Adding Filtering to Geometric Distortion to Visualize a Clustered Graph on Small Screens

Wanchun Li and Masahiro Takatsuka

ViSLAB, School of Information Technologies
The University of Sydney, NSW, 2006, Australia

lpaul@it.usyd.edu.au, masa@vislab.usyd.edu.au

Abstract

Presenting large amounts of information in a limited screen space is a significant challenge in the field of Information Visualization. With the rapid development and growing use of small handheld devices such as PDAs this issue has become more important. Many Focus+Context techniques have been developed to address it but very few of them would effectively aid visualization applications for small handheld devices.

In this paper we propose a new approach for visualizing a clustered graph by adding filtering to geometric distortion. Different from most existing techniques, our approach focuses on contextual information. Beginning with the assumption that not all contextual information is necessary to visualize a user's interests, we propose an approach for filtering out some "*useless*" context to compensate for the small screen size.

Samples of applying the proposed approach to visualization of a rooted tree and a clustered graph on a small screen are presented. The limitations are also discussed.

Keywords: small screen display, graph visualization, clustered graph, Focus+Context, fisheye, PDA.

1 Introduction

One of the major objectives of Information Visualization is to effectively and clearly convey large amounts of information on a relatively small screen. This problem is exacerbated by the rapid development and growing use of small handheld devices such as PDAs (Personal Digital Assistants).

There are many visualization techniques for addressing the effective use of screen space. Zooming and panning are traditional tools of information visualization. As mentioned by Card, Mackinlay, and Shneiderman (1999), they are simple to implement and understand, and can provide rapid access to details of the drawing that are too large to fit in a window. However, it does not integrate a user's focus and global context very well. The Focus+Context technique is a good alternative. In any case of visualization or user interface interaction, a user's attention is usually focussed on a particular area. The Focus+Context technique allocates more display resources to the region of the user's interest and maintains contextual information at the same time.

Focus+Context techniques can be categorized as non-graph-oriented and graph-oriented depending on whether or not an approach is specifically geared towards graph visualization (Noik 1994). They may also be classified as distortion-oriented if view is geometrically distorted. Leung (1994) provided a review and taxonomy of distortion-oriented techniques. Distortion-oriented techniques can be further grouped into linear distortion and non-linear distortion techniques according to an underlying mathematic transformation formulation. Keahey (1996) discussed the non-linear magnification transformations.

There are many exciting results of early work using Focus+Context techniques such as: Bifocal Lens (Spence and Appperley 1982), Fisheye View (Furnas 1986), Perspective Wall (Mackinlay, Robertson, and Card 1991), Cone Trees (Robertson, Mackinlay, and Card 1991), Graphical Fisheye View (Sarkar and Brown 1992), Table Lens (Rao and Card 1994) and Hyperbolic Tree (Lamping and Rao 1996). Some of these are aimed towards data visualization, while some are intended more for structure visualization.

Unfortunately, very few of these techniques would aid visualization applications for small handheld devices. The display resource of many small handheld devices is very limited because both the screen size and the number of pixels available are significantly smaller than desktop systems. The result is that most of these techniques could not be directly applied to small handheld devices.

New approaches, therefore, still need to be developed in order to create visual presentations that are suitable for small handheld devices. We believe that one important principle of a small screen display approach is that it cannot compensate for the small size at the expense of aesthetics. So the objectives of a new approach should be: 1) to use the screen more efficiently, and 2) that the presentation is aesthetically pleasing.

With the assumption that not all contextual information is necessary to visualize a user's interests, we concentrate on finding rules to filter out some "*useless*" contextual information. Our basic idea is to compensate for the small screen size by computing the contextual

information in order to 1) allocate enough space to the focus area to display details of the user's interest, 2) reduce the amount of information to be suitable for a small screen.

Based on this idea, we present a new approach for visualizing a clustered graph by adding filtering to the geometric distortion of a graph. In this method, the "degree of interest" (DOI) of Logical Fisheye View (LFV) is dynamically recomputed according to structural information as well as the results of geometric distortion. LFV controls the contextual filtering based on the recomputed DOIs.

The approach finds a way to convert the geometric information to structural expressions by adapting DOI of LFV. The contribution of this approach is that it can modify existing visualization techniques or graph layout algorithms that use hierarchical or clustered graphs so that this new hybrid technique can be applied.

2 Background

2.1 Logical Fisheye View (LFV)

LFV was initially proposed by Furnas (1986). The idea is that during any given interaction, users are not equally interested in all of the data. Furnas proposed the DOI function to represent user's interest in the data. The data whose DOI are lower than a given threshold value *k* will be filtered out, and the data with high DOI values (due to being depicted at the focused area) can be efficiently displayed in the confined space. In a tree structure, the DOI of a given vertex is the negative sum of its depth and distance of the vertex to the focal vertex as:

$$\mathrm{DOI}(x \mid .) = -(d(r,x) + d(.,x)), \qquad (1)$$

where

- '.' is the focal vertex
- $DOI(x \mid .)$ is the DOI value of a given vertex x
- $d(r,x)$ is the distance from root to x, i.e. the depth of x
- $d(.,x)$ is the distance of the vertex x to the focal vertex

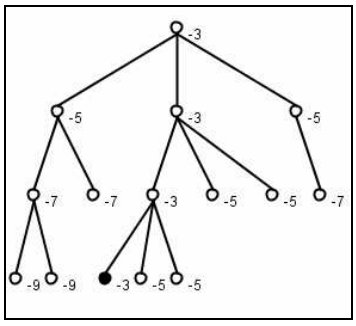
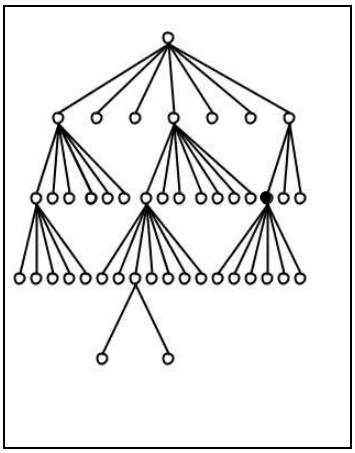


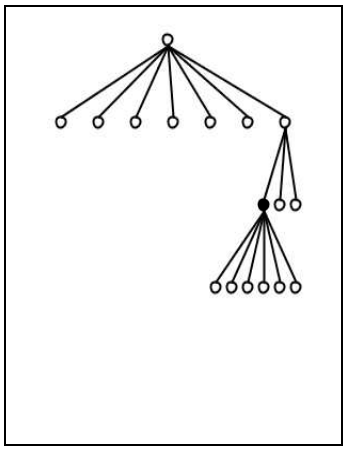
Figure 1 DOI of a tree. The solid vertex is the focus.

Figure 1 shows the DOI value of all vertices in a tree when the solid vertex is chosen as the focal vertex.

Figure 2 shows a LFV of a tree with 45 vertices. As the threshold value *k* becomes smaller, more structural information will be shown. From Figure 2 we can find that the LFV can clearly present the structural contextual information of the focus.



(a)



(b)

Figure 2 Logical Fisheye View of a tree. (a) Original view of the tree. (b) The logical fisheye view, k = -6, and the solid vertex is the focus.

However, there are some drawbacks of LFV when applied to graph visualization on a small screen. First, it loses some information of physically neighbouring vertices. For example, in Figure 2 (a), we can find that there are some vertices close to the left of the focal vertex. But all of the left side neighbour vertices are filtered out and only its sibling is presented at that level in 2(b). Moreover, the focal vertex has the same size as other vertices so that it is hard for a user to get the detail of the focus. Finally, the space is not efficiently used. As we can see in Figure 2(b) there is a big blank space after the thresholding.

Therefore, LFV is a powerful tool for representing the structure of data, but has limitations in utilizing the screen space.

2.2 Geometric Distortion (GD)

Generally, GD is used with graph-oriented techniques. The essence of these techniques is the concurrent

presentation of local detail together with global context at reduced magnification (Leung1994). The distorted view is computed by applying a transformation function to a presentation. The transformation function can be either nonlinear or linear, and the techniques are grouped into nonlinear distortion and linear distortion correspondingly. Since our implementation uses the Geometric Fisheye View (GFV), we discuss GFV as a sample of GD in the sequel.

GFV is a non-linear distortion-oriented graphical visualization technique. It imitates the fisheye lens effect by magnifying and reducing the focused and peripheral contexts respectively. Its transformation function is:

$$T(D) = \frac{(1+d)\times D}{(d\times D+1)}, \qquad (2)$$

$$D \in [-1,1], d > 0.$$

where d is called the distortion factor, and D is the normalized distance of a point to the focus.

$$D = v_d / f_d \qquad (3)$$

where v_d is the distance of a given point to the focal point, and f_d is the distance of focal point to the boundary.

The larger *d* is, the more space the magnified area will be allocated. The larger D is, the smaller the transformed vertex size will be. Figure 3 is the GFV of the tree in Figure 2(a). Sarkar and Brown 1992 have a detailed discussion of the related mathematical issues.

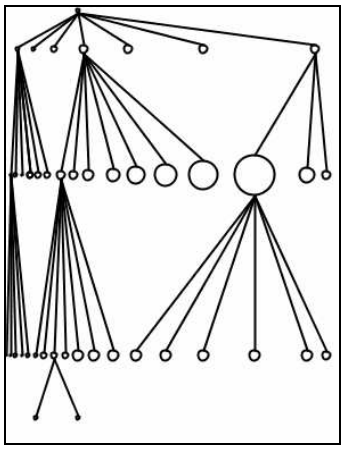


Figure 3 The Geometric Fisheye View of the tree in Figure 2(a), *d* = 3.0.

GFV overcomes the shortcomings of the LFV when applied to visualization on small screens. First, it allocates more space to the focus and the nearby vertices. The space is used more efficiently with less blank area. Although the presentation of a graph is distorted, it doesn't have much affect on user's navigation and interaction capabilities. (Carl Gutwin and Amy Skopik, 2003)

However, it does have disadvantages. The first problem is the readability in the area of reduced magnification. As we can see in Figure 3, there are too many tiny vertices to read in the area close to the left boundary of the screen. They don't present any useful information, but just increase the visual complexity. In turn, an aesthetic problem is raised in that the angular resolution is too small.

2.3 Hybrid Techniques

Both LFV and GD can be added as a separate processing step on a specific graph layout to form a hybrid approach that produces a new layout output. The hybrid technique allocates more space to the user's focus while keeping the context information based on the specific layout. Therefore, as Leung (1994) mentioned, the hybrid approach "*is potentially powerful, and greater research effort should be focused on exploring the application domains for such hybrid approaches*."

There are now many hybrid techniques such as Continuous Zoom (Bartram *et al* 1995) and DOI tree (Card S.K., Nation D. 2002), etc.

2.4 Clustered Graphs

There are various definitions of clustered and hierarchical graphs (Lengauer and Wanke, 1988; Eades and Feng, 1996; Sugiyama and Misue, 1991), but in this study we use Clustered Graphs as introduced by Eades and Feng (1996): A clustered graph is a graph with recursive clustering structures over the vertices. A clustered graph $C = (G,T)$ consists of an undirected graph $G = (V,E)$ and a cluster tree T such that the leaves of T are exactly the vertices of G. Each node of T represents a clustered vertex of G. As discussed by Eades and Huang (2000), a clustered graph has some elementary operations such as *close cluster*, *create cluster*, and *open cluster*.

Many graph visualization techniques use two basic operations: filtering and searching/navigating. These operations can be naturally mapped to the operations of a clustered graph (see Figure 4). Therefore, clustered graphs are an important data structure for graph visualization since it will be easy to visualize a given dataset in a small screen if we can discover the clustered relationships within it.

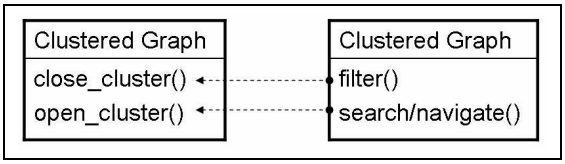
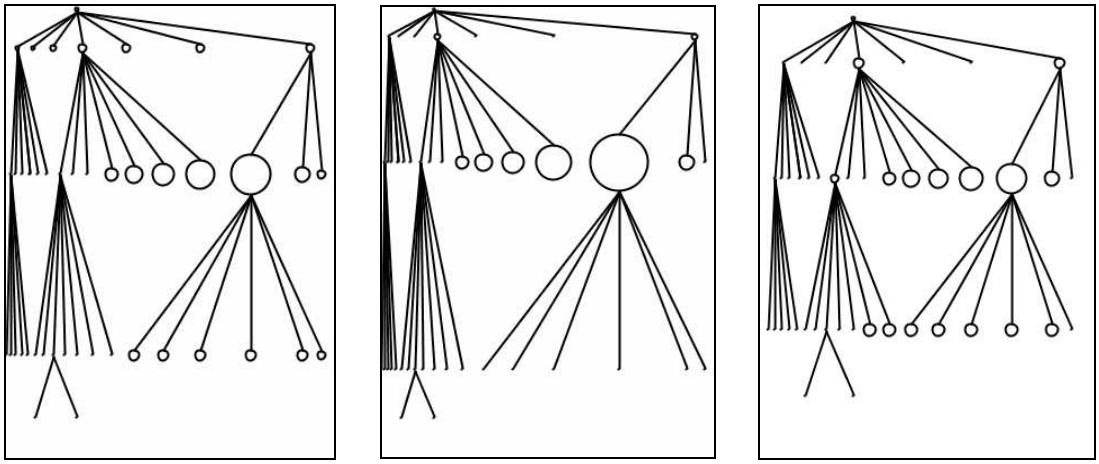


Figure 4 Mapping between a clustered graph and a visualization system.

A tree is the simplest clustered graph in which there is no edge between vertices in different clusters. In Section 3, we first introduce our approach by applying it to a rooted tree and discuss related issues of the



(a) (b) (c)

Figure 5 *Small screen view* of 2(a). (a) GFV d = 3.0, LFV k= -4. (b) d = 5.0, k = -2. (c) d = 2.0, k = -2.

approach. Then we apply this approach to visualizing a general clustered graph in Section 4.

3 Proposed Approach

Our approach is a hybrid technique. It visualizes a clustered graph on small screens by adding a filtering process to GD of a layout of a clustered graph.

In this study, we call the original presentation of a graph an *initial view* and the results of this approach as *small screen view.*

3.1 Small screen view of a tree

There are three steps to visualizing a tree using the new approach: GD, recomputation of DOI, and contextual filtering.

First, distort the tree by a specific GD. In our sample, we use GFV.

Second, specify a threshold value of LFV as k, and recompute DOIs of all vertices by the following rules:

1. If a DOI is not smaller than k, keep it
2. Else
   a. Set DOI as 1 if the vertex is magnified
   b. Set DOI as 1 if the vertex is an ancestor of a magnified vertex
   c. Set DOI of other vertices as $-\infty$

Third, filter out vertices whose DOIs are lower than the specified threshold value k.

In other words, a vertex will be filtered out if it meets all the following conditions:

1. Demagnified (reduced magnification)
2. DOI is lower than k
3. Not an ancestor of any kept vertices

Figure 5(a) shows the small screen view of Figure 2(a). After applying geometric transformation to Figure 2(a), the four neighbouring vertices on the left-hand side of the focal vertex and the first right sibling of the focal vertex are magnified, so their DOIs and their ancestors' DOI are set as 1. The children of the focal vertex, the siblings of the focal vertex and the siblings of ancestors of the focal vertex are kept because their DOIs are bigger than k. The DOIs of other vertices are set as $-\infty$.

Note that in Figure 5(c) the leftmost vertex at the second level (hereafter, we refer to the root as the zero level) is in a demagnified area and its DOI value, -6, is smaller than the threshold value k = -2. However, this vertex is kept because it is the ancestor of the vertices that are magnified at the third level (the left two vertices). This vertex's parent is also kept. The reason for keeping the direct ancestor of a magnified vertex is that the ancestor can offer contextual information for the magnified vertex and also for the focal vertex.

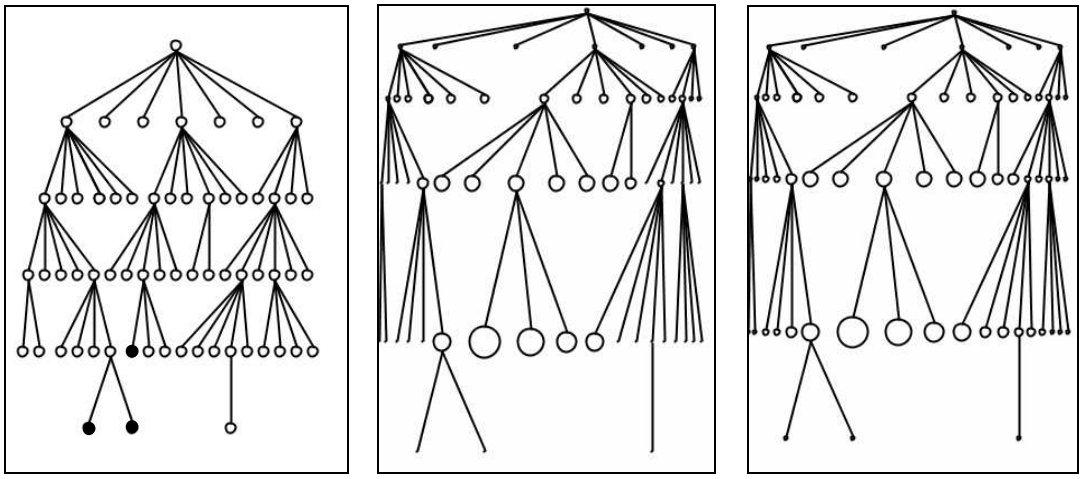
3.2 Discussion

3.2.1 Preservation of contextual information

Comparing figure 5(a) with figure 3 we can see that this approach uses the space efficiently. In Figure 5(a), both the structural context and the geometric context are displayed, and the focal vertex is enlarged in the magnified area.

It can also be seen that some "useless" vertices are filtered out in Figure 5(a). Because their sizes are too small, those vertices close to the left boundary of the screen in Figure 3 cannot show much information, but just increase the visual complexity. After filtering out such vertices, the drawing (see Figure 5(a)) has less visual elements than Figure 3, but no less of the details of the focal vertex and contextual information. Furthermore, because of the decrease of visual complexity, the user can more concentrate on the focal vertex.

Our approach allows users to control the visual appearance by changing the distortion factor d and the threshold value k. By adjusting the two factors, users get different contextual data and a different level of detail in the focus area (see Figure 4 (b) and 4(c)). Hence, users



(a) (b) (c)

Figure 6(a) A graph with 64 nodes. (b) Small screen view of (a), d = 2.0, k = -6. (c) Small screen view of (a), d=2, k=-14. The small screen view becomes a pure GFV.

can correctly emphasize what they are interested in and de-emphasize what they are not. These controls also change the visual complexity to improve the aesthetic of the small screen view.

3.2.2 Modification of existing visualization techniques or graph layout algorithms

One contribution of this approach is that it can be applied to any visualization and graph layout techniques as long as they employ hierarchical or clustered data. The approach can be added as an independent process to refine the presentation. The new output is the *small screen view* of the original presentation. Thus, the modified visualization/graph layout can be applied to visualization on small screens.

In our current implementation, GD is computed using a GFV transformation function. But different kinds of magnification techniques might be needed by different layout algorithms or visualization techniques. For example, for an orthogonal layout algorithm or boxing related visualization technique such as "*inclusion tree*" (Eades, Lin, and Lin, 1993) or "*flip zooming*" (Holmquista and Ahlbergb, 1997), a linear magnification such as a bi-focal lens would be better than a non-linear magnification such as GFV, because the distorted view of the linear transformation will not destroy the orthogonal or rectangle presentation. On the other hand, multiple foci or more complicated transformations might be necessary for some special tasks. Whatever GD is used, there will be some magnified vertices and demagnified vertices, and the DOI recomputation of the second step can be applied.

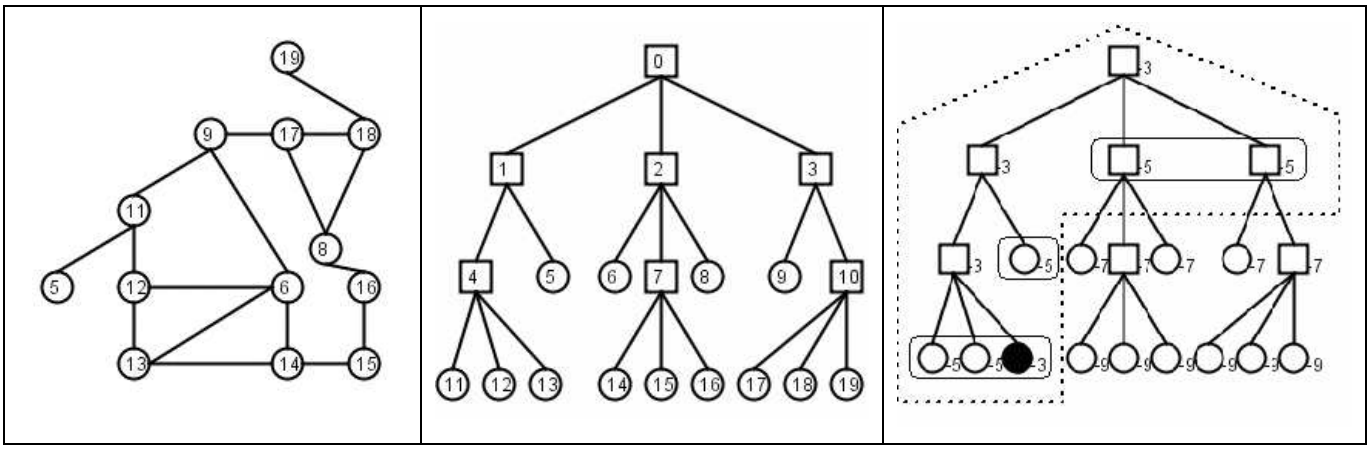
This approach can also be applied to visualize an image. When a point on an image is selected as the focus, we can decompose the image to some segments based on the geometric distance or can have images of different levels of detail (LOD). A tree or a hierarchical graph can be built to represent these segments or LOD. Then our approach can be applied to create a *small screen view* for this image.

3.2.3 Current Problems

The proposed approach still needs some improvement in modification of its filtering rule. The current filtering rule is to threshold those demagnified vertices that have no magnified direct descendants and whose original DOI values are lower than the specified threshold value *k*. However, in some situations, some relatively important context could be filtered out, while relatively less important context is displayed.

For example, Figure 6(b) is the *small screen view* of Figure 6(a). In 6(a), there are two solid vertices at the bottom level left of the focal vertex. In some situations, these two vertices can offer more contextual information of the focal vertex than some vertices in the second level do. But all the second level vertices are displayed, while the two vertices are filtered out. Unless *k* is set to –14(see Figure 6(c)), the lowest logical DOI in the tree, the two vertices at the bottom level won't be displayed by using the current filtering rule. However, Figure 6(c) is a pure GFV, no structural factor affects this view. In this case, the current rules failed to create an effective *small screen view.*

This problem is caused by the GFV transformation function. We know from (2) that the normalized distance *D* is another factor of distortion computation. From (3) we can discover that when the focal point/vertex is close to the boundary, *D* of a certain point/vertex could be large. Consequently, the point/vertex will be demagnified although the physical distance of the point/vertex to the focal point/vertex is small.



(a) (b) (c)

Figure 7 (a) A clustered graph. (b) The cluster tree of (a), The leaves of the tree are the vertices of (a). (c) The vertices circled by rectangles are *boundary nodes* of (b), when solid vertex is focus and *k*=-5.

4 Visualization of a clustered graph

4.1 Filtering rules for a clustered graph

The same three steps are applied to visualize a clustered graph $C = (G,T)$: GD, recomputation of DOI, and filtering. However, there is are differences from the tree visualization, the GD is applied to G , whilst recomputation of DOI and filtering are applied to T . Another difference is that there is a *create cluster* or *close cluster* operation in the filtering step.

Before further discussion, we introduce the *boundary node* of a clustered graph $C = (G,T)$. When a threshold value *k* of LFV is specified to T , we can get a truncated tree T_c that is composed of all the nodes of T where the DOIs are not smaller than *k*. We call a leaf node of T_c a *boundary node*.

Figure 7(a) is a clustered graph, and Figure 7(b) is its cluster tree. When the solid node in Figure 7(c) is selected as focal node and threshold value k=-5, T_c is the part inside the dashed border of T . The *boundary nodes* are the nodes circled by the rectangles (see Figure 6(c)).

Now we can describe the creation of the *small screen view* of C .

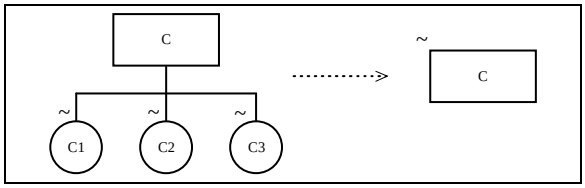
First, distort the *initial view* of G by a specific GD. In our sample, we use GFV. Note that only the leaves of T will be distorted because no super vertex of T will be displayed in its *initial view*.

Second, specify a threshold value *k* of LFV of T , and recompute DOIs of the leaves of T by the following rules:

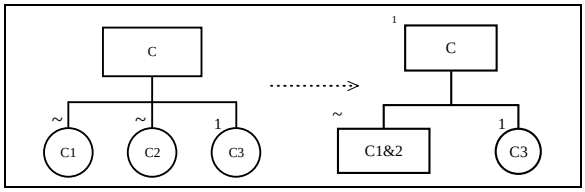
1. If the DOI of a leaf is not smaller than *k*, keep it
2. Else
   a. Set DOIs of magnified leaves as 1
   b. Set DOIs of demagnified leaves as $-\infty$

Third, iterate the following procedure from the leaf vertex of T upwards in order to create a new cluster tree Ts for the *small screen view*. For each iteration, it stops when it meets a *boundary node*

1. If the DOIs of a leaf and all its siblings are $-\infty$
   a. Cluster all these vertices to its parent
   b. Set DOI of its parent as $-\infty$
2. Else if the DOIs of some siblings' are $-\infty$, but others' are not
   a. Create a new vertex for these vertices which DOIs are $-\infty$
   b. Set DOI of the new created vertex as $-\infty$
   c. Keep the DOI of a vertex if it is not $-\infty$
   d. Keep the DOI of the super vertex



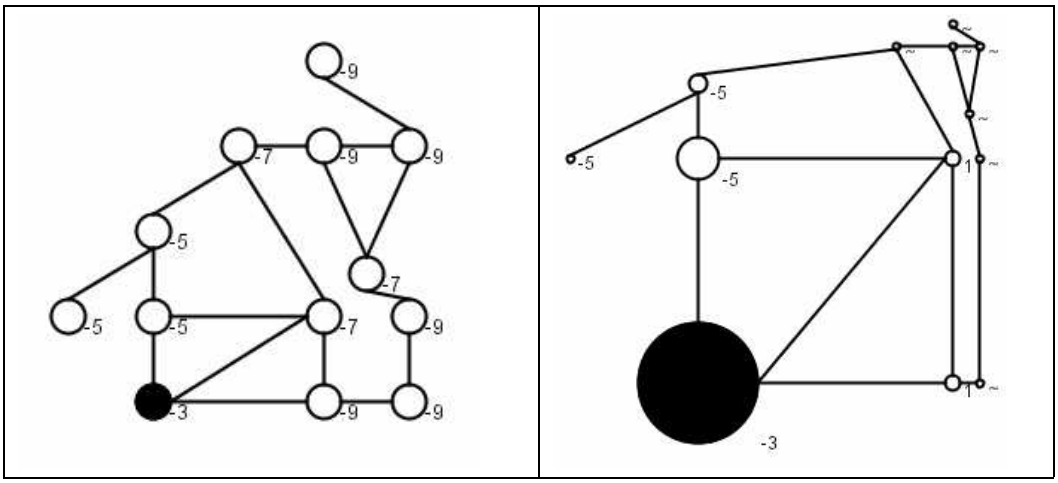
(a)[1]



(b)

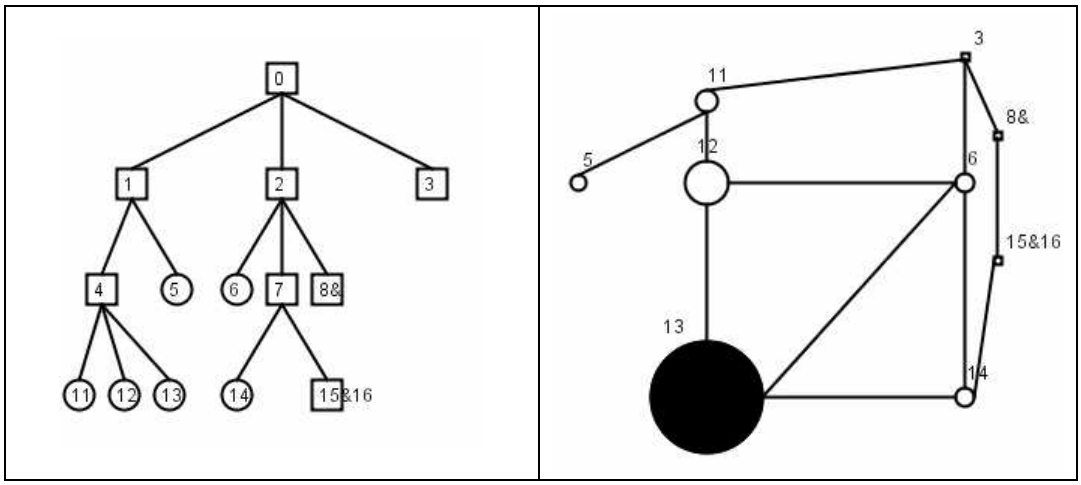
Figure 8 Operation *of close cluster and create cluster*. (a) Closing cluster. (b)Creating cluster.

[1] Hereafter, "~" is used to represent $-\infty$ in pictures.



(a) (b)

Figure 9 DOI recomputation. (a) The original DOI values when the solid vertex is focus. (b) The DOI values after geometric distortion *d*=4, and structural thresholding *k*=-5.



(a) (b)

Figure 10 *Small screen view* and its cluster tree. (a) The cluster tree of the *small screen view*. All children of Vertex 3 are closed, and some children of vertex 7 are closed. (b) The leaves of (a) are the vertices of (b).

3. Else do nothing, there is no vertex of $-\infty$ DOI

The created *small screen view* will be composed of the leaves of Ts .

Figure 8(a) describes the case 1 in the third step. The super vertex C has three leaf children C1, C2, and C3. All the DOIs of the three leaves are $-\infty$. All three leaves are clustered and C will become a leaf vertex, the DOI of which is recomputed to $-\infty$. This is an operation of *closing cluster*.

Figure 8(b) describes case 2. DOIs of C1 and C2 are $-\infty$, while that of C3 is 1. A new vertex, C1&2, is created to cluster C1 and C2. The DOI of C1&2 is $-\infty$. C3 does not change its DOI value which is still 1. In such a case whatever DOI of C is, it will be changed to 1. This is an operation of *creating cluster*. Eades and Huang (2000) give a very detailed demonstration of the two operations.

The *boundary nodes* are important to determine which vertices will be displayed in the *small screen view*. We know that only leaves of Ts will be displayed in the *small screen view*. If a *boundary node* is a leaf of T , it will be a leaf of Ts . If a *boundary node* is a super vertex of T and all descendants are demagnified, the *boundary nodes* will be a leaf of Ts as well. In the case that a *boundary node* is a super vertex of T , but not all descendants of a *boundary node* are magnified, the *boundary node* won't be a leaf of Ts .

5 An Example

Figure 9 shows the DOI recomputation of Figure 7(a). When the solid vertex, vertex 13, is selected as the focal vertex, all vertices get a LFV DOI according to their structure position in the cluster tree T (see Figure 7(b)).

After GD (see Figure 9(b)), vertices 5, 11, 12, and 13 do not change their DOI because their DOIs are not smaller than the threshold value k=-5. The DOIs of vertices 6 and 14 are set as 1 because their initial DOIs are smaller than k and they are magnified. Vertices 9, 17, 18, and 19 and vertices 15 and 16 are set a DOI of $-\infty$ because their initial DOIs are smaller than k and they are demagnified. (See Figure 9(b))

From 7(b) we know that vertices 17, 18, and 19 are siblings. After GD their DOIs are $-\infty$. They will be closed and their parent, vertex 10, will be set a DOI of $-\infty$. Now both vertex 9 and vertex 10 (in Figure 7(b)) have a DOI of $-\infty$, that is, they are clustered. Their parent, vertex 3, is a *boundary vertex*, (see 7(c)). Now the iteration of DOI recomputation and *closing/creating* is complete.

Figure 7(b) also shows that vertices 14, 15, and 16 are siblings. After GD, DOIs of vertex 15, 16 are $-\infty$, and DOI of vertex 14 is 1 (see Figure 7(b)). A new cluster, vertex 15&16, is created for vertices 15 and 16, and DOI of vertex15&16 is $-\infty$. No change happens to vertex 14, but vertex 7, the parent of vertex 14, 15&16, is set as DOI of 1. Continuing the DOI recomputation upwards, there are three vertices: vertices 6, 7, and 8 (in Figure7 (b)). Vertex 6 has a DOI of 1, and DOI vertex 7 has been recomputed as 1. Vertex 8 is the only one whose DOI is $-\infty$, and a new vertex, vertex8&, is created to cluster it. The parent of vertices 6, 7, and 8& is vertex 2. Vertex 2 is a *boundary node* (see Figure 7(c)), so the recomputation and *closing/creating* is finished.

Vertices 11, 12, 13, and 5 are *boundary nodes*, so no recomputation or clustering/creating is needed.

Now a new cluster tree (Figure 10(a)) is created, and the *small screen view* is Figure 10(b).

6 Conclusions

It is challenging to present large amounts of information on small screens of handheld devices such as PDAs. Clustered graph visualization is of special importance for information visualization techniques when there is limited screen space. In this paper, we proposed and demonstrated a new approach to the clustered graph visualization by adding filtering to geometric distortion of a clustered graph presentation.

The Logical Fisheye View is one of Context+Focus techniques. It is very powerful to present structural information of a hierarchical or clustered graph by DOI thresholding. In our approach, the DOI of LFV is computed by both the conventional DOI formalization and the result of geometric distortion. This method allocates more space for displaying the details of focal vertex than the ordinary logical fisheye view techniques, and displays both geometric and structural contextual information of user's interest. It can effectively present information by giving users the control to adjust key factors in order to correctly emphasize or de-emphasize information. Visualization techniques or graph layout algorithms for hierarchical or clustered graph can be modified by this approach in order to be applied to visualization applications on small screens. The approach also has application to viewing images on a small screen.

7 Acknowledgements

We would like to thank Tim Dwyer and Amelia de Bie of the University of Sydney for their advices and help.

8 References

Apperley, M.D., Tzavaras, I. and Spence, R. (1982): A Bifocal Display Technique for Data Presentation. *Proceedings of Eurographics'82, Conference of the European Association for Computer Graphics*, 27-43.

Bartram, L., Ho, A., Dill, J., and Henigman F. (1995): The continuous zoom: A constrained fisheye technique for viewing and navigating large information spaces. *ACM Symposium on User Interface Software and Technology*, 207-215.

Card S.K., Nation D. (2002): Degree-of-Interest Trees: A Component of an Attention-Reactive User Interface. *Advanced Visual Interface '02*, Trento, ITALY.

Furnas, G. (1986): Generalized fish-eye views. *Proceedings of the 1986 ACM Conference of Human Factors in Computing Systems, CHI '86*, Bostons, New York, 16-23, ACM Press.

Gutwin, C., Amy, S. (2003): Fisheyes are good for large steering tasks. *Proceedings of the conference on Human factors in computing systems, Conference on Human Factors and Computing Systems*, Ft. Lauderdale, Florida, USA, p.201 - 208, ACM Press.

Keahey, T., and Robertson, E. (1996): Techniques for nonlinear magnification transformations. *Proc. of the IEEE Symposium on Information Visualization*, 38-45.

Mackinlay, J., Robertson, G., and Card, S. (1991): The perspective wall: Detail and context smoothly integrated. *Proc ACM CHI'91 (1991)*, New Orleans, 173 - 180.ACM Press.

Rao, R., and Card, S. K. (1994): The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus+Context Visualization for Tabular Information. *Proceedings of Human Factors in Computing Systems (CHI 94)*, 318-322, ACM Press.

Robertson, G., Mackinlay, J., and Card, S. (1991): Cone Trees: Animated 3D Visualizations of Hierarchical Information. *Proc. CHI '91*, New Orleans, LA, 189--194. ACM Press,

Holmquist, L.E. and Ahlberg, C. (1997): Flip Zooming: A Practical Focus+Context Approach to Visualizing Large Information Sets. *Proc. HCI International '97*, Elsevier, Amsterdam, 763-766.

Card S.K., Mackinlay J.D. and Shneiderman B. (1999): *Readings in information visualization: using vision to think*. Morgan Kaufmann Publishers.

Eades, P., Feng, Q., Lin, X.(1996) Straight-Line Drawing Algorithms for Hierarchical Graphs and Clustered Graphs. Graph Drawing, 113-128.

Eades, P. and Huang, M. L. (2000): Navigating Clustered Graphs using Force-Directed Methods. *Journal of Graph Algorithms and Applications*, 4 (3), 157-181.

Eades P, Lin T, and Lin X (1993): Two tree drawing conventions. *Int J Comput Geom Appl* 3:133-153.

Leung, Y., and Apperley, M. (1994): A Review and Taxonomy of Distortion-Oriented Presentation Techniques. *ACM Trans. on Computer-Human Interaction*, 1 (2), 126-160.

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.

Lengauer, T., and Wanke, E. (1988): Efficient Analysis of Graph Properties on Context-free Graph Languages. ICALP 379-393

Sarkar, M. and Brown, M. H. (1994): Graphical fisheye views. *Communications of the ACM*, 37 (12): 73-8.

Sugiyama K.; Misue K.: *Visualization of Structural Information: Automatic Drawing of Compound Digraphs*, IEEE Trans. Sys., Man, and Cybernetics, 21(4), pp. 876-892, 1991.