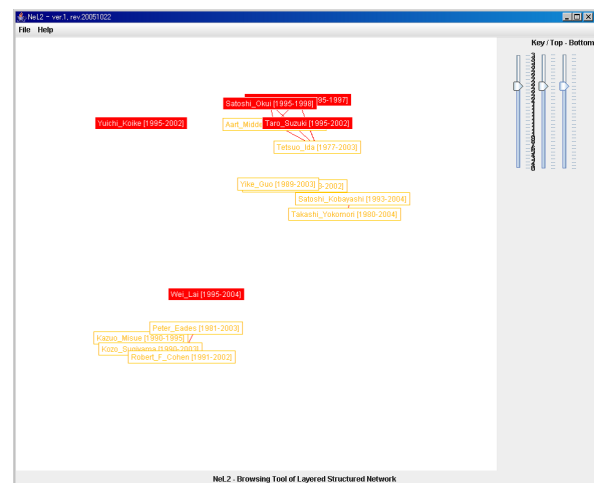


Figure 4: Difference layer

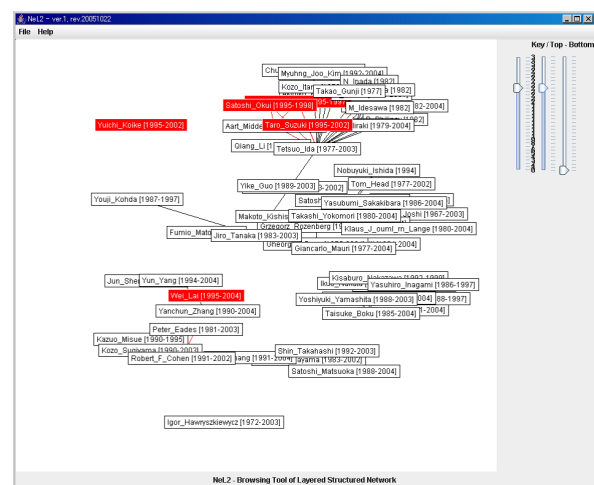


Figure 5: Diagram after difference layer is added

3.2 Advantage of Using Layered Structure

There are several advantages in using network diagrams with the layered structure.

First, only a single network diagram is needed when using a layered structure. Previously, several diagrams must be arranged and compared to see the differences in a network. It is possible for layered structure to express these changes in one network diagram, because the network data is composed of “a gathering of differences” in information between layers. Existing methods of network visualization display differences made from all network data. On the other hand, NeL² determines difference expressions at the data level and expresses all data by integrating these differences. This method is suitable for visualizing the evolution of networks.

Second, layered structure makes the operation of network diagrams easy to understand and intuitive. We usually perform actions similar to those performed with a layered structure, for example “extract pp. 10–16 from stack of documents (papers).” In the same way the operation is possible even in the network diagram which uses layered structure.

Third, layered structure helps users grasp the features of networks easily. Users cannot see changes over time in network diagrams that are not divided into layers, because all elements are indicated in one diagram simultaneously. By introduction of layered structure, users can change the indications of nodes and edges on every layer.

4 Definition of Layered Structure

4.1 Graph

In the layered structure which we propose, layers are accumulated in chronological order, and the network diagram represented by the layers is visible at each time. We considered a network diagram at each time as a graph, and express it as the following triplet:

$$G = (N, E, \varphi),$$

where N and E are a set of nodes and a set of edges, respectively. The function $\varphi : E \rightarrow N \times N$ is the incidence function of edges. When $\varphi(e) = (v, w)$, edge e is connected to nodes v and w .

4.2 Layers

Layers express differences in the graph at different times. l_i is the i^{th} layer from the bottom and is expressed as the following:

$$l_i = (N_i^+, E_i^+, N_i^-, E_i^-, \varphi_i^+)$$

where N_i^+ and E_i^+ are sets of nodes and edges added to the i^{th} layer, and N_i^- and E_i^- are sets of nodes and edges removed from the i^{th} layer. The function φ_i^+ is an incidence function of the set of added edges. The domain and range of this function are as follows:

$$\begin{aligned} \varphi_i^+ &: E_i^+ \rightarrow N' \times N' \\ N' &= (N_{i-1} \cup N_i^+) \setminus N_i^- \end{aligned}$$

4.3 Accumulated Layers

The accumulation of layers is represented by \oplus . The operation of piling on top of the next layer on graph

G_{i-1} , which is visible by piling up all layers to the $(i-1)^{\text{th}}$ layer, is represented as follows:

$$G_{i-1} \oplus l_i = G_i = (N_i, E_i, \varphi_i),$$

where N_i , E_i and φ_i are represented as below. G_0 is an empty graph, which has no nodes or edges.

$$\begin{aligned} N_i &= (N_{i-1} \cup N_i^+) \setminus N_i^- \\ E_i &= (E_{i-1} \cup E_i^+) \setminus E_i^- \\ \varphi_i &: E_i \rightarrow N_i \times N_i \\ \varphi_i(e) &= \begin{cases} \varphi_i^+(e) & \text{if } e \in E_i^+ \\ \varphi_{i-1}(e) & \text{if } e \in E_{i-1} \end{cases} \end{aligned}$$

4.4 Integrated Layers

The integration of a series of layers is also represented as \oplus and is defined as follows.

$$l_i \oplus l_{i+1} = (N_i^{++}, E_i^{++}, N_i^{--}, E_i^{--}, \varphi_i^{++}),$$

where N_i^{++} , E_i^{++} , N_i^{--} , E_i^{--} and φ_i^{++} are

$$\begin{aligned} N_i^{++} &= (N_i^+ \cup N_{i+1}^+) \setminus N_{i+1}^- \\ E_i^{++} &= (E_i^+ \cup E_{i+1}^+) \setminus E_{i+1}^- \\ N_i^{--} &= (N_i^- \cup N_{i+1}^-) \setminus N_{i+1}^+ \\ E_i^{--} &= (E_i^- \cup E_{i+1}^-) \setminus E_{i+1}^+ \\ \varphi_i^{++} &: E_i^{++} \rightarrow N'' \times N'' \\ N'' &= (N_{i-1} \cup N_i^+ \cup N_{i+1}^+) \setminus (N_i^- \cup N_{i+1}^-) \\ \varphi_i^{++}(e) &= \begin{cases} \varphi_i^+(e) & \text{if } e \in E_i^+ \\ \varphi_{i+1}^+(e) & \text{if } e \in E_{i+1}^+ \end{cases} \end{aligned}$$

4.5 View of Layer

Changes in a network diagram can be seen by looking at only one layer. In the difference expression, the added and the deleted elements are drawn in different colors as a view of a layer. A colored graph G^c , which shows differences using colors, is expressed as following:

$$G^c = (N, E, \varphi, \gamma_N, \gamma_E),$$

where N , E and φ are similar to the definitions for a graph G . The functions $\gamma_N : N \rightarrow C$ and $\gamma_E : E \rightarrow C$ specify the colors of nodes and edges. C is a set of colors. Here, only three colors are considered abstract colors: c_n for unchanged nodes, c_a for added nodes or edges, and c_d for deleted nodes or edges.

A colored graph $G^c = (N, E, \varphi, \gamma_N, \gamma_E)$, which is the view of layer $l_i = (N_i^+, E_i^+, N_i^-, E_i^-, \varphi_i^+)$, is represented as follows:

$$\begin{aligned} N &= N_i^+ \cup N_i^- \cup N^u \\ N^u &= \{v \mid (v, w) \in \varphi_i(E) \vee (w, v) \in \varphi_i(E)\} \\ &\quad \setminus (N_i^+ \cup N_i^-) \\ E &= E_i^+ \cup E_i^- \\ \varphi &: E \rightarrow N \times N \\ \varphi(e) &= \begin{cases} \varphi_i^+(e) & \text{if } e \in E_i^+ \\ \varphi_{i-1}(e) & \text{if } e \in E_{i-1} \end{cases} \\ \gamma_N(v) &= \begin{cases} c_n & \text{if } v \in N^u \\ c_a & \text{if } v \in N_i^+ \\ c_d & \text{if } v \in N_i^- \end{cases} \\ \gamma_E(e) &= \begin{cases} c_a & \text{if } e \in E_i^+ \\ c_d & \text{if } e \in E_i^- \end{cases} \end{aligned}$$

where N^u is a set of nodes that does not change in the layer and connects to edges changed in the layer. Because neither N_i^+ nor N_i^- include such nodes, a graph cannot be constructed with only a set of nodes $N_i^+ \cup N_i^-$. Therefore N^u is used to construct a graph.

5 Implementation

We developed a tool called “NeL²”¹ which displays and operates layered structured network diagrams in Java (JDK 5.0).

5.1 Data Format of Network

We described the data representing a network diagram in the original XML document format. Data formats used by existing network visualization techniques have no concept of layers, unlike our technique. Thus, we designed a new data format to express layer information. An example of network data with a layered structure is shown in Figure 6.

```
<?xml version="1.0" encoding="UTF-8"?>
<graph>
  <layer id="0" date="1959">
    ....
  </layer>
  ....
  <layer id="12" date="1983">
    <node id="40">
      <label>Jiro_Tanaka</label>
      <position x="411" y="439" />
    </node>
    ....
    <edge id="0">
      <from>36</from>
      <to>1</to>
      <label>1983</label>
    </edge>
    ....
    <node_event id="0"
      node_id="2" command="del" />
    ....
  </layer>
  ....
</graph>
```

Figure 6: Description example of the network data with layered structure in XML

A **layer** element defines a layer. This element has an updated date with its ID because a network diagram is updated each time when a layer is added. A **node** element defines a node and corresponds to $v \in N$ of the mathematical definition. The **node** element has two attributes: its ID and the ID of the layer the node belongs to. The element also has a **label** element and a **position** element that represent the label and the x-y coordinates of a node, respectively. An **edge** element defines an edge that connects two nodes. It corresponds to $e \in E$ of the mathematical definition. The **edge** element has two attributes: its ID and the ID of the layer the node belongs to. The element also has **from**, **to** elements and **label** element, which represent nodes connected at both ends and a label of an edge, respectively.

Nodes or edges can be deleted when the network diagram is updated. In this case, **node_event** or **edge_event** elements are added instead of modifying **node** or **edge** elements corresponding to deleted elements. These elements correspond to N^- and E^- of the mathematical definition, respectively. Operations to nodes and edges are recorded on these elements. When the tool draws a graph, these elements overwrite the nodes and edges themselves. Using this

data format, there is the merit that past states are saved even if nodes or edges are deleted.

5.2 System of Data Operation

At the initial state immediately after loading, the XML document, all nodes and edges are visible regardless of the layers they belong to.

We used sliders as an interface for the tool to change visible layers, so that users can operate network diagrams intuitively, as shown in Figure ???. We prepare three sliders; the two sliders on the right side specify the range of visible layers, specifically the top and bottom layers. User can appoint a layer which he or she pays attention to (key layer) by using slider on left side. Then nodes and edges belonging to the appointed key layer are highlighted (colored red) on the network diagram. This function makes comparing the key layer and other layers easy.

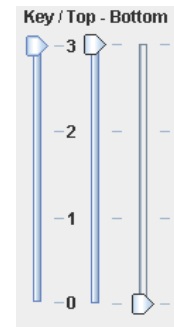


Figure 7: Sliders

6 Application to Visualize Co-authorship Networks

Network diagrams with layered structure can be applied to various situations. In this section, we used this diagram to visualize co-authorship networks of scientific literature.

6.1 Visualization of Coauthorship

Co-authorship networks of scientific research continuously change. Understanding these changes of co-authorship networks can help explain changing circumstances, trends for researchers or within research fields, and other things.

Many researchers have analyzed co-authorship networks (Brandes & Willhalm 2002). For example, Yoshikane *et al.* studied global network analysis, which looked around the field of the whole research area (Yoshikane, Nozawa & Tsuji 2005).

Generally, co-authorship is analyzed quantitatively using mathematical techniques. However, in order to understand tendencies in network data, conjecture using visual diagrams is more convenient than using mathematical formula. This kind of presumption suggests the result of analysis that follows afterwards, and helps analysis. Layered structured network diagrams are useful in these kinds of cases.

6.2 Convert Co-authorship Network Data

We used a co-authorship network for scientific papers, the Digital Bibliography and Library Project (DBLP)² which is a scientific coauthorship database on the Web. We made the data for NeL² using the

¹The name NeL² results from the first (and second) letters of Networks, Layer, and Layout.

²<http://www.informatik.uni-trier.de/~ley/db/>

NeL2 - Browning Tool of Layered Structured Network

File Help

Key / Top - Bottom

Diagram showing a network structure with nodes and edges. Nodes are labeled with names and time ranges in brackets. Some nodes are highlighted in yellow (e.g., Tetsuo_Ida, Yike_Guo, Makoto_Kishis, Takashi_Yokomori, Taisuke_Boku, Satoshi_Matsuoka) and others in red (e.g., Jun_Shen, Yanchun_Zhang, Kazuo_Misue, Kozo_Sunawama). The diagram shows a complex web of connections between these entities.

NeL2 - Browning Tool of Layered Structured Network

113

same technique as Misue (2005). On DBLP, approximately 350 thousand authors and approximately 880 thousand collaborative relationships were registered as of September 2, 2004. If a paper was written by authors A and author B, there was one relationship, and if a paper was written by three authors, there were three.

With our application, authors were regarded as nodes, and co-authorship was regarded as edge. We selected Jiro Tanaka, one of the authors of this paper, as the basis node, and searched for nodes within distance of two steps of him. Then, we extracted a sub-graph with 118 nodes and 535 edges. The layout was calculated on the basis of this data using another tool, and string manipulation of the data was performed with a script written in Ruby. As a result, the data was converted to an XML document that could be used in NeL².

6.3 Visualization using Layered Structure

This data was read and visualized by NeL², as shown in Figure 8. One layer corresponded to one year. The key layer, as indicated by the slider on the left side, was the latest year, 2004, automatically, just after the tool load the data. Also, all layers were visible, and the two right sliders indicated 1959, the oldest year, and 2004, the latest year, respectively.

When looking at the network diagram with Jiro Tanaka as the basis node, relatively large clusters can be seen at the top, right side, and bottom. However, from this network view, we cannot understand which cluster was active or when it was active.

Accordingly, we operated the two right sliders and limited the visible layers to only 10, between 1983, the year Jiro Tanaka was added, and 1992. The key layer was set to 1990 by operating the left slider as shown in Figure 9. Modifying indication doesn't need complex operations, everyone can use it naturally with only simple explanations.

The result of modifying indication is that the node for Jiro Tanaka connected to nodes belonging to the top cluster, but not to the bottom cluster. Thus, we can understand the following things about Jiro Tanaka: in the 10 years from 1983 to 1992 (which are visible), he worked closely with the community corresponding to the top cluster, while his connection to the cluster representing the bottom community has been relatively recent.

When we set the key layer to 1990, many nodes located under the left were highlighted. This suggests that the community corresponding to this cluster under the left was very active during this period.

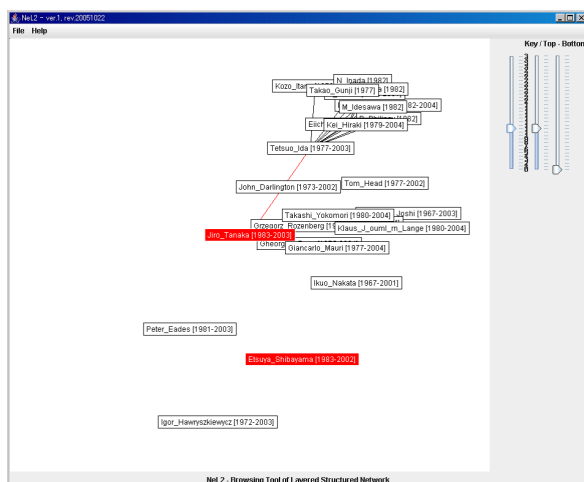


Figure 10: Basis node Jiro Tanaka appears in 1983

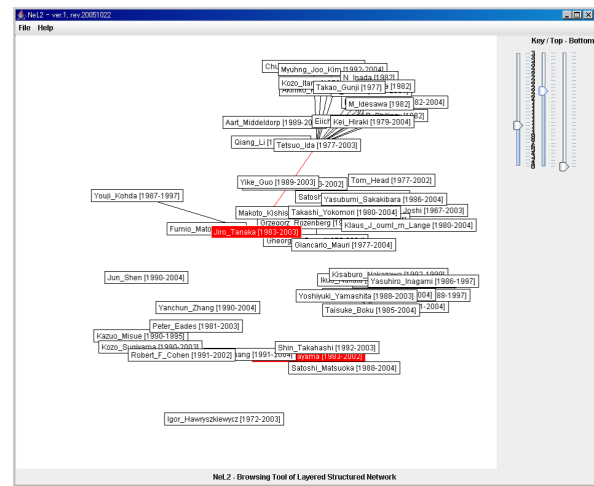


Figure 11: Communities build up

In addition, NeL² can express the alteration of networks as motion by piling layers one by one in the old order by using the slider. At first, we adjusted all three sliders to 1959 (the oldest layer); only the data 1959 is indicated in the canvas. Next, we moved the central slider (indicating the topmost visible layer) upward, adding layers one by one in the old order. At early stages, world-renowned researchers appeared, and Jiro Tanaka appeared in 1983, as shown as Figure 10. As more layers were added, connections between Jiro Tanaka and world-renowned researchers formed, and communities, such as laboratories, were built up, as shown in Figure 11. These kinds of network changes can be easily perused using only the slider operation in the layered structured network diagram.

As such, introduction of a concept of layered structure helps users to change the view of network diagrams easily, and it becomes possible for us to presume features of co-authorship networks.

7 Other Example Applications

Layered structured network diagram can be applied to the visualization of various kinds of data, not only the co-authorship of scientific papers. Several examples are introduced in this section.

7.1 Changes of Websites

Due to the spread of blogs and related tool, the number of people who have personal websites is increasing rapidly. Such sites are created, modified, or deleted on a day-to-day basis. Though much research has been done about visualization of the Web (Chi, Pitkow, Mackinlay, Piroli, Gossweiler & Card 1998), NeL² also can be applied here. By saving changes of websites on layers periodically, users can easily check the state of websites for any period, or how to strengthen or weaken relationships between sites.

7.2 Management Change Log of Visual Debugger

Zanden *et al.* developed a domain-specific debugger for one-way constraint solvers (Zanden, Baker & Jin 2004). This debugger uses a dataflow graph and saves snapshots of each time and makes differences based on them. With NeL², this kind of differences indication reaches the point where it can be actualized by designating specific layers.

7.3 Integration with Idea Processors

Idea processors are a category of software that assists in rearranging thoughts and documents, such as mind mapping³. Some tools mark ideas as tags or nodes, rearranging and making relationships between them (connecting them with lines). Several such tools exist, which use as foundation mind mapping, KJ method, and so on.

When tags are regarded as nodes and relationships are regarded as edges, we can consider that we make a network diagram. Using layered structured network diagram for some functions of idea processors, like save and play on, users can easily understand when tags (ideas) were designed or modified. Users can also see the evolution of ideas, since they are saved in chronological order using the layered structure. By the fact that this kind of past record management is introduced into the idea processor, it becomes easy to follow the arrangement and development of thought.

8 Conclusion and Future Work

In this paper, we proposed the notion of a layered structure for network diagrams and defined mathematically. We also developed a tool called “NeL²” which we developed to handle layered structured network diagrams. The tool uses layers as units expressing the changes of networks over time. In addition, it provides natural operations in order for obtaining intended diagrams. Using a layered structured network diagram, it has become possible to grasp various kinds of information, such as tendencies and processes of evolution, in networks.

Visualization of the co-authorship of scientific papers was shown as an example application for NeL². A layered structure was used to visualize the changes of co-authorship networks with community formation and activity liveness shown explicitly.

In the future, we intend to improve the drawing of the graph structure so that it appears to be truly in the form of a layer, as in Figure 2. We also want to replace the slider with an operation that allows users to choose a layer and make it visible by dragging it. We also plan to color differently not only the layer units, but also data related to a chosen node. We will design a usability test to see whether users like this kind of interface to clarify the properties and merits of this system.

References

- Brandes, U. & Willhalm, T. (2002), Visualization of Bibliographic Networks with a Reshaped Landscape Metaphor, *in* ‘Proceedings of the 4th Joint Eurographics-IEEE TCVG Symposium on Visualization 2002’, pp. 159–164.
- Carlis, J. V. & Konstan, J. A. (1998), Interactive Visualization of Serial Periodic Data, *in* ‘Proceedings of the 11th annual ACM symposium on User interface software and technology’, ACM Press, New York, USA, pp. 29–38.
- Chen, C. & Carr, L. (1999), Visualizing the Evolution of a Subject Domain: A Case Study, *in* ‘Proceedings of the conference on Visualization ’99’, IEEE Computer Society Press, Los Alamitos, USA, pp. 449–452.
- Chi, H., Pitkom, J., Mackinlay, J., Pirollo, P., Gossweiler, R. & Card, S. K. (1998), Visualizing the Evolution of Web Ecologies, *in* ‘Proceedings of the SIGCHI conference on Human factors in computing systems 1998’, pp. 400–407.
- Erten, C., Kobourov, S. G., Le, V. & Navabi, A. (2003), Simultaneous Graph Drawing: Layout Algorithms and Visualization Schemes, *in* ‘The 11th Symposium on Graph Drawing’, pp. 437–449.
- Heer, J., Card, S. K. & Landay, J. A. (2005), prefuse: a toolkit for interactive information visualization, *in* ‘Proceedings of the SIGCHI Conference on Human Factors in Computing Systems 2005’, ACM Press, New York, USA, pp. 421–430.
- Misue, K. (2005), Visual Destruction for Understanding Small-world Networks, *in* ‘Proceedings of HCI International 2005’ CD-ROM.
- Toyoda, M. & Kitsuregawa, Y. (2005), A System for Visualizing and Analyzing the Evolution of the Web with a Time Series of Graphs, *in* ‘Proceedings of the 16th ACM conference on Hypertext and hypermedia’, ACM Press, New York, USA, pp. 151–160.
- Yee, K. -P., Fisher, D., Dhamija, R. & Hearst, M. A. (2001), Animated Exploration of Dynamic Graphs with Radial Layout, *in* ‘INFOVIS’, pp. 43–50.
- Yoshikane, F., Nozawa, T. & Tsuji, K. (2005), Comparative Analysis of Co-authorship Networks Considering Authors’ Roles in Collaboration: Differences between the Theoretical and Application Areas, *in* ‘Proceedings of 10th International Conference of the International Society for Scientometrics and Informetrics. Vol.2’, pp. 509–516.
- Zanden, B. T. V., Baker, D. & Jin, J. (2004), An Explanation-Based, Visual Debugger for One-way Constraints, *in* ‘Proceedings of the 17th annual ACM symposium on User interface software and technology’, ACM Press, New York, USA, pp. 207–216.

³A representative example is FreeMind (<http://freemind.sourceforge.net/>).

Interactive Optimization in Cooperative Environments

Joelma de Moura Ferreira Hugo A. Dantas do Nascimento
Eduardo Simões de Albuquerque

Instituto de Informática- Federal University of Goiás
Post office box 131-
74001-970-Goiânia, GO, Brazil
Email: {joelma, hadn, eduardo}@inf.ufg.br

Abstract

In the present paper, we introduce a multi-user interactive framework for solving complex optimization problems. The framework, called Co-UserHints, provides a visual computational environment for interaction and visualization. We demonstrate its functionality by presenting a multi-user system for interactive graph drawing.

Keywords: interactive optimization, multi-user framework, user hints, graph visualization.

1 Introduction

The Interactive Optimization research area studies ways of combining automatic methods with human interaction in order to solve combinatorial optimization problems. Many practical optimization processes are dynamic and complex, thus demanding human intervention to insert knowledge into the computer and/or to help a semi-automatic optimization method to produce high quality solutions.

Some approaches for interactive optimization have been developed (Do Nascimento 2003, Anderson, Anderson, Lesh, Marks, Mirtich, Ratajczak & Ryall 2000). However, they focus only on single-user interaction, while many real optimization problems are handled in a cooperative way by a group of people. In general, these optimization problems involve subjective domain knowledge and problem-solving skills that are aggregated by several users.

In this paper, we present a new cooperative and interactive framework, called Co-UserHints, for optimization processes. We also describe a multi-user system for interactive graph drawing based on our approach.

The remainder of the paper is organized as follows: Section 2 gives an overview of the related work. Section 3 introduces our Co-UserHints framework, and Section 4 describes the multi-user graph drawing system. Section 5 concludes with a discussion of the results obtained so far and our future research.

2 Related Work

Recent studies have investigated effective forms of allowing a user to interact with automatic methods. Among such studies, we have been investigating the *User Hints* framework, introduced by Nascimento and Eades (Do Nascimento 2003). In this framework,

an algorithm is responsible for seeking an optimal solution for an optimization problem, while a user helps the algorithm to complete its task by performing interactive actions.

The framework is composed of nine elements, which are described below:

- **User:** he/she is responsible for refining the problem and guiding the method to a high quality solution.
- **Objective function:** this function measures the cost of a solution.
- **Constraints:** these are the conditions imposed to the variables of the problem.
- **Working solution:** it is the solution being refined by the user.
- **Optimization methods:** they are traditional algorithms to solve the optimization problem.
- **Quality function:** this function includes an additional cost for the non-satisfaction of some constraints. This means that a set of constraints can be relaxed.
- **Best solution agent:** measures the quality of the working solution and store the best result found so far.
- **Visualization tool:** it is through the visualization tool that the user receives *feedback* from the optimization process.
- **Visualizations:** they present the working solution and other elements of the optimization process visually.

In the *User Hints* framework, the user can perform some operations that help to include domain knowledge in the processing, to escape from local minimum, and to reduce the solution space to be explored. Those operations are called *hints*. The main types of hints are: adjustment of objectives and constraint of the problem, focus of the optimization method, and manual changes of the working solution. For more details about the *User Hints* framework and its application to combinatorial optimization, see (Do Nascimento 2003).

Unfortunately, the *User Hints* framework considers human-computer interaction for only one user. To our knowledge there is no general interactive approach for combinatorial optimization problems that supports multi-user cooperation. Proposing such an approach is the aim of the present paper.

3 The Cooperative User Hints Framework

Our approach for interactive optimization extends the original *User Hints* framework in order to support multiple co-located and remote users. We call it *Co-operative User Hints framework* (Co-UserHints).

In the *User Hints* framework the user acts in an individual level, only interacting with the system. In the Co-UserHints, it is possible to have two levels of actions: individual and cooperative.

In the individual level, each user works on an independent workspace that implements the complete functionality of the original *User Hints* framework.

In the cooperative level, a general *User Hints* framework aggregates the individual workspaces allowing several users to work together. The users can discuss aspects of the optimization problem and then merge their individual results in order to produce a new solution with better quality.

User actions in the individual and in the cooperative level can be performed simultaneously or alternately, in order to build and/or improve a global solution.

3.1 The Components of the Framework

The Co-UserHints framework has the same elements of the *User Hints* model and adds new resources for allowing multi-user cooperation. Its general structure is shown in Figure 1, and is composed of sixteen elements:

- **Users:** they work cooperatively for obtaining a high quality solution.
- **(Global) objective function:** it defines the cost of a solution (see Section 2).
- **Global constraints:** these are constraints on the variables of the problem applied to all users and optimization methods.
- **User local constraints:** the variables of the problem can be limited by constraints defined by the users individually.
- **User working solution:** this is the solution improved by a user individually.
- **Local best solution:** this is the best solution solution created by a user individually.
- **Global best solution:** it is the current best solution among the solutions produced by all users.
- **Global quality function:** this function defines the cost of a solution globally.
- **User local quality function:** this function measures the cost of a solution locally.
- **Optimization methods:** they are optimization algorithms that can be executed by any user in order to improve his/her working solution.
- **Shared solutions:** these are complete or partial solutions that are saved in a shared area and are available for all users.
- **Local best solution agent:** it is an automatic agent that checks every new working solution produced by a user.
- **Global best solution agent:** it is responsible for saving and updating the global best solution.

- **Integration operators:** these operators are computational tools that allow users to integrate their results by joining parts their solutions and/or merging complete solutions for the optimization problem. They work on the shared area.

- **Visualization tool:** this tool provides visual feedback for the users.

- **Visualizations:** several visualizations are created by the visualization tool for presenting information about the working solution assigned to each user and about the general progress of the optimization process.

Each user has a set of local constraints, a working solution, a local best solution, a local quality function and a best solution agent assigned to his/her workspace.

3.2 Cooperative Hints

We identified the need for having new types of hints for solving optimization processes in a cooperative way. We call them *cooperative hints*. Three types of cooperative hints can be explored:

- **Solution Sharing:** during the optimization process each user can copy his/her working solution, or part of it, to the shared area of the system. Everything saved in this area can be accessed and visualized by all members of the group.
- **Solution Integration:** this hint is similar to Solution Sharing. However, in this new type of hint, every element inserted into the share area is automatically merged with the previous solutions saved there. The merge procedure is performed by the Integration Operators, which join partial solutions or produce a new result based on the best attributes of several distinct solutions.
- **Global Solution Access:** it is responsibility of the Global Best Solution Agent to find the Global Best Solution among all Local Best Solutions produced by the users individually. This task is done automatically by the agent without human interference. The users, however, can access (and recover) the Global Best Solution.

3.3 The Optimization Process

In our new framework the optimization process starts with the generation of an initial solution, which is then replicated for all users, as their local working and local best solutions. Each user is responsible for the improvement of his/her working solution through the use of hints and the execution of optimization methods. No user can modify the solution of another user.

Every change of a working solution forces its related local best solution agent to run. This agent verifies whether the new working solution has a better quality than the local best solution. If this condition is true, then the local best solution is updated and the global best solution agent is notified. The global best solution agent works in a similar fashion to the local best solution agent.

Note that the visualizations are the key elements of the Co-UserHints framework, since they provide useful feedback about the actions performed by all users and how such actions affect the optimization process. Another important issue is that new visualization techniques can be developed for helping

cooperation. Examples of helpful visualizations are drawings that highlight the difference between two solutions, produced by two different users, and histograms (Solution Quality X Time) that describe how the users improved the global best solution over time.

4 A Multiuser Graph Drawing Application

In order to test the Co-UserHints framework, we started the development of a multi-user application for the problem of drawing undirected graphs. Our system, named Co-GDHints, uses multiple pointing devices.

The system has a multi-user graphical interface that allows a group of up to four co-located people to improve graph drawings simultaneously. The main optimization goal is to reduce edge crossings, but improving edge orientation and symmetry is also being incorporated.

The interface is divided into four editing areas, with each user responsible for one of them. There are also two shares areas; one just for sharing partial and complete solutions, and another for saving integrated solutions. All editing areas have a toolbar with options to run the optimization method, to retrieve the best (local and global) solutions, and to insert/retrieve solutions from/to the shared areas. No user has permission to edit the area of another user.

As the application was developed to be executed on an *optimization desk* (Do Nascimento 2003, Anderson et al. 2000), the orientation of the editing areas and the mouse pointers are set up according to the users' positions around the environment. Figure 2 presents the Co-GDHints graphical interface.

We implemented five types of hints into the system:

- manually changing a drawing through selection and movements of vertices;
- focusing the execution of an optimization method¹;
- forcing the best solution agents to save a particular solution or retrieving the local or the global best solution at any time;
- inserting/removing a solution to/from the shared areas;
- integrating solutions through an integration operator that tries to reduce edge crossings.

4.1 Tests

We performed preliminary tests with the Co-GDHints system to check whether more than one user, cooperating in a graph drawing problem along with optimization methods, could improve a given initial solution.

Three groups of two users took part of the experiments. Each group had to perform two tests, which consisted of improving an initial drawing of a graph. A different graph was used in each test.

In the first test every user had his/her own workspace, and could see the other users drawings. The users could share their drawings, but the integration operator was not available. In the second test all users worked on the same solution in a single workspace. In both cases, the users had their own pointing device.

From the experiments we observed that:

- Users working in individual workspaces tended to be competitive rather than cooperative. Still, as a user could see the others solution, it was difficult to identify similarities because the drawing looked up side down. This increased further the competition. In addition, the system had no features to force the users to cooperate.
- In the test where the users worked on the same workspace, they naturally divided the drawing in small regions and each one worked on his/her own area. Although the division was not enforced by the system, the framework provided satisfactory features for cooperation.

In order to provide a fully multi-user approach, we also noticed that the system has to support both individual and the group viewing of a solution. Therefore, we have incorporated two new visualization technics into the Co-GDHints system to improve the users interaction experience. These techniques are based on our observation during the tests:

- Graph drawing flip: to support face-to-face cooperation we consider the possibility of flipping a workspace so that its drawing looks oriented to the other users.
- Graph drawing overlaying: before the user recovers a shared or integrated solution, he/she can visualize how different it is from his/her working solution by highlighting vertices that will be modified, and showing their current and new positions.

5 Conclusion

In this paper, we presented a new framework, called Co-UserHints, for interactive optimization. The framework extends the original *User Hints* approach in order to support multi-user cooperation.

In addition to the cooperative framework, we also started the implementation of a system for solving a graph drawing optimization problem. The system is not fully functional yet, but we have completed a reduced number of experiments that show promising results. The initial tests indicated that the framework offers freedom for the users to interact with an optimization process, and provides space for group discussion. In addition, users working in a cooperative environment may obtain better results. We are now completing the implementation of the system, and will perform more intensive controlled experiments of its cooperative resources. The new results of this research will be available at <http://www.inf.ufg.br/funcomp>.

References

- Anderson, D., Anderson, E., Lesh, N., Marks, J., Mirtich, B., Ratajczak, D. & Ryall, K. (2000), 'Human-guided simple search', *AAAI'00: Seventeenth National Conference on Artificial Intelligence* pp. 209–216.
- Di Battista, D., Eades, P., Tamassia, R. & Tollis, I. G. (1999), *Graph Drawing: Algorithms for the Visualization of Graphs*, Prentice Hall.
- Do Nascimento, H. A. D. (2003), *User Hints for Optimization Processes*, PhD thesis, University of Sydney.

¹The method is a *Spring* algorithm, which defines a force system where vertices repel themselves while edges attract adjacent vertices. (Di Battista, Eades, Tamassia & Tollis 1999)

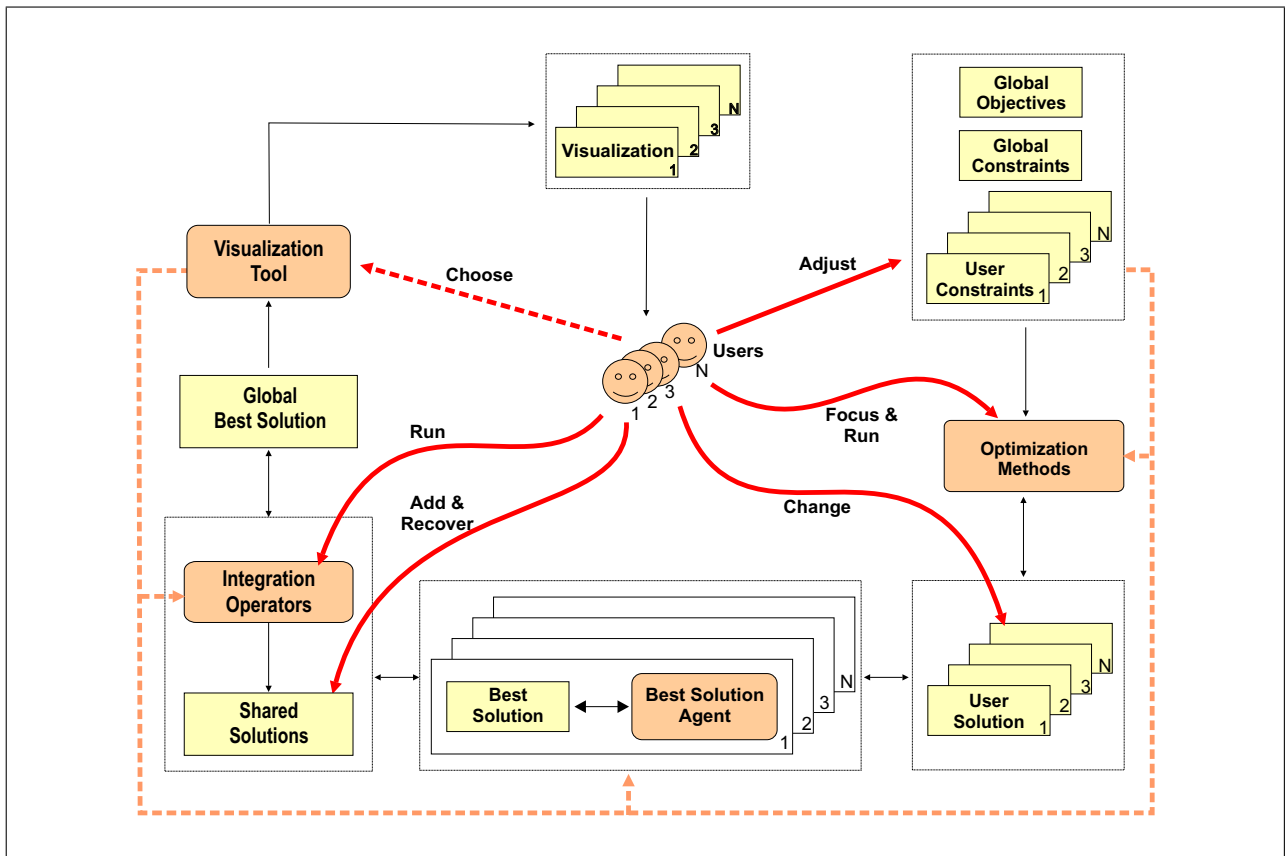


Figure 1: The Co-UserHints framework

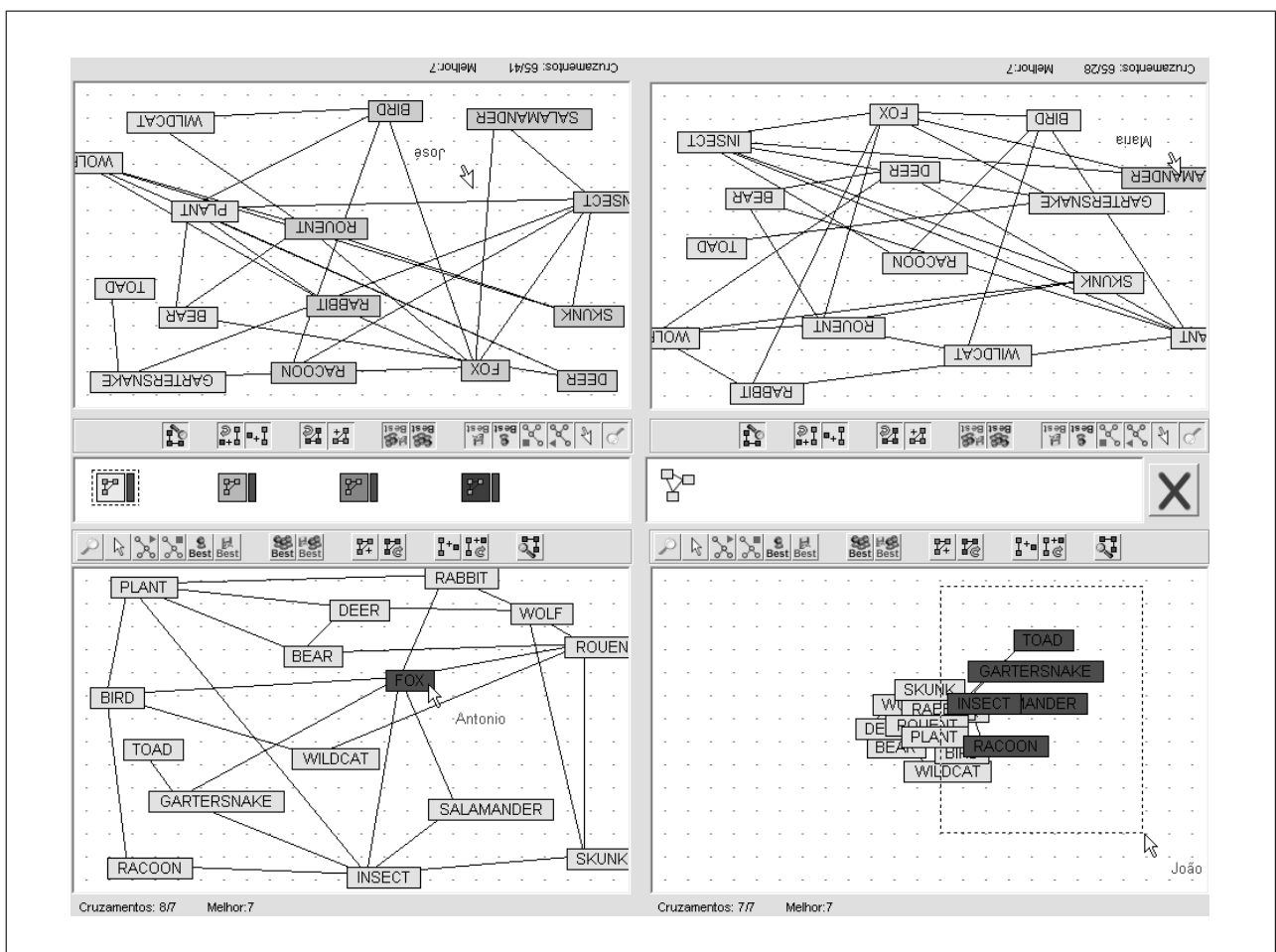


Figure 2: The Co-GDHints graphical user interface

Enhancing the Visualization Process with Principal Component Analysis to Support the Exploration of Trends

Wolfgang Müller¹, Thomas Nocke², and Heidrun Schumann²

1) Media Education and Visualization Group
University of Education Weingarten
D-88250 Weingarten, Germany
muellerw@ph-weingarten.de

2) Department of Computer Science
University of Rostock
D-18051 Rostock, Germany
(nocke/schumann)@informatik.uni-rostock.de

Abstract

This paper describes the integration of the Principal Component Analysis into the Visualization Process. Although, the combination of Principal Component Analysis (PCA) and visual methods is a common approach to the analysis of high-dimensional datasets, it is mostly limited to a pure preprocessing step for dimension reduction. In this paper we will discuss, how PCA results can be used to control all steps of the visualization pipeline to generate more effective visual representations, and thus, a higher degree of understanding of the PCA values as well as of original data.

Keywords: Information Visualization, Visualization process, Principal Component Analysis.

1 Introduction

The amount of data in public and corporate databases is growing rapidly day by day, and databases with gigabytes or terabytes of data are no longer unusual. For long the visual representation of such data was proven to be of high value in exploring this data, detecting correlations as well as trends and outliers. For doing so, the high dimensional data must be somehow converted to low dimensional geometry for display. One well-established method in this context is the Principal Component Analysis (PCA).

PCA is a transformation technique that aims to identify main factors accounting for variances in the data. Identifying these factors leads to a more compressed description of correlations in the data and, thus, for a better understanding of the underlying features. Therefore, it is a powerful approach to extract general trends in a data set. Moreover, since PCA provides principal components ordered by their significance, it also offers an excellent basis for data dimension reduction in case of multidimensional data. This can be achieved by

omitting less relevant trends in the data set and concentrating on main principal components.

The combination of PCA and visual methods is a common approach to the analysis of high-dimensional datasets in many application domains (see e.g. Yang 2003, Komura 2004, Landgrebe 2002, Santos and Brodlie 2004). However, nearly all of these approaches use the PCA as a kind of preprocessing computing significant trends in the data and visualize them. This may particularly cause 2 problems:

1. The PCA may generate principal components which are difficult to interpret and which sometimes even appear orthogonal to what intuitively seems to be the dominant trend in the data. A subsequent rotation of principal components is therefore sometimes appropriate (see Jolliffe 1986 for more details). Also, the necessary assumption of global linearity of the data oftentimes proves inadequate. (Müller and Alexa, 2004) therefore suggest an interactive, visually-guided improvement of the results of a PCA.
2. The coordinate transformation into principal component space involved with the PCA makes it difficult to relate identified trends to original variables. In general, it is often not easy to explain the variation of data with respect to a principal component.

Our aim is to reduce these problems by a stronger integration of the PCA into the visualization process. That means, we want to discuss, how the PCA can be used to support the different steps of the visualization pipeline to generate more effective visual representations, and thus, a higher degree of understanding of the PCA values as well as of original data.

The remainder of this paper is structured as follows. Section 2 describes the background including the fundamentals of the PCA as well as of the visualization process. In section 3 we look at each step of the visualization process, and discuss the main aspects of combining these steps with PCA. Moreover, we demonstrate the usefulness of our approach by different examples. We conclude with a summary and an outlook on further work in section 4.

Copyright © 2006, Australian Computer Society, Inc. This paper appeared at the Asia-Pacific Symposium on Information Visualization (APVIS 2006), Tokyo, Japan, February 2006. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 60. K. Misue, K. Sugiyama and J. Tanaka, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

2 Background

2.1 Fundamentals of PCA

The Principal Component Analysis is a technique commonly applied to *reduce* the number of variables and to *detect structure* in the relationships between variables in multi-dimensional data sets. In the PCA the extraction of principal components amounts to a *variance maximizing rotation* of the original variable space. That is, the PCA provides a transformation of the original data space in such a way that the first coordinate of the resulting principle component space would resemble most of the variance in the data set, the second variable the most of the remaining variance, and so on.

More formally, we assume that our original data set is given in matrix notation

$$Y = \begin{pmatrix} y_{11} & y_{21} & \dots & y_{d1} \\ y_{12} & y_{22} & \dots & y_{d2} \\ \vdots & \vdots & \ddots & \vdots \\ y_{1n} & y_{2n} & \dots & y_{dn} \end{pmatrix} \quad (1)$$

where each column is associated with a single variable, thus representing an individual dimension of the data set. Consequently, d describes the number of dimensions. Each row y_i , $i = 1 \dots n$ represents a different case in the data set. For instance, each row may stand for a different time step. In the following, we assume a normalized and mean-centered data matrix Y . There are several ways to determine the principal components for Y . An often applied approach is to apply a *Singular Value Decomposition* (SVD). The SVD factors Y directly into

$$Y = C\Lambda^2 W \quad (2)$$

where W represents the loadings that is the basis vectors w_i of the transformed vector space. Λ^2 represents a diagonal matrix with the significance of each principal component corresponding to the amount of variance described. Finally, C represents the scores c_i , that is the coordinates of the data elements in the transformed vector space.

Once this information is obtained it can be exploited in several ways. First of all, the principal directions in W may be interpreted as prominent trends in the data. Also, the loadings in W provide information on the correlation of different variables with these trends. For data reduction, on the other hand, we can simply neglect less significant axes in the principal component space and reduce the analysis to the remaining dimensions.

2.2 The Visualization Process

Santos and Brodlie (2004) introduce an extended data flow model to accommodate the visualization process (see figure 1). In contrast to the common visualization pipeline they replace the preprocessing step by two separate components distinguishing between data analysis and filtering. The data analysis component is computer-

centered and provides some computational methods to analyze or enrich the data. Typical functions for this purpose are interpolation functions, PCA or Multidimensional Scaling (Santos and Brodlie, 2004). The filtering-component is user-centered, and provides different functionality for selecting data of interest. Thus, the portion of data to be visualized can be extracted. This is done in two levels. First, an n -dimensional window with upper and lower bounds is specified within the data domain. Moreover, a focus point within these bounds is set. In doing so, the representation within the window can be controlled, e.g. by accenting data of interest. Although (Santos and Brodlie, 2004) assume a fixed order of these two components (data analysis first, then interactive data filtering, whereby the data selection process can be repeated more than once), it can be expedient to swap these steps. For example, it can be useful to apply a PCA after a data selection step.

If a PCA is applied to all variables of a data set, the interpretation of the visualized PCA-components could be difficult. In this case, natural references such as time in the case of temporal data, or spatial dimensions in the case of spatial data, are also a target of the coordinate transformation into principal component space. Consequently, they are no longer available as a point of reference to relate to. Therefore, it can make sense, to select variables for PCA-processing, while leaving others as frame of reference.

The next step in the visualization pipeline is the mapping of selected and computed data to be visualized onto a geometrical representation considering the given focus point. This is the most crucial step during the visualization process deciding about the expressiveness and effectiveness of a visual representation. Different approaches were developed to support the mapping step, and to generate an appropriate visualization design (see e.g. Mackinlay, 1986; Senay, and Ignatius, 1994; Roth et al 96; Fujishiro et al, 2000; Zhou+ 2002; Almar and Stasko, 2004; Tang et al., 2004). However, none of these approaches apply PCA results to control the mapping. We will discuss in section 3.3 how PCA results (loadings from the matrix W and eigenvalues from the matrix Λ^2) can be used to define and parameterize the mapping, and in doing so, creating intuitive visual representations for a better insight of the data.

The last step of the visualization process is the rendering step. Here, image data are created for display on a monitor. Usually, this step is not in focus of recent research. However, it is worth to investigate, how a combined handling of data and PCA-values can improve the display. For example, showing trends in the data by rendering principal components requires a mental mapping of axes of the principal component space onto data variables from the user. Therefore, an important step in the application of the PCA is the adequate labeling of the resulting axes. This includes the determination of adequate axes labels describing the background of a principal component, and providing suitable axis value marks to provide an adequate relationship to the original data space. To achieve this, it is necessary to identify in how far certain variables correspond to principal components. Nearly all PCA visualizations forbear from

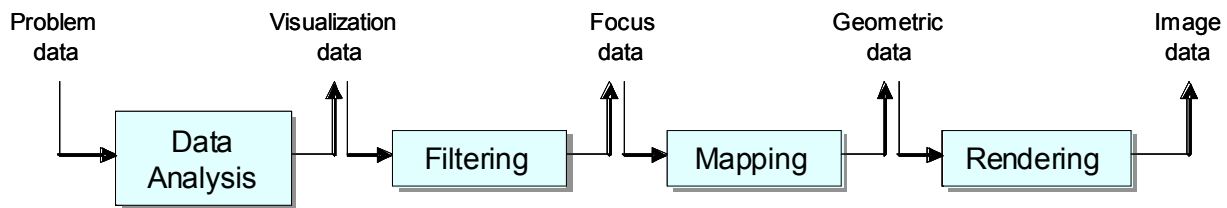


Figure 1: The visualization process, adapted from (Santos and Brodlie, 2004).

doing that. We will get back to this problem in section 3.4.

After discussing the separate steps of the visualization pipeline, finally we want to consider the process as a whole. The visual exploration of data sets has a high degree of interactivity. Thus, the visualization process can not be understood as one pipelining queue. Rather, different steps are repeated more than once to create different visual representations supporting different exploration tasks. For this, (Shneiderman, 1996) introduced the so-called *Visualization Mantra* “Overview first, Zoom and Filter, then Detail on Demand”. Hereby, the fundamental procedure of visually exploring data is defined.

Starting with a general overview image representing general trends of a data set, zooming is applied to focus on regions of interest. Moreover, information hiding is provided to filter out non-relevant data. Finally, details on demand have to be added by request of a user.

PCA can provide useful information to support the *Visualization Mantra*. Since PCA provides principal components ordered by their significance, it offers an excellent basis for data dimension reduction, and thus for creating overview images. Moreover, Zooming can be controlled by PCA values, e.g. to set the focus on interesting trends or even on outliers. Similarly, certain trends or outliers can be filtered out depending on the exploration task. Finally it must be possible to add details, e.g. adequate annotations.

In sum, we can notice that a stronger interrelation between PCA and the visualization process can be very useful, and therefore it would be worth for further investigations. We will discuss this topic in more detail in the next section.

3 Integration of PCA into the Visualization Process

3.1 Data analysis and PCA

The first step of the visualization pipeline is the data analysis step. As already mentioned, PCA is a common procedure in this context, especially to achieve dimension reduction. The PCA provides principal components ordered by their significance, and thus less relevant trends in the data set can be omitted concentrating on main principal components only. Moreover, we want to provide a better insight of the PCA-results, and in doing so, a better insight of the features of the original data, by a stronger coupling of PCA and visual methods. Figure 2 shows our visual representation of the loadings of the matrix W (see formula 2) for a demographical data set.

High loadings in a column act as an indicator for a high relevance of the associated axis in the PCA-space representing, and thus, as a relevant trend of the given data set. The values in a row show the influence of the different variables on the trends represented by the PCA-axes.

More precisely, in Figure 2 we see a major trend represented by the principal component 1 (PC1). All the positive loadings (in blue color) in PC1 indicate a direct proportional relationship for the variables *literacy*, *infant mortality (Babymort)*, ..., and *life expectation*. *Population* and *density* do not influence this trend. The second trend (PC2) is constituted by *gross domestic product (GDP)* and *life expectation* and is indirect proportional to *infant mortality*, *death rate* and *birthrate*. This example illustrates how the user gets fast insight into the main trends in a high dimensional data set. However, he gains as well information about more hidden trends. Instance for such an “outlier trend” are the oppositional loadings of Life expectation of men and women in the principal component PC9.

A better understanding of the insights of the PCA can be used to control the following steps of the visualization pipeline (data selection, mapping and rendering) to generate more expressive visual representations. Furthermore, this provides a powerful basis to create overview images as well as detailed displays. To explore relevant trends for example, the user can switch to further views e.g. line charts or scatter plots (see section 3.3).

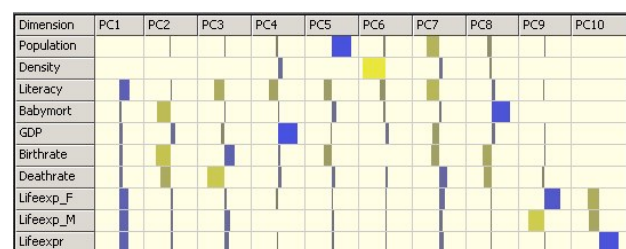


Figure 2: Visualization of the Loadings matrix W for a demographical data set

3.2 Filtering and PCA

The second step of the visualization pipeline is the filtering process. Here, variables of interest are selected for further procedure by the user. Thus, this process is highly interactive and user-centered. With regard to the PCA we distinguish two strategies:

1. Filtering before executing PCA,
2. Filtering after executing PCA.

Filtering before executing the PCA includes the selection of variables as well as of data objects (defined by columns resp. rows in the input matrix Y) for PCA processing. Usually, natural references such as time in the case of temporal data, or spatial dimensions in the case of spatial data, are excluded from PCA to leave them as a frame of reference. Moreover, special variables or data objects of interest can be selected for PCA-processing. Contrarily, high correlated variables or outliers (variables as well as data objects) can be excluded from PCA. For supporting the user to include or exclude variables and data objects for PCA, a visual representation of the original data can be very helpful.

Figure 3 illustrates how the user gets hints to filter out certain data objects (table rows) or variables (table columns) by example of a data table visualization of the demographic data set from figure 2. For instance, the user can examine high outlier values of *population* and *density* (the first two columns)

To exclude these outliers from PCA, the referring data objects respectively the referring variables have to be deselected. Furthermore, this data set includes three *life expression* variables (women, men, all, the last three columns), which are expectedly highly dependent. This fact will strengthen certain trends calculated by the PCA (may be more then the original data express). It is up to the user to exclude such dependent variables from the PCA.

Filtering after executing the PCA has high potency to guide users through a variety of trends in an overview and detail manner. Using a tabular loadings visualization (see figure 2) the user gains an overview about the most important trends and the involved variables. Based on such a loadings table he can select principal components, original variables or combinations of both. This is a flexible approach to analyze the trends and the contributions of certain variables to these trends by appropriate visual representations in more detail (see section 3.3).

Furthermore, PCA does not only calculate the trends, but even orders them by their significance (PC1 for highest to

PCn for lowest significance). Additionally, the normalized PCA loadings (by the eigenvalues from the matrix Λ^2) can be used as a quantitative measure to express the significance information. This information is very helpful to guide users through the “trend space”. That means, the calculated measures can be used to visually direct the user’s attention to the most relevant trends.

Figure 4 shows a *Table Visualization* with normalized loadings. The first trend (PC1) - primarily established by *literacy* and *life expression* - has a high significance for almost all variables. Normalized loading values for the next relevant PCs are strongly decreasing. Thus, the user can select low dimensional subspace of the original space for visualization purposes concentrating on the exploration of main trends. In this case, for instance, a simple line chart can represent the first principal component.

3.3 Mapping and PCA

Mapping is the most crucial step in the visually guided process of analyzing data. It involves both the selection of an appropriate visualization technique and the mapping of data values onto expressive and effective visual attributes, such as position, marker sizes, and marker color.

When applying a PCA we enrich our original data by a principal component representation making additional aspects of the data explicit and, thus, available in the mapping process. Consequently, the gained information may influence our decisions on what aspects of the data to map onto visual attributes (i.e. PC values vs. original data).

Visualizing the data from principal component space instead of mapping the original data follows the main objective of the PCA to reduce the number of dimensions of the original data space and to present prominent trends in the data only. We will demonstrate this strategy by an example. Figure 5 displays a reduced version of the countries data set in the original data space in terms of a Scatterplot Matrix.

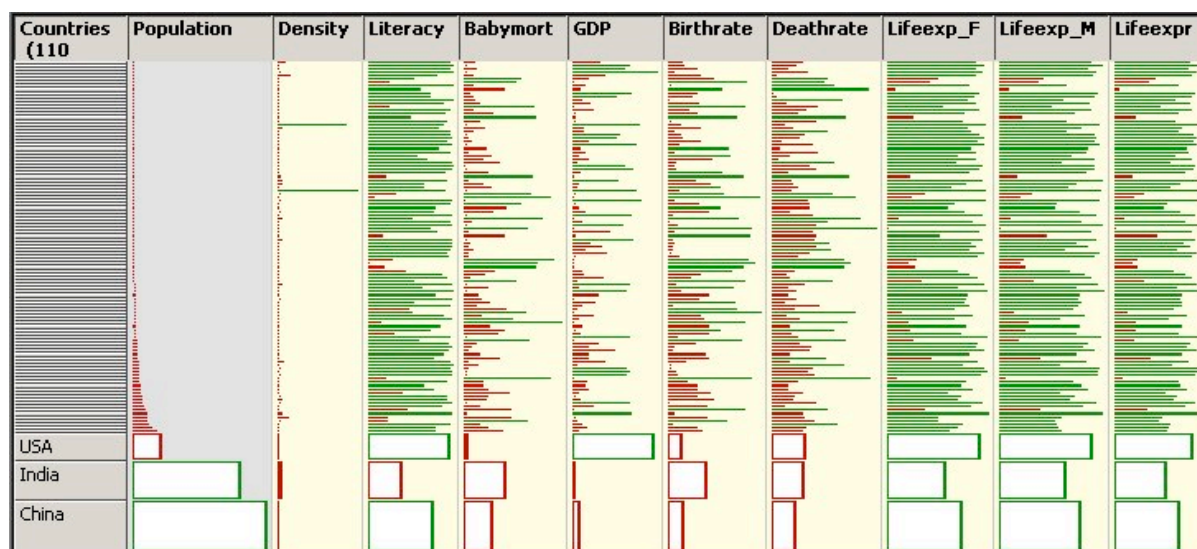


Figure 3: Data table visualization of the demographic data set (with table lens, see Kreuseler (2002))

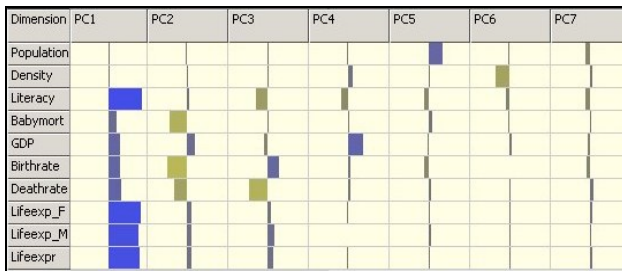


Figure 4: Normalized loadings visualization of the demographic data set

We had to reduce the original data set to generate this intuitive representation. The visualization of the whole data set (in this case 10 variables) or of even higher dimensional data sets will generate rather complex visual representations with many recurring dependencies and thus, quickly exhaust users. A big drawback of manually filtering variables is the loss of possibly relevant trends in the data. The question is how to decide which original variables to show and which not, or in the case of the demographic data set: why do we display the variable *literacy* and not the variable *population*. To answer this question, PCA can be a good guide to select (depending) variables keeping the remaining trends in mind (see section 3.2). For instance, in figure 5, the selection of variables is based on the main PCs (see figure 4).

In contrast to the PCA controlled visualization of variables in the original space, we may decide to utilize the data in principal component space for a direct visualization. That is we utilize the PCA score matrix data and map those transformed coordinates directly to visual attributes.

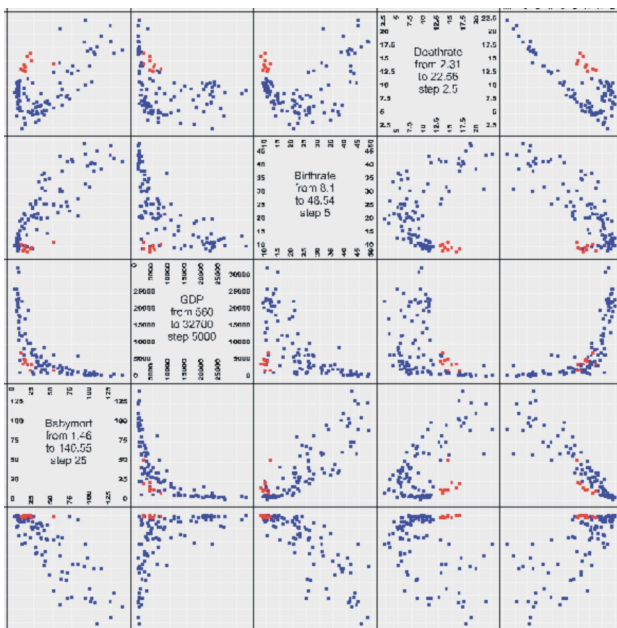


Figure 5: Scatterplot matrix of a subspace of the demographic data set; countries with high death rate, low birth rate, high literacy and high life expectation highlighted (all are former east European countries)

In the case of the dataset discussed before, for instance, a scatter plot matrix of the first three principal components presents the most noticeable trends in the data directly (see figure 6). Interestingly, in our example there is a strong correlation between the trends represented by the first two principal components. This is usually a sign for a more complex correlation between the corresponding variables that cannot be described easily in terms of a variance in a linear direction.

As an alternative to these straightforward approaches we suggest to apply a table visualization of the scores ordered by the values of the first PC (see figure 7). The advantage of this kind of visualization is that users can directly identify data objects with certain PC values, and thus, get deeper inside about the meaning and significance of the trends. For example, we can see that the last three rows (representing the countries *Japan*, *Norway* and *Luxembourg*) are prominent representatives of the first two trends.

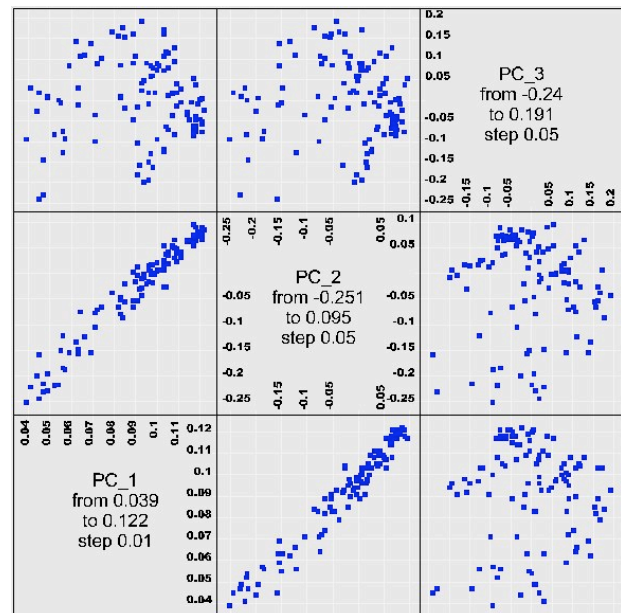


Figure 6: Scatter plot matrix with the first three principal components

Besides presenting the most significant principal components it may also be appropriate to visualize the least significant ones, for instance to identify outliers or erroneous data.

In general, the presentation of principal component scores results in an abstraction of the original variables and in abstract axes. Both make it difficult to relate the visible patterns to trends with respect to the original variables.

This represents a specific form of a correspondence problem. Consequently, graphing data in principal component space requests for an adequate labeling of the resulting PC axes. We will come back to this issue in chapter 3.4.

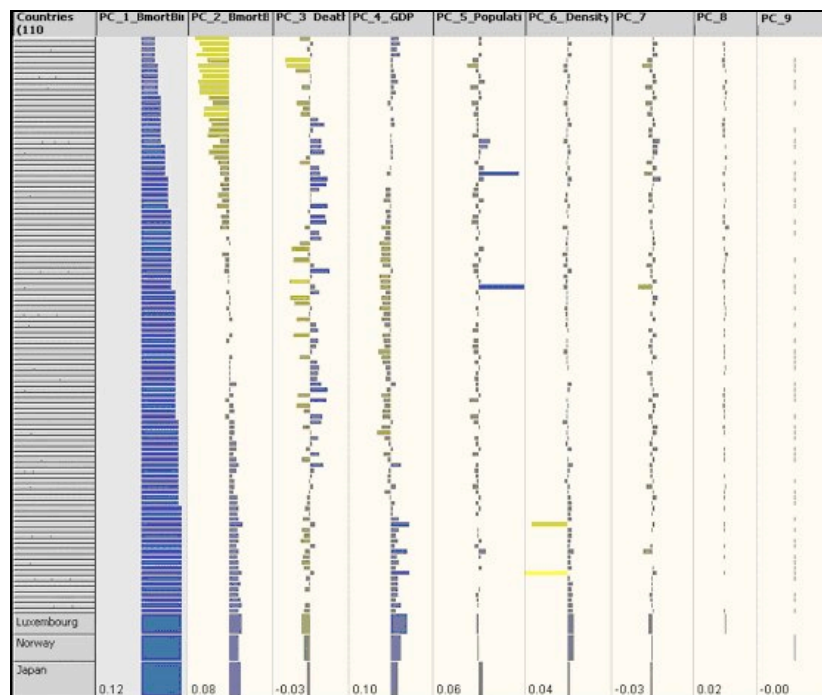


Figure 7: Table visualization of the normalized PC scores of the demographic data set ordered by the first PC

A useful aid for the interpretation of original and PCA data provides the possibility to combine both in a single plot. For instance, in the example of the demographic data set a similar scatter plot relating *literacy* and *life expectation* to the first principal component can clarify the correspondence of the first identified trend to these variables. Figure 8 shows a strong correlation between these three variables, whereby the correlation between *literacy* and PC1 is more scattered than the correlation between PC1 and *life expectation*.

For a combined representation of PCA values and original data we have to decide which variables to present directly. A common approach in this context is to distinguish between independent and dependent variables. Usually, dependent variables are processed by PCA, whereas independent are displayed in the original data space. In our case it can make sense to show PCA values of the demographic data set in their spatial frame of reference (see figure 9).

Another approach is to combine PCA loadings instead of the scores to the original data. Different from the score data the loadings represent the prominent trends as identified by the PCA directly. We may utilize this information to depict the identified trends in the context of the original data (Müller and Alexa 2004). Figure 10 depicts an example for such an approach.

3.4 Rendering and PCA

So far we discussed approaches based on a direct visualization of original data and/or principal components and their factors. We mainly concentrated on combining original and PCA data in visualizations and on performing a direct mapping of PCA data to visual attributes. Another promising approach is to utilize the trend information made explicit by the PCA for tuning

the rendering process. We denote this *implicit visualization* of PCA values.

An implicit visualization may especially aid in the filtering and the mapping stage. For instance, variables can be mapped automatically onto visual attributes based on the correspondence of the variable to a major trend as identified by the PCA and the effectiveness of certain visual variables (e.g. position vs. color, see Mackinlay 1986). This opens up a new field of automatically controlling the mapping of data variables based on detected trends in the data.

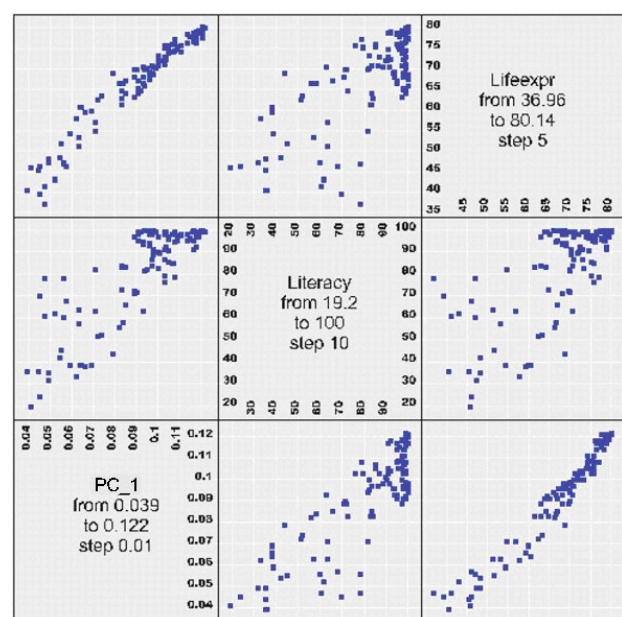


Figure 8: Scatter plot relating the first PC to the variables literacy and life expectation

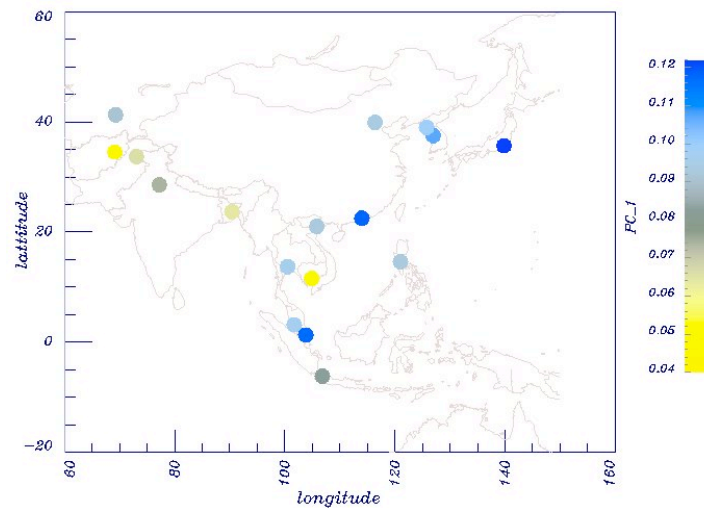


Figure 9: Spatial visualization of the first PC of the demographic data set (cutout of Asian countries; PC1 is represented by a colored circle at the capital location)

3.4.1 Focusing and arrangement

Another application of principal component scores is their utilization for an indirect visualization, for instance for brushing & linking. In figure 11 high values with respect to the most prominent data trend are selected (left), and highlighted in a scatter plot visualization (right) representing three original data variables (two axes and color). Besides a simple highlighting, further details can be presented for the selected data elements, e.g. annotations or semantic zooming. Brushing and linking of PC score values is not limited to standard graphs but can also be utilized in the context of other information visualization techniques for multi-dimensional data. Figure 12 depicts an example for focusing on information

in a table view representation by applying a table lens. Data objects corresponding to high trend values are rendered with more detail (by larger row space).

The additional information available in terms of the PCA scores can be used to steer the rendering process either manually or automatically by setting for instance the focus on most prominent trends or outliers automatically in an initial view (see e.g. figures 11 and 12).

Besides changing the focus and highlighting certain objects, PCA can be applied to automatically arrange original variables or data objects as well. Figure 13 depicts the table visualization from figure 3 reordered by the first PC.

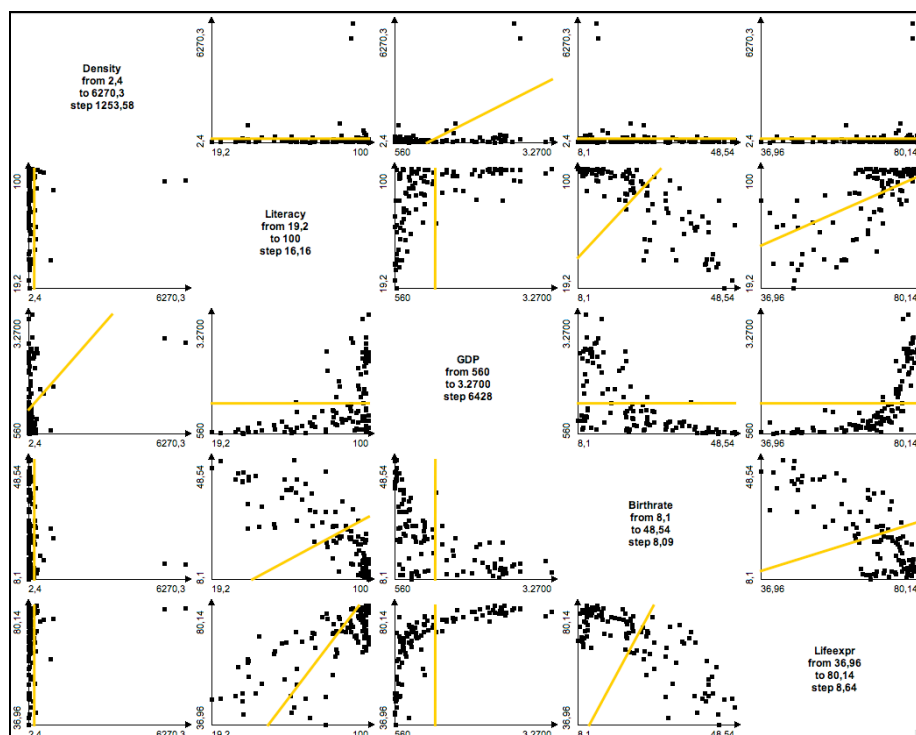


Figure 10: Scatter plot matrix of the original demographic data set with PC1 directions

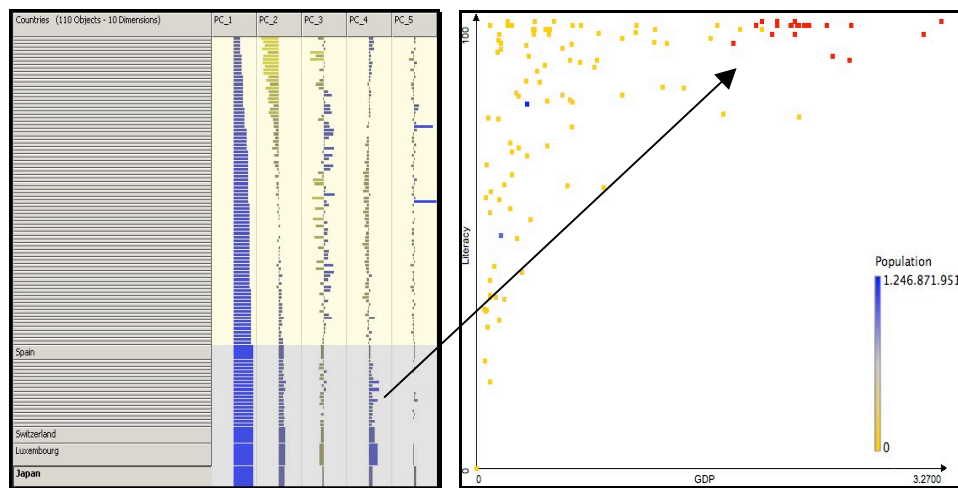


Figure 11: Example of Brushing & Linking utilizing PCA score data. Selecting high PC score values in the table view allows to link extreme values with respect to a prominent trend in the data with original data (scatter plot visualization of the variables literacy (y-axis), GDP (x-axis) and population (color))

3.4.2 Annotation

As already discussed, a general problem in the visualization of principal components is to relate the identified trends to the original variables (see section 3.3). The principal component space represents an abstraction of the original data space and its basis vectors contain usually aspects of several data variables.

As one solution to this problem, we propose combined visualizations of original variables and PC (figure 8) and implicit visualization of PCs in the original space (figure

11 and 13) to enable users to interactively approach to the meaning of the trends calculated by the PCA, and thus, improve the understanding of the rather abstract PCs.

A second solution to the mentioned problem is to use the information given by the significances (eigenvalues) and the loadings to encode the meaning of the PC more explicitly in the visualization. Challenge in this context is to enrich the rather abstract visualizations in the PC space, especially by generating meaningful axes labels.

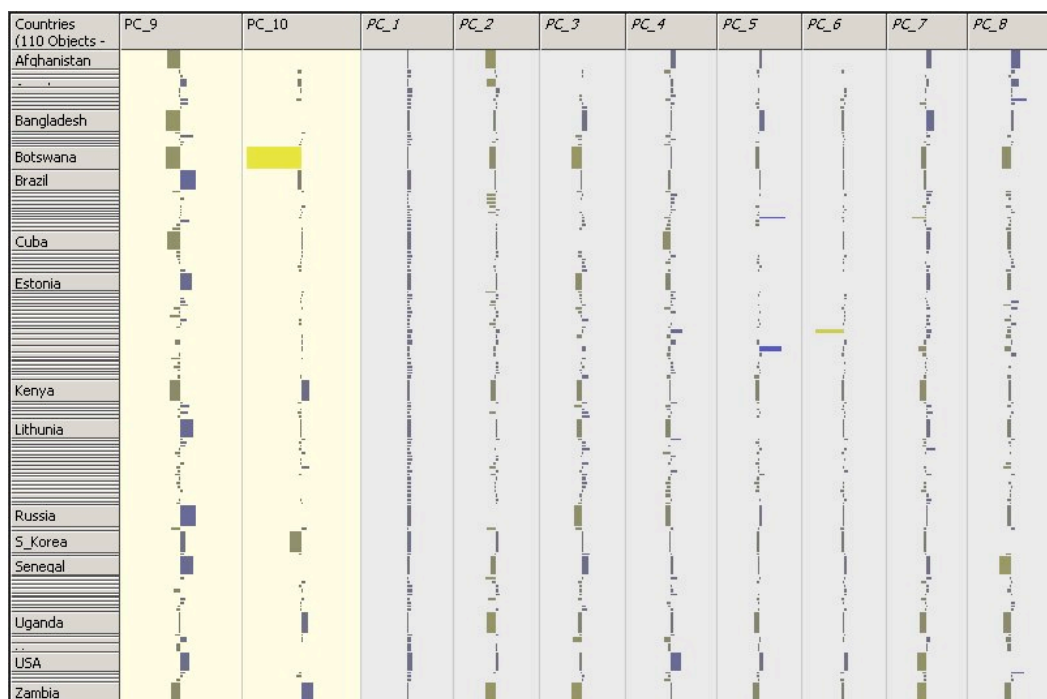


Figure 12: Table Visualization with adapted table lens function to focus on countries with extreme score values in the trends PC_9 and PC_10

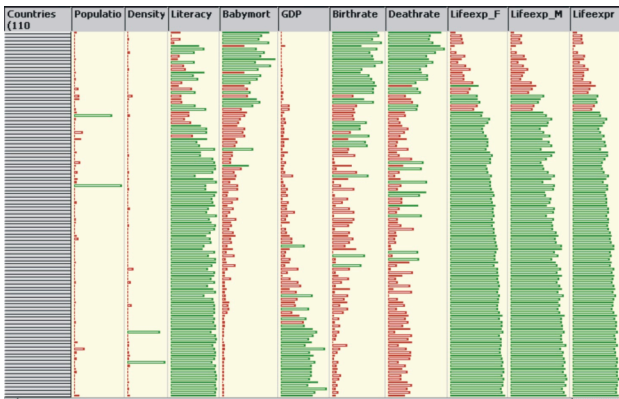


Figure 13: Data table visualization of the demographic data set, reordered by first PC (with table lens, see Kreuseler, 2002)

Figure 14 displays a first solution to this problem, enriching PC axes labels by the variables they mainly represent (applying a threshold on normalized loading values), ordered by their significance. However, determining when a trend based on a principal component corresponds to a variable or not is not always easy. (Jolliffe 1986) recommends to utilize the PCA loadings and to accept a correspondence in case of 70% accordance of original variable and principal component. Unfortunately, we often may have the situation that a trend corresponds to a large number of variables which leads to long label lists that dominate the visual representation (PC1 label in figure 14) or stay hidden because of space restrictions (see figure 7). Therefore, it is useful to let the user decide and to provide adequate settings for this purpose. For tuning the labeling parameters he is supported by our proposed visual representations for PCA results and original data.

Besides understanding the PCA axes, a further challenge is to annotate the meaning of the score values in the visualization to enable users to estimate which score value corresponds to which original variable value.

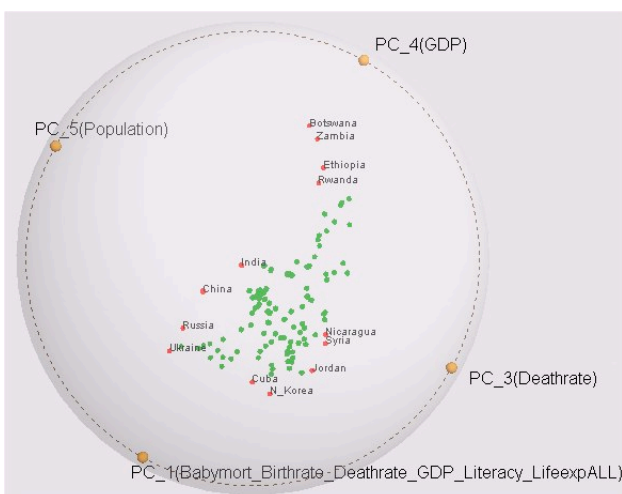


Figure 14: ShapeVis visualization of data objects in PC space using enriched labeling with data variable names (mapping objects to 2D locations using a spring model, see Theisel and Kreuseler, 2002)

As a first solution, we tested annotating minimum and maximum score values at the axes with corresponding original values. Unfortunately, this kind of annotation even extends the long list of original variable names. An alternative approach is to render a brief and abbreviated labeling and to provide additional information on request via brushing, e.g. depicting on PCA axes resp. data objects using the mouse. However, abbreviated labels are usually application dependent and therefore difficult to generate automatically. To conclude, further research needs to be done to adequately enrich visualizations in PC space by appropriate annotations.

4 Conclusions

In this paper, we systematically discussed how to combine information visualization and PCA in the different stages of the visualization process. As a result, we conclude that the enrichment of the original data by PCA generated data and the application of these PC values in the visualization process is a strong support for users studying trends. A variety of representations explicitly and/or implicitly visualizing PC values (scores, loadings eigenvalues) and original data values have been developed and presented. Table 1 demonstrates the tasks performable with these representations on the basis of the kind of PC values applied.

However, there are still challenges for future work. First of all, we want to supply the proposed PCA based visualizations to non-statistic expert users and evaluate them. Finally, further research needs to be done to improve the understanding of trends generated by the PC, for instance by improved annotation strategies for PC axes.

References

- Almar, R.; and Stasko, J. (2004): A Knowledge Task-Based Framework for Design and Evaluation of Information Visualizations. Proc. IEEE InfoVis'04, Austin, USA, 143-149, IEEE Computer Society Press
- Fujishiro, I.; Ichikawa, Y.; Furuhashi, R.; and Takeshima, Y. (2000): GADGET / IV : A Taxonomic Approach to Semi-Automatic Design of Information Visualization Applications Using Modular Visualization Environments. Proc. IEEE InfoVis'00, Salt Lake City, USA, 77-83, IEEE Computer Society Press
- Jolliffe, I. T. (1986): Principal Component Analysis. Series in Statistics. Springer.
- Komura, D., Nakamura, H., Tsutsumi, S., Aburatani, H. and Ihara, S. (2004): Multidimensional support vector machines for visualization of gene expression data. Proc. ACM symposium on Applied computing, Nicosia, Cyprus, 175 – 179.
- Kreuseler, M.; Schumann, H. (2002): A Flexible Approach for Visual Data Mining. IEEE Transactions on Visualization and Computer Graphics **8**, S: 39-5, IEEE Computer Society Press.
- Landgrebe, J., Wurst, W. and Welzl, G. (2002): Permutation-validated principal components analysis of microarray data. Genome Biology, **3**: research 0019.1-0019.11.

- Mackinlay, J. (1986): Automating the Design of Graphical Presentations of Relational Information. Transactions on Graphics, **5**: 110-141, ACM Press.
- Müller, W. and Alexa, M. (2004): Visual Component Analysis. Proc. of Joint IEEE/EG Symposium on Visualization, VisSym'04, Konstanz, Germany, 129-136
- Roth, S.F. et al (1996): Visage: A User Interface Environment for Exploring Information. Proc. IEEE InfoVis'96, San Francisco, USA, 3-12, IEEE Computer Society Press
- dos Santos, S., and Brodlié, K. (2004): Gaining understanding of multivariate and multidimensional Data through Visualization. Computers & Graphics, **28**: 311-325, Elsevier.
- Senay, H.; and Ignatius, E.(1994): A Knowledge-based System for Visualization Design. IEEE CG&A; **14**: 36-47.
- Shneiderman, B. (1996): The Eyes have it: A task by data type taxonomy of information visualizations. Proc. IEEE Symposium on Visual Languages'96, 336-343 IEEE Computer Society Press.
- Tang, D.; Stolte, C.; and Bosch, R. (2004): Design Choices when Architecting Visualizations. Information Visualization, **3**: 65 – 80, Palgrave Macmillan Ltd.
- Theisel, H.; and Kreuseler, M. (1998): An Enhanced Spring Model for Information Visualization. Computer Graphics Forum, vol. 17, no. 3, (Proceedings Eurographics '98)
- Yang J., Ward M. O., Rundensteiner, E. A., and Huang, S. (2003): Visual hierarchical dimension reduction for exploration of high dimensional datasets. Proc. of Joint IEEE/EG Symposium on Visualization, VisSym'03, Grenoble, France, 19-28.
- Zhou, M. X.; Chen, M.; and Feng, Y. (2002): Building a Visual Database for Example-based Graphics Generation. Proc. IEEE InfoVis'02, Boston, USA, 23-32, IEEE Computer Society Press.

| | PC scores | PC loadings | PC eigenvalues | Original data |
|-----------------------|--|---|--|---|
| PC scores | Analyze trends in the PC space and the relationships of main trends (figures 6, 11 left, 12, 14) Interactively reorder PCs (7 and 11 left) Focus on data objects strongly representing trends (figures 7, 11 left, 12, 14) | Annotate PC axes (figures 7 and 14) | Annotate PC axes (figures 7, 14) Focus on major trends (figure 7) | Get deeper insight into meaning of PC axes (figures 8, 9, 11 right) Analyze trends in observation space (figure 9, 11 right) Focus and/or order on data objects in original space (figure 13) |
| PC loadings | / | Get a compact overview about strength and relationship of the main trends (figures 2, 4) | Focus on major trends (figure 4) | Get overview and deeper insight into meaning of PC axes (figures 2, 4) Compare dependencies in original data with PC axes directions (figure 10) Focus and/or order original data variables |
| PC eigenvalues | / | / | ? | ? |
| Original data | / | / | / | Typical InfoVis tasks (figures 3, 5, 8, 10) |

Table 1: Possible combinations of PCA results and data for visualization purposes and the possible tasks; each cell represents a specific combination of certain kinds of input data to be used explicitly or implicitly in the visualization process; the lower triangle matrix is left empty for symmetry reasons

STARMINE : A Visualization System for Cyber Attacks

Yusuke Hideshima

Hideki Koike

Graduate School of Information System
The University of Electro-Communications
1-5-1, Chofugaoka, Chofu, Tokyo 182-8585, Japan
Email: hidesuke@vogue.is.uec.ac.jp koike@acm.org

Abstract

In cyber attack monitoring systems, various types of visualizations, such as geographical visualization, temporal visualization, logical visualization, are being used. Each visualization has its own advantages and disadvantages. Since it is important to analyze the information from different viewpoints and to make a right decision in practical cyber attack monitoring, these visualization should be highly integrated.

This paper describes a visualization system for cyber threat monitoring named *STARMINE*, which integrates three different views, that is geographical, temporal, and logical views, of the cyber threat in 3-D space. Since three views are seen simultaneously and are synchronized, it is helpful for administrators to analyze the threats much more easily. As an example, the propagation of Sasser.D worm were shown.

Keywords: intrusion detection, information visualization, information security, virus visualization, Internet worm

1 Introduction

In recent years, cyber attacks such as Internet worms or DDoS (Distributed Denial of Service) attacks are serious concerns in our society. To monitor and analyze these attacks, a variety of cyber threat monitoring systems are in use (DSshield, SANS Internet Storm Center). These systems utilize traditional visualization methods, such as geographical visualization and temporal visualization, to show the result of its analysis.

These visualizations are independent and it is difficult to know the relation between them. For example, the temporal visualization shows us transition of the number of attacks. On the other hand, the geographical visualization shows us the number of attacks of different locations at a particular moment. Even if we click on the geographical visualization, it does not reflect to the temporal visualization.

In practical cyber threat monitoring, it is important to analyze all the information from different viewpoints and to make a right decision. It would be helpful for the administrators that these different views are integrated and synchronized.

This paper describes a visualization system for cyber threat monitoring, named *STARMINE*, which integrates three different views of the cyber threat such

as geographical view, logical view, and temporal view, in 3-D space.

The next section discusses visualization methods used in cyber threat monitoring. Section 3 describes implementation details of our system and an example visualization. Section 4 discusses the advantages and disadvantages of the system. Then, Section 5 concludes the paper.

2 Visualization for Cyber Threat Monitoring

2.1 Geographical Visualization

In cyber threat monitoring, geographical visualization is one of the popular visualizations. For example, a world map is often used to show country-by-country statistics of cyber attacks. By using the map, we can easily know which attacks are currently active in the world, or how many attacks are observed in each country or in each continent. We can even plot the primary source and the destination of the attack on the map. However, the recent Internet worms use a propagation algorithm called local scan, which uses IP addresses to select the next target. In this case, the logical relation is more important than the geographical information as is described in (Ohno 2005).

2.2 Logical Visualization

Logical visualization focuses on the logical relation of IP addresses. It becomes popular in visualizing security information (Ohno 2005, Lakkaraju 2003, Yin 2004).

IP Matrix (Ohno 2005) is a 2-D matrix representation of IP addresses (IPv4) as shown in Figure 1. In the figure, the vertical axis indicates the highest 8 bits of the IP address, and the horizontal axis indicates the next 8 bits. If an attack is observed, the source of the attack is plotted on the corresponding coordinates on the matrix. By using this visualization, administrators can understand the propagation of the Internet worms intuitively.

Although the IP Matrix is effective for visualizing the worm based on the local scan, it is not appropriate for mail viruses which use email addresses instead of IP addresses.

Another issue in the IP Matrix is that one coordinate in IP Matrix is often shared by individuals or organizations in different countries. According to Koizumi's analysis (Koizumi 2004), the address block represented by the fixed higher 16 bits (e.g. A.B in A.B.C.D. we refer to it as *site*) has 1.4 countries in average. The largest number in one site is 79. This means that it is necessary to see both the logical relation and the geographical relation when analyzing the IP address.

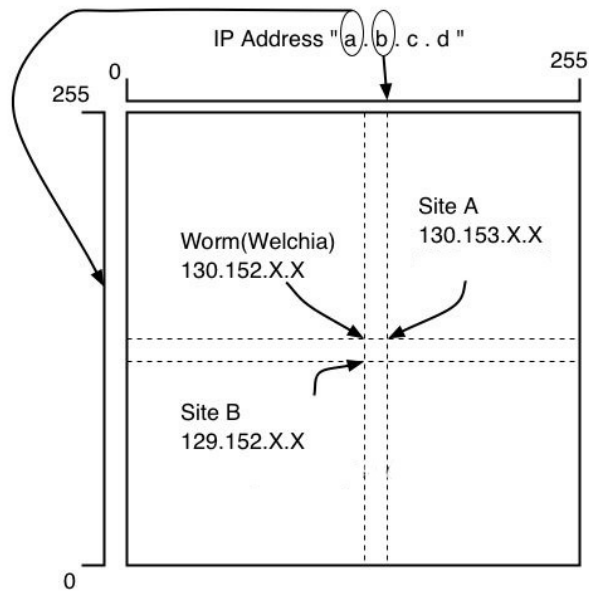


Figure 1: IP Matrix

2.3 Temporal visualization

Temporal visualization, particularly the time-diagram is often used in cyber threat monitoring. In the time diagram, X-axis indicates time and Y-axis indicates, for example, the amount of attacks. This visualization is useful to understand time transition of the attacks. However, it is not possible to see the source or the destination of the attack.

2.4 Integration

To summarize, the geographical visualization is appropriate to show statistical information of cyber attacks, and the geographical location of the source and the destination of each attack. The logical visualization is better to see automatic attacks such as scans or Internet worms. The temporal visualization is useful to understand transitions of the attacks. Each visualization has its own advantages and disadvantages.

When investigating information security, the security administrators have to see various kinds of information, compare each information, make a right decision, and take an appropriate action. Therefore, an effective visualization tool should provide a way to see such various kinds of information simultaneously.

Another important requirement is that these visualization should be integrated. For example, if the administrator finds an attack from a certain IP address in logical visualization, he or she might want to know in which countries the IP address belongs to. If the system shows each visualization on individual window, the system administrator need to make efforts to understand relations between different views. On the other hand, it would be better, when the administrator clicks on a point on the logical visualization, the corresponding countries are highlighted on the geographical visualization.

3 Implementation

3.1 Overview

To address issues in existing visualization, we developed an integrated visualization system named STARMINE. Figure 2 shows an overview of the system. As seen in the figure, STARMINE integrates the geographical view, the logical view, and the temporal

view in 3-D space. The system is written in Java 1.4.2 and Java 3D. Therefore, it will run on any platform supporting Java and Java 3D.

3.2 Cyber Threat Monitoring System

Our research group started monitoring cyber attacks in May 2003 (Fig. 3). Currently five network sensors are deployed in Japan. Three of these sensors are observing university networks, and others are observing enterprise networks. Operating System we use are RedHat Linux 9.0 and MacOS 10.2. In order to capture cyber attacks, a network-based intrusion detection system Snort(M. Roesch 1999, Snort) is being used. For the signature base, all rules files in Snort ver. 2.1 are included. Each sensors are running cron daemon to sent collected data hourly to the central log server using ssh and rsync commands. The logs collected from each sensor are used in the visualization.

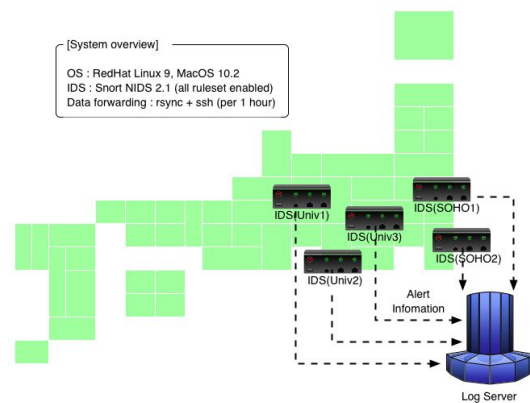


Figure 3: Cyber Attack Monitor System

3.3 Information flow

Figure 5 shows the information flow in STARMINE. The system uses an alert log gathered by the cyber threat monitoring system. The user first selects an alert log which he or she wants to visualize. The system analyzes information from the selected alert log. The information used in the system are: (1) type of attack (i.e. signature name in Snort), (2) date, (3) time, (4) source IP address, and (5) destination IP address.

At the same time, the system obtains geographical information such as longitude and latitude from IP address. To get geographical information, we used GeoIP (GeoIP). GeoIP is a geographic database provided by MaxMind LLC.. We can get geographical information, such as latitude, longitude, and name of country, from IP address by using GeoIP. GeoIP supports various kinds of programming languages, including Java, C, C++, Perl, PHP, and so on. On our system, we get latitude and longitude from GeoIP.

These information are written into a "result file". Then, the user selects the type of the attack he or she wants to visualize (Fig.4). When selecting the attack, the system gives the following information: (1) amount of attacks, (2) relative percentage of attacks, (3) name of attacks. In the same time, statistics of each attack are estimated. The system also produces a "data file". This file keeps information which is necessary in visualizing other types of visualization.

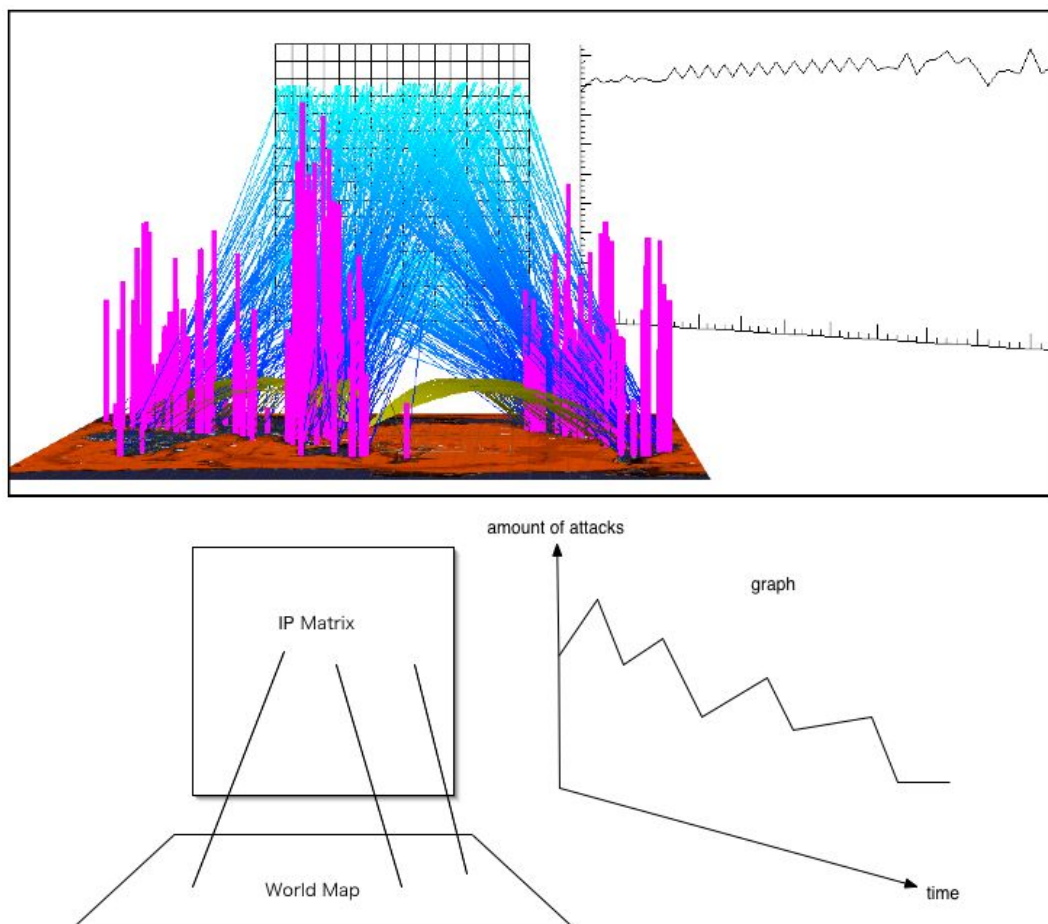


Figure 2: STARMINE Visualization

The system checks the selected alert log periodically. Whenever the alert log is updated, the system will repaint the visualization.

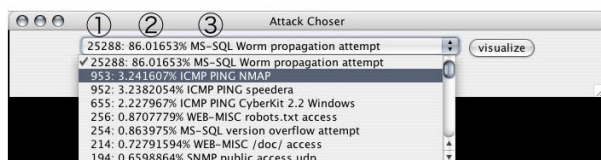


Figure 4: Selecting an attack – Figure indicates: (1) amount of attack, (2) relative percentage of attacks, (3) name of attack

3.4 Visualization

We implemented three types of visualizations. Integrated Visualization (Integrated View (Fig.2)) and optional two geographical visualizations (Globe View (Fig.6) and Map View (Fig.7)). Figure 6 and Figure 7 are visualizing the same data.

3.4.1 Integrated View

Integrated View consists of a world map, IP Matrix, and a time-diagram as shown in Figure 2. IP Matrix is shown on the center of STARMINE. The vertical

axis indicates the highest 8 bits of the IP address, and the horizontal axis indicates the next 8 bits.

Under the IP Matrix, a world map are shown. The statistical information, such as amount of attacks from each location are shown as a bar chart. Arcs on the map indicate the source and the destination of each attack.

On the right, there is a time-diagram. This graph shows time transition of the amount of attacks. The vertical axis indicates the amount of attacks, and the horizontal axis indicates the time. The scale of the graph dynamical changes depending on the amount of data and date.

These three views are integrated and synchronized. For example, each location and the corresponding IP address block in the IP Matrix are connected with a line. It is helpful to understand the relationship between IP addresses and its geographical location.

3.4.2 Globe View

Globe View (Fig. 6) shows statistical information of attacks using globe. It shows where the attacks come from. Lines are drawn inside the sphere. The source and the destination are connected with a line. The source and the destination are colored as green and white, respectively.

The globe view rotates slowly. The user can rotate the globe by using mouse, too.

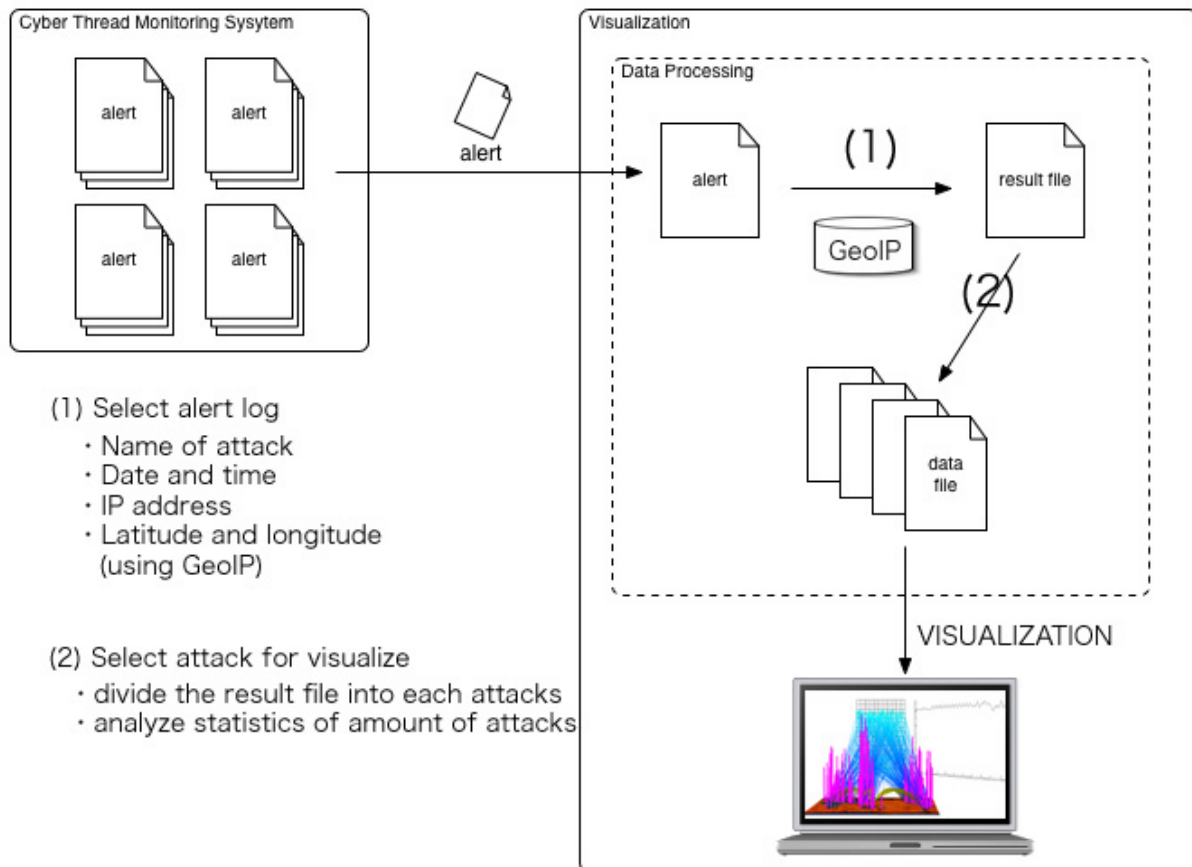


Figure 5: Information flow.

3.4.3 Map View

Map View (Fig.7) is a visualization which uses 2-D world map. It is a simple version of the world map in Integrated View. It can show the amount of the attacks and where the attacks come from. The source and the destination of the attacks are linked by an arc.

3.5 Visualization Examples

Sasser.D

Figure 8 and Figure 9 are visualizations of Sasser.D worm observed by our cyber threat monitoring system. Sasser.D appeared on April, 2004. The snort recorded the alerts on Sasser.D worm as "ICMP PING NMAP". So we can easily find Sasser.D worm. Figure 8 shows the situation on April, 2004, and Figure 9 is on May, 2004.

Figure 8 has few "ICMP PING NMAP" alert. On the other hand, on Figure 9, Sasser.D worm was spread over east Asia. Also, when we look at IP Matrix, we can know that Sasser.D worm was spread at IP address 218.*.*. This is a typical feature of the local scan. The figure illustrates that east Asia were seriously damaged by Sasser.D.

Welchia

Figure 10 is visualization of Welchia worm. Welchia uses a dictionary to select targets. The dictionary is a list of a popular first octets of IP addresses, such as 61, 202, 203, 210, 211, 218, 219, and 220. Welchia

uses these numbers to decide the first octet of the target IP address. The snort recorded the alerts on Welchia as "ICMP PING CyberKit 2.2 Windows".

Figure shows that the Welchia were spread all over the world.

Blaster

Figure 11 is visualization of Blaster worm. Blaster discovered on August 11, 2003. Blaster uses local scan to spread. Snort logs the Blaster as "SHELLCODE x86 NOOP", "NETBIOS DCERPC ISystemActivator path overflow attempt little endian unicode", "NETBIOS DCERPC Activation bind attempt", and "NETBIOS DCERPC IActiveation little endian bind attempt".

On Figure 11, all of the blaster were came from China in contrast to Welchia (Fig 10). When we look at IPMatrix, infected IP addresses are 211.*.*.

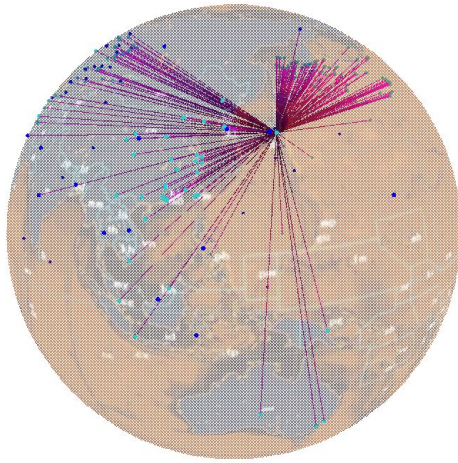


Figure 6: Globe View

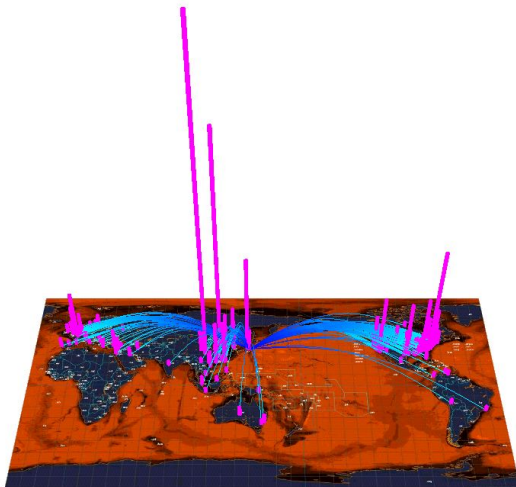


Figure 7: Map View

4 Discussion

4.1 Integrated View

Many Internet worms use IP address for propagation. Therefore, it is commonly said that geographical factor has less relation to the propagation of the Internet worms. However, Figure 9 shows that the place which has the same highest 8 bits of IP addresses are on east Asia. This indicates that IP address and geographical location has some relation.

The infection of computer viruses or Internet worms would be effected by actual movement. For example, we can easily carry our notebook computer to our office or Internet cafe. If the computer was infected by the computer viruses or Internet worms, the computer will spread worms or viruses to the office's computers or Internet cafe's computer. Our home and office might be far from the viewpoint of IP address, but it is possible to infect worms or viruses by actual movement.

By using STARMINE, we can take some actions against Internet worms or computer viruses from the

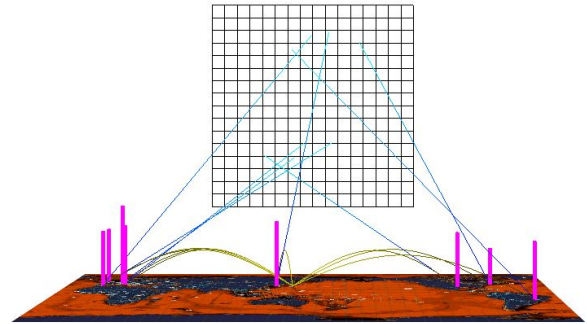


Figure 8: Visualizing Sasser worm (April 2004)

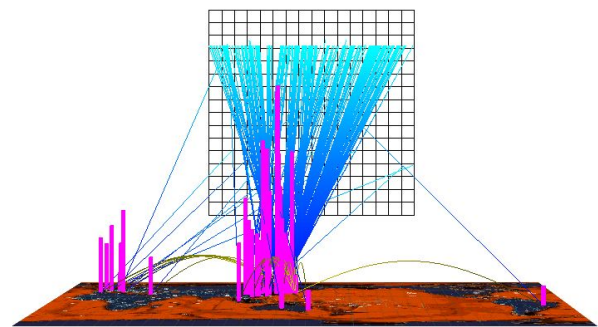


Figure 9: Visualizing Sasser worm (May 2004)

IP address's point of view as well as the geographical point of view.

4.2 Globe View and Map View

Using Globe View, we can see the trends of cyber attacks in the world, and the source and the destination of the attack intuitively. However, it is difficult to compare the amount of attacks by using bar charts on Globe View.

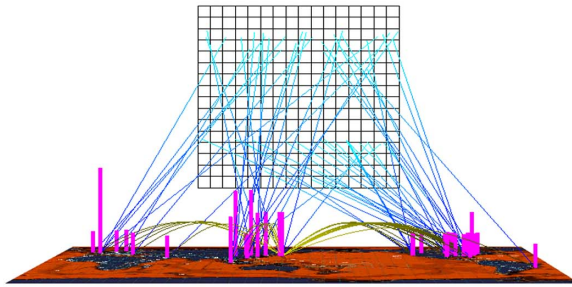
Map View can show the trends of cyber attacks and the path of attacks intuitively as Globe View. Furthermore, it is easy to compare the amount of attacks by using bar charts. However, Map View can not indicate the place which has possibility of infecting Internet worm like Globe View. Putting spheres like Globe View is not good idea, because the arcs on the map would hide the spheres. We have to make other representation method.

4.3 Future Work

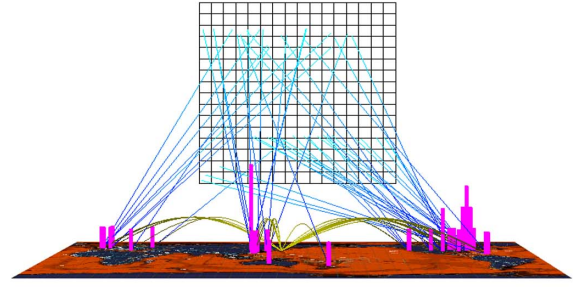
The followings are future work to be done.

- animation
To understand the time-transition of Internet worms, the animation is effective. If the administrators could know the situation of spreading worms in real-time, they can take some actions against worm rapidly.
- visualizing multiple logs
STARMINE do not support visualizing multiple logs. By comparing multiple logs, the administrators can know a lot of information and they

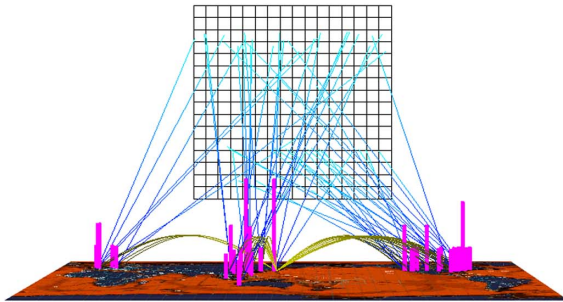
9th March 2004



10th March 2004



12th March 2004



14th March 2004

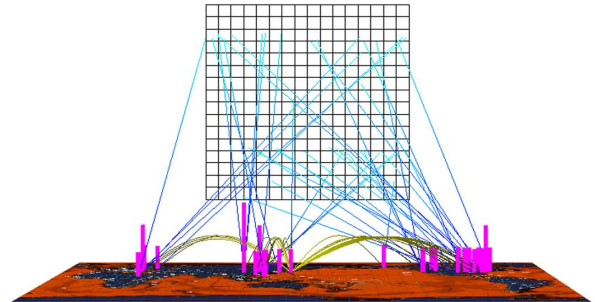
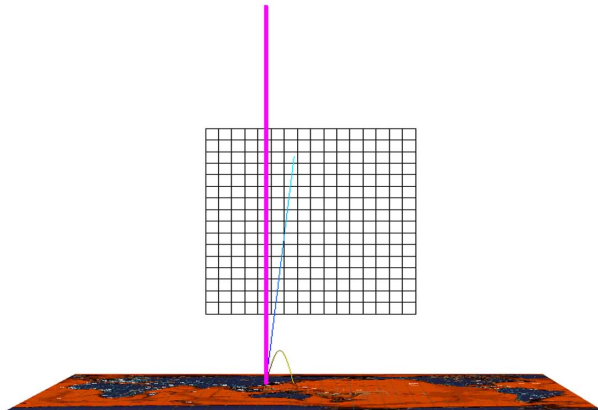
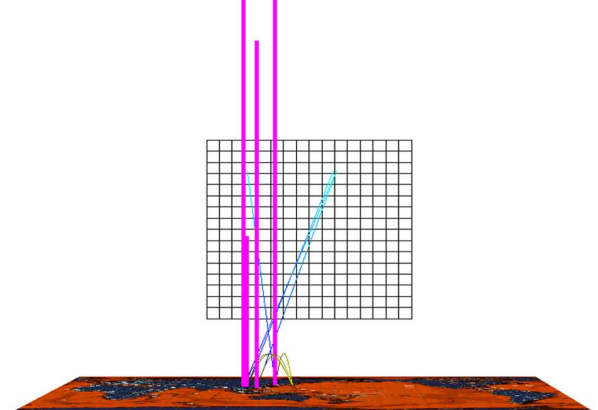


Figure 10: Visualizing Welchia worm (March 2004)

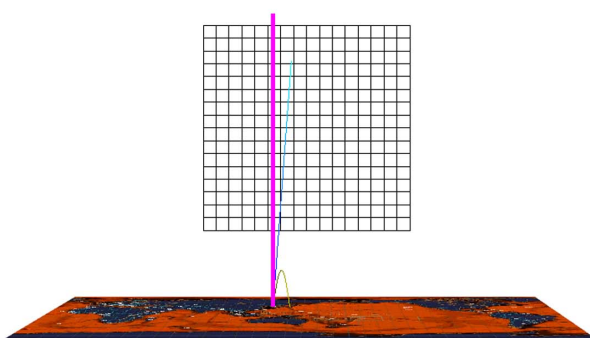
9th March 2004



10th March 2004



12th March 2004



14th March 2004

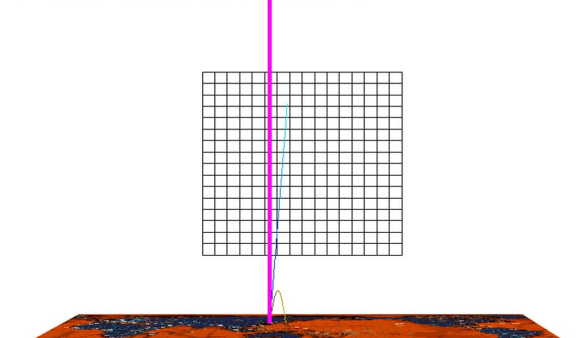


Figure 11: Visualizing Blaster worm (March 2004)

can utilize that information for our action against cyber attacks.

- visualizing multiple attacks
Visualizing different kinds of attacks simultaneously is also important. This would show us the relation between the different attacks. For example, Welch worm cleans Blaster worm and makes a patch to the vulnerable system. By visualizing multiple attacks, we could observe such a interesting behavior of Welch and Blaster.
- visualizing other logs
Comparing various types of logs, such as firewall log, system log, packet sniffer log, is important. The intrusion detection system log has a lot of false positives. To reduce the false positives, the best way is to compare with other logs. Also, the intrusion detection system cannot detect unknown attacks. In this case other logs will help us. Thus, we need visualization of other types of logs.

4.4 Toward Security Forecast

One of the important challenges in cyber attack monitoring is prediction of attacks. If the attack is predicted, the damage of the attack could be minimized. Although IDS can detect attacks in real time, it cannot predict the attacks.

On the other hand, our system might be used for prediction, because the user can understand the spatial diffusion of attacks as well as their transition in time. Recent worms use random scan¹ and local scan. When one site is infected by a worm, it is highly possible that its logical neighborhoods are the next target. IP Matrix helps predicting the attacks. Geographical visualization will help to predict the mail virus. Mail virus use address book of infected computer and send mail with virus to your friends. So, it is considered that propagation of mail virus would rely on geographical matter. This is why geographical visualization helps to predict mail virus.

Some computer virus, like Nimda, has more than two propagation algorithm. For example, using email, local scans, random scans, and so forth. STARMINE would be effective against these complex types of virus.

4.5 Related Work

Starlight (Rich 1997) is a 3-D visualization system of multimedia information. Starlight visualizes various types of information, including structured and unstructured text, geographic information, and digital imagery. These multimedia information are visualized in 3-D space altogether. Relational information are linked together. Starlight are developed for intelligence analysis. Starlight can not be used for network monitoring.

(Tominski 2005) developed a 3D visualization for time dependent data using maps. This system visualizes time dependent data on maps. To represent time dependent data on maps, they use 3D icons, Pencil icons and Helix icons. On pencil icons, linear time is encoded along the pencil's faces. Each faces of a pencil can be adjusted according to the number of attributes to be visualized. Helix icons can emphasize the cyclic character of a data set. Both icons represent time as height of the icons. This system is a good example of integrating maps and time-transition data.

¹random scan - The target IP address is randomly scanned. This algorithm is inefficient because it has to scan all IP addresses space (32-bit). CodeRed and SQL Slammer used this algorithm

As we described previously, a variety of cyber attack monitoring systems are in service today (DShield, SANS Internet Storm Center). They are providing information on cyber attacks using country-by-country statistics visualization or time diagrams.

The SIFT (Security Incident Fusion Tools) project visualizes security information. NVisionIP (Lakkara 2003) visualizes both transfer information and the connection information in a small-scale network. VisFlowConnect (Yin 2004) visualizes traffic between an internal network and an outer network using a line. This system focuses on a local area network and is not appropriate for use in large-scale network monitoring.

Tudumi (Takada 2002) visualizes connection to a particular server in one 3-D visualization by taking several logs (syslog, sulog, wtmpx, etc.) as input. In Tudumi, the connecting hosts are categorized by network domain and displayed as a stack of circles. It does not have real-time monitoring capability.

Snort View (Koike 2004) visualizes Snort logs using a time-diagram. It shows each attack as an icon whose shape and color indicates the type and priority of the attack. It overlays a histogram of the attack on each icon, which helps to avoid overwhelming the screen large numbers of the same icon.

5 Conclusion

This paper described a visualization system named STARMINE. STARMINE is a visualization system for cyber threat monitoring. It integrates logical, geometrical, and temporal visualization. STARMINE can show the relationship between geographical space and cyber space. We visualized Sasser.D worm propagation using STARMINE. We can see the typical features of Sasser.D worm propagation.

References

- DShield, Distributed Intrusion Detection System, <http://www.dshield.org/>
- SANS Internet Storm Center, <http://ics.sans.org/>
- M. Roesch (1999), Snort - Lightweight Intrusion Detection for Networks, Proc. of the 1999 USENIX LISA conference, 1999.
- Snort, The Open Source Network Intrusion Detection System, <http://www.snort.org/>
- Kazuhiro Ohno, Hideki Koike, Kanba Koizumi (2005), IP Matrix : An Effective Visualization Framework for Cyber Threat Monitoring, in 'Ninth International Conference on Information Visualization (IV05)', London, England, IEEE/CS, pp. 678-685
- Kanba Koizumi, Hideki Koike, Michiaki Yasumura, (2004), The Relationship Between Virus Spread Process and The Infected Number of Countries, Computer Security Group, Information Processing Society of Japan. (in Japanese)
- MaxMind LLC., GeoIP, Geolocation IP Address to Country, <http://www.maxmind.com/>
- J.S. Risch, D.B. Rex, S.T. Dowson, T.B. Walters, R.A. May, B.D. Moon, The STARLIGHT Information Visualization System, Proceedings of IEEE International Conference on Information Visualization, IEEE, 1997

- Christian Tominski, Petra Schulze-Wollgast, Heidrun Schumann, 3D Information Visualization for Time Dependent Data on Maps, 'Ninth International Conference on Information Visualization (IV05)', London, England, IEEE/CS, pp. 175–181
- Kiran Lakkaraju, Ratna Bearavolu, William Yurcik, NVisionIP - A Traffic Visualization Tool for Security Analysis of Large and Complex Networks, International Multiconference on Measurement, Modeling, and Evaluation of Computer-Communications System (Performance TOOLS), 2003.
- Xiaoxin Yin, Cristina Abad, VisFlowConnect: Providing Security Situational Awareness by Visualizing Network Traffic Flows, Workshop on Information Assurance (WIA04) held in conjunction with the 23rd IEEE International Performance Computing and Communications Conference (IPCCC 2004), 2004.
- Tetsuji Takada, Hideki Koike, (2002), Tudumi: Information Visualization System for Monitoring and Auditing Computer Logs, Proc. on Information Visualization (IV2002), IEEE/CS, pp.570-576, 2002.
- Hideki Koike, Kazuhiro Ohno, SnortView: Visualization of Snort Logs, Proc. of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security (VizSEC '04), 2004.

Spatial analysis of centralization and decentralization in the population migration network

S. Tomita and Y. Hayashi

School of Knowledge Science, Japan Advanced Institute of Science and Technology
1-1 Asahidai, Nomi-city, Ishikawa 923-1292

s-tomita@jaist.ac.jp, yhayashi@jaist.ac.jp

Abstract

Although overcrowded and the depopulation of the population are one of the social issues in Japan, the complex mechanisms are not comprehended. Then, we tried to show this phenomenon by catching as a network. We interpreted prefectures as nodes and migration as links. We observe power law distribution in both population and migration. We insist on the necessity of reconsider the prevailing assumption that cities size or influential potential follow normal distributions. We investigate what factor influence on the migration. We compare the population and distance between prefectures. As a result, we showed that the population of destination has strong influence in migration. We visualize the migration. We characterized quantitatively and showed the whole image of the migration. This analysis may be useful at various stages in decision-making and planning support of government and municipalities.

Keywords: spatial structure, scale-free, gravity model

1 Introduction

We study the population migration between 47 prefectures in Japan. In general, concentrates and depopulation have progressed simultaneously with economy developing. It is pointed out that the generation factor of this phenomenon is a failure of the policy. However, is it really so? In such phenomenon observed not only in Japan but also in other countries (K. Doo-Chul 1997). Individual may migrate by the convenience of work, own intention and so on. It might be difficult to find the common law to the individual's migration. But in a huge amount of migration decided by the great number of people, an objective and common law can be found. Our purpose is to show the whole image of the population migration. In particularly, we devote to developing the quantitative analysis and the visualization.

The paper is organized as follows. In the next section, we investigate the migration transition in Japan. We explain the rank size rule that is empirical law in the population distribution of the city. We also describe the dataset of migration between inter-prefectural migrants. We show that this rule applies to the migration. Furthermore, we analyse the data according to the gravity model. In section 3, we visualize the migration network. In section 4, we summarize the visualization of the migration network.

Copyright © 2006, Australian Computer Society, Inc. This paper appeared at *Asia Pacific Symposium on Information Visualization 2006 (APVIS 2006)*, Tokyo, Japan, February 2006. Conferences in Research and Practice in Information Technology, Vol. 60. K. Misue, K. Sugiyama and J. Tanaka, Eds. Reproduction for academic, not-for profit purposes permitted provided this text is included.

2 Investigation of migration transition

The population distribution is one of the typical indexes that represent the power of the city. It is known that there is a rule between the population and ranking. A city size can be expressed in terms of the power law (N. Hayashi 1991). This rule was pointed out by Auerbach (F. Auerbach 1913) and applied to size of city by Zipf (G. K. Zipf 1949). The following expressions consist experiencing when the country is in the stable state.

$$y = bx^{-\gamma} \quad (1)$$

where y denotes the population, x denotes ranking, γ denotes power law index, b denotes parameter. We take the logarithm of expression (1), it's rewritten as

$$\log y = \log b - \gamma \log x \quad (2)$$

Power law distribution is linear logarithm with exponent $-\gamma$ on both logarithm graph distribution. It doesn't concentrate around the mean value like normal distribution. For the scale-free minority has high value, while majority has gathered near minimum value. The γ represents concentrated level to minority. Figure 1 shows the relation between population and ranking of prefectures. These lines exhibit power laws.

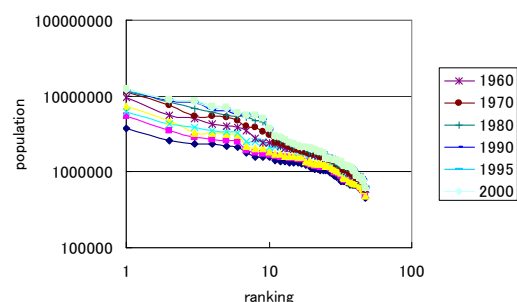


Figure 1: Relation between population and ranking

We used the least square method to estimate the exponent γ . The transition of the exponent γ is shown in Table 1.

| year | γ | Coefficient of determination |
|------|----------|------------------------------|
| 1960 | 0.66 | 0.96 |
| 1970 | 0.76 | 0.97 |
| 1980 | 0.78 | 0.96 |
| 1990 | 0.80 | 0.95 |
| 2000 | 0.81 | 0.95 |

Table 1: Transition of γ of population

The coefficient of determination indicates how much of the total variation in the dependent variable can be accounted for by the Regression Function (expression (2)). Most statisticians consider a coefficient of determination of 0.7 or higher for a reasonable model. It is understood that relation between population and ranking suit power law by height of coefficient of determination. And γ is growing year by year. It is mean that the difference of populations among prefectures has expanded. Then, how about migration? Does the migration also follow power law? The answer is YES. Figure 2 shows the relation between migrants and ranking of prefectures. In the box, “in” represents the migration from the outside to the inside and “out” represents the migration from the inside to the outside. The transition of γ in Figure 2 is shown in Table 2. It is understood that relation between migration and ranking also suit power law by height of coefficient of determination. Note that “ γ -in” maintains a very high value. On the other hand, “ γ -out” has been rising gradually though is low the start.

In a word, we clarified not only population but also migration according to rank size rule.

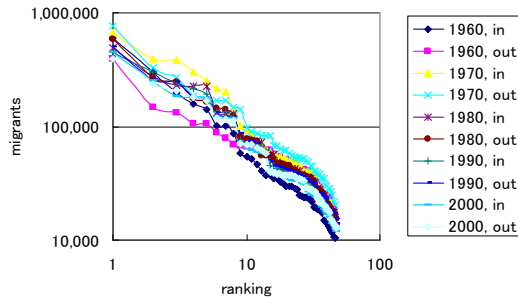


Figure 2: Relation between migrants and ranking

| year | γ -in | Coefficient of determination | γ -out | Coefficient of determination |
|------|--------------|------------------------------|---------------|------------------------------|
| 1960 | 0.96 | 0.99 | 0.67 | 0.95 |
| 1970 | 1.07 | 0.95 | 0.85 | 0.98 |
| 1980 | 0.94 | 0.97 | 0.90 | 0.97 |
| 1990 | 0.97 | 0.97 | 0.91 | 0.98 |
| 2000 | 0.95 | 0.97 | 0.91 | 0.97 |

Table 2: Transition of γ of migration

In Japan, urbanization that population concentrates on city and depopulation that population of mountainous village decreases extremely have progressed simultaneously with economy developing after the war. We showed population and migration follow power law in the previous section. Next, we investigate the relation between population and migration. We used the migration data between prefectures from 1960 to 2004 (RIMJ). Figure 3 represents the relation between population and migrants (data of 2000). In this Figure, x-axis represents population and y-axis represents migrants. There are a lot of migrants like a populous prefecture. This relation can be approximated by the straight line. In a word, population and migration

are in the proportion. The slope of this approximated line is denoted b .

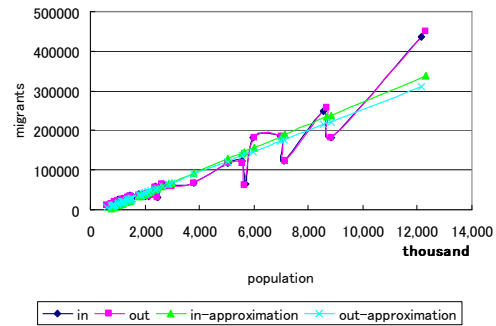


Figure 3: Relation between population and migrants

Figure 4 shows slope b transition from 1960 to 2004. If “ b -in” is higher than “ b -out”, it represents the centralization of the population on populous prefectures has happened. Inversely, if “ b -out” is higher than “ b -in”, it represents the decentralization has happened. Slope b has become small year by year. This means that the difference of the migration in populous prefectures and few prefectures has become small. “ b -in” is higher from 1960 to 1970 (①). “ b -out” is reversed from 1970 to 1990 (②). In addition, “ b -in” is reversed since 1990 (③). Centralization and decentralization of the population are repeated. This is different from a general opinion that the centralization of the population has advanced consistently.

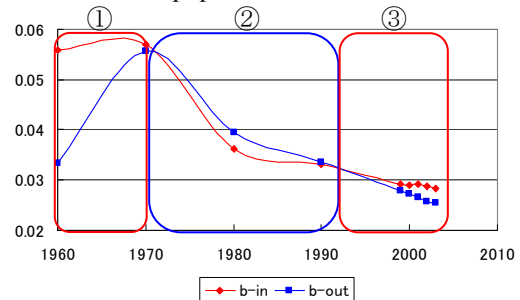


Figure 4: Transition of b

For the whole set of prefectures partitioned in 47 the Ministry of Internal Affairs and Communication (MIAC) has issued the report on internal migration. The report constructed on the output of a survey about migration between prefectures. In particular we have considered the local flows $i \rightarrow j$ which measure the movements from any prefecture i to the prefecture j .

The adjacency matrix is then becomes a complete graph because there is a migration between any prefectures. We can interpret this network as directed network. We construct the weighted adjacency matrix W in which the elements w_{ij} are calculated as the sum of the migrants $i \rightarrow j$.

The weighted graph provides a richer description since it considers the topology along with the quantitative information on the dynamics occurring in the whole network. It is also important to stress that while the nodes correspond to migrants located in the physical space.

What factor influences on the migration? The method of analysing the interaction between two regions with a spatial distance is called to be a spatial interaction model

generically (J. K. Stewart 1947). In Newton's law of universal gravitation, the gravitational force f is directly proportional to the mass of both interacting objects m_1, m_2 ; more massive objects will attract each other with a greater gravitational force and is inversely proportional to the separation distance d between the two interacting objects; more separation distance will result in weaker gravitational forces.

$$f = G \frac{m_1 m_2}{d^2} \quad (3)$$

where G is a constant. In general (3) is expressed as

$$f_{ij} = K \frac{P_i^\alpha P_j^\beta}{d_{ij}^\gamma} \quad i \in N, j \in N \quad (4)$$

where, f_{ij} denotes gross migration, P_i denotes the population of origin, P_j denotes the population of destination, d_{ij} denotes distance between i and j (prefectural government addresses), K, α, β, γ denotes parameters, and N denotes sets of prefectures.

In this section, we apply gravity model to the dataset of migration between inter-prefectural migrants from 1960 to 2004. We assume that gravity model can explain migration on some level and investigate which element is the dominant in α, β, γ . The exponent α represents the source power level that population of origin gives migration. β represents the destination level that population of destination gives migration. γ represents the influence level that distance between i and j gives migration (distance resistance). The estimation of these parameters $\{\alpha, \beta, \gamma\}$ takes natural logarithm of both sides of the expression (4),

$$\log f_{ij} = \log K + \alpha \log P_i + \beta \log P_j - \gamma \log d_{ij} + \varepsilon_{ij} \quad (5)$$

the least square method is applied to data $\{f_{ij}, P_i, P_j, d_{ij}\}$. ε_{ij} is error term. Figure 5 shows the transition of α, β, γ . α and β change similarly a constant distance.

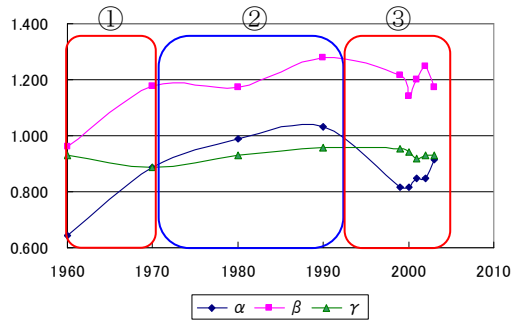


Figure 5: Transition of α, β, γ

β takes a highest value in constant. It can be said that the population of destination has the strongest influence on the migration. γ shows the steady value (about 0.9) compared with α and β . Though it is thought that the development of transportation pressed the migration in general, the distance resistance to migration has not been changing for 40 years almost. As soon as not, it is understood that the change of the size of population influences the migration.

It is interesting to note that the period when α exceeds γ in Figure 5 is corresponding to the period when “b-out” exceeds “b-in” in Figure 4. It is expected that dispersal of population and rise of influence level that population of origin gives migration are related. The dispersal of population occurs when α exceeds γ . Oppositely, the concentration of the population occurs when α dip from γ .

3 Visualisation of the migration

We visualize the migration in 2003. The migration between prefectures is various from a small amount to a large amount. For example, Tottori \rightarrow Iwate is 5, Tokyo \rightarrow Kanagawa is 85437. Thinking about small link and big link as the same link is unnatural. To express these differences, we did the rank division of links.

3.1 Rank division of migration

We divided the migration and extracted rank. Figure 6 represents the rank division of migration. We can see the region with influence power on a nationwide scale (① SAITAMA, TIBA, TOKYO and KANAGAWA), region with influence power within the medium-scale range (② KYOTO, OSAKA and HYOGO) and region with influence power within the small-scale range (③ FUKUOKA). We can also see a lot of migration in neighbouring prefectures (④) and a little migration in far prefectures (⑤). In a word, the instructions exist not only related to spatial structure but also unrelated to spatial structure.

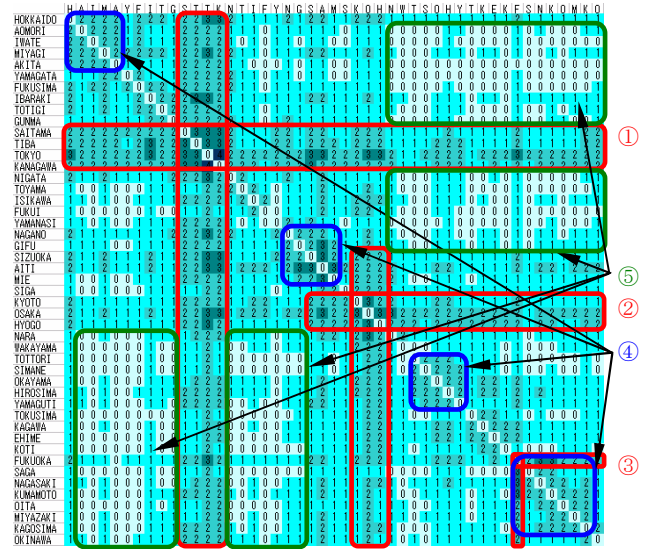


Figure 6: rank division of migration (color coding: number of migrants 1~80: rank0, 81~800: rank1, 801~8000: rank2, 8001~80000: rank3, 80001~: rank4)

3.2 Geographical mapping

Next, we put the nodes and the links on an actual Japanese map. The migration is intuitively understood by geographical mapping. We visualize over rank 3 (Figure 7). This is a method by the manual operation. In Figure 7, the prefecture where the amount of the migration is

extremely large is extracted. A lot of long links are seen. In a word, there are many migrations in far prefectures. The influence power to migration of distance is not too large as obtained by the result of the gravity model. Then, how should be visualized the actual migration space that doesn't depend on the distance? Next, we visualize migrating by the algorithm that automatically calculates the position of the node.

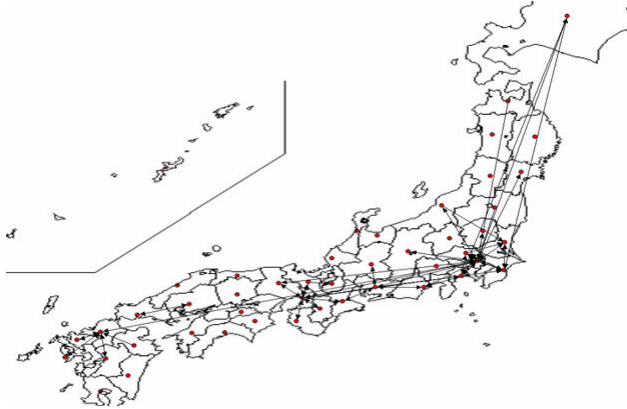


Figure 7: Geographical mapping

3.3 Relational mapping

Automated procedures for finding an optimal layout are a better way to obtain a basic layout than manual drawing, because the resulting picture depends less on the preconceptions and misconceptions of the investigator. In addition, automated drawing is much faster and quite spectacular. Kamada-Kawai lays out is assumed that the node is connected by the springs (T. Kamada and S. Kawai 1989). Essentially, the network is modelled as a collection of nodes connected by springs with resting lengths proportional to the length of the shortest path distance between each node pair. We visualize over rank 3 initial location as geographical mapping (Figure 8).

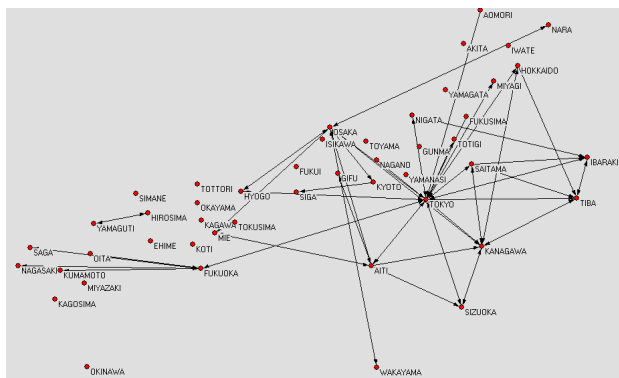


Figure 8: Relational mapping

By this figure, we can see the spatial structure and distance of migration which different from actual location and distance between prefectures. The prefectures which have a lot of links come near to center and the prefectures which have few links scatter in the surrounding. The distance of the prefecture that exists mutually at a far position has actually shortened. Figure 8 shows the migration relation between prefectures more clearly than Figure 7.

In a migration space independent of the actual location, the distance between TOKYO and HOKKAIDO is shorter than the distance between TOKYO and AOMORI.

4 Conclusions

In this paper, we study the quantitative feature of migration and how to visualize it. We treat the migration between prefectures as a directed weighted network. First thing we understood is that not only population but also migration according to rank size rule. It can be said that this is a crucial discovery. It doesn't concentrate around the mean value like normal distribution, but the same distribution is shown. Understanding concerning this is very important for thinking about the policy. This is the evidence to prove a past policy of giving subsidy technique did not go well. Second thing we understood is that the population of destination has a strong influence on migration. However, the distance is not strongly dominant factor. There might be correlations in dispersal of population and rise of influence level that population of origin gives migration. Finally, we visualize the geographical mapping and relational mapping. It has been understood that the migration is independent of the distance by this visualization. The migration receives a stronger influence from the population than distance. If the mechanism that the person calls the person works, populous prefecture becomes an attractive prefecture. It is necessary to do development of original charm from reduce the distance by building the road and the airport, etc.

Further studies include spatial aspects of the network. This forthcoming analysis will hopefully provide decision makers and planners with relevant indications on the global effect of new infrastructure.

5 References

- F. Auerbach (1913): Das Gesets der Bevölkerungskonzentration. *Petermann's Mitteilungen*, **59**:74-76
- G. K. Zipf (1949): Human Behavior and the Principle of Least Effort. *Addison-Wesley Press*
- K. Doo-Chul (1997): Economic Growth, Migration and Rural Depopulation in the Republic of Korea :Comparison with Japan's Experiences. *Regional Development Studies*, **3**:239-259
- N. Hayashi (1991): Space system and location of city. *Daimeido*:62-63
- T. Kamada and S. Kawai (1989) : An Algorithm for Drawing Undirected Graphs. *Information Processing Letters*, **31**:7-15
- MIAC: Ministry of Internal Affair of Communication, <http://www.stat.go.jp/english/index.htm>. Accessed 12 September 2005
- RIMJ: Report on Internal Migration in Japan, <http://www.stat.go.jp/english/data/idou/index.htm>. Accessed 12 September 2005

Visualization by information type on mobile device

Hee Yong Yoo

Suh Hyun Cheon

Department of Computer Engineering

Dong-Guk University

26-3Ga Pil-dong, Chung-ku, Seoul, South Korea

yooheeyong@nate.com

shcheon@dgu.ac.kr

Abstract

The number of mobile device users is increasing for its convenience of information searching and its guaranteed mobility. Unlike desktop environment with large display screens, these mobile devices have a limitation to present information on small screens. Though hardware performance of mobile device is improving, the problem of displaying information on such small screens will still remain. This paper proposes information visualization method as a solution to the above problem. Information visualization method effectively layouts and efficiently navigates information on small a screen considering given resources and process power of mobile devices.

Keywords: Visualization, Fisheye view, Layout, Mobile

1 Introduction

The number of people who uses internet on their mobile device is continually increasing, therefore there is a demand for efficient mobile user interface that can effectively display information on small sized mobile screen. Mobile processing power has been greatly improved with the enhanced hardware. However, the limitation of display screen size may remain as an important issue to present information on mobile devices(Kris Luyten, Karin Coninx 2001). Many methods to better present information with limited screen have been developed from desktop. Information visualization methods had been studied in various forms and are developed for desktop. Though these visualization methods adopted from desktop and apply to mobile device, the following restrictions still remain(Staffan Björk et al. 2000).

- Limited calculation ability
- Limited screen size
- Small memory volume
- Low network bandwidth

Also, there is no standard for the sizes of screen on mobile devices, therefore the screen sizes vary from different manufacturers, models, purposes etc(see table 1)(Steve Jones et al. 2004). In order for effective visualization to these mobile devices, we need information layout methods

Copyright © 2006, Australian Computer Society, Inc. This paper appeared at Asia Pacific Symposium on Information Visualization (APVIS 2006), Tokyo, Japan. Conferences in Research and Practice in Information Technology, Vol. 60. K. Misue, K. Sugiyama and J. Tanaka, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

| Feature | PDA | Smart Phone | Cellular Phone |
|------------------------------|--|--|--|
| Screen size | QVGA(240x320), 160x160 etc | 640x480, QVGA, etc. | 128x128, 128x160, 176x220, QVGA etc. |
| Color facilities | 4gray, 16gray, 256 color, 16bit color etc. | 4gray, 16gray, 256 color, 16bit color etc. | b/w, 4gray, 16gray, 256 color, 16 bit color etc. |
| Input facilities | touch screen, pen, buttons | pen, buttons | buttons |
| O/S | WinCE, PalmO/S etc. | WinCE, PalmO/S etc. | many |
| Computational power(v s. PC) | low | low | low |

Table 1: Feature of various mobile device

and effective visualization algorithm that utilize small screen to solve the restrictions of mobile devices. Among the desktop-based developed visualization algorithms, focus+context algorithm is suitable to apply to mobile devices(Brend Karstens et al. 2003).

This paper analyzes information types and match suitable layouts according to the type of information and presents information applying fisheye view algorithm that is one of focus+context visualization methods. Also, the user's recognition can be improved by displaying an animation during browsing or calculating.

In section 2, we study related research of information layout and visualization algorithm. In section 3, we propose layout method, visualization algorithm and animation according to the type of information. In section 4, we look at the examples that apply application to PDA and Cellular Phone. In section 5, we give our conclusion and the direction for future improvement.

2 Related Works

Information gathered from Internet or database is provided to user in various forms. Some information is delivered in sequential form, and some information is delivered in hierarchical form such as tree structure or graph that is more general form. There is active research in progress on how to effectively convert and deliver web

information on desktop to mobile (Tamara Munzner 1997). And various layout methods and visualization algorithm have been developed to effectively delivery various types of information. Also, various criteria were proposed to estimate usability of this methods (Freitas et al. 2002). We discuss about techniques of the currently developed information visualization in this section.

2.1 Layout

Layout is the stage to decide visual size and location of information to be displayed (S. Lok et al. 2001). Effective layout is one of the most important steps in optimization of information visualization. Information layout is applied differently according to types of the information. For example, method that ranks orders of search results with the search word like Google prefers sequential layout. This type of information is less inter-relational and existing scrollbar is used to present the information. Unlike this, information such as top-down menu prefers hierarchical layout such as tree. One of the most general hierarchical information presentation methods is graph, and G. di Battista introduction of various kinds methods to do layout in display space is shown in his book (Battista et al. 1999). Among those methods, radial drawing can efficiently lay out information of tree or graph according to the number of information or types (Ka-Ping Yee et al. 2001, Jankun-Kelly et al. 2002, Battista et al. 1999).

2.2 Visualization Algorithm

Many information visualization algorithms have been developed in existing desktop environment. As mentioned before, most web browsers use scroll as one of the visualization methods as Google's search results visualization algorithm. This method has a weak point that small screen of mobile device that cannot display whole information therefore requires too much scrolling. Also, this method has a problem that current cellular phone does not offer such special UI control. Many visualization algorithm methods to display hierarchical information forms such as top-down menu or file and directory have been developed. As a solid example, tree map offers an advantage that can use whole screen; however, users can find difficulties to understand the whole information with hierarchical structure. And there is fractal method that uses self-similarity. Also, there is focus+context method that presents overall information structure and detailed information on parts that a user is particularly interested. Two typical methods that are effective on small sized screen mobile devices are hyperbolic geometry visualization method and fisheye view visualization method.

2.3 Animation

Effectively to inform information to user, many visualization method is introducing animation way to improve user's understanding at information layout's reconstruction by user interest. John Lamping (John Lamping et al. 1995) applied animation effect when arrange new information location to display space by applying hyperbolic geometry visualization algorithm information. Tamara Munzner (Tamara Munzner 1997)

applied visualization algorithm of John Lamping to third dimension space and apply animation effect in 3D. By other visualization method Cone Tree and Perspective Wall are applying third dimension animation effect to help that user follows information trace (Ka-Ping Yee et al. 2001).

3 Visualization of Information on Mobile Device

Mobile device resources are much limited compare to desktop resources. Therefore in order to effectively process information, the followings are required. First, we apply each layout to use limited screen effectively according to the type of information. Second, we apply visualization algorithms that consider limited calculation ability of mobile devices. Third, we must supply efficient animation effects for users to better recognize changes of information. Considering the three conditions mentioned above, our approach applies proposed visualization method. Visualization method is processed according to the procedure in figure 1.

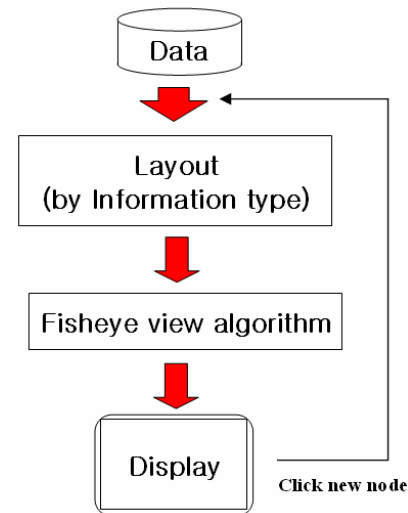


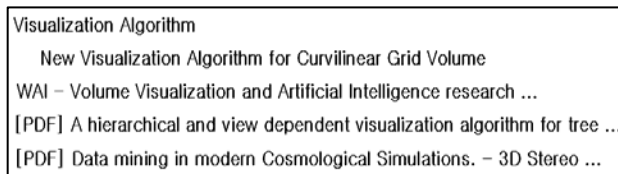
Figure 1: Information visualization process

3.1 Layout by Information Type

As information increases and the relation between information becomes complicated, efficient information layout on display screen is being more and more important. Types of Information can be divided into two forms. One is sequential type and the other is hierarchical type where there are relations between information. This research applies different layouts according to the two information types.

3.1.1 Sequential Layout

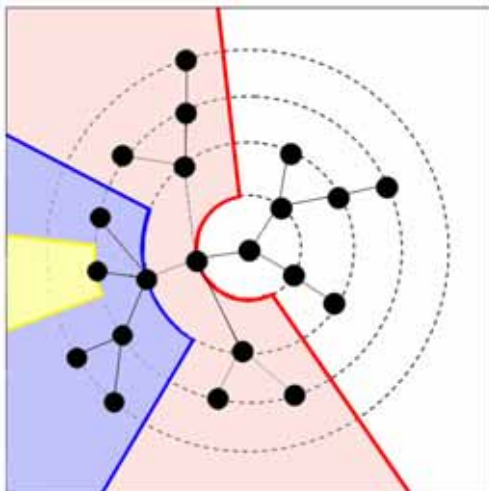
Sequential layout is a suitable for information that has less relation between information. For example, it is used to give search word in search site such as Google in Internet and show search result or to give query in database and lists correct data results according to some criteria to relevant query. This method does not require special formula since it lists information by some criteria simply so the calculation amount is little.

**Figure 2: Sequential layout**

In order to present the information on such small screen of mobile device, the allowed text character length is presented. When the text character exceeds the allowed length, it marks omission sign.

3.1.2 Radial Layout

For effective arrangement of information and efficient moves, we apply radial layout for information that has hierarchy. we use circle as radial layout. Such method using circles is called radial layout, the method of Ka-Ping Yee(Ka-Ping Yee et al. 2001) and Battista(Battista et al. 1999) considering mobile device's restrictions is applied. We can apply weight to control display area that each sub tree is allocated according to size of each node in this mode.

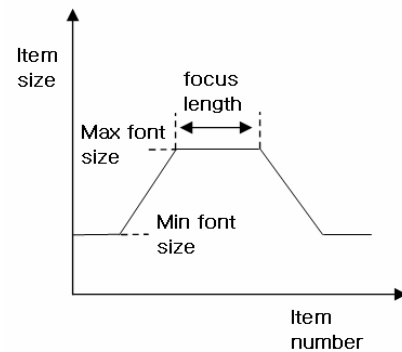
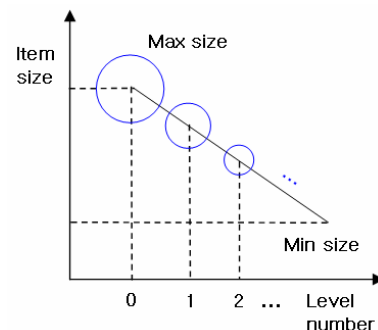
**Figure 3: Radial layout for tree**

3.2 Proposed Visualization Algorithm

Considering CPU calculation ability, memory capacity etc. of mobile devices, our research uses the simple and efficient fisheye view algorithm.

First, we apply method shown that applies fisheye view algorithm to linear menus in sequential layout(Bederson 2000). Figure 4 shows DOI(Degree Of Interest) that illustrates text font size that is decided when applying fisheye view algorithm to sequential layout information.

The following examines visualization method that is applied to radial layout for hierarchical information. Here, the information that user has most interest is located in the center of the circle the information that is located further from the center means the user has less interest. Figure 5 shows DOI of radial layout which displays the center near information bigger and center far information smaller.

**Figure 4: DOI of sequential layout****Figure 5: DOI of radial layout**

3.3 Animation

When fisheye view algorithm is applied to the information in sequential layout and radial, in order to not disorient user's attention, consecutive display of reconstructed information should be shown to the users.

4 Application

This paper developed PocketPC 2002 and 2003 application of Windows CE based using C language to build PDA applications. And cellular phone applications were developed in WIPI(Wireless Internet Platform for Interoperability) environment that is a current mobile standard platform in Korea.

- SequentialMobile : use information created based on the information searched in web through Google
- RadialMobile : use number information that has a hierarchical structure

Information that is used is refined by a person passively. Especially, the information in SequentialMobile displays omission marks for the parts when the space of full information exceeds the length and the width of the screen. The information used in RadialMobile can also be used for the information that can be displayed in graph or tree. The resolution used in test is 280 x 320 in case of PDA and 176 x 220 in case of cellular phone. And each input method used common keys to keep reciprocity compatibility.

Figure 6 shows execution image of SequentialMobile application developed for PDA and cellular phone in each PocketPC and WIPI platform.

A schematic of RadialMobile application is shown in figure 7. With regards to the amount of data processing, the application of RadialMobile is currently limited to PDA applications.



Figure 6: SequentialMobile



Figure 7: RadialMobile

5 Conclusion

This paper applied steps that provide suitable layouts according to information types and steps that apply information visualization algorithm. Also, our approach applied animated effects to improve user's recognition for information. These applied visualization techniques have shown not only that they are fast in informational processes, but they use maximal resources within a given, limited capacity. This paper suggested sequential layouts to support information which is sequential type and radial layouts for hierarchical information such as tree. Also, our approach applied fisheye view algorithm to information arrangement to offer efficient information navigation to users. And in the event of focus information change, we employed animated effects. According to each information form, proposed visualization method helped users to navigate through information more easily. Also, we used common up-down keys to guarantee user's mobility. We need research that supplies various visualization methods according to different types of information in future work. We will compose and supply

information in form that is normalized such as XML. And we will develop agent that applies suitable information layout and visualization algorithm. Also, we will apply rectangular layouts rather than radial layout.

6 References

- Brend Karstens, Matthias Kreuseler, Heidrun Schumann (2003): Visualization of Complex Structures on Mobile Handhelds. *Proc. International Workshop on Mobile Computing*
- Staffan Björk, Johan Redström, Peter Ljungstrand, Lars Erik Holmquist (2000): POWERVIEW: Using Information Links and Information Views to Navigate and Visualize Information on Small Displays. *HUC 2000*, 46-62.
- Ka-Ping Yee, Danyel Fisher, Rachna Dhamija, and Marti Hearst (2001): Animated Exploration of Dynamic Graphs with Radial Layout, *Proc. of Information Visualization*, San Diego, CA, USA, 43-50.
- Bederson, B. B. (2000): Fisheye Menus. *Proc. of UIST*, 217-226, ACM Press.
- Kris Luyten, Karin Coninx (2001): An XML-Based Runtime User Interface Description Language for Mobile Computing Devices. *DSV-IS*, 1-15.
- Freitas, C. M. D. S., Luzzardi, P. R. G., Cava, R. A., Winckler, M. A., Pimenta, M., Nedel, L. P. (2002): Evaluating Usability of Information Visualization Techniques. In *5th Symposium on Human Factors in Computer Systems (IHC)*.
- S. Lok and S. Feiner (2001) A survey of automated layout techniques for information presentations. In *Proc. SmartGraphics Symposium '01*, 61-68.
- John Lamping, Ramana Rao and Peter Pirolli (1995): A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies. In *Proc. of the CHI 95*.
- Xinyi Yin and We Sun Lee (2004): Using Link Analysis to Improve Layout on Mobile Devices. In *Proc. International WWW Conference*, New York, USA.
- Tamara Munzner (1997): H3: Laying Out Large Directed Graphs in 3D Hyperbolic Space. *Proc. of the IEEE Symposium on Information Visualization*, Phoenix, AZ, 2-10.
- WIPI Forum, <http://www.wipi.or.kr/>
- T.J.Jankun-Kelly and Kwan-Liu Ma (2002): Focus+Context Display of the Visualization Process. *Technical Report CSE-2002-13*, Computer Science Department, University of California, Davis.
- G. di Battista, P. Eades, R. Tamassia, and I. G. Tollis (1999): *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall.
- Steve Jones, Matt Jones, Shaleen Deo (2004): Using Keyphrases as Search Result Surrogates on Small Screen Devices. *Personal and Ubiquitous Computing* 8(1): 55-68.

FindFlow: Visual Interface for Information Search based on Intermediate Results

Tomoyuki Hansaki

Buntarou Shizuki

Kazuo Misue

Jiro Tanaka

Graduate School of Systems and Information Engineering
University of Tsukuba,
Tennoudai 1-1-1, Tsukuba, Ibaraki 305-8571, Japan
Email: {hansaki, shizuki, misue, jiro}@iplab.cs.tsukuba.ac.jp

Abstract

FindFlow is a search interface that enables users to construct queries visually on the screen. Because the constructed queries show the process of the search, the user can take a step forward in the search while monitoring the process. The queries can be recombined freely, and previous search results can be reused. In addition, FindFlow shows results interactively for each operation. FindFlow is advantages for searches in which the user repeats trial and error many times.

Keywords: Visual Query, Information Search, Interactive Interface, Visual Interface.

1 Introduction

Searching for information, whether it is for renting an apartment, making a purchase, reserving a hotel room, or various other purposes, is something many of us do in our daily lives. To find the information we want, we have to construct the query appropriately. For example, in searching for an apartment, factors that we might include in the query are rent, location, age, size and distance to a train station. The search is then conducted by trial and error. If the results are not satisfactory, we change some conditions and repeat the search.

Many systems show only results that match the query completely. In many cases, although the result matches the query, we are not satisfied with what the search has returned, but we do not know which part of the query we should change to find what we want. We have to change the query partly or entirely through trial and error many times to find the needed result (Teeven, Alvarado, Karger & Ackerman 2004).

At other times, we may want to reuse a previous search. However, this is often not possible because a system may keep only the last result even if it is used many times. This means we have to construct the same query over and over.

When we repeat a search many times, the query often changes as we get new ideas about how to find what we are looking for. Often the system keeps returning a result that does not really satisfy our original request, but accept it anyway and give up the search.

In any search, the user has to formulate a query, input the query, execute the search, and monitor the results. We think the search system should support

these steps to make the whole search process easier and faster.

We have developed a visual search interface called "FindFlow". FindFlow features a directed graph and supports query construction, trial and error, and reuse. FindFlow shows not only the final result but also the intermediate results, based on which the user can construct or modify queries. We expect that FindFlow will help users find what they the desired results, because queries can be constructed easily, quickly, and adequately.

2 Design

2.1 Support for Constructing a Query

The system should support query construction. A complicated text query is difficult to understand even if we have constructed it ourselves. The system should therefore provide a view of the query that the user can easily comprehend. The system should also provide functions for easy and fast query construction. The user has to repeat search steps many times during a search. The system should therefore offer support to reduce search time.

2.2 Support Trial and Error

Since the user repeats trial and error in a search, the system should support this process. When a result is unacceptable, the query has to be modified, but, on the basis of only the final result, it is difficult to determine exactly how the query should be changed. If the system shows how each condition in the query influences the results, the user will know how to obtain better results. The system should also support search with multiple queries at the same time. The user changes the query and searches many times, when the user considers how a certain query might return a good result and changes the wording of the query accordingly. If searches could be performed using multiple queries at the same time, it would not be necessary to repeat the search steps so many times.

The system should show the result interactively so that the user does not have to do execute a search after each operation, and can confirm the result immediately.

2.3 Support for Reuse

Sometimes, the user wants to reuse a previous search when starting a similar one. The system should therefore support the reuse of previous searches.

3 FindFlow

FindFlow is a visual search interface that allows users to construct queries visually using a dataflow diagram

(Fig. 1).

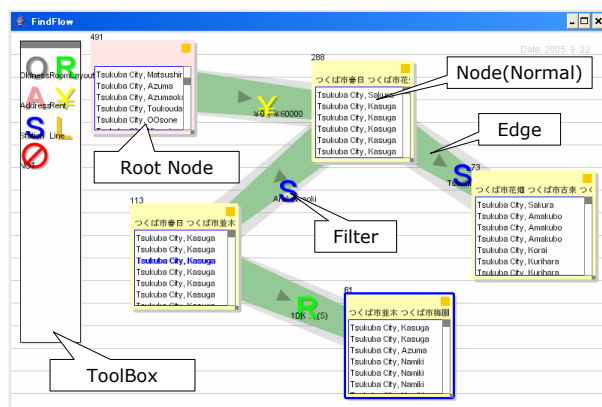


Figure 1: Screenshot of FindFlow: The user is searching for an apartment in Tsukuba.

Several visual interfaces that allow users to construct queries on the screen have been developed (Young & Shneiderman 1993, Guo 2003, Angelaccio, Catarci & Santucci 1990, Batini, Catarci, Costabile & Levialdi 1996, Catarci, Costabile & Levialdi 1993, Krishnan & Kunii 1991). We chose a visual interface based on a dataflow diagram because the whole process can be represented in one diagram, with makes things easy to understand, and because it is easy to compose queries with it (Tanimoto 2003).

FindFlow shows a visual query that describes both the query conditions and results, so that the user can know how the query should be changed. It also shows the results interactively after each user operation, and the user can construct the query while monitoring results.

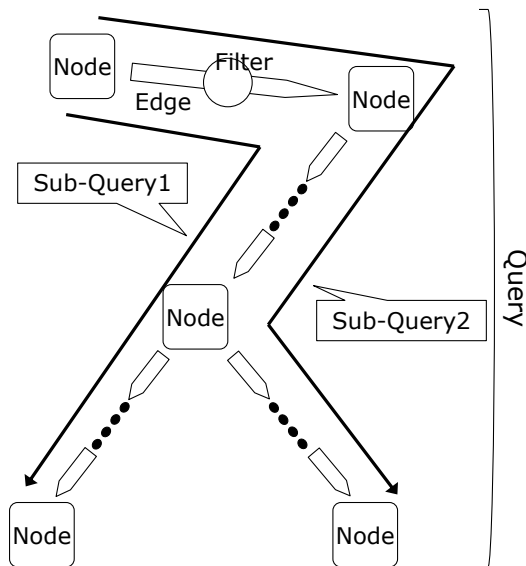


Figure 2: The query and a sub-query.

In FindFlow, a query is constructed using an edge, node, and filter. Various queries can be constructed at the same time (Fig. 2). The part of the query from the root node to another node is called a sub-query. The user can execute various searches at the same time using sub-queries.

3.1 Compositions for Constructing Query

Edge The edge connects two nodes and sends data from one node to the other. An arrow shows the direction the data is sent. A filter on the edge filters data from the sending node, and the thickness of the edge indicates how much data has been filtered. The thickness of the edge changes depending on the amount of data in both the sending node and the receiving node. This helps the user know how the filter works and determine how the query should be changed.

Node The node receives data from edges and shows the received data. The node shows headings for the data in a list and highlights frequently appearing words. The order of data shown in the list depends on the filters (described later). The user can know what kind of data the node contains and whether the filter is appropriate or not. If the filter is not appropriate, the user changes or rearranges the compositions. If the node is too small to see the result, the view can be enlarged by dragging from the right bottom corner of the node.

There are two types of nodes: a normal node, which is yellow, and a root node, which is red. The root node contains all data from the search. The user connects some normal nodes to the root node and starts the search.

Filter To help the user construct queries, the filter is shown as an icon on the screen. By dragging the icon, the user can add, remove, or exchange it anytime. The user can set up the filter on the edge by moving the icon to it. When the mouse pointer touches the edge as the icon is dragged, FindFlow interactively shows the results of applying the filter. This helps the user can determine how to construct a query while confirming the results. The filter shows the setting window when the icon is clicked. The user can decide how the filter will work in detail by changing the settings. Figure 3 shows a rent filter. The user can change the rent criteria by moving the scrollbar, which is shaded according to the amount of matched data. The scrollbar shows the results by shading, which supports trial and error.

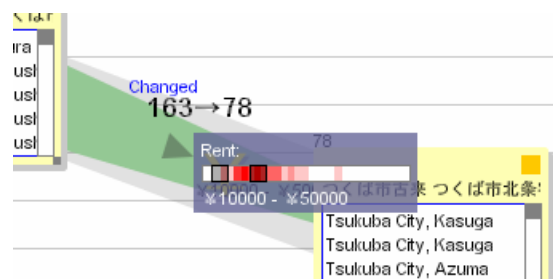


Figure 3: Setting window of rent filter

3.2 Operations

FindFlow's simple operations help make query construction easy. FindFlow shows the results interactively after each operation.

Creating a Node To create a node, the user simply clicks the right top button of a node.

Connecting Two Nodes Figure 4 shows the operation for making connections. To connect existing nodes, the user clicks the right upper button

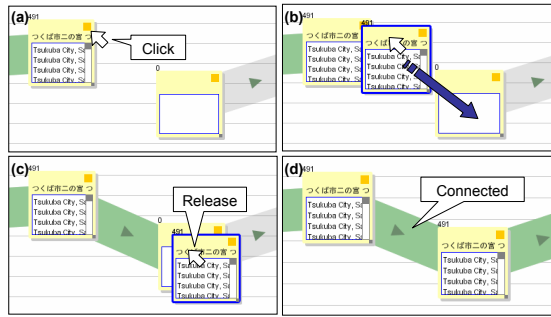


Figure 4: Operations for connecting nodes.

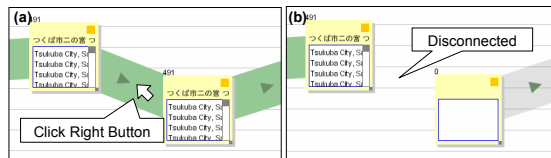


Figure 5: Operations for disconnecting nodes.

of the sending node (a). The new node is created (b), and the user drags it to the receiving node (c). FindFlow makes the edge work as the connection (d). When the connection is made, the sending node sends the data to the receiving node.

Disconnecting Two Nodes Figure 5 shows the operation for disconnecting nodes. To disconnect two nodes, the user moves the mouse pointer to the edge and clicks the right button (a). The connection between nodes disappears immediately, and no further data is transferred. (b). Any data in the receiving node is expunged.

Deleting a Composition In revising a query, the user may want to delete compositions. The operation for deleting composition is the same as the disconnecting operation. The user moves the pointer to the composition and right-clicks the mouse.

Making Divergences and Junctions The user can make divergences and junctions in a query. When setting up different compositions for a diverged query, the user can push different searches forward at the same time. This helps the user search with multiple queries at the same time. Figure 6 shows a divergence and junction. When the user makes a divergence, the divergence node sends the same data to each receiving node. When the user lets some queries combine, the junction node merges the data from each edge.

The operation for combining or diverging queries is the same as the one for making connections.

3.3 Weight of filters

The node shows lists the results and sorts the data based on the weight of placed filters on the query. The weight of the filter used near the root node increases, because it is reasonable to assume that users construct queries in the order they believe to be important.

By assigning a weight to filters, FindFlow supports the user in finding the needed data.

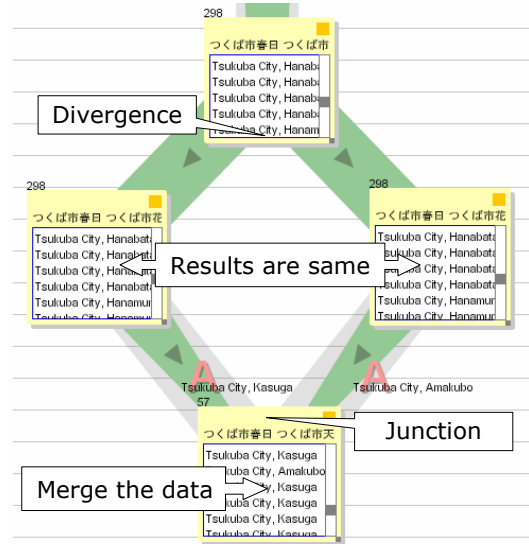


Figure 6: Divergence and junction

3.4 Step-by-Step Indication

FindFlow indicates how each filter influences the results in a step-by-step manner. When the user changes constructed query, the thickness of the edges changes and nodes change their result step by step from the changed composition to the end of query. This supports the user in finding how the operation influences the result, and makes it easy for the user to understand how data are filtered.

3.5 Packaging

FindFlow has a function for compiling sub-queries and generating new filters. This function helps the user in constructing the queries that will be used continually. Once it is made, the package can be reused many times. When a query becomes long, this helps the user in understanding the query.

Figure 7 shows the operation for packaging. To use the packaging function, the user drags the node to the toolbox and drops it (a-1). Then, a new package is generated as a filter (a-2), and the sub-query from the root node to the dropped node is compiled into the new package. The user names the generated package, for instance, "MySearch" (a-3). The user can use the new package just like a filter. After dragging the package to the edge (b-1), the user obtains the same result as the result without a package (b-2).

3.6 Save the State of Searching

FindFlow provides a function for saving the search status in a file. This supports the user in reusing previous searches.

Sometimes the user starts a search is similar to a previous one. If the search can be continued from the previous one, the user does not have to waste time constructing similar queries.

FindFlow saves the search status automatically. To continue a search, the user has to do is choosing the file. FindFlow reads the file and restores the previous search immediately, and the user can start the new search immediately.

4 FindFlow System

Figure 8 shows the structure of the FindFlow system. FindFlow consists of an interface module and a

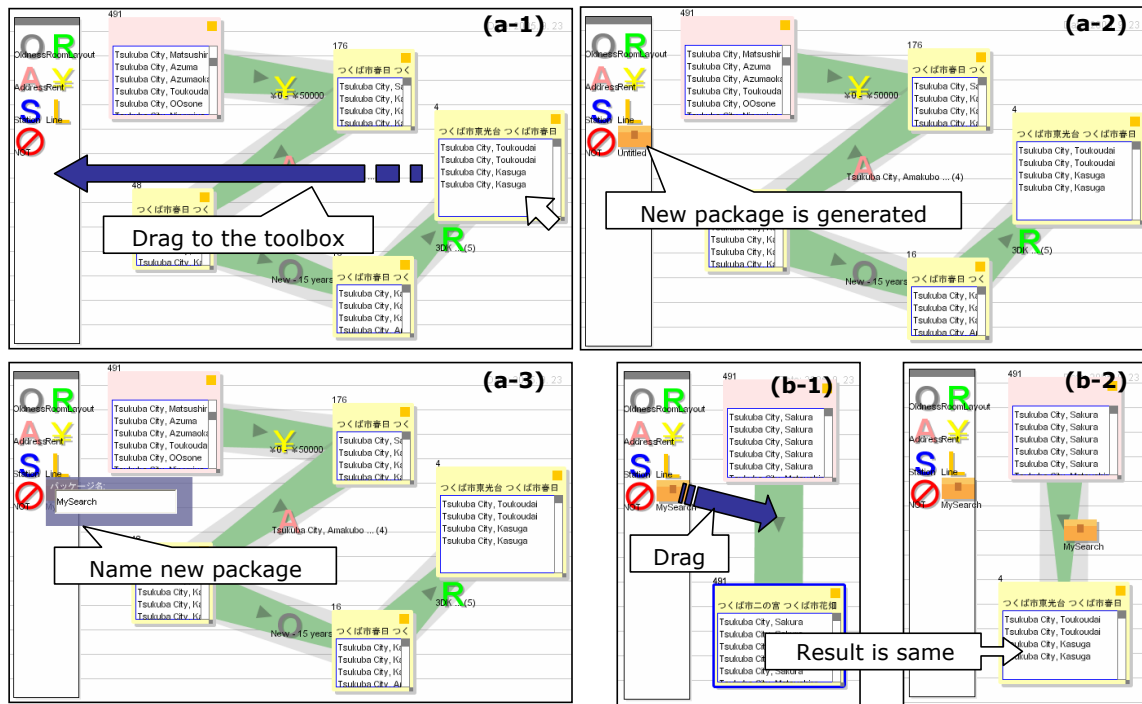


Figure 7: Packaging

database module.

The interface module is operated by the user and it sends queries to the database module. After receiving the result from the database module, it shows it to the user. The interface module comprises the edge, node, and filter. The user manipulates each to construct a query. The database module receives the query from the interface module and converts it into the database query. After conversion, it sends the query to the database and receives the result. Then, it converts the received result into the data for the interface module and returns the converted data as the result.

When the user operates FindFlow, each node requests the query data and weights of filters from the sending nodes. The sending node sends these data, and each edge creates new query data and new weights from the filter and the received data. After receiving the new query data and weights, the node sends the query data to the database module. The database module receives the query data and converts it into a database query. It then sends the converted query to the database server and receives the result. Database module converts the received result for the interface module, and returns the converted result to the node. After receiving the result from the database module, the node sorts the data based on new weights of filters. After that, the interface module shows the result to the user by changing the result in the node, the thickness of edges, and so on.

5 Application Example

5.1 Searching for Apartments in Tsukuba by Trial and Error

We made a search system for rental listings in the city of Tsukuba. We prepared filters for age, layout, address, rent, and station proximity.

Figure 9 shows screenshots of a user searching for apartments in Tsukuba. The criteria are:

- Location is “Amakubo”, “Sakura” or “Kasuga”.
- As new as possible.

- Rent as low as possible.

First, FindFlow shows the screen (a), and the user constructs a query specifying 70,000 yen rent and Amakubo, Sakura or Kasuga as the location (b). Many results are returned, and the user changes the rent filter, making the rent lower little by little. The user finds that the lowest rent is 30,000 yen (c).

The user focuses attention on the Kasuga area. He wants to push forward the search, but every apartment found is older than 30 years. He therefore stops filtering data (d) and decides to reconsider the whole query.

The user rearranges the query. The user inserts a filter with the condition that apartment should be less than 15 years old (e). He changes the rent filter to 50,000 yen and finds matches with the query (f). He then changes the age condition by requesting a newer apartment (g), and finds the apartment to match the query (h). This listing meets the users requirements.

5.2 Classification of Mails and Files

By regarding the node as a classification folder, FindFlow can serve as a classification tool.

Suppose the user wants to search for or classify some e-mails. FindFlow generates filters automatically from mail headers. The user classifies e-mails just like in searching and saves the search status after finishing the classification. The user loads newly received e-mails to FindFlow. After loading the saved status, FindFlow classifies new e-mails by the query automatically. When the classification does not go well, the user changes the query partly or entirely.

FindFlow is useful for classifying files, too. The user can put files in one place and does not have to classify files with traditional directories. With FindFlow, even when new files are added, the user does not have to be remember where files are placed. FindFlow automatically classifies files by the queries. Because the search status has been saved, the user can load the same status and can find needed files anytime.

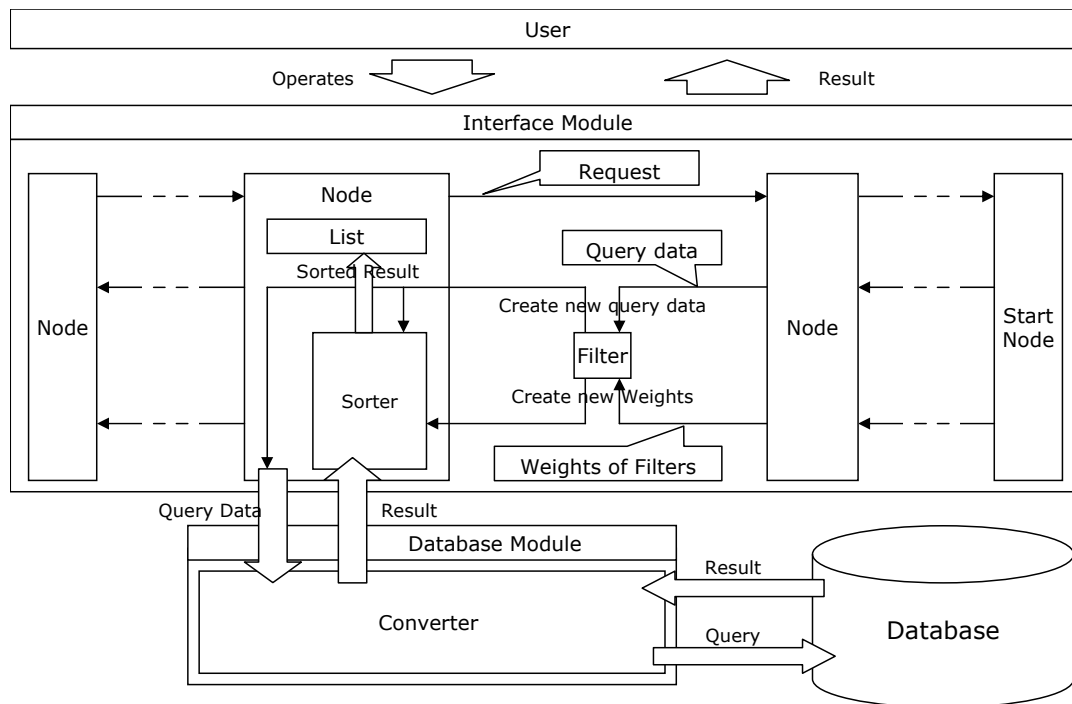


Figure 8: System

6 Discussion

We had various users try FindFlow. The users were businessmen, students, or researchers. After the trials, the users were interviewed. The users found that the operations were simple and easy to understand. They also reported that it was easy to determine which conditions were impeding the search and that it was good that the result was simple. On the negative side, they mentioned that the screen was too small and that, although the node shows the result, it was too small to see and did not contain enough information.

The screen size problem could be solved by providing a scrolling function. Scrolling would allow the user to get more space for query construction. An automatic zooming function would also be useful. When constructed queries reach the edge of screen, FindFlow changes the scale. A zoom would enable users to zoom in on the screen near the mouse pointer.

7 Conclusions and Future Work

We have developed an interactive visual search interface called FindFlow. FindFlow shows not only final results but also intermediate results. The user can change a query based on the intermediate results. FindFlow supports query construction, trial-and-error search, and reuse, enabling the user to construct, recombine and change queries easily, quickly, and properly. FindFlow can be applied to various kinds of searches.

We plan to evaluate FindFlow experimentally to clarify its advantages and problems. We are also considering how to better use FindFlow in cooperation with the Web and how to improve the methods for showing data.

References

- Jaime Teeven, Christine Alvarado, Devid R. Karger & Mark S. Ackerman (2004), The Perfect Search Engine is Not Enough, *in* 'Proceedings of Conference on Human Factors in Computing Systems', pp. 415–421.
- Angelaccio, M., Catarci, T., & Santucci, G. (1990), QDB: A Graphical Query Language with Recursion, *in* 'IEEE Transactions on Software Engineering', pp. 1150–1163.
- Batini, C., Catarci, T., Costabile, M.F. & Levialdi, S. (1996), Visual Query Systems: A Taxonomy, *in* 'Proceedings of the 2nd IFIP WG2.6 Working Conference on Visual Databases'.
- Catarci, T., Costabile, M.F. & Levialdi, S. (1993), On Visual Representations Database Query Systems, *in* 'Proceedings of the Interface to Real and Virtual Worlds Conference', pp. 273–283.
- Deepa Krishnan & Tosiya L. Kunii (1991), A Visual Query Interface for an Engineering Database, *in* 'Database Systems for Advanced Applications'.
- Aiken, A., Chen, J., Lin, M., Spalding, M., Stonebraker, M. & Woodruff, A. (1995), The Tioga-2 Database Visualization Environment, *in* 'Proceedings of the IEEE Visualization Workshop'.
- Degi Young & Ben Shneiderman (1993), A Graphical Filter Flow Representation of Boolean Queries: A Prototype Implementation and Evaluation, *in* 'Journal of the American Society for Information Science', Vol. 44(6), pp. 327–339.
- Diansheng Guo (2003), A Geographic Visual Query Composer (GVQC) for Accessing Federal Databases, *in* 'Proceedings of National Conference for Digital Government Research', pp. 397–400.
- Steven L. Tanimoto (2003), Programming in a Data Factory, *in* 'Proceedings of Human Centric Computing Language and Environments', pp. 100–108.

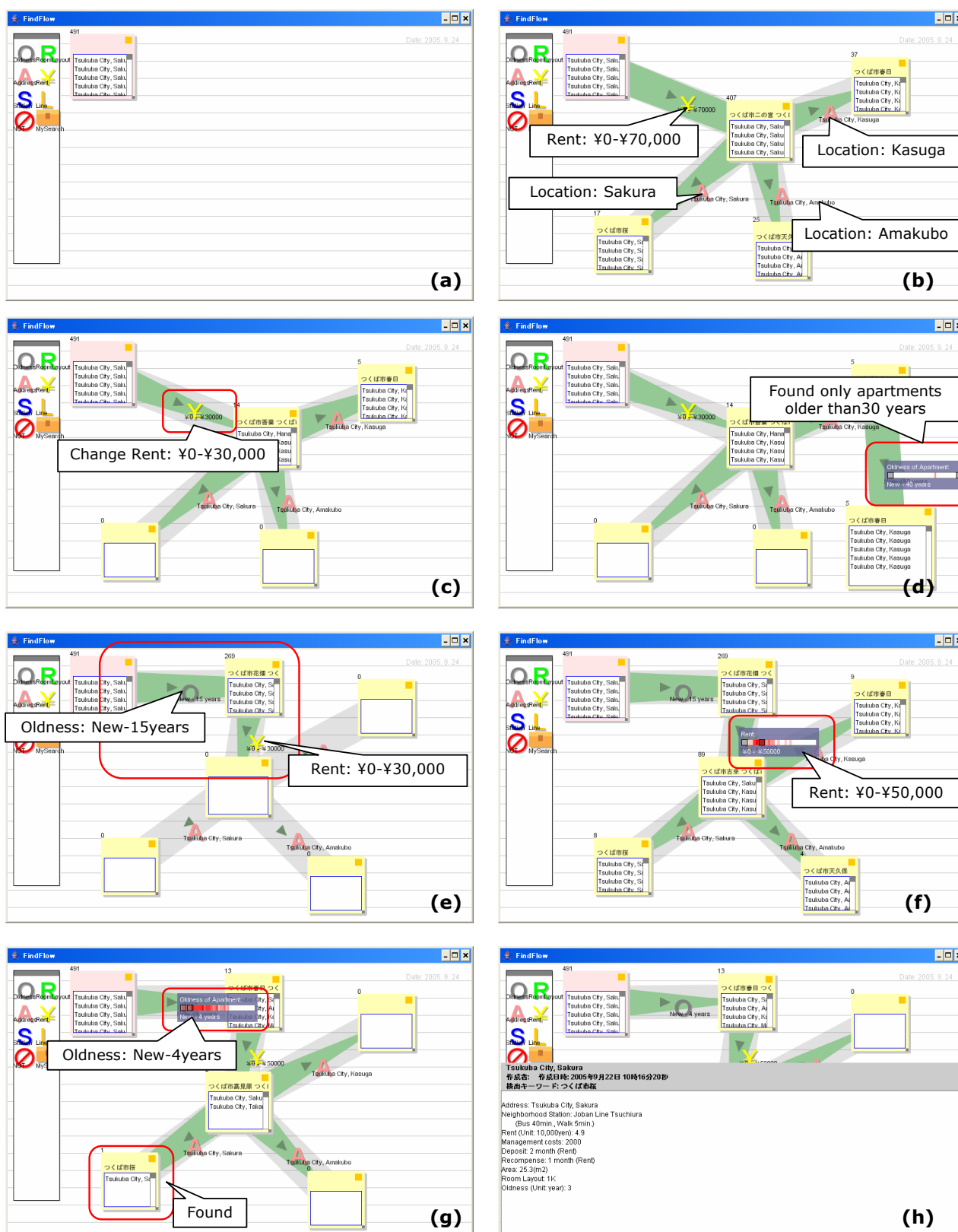


Figure 9: Example of search with FindFlow: Finding an apartment in Tsukuba.

Ripple Presentation for Tree Structures with Historical Information

Masaki Ishihara Kazuo Misue Jiro Tanaka

Department of Computer Science,
University of Tsukuba

1-1-1 Tennoudai, Tsukuba, Ibaraki, 305-8573, Japan

{ishihara, misue, jiro}@iplab.cs.tsukuba.ac.jp

Abstract

We propose a new method for representing tree structures with historical information. We call this method Ripple Presentation. Categories of nodes are represented by the angles of edges and elapsed time is represented by the length of the edges. In this way, the method can express both the time series and categories, which has been difficult to achieve with either tree structures or lists. As a result, users can understand the overall information from their viewpoint view and discover target information effectively. We applied the method to trackback links of Weblog articles and the latest articles of News sites using RSS on Web as a resource.

Keywords: Information Visualization, Graph Drawing, Historical information, Tree structures.

1 Introduction

Finding information on the Internet can be a very time-consuming task. Even if we can obtain a list of relevant Websites using a search engine, the latest information is often difficult to find. This is particularly problematic when there is an urgent need for the latest information.

Recently, an increasing number of Websites distribute their content using RSS (Rich Site Summary). RSS provides indexes and summaries of Websites and updated information. Updated information from multiple Websites can be easily obtained using tools such as RSS Reader and News Reader.

However, information is often displayed using a tree or list in a time-series order, so finding the exact information from the perspective of category and time still takes time. We believe that there is need for and improved and more intuitive approach to finding relevant and up-to-date information.

In this paper, we address the problem of effective discovery of requested information by introducing

approach of visualizing a large amount of information on the Internet.

The tree structure is widely used as a method of classifying information. It is commonly used for computer directories, the class structure of JAVA, organization charts, and distribution diagrams. The advantage of the tree structure is its ability to organize information in categories.

Once the root is decided, the tree structure becomes a directed graph. For instance, each hierarchy shows the unit of organization in organization charts or the age in distribution diagrams.

The weakness of the tree structure is that the latest information is always laid out at the bottom of the hierarchy because of its drawing convention when the hierarchy means time-series. Furthermore, the link relations between leaf nodes are not fundamentally expressible due to a tree structure that does not have closed loops.

In a tree structure, information can also be sorted by time. However, it then becomes difficult to organize the information in categories.

We propose an expression method called Ripple Presentation. As the name suggests, as time passes, older information (historical information) is moved outwardly away from a central point, where new information is positioned first. Though the targeted data structure is the tree structure, our method differs from the past method in two points:

1. The latest information is positioned in the neighborhood of parent nodes.
2. The relationships between child nodes are indicated by the angles of edges from the parent nodes.

2 Outline of Ripple Presentation

The concept of Ripple Presentation is that older information is positioned far away from and newer information closer to the user's viewpoint.

In the expression of a tree structure with a time-series hierarchy, such as in a distribution diagram, the hierarchy of the time series is normally defined in a parallel manner from the root node to leaf nodes as shown in Fig. 1(a). In Ripple Presentation, concentric circles (ripples) whose

centers are the parent nodes of each subtree define the hierarchy as shown in Fig. 1(b).

Furthermore, the arrangement of the hierarchy is different in the two methods. In a tree structure with the time series, the hierarchy is generally arranged from the root node toward leaf nodes in ascending order. In our approach, it is arranged from leaf nodes toward the parent nodes of each subtree in the same order. Only the position of the root nodes is decided at its creation time. In other words, the arrangement of the time-series hierarchy is opposite to the arrangement of a tree structure with a time series.

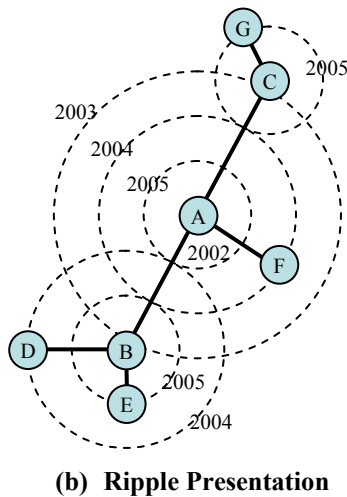
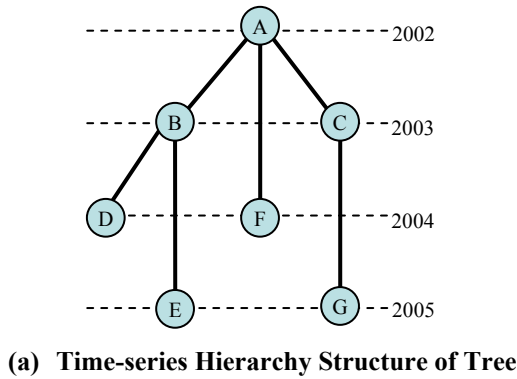


Fig. 1: Comparison of Time-series Hierarchy Structure with Ripple Presentation for Tree.

Another analogy in the natural world is the annual rings of a tree. An annual ring represents one year of the growth cycle, and the interval between the ripples is proportional to the reciprocal of the frequency of the wave. Although the cycles vary, the common point is that the interval has a constant cycle (assuming that there is no attenuation) and its shape is a concentric circle.

In Ripple Presentation, the cycle of each ripple is constant. Assuming that the attribute time of the node is the same as the creation time of the ripple, the node is positioned on the spreading ripple because the ripples are synchronized with the nodes. The parent nodes are positioned at the center of the ripples, and the ripples are generated at certain constant intervals as concentric circles. That is, all nodes on the same ripple means the nodes were generated at the same time.

Thus, once the time cycle of each ripple is understood, the history of the tree from some point in time to the oldest time is understood. In addition, it is only necessary to pay attention to the neighborhood of the generation points of the ripples (parent nodes) in order to know the latest information (leaf nodes).

3 Layout Design

This section explains how to lay out the tree structure data by Ripple Presentation in detail. First the targeted data is modeled and the layout design is mathematically explained using a graph model.

3.1 Rooted Tree with Historical Information

Our system targets a rooted tree with historical information. Suppose that $G = (V, E)$ is a rooted tree. Sets $E \subset V \times V$ of edges and sets V of nodes compose G . Sub trees described with G_{sub} are rooted trees composed of only the leaf nodes as the child nodes. Each node $v \in V$ of G includes some attributes such as category and creation time in the system. In this case, V represents the time series data. The creation time of v is described as $d(v)$.

3.2 Drawing Conventions

According to the strict definition (K. Sugiyama 2002) drawing conventions are composed of placement conventions for nodes and routing conventions for edges.

3.2.1 Placement Conventions for Nodes

In Ripple Presentation, the placement convention for G_{sub} is a *concentric circle*. The placement convention for G is a *recursive concentric circle*, since G is made up of G_{sub} as a nested structure.

3.2.2 Routing Conventions for Edges

The routing conventions are explained by line type and the relationship with the coordinate system. In Ripple Presentation, the line type is *straight line routing* and routing is *independent of coordinate system*.

3.3 Drawing Ripples

Ripples start generating when child nodes are created. That is, assuming that the oldest node (the most outside node) in all nodes except for root node that composes a subtree is v , ripples start generating at time $t_0 = d(v)$.

The ripples are not only generated synchronously with child nodes but also at a constant frequency. Therefore, by viewing the ripples, users can be aware of the elapsed time to accommodate the length of the edges. More accurately, from the projection lines (time scale) provided by the ripples, users can understand visually the scale of the elapsed time.

Suppose that $\{c_1, c_2, \dots, c_n\}$ is a set of ripples centered on a parent node of subtree G_{sub} . At time t , the radius $r_1(t)$ of ripple c_1 (the most outside ripple) generated at time t_0 is expressed by

$$r_1(t) = \lambda \cdot f \times (t - t_0) \quad (3.1)$$

where f is the frequency of the ripples and λ (common parameter in all the ripples) is their wavelength. Then, the number of ripples n is

$$n = \left\lceil \frac{r_1(t)}{\lambda} \right\rceil \quad (3.2)$$

Therefore, at time t , the radius $r_i(t)$ of ripple c_i (inside ripples from c_1) generated after time t_0 is defined as

$$\begin{aligned} r_i(t) &= r_1(t) - \lambda(i-1) \\ (i &= 1, 2, \dots, n) \end{aligned} \quad (3.3)$$

3.4 Layout of Nodes

A polar coordinate system is used for Ripple Presentation as shown in Fig. 2. To put it more concretely, the length and the angle of the edges are required as parameters at the time of drawing G , and then, based on these parameters, the position of each node is decided. For example, position $p(v)$ of child node v is defined by polar representation $(l_v(t), \theta)$, where the length of edge $e \in E$ is $l_v(t)$ and the angle is θ . That is, positions of all nodes are relatively decided when the position of the root node is decided.

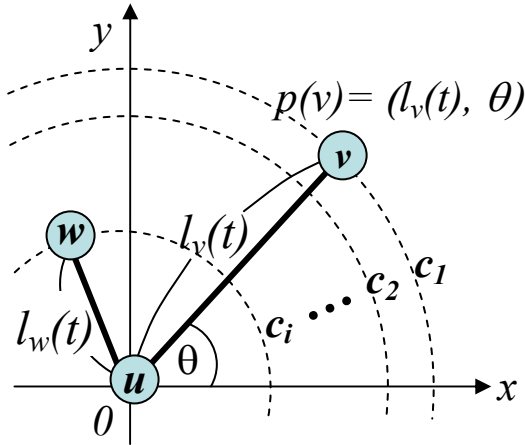


Fig. 2: Polar Representation of Child Nodes.

3.5 Length of Edges

At time t , the length $l_w(t)$ of the edge from parent node $u \in V$ to child node $w \in V$ is defined as

$$\begin{aligned} l_w(t) &= r_1(t) - \lambda(j-1) \\ (j &= \lceil f \times (d(w) - t_0) \rceil) \end{aligned} \quad (3.4)$$

Noteworthy is that expression (3.4) shows that the lengths of each edge synchronize with the radius of the ripples depending on frequency f of ripples.

3.6 Angles of Edges

How to decide angle θ is entrusted to the applications.

The angles of edges could be decided from the relation between the parent nodes and each child node. The parameter used for drawing would then be the category attributes of the nodes. Assuming that the content of each node is a news article, categories such as "Society," "Politics," or "Economy" could be obtained from the content of the article. The angle of each edge having child nodes with the same category is set within a certain constant range of the angle.

In the management of the directory and bookmarks, folders (parent nodes) are made in categories division, and the files (child nodes) belonging to the categories in the folder are stored. In this case, each hierarchy of the tree structure is arranged not as a time series but as categories.

In Ripple Presentation, we treat each hierarchy as a time series. The categories mean angles. That is, our layout method can express both the implicit links of leaf nodes (clustering) and the time series.

4 Implementation

4.1 System Architecture

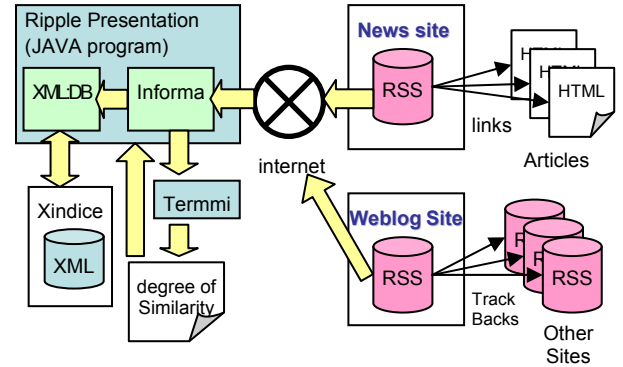


Fig. 3: System architecture.

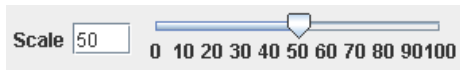
In our systems, we adopt Java2 SDK 1.4 as the development language and use Xindice1.0 for the native XML database. The Informa library is used for parsing the RSS feed acquisition. If there is no category description in the RSS feed of articles, the degree of similarity among articles is calculated with the external tool "Termmi," so the angle of edge of each article is decided based on this.

First, the user sets the URL as the location of RSS or trackback URI of the Weblog article to the root node, and the system collects linked information. This information is then converted into XML data, which is registered in the database. Finally, the system draws the graph and

ripples based on this XML data in the view. The animation is drawn by synchronizing two threads for ripples and the graph.

4.2 Feature Description

Users can adjust each parameter for an easy-to-view layout while drawing it in the view by animation. As a result, a view is obtained that is well adapted to the characteristics of the data or to the user's viewpoint. Concretely, three parameters are mounted on the upper portion of the view screen and the users are able to control slider (Fig. 4).



(a) Scale Slider



(b) Speed Slider



(c) Time interval Slider

Fig. 4: Sliders for parameter setting.

An example of using these operations is shown in Fig. 5. The viewing time in Fig. 5(b) is 9 days after that in Fig. 5(a). Figure 5(c), which has the same view time as Fig. 5(b), shows a view that has been zoomed out and whose time interval has been changed from 1 hour to a day. Thus, various views can be obtained by changing each parameter according to the purpose.

4.2.1 Scale

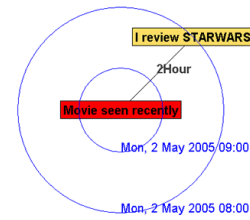
This system has a function that changes the scale parameter of the view. In terms of the model, this parameter reflects the wavelength λ in expression (3.1). Concretely, the view can be controlled using a scale slider to zoom in or out. The default setting of the scale parameter is 50.

4.2.2 Speed

The second function is a controller for the progress speed of the view. This parameter is proportionality constant for progress speed of time t in description of section 3. The default setting is 1.

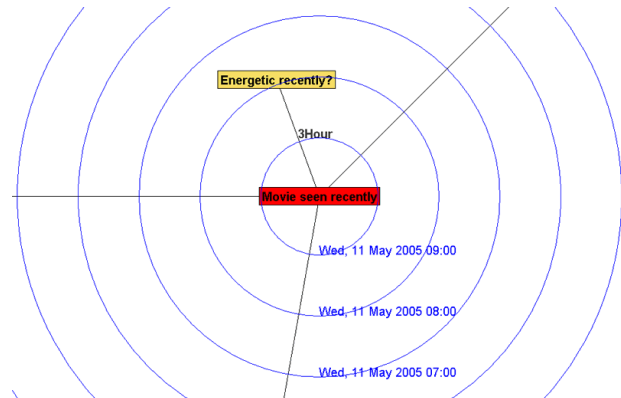
4.2.3 Time interval of Ripples

The third function is a controller for the time-interval of each ripple. This parameter is reflected in the frequency f of ripples in expression (3.1). The initial setting for this parameter is 1 hour (for instance, if a certain ripple is "9 May 2005 10:00," the following ripple is "9 May 2005 11:00," that is after one hour). The time-interval of ripples can be adjusted from 1 to 24 hours (1 day) by 1 hour and from 1 day to 7 days (1 week) by one day.



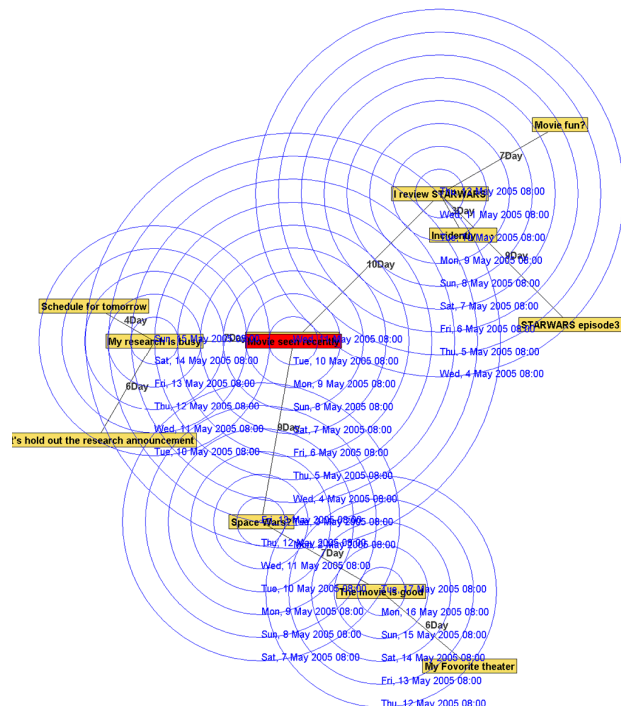
(a) Scale: 50. Time-interval: 1 hour.

(View-time: 2 May 2005 9:00 AM)



(b) Scale: 50. Time-interval: 1 hour.

(View-time: 11 May 2005 9:00 AM)



(c) Scale: 25. Time-interval: 1 day.

[View-time: the same as (b)]

Fig. 5: Control flow of parameter setting.

5 Application Example

To demonstrate the utility of Ripple Presentation, we applied it to news articles on a news site, which simply have subtree structure with historical information. In this case, it is possible to decide the graph by comparatively simplifying the angles, because the category is assigned to each article beforehand as an attribute.

We also applied it to trackback articles on the Weblog site, which have rooted tree structure (composed by nested subtree) with historical information. In this case, angle of each edge is decided by the relative similarity between each article after analyzing its contents. Note that ripples are drawn recursively in this case.

5.1 Visualization of Articles on News Site

The information about the latest news with RSS is delivered every day on the News site. Users can watch the latest news or update information easily using the News Reader to collect RSS feeds.

The News Reader often displays news articles using a list with a time series. This presentation method is seen in various scenes, such as Mailer and file management. In general, the users see the titles of articles in a list and searches out a target articles.

In Ripple Presentation, it is possible to refine the search for the latest on-demand news because the articles are categorized by content before users open them in the browser. In addition, we also think that this expression method is useful in analyzing not only the latest article, but also the trends in the news during a certain period and the frequency of the news.

We used RSS (217 articles in total) delivered by the news site "U.S. News and World Report"¹ on September 26, 2005 as data.

The news articles visualized by the Ripple Presentation are shown in Fig. 6. In this case, the root node represents the homepage of the site, and each node is a news article. Each article is categorized by content.

Let us assume that the users want to see articles related to the culture category from the news delivered between 0:00AM, September 11, 2005 and 3:00 AM September 14, 2005. At this time, the users should pay attention to articles appearing in the direction of the angle of the cultural category. In the first step, Fig. 7(a) shows the view where the time interval of the ripples is one week. In the next step, when the time interval of the ripples will be changed to 12 hours, and the root node neighborhood is zoomed, Fig. 7(b) will become the current view. That is, in order to view the latest article from those that appeared during that week, one simply chooses the article nearest the root node and in the culture category.

5.2 Angles of Edges

In Ripple Presentation of the news site, the angle of each edge are divide up evenly in 360 degrees by the number of articles. In addition, all nodes are arranged by category and then, the nodes included every category are arranged by time-series. Therefore, all nodes are sequentially arranged by the time series in the range of the angle of each category.

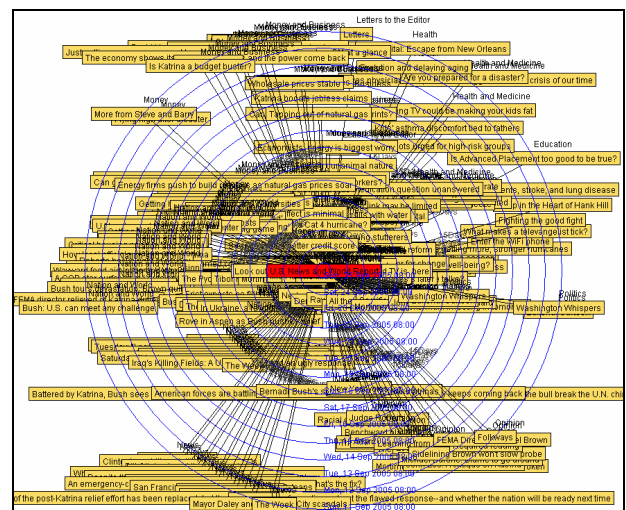


Fig. 6: News list with Ripple Presentation.

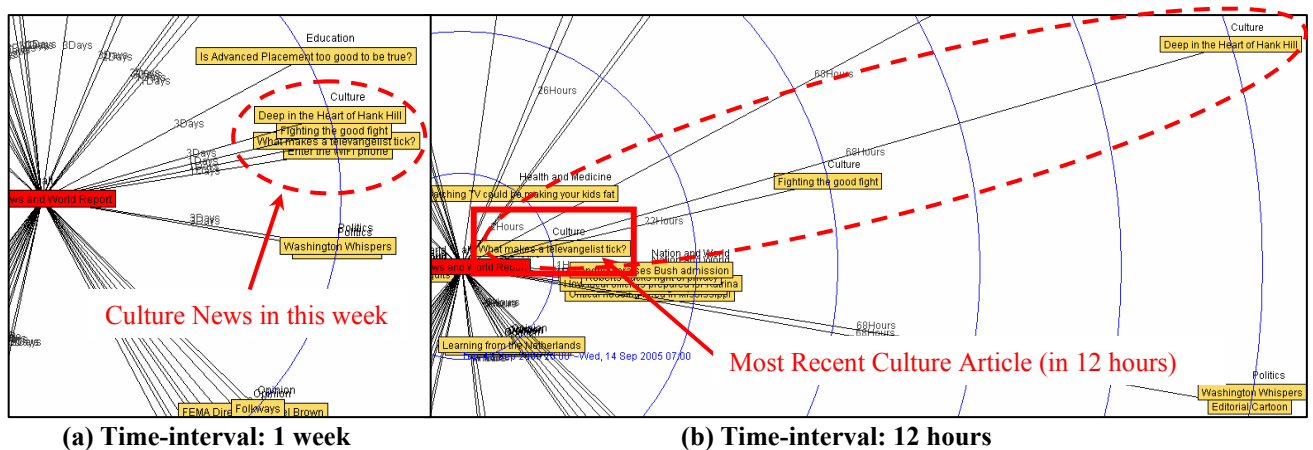


Fig. 7: Discovery of latest news articles in the culture category.

¹ <http://www.usnews.com/usnews/usnews.rss>

5.3 Visualization of Trackback on Weblog

In a nutshell, *trackback* was designed to provide a method of notification between Weblog sites: it is a method by which person A says to person B, "This is something you may be interested in." To do that, person A sends a *trackback ping* to person B. The URL of the article is described in the RSS feed of the Weblog site of person B by this *trackback ping*. Although the way of understanding the links of the trackback can do nothing but trace the link up to the present time, Ripple Presentation visualizes these links based on RSS.

The structure of the traceback links is different from structure of news articles. The traceback has continuous branching links. That is, the ripples are recursively drawn, because the link structure is a rooted tree.

The links of the trackback to the Weblog article “Google launches Blog Search1” are shown using Ripple Presentation in Fig. 8. The display period was from September 14 5:00 AM to September 18 5:00 AM. The total was 26 articles (until September 30, 2005). Overlapping articles were excluded.

5.4 Angles of Edges

When the traceback of Weblog is visualized, it is necessary to calculate the angles of edges from the relation between each traceback article and the parent article. The procedure is as follows.

1. An important key word is extracted by the *TF/IDF (Term Frequency / Inverse Document Frequency) method* from each traceback article including the parent article.

2. The score of the importance of the article is calculated by the *Vector Space method* from this important key word.
3. The relative difference of the importance score is defined as the degree of similarity between each traceback article and the parent article, and it corresponds to the angle of each edge.

We use the text-mining tool "Termmi" for this purpose (Fig. 3).

If the degree of similarity between the parent article and the trackback articles (child articles) is high, the edges move to the top of the screen. Oppositely, if the degree of similarity is low, they move to the bottom of the screen. Moreover, from the point of view of the parent article, the articles that include unique content are shown on the right side of the screen and those of general content are shown on the left side (Fig. 9).

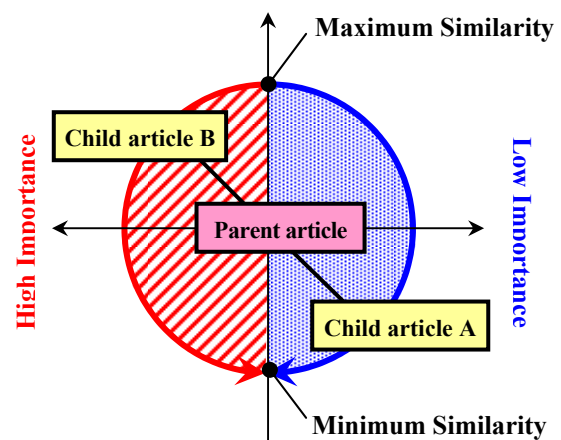


Fig. 9: Angles of edges.

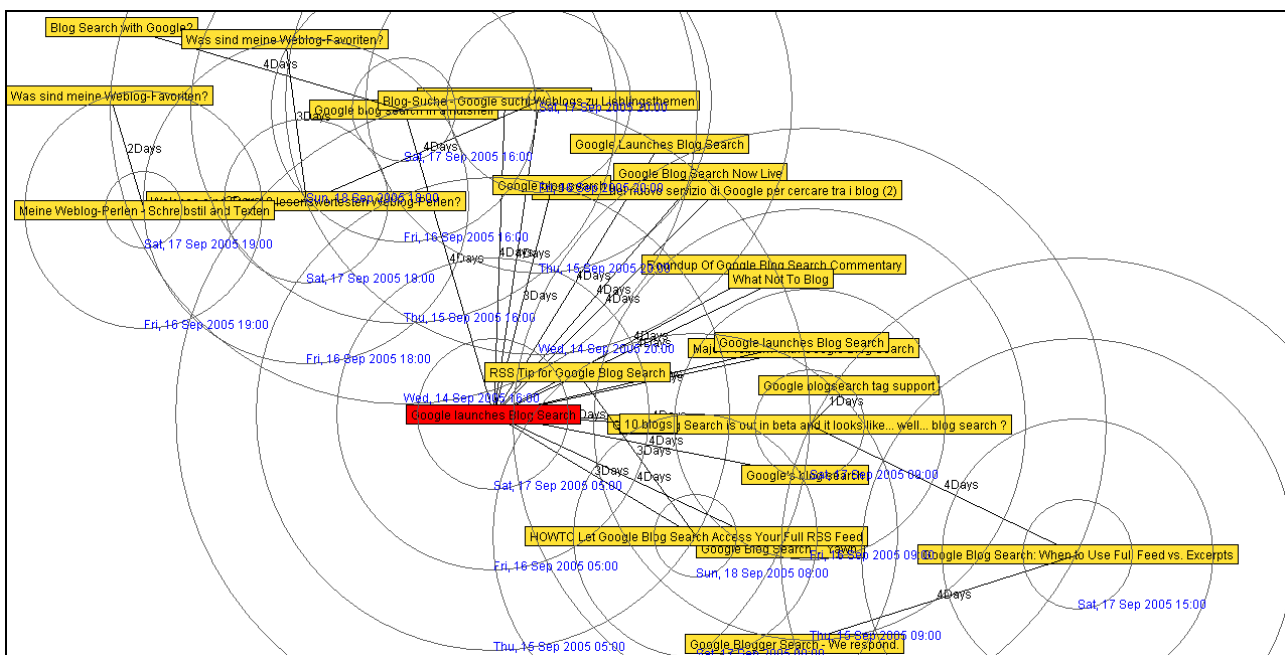


Fig. 8: Backtrack links of Weblog articles “Google launches Blog Search²” with Ripple Presentation.

² http://www.sixapart.com/pronet/weblog/2005/09/google_launches.html

6 Related Work

A lot of research has been performed for the visualization of the tree structure and many systems have been developed.

A typical tree structure expression technique is the Hyperbolic Tree (Lamping, J., Rao, R. 1993), where the tree diagram is arranged on a hyperbolic space. A node positioned far from the center is displayed smaller, while a node positioned near the center is displayed bigger. This expression method considers the aspect to which the user is paying attention.

Our Ripple Presentation also considers the user's viewpoint. Concretely, the root node is positioned in the center, and the latest node is arranged in the root node neighborhood.

Other research has examined other hierarchies of the tree structure. Tree-Maps (Johnson, B. and Shneiderman, B. 1991) uses space-filling recursive division of a rectangular area as a layout method. Information Cube (Rekimoto, J. 1993) is a visualization technique constructed with a three-dimensional space as a nested construction and features a translucent display of the layered structure. That is, the fundamental purpose of this method is hierarchical clustering for the tree.

On the other hand, our Ripple Presentation is effective when the targeted tree structure data is expressed in a time-series hierarchy. The time-series hierarchy is expressed by concentric circles. Thus, there exists other research to devise the arrangement of the nodes paying attention to the time of the nodes. John V. Carlis focuses on the cycle of time and studied an expression technique for arranging time series information on a spiral as the time-series hierarchy (Carlis, J. V., Konstan, J. A. 1998).

Various algorithms that use concentric circle placement as the drawing convention (like Ripple Presentation does) have been proposed (Wills, G. J. 1999). Eades presented a linear tree-drawing algorithm using a concentric circle layout and proved that the drawing rule of "no crossing" will always be achieved (Eades, P. 1991). Manning and Atallah proposed an algorithm for concentric circle layout that clearly shows "symmetry" (Manning, J. and Atallah, M. J. 1989). We focus on semantic rules, by which users are made aware of the novelty of information by the length of edges and made aware of the categories of information by the angles of edges, rather than by structural rules.

Ka-Ping Yee also expresses each hierarchy of the tree structure in the radial layout. The feature is that the layout is changed by animation dynamically according to user's attention nodes (Yee, K. -P. et al. 2001). Our approach is, dynamically speaking, a circle arrangement based on historical information of nodes rather than rearrangement for clear layout of the tree structure.

prefuse (Heer, J., Card, S. K., and Landay, J. A. 2005) is a toolkit for devising the layout through various effects of animation. The target data is a general graph structure instead of a tree structure. prefuse gives importance to the dynamic change of the layout of the graph based on the

user's viewpoint. We also emphasize the user's viewpoint. In Ripple Presentation, the graph is dynamically changed by animation, and the latest information always appears in the neighborhood of the parent nodes.

7 Future Work

There is room for improvement of implementation, such as enhancement of the interface and the data-collection function.

For example, when a category is described for each article like in the RSS of a news site, we want to present to users the range of the category in the view explicitly. By implementing the reading RSS feed of two or more news sites and merging articles, we want users to be able to compare articles about the same news from various viewpoints. It is also important that the drawing of the ripples does not obstruct the user's view. Concretely, it is necessary to devise a system that draws ripples only in the subtree part corresponding to the mouse's position from the user's point of view.

It is necessary to prevent node overlapping, since child nodes closer to a parent node are placed in a more crowded way. An effective solution might be to use the fish-eye lens approach (Lamping, J. and Rao, R. 1993, K. Misue et al. 1995) to zoom the neighborhood of the parent node into focus. Moreover, the crossing of the edges should be reduced to improve readability in Ripple Presentation. It is necessary to model the calculation for angles of edges in designing the layout.

8 Summary

To support efficient information search by presenting tree structure data with historical information visually, we proposed the Ripple Presentation as a visualization technique. This method puts the focus on the expression of the novelty of information, which is difficult to visualize using a tree-structure expression like a directory.

The advantage of this expression method is that it becomes easy to search for the latest information, because the latest information is arranged in the neighborhood of parent nodes by making a correspondence between the length of the edges and the elapsed time. Moreover, the view in the categorization graph is obtained by making a correspondence between the angle of an edge and a category or similarity. As a result, the users focus attention only to the node of the category of interest.

As an application example, articles on a News site and trackback on a Weblog were expressed using Ripple Presentation. In both cases, the value lies in the novelty of the article. In addition, we can obtain not only the latest information, but also, from an overview of the article group, we can analyze trends in the article group, the appearance cycle, frequency, and so on.

The Ripple Presentation is applicable only for tree structure data with historical information. However, it

seems that the range of applications of the Ripple Presentation is wide. In the real world, quite a lot of data with historical information exists. In this paper, we used RSS as target data, which is available as well-structured data resource in spite of a large amount of information. We hope that Ripple Presentation will be used in the various areas of information visualization in the future.

References

- Carlis, J. V., Konstan, J. A. (1998): Interactive Visualization of Serial Periodic Data. *Proc. the 11th annual Symposium on User Interface Software and Technology*: 29-38.
- Eades, P. (1991): Drawing Free trees. Res. Rep. IIAS-RR-91-17E, Intern. Inst. For Advanced Study of Social Information Science, Fujitsu Lab. Ltd.: 1-29.
- Heer, J., Card, S. K., and Landay, J. A. (2005): prefuse: a toolkit for interactive information visualization. *Proc. of Human Factors in Computing Systems 2005*: 421-430.
- Johnson, B. and Shneiderman, B. (1991): Treemaps: A Space-Filling Approach to the Visualization of Hierarchical Information Space. *Proc. of the 2nd International IEEE Visualization Conference*: 275-282.
- Lamping, J. and Rao, R. (1996): The Hyperbolic Browser: A Focus+context Technique for Visualizing Large Hierarchies. *Journal of Visual Languages and Computing* 7(1): 33-55.
- Manning, J. and Atallah, M. J. (1989): Fast detection and display of symmetry in trees. *Congressus Numerantium*.
- Misue, K., Eades, P., Lai, W. and Sugiyama, K. (1995): Layout Adjustment and the Mental Map. *Journal of Visual Languages and Computing* 6(2): 183-210.
- Rekimoto, J. (1993): The Information Cube: Using Transparency in 3D Information Visualization. *Proc. of the Third Annual Workshop on Information Technologies & Systems*: 125-132.
- Sugiyama, K. (2002): *Graph Drawing and Applications for Software and Knowledge Engineers*. USA, World Scientific Publishing Co. Pte. Ltd.
- Termmi: Text Mining Shareware Software, <http://gensen.dl.itc.u-tokyo.ac.jp/termmi.html>, Accessed 27 September 2005.
- Wills, G. J. (1999): Niche Works - Interactive Visualization of Very Large Graphs. *Journal of Computational and Graphical Statistics* 8(2):190-212
- Yee, K. -P., Fisher, D., Dhamija, R. and Hearst, M. (2001): Animated Exploration of Dynamic Graphs with Radial Layout. *Proc. of the IEEE Symposium on Information Visualization 2001*:43.

SnapShoot: Integrating Semantic Analysis and Visualization Techniques for Web-based Note Taking System

Soichiro Iga

Makoto Shinnishi

Advanced Technology R&D Center
Ricoh Co., Ltd.
16-1 Shinei-cho, Tsuzuki-ku Yokohama, Japan
Email: igaiga@rdc.ricoh.co.jp

Abstract

We have many application software today to support reading and writing respectively. However, we believe that few applications available today fully support both reading and writing process of the note taking task. This paper presents a novel prototype system called "SnapShoot". The SnapShoot system integrates existing technology - a WWW browser, an out-line processor, context-awareness - with a unique information visualization tool in support of the users who edit and manage electronic documents online. The current system composed of a web based portal to manage the users' documents and an out-line processor tool responsible for editing and visualizing the users' documents. The out-line processor tool visualizes each line of the document by the semantics of the sentence. Early user experience shows that the system provides many benefits over the current application softwares.

Keywords: note taking, information visualization, interaction technique, sowing interface, context-sensitive, world wide web

1 Introduction

Composing electronic documents is a commonly practiced activity which forms a combination of a wide range of different activities, and which serves many purposes. For example, a document may be read by skimming it rapidly or scanning for a specific piece of sentence, or it may be written by summarizing the process of the discussion or arranging work for someone.

One of the casual instance of composing electronic documents is the note taking. There are several difference between note taking and rather formal document production. In case of notes like conference minutes or personal notes, compared to the formal documents, they may not be strictly structured from scratch, and they may require immediacy in finding the right location to read and write during the session.

Another important factor for note taking is the document management. We may refer to the several documents during a meeting, however, it is problematic for us to manage these dispersed documents in the whole process of the meeting.

There are several researches and applications that partly contribute to some of the activities in the note taking. The Notes Program(Neuwirth, Kaufer,

Chimera & Gillespie 1987) is a hypertext application which supports writing process by recording annotation texts to the source text. The system provides several views for representing annotation texts. The Sparrow system(Chang 1998) allows contributors to add information to a web page over the network which would facilitate collaborative work. The Reader's Helper(Graham 1999) helps the users to read web pages by showing dynamic thumbnail representation of the document, and highlighting the words which are relevant to the users' topic of interests. XLibris(Schilit, Golovchinsky & Price 1998) supports the user's reading by providing a way to add free form annotation on the digital document.

Many visualization technique have been proposed for representing large scale of data. MagicLens(Stone, Fishkin & Bier 1994) introduces an interaction technique which shows the details of documents when the user overlays the object called "lens" to the thumbnail information. For instance, when you hold lens over to the large-scale map, the detailed map would appear in the area of the lens. Another system visualizes the process of calculation of the spreadsheet application(Igarashi, Mackinlay, Chang & Zellweger 1998). It may reduce the cognitive load of the user by externalizing the internal calculating process. The Alphaslides(Ahlberg & Shneiderman 1994) proposes a scroll bar widget to rapidly scan through and select from lists of alphanumeric data. The LensBar(Masui 1998) provides same kind of technique integrating keyword filtering and zooming technique. The Information Mural(Jerding & Stasko 1998) supports visualizing and navigating large information spaces. The technique can be applied to create a reduced representation of a text file even when the number of lines is greater than the available display space.

Ajax is a web development technique for creating interactive web applications using a combination of rather existing web-based technologies such as HTML, Javascript, and XML. Applications such as a word processor(Upstartle 2005) and a spreadsheet(Trimpath 2005) work in a general web browser window. Most of the information manipulation can be done on the web browser, and the documents are managed in the central server. In this way, the application softwares are location-independent and the users can access to their documents from anywhere.

Despite we have many application software today to support reading and writing respectively, however, we believe that few applications available today fully and seamlessly support both reading and writing process of the note taking task.

We have created a new web-based note taking environment called the SnapShoot which is to facilitate reading and writing of electrical documents with a unique information visualization techniques. The sys-

tem also have a web-based document portal which could manage a number of electronic documents referred to while taking the note.

In the following sections, the SnapShoot system is described in terms of the user interface with some unique visualization techniques. Future issues and potential research directions are also discussed.

2 SnapShoot

The SnapShoot system integrates existing technology - a WWW browser, an out-line processor, context-awareness - with a unique information visualization tool in support of the users who edit and manage electronic documents online. The current system composed of a web based portal page to manage the users' documents and an out-line processor tool responsible for editing and visualizing the users' documents. The out-line processor tool visualizes each line of the document by the semantics of the sentence. The way the tool understands the user's topic is through a *editing profile* which contains the list of the topics and the relevant keywords for each topics. Several visualization techniques are incorporated into the out-line processor tool to support reading and writing of the electronic document. The out-line processor has a text-based markup sticky notes that the user can promptly visualize the mark for particular line just by typing specific markup keywords. The system also can convert most of the commercial available file formats into HTML format, and the user can label visual sticky notes onto the particular position on the HTML formatted document. The user can get a united document in HTML format which assembles the note edited on the out-line processor and the sticky notes that applied to the relevant documents on a portal page.

In the following sections the out-line processing environment is described in terms of the user interface and followed by several features to facilitate the user's editing task.

2.1 The User Interface

Reading electronic documents is more difficult task than reading paper documents(O'Hara & Sellen 1997). One of the most important aspects of on-line electronic tools is the user interface that it should support annotation, navigation and flexibility and control in spatial layout.

Reading and writing electronic documents might be more complicated than just reading. The document may dynamically change its semantics as the user edits, and the user's focus of the location in the document would move backward and forward. The user's writing operation of the electronic tools should be done in real time, and it also has to be done in parallel with the support for reading which is not to burden the user.

Figure 1 shows the SnapShoot editor displaying a conference note. The prototype system runs on a web browser, and it consists of a portal page and a editing page. Basic operation for using this system is to create or select "note", which is the document structure provided in the system, from the portal page and edit it on the tools provided in the editing page. The user can upload electrical documents or capture screen shot in the editing page, and the system would associate the user's uploaded documents to the note.

The user interface of the portal page mainly provides a user login facility and a note document management facility. Note files and associated documents are managed with respect to each user.

The user interface of the editing page can be classified broadly into two parts - menu(a) and editor(b).

On the left side of the display is the menu part(a). The menu is to show commands(d) and a list of files(c) which the user can manipulate. Commands include function for editing keyword list to be hereinafter described, capturing screen shot, sending e-mail, and so on. The list of files are the files that the user uploaded to the system, and the user can execute them and also the user can put annotation by sticky notes freely onto the contents.

On the right side of the display is the editor(b). The editor part is composed of the out-line processor(e), *Topic Bar*(f), and a list of keywords(g). The out-line processor is to edit electronic documents stored in the document portal server, and it offers several visualization techniques to facilitate note taking activities. The *Topic Bar* is a dynamic brief representation of the document which the user is editing. The list of keywords is to assign unique character string(A1, A2, ...) for an arbitrary sentence, and the user can call it up by typing the character string on the out-line processor. In current prototype, the user can create up to three categories of list of keywords, for example, list of agendas, participants, electronic documents, and so on.

We will describe each components in the following sections.

2.1.1 Out-line Processor

On the SnapShoot system, the user can edit the electronic documents stored in the document portal server on the out-line processor online. The out-line processor obviously incorporates function to read and write texts, to indent texts, and to move texts backward and forward by line. The out-line processor should support both reading and writing task that the interaction techniques should be seamless and continuous between reading and writing for operating the out-line processor. The out-line processor incorporates following three zooming techniques(Bederson & Hollan 1994, Furnas 1986) that the user can choose one of them depending on the situation.

1. Linear zoom: Linear zoom is a default editing mode that scales of all of the text lines on the out-line processor would change in a single uniform way as the user controls the scale scroll bar.
2. Distortion zoom: This is a distortion-oriented zooming techniques where the user can smoothly zoom into the focusing area of the text lines, and the colors and sizes of text lines would change according to the distance from point of focus.
3. Context-sensitive zoom: The colors and sizes of the text lines would change in proportion to the the topic of each text lines. Topic categories are automatically recognized by means of language processing.

Especially like taking note during the meeting, the topic would change from one moment to the next. Hence, the user's focus of the location in the document would move backward and forward. The position of the line where the user is currently editing is highlighted in a dark color. When the user moves the cursor, the highlight of the color would be changed by gradation, and the gradation would decay in proportion to how fast the user moved the cursor position(Figure 3). Hence, the user can easily find the previous and the current cursor position.

Information handled on our system can be inter-linked by markup-based annotation. The list of keywords stores a set of an unique character string and

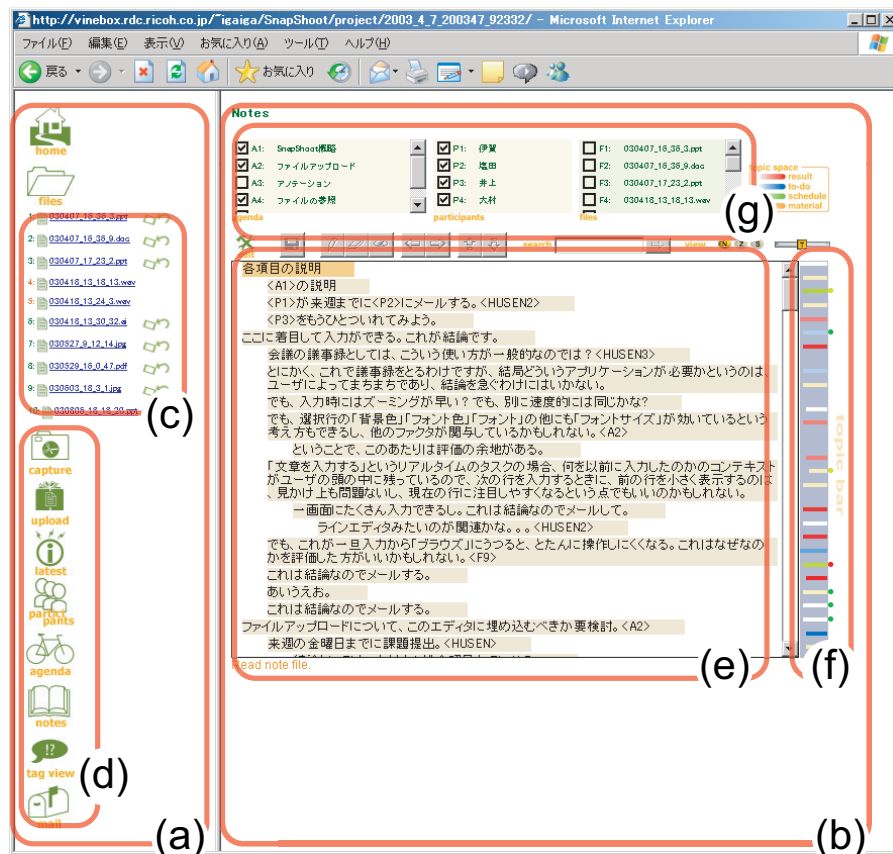


Figure 1: SnapShoot: the user interface of the editing page

an arbitrary sentence that the user can use the unique character for the alias for typing the sentence. For example, assume that the character string “<A1>” and the sentence “Review calendar” are stored in the list of keywords. The sentence “Review calendar” would be linked, when the user types the character string “<A1>” in the out-line processor. This function would be helpful for taking note during the meeting that the user can easily markup the current topic of the meeting by registering agenda of the meeting.

The user can create a bookmark for each line of the document also by markup-based annotation technique. When the user types the character like “<STICKY>”, colored circle sign would appear at the corresponding position on the *Topic Bar*. The color of each appearing sign would automatically defined according to the numbering in the typed command, i.e. <STICKY1>, <STICKY2>, <STICKY3>, and so on. The user can easily find marked positions in the document and also the user can categorize some information by this method. For example, the user can easily jump to the point of important affairs or pending discussion or the user can visually classify the items just by putting markup-based annotation.

2.1.2 Topic Bar

Many interaction techniques have been proposed to improve a performance of a scroll bar (Ahlberg et al. 1994, Jerding et al. 1998, Masui, Kashiwagi & Borden IV 1995, Masui 1998). We also need to incorporate the user interface which fits to manipulate note documents. Design issues are that notes are not strictly structured that the user interface should adapt to the change of the content of the document, and it has to be simple enough to scan and search of the document that the system would be utilized during the session

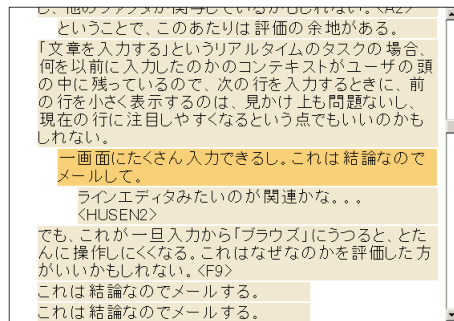
such as a conference and a meeting.

Figure 4 shows the user interface of the *Topic Bar*. The *Topic Bar* is a unique information visualization tool which composed of a functionality of visualizing the contents in each line of the note document and a functionality of a scroll bar. Color of each line changes according to the semantic meaning of each sentence in the note. The system tracks the change of the meaning of the sentence in real time. In current system, the user can create up to four categories for semantic meanings, for example, result, to-do list, schedule, files, and so on. Unique color space would be assigned for each category, and contrasting density of each color space would change according to how relevant to the particular semantic category. Semantic meaning measure is computed by keyword phrases and weights of their frequency defined in the *editing profile* and the proximity of keywords in the sentence.

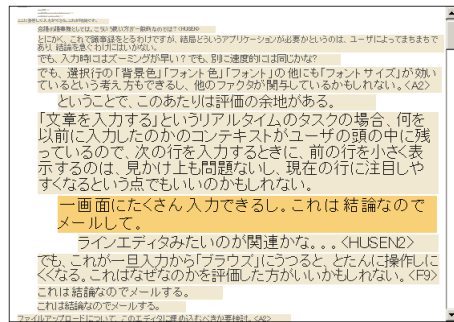
The *Topic Bar* provides comprehensive view of whole the note document. The user can quickly skim the position of the topic inside the note and navigate through the contents where significant information should be comprised in. Contrasting density of the color represent the relevancy to the defined topics that the user can understand the distribution of significant location in the note in a relative manner. The signs for markup-based annotations are shown alongside of the *Topic Bar* so that they also be utilized as a key to find the particular sentence in the note document.

2.1.3 Sowing Interface

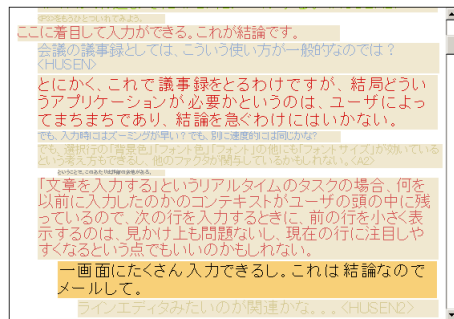
Electronic documents enables the users to handle large scale of information compared to the paper documents, however, it is problematic for the users to search for the information he or she is seeking. For



(a) Linear zoom



(b) Distortion zoom



(c) Context-sensitive zoom

Figure 2: Variation of visualization in out-line processor: the users have a choice of visualization styles depending on the situation of their task.

instance, when you are taking notes of the conference and you are to extract some related agenda at the end of the conference, it involve an immense amount of time and effort for you to find relevant topics you are finding. Some researches proposes the control of large scaled information in proportion to the speed of the user's operation(Iga, Kadota & Yasumura 1997, Igarashi & Hinckley 2000). Applying these techniques would improve the efficiency of browsing a large document. We propose a visualization technique called the *Sowing Interface*. *Sowing Interface* integrates a speed-dependent browsing and a simple text summarization technique.

Figure 5 shows the general idea of how *Sowing Interface* works. When the user drag over the thumbnail image in the *Topic Bar* screen with the pointing device, the corresponding texts would be transudatory visible according to the speed of the drag movement. The faster the user drags, for instance, the more distant text from the start point of the drag would appear and the appearance of the texts would be sparse.

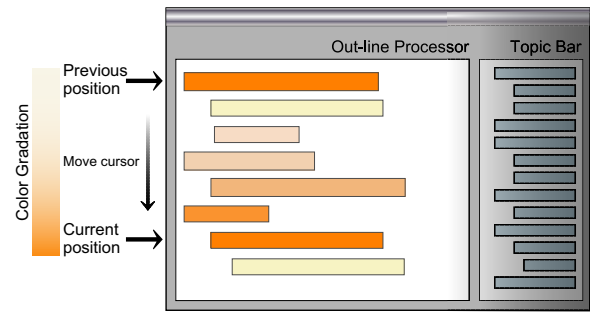


Figure 3: Cursor trajectory gradation

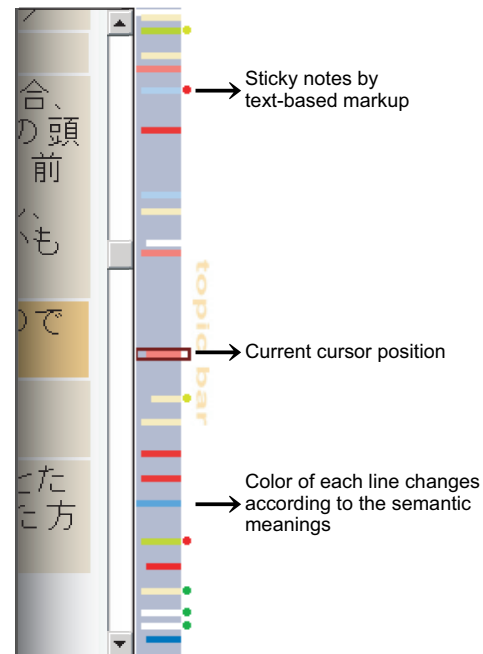


Figure 4: Topic Bar

At the same time, the text would be summarized in proportion to the speed of the drag movement and the distance from the start point of the drag movement. The distant text would be strongly summarized while the near text would stick closely to the original text.

As the note document gets larger, it is difficult for the user to remember exact position where the things are written on. Using this method, the user can efficiently search for the rough position by dragging fast. The user can find exact position of the destination by interactively and gradually slow down his or her dragging operation.

2.2 Sticky Note

The user can easily incorporate the electronic documents to the system just by the drag and drop operation on the web browser. The system also supports to capture screen shot of the user's computer screen. The documents which the user uploaded to the server are automatically converted into HTML formatted document, and the user can put sticky notes onto the converted document on a web browser(Figure 6). Converted HTML file includes a user interface button to create sticky notes. Clicking the button would invoke the small window (sticky note) overlaid on the HTML file. The user can freely move the sticky note window and he or she can type texts from the keyboard into the sticky notes on the web browser. The

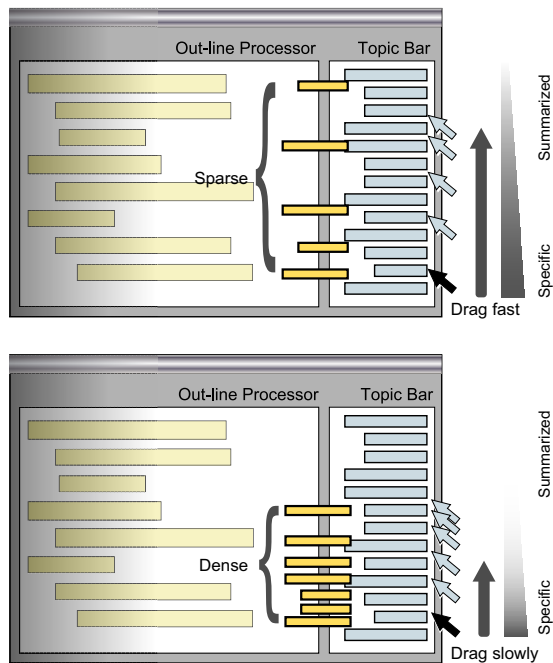


Figure 5: Sowing Interface

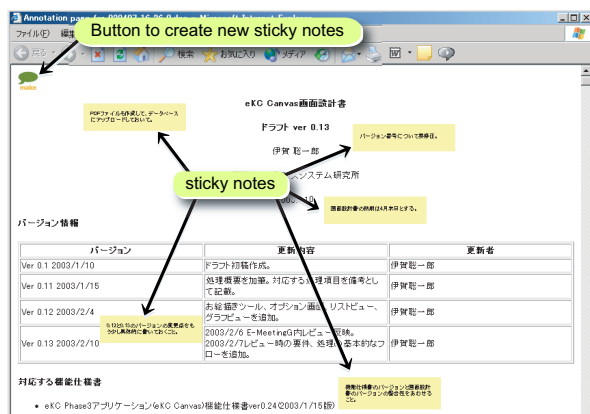


Figure 6: Sticky notes attached to the uploaded document

system currently supports image files(jpg, bmp, gif), plain texts, Microsoft Office documents, and Adobe PDF files.

The system assembles the annotated sticky notes for each uploaded documents, and the system provides the view to show the list of collected information. Figure 7 shows how the sticky notes are utilized. The user can have a clue to get rough positions where each sticky notes are placed on the HTML by the indicated sign on a matrix-shaped thumbnail. In current prototype, we provide 3×3 sized matrix to show the position of the sticky note. Width and height of the document are estimated from the HTML file, and the rough position can be calculated where the actual coordinates of the sticky note are contained within the matrix.

3 Implementation

Figure 8 shows the system architecture of the SnapShoot. The system is a web-based server/client networked system. The client system runs on commercial web browser (Microsoft Internet Explorer 6.0) and it communicates with a web server via HTTP

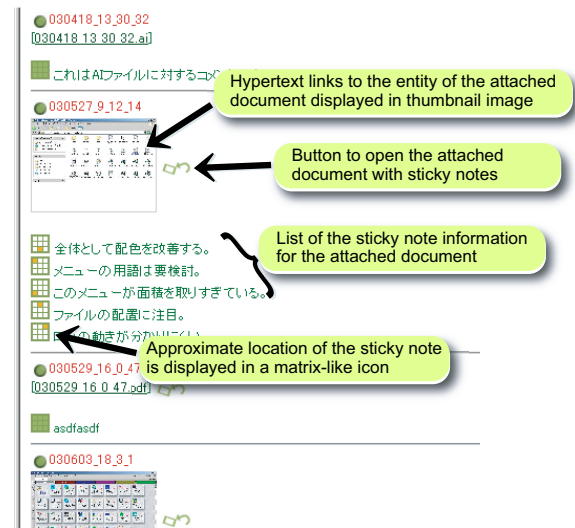


Figure 7: Sticky notes assembled in an united document

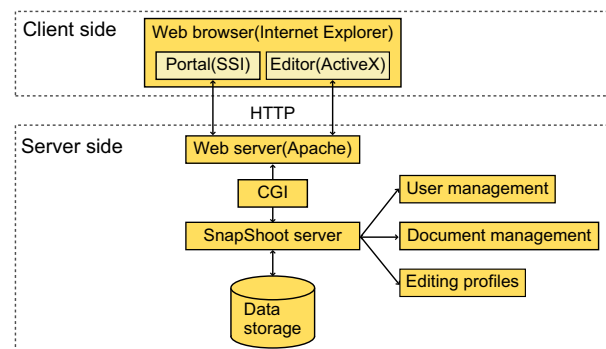


Figure 8: SnapShoot system architecture

using Common Gateway Interface (CGI). The server consists of a web server and a database management system. The documents edited on the clients system are sent and received through CGI modules and will be managed in the database. The server works on Linux operating system and the CGI modules are implemented in Perl language. Out-line processor part of the system was implemented in Microsoft Visual Basic as an ActiveX user control and Javascript on a PC running Microsoft Windows XP.

The *editing profile* contains the multiple sets of a semantic category and relevant keyword phrases and their weights of frequencies. In current prototype system, several semantic categories predefined that should be manually created. We used a hundred of conference minutes and memos for source documents, and we manually labeled categories - result, to-do, schedule, files - for each sentences contained in each source documents. Then, we extracted noun phrases and counted frequencies of each phrases from each sentences using Chasen system (Matsumoto, Kitauchi, Yamashita, Hirano, Matsuda, Takaoka & Asahara 2000). These extracted data are stored as *editing profile* data at the server system. Semantic meaning measure is computed by degree of similarity which is a pattern matching of noun keyword phrases and a cumulative number of the frequencies of each matched keyword phrases.

Text summarization in the *Sowing Interface* also uses same kind of technique. The rate of how briefly summarize each sentence is calculated through the weights of frequencies.

4 User Experience

Although we have not precisely tested the system yet, several users have been tested current prototype system. The purpose of the test was to accumulate informal user experiences mainly on editor part of the system. Subjects were three laboratory members. Every subjects were the experts in using computer systems. The given task was to use the system for taking notes during their casual meetings and we observed how they operate the system. We also interviewed after the week of the test period for each subject.

The result for the user experience of out-line processor was that every subjects have been observed to use distortion zoom mode for editing their notes. One of the comments from the interview says that distortion zoom makes the most sense for getting a quick overview of whole the note while it is feasible enough for editing at the same instant. Context-sensitive zoom has not been utilized so much. Most of the subjects have been using *Topic Bar* for finding topics that same function has been provided by *Topic Bar* with greater availability. Current visualization technique for context-sensitive zoom was not scalable enough for representing large document that it might be a future work.

Current *Topic Bar* can classify the sentence into only a single semantic category that unintended classification has been observed sometime. One of the subject has been observed to type the keyword like "result" on the out-line processor after the several trials which would lead to recognize the sentence to the particular semantic category. For the markup-based sticky note function, it was helpful for the most subjects to manually label it for their remembrance.

All the subjects mentioned on the emotional aspect of visualization that it was really fun to see the dynamic movement of texts during their uses. In interview, one of the subject says that by casually browsing the markups without much thought, it was helpful for him to remind the structure of what he wrote in the note. D.A. Norman gave a framework of three different aspects of design - visceral, behavioral, and reflective (Norman 2004). More emotional aspect of design should be taken into consideration for the visual representation and operation of the user interface.

5 Future Work

Interoperability of the note documents would be the important issue for the system to be put to practical use. Current prototype only supports to distribute note documents in HTML format, however, the early user experiment shows that most of the users mentioned for the needs of the varieties of file formats. In future work we will enhance the prototype system including supporting standard file format like OPML (Outline Processor Markup Language) (OPML 2000). OPML is an XML-based format that allows exchange of outline-structured information between applications running on different operating systems and environments.

The system is a network-based system that it needs networked environment to use. At least the editor part of the system has to be operated without network. The system also has a restriction that it only could handle Japanese language due to the capability of language processing module which we have been incorporated. It is possible to extend it to multilingual combining several language processing softwares.

In the current prototype, the *editing profile* is manually created and predefined that the user does not have a way of customizing it. MSBNx (Kadie, Hovel

& Horvitz 2001) is a component-based application for creating, assessing, and evaluating Bayesian Networks. This kind of editor and toolkit-based support should be provided for the users to create and customize *editing profile*.

Some basic observations examining note taking on physical papers have been conducted (Cunningham & Knowles 2005). Motivations for taking individual notes observed from participants have been characterized in several ways - references, reminders, ideas, paper conversations, negative notes, boredom notes, and doodles. Future work also include this kind of long term user observation on our system.

6 Conclusion

In this paper, we have proposed a new web-based note taking environment called the SnapShoot which is to facilitate reading and writing of electrical documents with a unique information visualization techniques. Based on the preliminary user experience using the SnapShoot, we think that integration of visualization techniques on the out-line processor, used in conjunction with the *Topic Bar*, provides many benefits over the current application softwares.

In the future we will investigate how this approach can facilitate the note taking activities in the practical situations.

References

- Ahlberg, C., & Shneiderman, B. (1994), The Alphaslider: A Compact and Rapid Selector, *in Proc. of CHI'94*, pp. 365-371.
- Bederson, B.B., & Hollan, J.D. (1994), Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics, *in Proc. of UIST'94*, pp. 17-26.
- Chang, B. (1998), In-place editing of web pages: Sparrow community-shared documents, *in Proc. of Seventh World Wide Web Conference*, pp. 489-498.
- Cunningham, S.J. & Knowles, C. (2005), Take note: academic note-taking and annotation behavior, *in Proc. of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries*, pp. 374-374.
- Furnas, G.W. (1986), Generalized Fisheye Views, *in Proc. of CHI'86*, pp. 16-23.
- Graham, J. (1999), The Reader's Heler: A Personalized Document Reading Environment, *in Proc. of CHI'99*, pp. 481-488.
- Iga, S., Kadota, M. & Yasumura, M. (1997), Sensory Substitution Technology for The Blind based on Human Visual Space Searching Model, *in IPSJ-SIG-HI Technical Reports Vol.1997 No.088* (in Japanese).
- Igarashi, T., Mackinlay, J.D., Chang, B.W. & Zellweger, P.T. (1998), Fluid Visualization for Spreadsheet Structures, *in Proc. of 14th IEEE Symposium on Visual Languages*, pp. 118-125.
- Igarashi, T. & Hinckley, K. (2000), Speed-dependent automatic zooming for browsing large documents, *in Proc. of UIST'00*, pp. 139-148.
- Jerding D. & Stasko, J. (1998), The Information Mural: A Technique for Displaying and Navigating Large Information Spaces, *IEEE Transactions on Visualization and Computer Graphics*, Vol. 4, No. 3, pp. 257-271.

- Kadie, C.M., Hovel, D. & Horvitz, E. (2001), MSBNx: A Component-Centric Toolkit for Modeling and Inference with Bayesian Networks, Microsoft Research Technical Report MSR-TR-2001-67, [on-line] <http://research.microsoft.com/adapt/MSBNx/MSBNxTechreport.pdf>
- Masui, T., Kashiwagi, K. & Borden IV, G.R. (1995), Elastic graphical interfaces to precise data manipulation, *in* Proc. of CHI'95 Conference Companion, pp. 143–144.
- Masui, T. (1998), LensBar Visualization for Browsing and Filtering Large Lists of Data, *in* Proc. of IEEE Symposium on Information Visualization, pp. 113–120.
- Matsumoto, Y., Kitauchi, A., Yamashita, T., Hirano, Y., Matsuda, H., Takaoka, K. & Asahara, M. (2000), Japanese Morphological Analysis System Chasen version 2.2.1, [on-line] <http://chasen.aist-nara.ac.jp/chasen/doc/chasen-2.2.1.pdf>
- Neuwirth, C., Kaufer, D., Chimera, R. & Gillespie, T. (1987), The Notes program: a hypertext application for writing from source texts. *in* Proc. of HYPERTEXT '87, pp. 121–141.
- Norman, D.A. (2004), Emotional Design: Why We Love (Or Hate) Everyday Things, Basic Books.
- O'Hara, K. & Sellen, A.J. (1997), A Comparison of Reading Paper and On-Line Documents, *in* Online Proc. of CHI'97, [on-line] <http://www.acm.org/sigchi/chi97/proceedings/paper/koh.htm>
- OPML - Scripting News, Inc. (2000), OPML 1.0 Specification, [on-line] <http://www.opml.org/spec>
- Schilit, B., Golovchinsky, G., Price, M. (1998), Beyond Paper: Supporting Active Reading with Free Form Digital Ink Annotations, *in* Proc. of CHI'98, pp. 249–256.
- Stone, M.C., Fishkin, K. & Bier, E.A. (1994), The Movable Filter as a User Interface Tool, *in* Proc. of CHI'94, pp. 306–312.
- Trimpeth Project, Num Sum (2005), [on-line] <http://numsum.com/>
- Upstartle LLC., writely (2005), [on-line] <http://www.writely.com/>

Drawing Bipartite Graphs as Anchored Maps

Kazuo Misue

Graduate School of Systems and Information Engineering
University of Tsukuba
1-1-1 Tennoudai, Tsukuba, 305-8573 Japan
misue@cs.tsukuba.ac.jp

Abstract

A method of drawing anchored maps for bipartite graphs is presented. Suppose that the node set of a bipartite graph is divided into set A and set B. On an anchored map of the bipartite graph, the nodes in A, which are called “anchor nodes,” are arranged on the circumference, and the nodes in B, which are called “free nodes,” are arranged at suitable positions in relation to the adjacent anchor nodes. This article describes aesthetic criteria that are employed according to the purpose of drawing anchored maps and a method to arrange the anchor nodes so that they satisfy the criteria. The effectiveness of the proposed method is discussed in terms of the aesthetics of drawing results.

Keywords: graph drawing, anchored map, bipartite graph

1 Introduction

Bipartite graphs can be found in various fields. Examples include graphs that show relationships between documents and the words included in them, graphs that show relationships between customers and goods they bought, and graphs that show relationships between communities and their members.

Such relationships are often used to compute similarities. For example, when customer A and customer B have bought a lot of the same goods, we may think that A and B share similar properties. On the other hand, when a lot of customers bought goods X together with goods Y, we may think X and Y have some similarities. Such similarities are used for information retrieval, E-commerce recommendations, and so on.

The purpose of this research is visualization of the relations represented by bipartite graphs. In particular, its target is to facilitate an understanding of the node clusters’ meaning and the relationships among the clusters. In the examples mentioned above, we expect that bipartite graphs can help us to understand the relational structures such as the scale of clusters of similar goods, the relationships among clusters of goods, and the relationships among clusters of goods and customers.

We developed a drawing technique for expressing bipartite graphs called “anchored maps.” It has two kinds of node: “anchors” and “free nodes.” The anchors are arranged on the circumference at equal intervals, and the free nodes are arranged at suitable positions in relation to the adjacent anchor nodes. The problem is how to decide the order of anchor nodes on the circumference and the positions of free nodes. We do not need to worry about the routing of edges because we use straight-line drawing.

In this article, we describe the technique for drawing anchored maps. We explain the aesthetic criteria that are employed according to the map’s purpose and a method to arrange anchor nodes to fulfill the criteria. We also show the technique’s effectiveness in an aesthetic evaluation of its drawing results.

2 Anchored maps of bipartite graphs

2.1 Bipartite graphs

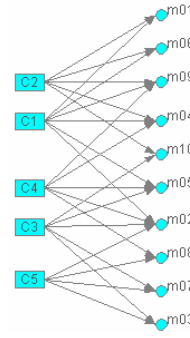
Suppose that $G = (A \cup B, E)$ is a bipartite graph. A and B are finite sets of vertices, and A and B are disjoint. E is a finite set of edges, and E is a subset of $A \times B$.

2.2 Expression styles of bipartite graphs

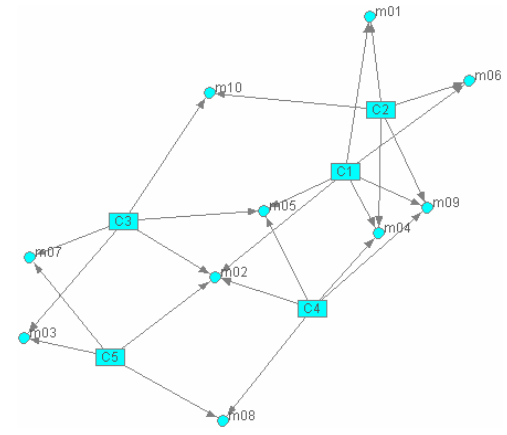
Figure 1 shows various styles of expressing bipartite graphs. Virtual data is used as an example. Suppose that there is a bipartite graph that represents relationships between five communities and ten members. Figure 1(a) shows the matrix representation. Each row corresponds to a community, and each column corresponds to a member. A “1” in a cell indicates that a member belongs to a community; for example, member m01 belongs to communities C1 and C2. This matrix cannot intuitively show relational structures because it is not visual. Figure 1(b) shows a two-layer graphical representation of the same bipartite graph. The five nodes on the left-hand side represent communities, and the ten nodes on the right-hand side represent its members. A line between a community and a member shows that the member belongs to the community. Figure 1(c) shows the same graph as laid out by using spring embedding (Eades 1984). While Figure 1(b) gives the impression that the arrangement is more orderly than (c)’s, it doesn’t emphasize the relations between communities and members. On the other hand, while (c) illustrates these relationships, its distinction between communities and members is not clear because the two kinds of node exist together.

| | C1 | C2 | C3 | C4 | C5 |
|-----|----|----|----|----|----|
| m01 | 1 | 1 | | | |
| m02 | 1 | | | 1 | 1 |
| m03 | | | 1 | | 1 |
| m04 | 1 | 1 | | 1 | |
| m05 | 1 | | 1 | 1 | |
| m06 | 1 | 1 | | | |
| m07 | | | 1 | | 1 |
| m08 | | | | 1 | 1 |
| m09 | 1 | 1 | | 1 | |
| m10 | | 1 | 1 | | |

(a) Matrix style



(b) Two-layer style



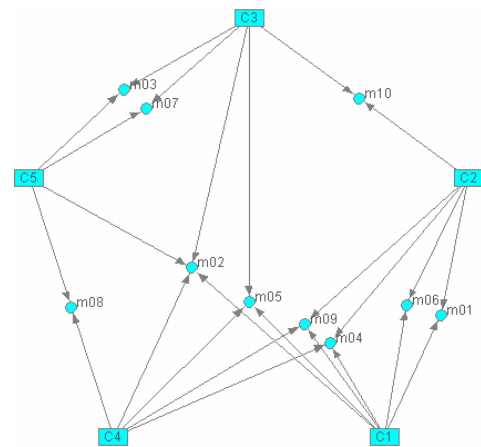
(c) Spring-embedded style

Figure 1: Various styles of representing bipartite graphs

2.3 Anchored Maps

An anchored map restricts some nodes to certain positions, but leaves other nodes so they may be arranged freely. The restricted nodes are fixed like anchors, hence the terms “anchors” and “free nodes.”

There are many variations in the restriction of the anchors. In the simplest one, each anchor is fixed to a coordinate. It is also possible to restrict anchors to a certain curve, in a certain area, and so on. Figure 2 shows the same bipartite graph shown in Figure 1 as an anchored map. The communities are arranged on the vertices of a pentagon, and members are arranged at a suitable position to represent their relationship to the communities. The distinction of the nodes is clearer than in Figure 1(c), and it is easy to see the relations between members.

**Figure 2: Anchored-map style**

2.4 Drawing Convention

The drawing convention of anchored maps is as follows:

(Arrangement convention for the nodes)

- Compound coordinate system: For a bipartite graph $G = (A \cup B, E)$, the elements of set A (the anchors) depend on a circular coordinate system; they are arranged on the circumference at equal intervals (in other words, they are arranged on the vertices of a regular polygon if $|A| \geq 3$), while the elements of set B (the free nodes) are independent of the coordinate system.

(Routing convention for the edges)

- Straight line wiring: The adjacent nodes are connected by a straight line.
- Edges are independent of the coordinate system.

2.5 Features of anchored maps

The features of the anchored maps depend on how one restricts the anchor positions and how one arranges the free nodes.

For example, when anchor nodes are separated by enough distance and the free nodes have spring embedding, the anchor nodes connected to a common free node pull it in different directions. By the nature of the spring embedding, the free node will “move” to an appropriate position that expresses its relation to the connected anchor nodes.

Misue and Watanabe showed an example of expressing the relations between concepts extracted from a large volume of textual data in maps with three anchors (Misue & Watanabe 1999). Compared with simple spring embedding, the anchored maps were able to clarify the concepts that were worth attention, and thereby facilitate an easier analysis of large textual data.

2.6 Purpose in drawing anchored maps

The purpose of a bipartite graph visualization obviously varies with the application. However, the essential reasons for using anchored maps are as follows.

- (a1) The positions of free nodes are understandable in relation to the anchors.

Free nodes should be arranged close to anchors. Two or more anchors connected to the free node pull it towards them. This is advantageous for understanding which anchors are related to which free node.

(a2) Clusters of free nodes can be discerned based on their relation to the anchors.

The free nodes are divided into clusters by the connective relationships with the anchors (subsets of anchors which are connected to the free nodes). This illustrates what kinds of cluster exist and which anchor subset is responsible for constructing each cluster.

3 Drawing anchored maps

3.1 Aesthetic criteria

We employed the following aesthetic criteria for our anchored map drawing method. Note that some of these criteria are employed in many drawing methods. We also judged them to be important in consideration of the adaptability to the purpose of drawing of anchored maps.

(e1) Nodes connected to each other are laid out as closely as possible (minimize the total length of edges).

(e2) The number of crossings among edges is minimized.

(e3) Anchors connected to common free nodes are laid out as closely as possible.

(e4) Free nodes belonging to the same cluster are laid out as closely as possible (free nodes do not belong to the same cluster are not laid out closely).

3.2 Drawing procedure

The map is laid out in two steps:

(Step 1) Arrange anchors on the circumference at equal intervals.

(Step 2) Fix the anchors, and arrange the free nodes at the positions in which their relationships to the anchors are appropriately expressed.

In Step 1, the size (i.e., radius) of the circumference is decided, and the order of the anchors on the circumference is decided. The length of the circumference doesn't influence the quality of the layout; it influences only the size of the drawing. Here, the circumference is decided according to the size of the drawing area (i.e., window). On the other hand, the order of the anchors has a large influence on the quality of the layout. The next section describes how to decide the order of the anchors.

In Step 2, the positions (i.e., coordinates) of the free nodes are decided. Spring embedding with restrictions is used to find the positions. That is to say, in every iteration of spring embedding, the anchors are fixed and their positions are not changed, and only the free nodes are moved toward stable positions.

3.3 How to decide the order of anchor nodes

The distance between adjacent nodes (length of the edges) and the number of edge crossings are essentially

influenced by the order of the anchor nodes. However, these features are not deterministically decided from only the anchor node order. The spring embedding in Step 2 and the initial positions of the free nodes also influence the distance. However, spring embedding has negligible influence compared with the influence of the anchor order.

Thus, the order of the anchors is the most critical problem. We can evaluate the goodness of a certain order only after the spring embedding has been processed for a certain anchor arrangement, but to do this for all the possible candidate orders would require too much computing time. Thus, we need an alternative index that is computable in a deterministic way and with a low cost.

We propose to use the distance along the circumference between anchors connected to the same free nodes as such an index.

Although the computing cost of this index is not very low, we use it as the initial choice because it can be computed deterministically.

3.3.1 Preparation

Suppose that M is the number of anchors, that is, $M = |A|$.

The anchors are arranged on the vertices of a regular M -gon (a polygon with M vertices). The vertices of the M -gon are labeled clockwise from 1 to M . It doesn't matter which vertex is chosen to be 1.

Suppose that $p(a)$ is the position of anchor a .

3.3.2 Gap

The "clockwise distance" between the i -th vertex and the j -th vertex is the number of vertices we meet when we trace the vertices of the M -gon from the i -th to the j -th clockwise. The **clockwise distance** l_M is defined as

$$(1) \quad l_M(i, j) = (j - i + M) \bmod M$$

Next, we introduce a "gap" for every anchor connected to a certain free node. Suppose that f is a free node and $A' = \{a_1, a_2, \dots, a_k\}$ is a set of anchors connected to f . Moreover, suppose that a_1, \dots, a_k have been sorted by the numbers of vertices of the M -gon. In other words, $p(a_s) < p(a_t)$ if $s < t$. The gap for each anchor is the clockwise distance between one node and the next node (the next of a_k is a_1). The **gap** $g(a_s)$ of node a_s is defined as

$$(2) \quad g(a_s) = l_M(p(a_s), p(a_{(s+1) \bmod M}))$$

3.3.3 Penalty

When k anchors are connected with a certain free node, a sequence of k gaps is obtained. Removing the maximum gap from the sequence leaves $k-1$ smaller gaps. The "penalty" is the sum of powers of $k-1$ gaps. A small penalty indicates that the anchors connected to the same free node should be arranged close to each other. Therefore, the penalty could be used as an index of how to decide the order of the anchor nodes.

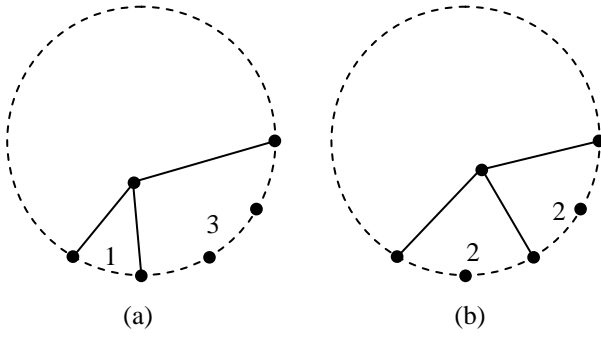


Figure 3: Effect of the parameter q in the penalty

Suppose that f is a free node and g_1, g_2, \dots, g_{k-1} is the remainder sequence that excludes the maximum gap from the gap sequence of the anchors connected to f . The **penalty** $p(f)$ of free node f is expressed as

$$(3) \quad p(f) = \sum_{i=1}^{k-1} g_i^q$$

Figure 3 illustrates the reason why we employed the power of the gaps with the parameter q . We introduced q to judge which of (a) or (b) is better when there is a free node in Figure 3. When $q=1$, the penalty $p=1^1+3^1=4$ in the case of (a), and $p=2^1+2^1=4$ in the case of (b). The penalty is the same value in either case; it is a useless index for deciding which arrangement is better. To take account of the balance and symmetry and to prefer the case of (b) we should have $q>1$ (for example, $q=2$). When $q=2$, $p=1^2+3^2=10$ in the case of (a) and $p=2^2+2^2=8$ in the case of (b); hence, (b) is chosen for the graph.

3.3.4 Searching for the optimal order

It is necessary to search for the anchor order that minimizes the penalty.

A simple technique is to compute the penalties for all the possible orders (isomorphism cause by rotation and mirroring is negligible), and then finds the order with the minimum penalty. This technique gives the optimal solution for the penalty but its computational cost is very large. Moreover, we should recall that the penalty is an alternative index. There is no guarantee that the optimal solution for the penalty is the optimal solution for the aesthetic criteria.

We created a quasi-optimal solution for the penalty and devised a technique with a lower computational cost. The technique begins with a random anchor order, and exchanges two anchor nodes while comparing the penalty before and after the exchange. It continues to choose pairs of anchor nodes and it exchanges them if the penalty decreases by doing so. The technique is as follows.

(a) First, we sequentially exchange adjoining nodes clockwise in a manner like bubble sorting. However, unlike bubble sorting, there is no terminal element of a list because the nodes are on a circumference. If the penalty does not change after the exchange procedure makes a

complete circuit around the circumference, it stops. This procedure must stop because the penalty does not keep decreasing indefinitely. However, if we use only this condition for stopping, the solution tends to fall into a local minimum, which may not yield good results.

(b) Next, we vary the distance between the compared nodes from a large value to a small one. We start the comparison by choosing nodes far from each other, then gradually shorten the distance between nodes to be compared, and finally compare adjoining nodes at the end. For each distance, the comparison stops when the penalty does not change. In this way, it becomes unlikely that the solution will fall into a local minimum solution.

(c) To improve the efficiency of (b) at the end, the distance between nodes is made to be half the node distance of (b).

While the frequency of loop of every distance between compared nodes is $O(|A|^2)$ in (b), we expect that it becomes $O(|A| \log |A|)$ in (c). The procedure is listed as Algorithm 1 (the vertices run from 0 to $|A|-1$ in the algorithm).

Algorithm 1

```

int p0 = current_penalty;
int d = |A| / 2;
while (d > 0) {
    do {
        boolean c = false;
        for (int i = 0; i < |A|; i++) {
            int j = (i + d) mod |A|;
            swap i-th node and j-th node;
            int p1 = current_penalty;
            if (p1 < p0) {
                p0 = p1;
                c = true;
            } else {
                swap i-th node and j-th node;
            }
        }
    } while (c);
    d = d / 2;
}

```

4 Evaluation concerning aesthetic criteria

We implemented the technique described in Section 3 in Java and performed an experiment to evaluate its effect in terms of aesthetic criteria.

4.1 Evaluation Items

We wanted to assess

- (a) the validity of the penalty and
- (b) the effectiveness of the penalty minimization algorithm.

Regarding (a), we obtained layouts of many variations for a number of graph instances and examined the correlations between the penalty and average edge length and between the penalty and the number of edge crossings for each

graph. We judged the penalty to be valid as an alternative index if there was a certain correlation between the penalty and these values.

Regarding (b), we assumed that the penalty from (a) is valid. We examined how well the solutions obtained by the algorithm compared with the optimal solutions. We judged the algorithm to be effective if the quasi-optimal solution obtained by the algorithm was closer than a random order to the optimal solution.

4.2 Outline of the evaluation

Besides the proposed algorithm, we implemented two programs in Java; one obtained the optimal order concerning the penalty, and the other obtained a random order. We generated six kinds of bipartite graph. The three programs obtained a total of 1000 kinds of layout for each of the six graphs. The 1000 layouts had the following properties.

1. One optimal arrangement of the penalty
2. One Quasi-optimal arrangement of the penalty (obtained by Algorithm 1)
3. 998 random arrangements

Although the random arrangements do not need the penalty computation, we computed the penalties of all layouts as well as the correlations between the penalties and total edge lengths and the correlations between the penalties and the number of edge crossings.

Table 1 shows the size of the six bipartite graphs used for the evaluation. Every graph had 10 anchors. Note that our technique does not need to be used when there are only a few anchors because it would not take a long time to compute all possible orders in this case. On the other hand, when the anchor nodes exceed 10 in number, it becomes difficult to obtain optimal solutions for all graphs.

Table 1: Size of the graphs used for the evaluation

| | $ A $ | $ B $ | $ E $ |
|----|-------|-------|-------|
| G1 | 10 | 16 | 65 |
| G2 | 10 | 19 | 84 |
| G3 | 10 | 40 | 182 |
| G4 | 10 | 48 | 251 |
| G5 | 10 | 98 | 473 |
| G6 | 10 | 99 | 512 |

4.3 How to generate random graphs

Let n and p be parameters to generate a random graph. n denotes the number of anchors, and p denotes the appearance probability of free nodes. There can be 2^n kinds of free node when paying attention to the connecting pattern to the anchors (isolated free nodes, which are not connected any anchors, are also included in this number). The arrangement of free nodes connected to at most one

anchor is not influenced by the order of the anchor nodes. Thus, we generate only random graphs whose free nodes have at least degree 2. We choose free nodes of degree 2 or more with probability p . Two graphs generated with the same n and p might have different numbers of edges because p is not dependent on the connecting patterns to the anchor nodes.

The values of p for graphs G1, ..., G6 are respectively 0.02, 0.02, 0.05, 0.05, 0.1, and 0.1. We assumed that p was at most 0.1 because the density of the edges rises too much when there are more free nodes, and readable layouts cannot be expected for such anchored maps.

4.4 Evaluation result

We obtained 1000 sets of penalty, average edge length, and edge crossings for each of the six graphs. Table 2 shows the correlations calculated from them, and all of them are 0.8 or more. The penalty thus seems to be useful as an alternative index as long as we can see the correlations between it and the average edge length and between it and the number of edge crossings. We also found that scatter charts could represent relations between the penalty and the average edge length, and between the penalty and the number of edge crossings. Figure 4 shows a scatter chart, which represents the relation between the penalty and the number of edge crossings in G4. Certainly, a general trend is that the number of edge crossings is small in a layout whose penalty is small. However, we can see that some cases with few edge crossings exist and their penalties are not the minimum values.

Table 2: Correlations of the penalty with the average edge length and the number of edge crossings

| | Correlation between penalty and average edge length | Correlation between penalty and number of edge crossings |
|----|---|--|
| G1 | 0.884 | 0.828 |
| G2 | 0.872 | 0.829 |
| G3 | 0.893 | 0.878 |
| G4 | 0.831 | 0.816 |
| G5 | 0.904 | 0.893 |
| G6 | 0.875 | 0.863 |

Next, we ranked the 1000 kinds of layout according to the aesthetic criteria for each graph. Table 3 shows the ranks of the optimal layout and the quasi-optimal layout. From the left, it shows the ranks for the penalty, the ranks for the average edge length, and the ranks for the number of edge crossings, respectively. The ranks of the optimal layout for the penalty are omitted because these were always the first place. The quasi-optimal solutions give considerably good results if we regard only the ranking for the penalty. However, some ranks of quasi-optimal layouts are rather low in regard to the average edge length or number of edge crossings. All cases except the number of edge crossings

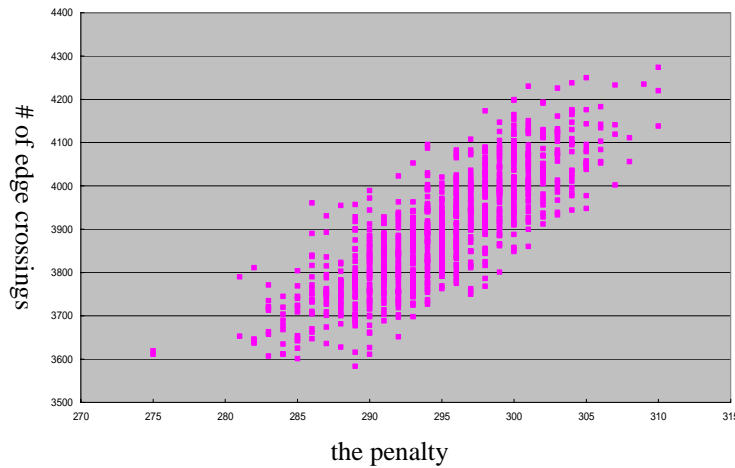


Figure 4: Correlation between the penalty and the number of edge crossings

in G2 are ranked within the 65th place in 1000; the rank of the number of edge crossings in G2 is 144th.

These data led us to conclude that the penalty works moderately well. Moreover, we think that the proposed algorithm works well enough. We guess that the goodness (or badness) of the index caused the ranks concerning the aesthetic criteria to be a little low. Therefore, it will be necessary for us to look for a better alternative index.

By the way, the penalty used in the experiment assumed that $q = 1$ in Expression (3). We also examined a penalty of $q = 2$ and $q = 3$. When q becomes large, both the correlation between the penalty and the average edge length and the correlation between the penalty and the number of edge crossings tended to become small, against the expected effect of q as described in Section 3.3.3.

We also computed the correlation between the average edge length and the number of edge crossings. For all six graphs, we obtained high correlations of more than 0.94. This means that we can use the common measure to decrease both of the average edge length and the number of edge crossings.

Table 3: Ranks of optimal solutions and quasi-optimal solutions concerning penalty, average edge length, and number of edge crossings

| | penalty | Average edge length | | Number of edge crossings | |
|----|---------|---------------------|---------|--------------------------|---------|
| | q.-opt. | opt. | q.-opt. | opt. | q.-opt. |
| G1 | 2 | 1 | 6 | 1 | 33 |
| G2 | 3 | 1 | 53 | 1 | 144 |
| G3 | 3 | 2 | 18 | 1 | 65 |
| G4 | 1 | 3 | 2 | 9 | 4 |
| G5 | 2 | 1 | 21 | 1 | 26 |
| G6 | 1 | 3 | 4 | 2 | 3 |

5 Application

Below, we show examples of anchored maps that visualize protein interaction networks.

When a protein is generated based on DNA, a transcription factor works. The kind of protein that is generated depends on the transcription factor. A transcription factor generates one or more kinds of protein, and a protein is generated from one or more kinds of transcription factor; there is an n-to-n relation between the proteins and the transcription factors. We guessed that proteins generated from the same transcription factor would have a similar character and that transcription factors generating the same protein would have a similar character. We expected that we could intuitively grasp the global relational

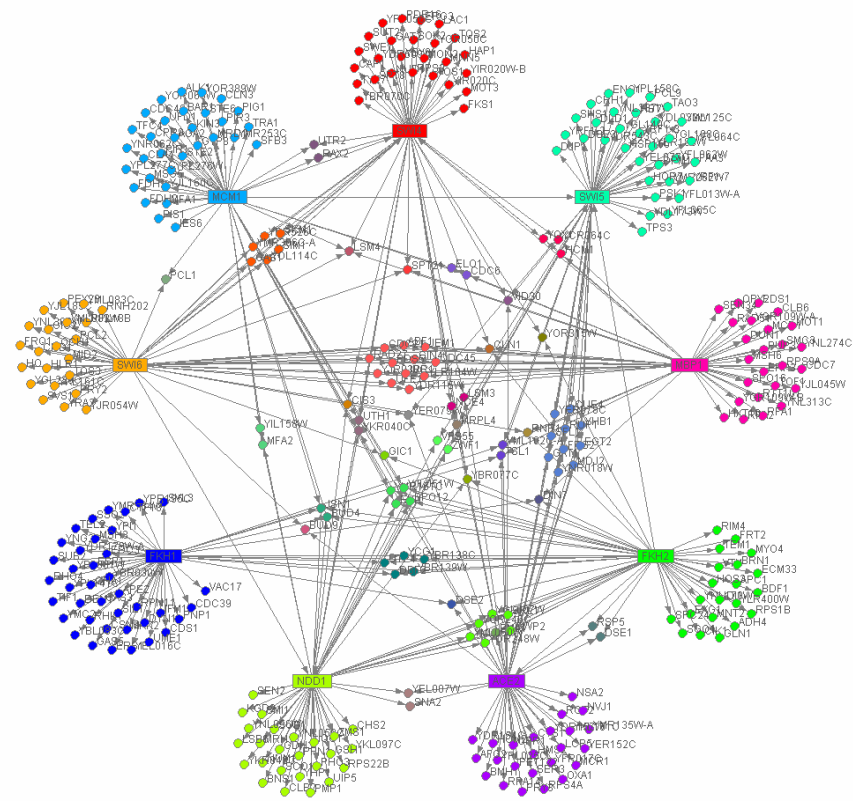
structure between proteins and transcription factors if we see such relationships graphically.

We expressed the relationships between the proteins and the transcription factors as anchored maps in which the transcription factors are anchors. Note that, in fact, the relationships between proteins and transcription factors are not bipartite graphs because transcription factors are also proteins.

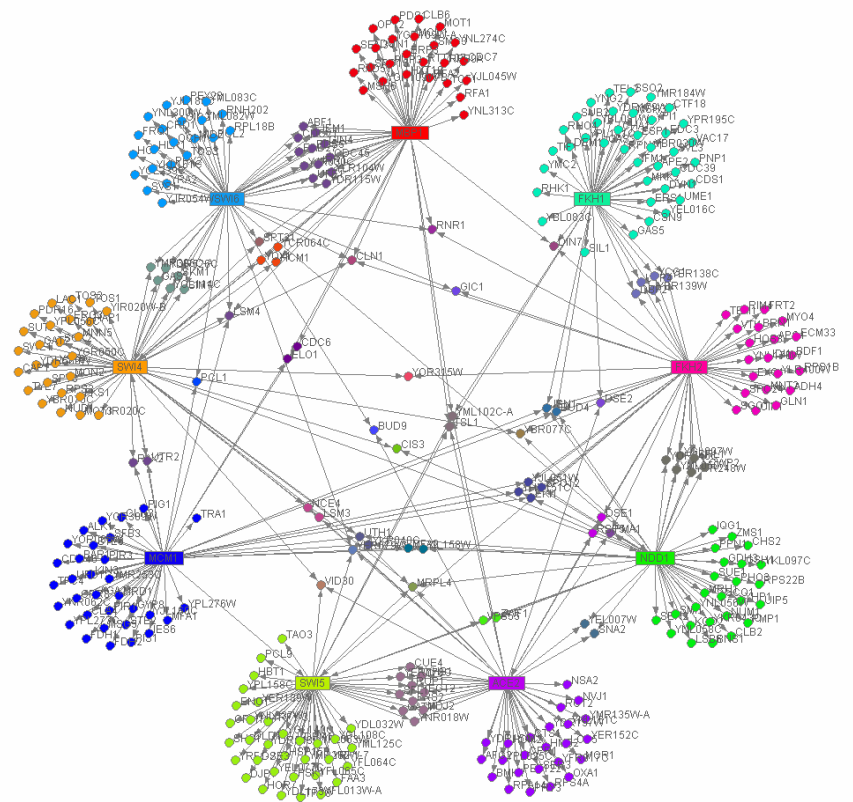
Figure 5 and Figure 6 show interaction networks of yeast proteins. Figure 5 shows the relationships between nine representative transcription factors related to the cell cycle and 276 proteins related to the factors. Figure 6 shows a larger example; it shows 63 transcription factors whose functions are comparatively clear and 1315 proteins related to the factors. As described above, the relations between the proteins and the transcription factors are not bipartite graphs. Here, we deleted the edges toward the transcription factors to make the bipartite graphs. Refer to (Harbison et al 2004) for more information about the protein data.

We drew two kinds of diagrams to see the effect of the proposed technique. In Figure 5(a) and in Figure 6(a), the anchors are arranged at random. In Figure 5(b) and in Figure 6(b), the orders of the anchor nodes are computed by using the proposed technique.

In the random layouts, we can see that the density of the central parts of the diagrams is high, and some transcription factors related to the common proteins are placed far away from each other. As the result, proteins which might have no relationship with each other (in other words, which are generated from quite different transcription factors) gather in a central part of the diagram; we cannot see features like clusters of proteins with similar natures. In contrast, the effect of the proposed technique is obvious. In Figure 5(b) and in Figure 6(b), transcription factors that relate to each other are placed together, and the central parts of the diagrams are sparser. In addition, it seems that the proteins are moderately scattered, and proteins with similar relationships with the transcription factors are placed close to each other.

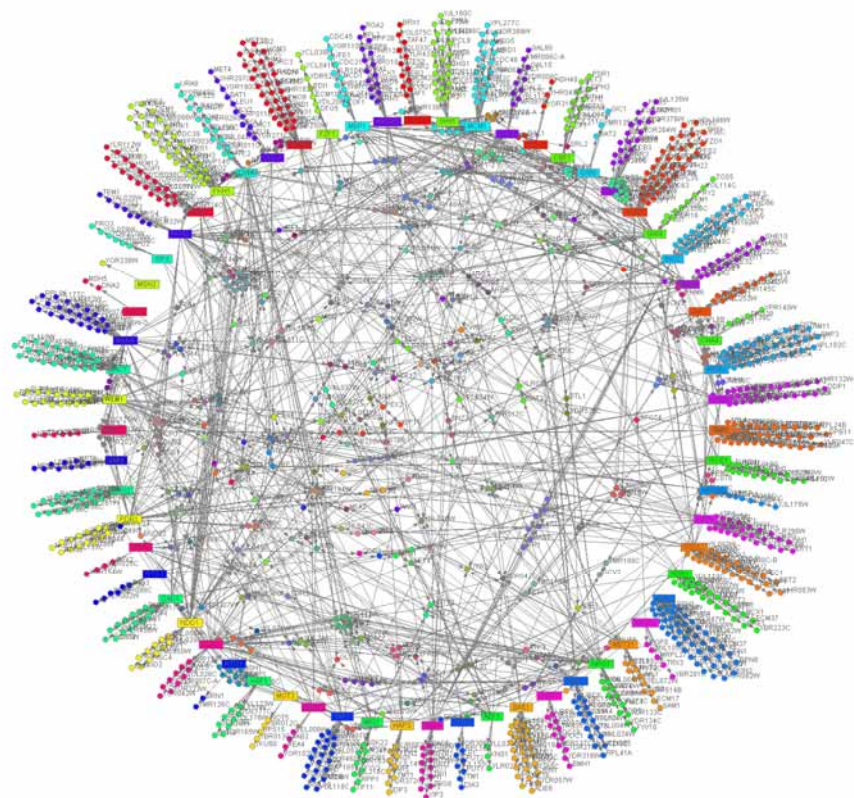


(a) Anchor nodes in random order

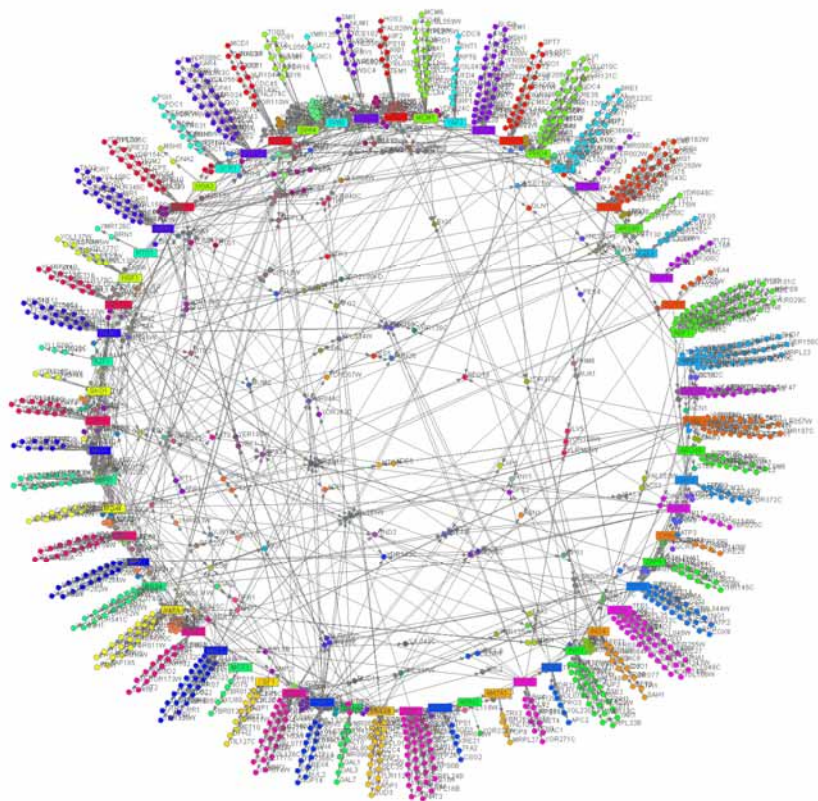


(b) Order of anchor nodes computed by using the proposed technique

Figure 5: Interaction networks of yeast proteins (9 transcription factors)



(a) Anchor nodes in random order



(b) Order of anchor nodes computed by using the proposed technique

Figure 6: Interaction networks of yeast proteins (63 transcription factors)

6 Related work

Visual Who (Donath 1995) is a tool whose purpose is to visualize communities. The tool visually expresses the appearance of the communities based on some mailing lists by using information statistically extracted from the text data of the mailing lists. The positions of the nodes are computed by spring embedding. The user can arrange an arbitrary mailing list as anchors, and the layout of the nodes represent member changes. This tool is interactive, and the user may arrange the anchors manually. It does not provide automatic layout facilities for the anchors.

SQWID (McCrickard & Kehoe 1997) is a Web search tool that expresses the retrieval result of WWW by using anchored maps. Terms used in the query are placed on the vertices of a triangle as anchor nodes. Web pages (or sites) are placed according to a related level with these terms. They limited anchors to three so that the relationships between the arranged Web pages and the fixed terms should not become vague. However, we think that there are a lot of situations in which four or more anchors are needed, and that a automatic technique to find the arrangement of anchor nodes should be developed.

7 Conclusions

We proposed an anchored map drawing technique for bipartite graphs, and explained the two main ideas of the technique. One idea concerns the index to decide the order of anchor nodes. As the index, we defined a penalty based on gaps between anchor nodes connected with a common free node. The other idea is an efficient algorithm to decide the order of anchor nodes by using the index. We implemented these ideas in Java and evaluated the effectiveness of the ideas. We generated six kinds of random bipartite graphs as data for the evaluation experiment. We computed layouts of these graphs by using our technique and one using a random ordering of

anchor nodes, and considered the results according to aesthetic criteria. The penalty seems to be useful as an alternative index as long as we can see the correlations between it and the average edge length and between it and the number of edge crossings. Moreover, we think that the proposed algorithm is good enough.

We have yet to complete our evaluations for all aesthetic criteria. In addition, we should try to find an alternative index that is better than the penalty described in this article, and to develop other algorithms to find good orders of anchor nodes based on the index.

Acknowledgements

The author would like to thank Professor Yasubumi Sakakibara and Mr. Yuji Kawada of Keio University for their valuable advice on the interaction network of proteins. This research was supported by the Artificial Intelligence Research Promotion Foundation.

References

- Eades, P. (1984), A Heuristic for Graph Drawing, *Congressus Numerantium* 42, pp. 149-160.
- Harbison, C. T. et al. (2004), Transcriptional regulatory code of a eukaryotic genome, *Nature*, 431, pp. 99-104.
- Misue, K. & Watanabe, I. (1999), Visualization of Keyword Association for Text Mining, In *IPSJ SIG Technical Report 1999-FI-55-8*, Information Processing Society of Japan, pp. 65-72 (in Japanese).
- McCrickard, S. D. & Kehoe, C. (1997), Visualizing Search Results using SQWID, In *Proc. of the sixth International World Wide Web Conference (WWW 6)*.
- Donath, J. S. (1995), Visual Who: Animating the affinities and activities of an electronic community, *ACM Multimedia 95 – Electronic Proceedings*.

Mental Map Preserving Graph Drawing Using Simulated Annealing

Yi-Yi Lee, Chun-Cheng Lin, Hsu-Chun Yen*

Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan 106, R.O.C.

E-mail: {sthugo,sanlin}@cobra.ee.ntu.edu.tw, yen@cc.ee.ntu.edu.tw

Abstract

Information visualization has attracted much attention in recent years in many fields of science and engineering. In many applications, graphs are ‘dynamic’ in the sense that changes are constantly applied to a graph to reflect the evolution of the system behaviour represented by the graph. In the past, the concept of the so-called “mental map” has largely been ignored. Users often have to spend a lot of time relearning the redrawn graphs. This paper proposes an effective way to release the user from such kind of a distasteful job by maintaining a high degree of the “mental map” for general graphs when graphs are redrawn. Our experimental results suggest this new approach to be promising.

Keywords: mental map, simulated annealing, graph drawing.

1 Introduction

Using graphs to represent real-world concepts has been widely used in many areas such as social relationship, interactive systems, database, automata theory, and more. In many cases, users may interact with the graphs, such as adding/deleting one or more nodes/edges, on a regular basis. As drawing graphs manually becomes infeasible for graphs of high complexity in structure and size, many automatic graph drawing methods have been developed. Most of the existing automatic drawing methods draw graphs according to a pre-built model. Such drawing methods are convenient and simple but have some limitations. Since only a few methods maintain the previous information which was formed in the users’ mind after they read the graphs, these techniques cannot preserve the users’ “mental maps”. The redrawn graph may differ a lot in structure in comparison with the original drawing, even in the case when only a slight modification was made to the original graph. When this happens, the user often has to spend a considerable amount of extra time to recognize the new graph. When the frequency of modification increases, this problem becomes more serious.

Suppose Figure 1(a) is the drawing of a graph using the well-known spring algorithm. After adding node 4 and connecting it to the rest of the graph, running the spring algorithm produces the layout displayed in Figure 1(b). Clearly Figure 1(c) is a better one for preserving the mental map since it inherits more information from the previous graph (i.e., Figure 1(a)) in comparison with Figure 1(b). Figure 2, from (Kaufmann and Wiese, 2002), is another example illustrating the importance of the preservation of the mental map. In Figure 2, adding a single edge is going to harm the mental map considerably. The initial graph consists of a chain of quadrangles, which are displayed nicely by the layout algorithm “Circular with Radial” (see (Kaufmann and Wiese, 2002)). When we connect the two outermost quadrangles by a single edge which is marked by the thick curve connecting from the highest node to the lowest node as shown in Figure 2(a), all of the components collapse into one, as shown in Figure 2(b). The Circular with Radial algorithm places the vertices on a circle and then performs a crossing minimization heuristics, which still results in at least $n/4$ crossings, one from each of the quadrangles. Unfortunately, the new drawing suffers from a rather small angular resolution such that the previous structure can hardly be recognized. An alternative layout (Figure 2(c)) for this graph shows the complete ring of components as it arose from the construction. When the graph size grows, the problem of then mental map will be more critical.

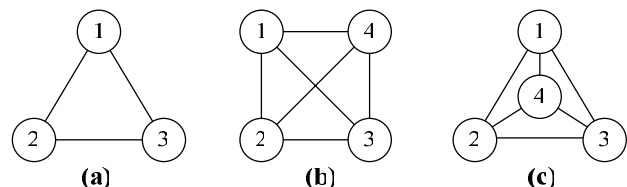


Figure 1. An example of the mental map problem.

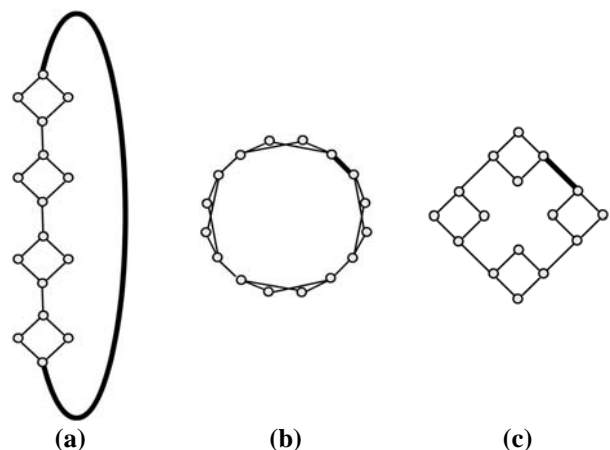


Figure 2. Example of the mental map problem in circular drawing (Kaufmann and Wiese, 2002).

Copyright © 2006, Australian Computer Society, Inc. This paper appeared at *Asia Pacific Symposium on Information Visualization 2006 (APVIS 2006)*, Tokyo, Japan, February 2006. Conferences in Research and Practice in Information Technology, Vol. 60. K. Misue, K. Sugiyama and J. Tanaka, Eds. Reproduction for academic, not-for profit purposes permitted provided this text is included.

* Research supported in part by NSC 94-2213-E-002-086, Taiwan.

The *mental map* concept was first proposed in (Eades et al., 1991). *Mental map* can be thought of as the presentation of a person's mind. When one sees a graph, there is a mental map constructed in his mind. Some people refer to the problem of preserving the user's mental map as the stability problem, see, e.g., (Paulisch and Tichy, 1990).

Preserving the mental map allows the user to quickly recognize the new graph which is redrawn after a modification is made to the original graph. The existing algorithms solving the mental map problem mostly focus on maintaining the orthogonal ordering, such as (Misue et al., 1995), or adding constraints to the graph, such as (K. Böhringer et al., 1990), and (He and Marriott, 1998). Others approaches preserve the mental map on certain kinds of graph layouts, see, e.g., (Kaufmann and Wiese, 2002). To our understanding, solving the mental map problem for general graphs under more sophisticated criteria remains a challenge.

Drawing graph nicely means placing nodes and edges in a pleasing way. There are different aesthetic criteria in different areas. In (Davidson and Harel, 1996), a nice-looking graph drawing refers to a drawing which: (1) distributes nodes evenly, (2) makes edge-lengths uniform, (3) minimizes edge-crossings, and (4) keeps nodes from coming too close to edges. To draw graphs nicely, a simulated annealing method was used in (Davidson and Harel, 1996) with the cost function designed to respect the above requirements. The implementation of their work allows the user to adjust the relative weights among various parameters, although some of these criteria may be in conflict with each other. Simulated annealing offers a more flexible way of drawing graphs than many other algorithms. The criteria used in (Davidson and Harel, 1996) will be referred to as '*nice criteria*' throughout the rest of this paper.

There are many intuitive factors that affect the mental map. For example, (Bridgeman and Tamassia, 2002) formally defined those intuitive factors and presented some statistical results to evaluate their effects on human perception of similarity. They used many measurements to measure the similarity between different pictures, and then asked users to measure the similarity between the same set of graphs. Finally, they used a statistical method to find the relationship between these measurements and users feelings, and concluded the best of the measures evaluated as follows: *ranking*, *orthogonal ordering*, *λ -matrix*, *nearest neighbour within*, *average distance*, *weighted nearest neighbour between*, *shape*, and *relative distance*.

In this paper, we propose a mental map preserving graph drawing algorithm for general graphs, based upon the simulated annealing graph drawing algorithm of (Davidson and Harel, 1996) with the cost function incorporating six criteria of (Bridgeman and Tamassia, 2002) to reflect people's mental maps. Our evaluation is divided into three stages. The first stage uses the original simulated annealing approach to draw graphs. The second stage is to modify the graph slightly. In most cases, we add two nodes, one with two edges and the other with four edges. In the third stage, our new approach is applied to redraw the modified graph nicely while preserving the mental map. The experimental results suggest this new

approach to be promising. To a certain extent, our approach can help us preserve the contour and the relative positions of the drawing of a graph when a sequence of operations are applied. Similar to (Davidson and Harel, 1996), our approach allows the user to adjust the relative weights among various parameters, and hence is very flexible. In addition, we can see from the experimental results that there exists a trade-off between drawing graphs nicely and preserving mental map.

The contributions of this paper are summarized as follows:

1. The previous work of preserving the mental map focused on how to preserve the orthogonal ordering and adding additional constraints to the graph. To our understanding, no graph drawing algorithms aiming at preserving the mental map on general graphs while using more sophisticated criteria simultaneously have been reported in the literature.
2. Our approach allows the user to adjust the relative weights among various parameters, and hence is very flexible.
3. By taking samples among a group of students with basic knowledge of graphs and graph drawing, (Bridgeman and Tamassia, 2002) suggested a number of good criteria suitable for capturing the notion of the mental map. Our graph drawing approach incorporates the criteria of (Bridgeman and Tamassia, 2002) into the cost function of a simulated annealing algorithm to preserve the contour and relative positions of the drawing of a graph when a sequence of operations are applied.
1. Because our approach obtains the location of each node in each iteration, rendering the graph in each iteration results in a smooth process from the initial drawing to the final drawing – a desirable property in information visualization.¹

The rest of this paper is organized as follows. In Section 2, we introduce the simulated annealing method on which our graph drawing algorithm is based. Section 3 involves the design of the cost function to maintain the mental map. Section 4 shows the experimental results of our algorithm. Concluding remarks and future work are given in Section 5.

2 Drawing Graphs Nicely

As mentioned in the introduction, the four requirements of a "*nice drawing*" include distributing nodes evenly, making edge-lengths uniform, minimizing edge-crossings, and keeping nodes from coming too close to edges.

2.1 Introduction to Simulated Annealing

The algorithm presented in (Davidson and Harel, 1996) was based on the technique of *simulated annealing* (SA), a generalization of the Monte Carlo method for examining the equations of state and frozen states of n -body systems.

¹ The experimental results are not given in this paper because the application to a smooth process is intuitive.

This method simulates the freezing of liquids or the recrystallization of metals in the process of annealing. In the beginning the system is at high temperature and disordered. As the process proceeds, the temperature cools down and the system becomes ordered.

SA is flexible and has successfully been applied to solving various combinational optimization problems. It allows “uphill” moves — moves that spoil, rather than improve, the temporary solution. This unique feature allows SA to escape from a local minimal solution, although there is no guarantee that a global minimum can be reached eventually.

The first step of SA is to generate an initial configuration (either randomly or heuristically constructed) and initialize the so-called temperature parameter T . Then the following is repeated until the termination condition is satisfied. A new solution from the neighbourhood of the current solution is randomly selected. The energy cost of the new solution is calculated over the objective function and T is decreased. Let E and E' denote the current and new energy costs, respectively. If $E' < E$, the new solution will be accepted. If $E' > E$, this solution will be accepted with a probability which is a function of T and $(E' - E)$. This probability is from the Boltzmann distribution: $e^{-(E' - E)/kT}$ where k is the Boltzmann constant. (We assume $k = 1$ in the subsequent discussion.) This kind of “uphill” moves will be fewer as SA proceeds because T is decreased after each iteration.

The following are the basic steps of SA:

1. generate an initial configuration δ and an initial temperature T ;
2. repeat the following until the termination condition is satisfied:
 - (1) choose a new configuration δ' from the neighbourhood of δ ;
 - (2) let E and E' be the value of the cost function at δ and δ' respectively; if $E' < E$ or random $< e^{-(E' - E)/kT}$ then set $\delta \rightarrow \delta'$;
3. decrease the temperature T ;
4. if the termination condition is satisfied, stop; otherwise go to step 2.

2.2 The Graph Drawing Algorithm

The input to our algorithm is a pair (V, E) stored in an adjacency list, where V is the set of nodes and E is the set of edges. Each node in V has a unique index. The output of our algorithm is a graph drawn on an $n \times n$ rectangular grid, where n depends on the size of the graph. The following is the setting of SA needed to carry out the simulation of the physical annealing process:

A *configuration* is the placement of nodes and edges of the graph. If there is no prior information about the graph, we place the graph randomly. The *neighbourhood* of a configuration C is a configuration that is obtained from C by changing the location of a single node. In each iteration, we choose a node and then randomly assign a new location

to it. In the beginning, the distance between a configuration and its neighbourhood is very large. As the algorithm continues, the distance between two consecutive locations of a node decreases. This setting can accelerate the convergence of the algorithm.

2.2.1 The Cost Function

The *cost function* constitutes the most important part of SA, on which the efficiency of the algorithm highly depends. A good cost function should reflect the properties of the nice-looking pictures. Intuitively, all desired features of the final picture must contribute to this function, in accordance with their relative importance. The following are the criteria that we use in the design of the cost function. These criteria are intuitively clear, using which nice drawings of graphs are obtained.

- (1) *Node distribution*: This criterion is concerned with how to spread the nodes of a graph evenly. In a nice drawing, any two nodes should not be too close to each other. Nodes should not be overcrowded and should be evenly distributed in the drawing area. These intentions can be accomplished by the “*Node Distribution*” and “*Borderlines*” components of the cost function. The former deals with the distances between any two nodes, while the latter deals with the distances between a node and four borderlines of the space. For each pair of nodes i and j , the term:

$$\alpha_{ij} = \frac{\lambda_1}{d_{ij}^2}$$

is added to the cost function, where d_{ij} is the Euclidean distance between nodes i and j and λ_1 is a normalizing factor that defines the relative importance of this criterion compared to others. Increasing λ_1 relative to other normalizing factors will produce a smaller graph.

- (2) *Borderlines*: This component prevents nodes from coming too close to the borderlines. This element and *Node Distribution* can disperse the nodes evenly. For each node i , the term is added to the cost function:

$$m_i = \lambda_2 \left(\frac{1}{r_i^2} + \frac{1}{l_i^2} + \frac{1}{t_i^2} + \frac{1}{b_i^2} \right)$$

where r_i, l_i, t_i and b_i stand for the distances between node i and the right, left, top and bottom sides, respectively; λ_2 is the normalizing factor, and increasing λ_2 relative to other normalizing factors will make the graph towards the center of the drawing space.

- (3) *Edge lengths*: The previous two ingredients are related to nodes. Now we discuss the criteria about edges. This component is slightly different from that used in (Davison and Harel, 1996). They used the term, $c_k = \lambda_3 d_k^2$ where d_k is the length of edge k . In our setting we use the term

$$c_k = \lambda_3 s^2$$

where s is the standard deviation of edge lengths and λ_3 is the normalizing factor. The reason why we use standard deviation instead is that using only edge length in designing the cost function tends to produce smaller graph layouts that are undesirable in some cases. Our new criterion ensures that edge lengths do not vary too much, though this alternative criterion costs more time than using that proposed in (Davison and Harel, 1996).

- (4) *Edge crossings*: Minimizing the number of crossing is an important goal, but it is in general difficult to achieve. Too many crossings degrade the quality of a drawing, which often cause unnecessary confusions when people reading the graph. Although the output of our SA approach may not reach the minimal number of crossings theoretically, it produces a quite pleasing picture in most cases as our experimental results indicate. To incorporate this component we simply add to the cost function the term:

$$e = \lambda_4 f_k f_l$$

where f_k is the weight attributed to edge k , again λ_4 is a normalizing factor that defines the relative importance of this criterion compared to others. In this case, the normalizing factor, λ_4 , is defined as

$$\lambda_4 = \frac{\lambda_5}{g_{\min}^2}$$

where λ_5 is the normalizing factor of the next component, *Node-edge Distance*, and g_{\min} is the minimal distance between all nodes and edges in the current configuration. This component constitutes the most time consuming one among all terms in the cost function.

- (5) *Node-edge distances*: The *Edge Crossing* component penalizes edges that cross with other edges, but does not penalize the condition in which nodes become too close to neighbouring edges. Nodes that are too close to their neighbouring edges are likely to confuse the user. So for node k and edge l with distance g_{kl} , the value of

$$h_{kl} = \frac{\lambda_5}{g_{kl}^2}$$

is added to the cost function. λ_5 is the normalizing factor, and larger λ_5 prefers longer distances between nodes and edges.

2.2.2 The Cooling Schedule

The cooling schedule is one of the key parts of the annealing algorithm. Most applications follow the basic geometric schedule described in (van Laarhoven and Aarts, 1987).

1. *Initial temperature*: Generally, we set the initial temperature high enough to accept almost every move in the beginning, in order for any configuration to work as the starting point of the algorithm. In some

situations, however, when the initial configuration is known to have some resemblance to the desired final result, a lower initial temperature can be chosen at the beginning..

2. *Temperature reduction*: The *temperature reduction* stage determines when to change the temperature and how to change it. This stage affects the number of moves at each temperature and the range of each move. Typical applications set the number of trials at each temperature to be polynomial in the size of the input. We set the moves at each temperature to be 30 times the number of nodes. The cooling schedule rule is geometric. If T_p is the temperature at the p -th stage, then the temperature at the next stage is

$$T_p = \gamma T_{p-1}$$

with γ is a real number between 0.6 and 0.95. The range of each move decreases when the temperature reduces. Given the range of movement at the p -th stage is R ; the range of movement serving for the next temperature will be γR , which accelerates the convergence of the algorithm.

3. *Termination condition*: This part is to decide when to stop the algorithm. Commonly used termination conditions include: stopping the algorithm when the solution does not change after several consecutive stages, or fixing the number of stages to be a constant. Evidence has shown that, in many cases, substantially changes happen in the first few stages and the following stages yield marginal improvement.

3 Maintaining the Mental Map

When a user looks at a graph, he or she learns about the graph's structure, and subsequently navigates through the drawing to understand its meaning. This is what we called the "*mental map*." This idea can be used in many areas. Since there are many real-world concepts captured by means of graph representations (such as flow-chart, social relationship, automata, ..., etc), preserving the user's mental map from the existing information is getting more and more important as the size of the graph increases.

In an interactive system, users or applications may modify the graph very often. As we explained earlier, adding one node and three edges to Figure 1 may cause some confusion. If we make more changes, the situation is likely to get worse. Although animation can be used to provide a smooth transition between consecutive drawings, it is still important to preserve some degree of similarity between drawings. If we cannot maintain one's mental map, he or she may have to spend a lot of time relearning the new graph.

In the past, maintaining the mental map in graph drawing has only been scarcely studied. Most of these works focused on how to preserve the orthogonal ordering and adding additional constraints to the graph. To our understanding, no algorithm aiming at preserving the mental map on general graphs while using more sophisticated criteria simultaneously has been reported in the literature. We feel that orthogonal ordering is not the only factor affecting one's mental map. Seeking additional

factors regarding the preservation of the mental map plays an important role in drawing, for example, dynamic graphs. Furthermore, adding additional constraints by the user becomes infeasible as the size of the graph increases. Regarded by many as the most complete work about the measurements of the mental map, Bridgeman and Tamassia (2002) delivered a user study based on several similarity measures, which reflect one's mental map according to the direct responses of users.

3.1 The Framework of Bridgeman and Tamassia

Bridgeman and Tamassia (2002) used a statistical method to test the mental map. Their samples include 103 students in a CS course at Brown University. Before taking the test the students had eight lectures on graphs and graph algorithms, including one on graph drawing. They had also been assigned a programming project involving graphs, so they had some familiarity with the subject. Though the study was done on orthogonal graphs, these measurements perform equally well on general graphs.

Each graph in the experiment was produced by Giotto and InteractiveGiotto. These graphs can be divided into two categories: base drawings and modified drawings. The base drawings were drawn by Giotto, while the modified drawings were produced by InteractiveGiotto, which simulated a user's action by adding two nodes, one with two edges and the other with four edges. InteractiveGiotto preserves edge crossings and bends.

In order not to affect the correction of the experiment, students were allowed to take a break after each task; so they did not have to stay focused for too long without a break. At each part students were asked to decide which graph resembles the base drawing or to find out which vertex in the modified graph is not in the base drawing. The choices as well as the response time the students made were recorded. It was assumed that the more time students spent, the less similar these graphs were. If the graphs were too different for the users to make a choice, they could choose the "cannot decide" answer. On the other hand, if the users spent less time answering a task, then the modified graph in the task tends to be more similar to the base graph.

Based on the above experiment, Bridgeman and Tamassia (2002) suggested a number of measurements that are best suited for capturing a user's mental map. We choose these criteria to enrich our cost function.

3.2 Redrawing Graphs while Preserving the Mental Map

In this stage, we use the simulated annealing method to draw graphs. Most of the settings in this stage are the same as those discussed in the previous section. We only explain the differences. The value of each component of the cost function is nonnegative. Given a value of a component of the cost function M , $M = 0$ implies that these two graphs are thought to be the same according to this component. The larger magnitude of M means a higher degree of difference between the graphs.

3.2.1 The Cost Function

To maintain the mental map, we add several criteria to our cost function. In addition to the components mentioned previously, we enrich the cost function with the following components. These components are designed to reflect the critical factors regarding the preservation of the mental map based on the experiments reported in (Bridgeman and Tamassia, 2002). In the following, P and P' refer to the point sets for graph layouts D and D' , respectively, and $p' \in P'$ is the corresponding point for $p \in P$ (and vice versa). Let $d(p, q)$ be the Euclidean distance between points p and q .

1. *Rankings*: Considering the fact that the relative horizontal and vertical positions of a node should not vary too much, we add the component, *ranking*, to the cost function. Let $right(p)$ and $above(p)$ be the numbers of points to the right and above of p , respectively. λ_6 is the normalizing factor.

$$rank(P, P') = \frac{1}{\lambda_6} \sum_{p \in P} \min\{|right(p) - right(p')| + |above(p) - above(p')|\}$$

where $\lambda_6 = 1.5(|P| - 1)$.

2. *Average relative distance*: The *average relative distance* is the average of the distances between pairs of nodes. Users normally do not want to see the distances between nodes in the redrawn picture to vary too much because too many changes on the distances would destroy the mental map.

$$rdist(P, P') = \frac{1}{|P|(|P| - 1)} \sum_{p, q \in P} |d(p, q) - d(p', q')|$$

3. *Shape*: The *Shape* factor records the orientation of each edge of the graph as one of the following directions: north, south, east, west, north-east, south-east, north-west and south-west. By keeping the relative positions of any two nodes, we can maintain a better mental map. λ_7 is the normalizing factor.

$$shape = \frac{1}{\lambda_7} \sum_{e \in E} edits(e, e')$$

where $\lambda_7 = |E|$ and $edits(e, e') = \begin{cases} 1, & \text{if } e \neq e' \\ 0, & \text{otherwise} \end{cases}$

in which e' is the corresponding edge of e in D' .

4. *λ -matrix*: Two sets of points P and P' have the same order type if, for every triple of points (p, q, r) , they are oriented counter-clockwise if and only if (p', q', r') are also oriented counter-clockwise. The λ -matrix is designed to maintain the kind of an order type. By using a simplified version, we calculate the number of nodes left to two designated nodes. Let $\lambda(p, q)$ be the number of points in P to the left of the directed line from p to q and n be the size of P . λ_8 is the normalizing factor.

$$\text{lamda}(P, P') = \frac{1}{\lambda_8} \sum_{p, q \in P} |\lambda(p, q) - \lambda(p', q')|$$

$$\text{where } \lambda_8 = n \left\lfloor \frac{(n-1)^2}{2} \right\rfloor.$$

5. *Nearest neighbour within*: This component captures the intuitive idea that a node's nearest neighbour should remain the closest one to it after the graph is redrawn. Let $nn(p)$ be the nearest neighbour of p in P and $nn(p)'$ be the corresponding node in P' to $nn(p)$. λ_9 is the normalizing factor.

$$nnw(P, P') = \frac{1}{\lambda_9} \sum_{p \in P} \text{nearer}(p)$$

$$\text{where } \lambda_9 = |P| \text{ and } \text{nearer}(p) = \{q \mid d(p', q') < d(p', nn(p)'), q \in P, q \neq p, q \neq nn(p)\}$$

6. *Nearest neighbour between*: This concept is based on the assumption that a node should not move too far from the original position. And its new location should be the closest one to its previous location. When the new drawing is presented, the user ought to be able to find a certain node according to the mental map. λ_{10} is the normalizing factor.

$$nmb(P, P') = \frac{1}{\lambda_{10}} \sum_{p \in P} \text{nearer}(p)$$

$$\text{where } \lambda_{10} = |P| (|P| - 1) \text{ and } \text{nearer}(p) = \{q \mid d(p, q') < d(p, p'), q \in P, q \neq p\}$$

Figure 3 is the overview of our algorithm.

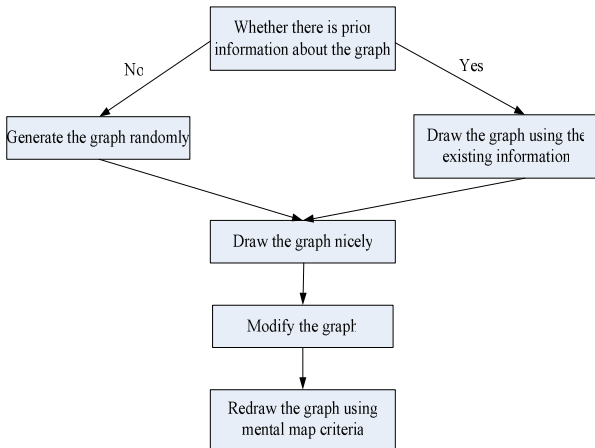


Figure 3. The overview of our algorithm.

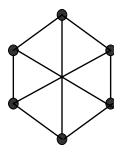


Figure 4. An example.

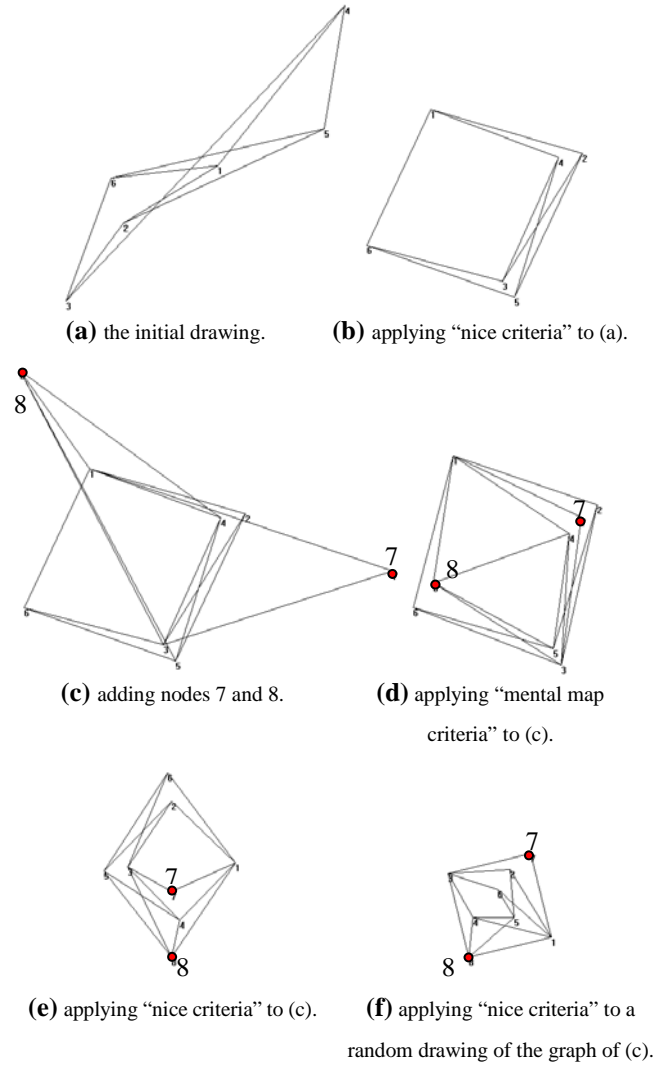


Figure 5. Various drawings illustrating the preservation of the mental map.

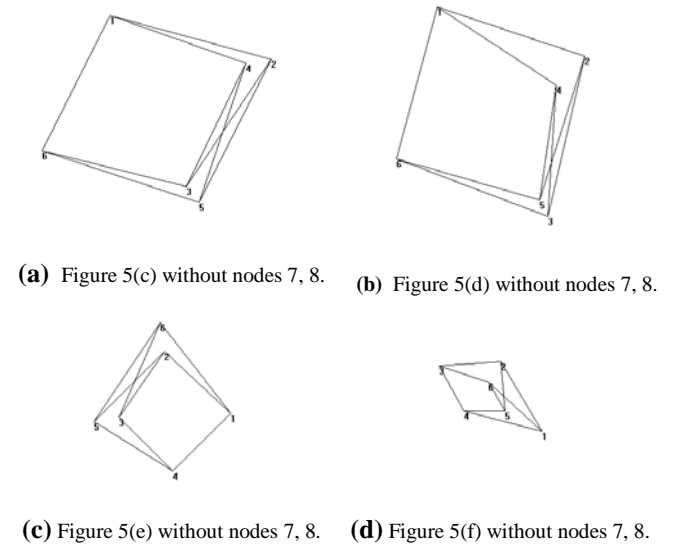


Figure 6. Removing nodes 7 and 8 from Figures 5(c)–(f).

4 Experimental Results

This section shows a number of examples to which our mental map preserving drawing algorithm is applied. We first draw a graph using the “*nice criteria*” explained in Section 2. Like the experiments carried out in (Bridgeman and Tamassia, 2002), we add two nodes, one with two edges and the other with four edges. The locations of these nodes are chosen randomly. The connections between the added nodes and the original graph are decided randomly. Finally, we redraw the modified graph using our algorithm which incorporates the “*mental map criteria*” discussed in Section 3. For a comparison purpose, we also present the modified graphs which are drawn using the “*nice criteria*” only.

Our prototype system runs on a Pentium IV 2.4 GHz PC with 512 MB memory. Consider a very simple example shown in Figure 4. In our implementation, a random drawing was generated as shown in Figure 5(a). After applying the “*nice criteria*”, the SA-based drawing algorithm produced the graph displayed in Figure 5(b). Then two nodes, 7 and 8, were added whose locations as well as the connections to the remaining graph were decided randomly (see Figure 5(c)). The modified graph was redrawn by using our modified SA approach based on the “*mental map criteria*.” The result is shown in Figure 5(d). The running time required to generate Figure 5(d) is 0.172 second. To compare with drawings without taking into account the concept of mental map, we show two additional drawings. Figure 5(e) shows the drawing after applying the “*nice criteria*” to Figure 5(c). Figure 5(f) shows the drawing based on the “*nice criteria*” with respect to a randomly generated initial drawing of the modified graph.

In view of Figures 5(d) and 5(b), it is easy to see that their contours and relative positions are very similar. On the other hand, Figures 5(e) and 5(f) share little similarity with Figure 5(c). Further, if we delete the additional nodes 7 and 8 from the drawings displayed in Figures 5(c)-(f), it becomes clearer why Figure 5(d) preserves a better mental map (see Figure 6). Like many aesthetic criteria which are often in conflict with one another, there is a trade-off between maintaining a high degree of mental map and other criteria such as minimizing the number of crossings. Our algorithm allows users to adjust the weights between different criteria, in order to fit their drawing requirements.

This next example is a tree-like graph with 21 nodes and 20 edges shown in Figure 7. With respect to the initial drawing shown in Figure 8(a), Figure 8(b) displays the output drawing using the ‘*nice criteria*’ algorithm. Figure 8(c) shows the graph after nodes 22 and 23 (together with some edges) are inserted into the graph. Note that in Figure 8(c), only node 19 was pulled upper, node 21 was pulled lower and some edges with the same parent changed their relative positions. Figure 8(d) is the drawing produced by applying our mental map preserving algorithm to Figure 8(c). There are no substantial differences between Figures 8(b) and 8(d). Figures 8(e) and 8(f) are drawings obtained from Figure 8(c) without taking the mental map property into account. Although Figure 8(c) is a planar graph, Figure 8(d) displays two crossings; however, there are no crossings in Figure 8(e) and 8(f). Like we mentioned

earlier, preserving the mental map and achieving optimality with respect to some other aesthetic criteria may not be met simultaneously. Figure 9 simply displays the drawings after the additional nodes 22 and 23 are removed from Figures 8(c)-(f). Figure 9(b) still preserves the shape and structure of Figure 9(a). In contrast, Figures 9(c) and 9(d) look somewhat different from Figure 9(a).

Finally, we present a tree example which has 63 nodes and 62 edges. It takes 39.313 seconds to finish the drawing which is based on the ‘*nice criteria*’. Using the mental map criteria, the drawing takes 45.266 seconds. See Figure 10. See also Figure 11 which shows the drawings when the added nodes are removed from Figure 10. It is fairly easy to see the advantage of our drawing algorithm when preserving mental map is the primary concern in drawing graphs.

5 Conclusion and Future Work

In this paper, we presented a simulated-annealing-based graph drawing algorithm, taking the preservation of the mental map into account. The mental map criteria help us better preserve the contour and relative positions of the redrawn graph when a slight modification is made to the original graph, which in turn allowing the user to spend less time to relearn the new graph. Preserving the mental map and optimizing other aesthetic constraints are often in conflict with one another. Our implementation also allows the user to adjust the weights of different criteria.

In addition to the factors considered in this paper, graph features such as colour, node size, shape or line width may also play a key role in the preservation of the mental map. See (Franconeri et al., 2005) and (Rensink, 2002), for example, for work in this area from a psychological viewpoint. It is of interest and importance to take such additional features of graphs into account in the design of mental-map preserving graph drawing algorithms.

6 References

- Böhringer, K. and Paulisch, F. (1990): Using Constraints to Achieve Stability in Automatic Graph Layout Algorithms. *Proc. of the ACM SIGCHI Conference on Human Factors in Computing Systems*, 43-51.
- Bridgeman, S. and Tamassia R. (2002): A User Study in Similarity Measures for Graph Drawing. *Journal of Graph Algorithms and Applications*, 6(3): 225-254.
- Davidson, R. and Harel D. (1996): Drawing Graphs Nicely Using Simulated Annealing. *ACM Transactions on Graphics*, 15(4):301-331.
- Eades, P. (1984): A heuristic for graph drawing. *Congressus Numerantium*, 42:149-160.
- Eades, P., Lai, W., Misue, K., and Sugiyama K. (1991): Preserving the Mental Map of a Diagram. *Proc. of Compugraphics*, 91: 24-33.
- Fernandez-Duque, D. and Thornton, I. (2000): Change Detection Without Awareness: Do Explicit Report Underestimate the Representation of Change in the Visual System? *Visual Cognition*, 7: 323-344.

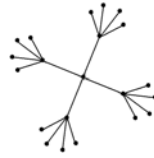
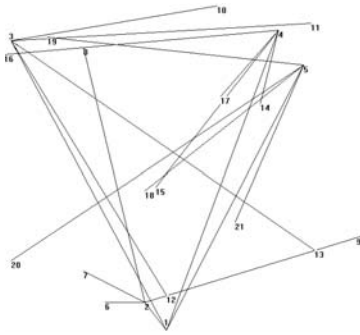
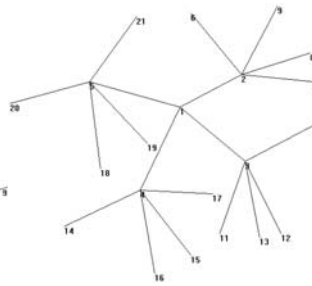


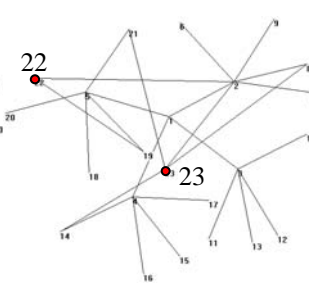
Figure 7. A tree example.



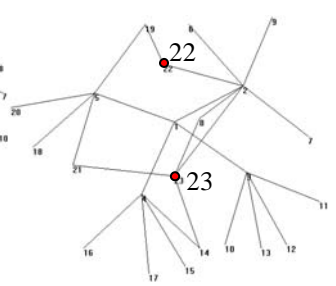
(a) the initial drawing.



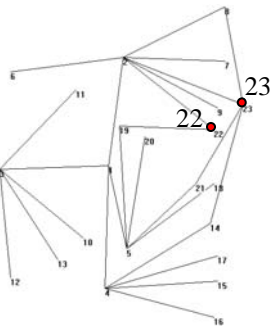
(b) applying "nice criteria" to (a).



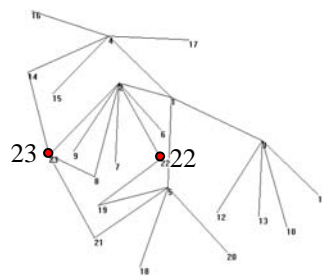
(c) adding nodes 22 and 23.



(d) applying "mental map criteria" to (c).

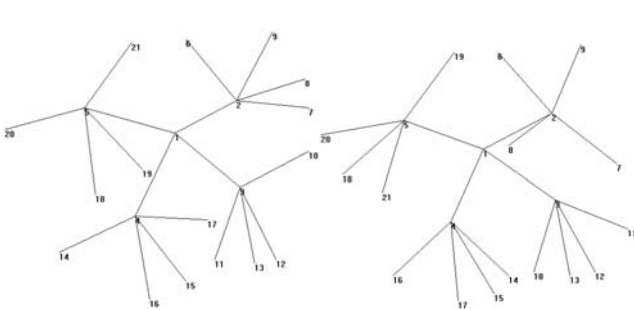


(e) applying "nice criteria" to (c).

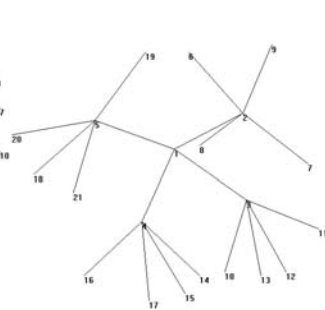


(f) applying "nice criteria" to a random drawing of the graph of (c).

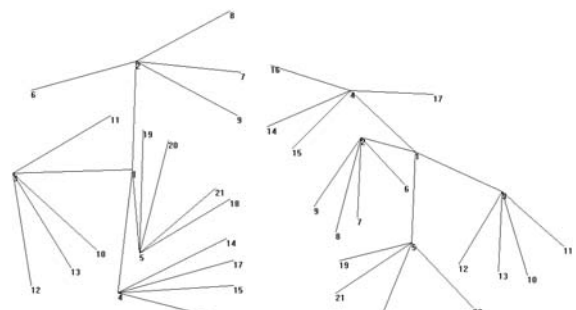
Figure 8. Various drawings illustrating the preservation of the mental map.



(a) Figure 8(c) without nodes 22 and 23.



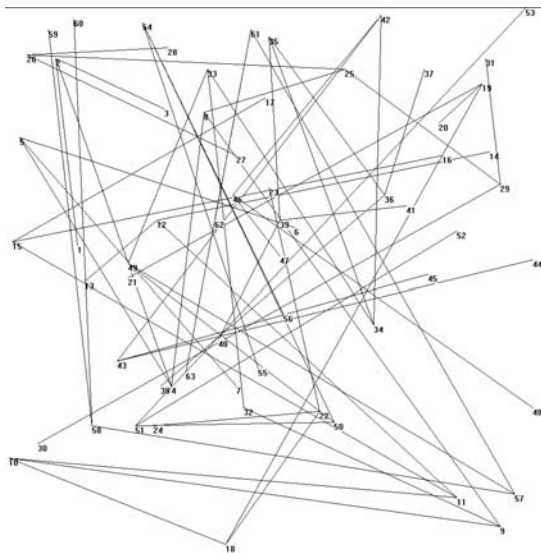
(b) Figure 8(d) without nodes 22 and 23.



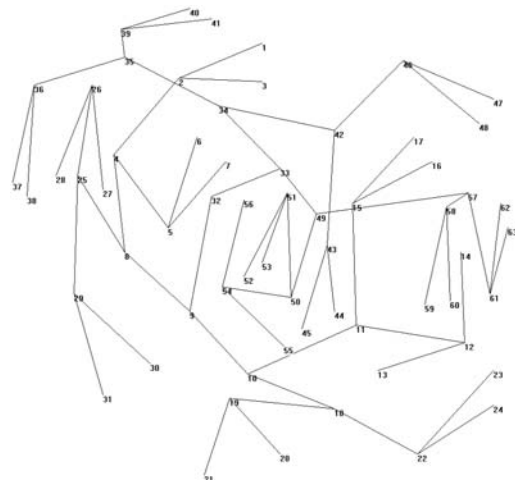
(c) Figure 8(e) without nodes 22 and 23.

(d) Figure 8(f) without nodes 22 and 23.

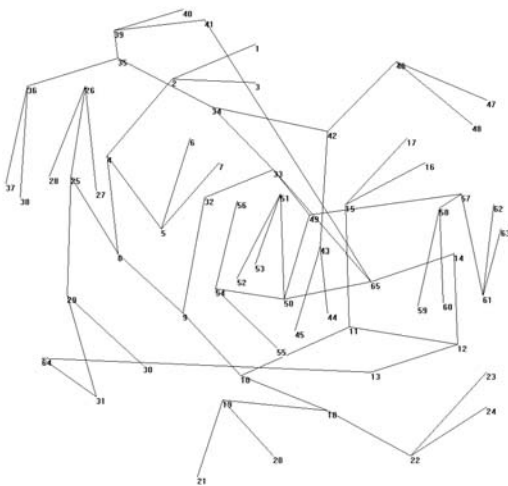
Figure 9. Drawings obtained from Figure 8 by deleting nodes 22 and 23.



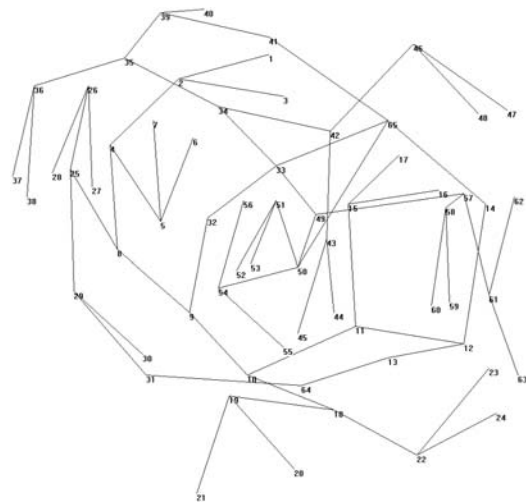
(a) the initial drawing.



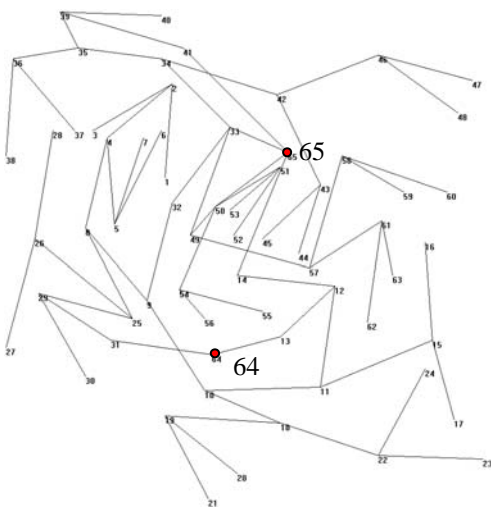
(b) applying "nice criteria" to (a).



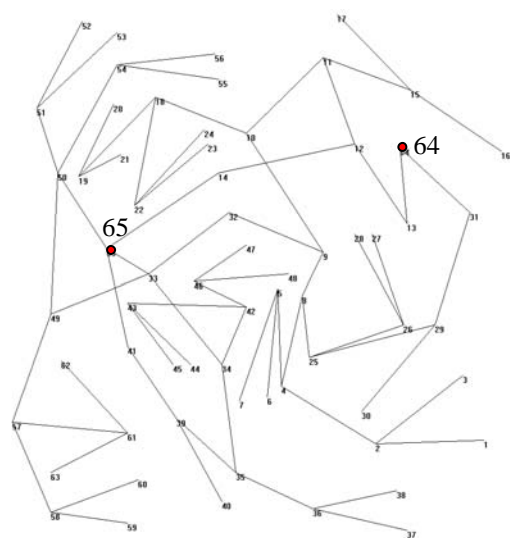
(c) adding nodes 64 and 65.



(d) applying "mental map criteria" to (c).

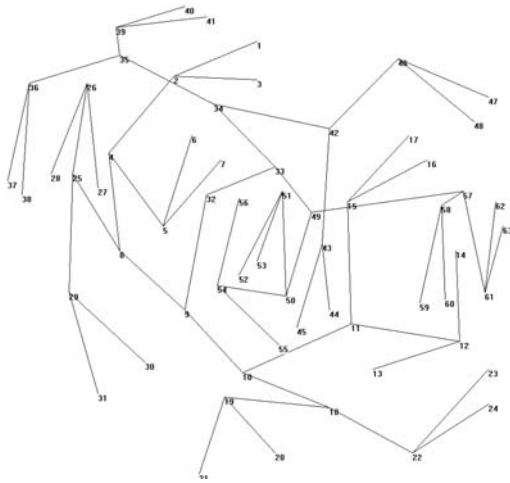


(e) applying "nice criteria" to (c).

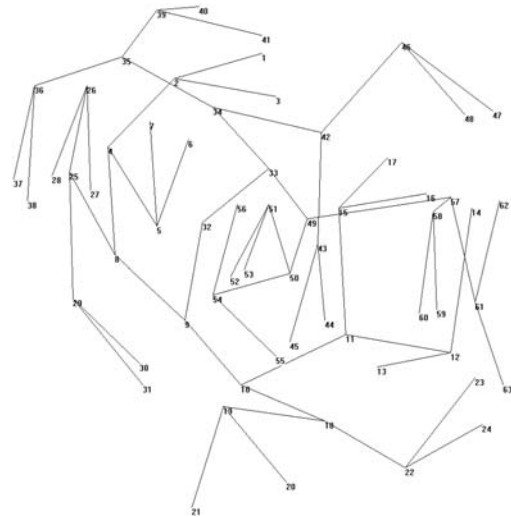


(f) applying "nice criteria" to a random drawing of the graph of (c).

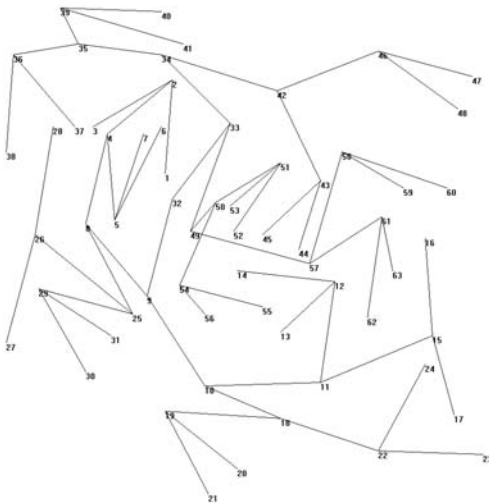
Figure 10. A larger tree example.



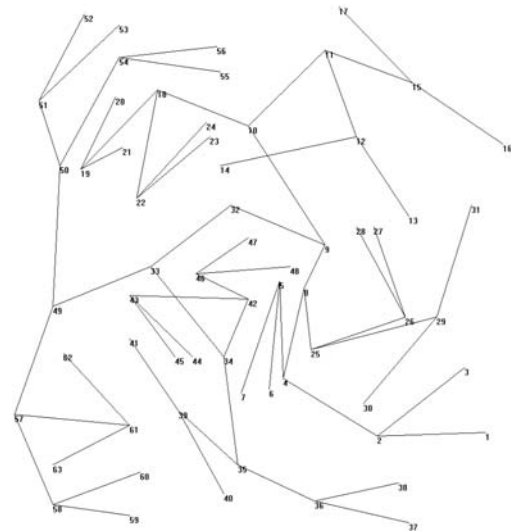
(a) Figure 10(c) without nodes 64 and 65.



(b) Figure 10(d) without nodes 64 and 65.



(c) Figure 10(e) without nodes 64 and 65.



(d) Figure 10(f) without nodes 64 and 65.

Figure 11. Drawings obtained from Figure 10 by deleting nodes 64 and 65.

- Franconeri, S., Hollingworth, A., and Simons, D. (2005): Do New Objects Capture Attention? *Psychological Science*, **16**(4): 275-281.
- Harel, D. and Sardas, M. (1998): Randomized Graph Drawing with Heavy-Duty Preprocessing. *J. Visual Language and Computing*, **6**: 233-253.
- Harrison, J., Rensink, R., and van de Panne, M. (2004): Obscuring Length Changes During Animated Motion. *ACM Transactions on Graphics*, **23**: 569-573.
- He, W. and Marriott, K. (1998): Constrained Graph Layout. *Constraints*, **3**(4): 289-314.
- Kaufmann, M. and Wiese, R. (2002): Maintaining the Mental Map for Circular Drawings. *Proc. of Graph Drawing GD'02*, volume 2528 of *Lecture Notes in Computer Science*, 12-22.
- Misue, K., Eades, P., Lai, W., and Sugiyama, K. (1995): Layout Adjustment and the Mental Map. *J. Visual Languages and Computing*, **6**: 183-210.
- Paulisch, F. and Tichy, W. (1990): Edge: an Extendible Directed Graph Editor. *Software – Practice and Experience*, **20**(S1): 63-88.
- Rensink, R. (2002): Change Detection. *Annual Review of Psychology*, **53**: 245-277.
- Storey, M. and Müller, H. (1995): Graph Layout Adjustment Strategies. *Proc. of Graph Drawing GD'95*, volume 1207 of *Lecture Notes in Computer Science*, 487-99.
- van Laarhoven, P. and Aarts, E. (1987): Simulated Annealing: Theory and Applications. Dordrecht: Reidel.

Visual Analysis of Network Centralities

Tim Dwyer*
Monash University,
Melbourne, VIC,
Australia

Seok-Hee Hong†
NICTA (National
ICT Australia) &
School of IT,
University of
Sydney, NSW,
Australia

Dirk Koschützki‡
Leibniz Institute of
Plant Genetics and
Crop Plant
Research,
Gatersleben,
Germany

Falk Schreiber§
Leibniz Institute of
Plant Genetics and
Crop Plant
Research,
Gatersleben,
Germany

Kai Xu¶
NICTA (National
ICT Australia),
Sydney, NSW,
Australia

Abstract

Centrality analysis determines the importance of vertices in a network based on their connectivity within the network structure. It is a widely used technique to analyse network-structured data. A particularly important task is the comparison of different centrality measures within one network. We present three methods for the exploration and comparison of centrality measures within a network: 3D parallel coordinates, orbit-based comparison and hierarchy-based comparison. There is a common underlying idea to all three methods: for each centrality measure the graph is copied and drawn in a separate 2D plane with vertex position dependent on centrality. These planes are then stacked into the third dimension so that the different centrality measures may be easily compared. Only the details of how centrality is mapped to vertex position are different in each method. For 3D parallel coordinates vertices are placed on vertical lines; for orbit-based comparison vertices are placed on concentric circles and for hierarchy-based comparison vertices are placed on horizontal lines. The second and third solutions make it particularly easy to track changing vertex-centrality values in the context of the underlying network structure. The usability of these methods is demonstrated on biological and social networks.

Keywords: network analysis, centralities, visualisation, graph drawing, biological networks, social networks.

1 Introduction

Network analysis methods support the study of structural properties in networks. One important method is centrality analysis which determines the relative

importance of vertices in a network based on their connectivity within the network structure. It is particularly useful in analysing social and biological networks. In social network analysis, a methodology which uses graph-theoretic concepts to analyse and understand the structure and behaviour of social networks (Wasserman & Faust 1994), centrality is a well known individual-level network analysis method. It measures the importance or prominence of the actors in a social network. For instance, in a research collaboration network, one can identify the most prominent and influential researchers in a particular research area. In a citation network, one can identify the most important and influential papers as cited by other scientists.

In the life sciences centrality measures help scientists to understand the underlying biological processes and have been successfully applied to different biological networks. Central vertices in protein-protein interaction networks are often functionally important and the removal of such vertices is related to lethality (Jeong, Mason, Barabási & Oltvai 2001). In metabolic networks metabolites with highest degree, i.e. with the highest number of neighbours, may belong to the oldest part of the metabolism (Fell & Wagner 2000) and are main metabolites in well-known pathways (Wuchty & Stadler 2003). Closeness centrality, a centrality that ranks vertices depending on the sum of their shortest paths to all other vertices is another method for identifying central metabolites in metabolic networks (Ma & Zeng 2003). Wuchty and Stadler applied different types of centralities to metabolic, protein-protein-interaction and domain-sequence networks to identify central network elements (Wuchty & Stadler 2003).

There are many different centrality measures (Bonacich 1972, Freeman 1977, Freeman 1979, Freeman, Borgatti & White 1991, Koschützki, Lehmann, Peeters, Richter, Tenfelde-Podehl & Zlotowski 2005, Newman 2003, Wasserman & Faust 1994) which can be used to analyse networks, and choosing the “right” measure for a specific problem is usually a difficult task. For example, in biological networks correlations between specific centrality measures and functionally important properties have been shown for some networks (Jeong et al. 2001, Wuchty & Stadler 2003). However, it has also been shown that the degree of a vertex alone is not sufficient to distinguish lethal proteins from viable ones (Wuchty 2002); that in protein networks there is no relation between network connectivity and robustness against amino-acid substitutions (Hahn, Conant & Wagner 2002); that different centrality measures for metabolic networks identify different sets of central metabolites (Ma & Zeng 2003); and

*e-mail: Tim.Dwyer@infotech.monash.edu.au

†e-mail: shhong@it.usyd.edu.au

‡e-mail: koschuet@ipk-gatersleben.de

§e-mail: schreibe@ipk-gatersleben.de

¶e-mail: KaiKevin.Xu@nicta.com.au

Copyright ©2006, Australian Computer Society, Inc. This paper appeared at Asia-Pacific Symposium on Information Visualization (APVIS 2006), Tokyo, Japan, February 2006. Conferences in Research and Practice in Information Technology, Vol. 60. K. Misue, K. Sugiyama and J. Tanaka, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

This paper is a result of the NICTA VALACON (Visualisation and Analysis of Large and Complex Networks) Workshop on Biological Networks organised by Seok-Hee Hong on November 2004 in Sydney, Australia. This work has been partially supported by the German Ministry of Education and Research (BMBF) under grant 0312706A.

that for biological network analysis several centrality measures have to be considered (Koschützki & Schreiber 2004, Wuchty & Stadler 2003). Furthermore, usually there is not enough data about the functional properties of network elements available to determine “good” centrality measures and centrality analysis is often used to simply explore a biological network to discover potentially important parts. A common approach to analysing such networks is therefore to compare different centralities within one network, build a hypothesis concerning discovered interesting “central” elements and test this hypothesis by experimental approaches.

In Table 1 and Figures 1 and 2 typical methods for the comparison of different centrality measures are shown using a protein-protein interaction (PPI) network as an example: correlation (Kendall’s correlation coefficient τ), scatterplot and parallel coordinates (Inselberg & Dimsdale 1990). This PPI network is based on *Mus musculus* (mouse) data from the DIP-database (Salwinski, Miller, Smith, Pettit, Bowie & Eisenberg 2004), release 26-01-2005 and shows the largest connected component of the network.

| | C_e eccentricity | C_c closeness | C_λ eigen- vector | C_d degree | C_r rwbet- weenness |
|-------------|-----------------------|--------------------|---------------------------------|-----------------|-----------------------------|
| C_e | 1.00000 | 0.85319 | 0.52707 | 0.35398 | 0.36017 |
| C_c | 0.85319 | 1.00000 | 0.58717 | 0.42786 | 0.46054 |
| C_λ | 0.52707 | 0.58717 | 1.00000 | 0.44396 | 0.43527 |
| C_d | 0.35398 | 0.42786 | 0.44396 | 1.00000 | 0.74693 |
| C_r | 0.36017 | 0.46054 | 0.43527 | 0.74693 | 1.00000 |

Table 1: Kendall’s correlation coefficient τ for the centrality ranks of the *M. musculus* PPI-network

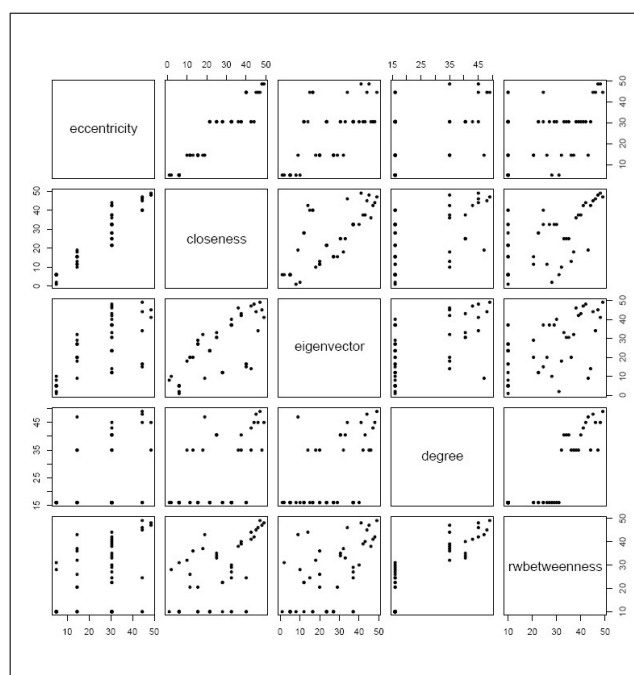


Figure 1: Scatterplot matrix of the centrality rank of the *Mus musculus* PPI-network. To the right and top of each square are vertices with high centrality. The different centrality measures used in this example are explained in Section 2

These commonly used methods for comparing centralities in biological networks have several disadvantages. Correlation coefficients only describe how well different centrality measures correlate without showing which parts of the networks are similar or dif-

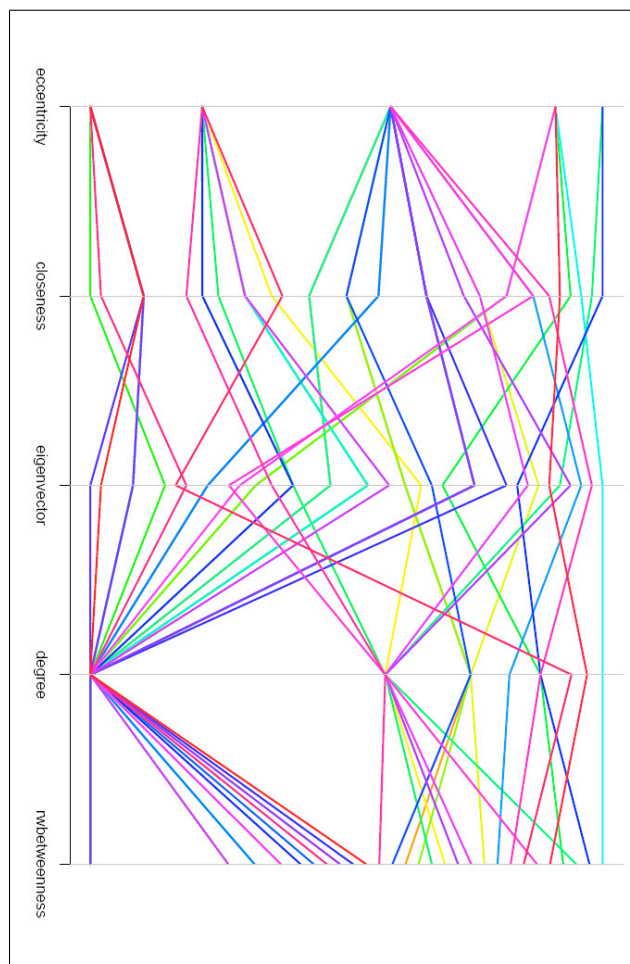


Figure 2: Parallel coordinates of the centrality rank for the same PPI network analysed in Figure 1

ferent. Scatterplots and parallel coordinates allow the comparison of the centrality of single vertices and show a general trend in correlation, but it is difficult to investigate larger sets of vertices even if colours are used. Furthermore, such methods do not support the investigation of the network structure to explore, for example, the distribution of centrality values or to find clusters of vertices with high centrality.

There are several approaches supporting such a structure-centred analysis. Centrality values of vertices can be visualised using the size of the vertices or by constraining their positions to fixed distances from the centre or the bottom of the drawing, see for example the Visone (Brandes & Wagner 2004) and Pajek systems (Batagelj & Mrvar 2004). To compare different centrality measures within one network such methods could be combined, e.g. by representing one centrality with different vertex sizes and another centrality with the positions of the vertices constrained to centrality-dependent distances from the centre. However, this solution is restricted to the comparison of only a few centralities. Further on, it does not show very well how the centrality value of a vertex changes from one centrality measure to the next.

The exploration and comparison of different centralities within one network is important, however, existing methods inefficiently support such tasks. This paper deals with three methods to assist scientists in the exploration and comparison of centrality measures within a single network. These three methods are based on a common underlying idea: for each centrality measure the graph is copied and drawn in a separate 2D plane such that the vertex positions depend on their centrality values. These planes are

then stacked into the third dimension. If the centrality value of a vertex does not change its position in the next plane with respect to the position in the previous plane it is stable. Changes of centrality values can be additionally emphasised by edges between the planes. The result is that different centrality measures can be easily compared.

This general idea of stacking 2D network information into the third dimension, also called $2\frac{1}{2}$ D visualisation, has been demonstrated to be useful in several applications (Brandes & Corman 2002, Brandes, Dwyer & Schreiber 2004, Dwyer & Eades 2002, Koike 1993), but to the best of our knowledge, $2\frac{1}{2}$ D techniques have not previously been applied to the study of network centrality. The details of how centrality is mapped to vertex position are different in our three methods. For *3D parallel coordinates* vertices are placed on vertical lines; for *orbit-based comparison* vertices are placed on concentric circles and for *hierarchy-based comparison* vertices are placed on horizontal lines. The second and third solutions make it particularly easy to track changing vertex-centrality values in the context of the underlying network structure.

This paper is organised as follows: in Section 2 we define the graph model on which we operate and introduce five centrality measures which are used as examples. Section 3 discusses the visualisation methods. These methods are applied to a biological and a social network in Section 4. Finally, Section 5 contains general discussion.

2 Definitions

In this section we define the graph model, introduce five centrality measures and discuss the comparison of such centrality measures.

2.1 Graphs

An undirected graph G consists of a finite set V of vertices and a set E of two-element subsets of V called edges. An edge $e = \{u, v\} \in E$ connects two vertices u and v . The vertices u and v are said to be *incident* with the edge e and *adjacent* to each other. The number of vertices and edges of G is given by $n = |V|$ and $m = |E|$, respectively. The set of all vertices which are adjacent to u is called the *neighbourhood* of u . A graph is called *loop-free* if no edge connects a vertex to itself. An *adjacency matrix* A of a graph is an $(n \times n)$ matrix, where $a_{ij} = 1$ if and only if $\{i, j\} \in E$ and $a_{ij} = 0$ otherwise. The adjacency matrix of any undirected graph is symmetric.

The *degree* $\deg(v)$ of a vertex v is the number of its incident edges. Let (e_1, \dots, e_k) be a sequence of edges in a graph G . This sequence is called a *walk* if there are vertices v_0, \dots, v_k such that $e_i = \{v_{i-1}, v_i\}$ for $i = 1, \dots, k$. If the edges e_i are pairwise distinct and the vertices v_i are pairwise distinct the walk is called a *path*. The *length* of a walk or path is given by the number of involved edges $k = |(e_1, \dots, e_k)|$. A *shortest path* between two vertices u, v is a path with minimal length. The *distance* $\text{dist}(u, v)$ between two vertices u, v is the length of a shortest path between them. Two vertices u, v of a graph are called *connected* if there exists a walk from vertex u to vertex v . If any pair of different vertices of the graph is connected, the graph is called *connected*. A *random walk* between u and v is a walk which starts at vertex u , chooses uniformly at random one of the incident edges of the current vertex until it finally reaches the target v .

In the remainder of this paper we consider only undirected, loop-free, connected, non-trivial (at least

two vertices and one edge) graphs. This restriction is required to be able to use all centrality measures defined in the following paragraph. Note that several centralities can easily be expanded to cover directed or unconnected graphs. Even an extension for weighted edges is possible.

2.2 Centralities

A centrality is a function \mathcal{C} which assigns every vertex $v \in V$ of a given graph G a value $\mathcal{C}(v) \in \mathbb{R}$. As we are interested in the ranking of the vertices of G we choose the convention that a vertex u is more important than another vertex v iff $\mathcal{C}(u) > \mathcal{C}(v)$.

Degree An obvious order of the vertices of a graph can be established by sorting them according to their degree. The corresponding centrality measure *degree-centrality* (\mathcal{C}_d) is defined as $\mathcal{C}_d(v) := \deg(v)$. Freeman gives a long list of references to the usage of degree-centrality in social network analysis (Freeman 1979). For biological network analysis, for example, proteins with high degree-centrality have been shown to be important in PPI-networks (Jeong et al. 2001).

Eccentricity This and the two subsequent centrality definitions operate on the concepts of paths within the given graph. A simple definition uses the distance between vertices. The *eccentricity* ecc of a vertex u is defined as $\text{ecc}(u) := \max_{v \in V} \text{dist}(u, v)$ and the *eccentricity-centrality* (\mathcal{C}_e) as $\mathcal{C}_e(u) := \frac{1}{\text{ecc}(u)}$. The reciprocal of $\text{ecc}(u)$ is used to ensure that more central vertices have a higher value of \mathcal{C}_e , since such central vertices are the ones with the smallest eccentricity value. An application of eccentricity within the biological context is shown by Wuchty and Stadler (Wuchty & Stadler 2003).

Closeness In contrast to eccentricity, closeness-centrality uses not the maximum distance between the vertex of interest and all other vertices but the sum of the distances of this vertex and all other vertices. The *closeness-centrality* $\mathcal{C}_c(u)$ is defined as $\mathcal{C}_c(u) := \frac{1}{\text{sumdist}(u)}$ with $\text{sumdist}(u) = \sum_{v \in V} \text{dist}(u, v)$. Closeness-based centrality measures are used in social (Freeman 1979) and biological (Wuchty & Stadler 2003) network analysis.

Random Walk Betweenness Within networks a communication between two vertices u, v may be visible to a third vertex w if this vertex lies in the path of the communication between u and v . To measure the centrality of a vertex the ability to observe communication is a feasible approach. Different methods to model communication are conceivable, e.g., over shortest paths, paths with maximum flow and random walks. All of these are potential models for *betweenness* (Freeman 1977, Freeman et al. 1991, Newman 2003). Newman's random walk approach models information transmission and therefore matches problems often modelled in biological networks. For the *random-walk betweenness centrality* (\mathcal{C}_r) the centrality of a vertex w is equal to the number of times that a random walk from u to v goes through w , averaged over all u and v .

Bonacich's Eigenvector Centrality A different approach to order the vertices of a graph was suggested by Bonacich (Bonacich 1972). It is based on the assumption that the value of a single

vertex is determined by the values of the neighbouring vertices. In contrast to the previous measures not only the position of a vertex within the graph is considered but also the centrality values of its neighbours. Bonacich suggested the following definition: $C_\lambda(u) := \sum_{v \in N(u)} C_\lambda(v)$. Considering the adjacency matrix representation of the graph this is equivalent to $C_\lambda(v_i) := \sum_{j=1}^n a_{ij} C_\lambda(v_j)$. This leads directly to the well known problem of eigenvector computation $\lambda S = AS$ and the eigenvector of the largest eigenvalue is the *eigenvector-centrality* ($C_\lambda := S$) (Bonacich 1972).

2.3 Correlation and Ordering

Given a graph and a set of centrality measures our method for the exploration and comparison of the different centralities is based on the stacking of copies of the graph, one for each centrality, into the third dimension. An important task is to compute an appropriate ordering of these copies. We want to order them such that those which are similar with respect to the correlation, i.e., which have a correlation close to 1.0, are close to each other. This is similar to Keim's one-dimensional ordering of dimensions (Keim 2000). The correlation $\tau(C_i, C_j)$ between the centralities C_i and C_j is our similarity measure with large values (close to 1.0) meaning high similarity, whereas low or negative values mean dissimilarity. In detail we use Kendall's correlation coefficient τ_b as this correlation coefficient is known to behave better than other coefficients (e.g. Spearman's ρ) in the case of ties, e.g., values which are equal for several objects. The definition and the details of the computation are beyond the scope of this paper and can be found in the literature (Lienert 1973, Press, Teukolsky, Vetterling & Flannery 1992).

Let C_1, \dots, C_k be the centralities under consideration. The optimal ordering of the centralities and therefore the corresponding copies of the graph is a permutation $\{\pi(1), \dots, \pi(k)\}$ of the centralities such that $\sum_{i=1}^{k-1} \tau(C_{\pi(i)}, C_{\pi(i+1)})$ is maximal. Note that the computation of an optimal ordering is NP-hard. In our example with five centralities we can compute the optimal ordering, for larger sets of different centralities several ordering heuristics exists.

3 Visualisation

In this section we discuss briefly the overarching visualisation concept and present our three visualisation approaches.

3.1 General Visualisation Method

The common idea behind the different visualisation approaches is that there is a copy of the graph for each centrality. Each copy is drawn in a separate 2D plane such that central vertices are easily detectable. The planes are then stacked into the third dimension in an optimal ordering and an arrangement of vertices within planes is computed with the goal of supporting comparison of centralities.

3.2 3D Parallel Coordinates-based Comparison

Parallel coordinates techniques have been used successfully for the visualisation of multi-dimensional or multivariate data (Inselberg & Dimsdale 1990). However, from our experiments with biological and social networks we observed that vertices in the network can

frequently share the same centrality value. Note that in standard two-dimensional parallel coordinates as in Figures 2 it is rather difficult to show each vertex separately, in particular where two or more vertices have the same centrality value.

This motivates the use of the additional dimension to display all the vertices which have the same centrality value in a horizontal line simultaneously. The main idea of our 3D parallel coordinates is very simple. Each two dimensional plane contains the information for a particular centrality and several centralities are stacked in the third dimension. For each centrality we use horizontal lines to place vertices with the same centrality values. See Figures 4 for an example.

More specifically, for each centrality measure we use a vertical line and several horizontal lines within the plane. That is, the vertical line represents the range of centrality values from the lowest centrality value to the highest. For each centrality measure all vertices belonging to exactly the same centrality value are placed on a horizontal line to separate the vertices. The ordering of planes is decided using the method described in Section 2.3. Note that two vertices may overlap if their centrality values are very close; however this problem can be solved by adjusting the size of the vertices.

Next, we add inter-plane edges between two adjacent planes, where each plane represents one centrality measure, such that the same vertices of the graph are connected by inter-plane edges. This helps to trace vertices between two different planes. Note that to reduce visual complexity it is possible to show edges only when its centrality difference is greater than a given threshold.

Finally, we need to decide the ordering of the vertices on each horizontal line, i.e. the ordering of vertices which have the same centrality value. We want an ordering which minimises the total edge length of inter-plane edges between the two planes, as the use of long edges will increase the visual complexity and thus decrease ease of comparison. Similarly, the ordering should avoid edge crossing. This problem can be solved easily considering the position of the same vertex in the previous plane. For tie-breaking, we use the difference in height.

Overall, the time complexity of comparison based on the 3D parallel coordinates is linear if the centrality values for each centrality measure and the optimal ordering of the different centrality measures are given. Thus, this method may scale well for large networks.

3.3 Orbit-based Comparison

The objective of our graph drawing algorithm for orbit-based comparison is to find coordinates for each vertex of the graphs G_1, \dots, G_k such that:

1. all vertices of graph G_i have the same z -coordinate (plane constraint);
2. the order of z -coordinates corresponds with the computed order of the graphs, see Section 2.3;
3. all vertices of graph G_i are constrained to lie on concentric circles (orbits) depending on the centrality value of the vertex. That is, each vertex v is assigned an *orbital constraint* with radius $r = f(C(u))$ where f is typically a partitioning such that there are only a small number of distinct radii;
4. the centre of the concentric circles for all graphs has the same x - and y -coordinate;

5. all vertices of graph G_i have x - and y -coordinates such that the distance of a vertex to its neighbours is as close to the optimal distance d as possible; and
6. each vertex of G_i should be as close as possible to the same vertex of the adjacent graphs.

A formulation for the “stress” of a graph layout is (Borg & Groenen 1997, Gansner, Koren & North 2004):

$$\sum_{i < j} w_{ij} (\| X_i - X_j \| - d_{ij})^2 \quad (1)$$

where X_i and X_j are the positions of the i^{th} and j^{th} vertices of a graph G , d_{ij} is the ideal distance between these vertices (typically a function of the graph theoretic distance between them) and w_{ij} is a normalisation constant (for example: d_{ij}^{-2}). A “good” layout is said to be one that minimises this stress function.

Since the plane constraints are constant and independent of the layout within planes, our layout scheme with plane and orbital constraints can be reduced to a 2D layout problem. That is, we seek to find an arrangement for the union graph $G = \bigcup_{i=1}^k G_i$ and we augment the edges of G with a set of inter-plane edges $\{\{u, v\} \mid \text{where } u \text{ and } v \text{ are the equivalent vertices in adjacent levels } G_i \text{ and } G_{i+1}\}$. The fifth objective in the list above is met by setting the ideal length $d = 0$ for such inter-plane edges. By restating this “stress” function in polar coordinates we can precisely define the orbital constraints of this new layout problem:

$$\sum_{i < j} w_{ij} (\| r_i(\cos \theta_i, \sin \theta_i) - r_j(\cos \theta_j, \sin \theta_j) \| - d_{ij})^2 \quad (2)$$

where r_i and r_j are constants corresponding to the radii of the orbit constraints for vertices i and j respectively. While Kamada and Kawai (Kamada & Kawai 1989) showed that an approximate solution for (1) could be found by iteratively fixing all but one vertex and solving the resulting convex quadratic form, and Gansner *et al.* (Gansner et al. 2004) were able to bound the global function with a quadratic form using functional majorisation, the cyclical nature (and hence non-convexity) of (2) means that no such simplification is readily apparent. For this reason Brandes *et al.* (Brandes, Kenis & Wagner 2003) chose a simulated annealing solution for a similar radial layout problem. Instead, we have had reasonable success adapting a naïve Fruchterman and Reingold (Fruchterman & Reingold 1991) force-directed method.

A standard force-directed method works by computing a “force” vector \vec{f} for each vertex u from the sum of attractive forces between u and all v where there exists an edge $\{u, v\}$ and repulsive forces between u and all other vertices in G . Each vertex is then moved by a small amount in the direction of this force, for example a vertex at position p is moved to $p' = p + c\vec{f}$ where c is a constant. The process repeats iteratively until the total length of all force vectors for an interaction falls below some threshold. We augment the standard force-directed layout method with orbital constraints by moving vertices only by the projection of \vec{f} on the orbital constraint arc. That is, to satisfy the orbital constraint centred about o with radius r the above vertex is placed at $o + r|\vec{op}'|$, see Figure 3.

To aid comparison of centralities it is more important that inter-plane edge lengths are minimised than the length of edges within each plane. Thus,

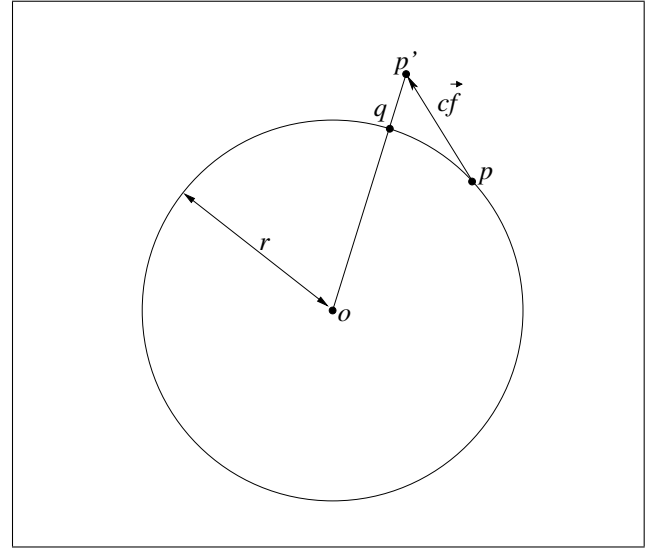


Figure 3: Calculating the new position q of a vertex at position p due to force f subject to an orbital constraint of radius r

we run the force-directed placement to completion in several stages. First, inter-plane attractive forces are activated; then forces within each plane are added, one plane at a time. This tends to avoid local minima of (2) with long inter-plane edges. Also, repulsive forces between vertices on different planar levels are not considered.

3.4 Hierarchy-based Comparison

In this section we describe a centrality comparison method based on a drawing algorithm which uses a hierarchy to display centrality values. More specifically, we use horizontal lines to represent centrality values. As with the previous methods, a drawing in each plane displays different centrality measures, and the ordering of the planes is decided based on the method described in Section 2.3.

To draw each graph in a plane by displaying centrality as hierarchy we first divide the vertices into layers depending on their centrality value. That is the vertices in the upper layers have higher centrality values than the vertices in the lower layers. Then we place each vertex in the layer on a horizontal line. Note that this method is different to the well-known *Sugiyama* method (Battista, Eades, Tamassia & Tollis 1999, Sugiyama, Tagawa & Toda 1981) for drawing hierarchical graphs as a layered drawing, in the sense that the layering is not based on a traversal of a directed graph and that there may exist edges between vertices in the same layer.

Inside each layer, we can define a few different ordering methods to order the vertices. For example, we can choose an ordering of vertices such that the vertices are ordered from the left to the right based on their centrality value in a decreasing way. Further, it may be possible to choose a different ordering of vertices in the layer with different optimisation criteria. For example, one may choose an ordering which minimises the length of edges between each plane. This problem can be easily solved as in the 3D parallel coordinates method. Alternately, one may choose an ordering which minimises the edge crossings in each plane. This problem is relatively well-studied in graph drawing literature, it is NP-hard even if there are only two layers (Eades & Whitesides 1994). However, there are several fast heuristics such as the *Median* and the *Barycenter* method available (Battista

et al. 1999).

To reduce edge crossings between adjacent layers within one plane we use a modification of the Barycenter method. More specifically, the position of a vertex is averaged over the positions of all the neighbours on upper layers, not necessarily just one level above. This considers edges which span more than two layers. Alternatively, these edges can be handled by introducing dummy vertices, as in the traditional *Sugiyama* method.

Finally, to reduce the total number of edge crossings in each plane, we perform a *layer-by-layer sweep* heuristic (Battista et al. 1999), that is, the algorithm sweeps from the top layer to the bottom layer, and then sweeps from the bottom layer to the top layer. We had reasonable success when this process was repeated ten times. Note that we can further combine these two different optimisation criteria. A simple heuristic is to use a variation of the Barycenter method, now also considering the position of the neighbour in the previous plane.

This hierarchy-based comparison method can be implemented in linear time, as the Barycenter heuristic can be implemented to run in linear time. Thus it can scale better than the orbit-based comparison method for large networks. Examples of this method are shown in Figures 6 and 7.

Again, as in the previous methods, we use filtering for displaying inter-plane edges to reduce cognitive load. That is we display the inter-plane edges only when the difference between the two centrality values of the same vertices differ more than some given threshold.

4 Implementation and Results

We implemented the presented methods in GEOMI (Geometry for Maximum Insight), a visual analysis tool for the visualisation and analysis of large and complex networks such as social networks, biological networks, scale-free networks and dynamic networks (Ahmed, Dwyer, Forster, Fu, Ho, Hong, Koschützki, Murray, Nikolov, Taib, Tarassov & Xu 2005). GEOMI is based on WilmaScope (Dwyer & Eckersley 2004). We used it for the analysis of the *Mus musculus* PPI network introduced in Section 1 and a social network based on Padgett's Florentine families marital relation data (Wasserman & Faust 1994). The PPI network from the DIP-database consists of 49 vertices and 54 edges. The social network has 16 vertices and 40 edges.

Figures 4-7 show some layouts generated using the three methods. In Figure 4 the biological network is presented with 3D parallel coordinates as described in Section 3.2. Each vertex of the network represents a protein and each edge a protein-protein-interaction. Each vertical column in the layout represents a specific centrality measure. If vertices have the same centrality value within one centrality measure they are placed on a horizontal line. The similarity to common 2D parallel coordinates makes this representation easy to understand. If viewed from the front the picture would look like a conventional 2D parallel coordinate visualisation as shown in Figure 2.

Even though it is easier to compare single vertices in the 3D representation than in common 2D parallel coordinates, there is still the disadvantage that centrality values and network structure cannot be explored within one representation. The orbit-based and the hierarchy-based comparisons overcome this disadvantage. Figure 5 shows the different centralities of the biological network with the orbit-based method described in Section 3.3. Each plane shows the network with the vertices placed on different or-

bits depending on their centrality values. If the orbit of a vertex changes between adjacent planes an edge is shown. Additionally in the small window the plane highlighted by the water-layer is displayed in a two dimensional representation. By moving the water-layer up and down it is easy to navigate through the different centralities and to see how the centrality values of a vertex change (move in and out) or are stable (a stable vertex has nearly the same x - and y -coordinates in each layer). In Figure 6 the biological network is presented with the hierarchy-based comparison described in Section 3.4.

As an example of social-network analysis we used Padgett's Florentine families marital relation data (Wasserman & Faust 1994). Each vertex in the network represents a family in 15th century Florence, Italy and each edge represents marital relations between the families. The 16 families are chosen for analysis from a larger collection of 116 leading Florentine families due to their historical prominence, such as the Medicis and Strozzi. Figure 7 shows a visual comparison of centralities for Padgett's Florentine families marital relation data. From the drawing, it is easy to see that Medici always belongs to the top rank in any centrality measure. Clearly, one can see the marital relations between the families in one plane and then compare the difference between different centrality measure.

5 Conclusion

We have demonstrated three different methods for the analysis and visualisation of centrality measures of a network. All methods represent the information in three dimensions and two of them allow the simultaneous visualisation of both the network structure and the ranking of vertices based on the centrality measures. All three visualisation methods support the exploration of different centrality measures for the same network. In this paper we presented the comparison of centrality measures, however, the visualisation methods could be also used to represent other multidimensional numerical data within an underlying network.

Our methods work well with networks of up to a hundred vertices. An important question is the scalability of our approach to larger networks with thousands of vertices. Larger networks result in very dense visualisations which are difficult to read and understand. To deal with such networks abstraction methods have to be applied before any of the methods described may show a useful result. The definition of such abstraction methods should be a focus for further work. Another issue is a formal user study to compare the different proposed methods with each other and with well-known 2D techniques.

References

- Ahmed, A., Dwyer, T., Forster, M., Fu, X., Ho, J., Hong, S.-H., Koschützki, D., Murray, C., Nikolov, N. S., Taib, R., Tarassov, A. & Xu, K. (2005), GEOMI: GEOMetry for Maximum Insight, in 'Proceedings of the 13th International Symposium on Graph Drawing (GD'05)'. to appear.
- Batagelj, V. & Mrvar, A. (2004), *Pajek - Analysis and Visualization of Large Networks*, in Jünger & Mutzel (2004), pp. 77-103.
- Battista, G. D., Eades, P., Tamassia, R. & Tollis, I. G. (1999), *Graph Drawing: Algorithms for the Visualization of Graphs*, Prentice-Hall.

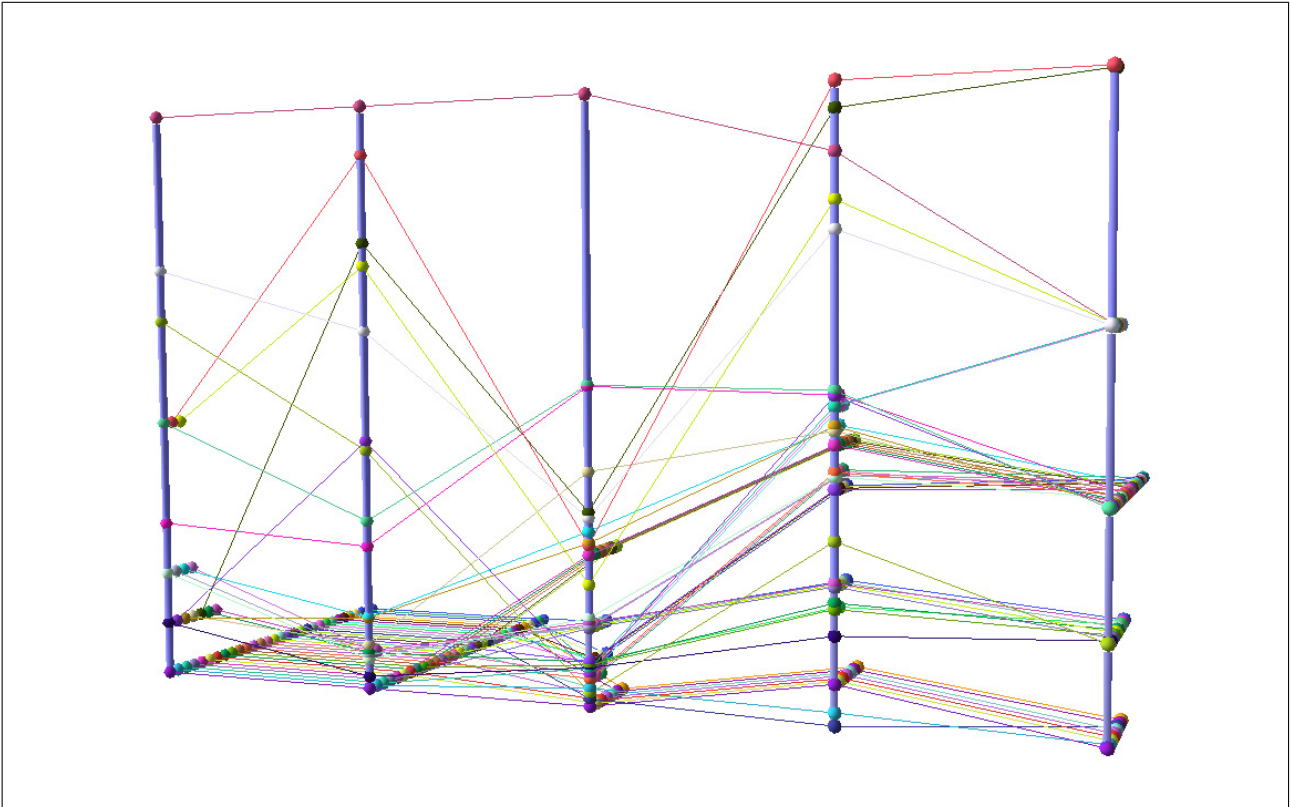


Figure 4: The network described in Section 1 in 3D parallel coordinates, a method described in Section 3.2. In particular, 3D parallel coordinates can display the exact centrality value for each vertex

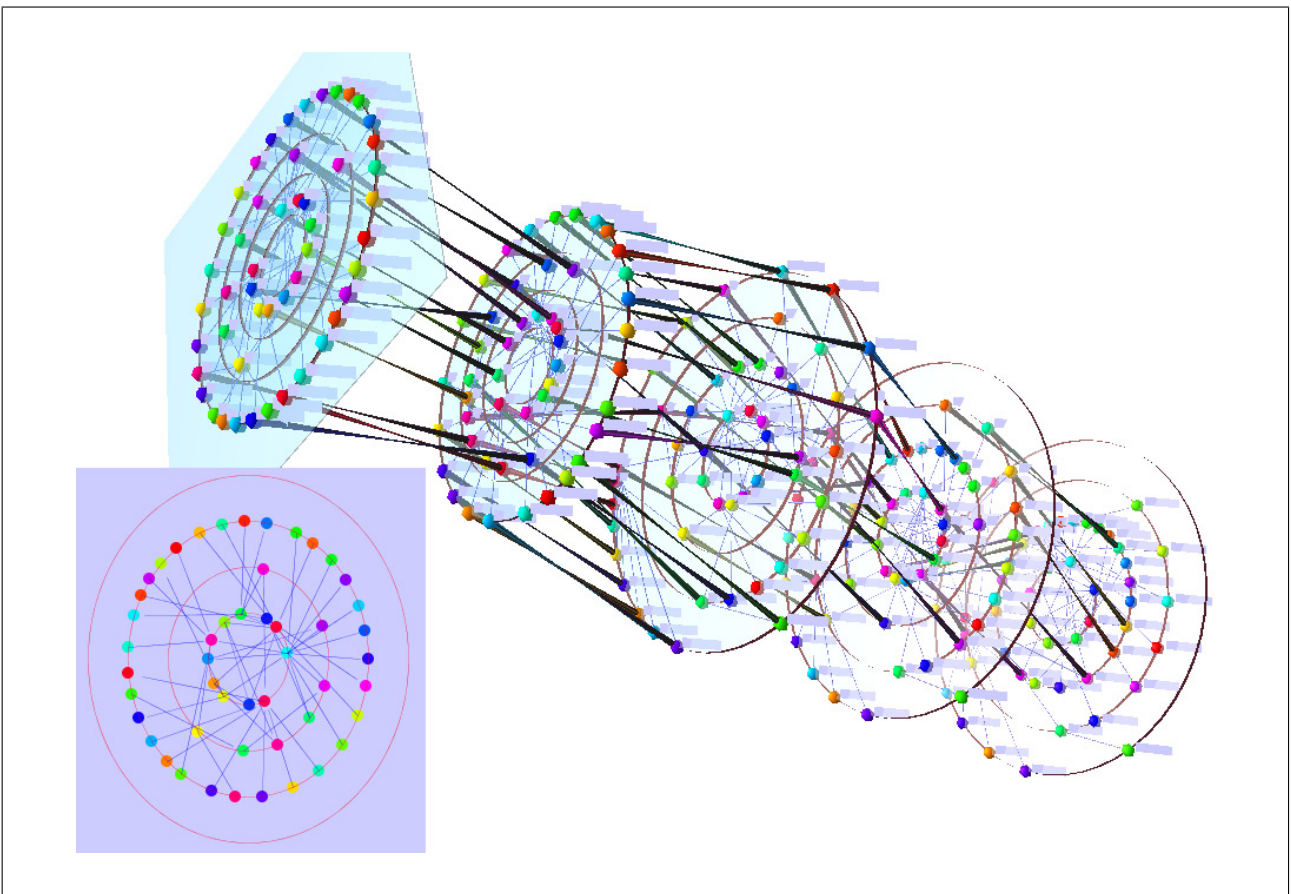


Figure 5: The same PPI network as shown in Figure 4 with the same centrality measures represented with the orbit-based comparison method. In the small window (left) the plane highlighted by the water-layer is displayed in a two dimensional representation

- Bonachich, P. (1972), 'Factoring and weighting approaches to status scores and clique identification', *Journal of Mathematical Sociology* **2**, 113–120.
- Borg, I. & Groenen, P. (1997), *Modern Multidimensional Scaling: Theory and Applications*, Springer.
- Brandes, U. & Corman, S. R. (2002), Visual unrolling of network evolution and the analysis of dynamic discourse, in 'Proceedings of the IEEE Symposium on Information Visualization 2002 (InfoVis '02)', pp. 145–151.
- Brandes, U., Dwyer, T. & Schreiber, F. (2004), 'Visual understanding of metabolic pathways across organisms using layout in two and a half dimensions', *Journal of Integrative Bioinformatics* **1**, e0002.
- Brandes, U., Kenis, P. & Wagner, D. (2003), 'Communicating centrality in policy network drawings', *IEEE Transactions on Visualization and Computer Graphics* **9**(2), 241–253.
- Brandes, U. & Wagner, D. (2004), *visone - Analysis and Visualization of Social Networks*, in Jünger & Mutzel (2004), pp. 321–340.
- Dwyer, T. & Eades, P. (2002), Visualising a fund manager flow graph with columns and worms, in 'Proceedings of the 6th International Conference on Information Visualisation (IV '02)', IEEE Computer Society Press, pp. 147–152.
- Dwyer, T. & Eckersley, P. (2004), *WilmaScope - A 3D Graph Visualization System*, in Jünger & Mutzel (2004), pp. 55–75.
- Eades, P. & Whitesides, S. (1994), 'Drawing graphs in two layers', *Theoretical Computer Science* **131**(2), 361–374.
- Fell, D. A. & Wagner, A. (2000), 'The small world of metabolism', *Nature Biotechnology* **18**, 1121–1122.
- Freeman, L. C. (1977), 'A Set of Measures of Centrality Based on Betweenness', *Sociometry* **40**(6), 35–41.
- Freeman, L. C. (1979), 'Centrality in social networks: Conceptual clarification', *Social Networks* **1**, 215–239.
- Freeman, L. C., Borgatti, S. P. & White, D. R. (1991), 'Centrality in valued graphs: a measure of betweenness based on network flow', *Social Networks* **13**, 141–154.
- Fruchterman, T. M. J. & Reingold, E. M. (1991), 'Graph drawing by force-directed placement', *Software - Practice and Experience* **21**(11), 1129–1164.
- Gansner, E. R., Koren, Y. & North, S. C. (2004), Graph drawing by stress majorization, in 'Proceedings of the 12th International Symposium on Graph Drawing (GD'04)', Vol. 3383 of *Lecture Notes in Computer Science*, Springer, pp. 239–250.
- Hahn, M. W., Conant, G. C. & Wagner, A. (2002), Molecular evolution in large genetic networks: Connectivity does not equal importance, Technical Report 02-08-039, Santa Fe Institute.
- Inselberg, A. & Dimsdale, B. (1990), Parallel coordinates: A tool for visualizing multidimensional geometry, in 'Proceedings of the 1st IEEE Conference on Visualization (Vis '90)', pp. 361–378.
- Jeong, H., Mason, S. P., Barabási, A.-L. & Oltvai, Z. N. (2001), 'Lethality and centrality in protein networks', *Nature* **411**, 41–42.
- Jünger, M. & Mutzel, P., eds (2004), *Graph Drawing Software*, Springer.
- Kamada, T. & Kawai, S. (1989), 'An algorithm for drawing general undirected graphs', *Information Processing Letters* **31**(1), 7–15.
- Keim, D. A. (2000), 'Designing pixel-oriented visualization techniques: Theory and applications', *IEEE Transactions on Visualization and Computer Graphics* **6**(1), 59–78.
- Koike, H. (1993), 'The role of another spatial dimension in software visualization', *ACM Transactions on Information Systems* **11**(3), 266–286.
- Koschützki, D., Lehmann, K. A., Peeters, L., Richter, S., Tenfelde-Podehl, D. & Zlotowski, O. (2005), Centrality indices, in U. Brandes & T. Erlebach, eds, 'Network Analysis', Vol. 3418 of *LNCIS Tutorial*, Springer, pp. 16–61.
- Koschützki, D. & Schreiber, F. (2004), Comparison of centralities for biological networks, in 'Proceedings of the German Conference on Bioinformatics 2004', Vol. P-53 of *Lecture Notes in Informatics*, Springer, pp. 199–206.
- Lienert, G. A. (1973), *Verteilungsfreie Methoden in der Biostatistik*, Verlag Anton Hain.
- Ma, H.-W. & Zeng, A.-P. (2003), 'The connectivity structure, giant strong component and centrality of metabolic networks', *Bioinformatics* **19**(11), 1423–1430.
- Newman, M. E. J. (2003), A measure of betweenness centrality based on random walks. arXiv cond-mat/0309045.
- Press, W. H., Teukolsky, S. A., Vetterling, W. T. & Flannery, B. P. (1992), *Numerical Recipes in C*, Cambridge University Press.
- Salwinski, L., Miller, C. S., Smith, A. J., Pettit, F. K., Bowie, J. U. & Eisenberg, D. (2004), 'The database of interacting proteins: 2004 update', *Nucleic Acids Res* **32**(1), 449–451.
- Sugiyama, K., Tagawa, S. & Toda, M. (1981), 'Methods for visual understanding of hierarchical system structures', *IEEE Transactions on Systems, Man and Cybernetics* **SMC-11**(2), 109–125.
- Wasserman, S. & Faust, K. (1994), *Social Network Analysis: Methods and Applications*, Cambridge Univ. Press.
- Wuchty, S. (2002), 'Interaction and domain networks of yeast', *Proteomics* **2**, 1715–1723.
- Wuchty, S. & Stadler, P. F. (2003), 'Centers of complex networks', *Journal of Theoretical Biology* **223**, 45–53.

How People Read Sociograms: A Questionnaire Study

Weidong Huang, Seok-Hee Hong, and Peter Eades

IMAGEN Program, National ICT Australia Ltd.
School of Information Technologies, University of Sydney, Australia
Email: {weidong.huang, seokhee.hong, peter.eades}@nicta.com.au

Abstract

Visualizing social network data into sociograms plays an important role in communicating information about network characteristics. Previous studies have shown that human perceptions of network features can be affected by the layout of a sociogram [McGrath et al. 1996, 1997]. An empirical user study has been conducted to investigate effectiveness of five different network visualization conventions and impact of edge crossings on sociogram perceptions, using both quantitative performance and preference measures and qualitative questionnaire study. This paper reports results and findings of the questionnaire study. We relate qualitative questionnaire results with quantitative findings and discuss their implications for sociogram design. We found that subjects had a strong preference of placing nodes on the top or in the center to highlight importance, and clustering nodes in the same group and separating groups to highlight groups. They had tendency to believe that nodes in the center or on the top are more important, and nodes in close proximity belong to the same group. Some preliminary recommendations for sociogram design and hypotheses about human reading behaviors are proposed.

Keywords: sociogram perception, social network, edge crossing, questionnaire.

1 Introduction

A social network [Wasserman et al. 1994] is a collection of *actors* (such as people, organizations or other social entities) and *relationships* among the actors, indicating the way in which they are connected socially (such as friendship, trade or information exchange). *Social network analysis* is a methodological approach to understand the structure of the network. A great deal of research revealed that representing data visually helps people understand data effectively [e.g. Larkin et al. 1987, Scott 2000]. Social networks can be modeled as graphs, and visualized as node-edge diagrams, or *sociograms* (see Figure 1 for an example) where nodes represent actors, and edges represent relationships between them. Recently, with the advance of display media, visualization has been increasingly popular and important in social network analysis.

Sociogram has been used as a major means to assist people in exploring and understanding social networks, and communicating their understandings with

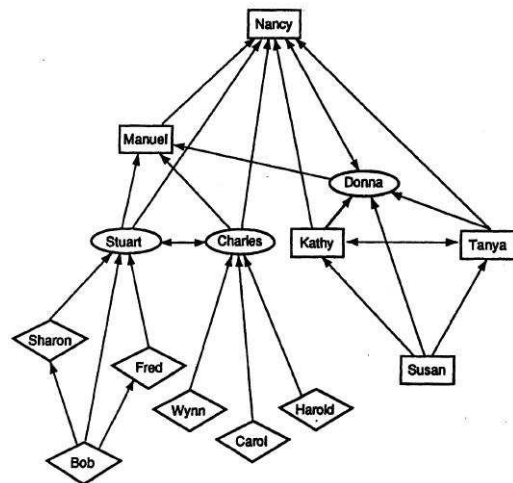


Figure 1: Advice network formed by an auditing team. Courtesy of Krackhardt [1996]. Ellipses represent managers; diamonds represent staff auditors and boxes represent secretaries. A line from Donna to Nancy indicates that Donna seeks advice from Nancy

others. Clearly, a network can be visualized in many different ways. Previous studies [McGrath et al. 1996, 1997] found that the spatial layout of nodes in a sociogram affects humans in perceiving social network characteristics, such as the existence of subgroups and the social positions of actors, although these characteristics are purely determined by the intrinsic network structure. The finding is significant because it indicates that people may perceive differently when a network is drawn differently, although according to Scott [2000, p. 64], when visualizing a network into a sociogram, what matters is relationship patterns, not the physical positioning of nodes.

Many techniques have been proposed and implemented mainly focusing on producing pictures of the network efficiently. Commonly used social network analysis and visualization tools typically use, for example, variants of the spring embedder [Eades 1984], or multidimensional scaling to compute node positions [Brandes et al. 2001]. On the other hand, due to a variety of differences in network structure characteristics, there is no single best layout existing to communicate all the network information effectively [McGrath et al. 1997]. Consequently, a particular spatial arrangement of nodes can only be used for highlighting and conveying only one or two aspects of the network structure.

Sociograms are drawn for people to understand. They are useful only if people who view them can retrieve meaningful information. Therefore, a natural question arises: when choosing a particular visualization technique to communicate our understanding

to others, how sure are we that viewers will understand the information the same as we do? Does the resulting sociogram really convey information which is intended to communicate?

To answer the questions, it is important to understand how people read sociograms in general, and how people interact with different visual factors of a sociogram in particular. Unfortunately, as pointed out by McGrath et al. [2003], our progress in understanding how people perceive sociograms does not well match the advances made in inventing new visualization techniques and tools.

As an attempt to fill in this gap, in a user study investigating communication effectiveness of five sociogram drawing conventions, and impact of edge crossings under each convention, we used not only quantitative performance and preference measures but also qualitative questionnaire study, asking subjects to perform domain-specific tasks, and indicate and explain their preferences for different drawing properties. Subjective data (preference and questionnaire) were gathered based on the methodology of Purchase et al. [2002]. Objective data (response time and response accuracy) were collected through an online system. The aims of the study were:

1. To understand how a particular visualization technique, or a sociogram drawing convention can affect human sociogram perceptions.
2. To investigate the impact of edge crossings, a major aesthetic which has been empirically validated in abstract graphs [Purchase 1997] and UML diagrams [Purchase et al. 2002], but not in sociograms.
3. To propose a set of preliminary recommendations for sociogram design. Currently, sociograms are drawn mainly based on either general aesthetics, personal intuitions, or some commonly used methods. Design guidelines generated on empirical basis should be more beneficial in facilitating human understanding.
4. Our long-time goal is to gain some new insights on how people perceive and interact with visual diagrams.

The results and findings based on quantitative data (task performance and user preference) obtained from the study have been reported elsewhere. For details, see [Huang et al. 2005b]. This paper reports the questionnaire study. We relate qualitative questionnaire results with quantitative findings and discuss their implications for sociogram design. Some preliminary recommendations for sociogram design and hypotheses about human reading behaviors are proposed (see section 4).

The rest of the paper is organized as follows. First in section 2, we introduce five social network visualization conventions and related work, followed by a brief description of our experiment in section 3. In section 4 the results of the study are presented and discussed. Finally we summarize our findings and envision some directions for future work in section 5.

2 Background and Related Work

2.1 Sociogram Perception Studies

McGrath, Blythe and Krackhardt have conducted a series of significant empirical studies in this area, investigating the spatial layout impact on human perceptions at different structure levels. At individual

level, in a study investigating how different drawings of the same network can affect human perceptions of actor's social status in terms of prominence or bridging, and groups present in the network, McGrath et al. [1997] found that both network structure and spatial arrangement of nodes influence viewers' perceptions. Further at group level, a user study conducted by McGrath et al. [1996] suggested the perception of network groupings can be significantly affected by visual clusters appearing in a sociogram. They proposed the "first principle" in designing sociograms, which is "adjacent nodes must be placed near to each other if possible, and Euclidean distance should be correlated with path distance".

More recently, at overall network level, in a large study in which 133 subjects were involved, McGrath et al. [2004] investigated:

1. The difference in effectiveness when a network is drawn to convey a particular aspect of data (spatial hierarchy), and drawn to conform to general graph drawing aesthetics (spatial centrality).
2. The impact of motion and layout on human perceptions of actor's status changes.
3. The effect of different layouts on overall understanding of the underlying network.

They found the following:

1. The difference of layout formats did not have a significant effect on user performance.
2. The introduction of motion with hierarchical layout significantly increased perception accuracy of network changes.
3. Subjects' prior knowledge and experience with a particular layout format could affect their overall network interpretations.

2.2 Readability Studies

In a project investigating the effects of aesthetics and algorithms on graph perception, Purchase et al. conducted a number of empirical studies both in abstract graph and application domains. Among them, Purchase et al. [1995] first validated three graph drawing aesthetics (crossings, bends and symmetry) using paper-based experiments, and revealed that these layout aesthetics do affect human graph understanding. Then in an online study, Purchase [1997] considered the five aesthetic criteria, and demonstrated that minimizing crossings was overwhelmingly beneficial in understanding graph structure; edge crossings was "by far the most important aesthetic".

Also they extended their approaches into the application domain of UML diagrams, and evaluated the effect of individual aesthetics on user preferences [Purchase et al. 2002]. In [Purchase et al. 2002] it is pointed out that there is no guarantee that the results of domain-independent experiments would automatically apply to the domain-specific diagrams.

Further, in an experiment which examined several aesthetics within the same set of computer-generated diagrams, Ware and Purchase [Ware et al. 2003] suggested and demonstrated that good path-continuity has a positive impact, and found that what matters is the crossings on the shortest paths for shortest-path searching tasks. Another notable contribution in this study is the introduction of new evaluation methodology which enables investigators to do cognitive cost assessment and is applicable to similar perception problems.

Instead of aesthetics, Ghoniem et al. [2004] compared readability effects of two visual representations

(node-edge and matrix-based representations of abstract graphs) and graph intrinsic properties (size and density), based on seven generic tasks. They found that readability effects of the two representations varied with size and density of graphs for different types of tasks. In particular, node-edge diagrams are more suitable for small graphs while matrices achieve better readability for large or dense graphs. And more significantly, they proposed a generic task list and methods for the evaluations of the similar type, and recommendations in choosing effective graph representations.

2.3 Questionnaire Approach

In both the sociogram perception and readability studies, either performance or preference was used as a major measurement. Questionnaires were often used for collecting demographic information. In [Purchase et al. 2002], subjects were asked to write an explanation for each of their preferences, which was intended for identifying confounding factors and providing support for the quantitative results. In our experiment, questionnaire study was employed as a complementary approach to the quantitative performance and preference measures. Therefore, the measurements in the whole study included task performance, user preference, and questionnaires mainly about how they read sociograms. By combining these methods, we believe that the qualitative data derived from questionnaires can help in gaining in-depth insights which otherwise is unlikely to obtain by just looking into static and silent quantitative data.

In particular, questionnaire approach was used in the study in order to:

1. Collect information about subjects' study status, sociogram reading and node-edge diagram experience, etc.
2. Identify possible confounding factors.
3. Provide support for findings of quantitative data.
4. Help to find roots of problems which may not be obvious in quantitative data.
5. Gain insights on sociogram reading behaviors.

2.4 Sociogram Drawing Conventions

Various sociogram drawing techniques have been proposed to highlight and communicate one particular piece of information about network structure, and conform to aesthetic criteria to improve readability. Of our particular interest are the following five sociogram drawing conventions. For examples, see Figure 4.

2.4.1 Circular Layout

For representing network data visually, circular layout, in which all nodes are put on a circle, is a common technique in social network analysis. This layout is intended to make the relationship patterns among actors salient [Scott 2000, p. 146].

2.4.2 Radial Layout

Radial layout was firstly proposed by Northway [1940] as a target sociogram (Figure 2). Nodes are laid inside of concentric circles whose radii increase while the status levels of nodes decrease, with the most central nodes in the center of the diagram. The ordering of nodes in the circles is manually determined so that the length of the lines connecting them are

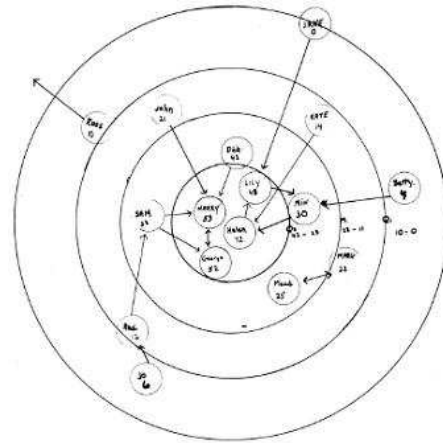


Figure 2: Target sociogram [Northway 1940], where node status is defined by how often the node is chosen

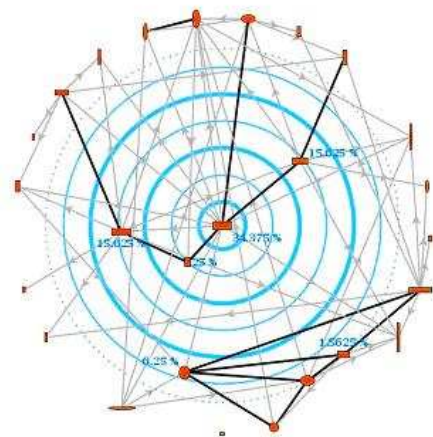


Figure 3: Radial layout of betweenness centrality. Courtesy of Brandes et al. [2003b, Figure 7]

short. Brandes et al. [2003a] suggested a similar approach which places nodes on the circumference of circles in a way that their distances from the center exactly reflect their centrality levels, and the nodes on the circles are arranged by so-called three-phase energy-based layout model to make the final layout readable [Brandes et al. 2003a] (e.g., Figure 3).

2.4.3 Hierarchical Layout

To aid people in exploring network structure and communicating information about actor status, Brandes et al. [2001] proposed a new approach, a prescriptive layout model which directly maps actors' status scores to the nodes' vertical coordinates; the horizontal positioning of nodes is "algorithmically" computed such that the overall readability in the final visual network representation is achieved.

2.4.4 Group Layout

Group layout (for a review, see [Freeman 1999]) is used to display information about groupings. It highlights the group existence by separating different groups and placing nodes in the same group close to each other.

2.4.5 Free Layout

This layout does not have any purpose of highlighting a particular network feature. Instead, sociograms are

drawn based on general aesthetic principles so that the layout is readable. Many automatic graph drawing methods can be used for this purpose [Di Battista et al. 1998, Kaufmann et al. 2001].

3 Experiment

The detailed experimental setting was described in [Huang et al. 2005b]. For completeness, we briefly summarize the experiment here.

The experiment employed a within-subjects design. A real world advice network formed by a small auditing team, which was described by Krackhardt [1996], was used in the study (Figure 1). To investigate drawing convention effects, the network was drawn using the five conventions, respectively; and to test edge crossings impact, for each convention, we drew two sociograms, one with minimum crossings, and the other with many crossings. Finally 2 (minimum crossings vs. many crossings) $\times 5$ (drawing conventions) = 10 sociograms were obtained (see Figure 4).

It is important to note that, by investigating *communication* effectiveness, we mean that we already have some knowledge about the network beforehand, e.g., individual's social status (importance), groups, then sociograms are drawn using different layout conventions to convey our knowledge and see whether subjects can perceive the same. Therefore, specific hypotheses to be tested are:

1. Group layout is more effective than others to convey group information; hierarchical and radial layouts are more effective than others to convey actor social status.
2. Edge crossings has a significant impact on sociogram perceptions.

3.1 Subjects

Subjects were twenty-seven computer science students. Six of them were graph drawing research students. They all were considered as novice subjects since none of them had either academic or working experience related to social networks. The reasons subjects should be novice in this study are:

1. In communicating information about social networks, it is quite common in real world that sociogram viewers are novice.
2. Experienced sociogram readers likely already have knowledge about drawing conventions and edge crossings, therefore findings based on their responses can be biased.

3.2 Tasks

Finding groups and identifying most important actors, which are two main measures frequently used in network analysis [McGrath et al. 1997], were used as online objective tasks. The subjective tasks were:

1. Usability acceptance rating: subjects were required to rate the usability for each of all drawings based on a scale from -3 (completely unacceptable) to +3 (completely acceptable) for importance tasks and group tasks, respectively.
2. Crossing preference rating: Subjects needed to indicate their preferences for each many-crossing (A) and minimum-crossing (B) pair for importance tasks and group tasks, respectively, based on a scale from -2 (strongly A) to +2 (strongly B), where, for example, "Strongly A" means A is strongly preferred over B.

3. Overall usability rating: with all 10 advice network drawings being shown in one page, subjects needed to choose 3 drawings that they least preferred and 3 drawings that they most preferred for their overall usability, using a scale from -3 to -1 and from 1 to 3, respectively.

4. Questionnaires: There were 2 questionnaires with each having a different focus, and to be presented to subjects before and after they were debriefed about edge crossings and drawing conventions, respectively. The first questionnaire asked subjects information about their study background, experience with node-edge diagrams and social networks, how they interpret sociograms, and any network structure and sociogram features that they think may influence (positive and negative) their sociogram perceptions. The second questionnaire asked about their thoughts about conventions and edge crossings.

For the above rating tasks 1-3, subjects were also required to write down a short explanation for each answer.

3.3 Procedure

The formal experiment took place in a computer laboratory, in which all the PCs had the same specifications.

Before starting the experiment, subjects were asked to read the information sheet, sign the consent form, read through and understand the tutorial material, ask questions and practice with the online system. The drawings used for practice were quite different from the ones used in formal tests, since the practice was only for familiarization with the procedure and system, not for them to get experienced with sociogram reading.

Once ready to start, subjects indicated to the experimenter, and started running the online system performing tasks formally. After the online reading tasks followed by a short break which was to refresh subjects' memory, they proceeded with the rating tasks 1-3, and the first questionnaire. Then, after being given a debriefing document explaining the nature of the study, edge crossings and drawing conventions, subjects were asked to do the rating task 1 and task 2 again, and finally finished with the second questionnaire. Subjects were also encouraged to verbalize any thoughts and feelings about the experiment. The whole session took about 60 minutes.

4 Qualitative Data and Discussions

Qualitative data collected from subjects' responses to the rating tasks and questionnaires were combined together and presented in this section. Throughout the discussion, we relate qualitative questionnaire results with quantitative findings (for details, see [Huang et al. 2005b]) and discuss their implications for sociogram design.

4.1 Edge Crossings Effects

Our quantitative results showed that edge crossings had significant effects on user preference, usability acceptance, and group task performance. Nearly 100% of subjects indicated that drawings with fewer crossings are desirable and easy to read. They commented drawings with many crossings "hard to read", "confusing", etc. This is consistent with previous research results on edge crossings effects [e.g. Purchase 1997].

However, no evident results were found suggesting that edge crossings posed a significant impact on

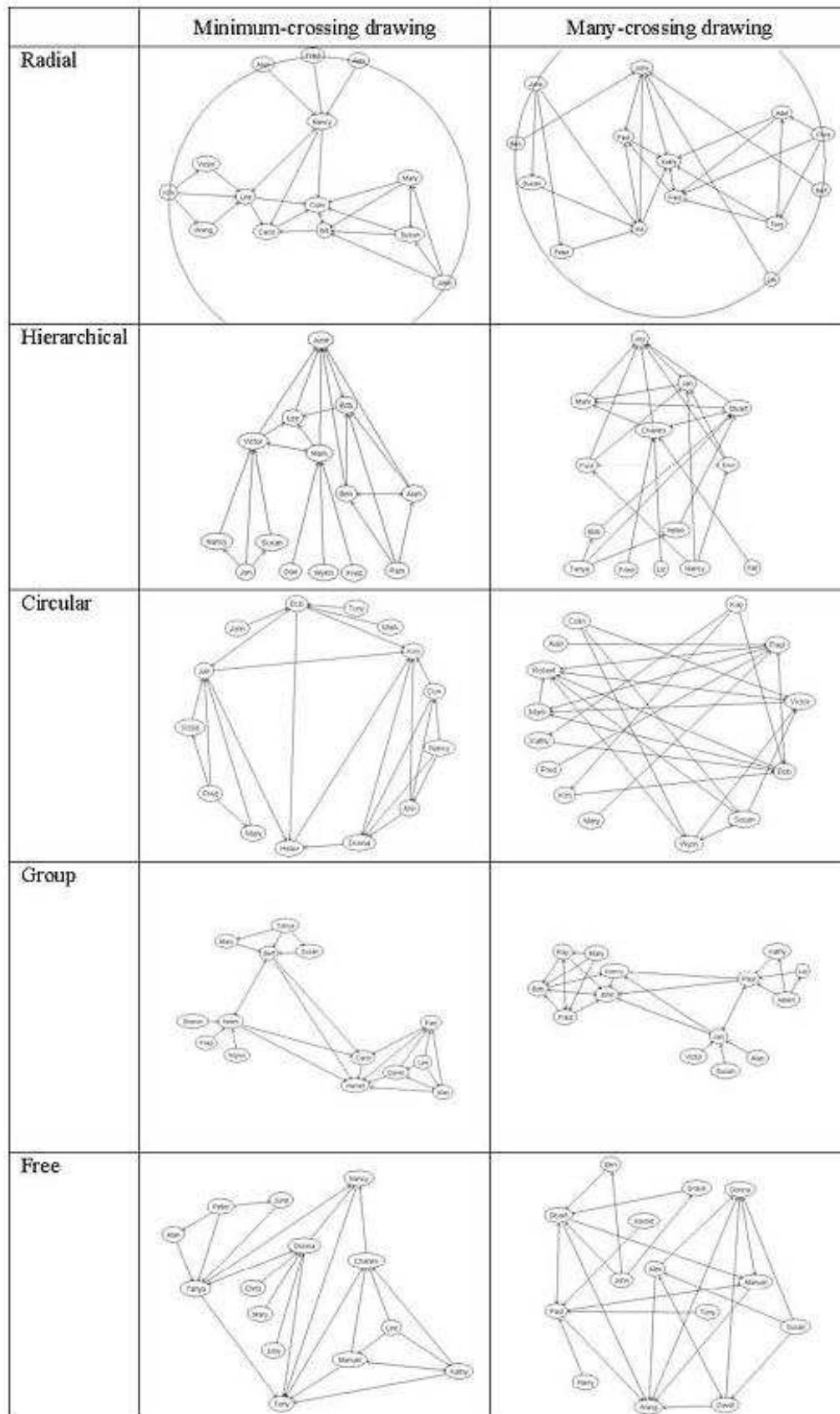


Figure 4: Sociograms used in the study

importance task performance. One reason for this may be that in perceiving an actor's importance, the viewer's main focus is on identifying the node's outgoing and incoming arrows. This can be done by just looking around the node, thereby edge crossings have little impact. Evidence for this explanation can be found in subjects' comments. For example, "In finding important people, maybe location is more important than crossings"; "For finding who is important, just look upon the arrows, but a more traceable graph is helpful". However, most subjects disliked edge crossings. "Crossings matter when finding which nodes are important in a directed graph, and edges with fewer crossings can reveal relationships easily"; "Crossings matter when counting the number of groups".

4.2 Convention Effects

The quantitative analysis revealed that there were significant convention effects on usability acceptance, and group task performance. Again, no apparent evidence was found that there was a significant convention impact on user importance task performance.

For importance tasks, hierarchical convention was strongly preferred, while for group tasks, group convention was strongly preferred. Subjects achieved the highest response accuracy with group convention for group tasks. However, the highest response accuracy did not come with hierarchical convention for importance tasks. For overall usability, group convention was the one for which the usability was rated high and user performance was good as well.

4.2.1 Circular Layout

Circular layout had moderate task performance, but was least accepted in usability. Indeed, it treats all nodes equally by placing them in a circle without any particular priority, thus making network structure features hard to discern. User comments include "I hate meaningless circular layout"; "nodes in the same circle look like they are equally important"; "it is frustrating following long edges from one side to the other side to find groups".

4.2.2 Radial Layout

Radial layout had moderate user ratings and task performance. Although there is visible big circle around the diagram, very few subjects realized nodes were actually arranged radially. Since people's reading habit is normally left-to-right and top-to-bottom, adding a big circle around the diagram does not necessarily make viewers read radially. Providing clear visual hints such as adding circles at each level could be helpful. But this also increases visual complexity, and likely confuses viewers. Some subjects commented like "what is the outer circle for? I do not think it was necessary"; "it is better to have circles visible if radial layout is to be used"; "it is good to put the most important node in the center, but it is crowded with others".

4.2.3 Group Layout

Group layout was rated high for group and overall usability, and moderate for importance usability. It also had very high user performance for group tasks, and moderate performance for importance tasks. User comments include "groups are already spatially separated"; "very clear overall structure"; "good for grouping"; "relationships are clear"; "it would be better if putting the group having the most important person on the top".

One finding worth to mention here is related to edge crossings impact on the task of finding groups. Some subjects commented that, as shown in Figure 5, when sociograms are drawn with group layout, crossing edges within groups gives a stronger sense that nodes are linked closely. In addition, since edges within a group are naturally dense, by crossing edges, it is easier to identify relationship patterns between the nodes. For example, "Crossings probably do not matter when nodes belonging to different clusters have already been separated"; "Crossings are helpful when there is a pattern". This indicates that the introduction of edge crossings into a sociogram may hardly have any negative impact for *communicating groupings*. This is supported by the quantitative user rating and performance data [Huang et al. 2005b].

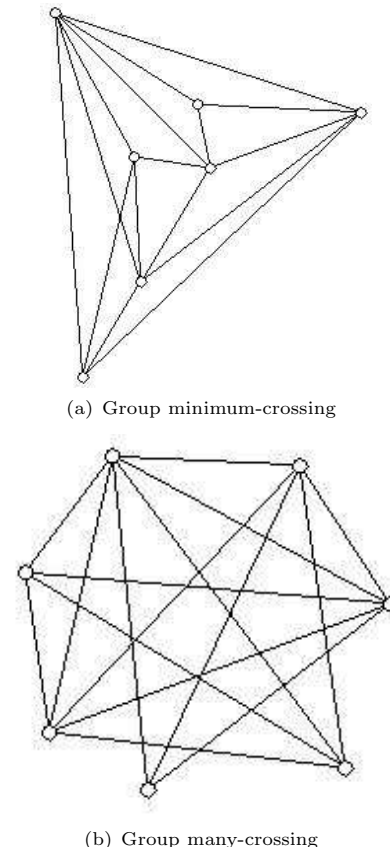


Figure 5: It is argued that Drawing (b) gives a stronger sense of group, and is easier to find relationship patterns than Drawing (a)

4.2.4 Free layout

Free layout had moderate user ratings for importance tasks, and low ratings and performance for group tasks. Some subjects commented that nodes seemed "not organized", "unordered, making grouping difficult"; "some edges were unnecessary long"; "each time I look at the diagram, it seems there are different ways of grouping"; "it is easy to think Nancy is the most important because she is placed on the topright corner, but clearly Tony is more important, However he is easy to miss because he is placed on the bottom"; "it is easy to identify important nodes".

4.2.5 Hierarchical layout

This layout was most preferred for importance tasks, but had low ratings and performance for group tasks.

Comments include “I prefer tree-like layout”; “it is good to arrange arrows pointing the same direction”; “it is easy to see the most important people on the top”; “the directions of edges attached to the top node are hard to tell to me”; “long edges make some nodes in the same group separate”.

4.3 Visual Factors in Sociogram Reading

Subjects’ responses to the questionnaires were categorized and presented as follows:

Category 1: What factors did you consider when determining your answers?

Almost everyone responded that only actual relationships should be considered.

Category 2: Did layout affect you when trying to find answers?

About 85% of the subjects indicated that their final answers were determined by both relations and spatial layout. For example, “50% relations, 50% layout”, “a little bit layout”, “only layout of drawing”, “try my best to find the answer according to relations; when the graph is confusing, I will rely upon the layout”, etc. The rest of the subjects indicated that their answers were only determined by relations. They responded like “The final answers are probably similar, but a bad layout needs more time to understand the answer”; “If a layout is nicer and clearer, I could answer questions faster”, etc.

Category 3: What visual features help or hinder your understanding?

Based on subjects’ responses, nodes and edges in a sociogram should be carefully organized and treated individually according to their individual roles in the network and visualization purposes. For example: nodes should not be evenly distributed; distance between nodes should reflect their relationship; arrows should point to the same direction. Here are some comments: “some important people are crowded with others”, “when relationship dependency makes a cycle, it is difficult to judge who is more important”; “Hierarchy layout really helps to find the important people”; “whether a node is important or not depends on edges it has, not on its position in the diagram”, etc.

Category 4: How to highlight importance and groupings?

Most people preferred putting nodes on the top to highlight importance; some mentioned putting nodes in the center and separating them with others. Almost everyone agreed that nodes which are intensively linked should be visually grouped together. Some examples are: “clustering groups ‘correctly’ helps”; “placing important people on the top or in the center helps”; “groups should be clearly separated”; “low degree nodes on the top should be avoided”, etc.

4.4 Summary and Limitations

In terms of sociogram perception, subjects had a strong preference of placing nodes on the top or in the center to highlight importance, and clustering nodes in the same group and separating groups to highlight groups. They had tendency to believe that nodes in the center or on the top are more important, and nodes in close proximity belong to the same group.

With regard to sociogram reading behavior, subjects tended to consider those nodes in their central attentions high in importance. Due to the close relation between visual attention and eye movements [Ware 1999, p. 154], four hypotheses were made based on subjects’ comments which need confirmation with specifically designed experiments:

1. When looking at a diagram, the first couple of fixations are on the top and central areas, and overall more time is spent on these areas than the other peripheral areas.
2. When looking at a diagram, the first couple of fixations are on sparse areas, and overall more time is spent on sparse areas than dense areas where nodes and edges are crowded.
3. However, their normal reading behavior (eye movement) can be changed and guided by introducing visual hints into diagrams. For example, circles in radial layout.
4. There are significantly more edge searching eye movements for group tasks than for importance tasks. For importance tasks, eye movements are mainly around nodes.

The recommendations for sociogram design are derived and listed below:

1. Reduce crossing number
2. Do not cross edges haphazardly
3. Highlight important nodes, by using color, size, etc.
4. Separate important nodes with others
5. Put the most important nodes on the top or in the center
6. Do not distribute nodes randomly
7. Do not treat nodes equally or distribute nodes evenly
8. Keep edges shorter when underlying relationships are closer
9. Arrange arrows to point the same direction
10. Increase angular resolution
11. Provide visual hints when necessary
12. Cluster nodes which belong to the same group
13. Cross edges when edges are dense in a group
14. Separate groups spatially or by adding boundaries

The hypotheses and recommendations we proposed should be interpreted within the experimental setting. In particular, the recommendation list was drawn mainly based on user responses for performing importance and group tasks. This list surely needs to be refined and extended as our understanding about human sociogram perceptions is getting better. Extra caution should be taken for generalization.

5 Conclusions and Future Work

Analysis of the qualitative questionnaire data not only revealed important insights which otherwise are not necessarily reflected in quantitative measures, but also provided complementary supports and explanations for the findings of quantitative data.

Based on user responses, a set of preliminary recommendations for sociogram design have been proposed. Hypotheses about people’s sociogram reading behaviors have also been made.

This study provides an additional empirical support for the remark of Purchase et al. [2000] that there is no guarantee that aesthetics derived

from domain-independent experiments could automatically apply to domain-specific diagrams. It is important to have visualization techniques evaluated with associated individuals to understand their actual effectiveness in communication, and extra caution should be taken when applying general design principles into the process of constructing domain and task specific diagrams, such as sociograms. The study also highlights a great need for well-proven domain specific aesthetic principles to guide the sociogram design. The recommendations proposed in section 4.4 may serve as the first step toward this aim.

Another contribution is that the study provides qualitative evidence for the conjecture we made in [Huang et al. 2005b]. We conjectured that, edge crossings matters only when the task to be performed requires intensive edge or path tracing, such as finding groups. For tasks such as perceiving importance, the positioning of nodes and angular resolution might be more important compared to reducing the number of edge crossings.

Questionnaire study reported in this paper is only the first step toward understanding how people read sociograms. The hypotheses about human sociogram reading behaviors need confirmation. Certainly questionnaire approach is simply not good enough for a thorough understanding of such a complex issue. More advanced devices and methods such as eye-tracking, think aloud, which have been successfully used in similar research [e.g. Ratwani et al. 2003, Huang et al. 2005a], should be adopted in more finely designed studies in future work.

References

- Brandes, U., Raab, J. and Wagner, D. (2001): Exploratory Network Visualization: Simultaneous Display of Actor Status and Connections. *Journal of Social Structure* 2(4).
- Brandes, U., Kenis, P. and Wagner, D. (2003a): Communicating Centrality in Policy Network Drawings. *IEEE Trans. on Visualization and Computer Graphics* 9(2): 241-253.
- Brandes, U. and Wagner, D. (2003b): visone - Analysis and Visualization of Social Networks. In Michael Jnger and Petra Mutzel (Eds.): *Graph Drawing Software*, pp. 321-340. Springer-Verlag, 2003.
- Di Battista, G., Eades, P., Tamassia, R. and Tollis, I. (1998) *Graph drawing: algorithms for the visualization of graphs*, Prentice Hall.
- Eades, P. (1984) A heuristic for graph drawing. *Congressus Numerantium*, 42: 149-160.
- Freeman, L. (1999) Visualizing Social Groups. *American Statistical Association, 1999 Proceedings of the Section on Statistical Graphics*, 47-54.
- Ghoniem, M., Fekete, J. and Castagliola, P. (2004) A Comparison of the Readability of Graphs Using Node-Link and Matrix-Based Representations. *Proceedings of the 10th IEEE Symposium on Information Visualization (InfoVis'04)*, Austin, TX, Oct 2004. IEEE Press. pp. 17-24.
- Huang, W., and Eades, P. (2005a) How People Read Graphs. *Proceedings of APVIS 2005 (Asia Pacific Symposium on Information Visualisation 2005)*, Australian Computer Society Inc, Conferences in Research and Practice in Information Technology, Vol. 45, pp. 47-53.
- Huang, W., Hong, S. and Eades, P. (2005b) Layout effects on sociogram perception. *Proceedings of 13th International Symposium on Graph Drawing (GD'05)*, September 2005, Limerick, Ireland.
- Kaufmann M. and Wagner D. (Eds.) (2001) *Drawing Graphs: Methods and Models*, Lecture Notes in Computer Science 2025, Springer Verlag.
- Krackhardt, D. (1996) Social Networks and Liability of Newness for Managers. In C. L. Cooper and D. M. Rousseau (eds.), *John Wiley Sons, Ltd. New York, NY. Trends in Organizational Behavior*, Volume 3, pp. 159-173.
- Larkin, J. H. and Simon, H. A. (1987) Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, 11, 65-99.
- McGrath, C., Blythe, J. and Krackhardt, D. (1997) The effect of spatial arrangement on judgments and errors in interpreting graphs. *Social Networks*, 19(3):223-242.
- McGrath, C., Blythe, J. and Krackhardt, D. (1996) Seeing Groups in Graph Layout. *Connections*, 19(2): 22-29.
- McGrath, C. and Blythe, J. (2004) Do You See What I Want You to See? The Effects of Motion and Spatial Layout on Viewers' Perceptions of Graph Structure. *Journal of Social Structure*, Vol. 5, No. 2.
- McGrath, C., Krackhardt, D. and Blythe, J. (2003) Visualizing Complexity in Networks: Seeing Both the Forest and the Trees. *Connections*, 25(1): 37-47.
- Northway, M. L. (1940) A method for depicting social relationships obtained by sociometric testing. *Sociometry*, vol. 3, no. 2, pp. 144-150.
- Purchase, H. C., Cohen R. and James, M. (1995): Validating graph drawing aesthetics. *Proceedings of the Graph Drawing Symposium, Passau, Germany*, 435-446, Springer-Verlag.
- Purchase, H. C., Carrington, D. A. and Alder, J. A. (2002) Graph Layout Aesthetics in UML Diagrams: User Preferences. *J. Graph Algorithms Appl.* 6(3): 255-279.
- Purchase, H. C. (1997) Which aesthetic has the greatest effect on human understanding? In Di Battista, G., editor, *Proceedings of the 5th International Symposium on Graph Drawing (GD '97)*, volume 1353 of *Lecture Notes in Computer Science*, pages 248-261. Springer.
- Ratwani, R. M., Trafton, J. G. and Boehm-Davis, D. A. (2003). Thinking graphically: Extracting Local and Global Information. In R. Alterman, D. Kirsch (Eds.), *25th Annual Meeting of the Cognitive Science Society*. Boston, MA: Erlbaum.
- Scott, J. (2000) *Social Network Analysis: A Handbook*. Sage Publications, 2nd edition.
- Ware, C. (1999) *Information visualization: perception for design*. Morgan Kaufman.
- Ware, C., Purchase, H. C., Colpoys, L. and McGill, M. (2002) Cognitive Measurements of Graph Aesthetics, *Information Visualization*, 1(2), pp 103-110.
- Wasserman, S. and Faust, K. (1994) *Social Network Analysis: Methods and Applications*. Cambridge University Press.

Predicting Graph Reading Performance: A Cognitive Approach

Weidong Huang, Seok-Hee Hong, and Peter Eades

IMAGEN Program, National ICT Australia Ltd.
School of Information Technologies, University of Sydney, Australia
Email: {weidong.huang, seokhee.hong, peter.eades}@nicta.com.au

Abstract

Performance and preference measures are commonly used in the assessment of visualization techniques. This is important and useful in understanding differences in effectiveness between different treatments. However, these measures do not answer how and why the differences are caused. And sometimes, performance measures alone may not be sensitive enough to detect differences. In this paper, we introduce a cognitive approach for visualization effectiveness and efficiency assessment. A model of user performance, mental effort and cognitive load (memory demand) is proposed and further mental effort and visualization efficiency measures are incorporated into our analysis. It is argued that 1) combining cognitive measures with traditional methods provides us new insights and practical guidance in visualization assessment. 2) analyzing human cognitive process not only helps to understand how viewers interact with visualizations, but also helps to predict user performance in initial stage. 3) keeping cognitive load induced by a visualization low allows more memory resources to be available for high level complex cognitive activities. A case study conducted supports our arguments.

Keywords: graph reading, cognitive load, mental effort, cognitive model, visualization efficiency, social network.

1 Introduction

Visualizing relational data as node-link diagrams has long been a hot topic and a key technique in research communities such as graph drawing, information visualization and network analysis. It is widely accepted that visualization assists people in understanding data and communicating information about the data with each other, by providing “cognitive support” [Tory et al. 2004] through a number of mechanisms. For example, representing data visually makes it possible that some tasks can be done by using few simple perceptual operations, which otherwise can be a laborious cognitive process; a visualization serves as external memory to reduce demands on human memory [Card et al. 1999].

However, this does not mean that visual representations are always more effective and less demanding than data in its original format. People process information in working memory [Baddeley 1986]; working memory has a limited capacity. An inappropriate vi-

ualization may impose high cognitive load on human cognitive system, thus overwhelming viewers and undoing its benefits of cognitive support.

Previous research in graph drawing and information visualization has mainly focused on creating visually pleasing pictures conforming to some predefined aesthetic criteria, designing fast algorithms for displaying large and complex data, and adopting new technologies into the visualizing process taking for granted that “cool” technologies and techniques will help to attract people and facilitate understanding. Many results in this area can be found in proceedings of annual conferences such as the Graph Drawing Symposium, the Visualization Conference, etc., and books [e.g. Di Battista et al. 1998, Card et al. 1999]. However, another side of the issue seems to be overlooked and we believe that it should equally receive attention and effort from researchers and practitioners if not more: how effective the final visual representation is in terms of human understanding. In other words, how much information users who view the visualization can actually retrieve should be taken into account in assessing the overall quality of visualization methods. The current visualization challenges can be summarized as follows:

Instead of concentrating on building more and more elaborated systems of rules, there must be an effort to accommodate the innate and vast human perceptual capacity. The deficiency in many computer graphics presentations is not in the output volume, but in the display itself. More intelligent computer programs are not needed, but more intelligently-designed computer displays are [Richards 1992].

Although progress is still slow, fortunately, more and more evaluations have been seen from information visualization and graph drawing communities [Chen 2000]. In order to evaluate visualization effectiveness, objective task performance (response time and response accuracy) and subjective user preference ratings are frequently used as measurements in previous empirical studies [e.g. Ghoniem et al. 2004, Ware et al. 2003, Purchase 1997, 1998, Purchase et al. 2002, Huang et al. 2005]. This is useful in revealing differences across experimental treatments and understanding relationships between visual factors and visualization comprehension. However, these individual measures fail to answer how and why those factors affect user performance, and do not explain how people use their perceptual and cognitive capabilities to complete the tasks. Thus it is no wonder that these measures can sometimes lead to conflicting results [Huang et al. 2005].

In this study, we propose a cognitive model of user performance, mental effort, and cognitive load (memory demand). Further, as a preliminary step,

Copyright ©2006, Australian Computer Society, Inc. This paper appeared at Asia-Pacific Symposium on Information Visualization (APVIS 2006), Tokyo, Japan, February 2006. Conferences in Research and Practice in Information Technology, Vol. 60. K. Misue, K. Sugiyama and J. Tanaka, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

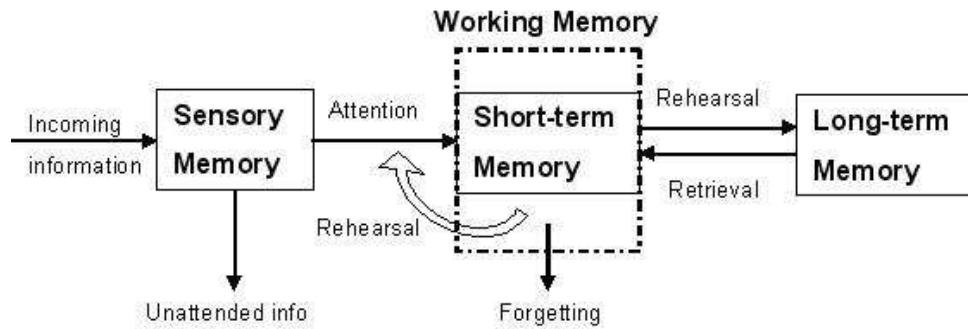


Figure 1: Information processing model

we partly validate this model by conducting a user study of multi-relational social network visualization. We argue that:

1. Combining cognitive measures with traditional performance measures provides us additional insights and practical guidance in assessing the overall quality of a visualization method.
2. Analyzing human cognitive process not only helps us to understand how people interact with visualizations and process information, and why a particular visualization is hard or easy to interpret, but also assists us in predicting user performance so that a cognitively user-friendly visualization method can be determined in as early as initial design phase.
3. To visualize is more than just to represent data. For example, while visually exploring social network data (primary task), we may actively compare what is perceived from a visualization with related prior knowledge retrieved from long time memory, and decide what to do next; that is, we perform a simultaneous decision making process (secondary task). Lowering cognitive load induced by a visualization allows more memory resources available for effective decision making. On the other hand, the high cognitive load induced may overload memory capacity and cause errors in primary tasks, let alone secondary tasks.

The rest of the paper is organized as follows. In section 2 the memory information processing model and measurement methodologies are briefly introduced. In section 3, we propose a theoretical model of user performance, mental effort and cognitive load (memory demand). A case study is presented to validate the model using mental effort and efficiency measures in section 4, followed by discussions and future work in section 5. Finally we conclude in section 6.

2 Background

2.1 Information Processing Model of Memory

Human memory can be considered as an information processing system. It has three basic components: *sensory memory*, *short-term memory* and *long-term memory* [Atkinson et al. 1968]. As can be seen from Figure 1, the incoming information from outside must be processed through sensory memory. Attended information is passed to short-term memory and processed in working memory [Baddeley 1986]. In order to have a good memory, efficient cognitive strategies are required for transferring information from sensory

memory all the way to long term memory. The information stored in long term memory can also be retrieved back to working memory for processing when it is needed.

The bottleneck for efficient information processing is that working memory has its limitations in *duration* and *capacity*. Working memory has a finite capacity, known as “the magical number seven plus or minus two” [Miller 1956]. Moreover, information stored in working memory decays over time; in other words, working memory retains information only for a limited period of time. Apart from information storage, working memory also is used for processing elements of information during the performance of cognitive tasks; therefore, there is considerable competition for working memory. A direct implication of these limitations is that, when a cognitive activity involves the following situations, high memory load and information loss can be expected, causing errors and resulting in poor performance.

1. Long time to process;
2. Many elements to be held in memory;
3. Many elements to be processed simultaneously (high element interactivity);
4. Many mental operations.

2.2 Measurements

In previous user studies investigating effects of visualization formats on human graph reading, response time (RT) and response accuracy (RA) are commonly used to measure effectiveness [e.g. Ghoniem et al. 2004, Purchase 1997, 1998, Bovey et al. 2004]. While it is normally claimed that a visualization which allows subjects to take less time and make fewer mistakes is more effective, these performance measures are either analyzed separately, or RAs are compared first, then RTs are compared if RAs are the same. However, there are difficulties here for interpretation of results, as it is not often that a visualization makes subjects obtain highest RA and shortest RT at the same time. Instead, it is very likely that subjects obtain a relatively high response accuracy, but take a very long time as well, which makes it difficult to judge the overall quality, thus offering little practical guidance in choosing a proper visualization.

Further, when working with, for example, two different visualizations (i.e. different in visual complexity) of the same data, within cognitive capacity, it is feasible and possible that the same viewer expends more mental effort to compensate for the increased cognitive load induced by higher visual complexity with one visualization, thereby obtaining the same

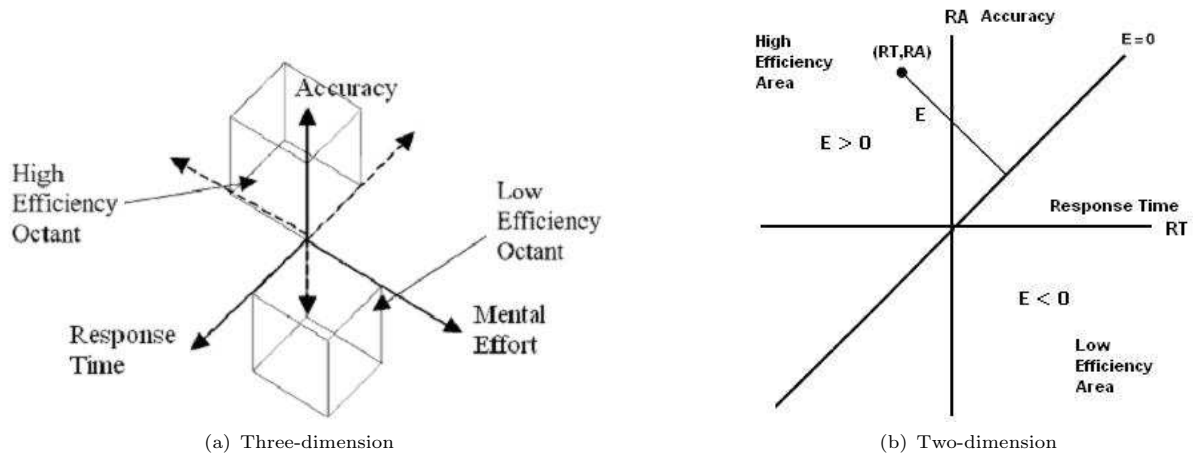


Figure 2: Multi-dimensional visualization efficiency. (a): modified from Tuovinen et al. [2004]; (b): modified from Kalyuga et al. [1999]

level of performance as with the other. Performance-based measures alone, therefore, may not be sensitive enough to reflect the differences between the cognitive costs and to distinguish effectiveness differences between the two visualizations.

To introduce new measurements and integrate them into the existing ones to reflect the actual human cognitive cost in graph reading, the cognitive load and efficiency methodologies, which are well known and have long been used in instructional science and educational psychology, were applied and experimentally tested for the first time in the context of graph reading.

2.2.1 Measuring Cognitive Load

Cognitive load is the amount of human cognitive resources needed to perform a given task. *Mental effort* is the cognitive capacity that is actually allocated to accommodate the demand imposed by the task, and considered to reflect the actual cognitive load. Therefore cognitive load can be measured by obtaining user perceived mental effort invested [Paas et al. 2003]. In this study, we ask subjects to rate the mental effort experienced based on the 9-point Likert scale from 1: very, very low mental effort to 9: very, very high mental effort, which was originally developed by Paas [1992].

2.2.2 Measuring Visualization Efficiency

Visualization efficiency can be calculated using the procedure of Tuovinen et al. [2004], which was originally proposed to measure the relative efficiency of different instructional conditions. By adopting this approach, mental effort is related to performance measures in the context of visualization efficiency, which is defined as the ratio between cognitive cost (in this study, mental effort and response time) and response accuracy. High efficiency is achieved when high accuracy is associated with low mental effort and short response time, whereas low efficiency occurs when low accuracy is associated with high mental effort and long response time. The efficiency (E) is calculated by converting mental effort (ME), response time (RT) and accuracy (RA) into z -scores, and combining them using the following formula:

$$E = \frac{RA - ME - RT}{\sqrt{3}}$$

According to this expression, an E of zero means

that cognitive cost and performance accuracy are balanced.

If the means of these z -scores are plotted in a 3D coordinate system with Accuracy on the vertical axis as shown in Figure 2(a), then E represents the perpendicular distance from the 3D dot to the plane whose equation is:

$$RA - ME - RT = 0 .$$

3 Model of Task Performance, Mental Effort and Cognitive Load (memory demand)

In performing a cognitive task, cognitive load can be induced by the following factors:

1. *Domain complexity*: different application domains have different data formats and contents, define different tasks, and require different visualizations. This factor defines specific visualization constraints and requirements, and provides a basis for overall complexity. For example, visualizing biological data should be different from visualizing social network data, due to the difference of information which biologists and sociologists look for and the distinction of data in nature which is determined by the biological features and social characteristics associated with the data.
2. *Data complexity*: this means the amount of the data. It includes objects in the data, attributes of the objects and relationships between them.
3. *Task complexity*: it determines the load imposed by task demand including the number of objects involved and interactivity of these objects. This is an objective measure, and should be distinguished from task difficulty, which refers to how a task is experienced and perceived by users, and can be affected by other factors, besides task complexity.
4. *Visual complexity*: visual complexity includes all visual elements and their spatial distributions. It is determined by how these elements are visually represented, how well their spatial relationships match their intrinsic structural relationships, and to what extent the spatial arrangement conforms to human visual perception conventions and cognitive process required by a particular task. Therefore a visualization with fewer elements or based on general aesthetics does not

necessarily always lead to lower visual complexity.

5. *Demographic complexity*: this includes motivation, age, gender, cognitive skills, domain knowledge, visualization interpretation experience, mental status, and so on. For instance, the more domain knowledge the viewer has, the lower the effort will be; elderly people have shorter memory span than youngsters.
6. *Time complexity*: time complexity not only has impact on the viewer's task performing strategy and performance, but also affects his mental and physical well-being, which in turn affects performance. Time pressure imposed by reactions to different time conditions has "the most profound and widespread effects on workload perceptions and performance" [Kerick et al. 2004].

Note that the factors above affect cognitive load not only individually, but also interactively. For example, a visualization which easily reminds the viewer of prior knowledge stored in memory may impose lower cognitive load. Time pressure induces anxiety; increasing anxiety results in task-irrelevant thoughts impairing task performance [Eysenck 1979, Coy 1997, Earles et al. 2004].

Informed by De Waard's model of driver's workload [De Waard 1996] and Wickens and Hollands' function of schematic resource supply and demand [Wickens et al. 1999, p. 460], we propose a theoretical model of user task performance, mental effort and cognitive load (memory demand), which is outlined in Figure 3.

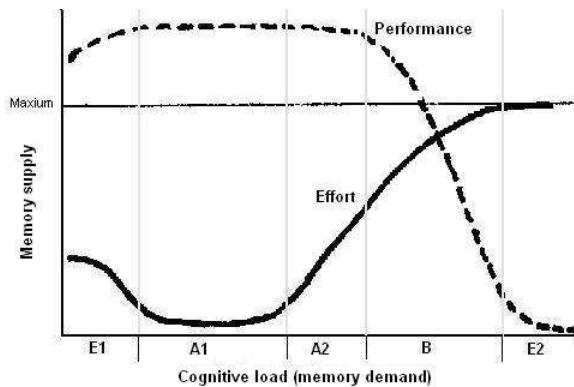


Figure 3: Model of user performance, mental effort and cognitive load

In region E1, where memory demand is consistently low, performance may be either maintained at the optimal level by the viewer investing extra effort to keep concentrated, or negatively affected due to boredom or disregard. In region E2, the demand on viewer's memory exceeds resources available; the viewer is overburdened and performance is at a minimum level. In region A1, the viewer can comfortably handle with demand increases without increasing effort; performance remains at a stable optimal level. In region A2, while demand increases, performance measures still remain unaffected. However, the viewer is only able to maintain the level of performance, by devoting more effort within the capacity of memory. With continuous increases in memory demand, exerting more effort is no longer workable to maintain performance optimal, a shift from region A2 to region B takes place. In region B, performance is affected and a decline appears. At a certain moment, memory demand will exceed the viewer's capability of effort

compensation and region E2 is entered, where performance level has dropped to a minimum level.

4 Case Study: Social Network Visualization

In visualizing social networks, a well considered approach is trying to incorporate information as much as possible into a single two-dimensional static graph visualization by means of various visual devices such as color, layout, etc. [McGrath et al. 2003]. Recently, to cater for the challenges introduced by large and complex networks, many new tools and techniques have become available. However, as pointed out by McGrath et al. [2003], our progress in understanding how people perceive from those fancy visualizations does not match the advances in visualization techniques. In this study, we visualize three multi-relational networks of different complexity levels (see Table 1) in two ways. One way is representing data as a single graph visualization using colors to represent different relations, called *combined version*. The other way is multiple graph visualizations with each representing one relation, called *separated version*. One specific goal of this study is to see how user performance and mental effort are affected by different visualization strategies (visual complexity) and network data (data complexity).

Graph reading is basically a visual search activity. We manipulated the complexity of tasks based on the theoretical model proposed by Wood [1986], in which component complexity was described as a function of the number of acts to be executed and the number of information elements to be processed for performing a task. Using search coverage or visual elements involved as a major consideration of component complexity, based on the analysis of cognitive processes of candidate tasks we proposed beforehand, three specific tasks of different complexity levels are identified as follows:

1. Simple: find one neighbor of a given actor (node),
2. Medium: find all common neighbors of two given actors, and
3. Complex: find all triangle patterns.

Further, in determining tasks, we also had the following criteria in mind, which were considered critically related to the control of confounding factors and the validation of experiment.

1. Tasks should not be too simple, and could not be accomplished by simple perception operations.
2. Tasks should not be too complex in terms of understanding and processing.
3. Task definitions themselves should be clear and short. Varying lengths and components in task definitions required varying amount of memory resources. In addition, tasks which require high effort to remember also competes resources with task performing.
4. Tasks should be social network specific and are actually being used in social network analysis; but do not require in-depth domain knowledge.

Therefore, another specific goal is to see how user performance is affected by task complexity.

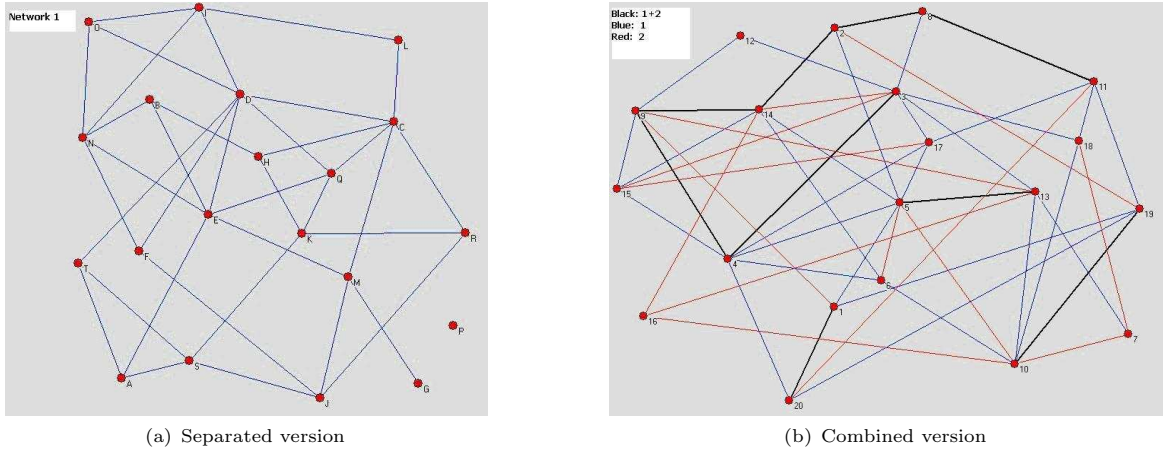


Figure 4: Examples of stimuli. Note that during the formal tests, only relation 1 was shown in (a) since other relations are irrelevant to the tasks

4.1 Hypotheses

In exploring data and communicating information using visualization, cognitive load induced by visual complexity can be reduced by proper arrangement of visual elements. By understanding cognitive process while performing a task and how this process is interfered by various complexity factors, we expect that the mental effort expended varies with levels of complexity, and performance is also affected accordingly. In general, if a cognitive process takes long time, requires users to hold too many items, involves intensive interaction between the items and mental operations in working memory, then it will induce high cognitive load and cause errors. The specific predictions are made as follows:

1. In terms of task (data) complexity, more complex tasks (data) require longer time and more mental effort to process than those less complex tasks (data).
2. With regard to visual complexity, combined versions require longer time and more mental effort to accomplish tasks, and are less efficient than separated versions.
3. When higher mental effort is expended, lower response accuracy, and lower user confidence can be expected.

4.2 Experiment

A within-subjects 3 networks (data complexity) \times 2 separated and combined versions (visual complexity) \times 3 tasks (task complexity) repeated measures design was employed in this experiment.

| Network | 1 | 2 | 3 |
|------------------------|----|----|----|
| Size | 16 | 20 | 25 |
| Links in relation 1 | 15 | 36 | 51 |
| Includes: shared links | 10 | 9 | 12 |
| Links in relation 2 | 21 | 23 | 27 |
| Links in relation 3 | | | 20 |

Table 1: Network data used in the study

Since the networks used were of multi-relations (see Table 1), to control the complexity of design and analysis, we decided that tasks were all about relation 1 for all the networks. We drew each of the three networks as combined and separated versions, leading to six diagrams (or *sociograms*) in total (see Figure 4

for examples). To avoid any possible bias introduced by a layout particularly suitable for a task, two sets of diagrams were produced by Pajek [Batagelj et al. 1998], which is a very popular social network analysis and visualization tool among sociologists, with hand tuning to remove overlaps. Only one set was randomly chosen for each subject. All diagrams were presented online by a custom-built system, using a 6×6 Latin Square design to counteract order effects. The order of the three tasks was also randomized.

For task 1 and task 2, actors (nodes) required had been carefully chosen and highlighted as green so that:

1. For each network, the given actor for task 1 and two actors for task 2 were the same in both the separated version and combined version, respectively.
2. For each task, the number of links involved increased with the complexity of data.
3. In particular, for task 2, the two given actors were ensured to have the same link distribution across relations, i.e., for example, in the combined version of a two-relation network, if one chosen actor had three shared links, one link for relation 1, and four links for relation 2, then the other one should also have the same relation pattern, which means three links in common, one link for relation 1 and four for relation 2. By doing this, in performing task 2, no matter which actor subjects started with to look for common neighbors, the same cognitive operations were needed.

4.2.1 Subjects

Subjects recruited were 30 students who were enrolled with the School of Information Technologies. They were either postgraduate coursework or research students. All of them were familiar with node-link diagrams. When presented with a sociogram, all of them had no trouble performing tasks based on the social aspect of the underlying network, while none of them had formal academic or working experience related social networks. The subjects participated in the experiment on a completely voluntary basis, and were reimbursed \$20 upon the completion of their tasks.

4.2.2 Procedure

The experiment was conducted in a computer laboratory on individual basis. First subjects read through

a tutorial sheet including graph theory concepts, social networks, tasks and procedure. They also had chance to ask questions and practice the system so that they understood everything needed. Then subjects started experiment formally by pressing a button on the screen. Since subjects needed to perform three tasks for each diagram, the same diagram was presented three times. Each time when a diagram is shown, subjects were required to answer a question as quickly as possible without compromising accuracy, and immediately after writing down the answer on the answer sheet provided, they pressed a button. The corresponding response time was logged starting when a diagram was shown and ending when a button was pressed. They continued to indicate on the answer sheet their mental effort based on the scale from 1 (very, very low mental effort) to 9 (very, very high mental effort) and confidence level about their performance from 1 (very uncertain) to 7 (very confident). Then they pressed the button again to proceed with the next diagram. After online tasks, subjects finished the experiment with a simple questionnaire. During the tutorial, subjects were also instructed that they were not allowed to take notes, talk aloud, or use mouse to help in performing reading tasks. The whole session took about 50 minutes.

4.2.3 Results

For task 1 and task 2, accuracy was recorded as 1 if given answer was correct and 0 otherwise. For task 3, the accuracy was used measuring how far the response was from zero, and calculated in the following way: if the given answer was no more than the correct answer, accuracy was recorded by dividing the given answer by the corresponding correct answer. Otherwise, accuracy was obtained by subtracting the difference between the given answer and correct answer from the correct answer first, then dividing the obtained value by the corresponding correct answer; if the obtained accuracy is less than 0, 0 was recorded. The higher the accuracy value, the better the performance.

Due to the limit of space and time, only partial results are reported here. However, it is our desire to fully report the experimental results and findings elsewhere, while results reported here are considered sufficient for the purposes of this paper. Dependant variables analyzed included response accuracy, response time, mental effort and confidence. For different visualization formats (separated version vs. combined version), efficiency was analyzed as well. Throughout our analysis, the significant level of 0.05 was used.

Data Complexity Effect: Table 2 shows the mean and standard deviation (in parentheses) scores of separated versions for the three networks. As we can see, with an increase in data complexity, mental effort increased as well. *Analysis of variance* (ANOVA) showed that there was a significant data complexity effect on mental effort expended ($p < 0.001$). Analysis also revealed that all other dependent variables were significantly affected ($p < 0.05$).

Task Complexity Effect: The mean and standard deviation (in parentheses) scores of the combined version of network 1 for the three tasks were shown in Table 3. It can be seen that mental effort increased with task complexity. Subjects exerted lowest effort while performing task 1, followed by task 2; the highest effort was expended for task 3. ANOVA revealed that task complexity posed a significant impact on mental effort expended ($p < 0.001$). Analysis also revealed that all other dependent variables were significantly affected ($p < 0.001$) except accuracy. A possible reason for this exception might be because network 1 was so simple that subjects could get all tasks done at the optimal level.

Visual Complexity Effect: We compared all dependant variables between the separated version and the combined version for each network for task 3. The scores of mean and standard deviation and analysis results (p values) are shown in Tables 4 - 6.

It can be clearly seen that for tasks 3, the separated version was more effective than the combined version for each network. Subjects spent shorter time and achieved higher accuracy with the separated version; they also expended lower effort and felt more confident about their performance with the separated version than the combined version. *Two-sample t* tests showed that all these differences were statistically significant ($p < 0.05$) with some exceptions as shown in Tables 4 - 6. In particular, for network 1, while subjects expended significantly less effort with the separated version than the combined one, the mean values are both very low (2.17 and 3.23, respectively). On the other hand, the high mental effort values and significant effort difference for network 3 (see Table 4) suggested that, when data complexity increases, using separated version can be more beneficial in effectiveness. From Table 6, it can be seen that there was no statistical difference with accuracy ($p > 0.05$); however, mental effort values did show a significant difference. This supports our analysis that sometimes performance measures are not sensitive enough to detect differences, but mental effort was in this case.

Although the statistical analysis did not show significant effects for all dependent variables, the efficiency measure consistently showed that the separated version was more efficient than the combined version, and *t* tests revealed that this difference was statistically significant ($p < 0.001$), as predicted.

5 Discussions

The experimental hypotheses that the means of all responses in terms of data complexity, task complexity and visual complexity would differ respectively, were supported by the experimental data.

The results of this study have shown that when a graph becomes large and dense, even with only 25 nodes and 98 links, human perception and cognitive systems can quickly be overburdened causing errors in performing relatively complex tasks, for example, finding relationship patterns. When visualizing large and complex data, the efficient use of limited human cognitive capacity stands out to be an important and crucial issue, becoming a major factor that determines the success of a particular visualization technique.

Cognitive load consideration provides us with additional insights and practical guidance in visualization assessment. Of course, visualization effectiveness depends on many factors apart from those affecting working memory load. However, since any visual information has to pass through human sensory system first, and be processed in working memory, failure to respect human cognitive capacity might override any positive benefits of new technologies and techniques introduced in the course of visualization.

Some people may argue that when tasks are easy, or data sets are small, it does not matter how much mental effort is devoted as long as it is within the limitations of human cognitive capacity. It is true on one hand; as can be seen from our model (Figure 3), optimal performance can be maintained by increasing effort. However, on the other hand, persistent high effort may cause, for example, fatigue, which in turn leads to effort increases. Besides, increased mental effort devoted during performing a reading task leaves little memory resource for simultaneous decision mak-

| Sep. ver. | Accuracy | Time | Effort | Confidence |
|-----------|-------------|----------------|--------------|--------------|
| Net. 1 | 2.96 (0.19) | 46.90 (29.08) | 5.67 (2.95) | 20.53 (0.90) |
| Net. 2 | 2.54 (0.56) | 100.14 (44.74) | 9.37 (3.34) | 18.80 (1.61) |
| Net. 3 | 2.44 (0.62) | 141.21 (69.12) | 10.37 (3.26) | 17.93 (1.89) |

Table 2: Mean and standard deviation scores of separated versions

| Com. ver. | Accuracy | Time | Effort | Confidence |
|-----------|--------------|---------------|-------------|------------|
| Task 1 | 1.00 (0.00) | 11.39 (9.18) | 1.67 (1.12) | 6.9 (0.31) |
| Task 2 | 0.933 (0.25) | 24.89 (14.73) | 2.60 (1.25) | 6.7 (0.54) |
| Task 3 | 0.925 (0.20) | 32.18 (16.54) | 3.23 (1.55) | 6.2 (0.89) |

Table 3: Mean and standard deviation scores of the combined version for network 1

| Net. 3 | Accuracy | Time | Effort | Confidence | Efficiency |
|-----------|-------------|----------------|-------------|-------------|---------------|
| Sep. ver. | 0.74 (0.24) | 100.16 (59.78) | 5.83 (1.86) | 4.47 (1.57) | 0.546 (0.54) |
| Com. ver. | 0.52 (0.25) | 139.43 (90.15) | 6.90 (1.79) | 3.70 (1.76) | -0.546 (0.73) |
| T value | 3.48 | -1.99 | -2.27 | 1.78 | 6.59 |
| P value | 0.001 | 0.052 | 0.027 | 0.081 | 0.000 |

Table 4: Means and p values of network 3 for task 3

| Net. 2 | Accuracy | Time | Effort | Confidence | Efficiency |
|-----------|-------------|----------------|-------------|-------------|--------------|
| Sep. ver. | 0.84 (0.14) | 64.84 (34.48) | 5.23 (1.77) | 5.20 (1.37) | 0.54 (0.77) |
| Com. ver. | 0.69 (0.18) | 100.65 (54.20) | 5.73 (1.76) | 4.47 (1.50) | -0.54 (0.82) |
| T value | 3.60 | -3.05 | -1.10 | 1.97 | 5.27 |
| P value | 0.001 | 0.004 | 0.278 | 0.053 | 0.000 |

Table 5: Means and p values of network 2 for task 3

| Net. 1 | Accuracy | Time | Effort | Confidence | Efficiency |
|-----------|-------------|---------------|-------------|-------------|--------------|
| Sep. ver. | 0.99 (0.05) | 19.90 (15.35) | 2.17 (1.21) | 6.73 (0.52) | 0.55 (0.73) |
| Com. ver. | 0.93 (0.20) | 32.18 (16.54) | 3.23 (1.55) | 6.20 (0.89) | -0.55 (1.35) |
| T value | -1.79 | 2.98 | 2.98 | -2.84 | -3.92 |
| P value | 0.083 | 0.004 | 0.004 | 0.007 | 0.000 |

Table 6: Means and p values of network 1 for task 3

ing process, which is quite common in many application domains while exploring and understanding data, as discussed in section 1. In cases where facilitating high level complex decision making is included as one of criteria of visualization assessment, or data to be visualized are considerably large and complex, cognitive load can be a critical and determining factor affecting the success of visualization.

It is important to note that, we were not proposing that the separated version was always better than the combined version since all the tasks used in this study were relevant to only one relation of multi-relational networks. What we suggested is that, when visualizing a large and complex network, it is necessary to be aware of the role of cognitive load induced by different visualization strategies. In some cases, multiple separated graphs might be a better option rather than putting all information into a single static graph; and this is supported by the cognitive results obtained in the user study. In addition, experiment conditions were manipulated in order to demonstrate our model.

Three limitations of the case study have been identified. First, subjects were told to use only memory and eyes to accomplish tasks. However, it is likely that they still could get help by switching between answers and diagrams when they were trying to write down answers on the provided answer sheet, although they might not consciously realize this. Second, the networks used in the study were typically small in general. We did this on purpose, as a result of pilot study. With complex visualizations, we got very strong reactions from subjects. Based on their feedback, they either simply did not want to go ahead with tasks, or took long time to think about how to approach diagrams, or answered questions by guessing after a few unsuccessful tries. Alternative experiment design and evaluation techniques should be used in this case, since it is impossible and undesirable to evaluate everything in just one single study. Third, interactions between variables were not included in our design and analysis. The last two will be the subjects of our future research.

5.1 Methodology

In this study, apart from traditional performance measures, we took mental effort devoted while performing tasks into account when measuring effectiveness. This is certainly a valuable addition to traditional methodologies, providing us with an additional dimension in assessing visualization effectiveness.

For the first time, we also introduced a multi-dimensional approach measuring visualization efficiency, which was originally proposed by Tuovinen et al. [2004], into the user studies of graph reading. This method incorporates both task performance and mental workload into one measure allowing us to obtain information on relative efficiency of a visualization, therefore enriching our knowledge about effects of different experimental conditions.

This methodology can also be adapted according to different analysis needs or variables available. In traditional studies, user preference (PRE) is also utilized as a major factor in evaluation. In this case, the efficiency formula can be modified accordingly by replacing ME with PRE:

$$E = \frac{RA - PRE - RT}{\sqrt{3}}$$

This formula treats every variable equally. Alternatively, we might assign different weights (w_i , where $i = 1, 2, 3$) to these variables according to individual priorities in the overall evaluation. Therefore the above equation can be refined as:

$$E = \frac{w_1 \times RA - w_2 \times PRE - w_3 \times RT}{\sqrt{3}}$$

where

$$\sum_{i=1}^3 w_i = 1$$

Further, in cases where only response time (RT) and response accuracy (RA) are measured, the similar two-dimensional approach [Paas et al. 1993] can be used, and efficiency is calculated as:

$$E = \frac{RA - RT}{\sqrt{2}}$$

In a coordinate system as illustrated in Figure 2(b), E represents the perpendicular distance from a point (RT, PA) to the line:

$$RA - RT = 0$$

The high efficiency area is located above the line; and the low efficiency area is located below it.

However, it is important to note that relying solely on this measure sometimes can be misleading, and it should always be considered together with traditional effectiveness measures such as correctness rate. After all, performance is such an important criterion for visualization evaluation, and absolute performance levels are not necessarily directly reflected in the efficiency measure itself [Paas et al. 1993].

5.2 Cognitive Approach

Although human factors, including memory capacity such as mental map, perception convention such as color, aesthetics such as edge crossings, have been taken into account in designing visualization techniques and related research, very few studies are available directly addressing the impact of human cognitive limitations on graph interpretation. This is the first study to empirically demonstrate the limitations of human cognitive capacity in the domain of graph reading.

In this study, we proposed a cognitive model of user task performance, mental effort and cognitive load (memory demand), which certainly needs refinement and improvement. As a preliminary step, in the user study, we have demonstrated that understanding human cognitive process of a graph reading task assists us in predicting user task performance. Therefore visualization researchers and designers are enabled to make a proper judgment of whether a particular visualization technique is good or not in the initial design stage, avoiding the waste of time and effort in making drawings which may turn out to be of little usability.

A good visualization needs not only to convey “right” information, but also to respect the limitations of human perception and cognition systems. A visual representation of given data which induces minimum cognitive load on memory not only frees viewers from laborious and error-prone effort in understanding data, thus avoiding fatigue and increasing confidence, but also allows more resources in working memory for other simultaneous information processing such as decision making.

Our advocacy for keeping cognitive load induced by visual complexity low is consistent with the underlying theory of *constant visual complexity*, which was proposed by Pulo [2005] and Li [2005]. By minimizing visual cognitive load, more memory resources can be

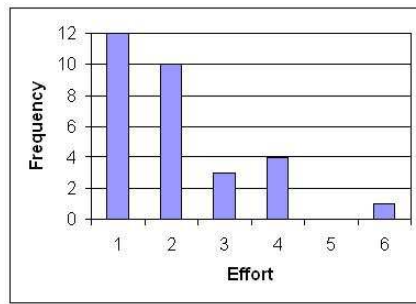


Figure 5: Effort distribution for task 2 for the separated version of network 1

allocated for cognitive load imposed by data complexity and task complexity, which should be increased so that more information can be visualized and effectively conveyed to the viewer. In other words, minimizing visual cognitive load and maintaining overall load at a *reasonable* level at the same time, not only make most use of limited human cognitive capacity, but also foster maximum information retrieving and effective processing.

5.3 Demographic Complexity and Time Factor

In this study, one important factor which was omitted from our investigation is demographical complexity. During the study, differences between individuals in terms of demographical complexity were removed by the within-subjects design. Figure 5 shows the distribution of effort across the subjects when performing task 2 with the separated version of network 1. It can be clearly seen that although task complexity is the same, mental efforts that individuals expended differed greatly for the same visualization. In addition, plenty of empirical evidence is also available in the literature suggesting that cognitive capacity differs with demographical factors [e.g. Van Gerven et al. 2003, Just et al. 1992, Velez et al. 2005], such as gender, age, spatial ability, cognitive skills, etc. However, specific research in graph reading domain is urgently needed to investigate how those human factors interact with different visualizations, and their impacts on visualization comprehension.

As regard to time complexity, during testing, time requirement was set by asking subjects to perform tasks as quickly as possible without compromising accuracy; we used time as a dependant variable in our analysis, instead of setting a fixed time limit. Enormous studies have been conducted in psychology, cognitive science and application domains [e.g. Sioberg 1977, Hwang 1995, Kerick et al. 2004, Earles et al. 2004], suggesting that there is an impact of time availability on perceived mental effort and task difficulty. Increasing time pressure induces anxiety [Tohill et al. 2000] and increased anxiety leads to task-irrelevant thoughts [Eysenck 1979, Coy 1997]; these thoughts consume working memory resources and may impair cognitive task performance. In addition, during tests, it is possible that subjects interpreted the time requirement differently and restricted the time availability at different levels, and changes in time availability may affect their graph reading strategies and behaviors, therefore bringing possible confounding noises into the resulting data. Clearly, further research is needed to specifically investigate time availability effects in the domain of graph reading.

In addition, we used the three dimensional ap-

proach which uses both mental effort and response time as cognitive cost against response accuracy to measure the efficiency of different visualization strategies. However, as mentioned by Tuovinen et al. [2004], it is unclear whether or not time spent had been taken into account as a factor when subjects indicated the mental effort exerted, while *regression* tests did not find obvious correlation between user response time and subjective mental effort. If this is the case, then the alternative two dimensional approach incorporating mental effort and performance accuracy could be more meaningful.

6 Conclusions

Evaluation should be treated as an essential part of the whole visualization process. It has become increasingly important and crucial, as nowadays more and more new technologies and techniques have been made available. We simply have many options when designing new visualizations, or choosing existing ones; evaluation helps us with this process. Traditional performance and preference measures have some limitations in detecting differences between visualizations, revealing the underlying mechanism resulting in the differences. Analyzing these measures separately makes it difficult to judge overall quality and provides little practical guidance in visualization design. This paper proposes a cognitive approach for the evaluation of visualization effectiveness and efficiency.

Human memory architecture and memory information processing model were presented first. Built on this, a cognitive model of user task performance, mental effort and cognitive load (memory demand) has been proposed. For the first time in graph reading domain, we introduced cognitive load and efficiency measures which are well known and commonly used in education psychology research. In the user study, we combined the cognitive measures with traditional performance measures to demonstrate that the cognitive approach equips us with great potential, in assessing overall quality of visualization and predicting graph reading performance in early stage. Revised versions of the efficiency formulas were also suggested to cater for different evaluation needs. Clearly, our cognitive model and suitability of the cognitive measures in graph reading domain still need further examinations. Therefore, this paper also calls for attention and research along this direction.

7 Acknowledgements

We thank Juhani Tuovinen for his valuable comments and explanations on cognitive methodologies, and students who willingly took part in the experiment. Ethical clearance for this study was granted by the University of Sydney, August 2005.

References

- Atkinson R., Shiffrin R. (1968) Human memory: A proposed system and its control processes. In K. Spence and J. Spence, editors, *The psychology of learning and motivation*. Academic Press.
- Baddeley, A. D. (1986) *Working memory*. Oxford, UK: Oxford University Press.
- Batagelj, V., Mrvar, A. (1998) Pajek: Program for large network analysis. *Connections*, 21(2):47-57.

- Bovey, J., Benoy, F., Rodgers, P. (2004) Using games to investigate movement for graph comprehension. In *Advanced Visual Interfaces: AVI 2004*, pages 71-79. May 2004, Gallipoli (LE), Italy.
- Card, S. K., Mackinlay, J. D., Shneiderman, B. (1999) *Readings in Information Visualization: Using Vision to Think*. San Francisco: Morgan Kaufmann Publishers.
- Chen, C. (2000) Empirical evaluation of information visualization: an introduction. *Int. J. Human-computer Studies*, 53, 631-635.
- Coy, W. B. (1997) Evaluation anxiety, task-irrelevant self-statements, and working memory: An evaluation of the cognitive interference theory. PHD thesis, Bowling Green State University, Ohio.
- De Waard, D. (1996) The measurement of drivers' mental workload. PHD thesis, University of Groningen.
- Di Battista, G., Eades, P., Tamassia, R., Tollis, I. (1998) *Graph drawing: algorithms for the visualization of graphs*, Prentice Hall.
- Earles, J. L., Kersten, A. W., Mas, B. B., Miccio, D. M. (2004) Aging and Memory for Self-Perfomanced Tasks: Effects of Task Difficulty and Time Pressure. *Journal of Gerontology: Psychological Sciences*, Vol. 59B, No. 6, 285-293.
- Eysenck, M. W. (1979) Anxiety, learning, and memory: A reconceptualization. *Journal of research in Personality*, 13, 363-385.
- Ghoniem, M., Fekete, J., Castagliola, P. (2004) A Comparison of the Readability of Graphs Using Node-Link and Matrix-Based Representations. *Proceedings of the 10th IEEE Symposium on Information Visualization (InfoVis'04)*, Austin, TX, Oct 2004. IEEE Press. pp. 17-24.
- Huang, W., Hong, S., Eades, P. (2005) Layout effects on sociogram perception. *Proceedings of 13th International Symposium on Graph Drawing (GD'05)*, September 2005, Limerick, Ireland.
- Hwang, M. I. (1995) The effectiveness of graphic and tabular presentation under time pressure and task complexity. *Information Resources Management Journal*. 8(3), 25-31.
- Just, M. A., Carpenter, P. A. (1992) A Capacity Theory of Comprehension: Individual Differences in Working Memory. *Psychological Review*. Vol. 99, No. 1, 122-149.
- Kalyuga, S., Chandler, P., Sweller, J. (1999) Managing split-attention and redundancy in multimedia instruction. *Applied Cognitive Psychology*. 13: 351-371.
- Kerick, S. E., Allender, L. E. (2004) Effects of cognitive workload on decision accuracy, shooting performance, and cortical activity of soldiers. <http://handle.dtic.mil/100.2/ADA433487>. Accessed 15 Nov. 2005
- McGrath, C., Krackhardt, D., Blythe, J. (2003) Visualizing complexity in networks: seeing both the forest and the trees. *Connections* 25 (1): 37-47.
- Li, W. (2005) Navigating clustered graphs. Masters thesis, University of Sydney.
- Miller, G. (1956) The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological Review*, (63):81-97.
- Paas, F. (1992) Training strategies for attaining transfer of problem-solving skill in statistics: A cognitive load approach. *Journal of Educational Psychology* 84: 429-434.
- Paas, F., Tuovinen J. E., Tabbers, H., van Gerven, P. W. M. (2003) Cognitive load measurement as a means to advance cognitive load theory. *Educational Psychologist*, 38(1), 63-71.
- Paas, F., Van Merriënboer, J. (1993) The efficiency of instructional condition: an approach to combine mental effort and performance measures, *Human factors* 35:734-743.
- Paas, F., Van Merriënboer, J. (1994). Instructional control of cognitive load in the training of complex cognitive tasks. *Educational Psychology Review*, 6, 51-71.
- Pulo, K. (2005) *Structural Focus + Context Navigation of Relational Data*. PHD thesis, University of Sydney.
- Purchase, H. C., Carrington, D. A., Alder J-A. (2002) Graph Layout Aesthetics in UML Diagrams: User Preferences. *J. Graph Algorithms Appl.* 6(3): 255-279.
- Purchase, H. C. (1997) Which aesthetic has the greatest effect on human understanding? In G. Di Battista (ed.): *Proceedings of Graph Drawing Symposium 1997*. LNCS, vol. 1351. Rome, Italy: Springer-Verlag, pp. 248-259.
- Purchase, H. C. (1998) Performance of layout algorithms: Comprehension, not computation. *Journal of Visual Languages and Computing* 9: 647-657.
- Richards, Method and apparatus for processing and displaying multivariate time series data, U. S. patent no. 5,121,469 (June 9, 1992).
- Sjoberg, H. (1977) Interaction of task difficulty, activation, and work load. *Journal of Human Stress* 3(1), 33-38.
- Tohill, J. M., Holyoak, K. J. (2000) The impact of anxiety on analogical reasoning. *Thinking and Reasoning*, 6, 27-40.
- Tory, M., Moller, T. (2004) Human factors in visualization research. *IEEE transaction and computer graphics* Vol. 10, No. 1.
- Tuovinen, J., Paas, F. (2004) Exploring multidimensional approaches to the efficiency of instructional conditions. *Instructional Science* 32: 133-152.
- Van Gerven, P., Paas, F., Van Merriënboer, J., Schmidt, H. (2003) Memory load and the cognitive papillary response in aging. *Psychophysiology*, 41, 167-174.
- Velez, M. C., Silver, D., Tremaine, M. (2005) Understanding Visualization through Spatial Ability Differences. *Proceedings of IEEE Visualization*, October, 2005, 23-28. Minneapolis, MN, USA.
- Ware, C., Purchase, H. C., Colpoys, L., McGill, M. (2003) Cognitive measures of graph aesthetics. *Information Visualization*, 1(2), 103-110.
- Wickens, C. D., Hollands, J. G. (1999) *Engineering psychology and human performance*. Third edition, Prentice Hall.
- Wood, R. E. (1986) Task complexity: definition of the construct. *Organizational behavior and human decision processes*, 37, 60-82.

Visual Narratives: The Essential Role of Imagination in the Visualization Process

Mark Baskinger

School of Design
Carnegie Mellon University
Pittsburgh, Pennsylvania U.S.A.
baskinger@cmu.edu

Ki-Chol Nam

School of Art and Design
Yeungnam University
Kyungsan, Korea
nam1@yumail.ac.kr

Abstract

The goal of this short form paper is to introduce ideas in contemporary visualization that use hand generated methods to engage the imagination of the author and audience to enhance the larger creative process. This approach, which focuses on exploring lateral ideas in a narrative format, blends many aspects of the fields of industrial design, communication design, and information design. “Analog” visual communication, defined as hand-generated sketches, diagrams and narratives, is of interest as it can bring powerful perceptual and conceptual processes to bear in making inferences and subsequent decisions. This method of visualization, translatable to many different disciplines, enables the author to think through and investigate his/her ideas by creating sequential image sets that simplify complex/dynamic information, illustrate key issues, and reduce abstract concepts into visually digestible, non-verbal narratives.

Keywords: “analog” visualization, visual narratives, graphical representations, conceptual models

Overview

Visualization, a term used by many disciplines, is interpreted in a variety of ways. It can range from a collection of loosely drawn scribbles to a highly refined storyboard. Regardless of media or method of creation, the intent behind any visualization is to capture and document an idea – to turn data or an idea into information and make it more accessible to a wider audience.

Since the terms “sketch,” “drawing” and “visualization” are commonly interchangeable, we can classify them as “visual communication.” Within this classification, we can investigate common themes or methods that make abstract ideas more concrete with respect to data visualization, product development, and information systems planning.

As designers, we approach visualization first with

hand-generated sketches to think through the abstract ideas in various permutations. We enjoy the journey of discovery in making relationships between intangible ideas/data and the formal elements that make the idea accessible. Our goals in visualization are 1. to externalize and convey the process of thinking – to transform intangible ideas to tangible information; 2. to reveal ideas/relationships, not results; and 3. to engage discussion around the subject/problem and make design an inclusive activity. In our minds, the common link to all of visualization is in constructing a graphical representation in a narrative format, one that speaks to lateral ideas, an evolution of a thought or alternative methods of describing the same content. The images that result from this process serve as vehicles to bring others into the designer’s mind to better facilitate collaboration.

This paper is written from a designer’s perspective and will introduce an approach that focuses on creating visual narratives and conceptual models. The narratives and models will engage the imagination of both the designer and the viewer and will be created first through a series of hand-generated or “analog” methods and progress into the digital realm.

Visualization for Communication: A Narrative Process

Using visual methods to communicate ideas entails creating a sub-structure of non-verbal communication. Too often do designers make hasty, unrefined drawings that must be laboriously over-explained to colleagues and clients. The very premise of visualization is that a conceptual model is created to convey thinking, or “tell a story” to someone else. Therefore, as a visual “story,” it must sequentially reveal information across the viewing plane in an orderly and scripted fashion. A narrative substructure built into the organization, hierarchy and composition of the piece will enable the non-verbal story to unfold. Narratives, which provide accounts for telling the story of events, experiences and ideas, offer concrete touch points for viewers with a sequential format divided into three distinct parts – beginning (to invite the viewer in), middle (to engage the viewer), and end (to provide closure). Simply, the viewer will immediately recognize a starting point, a main body or information, and an ending point to provide a comprehensive visual discourse of the concept. We often make a key distinction that sketching is the visual record of multiple ideas (i.e. idea sketching),

whereas visualization is a disciplined approach to making seamless visual connections among ideas. The differentiation of the two terms enables the use of narrative structures that lead the viewer through the page space and deliver ideas with greater impact. Regardless of the particular emphasis, pictures somehow possess a faster access than words to a common conceptual system. “Picture-word latency differences stem from a time-consuming translation from one symbolic code to another, and that semantic information required in the subsequent decision tasks is typically stored non-verbally” (te Linde, 1983).

Visual narratives can take many forms – from a page of loose sketches/visualization around a common theme to a highly structured and organized matrix (Figure 1) and can include aspects of storyboarding, hybrid drawing and diagramming. Maintaining a sequential structure across the image space will provide cues for orientation and establish some form of closure. By structuring parameters for the viewing experience through composition and hierarchy, the body of the work can manifest in a multitude of ways. Again, the ultimate goal is to communicate ideas and thinking to others. Visual narratives can also speak volumes about the design/creative process.

To properly structure a visual narrative, the author must know to whom they are communicating in terms of knowledge of the subject, familiarity with design processes, and visual/aesthetic sensitivity. Understanding both the character of the audience and the format for presentation will keep the narrative focused and succinct. Again, a narrative must have intent; meaning that it must have a clear and definable purpose to facilitate subsequent discourse around the subject.

Visualization and Imagination: Creating Conceptual Models

The mind’s eye is incredibly adept at revealing an inner vision that transcends the constraints of time and space. Manifesting in seemingly unbridled creativity, pictographic outcomes from imaginative exploration yield greater connections to tie the mind to the subject in more distinct and expedient ways.

Images created in the intangible, yet apparent mind’s eye prove to be elusive and easily lost to awareness and self-consciousness unless properly realized through physical record. Once recorded, drawn images feed back to the mind’s eye to further stimulate imaginative processes and perpetuate a dialogue between self and the image. This is the basis for conceptual visual modelling.

Although mental images often appear spontaneously in response to sensory perception (something heard, touched, tasted, smelled, or seen), we have the innate ability of envisioning recollected or imagined images through the mind’s eye. One might become aware that their mind is simultaneously processing the images drawn in front of them, making connections to previous experiences, mental records, or stimuli in the immediate environment. While imagination is predicated on visual memories, it is also incredibly suited to look forward in time to visualize a possible future. Therefore, conceptual models establish connections to the current information through visual bridges among the past, present, and future experiences. As an example, Figure 2 shows various conceptual models of a time schedule to emphasize time expenditures, focus/effort distribution, and management.

It is important to note here that there has been some debate in various philosophy and psychology circles about the distinction between *imagining* and *visualizing*.



Figure 1: Visualization in a matrix format presents an evolution of ideas for alarm clocks in panel 1 (pen and marker), toasters in panel 2 (pen and marker), and spheroid form exploration in panel 3 (digital modelling).

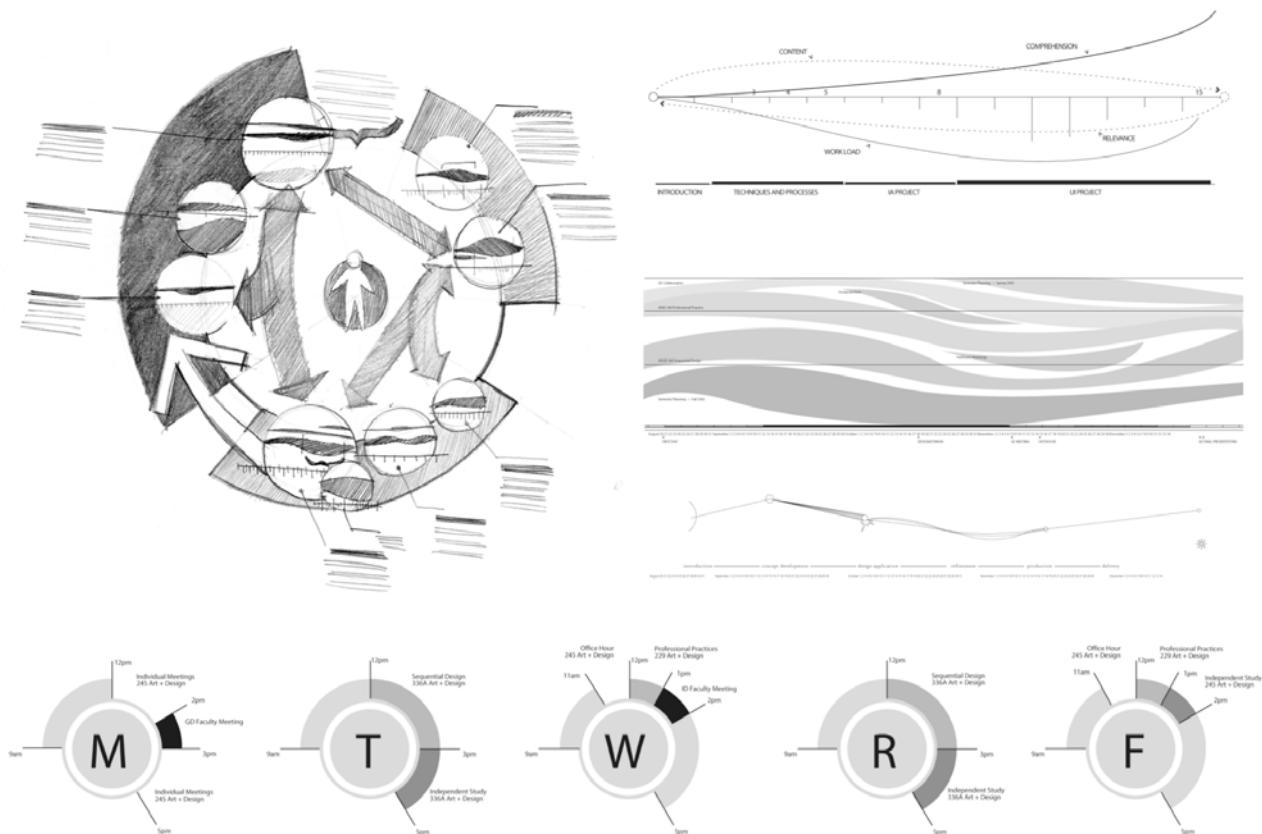


Figure 2: Visual representations of time and workflow can take many forms and yield different conceptual models.

Noted philosopher and author Alan R. White, wrote that “Visualizing and imagining are quite different concepts in the way that talking and thinking are different concepts. One can imagine, but not visualize, the non-visual, and even the non-perceptual, just as one can think, not only without talking in English but even talking at all” (White, 1990). While these terms may differ slightly in meaning, for the purposes of relating the role of imagination to the visualization process as designers traditionally know it, we can operate from a typical dictionary definition:

Imagination (*n*): the power of reproducing images stored in the memory under the suggestion of associated images or of recombining former experiences to create new images; a mental conception or creation, often a baseless or fanciful one; ability to meet and resolve difficulties, resourcefulness.

In reality, the activity of visualizing ideas is a combination of perception and imagination. When sketching, or transferring ideas from our heads to the page, we usually start with the concrete; then throughout the process of drawing, we relax constraints to abstract and emphasize what is important to us. As a highly purpose-specific process, representing knowledge in a visual manner must maintain a flexible and elastic process rather than some formalized representational

approach. Therefore, as such an ad hoc and ephemeral method of communication, designers can generate a number of small, temporary representations that can easily be picked up and utilized (Whitby, 1996). Hence, the drawing serves as a visual map of one’s thinking process – how they see the world and perceive what is important to them at a specific moment. Visualization then is the exercise of an ability to visually describe ideas based on knowledge gained from awareness, which may be easy or difficult to manifest on occasion or at which some may be better equipped than others. What one has visualized, as what one has in turn depicted, is not only some material object, existent or non-existent, but something whose identity depends not wholly on what it looks like but on what one intended it to be (White, 1990).

Del Coates stated in his recent book, *Watches Tell More Than Time*, that designers are equipped with artist’s antennae to sense what lies just out of range of normal sensibilities – designers (and artists) resonate with social and cultural forces that exist in the ether and remain undetectable by normal people. Designers build a mental catalogue of these factors that collectively serve as fodder for the next creative decision. Hanks and Belliston wrote in their popular book, *Rapid Viz*, “a mind filled with knowledge, experiences, and an acute observation of the surrounding world is more apt to bring forth creative ideas” (Hanks and Belliston, 1990). Hence a designer’s



Figure 3: This sketch captures ideas in a fluid and unscripted manner. Looking closer at the individual elements shows that each visualization builds upon its predecessors to create a seemingly linear evolution.

greatest asset is to work this memory bank into useable visual expressions of ideas, to challenge their imagination to change thinking, and to persuade viewers in the value of abstract approaches. In Figure 3, a sequence of hand-generated sketches illustrate alternatives in product configurations. At a size of 2' x 6', it creates an impressive statement about the flexibility in the development process.

Summary

Visualizations can be powerful and persuasive representations of ideas and data. A combination of hand generated and digital approaches can provide the flexibility to investigate alternative visual imagery, create parallel sequences and illustrate an idea in relation to a broader context. Carefully selecting the visual components of shapes, line, colors, and media can manage the inherent drifting towards complexity. As a statement, visualizations should illustrate the complex and abstract in a compellingly simple way. The common thread among the various manifestations of visualization is the structuring of the underlying communication and its ability to tell a story to the viewer.

References

- Coates, D. (2003): *Watches Tell More Than Time: Product Design, Information, and the Quest for Elegance*. New York, McGraw-Hill.
- Currie, G. and Ravenscroft, I. (2002): *Recreative Minds: Imagination in Philosophy and Psychology*. Oxford, Clarendon Press.
- Hanks, K. and Belliston, L. (1990): *Rapid Viz: A New Method for the Rapid Visualization of Ideas*. California, Crisp Publications.
- Narayanan, A. and Bodén, M. (1996): Representations for Changing One's Mind. In *Forms of Representation*. DONALD PETERSON (ed). Exeter, Intellect Books.
- Tabachneck-Schijf, H. and Simon, H. (1996): Alternative Representations of Instructional Material. In *Forms of Representation*. DONALD PETERSON (ed). Exeter, Intellect Books.
- Te Linde, J. (1983): Pictures and Words in Semantic Decisions. *Imagery, Memory and Cognition*. JOHN C. YUILLE (ed). New Jersey, Lawrence Erlbaum Associates.
- Whitby, B. (1996): Multiple Knowledge Representations: Maps and Aeronautical Navigation. In *Forms of Representation*. DONALD PETERSON (ed). Exeter, Intellect Books.
- White, A. (1990): *The Language of Imagination*. Oxford, Basil Blackwell Ltd.

Pattern Puzzle: A Metaphor for Visualizing Software Complexity Measures

Adam Ghandar, A.S.M Sajeed, Xiaodi Huang

School of Computer Science
University of New England
Armidale, New South Wales, Australia

{aghandar,sajeed, xhuang}@cs.une.edu.au

Abstract

Software systems have become increasingly complex over the years. Complexity metrics measures software complexity using real numbers. It is, however, hard to gain insight into different complexities by looking at these numbers. In this paper, we present a software complexity metaphor that uses a jigsaw puzzle. In particular, each component of a software system is modelled as a piece of a jigsaw puzzle. The problem complexity is modelled as a pattern on the surface of the piece, and the interconnection complexity as connectors between puzzle pieces. We demonstrate the benefits of this approach using case studies of the complexity measures of a real software system.

Keywords: Visualization, User Interface, Complexity Measures, Software Engineering

1 Introduction

Computer software has increased in complexity over the years. Software plays an important role in human life, from controlling advanced defense systems and power stations to remote education and games. Designing and maintaining complex software places an assortment of demands on all the actors in the design process. Managers must allocate human and material resources efficiently where they are needed; programmers and analysts need to form an understanding of the design of the software system. It is often necessary to present information about software systems to actors from different professional backgrounds.

We propose a method called the pattern puzzle metaphor that visualizes the measures of software complexity. The idea is based on the concept of a jig-saw puzzle – a set of pieces that fit together to create a larger picture. We implemented this method in a prototype to demonstrate the benefits of the metaphor and its use. In essence the pattern puzzle allows a user to form an understanding of at least two dimensions of complexity; that is, how they are related to each other, and how the complexity is distributed throughout a software system.

Copyright © 2006, Australian Computer Society, Inc. This paper appeared at *Asia Pacific Symposium on Information Visualization 2006 (APVIS 2006)*, Tokyo, Japan, February 2006. Conferences in Research and Practice in Information Technology, Vol. 60. K. Misue, K. Sugiyama and J. Tanaka, Eds. Reproduction for academic, not-for profit purposes permitted provided this text is included.

2 Complexity Measures

2.1 Problem Complexity

Some problem domains are inherently more complex than others. For instance, a software system for launching antiballistic missile in a defense establishment is likely to be much more complex than a software system that teaches the English alphabet. The Problem Complexity indicates complexities that arise from the problem domain. Furthermore, it is common that within the same application, some components have higher problem complexity than others. For example, in a search engine that is able to accept natural language questions, the processing component which analyses the natural language input is likely to be more complex than the user-interface component which displays the results to the user. The problem complexity of each component or a group of components (say, a package) is relevant to that of the whole system. As a result, managers may want to allocate additional resources and/or set up specific quality assurance procedures for components that are highly complex. However, as the number of components becomes large, it wouldn't be easy to get a good "picture" of the distribution of problem complexity among the different parts of the system or to locate those components that need a closer look. This is where visualization helps.

2.2 Design Complexity

Another class of complexity is the one arising from the way in which a software system is designed. For the same application, poorly designed software may exhibit higher complexity than a better designed one. Measures such as coupling, cohesion, weighted methods per classes, depth

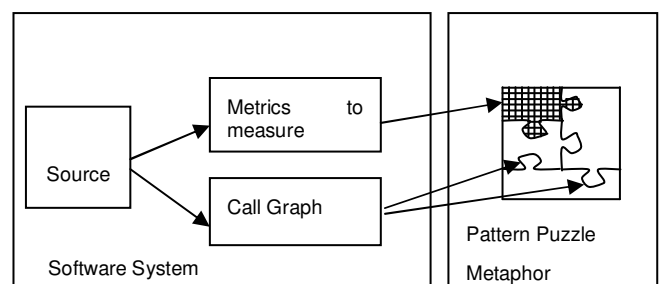


Figure 1. Overview of the Pattern Puzzle Metaphor: the call graph is the source of the pieces positions and their connections.

of inheritance tree etc. (Chidamber and Kemerer, 1994) are used in the literature to measure design complexity. Design complexity has implications both in developing reliable software as well as in maintenance of the system over its life cycle. Traditionally, graphs such as call-graphs (where nodes represent components and edges represent calls from one component to another) have been used to visualize aspects of design complexity.

3 Pattern Puzzle Metaphor

The Pattern Puzzle metaphor provides an integrated approach to visualizing complexity of software systems. It is based on the concept of a jig-saw puzzle – a set of pieces that fit together to create a larger picture. This larger picture is, in the same way that a real jig-saw puzzle is, the sum of its parts which are the pieces.

The metaphor has two main components: the *pieces* together with their individual pictures and the *joins* between the pieces along with the locations of pieces (see Figure 1). With reference to the concept of planar and retinal variables (Bertin, 1983), the dimensions of the pieces are planar and the patterns are a combination of several retinal variables.

The effectiveness of the metaphor is amplified by the careful choice of the surface pattern so that universal symbols (Fromm, 1976) are generated. If the pattern contains an intrinsic relationship with the attribute being visualized, then the Pattern Puzzle becomes a lucid and expressive way of looking at the software system.

3.1 Patterns

Surface patterns are a part of each puzzle piece and a target mapping for one or more attributes of the underlying software system. The patterns allow the viewing of at least one further attribute of a system in the same way that, say, the color would. However a pattern may be thought of as the amalgamation of a subset six retinal variables mentioned in (Bertin, 1983) (color, brightness, etc) in a coherent form where each variable complements the other. This means that a pattern, in some senses, can impart meaning by utilizing both the variables themselves and the relationships between them. Compared with approaches that use a single retinal variable such as the color to represent an additional attribute, patterns are a complimentary extension.

3.2 Connections

We elected to represent program complexity using an interesting class of patterns. They are symmetrical geometric shapes that are able to be generated in a computer graphics form using variables according to a method (Kaplan and Salesin, 2004). They are members of a class of designs termed Islamic Star Patterns. Figures 2 depicts a series of six pointed rosettes generated with the assistance of Kaplan's Taprats (Kaplan, 2000) library, and are a common motif used historically.

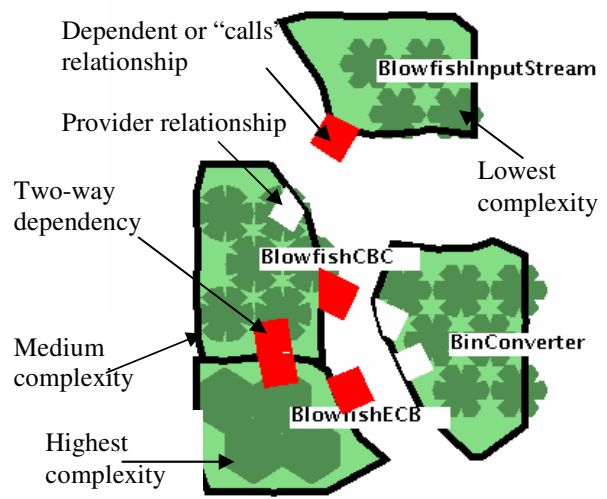


Figure 2. The meaning of symbols used in the implementation of the Pattern Puzzle metaphor.

The Pattern Puzzle pieces may be connected by puzzle joins to symbolize relational attributes in a software system. In fact, a number of problems arise in traditional graph visualization techniques when they are used to visualize software systems. The Pattern Puzzle metaphor attempts to lessen the confusion that arises from these problems by focusing on the puzzle pieces and by incorporating visual information about both the piece and its relationship to other pieces.

Puzzle joins are represented by protrusions of one piece's area into another. However, when creating a puzzle from a graph it is likely that a graph is not planar. In this case it is necessary to insert additional puzzle pieces. These pieces are drawn grey and the user is given a visual cue that the interconnection complexity is high.

A user is able to form an idea of the number of relationships in the data by briefly looking at the puzzle. This aspect of the visualization is enhanced by selecting a bright color for the links and by ensuring that they are of a reasonable size. Puzzles can be compared quickly with respect to the extent of interconnection among the components. Greater numbers of links indicate a higher interconnection complexity in the underlying system.

3.3 Hierarchy and User Interaction

The metaphor incorporates a hierarchical depiction of software systems in two ways.

First, patterns, which represent complex components in a puzzle that represents non-atomic components, (for example classes in a Java program can be divided into methods) imply that a complex structure lies 'below' the piece. The second level hierarchy allows users to 'dig' into the puzzle by selecting the pieces and to replace it with a new one in order to represent the subcomponents of a component, which is represented by the piece in the original puzzle. It is also possible to display several puzzles of different hierarchical levels simultaneously. The case study, illustrated in Figure 3, demonstrates the use of this method for visualizing hierarchy.

Users can also be encouraged to fit the puzzle pieces together themselves. This helps the users to build their mental maps of aspects of the software system incorporated in the visualization.

denoted by $Q = f_h(c)$; and (3) then find their mappings. That is, $D:Q \rightarrow V$. In particular, the mappings in our system include the followings:

- The complexity of software component

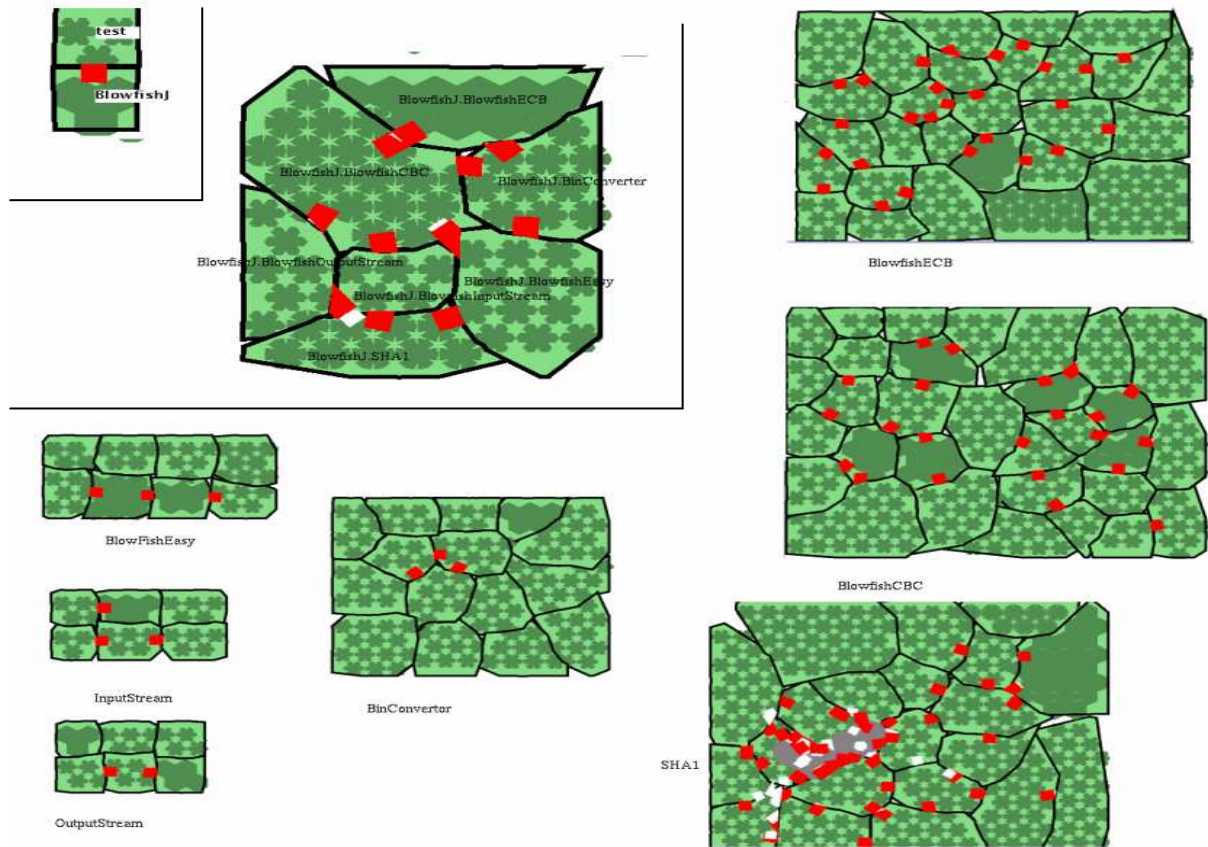


Figure 3. Visualizations of the BlowfishJ library by the Pattern Puzzle. Two packages: BlowfishJ and test, and their sub-components in the library are shown. Puzzles representing the three levels of hierarchy (package, class and method) are drawn. Note that the patterns in each puzzle indicate the relative value of complexity to those of other pieces in the same puzzle, instead of the overall value. The SHA1 pattern puzzle (bottom right) illustrates high interconnection complexity the diagram visually implies that the component could need more resources than the others.

3.4 Formal Description

In this Section, we formalize the problem of the Pattern Puzzle metaphor for visualizing software complexity measures.

A puzzle piece is the basic element in the visual vocabulary of the metaphor denoted by $V = \{\text{Pattern Puzzle}\} = \{\text{Puzzle piece, Connector}\}$. Each piece comprises of four elements: Puzzle piece = {surface picture, color, size, shape}. The relationships between Puzzle piece and Connector are represented as $\text{Connector} \subseteq \text{Puzzle piece} \times \text{Puzzle piece}$. The connection includes two types: protrusions or intrusions, namely $\text{Connector} = \{\text{intrusion, extrusion}\}$, both of which may be simultaneously present between two adjacent pieces.

In the design of the metaphor, we face three fundamental problems: (1) choose the software complexity measure H and a visual vocabulary V ; (2) calculate the complexity values associated with each components of software

Puzzle Piece, namely $f_m(c_j) \rightarrow \text{Puzzle piece}$ where j is a constant integer representing the structure level of components;

- $f_{cc}(m) \rightarrow \text{Pattern}$, which maps the complexity of class method m calculated by the cyclomatic complexity (CC) metric, into the Pattern;
- $R_j^c \rightarrow \text{Connector}$, which maps the relationships between components into the connections of puzzle pieces.

In addition, we illustrate the relationships between components R_j^c . For instance, in the method level, we have {Cyclomatic complexity, dependant upon, depended upon} {Pattern, intrusion, extrusion}. As an example, consider the following mapping:

$$\{f_{cc}(m), \{\text{true, false}\}, \{\text{true, false}\}\} \rightarrow \{ \text{Pattern}, \{\text{intrusion, extrusion}\} \}$$

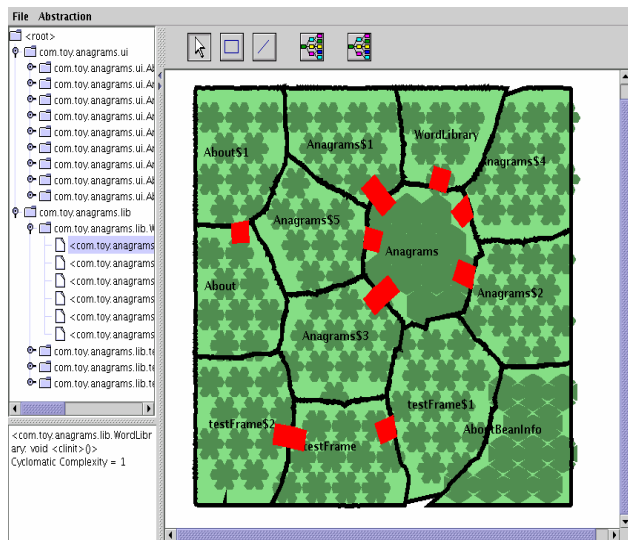


Figure 4. The user interface of the implementation of the Pattern Puzzle Metaphor prototype.

The true and false elements refer to whether or not a component is dependant on another specific component in a software system.

4 Case Study

We developed a tool which produces pattern puzzle representations of Java source files (see Figure 4). The tool is able to show three levels of hierarchy based on the package, class and method of a Java file.

We demonstrate the metaphor with the Blowfishj library (Hahn and Goeschl, 2004) which is of a reasonable size and exhibits large variations in complexity across components. It comprises 2 packages, 10 classes, and 2688 non-commented lines of code. Several classes have high interconnection complexity, and a number of methods have very high cyclomatic complexity (greater than 10). This is mainly because of the inherent problem complexity of the Blowfish algorithm (Schneier, 1993).

Figure 3 shows a visualization of the Blowfishj library in different levels using our Pattern Puzzle metaphor implementation. The three levels of hierarchy are shown. Complexity is indicated by the pattern on the surface of the pieces. Highest complexity is implied by solid blocks of colors. Elegant rosettes imply simplicity. Levels of complexity are implied by gradations between these extremes (see Figure 2). Note that the patterns are meaningful only within each puzzle and not relative over the whole of Figure 3.

A closer inspection of the puzzles in Figure 3, for example, reveals further properties of the software system. The method level puzzles of BlowfishCBC and BlowfishECB show simpler dependency patterns (indicated by the joins) than SHA1 that has fewer methods but with comparable or higher complexity. The method puzzle of SHA1 reveals a high level of interconnection complexity and a single method whose complexity dwarfs the others in the class. This would imply that the allocation of more resources (time, skills) to the implementation of the SHA1 class is necessary. It

may also signify that the design of this class should be looked at closely.

Many aspects of software can be perceived from the puzzle representation in addition to complexity. For example afferent and efferent couplings. At the package level the number of extrusions is equivalent to the number of efferent couplings and indicates packages independence. Conversely the number of intrusions indicates afferent couplings and indicates the package provides services to other components.

5 Conclusion

In this paper, we have presented our new approach called Pattern Puzzle metaphor for visualizing software complexity measures. One of the most successful metaphors used in computing is the desktop metaphor for the interfaces of operating systems. This success lies partly in the effective use of symbolism and user interaction which enable the user to make intuitive associations between the operating systems and the desktop metaphor without a great deal of learning. In other words, the desktop metaphor is almost a universal symbol (Fromm, 1976) for computer users. The Pattern Puzzle metaphor attempts to fulfil these criteria in the area of software complexity visualization using the jigsaw puzzle symbol.

ACKNOWLEDGEMENT

This research is supported by the Australian Research Council Discovery Grant, DP0209483.

REFERENCES

- Bertin, J. (1983): *Semiology of graphics*, University of Wisconsin Press.
- Chidamber, S. R. and Kemerer, C. F. (1994): A Metrics Suite for Object-Oriented Design *IEEE Transactions on Software Engineering*, **20**, 476-493.
- Fromm, E. (1976): *The Forgotten Language; An Introduction to the Understanding of Dreams, Fairy Tales, and Myths.*, Henry Holt & Co.
- Hahn, M. and Goeschl, S. (2004): Blowfishj. Blowfish encryption for java
<http://blowfishj.sourceforge.net/index.html>.
- Kaplan, C. S. (2000): Taprats
www.cgl.uwaterloo.ca/~csk/washington/taprats/.
- Kaplan, C. S. and Salesin, D. H. (2004): Islamic Star Patterns in Absolute Geometry *ACM Transactions on Graphics*, **23**, 97-119.
- Schneier, B. (1993): Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish). In *Fast Software Encryption* Springer-Verlag, Cambridge Security Workshop Proceedings. 191-204.

Author Index

- Baskinger, Mark, 217
Bekos, Michael A., 15
Bui, Michael, 35
- Cheon, Suh Hyun, 143
Clement, Christopher, 85
- de Albuquerque, Eduardo Simões, 117
do Nascimento, Hugo A. Dantas, 117
Dwyer, Tim, 189
- Eades, Peter, 199, 207
- Ferreira, Joelma de Moura, 117
- Ghandar, Adam, 221
Gudmundsson, Joachim, 67
- Hansaki, Tomoyuki, 147
Hayashi, Yukio, 139
Hideshima, Yusuke, 131
Ho, TuBao, 59
Hong, Seok-Hee, 189, 199, 207
Huang, Weidong, 199, 207
Huang, Xiaodi, 221
- Iga, Soichiro, 161
Ikeda, Katsuhiko, 49
Ishihara, Masaki, 153
Itoh, Takayuki, 23
- Jianu, Radu, 85
Jung, Seong Dae, 63
Jung, Soon Ki, 63
- Kaufmann, Michael, 15
Kawasaki, Saori, 59
Kim, Bok Dong, 63
Klukas, Christian, 39
Kobayashi, Takumi, 103
Koike, Hideki, 131
Koo, Sang Ok, 63
Koo, SungJa, 63
Koschützki, Dirk, 189
Kurokawa, Taketo, 9
Kwon, Hyok Don, 63
- Lee, Minho, 63
- Lee, Yi-Yi, 179
Lin, Chun-Cheng, 179
Lowe, Nick, 35
- Müller Wolfgang, 121
Ma, Kwan-Liu, 3, 93
Merrick, Damian, 67
Misue, Kazuo, iii, 49, 103, 109, 147, 153, 169
- Nakazono, Nagayoshi, 109
Nam, Ki-Chol, 217
Nguyen, DucDung, 59
Nocke, Thomas, 121
- Ogawa, Michael, 93
Omote, Hiroki, 89
Ong, Kenneth, 31
- Potika, Katerina, 15
- Rho, YongWoo, 63
Rusu, Adrian, 85
- Sajeev, A.S.M., 221
Santiago, Confesor, 85
Schreiber, Falk, 39, 189
Schumann, Heidrun, 121
Schwöebbermeyer, Henning, 39
Shen, Zeqian, 93
Shinnishi, Makoto, 161
Shizuki, Buntarou, 103, 147
Subramaniam, Ganesan, 31
Sugiyama, Kozo, 49, 89
Symvonis, Antonios, 15
- Takatsuka, Masahiro, 35, 77
Tanaka, Jiro, iii, 103, 109, 147, 153
Teoh, Soon Tee, 93
Tomita, Shinji, 139
- Watanabe, Isamu, 49
Wu, Yingxin, 77
- Xu, Kai, 189
- Yamashita, Fumiyoshi, 23
Yen, Hsu-Chun, 179
Yoo, Hee Yong, 143

Recent Volumes in the CRPIT Series

ISSN 1445-1336

Listed below are some of the latest volumes published in the ACS Series *Conferences in Research and Practice in Information Technology*. The full text of most papers (in either PDF or Postscript format) is available at the series website <http://crpit.com>.

Volume 41 - Theory of Computing 2005

Edited by Mike Atkinson, *University of Otago, New Zealand* and Frank Dehne, *Griffith University, Australia*. January, 2005. 1-920-68223-6.

Contains the papers presented at the Eleventh Computing: The Australasian Theory Symposium (CATS2005), Newcastle, NSW, Australia, January/February 2005.

Volume 42 - Computing Education 2005

Edited by Alison L. Young, *UNITEC, New Zealand* and Denise Tolhurst, *University of New South Wales, Australia*. January, 2005. 1-920-68224-4.

Contains the papers presented at the Seventh Australasian Computing Education Conference (ACE2005), Newcastle, NSW, Australia, January/February 2005.

Volume 43 - Conceptual Modelling 2005

Edited by Sven Hartmann, *Massey University, New Zealand* and Markus Stumptner, *University of South Australia*. January, 2005. 1-920-68225-2.

Contains the papers presented at the Second Asia-Pacific Conference on Conceptual Modelling (APCCM2005), Newcastle, NSW, Australia, January/February 2005.

Volume 44 - ACSW Frontiers 2005

Edited by Rajkumar Buyya, *University of Melbourne*, Paul Coddington, *University of Adelaide*, Paul Montague, *Motorola Australia Software Centre*, Rei Safavi-Naini, *University of Wollongong*, Nicholas Sheppard, *University of Wollongong* and Andrew Wendelborn, *University of Adelaide*. January, 2005. 1-920-68226-0.

Contains the papers presented at the Australasian Workshop on Grid Computing and e-Research (AusGrid 2005) and the Third Australasian Information Security Workshop (AISW 2005), Newcastle, NSW, Australia, January/February 2005.

Volume 45 - Information Visualisation 2005

Edited by Seok-Hee Hong, *NICTA, Australia*. January, 2005. 1-920-68227-9.

Contains the papers presented at the Asia-Pacific Symposium on Information Visualisation, APVis.au, Sydney, Australia, January 2005.

Volume 46 - ICT in Education

Edited by Graham Low, *University of New South Wales, Australia*. October, 2005. 1-920-68228-7.

Contains selected refereed papers presented at the South East Asia Regional Computer Confederation (SEARCC) 2005: ICT Building Bridges Conference, Sydney, Australia, September 2005.

Volume 47 - Safety Critical Systems and Software 2004

Edited by Tony Cant, *University of Queensland*. March, 2005. 1-920-68229-5.

Contains all papers presented at the Ninth Australian Workshop on Safety-Related Programmable Systems, (SCS2004), Brisbane, Australia, October 2004.

Volume 48 - Computer Science 2006

Edited by Vladimir Estivill-Castro, *Griffith University* and Gillian Dobbie, *University of Auckland, New Zealand*. January, 2006. 1-920-68230-9.

Contains the papers presented at the Twenty-Ninth Australasian Computer Science Conference (ACSC2006), Hobart, Tasmania, Australia, January 2006.

Volume 49 - Database Technologies 2006

Edited by Gillian Dobbie, *University of Auckland, New Zealand* and James Bailey, *University of Melbourne*. January, 2006. 1-920-68231-7.

Contains the papers presented at the Seventeenth Australasian Database Conference (ADC2006), Hobart, Tasmania, Australia, January 2006.

Volume 50 - User Interfaces 2006

Edited by Wayne Piekarski, *University of South Australia*. January, 2006. 1-920-68232-5.

Contains the papers presented at the Seventh Australasian User Interface Conference (AUIC2006), Hobart, Tasmania, Australia, January 2006.

Volume 51 - Theory of Computing 2006

Edited by Barry Jay, *UTS, Australia* and Joachim Gudmundsson, *NICTA, Australia*. January, 2006. 1-920-68233-3.

Contains the papers presented at the Twelfth Computing: The Australasian Theory Symposium (CATS2006), Hobart, Tasmania, Australia, January 2006.

Volume 52 - Computing Education 2006

Edited by Denise Tolhurst, *University of New South Wales, Australia* and Samuel Mann, *Otago Polytechnic, Otago, New Zealand*. January, 2006. 1-920-68234-1.

Contains the papers presented at the Eighth Australasian Computing Education Conference (ACE2006), Hobart, Tasmania, Australia, January 2006.

Volume 53 - Conceptual Modelling 2006

Edited by Markus Stumptner, *University of South Australia*, Sven Hartmann, *Massey University, New Zealand* and Yasushi Kiyoki, *Keio University, Japan*. January, 2006. 1-920-68235-X.

Contains the papers presented at the Third Asia-Pacific Conference on Conceptual Modelling (APCCM2006), Hobart, Tasmania, Australia, January 2006.

Volume 54 - ACSW Frontiers 2006

Edited by Rajkumar Buyya, *University of Melbourne*, Tianchi Ma, *University of Melbourne*, Rei Safavi-Naini, *University of Wollongong*, Chris Steketee, *University of South Australia* and Willy Susilo, *University of Wollongong*. January, 2006. 1-920-68236-8.

Contains the papers presented at the Fourth Australasian Workshop on Grid Computing and e-Research (AusGrid 2006) and the Fourth Australasian Information Security Workshop (AISW 2006), Hobart, Tasmania, Australia, January 2006.

Volume 55 - Safety Critical Systems and Software 2005

Edited by Tony Cant, *University of Queensland*. Late 2005. 1-920-68237-6.

Contains all papers presented at the 10th Australian Workshop on Safety Related Programmable Systems, August 2005, Sydney, Australia.

Volume 56 - Visual Information Processing 2005

Edited by Hong Yan, *City University of Hong Kong*, Jesse Jin, *University of Newcastle, Australia*, Zhiqiang Liu, *City University of Hong Kong* and Daniel Yeung, *Hong Kong Polytechnic University*. Late 2005. 1-920-68238-4.

Contains papers from the Asia-Pacific Workshop on Visual Information Processing (VIP2005), Hong Kong, December 2005.

Volume 57 - Multimodal User Interaction 2005

Edited by Fang Chen and Julien Epps, *National ICT Australia*. December, 2005. 1-920-68239-2.

Contains the proceedings of the Multimodal User Interaction Workshop 2005, NICTA-HCSNet, Sydney, Australia, 13-14 September 2005.

Volume 58 - Advances in Ontologies 2005

Edited by Thomas Meyer, *National ICT Australia, Sydney* and Mehmet Orgun, *Macquarie University*. December, 2005. 1-920-68240-6.

Contains the proceedings of the Australasian Ontology Workshop (AOW 2005), Sydney, Australia, 6 December 2005.