

Visualisation and Comparison of Image Collections based on Self-organised Maps

Da Deng

Jianhua Zhang

Martin Purvis

Department of Information Science
University of Otago
PO Box 56, Dunedin, New Zealand
E-mail: {ddeng, mpurvis}@infoscience.otago.ac.nz

Abstract

Self-organised maps (SOM) have been widely used for cluster analysis and visualisation purposes in exploratory data mining. In image retrieval applications, SOMs have been used to visualise high-dimensional feature space and build indexing structures. In this paper, we extend the use of SOMs for profiling and comparison of image collections, and present empirical results obtained in collection visualisation, visual and quantitative comparison of collections, and a prototype system implementation.

1 Introduction

Cluster analysis and visualisation techniques play key roles in exploratory data analysis. This is even more obvious in the area of multimedia information processing. Multimedia assets need efficient algorithms for information and knowledge extraction, indexing, and retrieval. The challenge arises not only from the sheer huge volume of multimedia data storage, but also from the fact that feature extraction process often results in feature space of high dimensionality. Besides these factors, there is a general difficulty existing in extraction of semantic information from unstructured raw multimedia data. The current content-based image retrieval (CBIR) approach (Smeulders, Worring, Santini, Gupta & Jain 2000), relies on using some combinations of low-level visual features, such as colour, texture, and shapes, for indexing. To name but a few representative works, there are (Smith & Chang 1996, Pass & Zabih 1999, Carson, Thomas, Belongie & et al. 1999). Higher level features such as colour semantics (Corridoni, Del Bimbo & Pala 1999) and visual concepts (Buijs & Lew 1999) have also raised attention.

In the context of multimedia asset management, one is more interested in examining collective information rather than individual multimedia resources. We propose to extend the CBIR approach from individual image retrieval toward image collection management (Deng 2003). This requires the employment of some clustering and visualisation techniques to analyse low-level visual features and build profiles as the collective representation of multimedia information within image collections, and employ some quantitative measurement of the similarity between these profiles. The basic

computational model we adopt for this study is the self-organised maps (SOM), see (Kohonen 1997).

In the next section we present a discussion on the self-organised map model we are to use for a number of purposes, and some relevant practical considerations. In Section 3 a system implementation is introduced, together with some empirical results we obtained on image search and collection comparison. We conclude with some discussion on future work.

2 The Computational Model

2.1 The SOM algorithm

SOM is a popular neural network model that has been applied widely in data clustering, visualisation, and even time series modelling. In the field of information retrieval, it has found application in a number of works, such as (Rauber & Merkl 1999, Laaksonen, Koskela & Oja 1999, Nürnberger & Klose 2002). Although bearing a rather simple mathematical form, the SOM algorithm bears some interesting characteristics.

The most significant advantage of SOM is its capability of carrying out vector quantisation and multi-dimensional scaling simultaneously. The map, usually set in 2-D or 3-D topology, consists of a regular lattice of neurons set in hexagonal or rectangular topology. Each neuron is associated with a weight vector. The map attempts to perform localised clustering on these node vectors, while in the meantime the ordering on the lattice works to match similar inputs to the same node or nodes close to each other, and dissimilar inputs onto nodes far from each other. The nodes are sometimes also called as units, and unit vectors are called as prototypes.

Assume we have a N -prototype SOM to train. Denote $\mathbf{w}_i(t)$ as the weight vector associated with node i . Given an input $\mathbf{x}(t)$, the algorithm first finds the best-matching unit (BMU) \mathbf{w}_b among all prototypes, i.e., the weight vectors are then updated according to the following learning rule:

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \gamma(t)h_{b,i}(t)[\mathbf{x}(t) - \mathbf{w}_i(t)] \quad (1)$$

where $h_{b,i}$ is a neighbourhood function centred at BMU and shrinking over time, and $\gamma(t)$ the learning rate decreasing with the time. There are some variants proposed to this original learning rule, but generally it has been shown that these learning rules lead to the convergence of unit vectors.

The following characteristics of SOM make it a natural choice for clustering and visualisation applications:

- Good visualisation ability. Network nodes are located on a low-dimensional lattice which is easy for visualisation and human interpretation. This also keeps users' navigation in the

high-dimensional feature space traceable on a low-dimensional map.

- Topology preserving. Similar inputs are mapped onto the same node or nodes in a neighbourhood on the map. This means that similar images can be closely mapped onto the grid, also making browsing easier and robust. Hierarchical design of maps is also made possible.
- Density matching. Although not being able to match *exactly* to the probability distribution underlying in the input data (see (Haykin 1999), page 460-461), the algorithm of SOM manages to represent a cluster of frequently occurring input stimuli by a larger area in the feature map. If we denote the number of nodes in a small volume $d\mathbf{x}$ over the input space \mathcal{X} as $m(\mathbf{x})$, it is proved SOM manages to have $m(\mathbf{x}) \propto p_{\mathbf{x}}^{2/3}(\mathbf{x})$, where $p_{\mathbf{x}}(\mathbf{x})$ is the probability density function of the input \mathbf{x} (Ritter 1991).

For these reasons it is not surprising that a number of applications have been built on the SOMs or tree-structured SOMs for image retrieval. To extend the use of SOM for profiling is as straightforward as using SOM trained on content-based visual features to construct a snapshot of the whole collection so that operations such as browsing, search, and comparison can be carried out in efficiency. For purposes of navigation and browsing hierarchical structures are often employed, but for profiling oriented for quick comparison across collections, we adopt flat SOMs to facilitate efficient calculation. On the other hand, little mathematical work has been done on the optimal structure design, topology preserving and density matching ability etc. for hierarchical feature maps.

2.2 Visualisation by SOM

There are some practical issues to be considered when using SOM as an interface for image collection navigation.

The outcome of the training process is rather flexible in prototype positioning and orientation, basically owing to the random initialisation of prototype vectors, while variation in other parameters may also contribute to this. Consequently the profile generated for the same image collection may look different every time it has been trained. To work around this we adopt one feature in the SOM_PAK toolbox (Kohonen, Hynninen, Kangas & Laaksonen 1996), linearly initialising prototype vectors using principal components of the input space. In our experiments we found this also reduces the chance of having ill-trained maps with foldings, and training can be done quicker.

For the visualisation of the trained maps, multi-dimensional scaling (MDS) is often needed so as to project the high-dimensional vector space onto a low-dimensional visualisation space typically of 2-D or 3-D. Sammon's mapping (Sammon 1969) is a MDS method that carries out gradient descent on an error function defined by the difference of distance order between prototypes before and after projection. The outcome of this gradient descent technique is however not unique but depends on initialisation. Instead of using random initialisation, We use deterministic initialisation by PCA. This not only stabilises the outcome of MDS, but also speed up the convergence of Sammon's mapping as found in (Naud & Duch 2000).

2.3 Distance between feature maps

With image collection profiles generated as feature maps consisting of data prototypes, it paves the way for quantitative evaluation of the similarity of image collections, which is a formidable task to carry out directly in the original data space.

Surprisingly the problem of distance measuring of SOMs has received little attention in studies. In (Kaski & Lagus 1996) an approach was given based on evaluation of the quality of feature maps. As a more straight-forward solution, we investigate a number of point set distance metrics in (Deng 2003), including Hausdorff Distance, Sum of Minimal Distance, and Earth Mover Distance (Rubner, Tomasi & Guibas 1998). A new distance called *SAND* is proposed as a modification of the sum of minimal distance, taking into account of the probability density modelling property of SOMs.

Assume we need to compare two maps X and Y (not necessarily of the same size). For a prototype $x \in X$, its BMU on the other map, i.e., the prototype $y_b \in Y$ with the minimum distance to x , can be found. To calculate SAND, this minimum distance is averaged with the distances between x and y_b 's neighbours, before it is summed across all population of X . The same process is then repeated for set Y .

The calculation process is summarised in the following steps:

1. Find the BMU $y_b \in Y$ for any $x \in X$, with

$$b = \arg_{y \in Y} \min(\|x - y\|). \quad (2)$$

2. Find out all best-matching pairs (α, β) between the neighbourhood of x and y_b , and calculate the averaging distance:

$$d_n(x) = E\{\|\alpha - \beta\|\}, \forall \alpha \in \Omega(x), \beta \in \Omega(y_b). \quad (3)$$

Here $\Omega(\cdot)$ denotes the neighbourhood of a map node.

3. Sum up the individual measures:

$$\text{SAND}(X, Y) = \frac{1}{2} \left(\sum_{x \in X} d_n(x) + \sum_{y \in Y} d_n(y) \right). \quad (4)$$

The rationale behind this scheme is based on SOM's probability density matching ability. By examining the matching among a map neighbourhood can tell the difference between maps of similar range of spatial span yet originated from different probability distributions. As density differing in the original feature space will result in, on the low dimensional map, either dense grids or sparse grids, the difference can be better reflected by SAND than a plain point-to-point measure.

2.4 Hierarchical map construction

To leverage the use of SOM for large-scale image database, hierarchical maps can be constructed, using some pyramid structure, e.g., (Lampinen 1992, Laaksonen et al. 1999). This avoids the time-consuming sequential search for the BMU in large feature maps. Searching now starts at the root level, descending to the next-lower level to locate the winner, and so on, until the BMU is found on the leaf level. In this study we adopt the HSOM proposed in (Lampinen 1992). To search out other good matchings we do not need to conduct a brute force search among all prototypes. Rather, we restrict the search among the neighbours of the BMU and

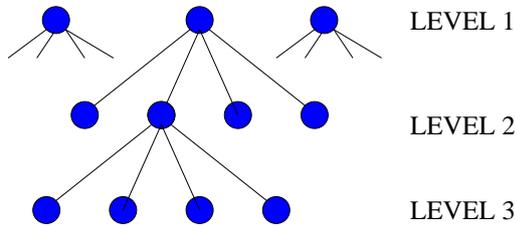


Figure 1: The Hierarchical SOM structure.

all those nodes connected to the immediately mother node of the BMU. We will show the effectiveness of such a search scheme later.

The HSOM structure is shown in Fig.1. The training process of HSOM can be summarised in the following steps:

- Define maximum number of levels, number of nodes neighbourhood size in each level, and initialise weights in each submap.
- Train the top level using the SOM algorithm.
- Split the training data set by picking the subset of data samples that matches the best to each of the node in the current level, and use the subset to train the submap under the node.
- Repeat the process until the number of subset samples reduces under a certain threshold.

3 An Empirical Study

3.1 Data processing

We use images of various image retrieval data sources available on the Internet, e.g. the Ground Truth Database from the Department of Computer Science, University of Washington, and pictures stored at the SUNET FTP site. The total number of images used in this study is about 3000.

For contend-based indexing of the images, we extract three types of low-level visual features:

1. Regional average colours in the LUV colour space from five rectangular zones of the image, similar to that defined in (Laaksonen et al. 1999). This gives feature codes of 15 dimensions.
2. Edge density histogram. For an image its edges are first extracted by Canny edge detection and local edge density statistics is collected (Pass & Zabih 1999).
3. Gabor filtering. Gabor filters are wide used for texture analysis (Turner 1986). A set of filters in 4 frequency levels and 8 orientations is adopted, resulting in 32-dimensional feature codes from the response energy of these filters.

3.2 System implementation

A software package, COVIC, is developed as a prototype system aimed at multimedia asset management. The user interface is written in Java and can be run either as an application or an applet embedded in web pages. For efficiency concern, visual feature extraction from raw image collections is written in C and runs at the server end. The training of hierarchical SOMs also is typically a time-consuming work and is done using a shell script calling SOM_PAK utilities originally written in C.

We use PostgreSQL DBMS to store information such as image collection title, URLs, image filenames,

and visual feature codebooks. Hierarchical feature maps built for image collections are also saved in data tables. COVIC handles database management tasks and image browsing and query via the JDBC interface of PostgreSQL.

The COVIC user interface, as shown in Fig.2, consists of the following elements:

- *Navigation pane*, where hierarchical maps can be displayed and navigated with the help of a toolbar. Two modes of SOM visualisation are available: 'MapView' shows the map in hexagonal grids and displays the similarity between nodes using some colouring scheme, while 'ProjectView' generates Sammon's mapping for the current feature map. Map nodes visualised in the latter mode bear a thumbnail label of the image that matches the best to the corresponding map prototype. This, together with the distance information provided by Sammon's mapping, helps the user to develop an interpretation of similarity between map nodes.
- *Examples pane*, shown on the upper right of Fig.2, where example images can be selected by right-clicking on any thumbnail image shown in the map.
- *Control pane* with a toolbar, which allows the user to choose from map visualisation modes, zoom-in and zoom-out the map display, travel up and down among the HSOM layers, switch between multiple feature schemes for map navigation or image searching, and carry out search once selected any thumbnail image from the 'Examples pane'.
- *Results pane*, where search results are displayed on descending order of similarity.

Some utilities provided by COVIC are -

- Generation of profiles of image collections for visualisation and navigation;
- Hierarchical SOM construction for image collections;
- Navigation of through the image collections via hierarchical SOMs. For instance, one can drill down a map to find more relevant images in lower layers of the map by clicking on the image node of interest. To broaden the browsing scope one can use the up-arrow button shown in the toolbar to move to the upper layer of the SOM structure.
- Image query-by-examples. If a user locates any image of interest from the map, he or she can search out all similar images, choosing different feature schemes for this purpose. The example image can also be uploaded from the user's end.

The hierarchical SOM structure has been found to be effective for users to navigate through and locate the image item of interest. The top level map of the image collection is shown on the left of Fig.2, while Fig.3 shows the submap no.55, accessible by clicking on the car image. The finer granularity of the submaps on lower levels allows users to locate the image of interest, while the top level helps to guide the users navigation by providing a rough all-in-one snapshot of the image galaxy.

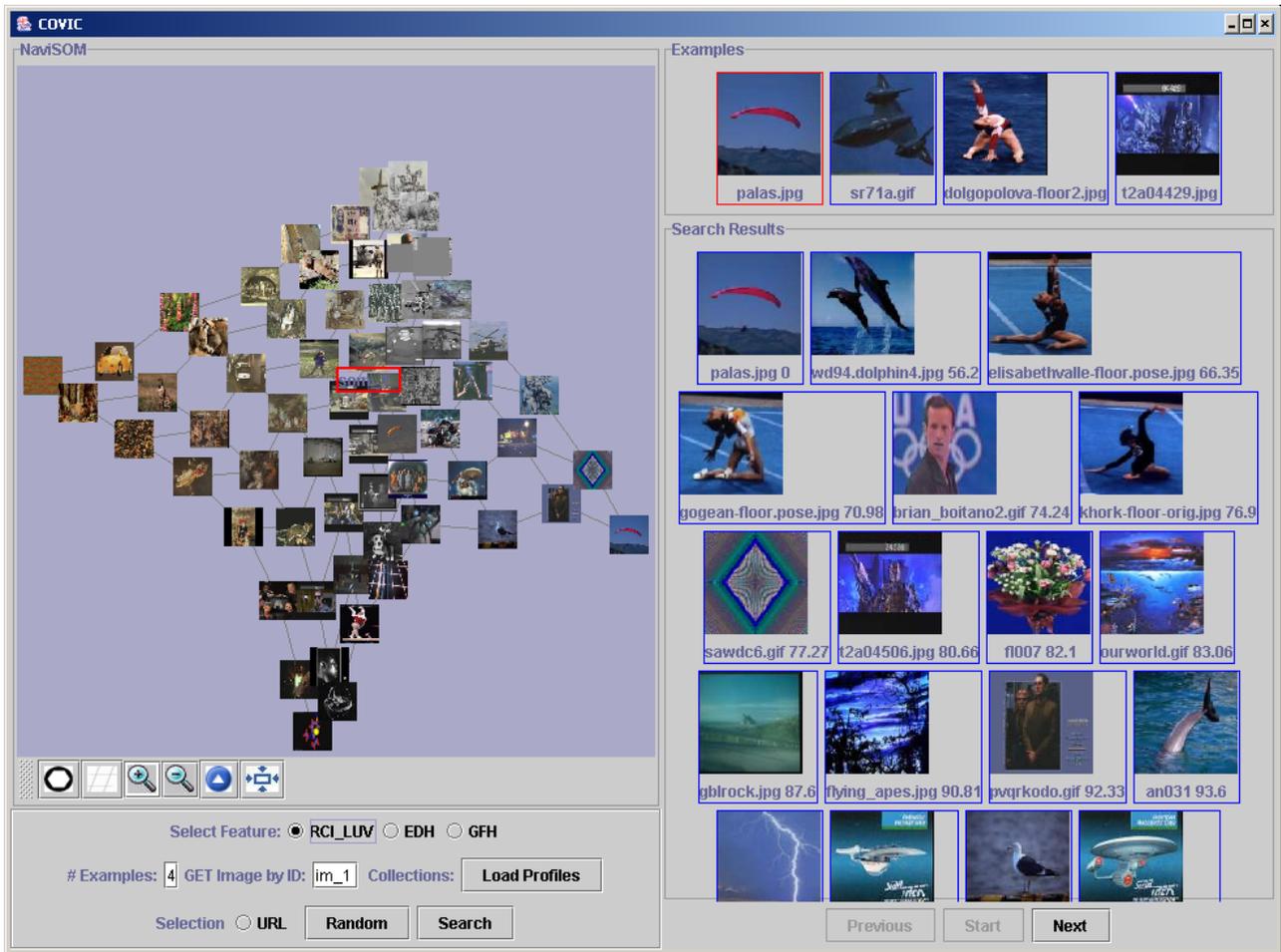


Figure 2: The COVIC system.

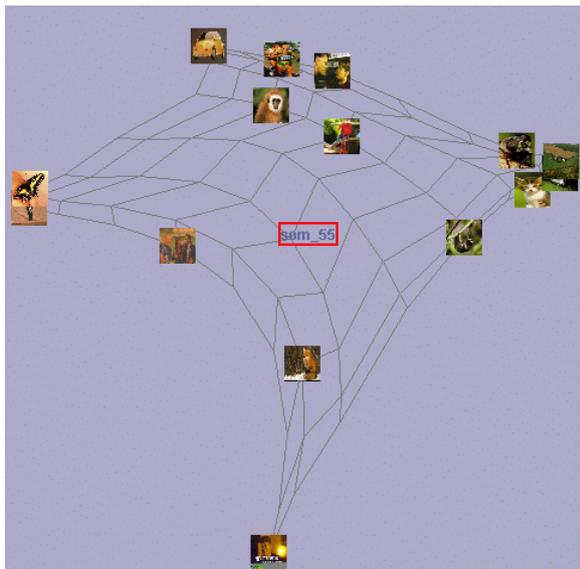


Figure 3: Submap no.55 in level 2 of the HSOM tree.

3.3 Search results assessment

Query-by-examples works by matching the visual feature codes of the query images with those stored in database. Because of the hierarchical structure of the feature maps, one can expect improved efficiency by guiding the search process on the map pyramid. Our method is to search from the top node and then gradually locate the best matching node going through successive map layers. Because of the content-based approach, the best matching unit may not match exactly the best to the query, so the system needs to return more results by examining map nodes connected in the direct neighbourhood, and even reverse one level upward so as to include a larger neighbourhood.

During the implementation we carried out experiments assessing the quality of such an image search mechanism. Since of the Java implementation of COVIC, no significant improvement in efficiency was observed using this approach, as the response time is mostly spent for the Java VM to load in the image thumbnails for display. On the other hand, we managed to evaluate the effectiveness of such a search process, by comparing the search results with that of doing a global search among all prototype vectors in brute force. The comparison results, shown in Fig.4, are from two scenarios - the best case and the worst case. The best case is when the query image matches exactly to the BMU map prototype, while the latter when it matches to an image that lies the furthest from the BMU. Consequently we can see the worst case produces more mismatches in our search outcome, as the example stays at the boundary of its Voronoi region and therefore are quite similar to

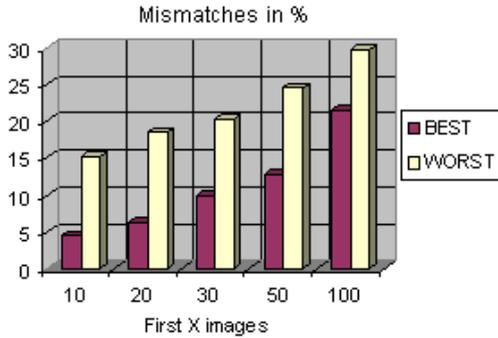


Figure 4: Mismatches of the SOM-based search compared with the global search.

the images belonging to other map nodes. For the worst cases, our search utility generates on average about 30% mismatches within the first 100 images. Although this appears to be quite unsatisfactory, it can be mitigated by using multiple images as ‘examples’, or, further introducing relevance feedback mechanism into the search utility.

3.4 Distance between collection profiles

Our evaluation uses four collections from the SUNET pictures: Views, Vehicles, Animals, and Sports, with more than 1000 images in total. Thumbnail images from the first three collections are shown in Fig.5.

These images from different collections share elements of strong visual similarity: blue sky, water, plants, and sometimes even vehicles. Consequently one can expect overlap of low level feature space across these collections. Take the regional average colour feature as an example. Four map profiles, each of 64 nodes, are generated from the collections. Their partial overlapping is obvious as shown in Fig.6, where these maps are projected together by doing a 2-D PCA of all prototypes. Visual inspection of this joint projection can give us some clue, e.g., that the Views collection lies closer to the Vehicles collection than to Sports. This can be explained by the fact that many images from the Vehicles collection have outdoor background, while those images in the Sports collection are often close-up of players.

Results of the SAND map distance corresponding to this visual inspection are:

$$\begin{aligned} \text{SAND}(\text{Views}, \text{Sports}) &= 76.1 \\ \text{SAND}(\text{Views}, \text{Vehicles}) &= 57.5 \end{aligned}$$

One can use profiles generated from different feature schemes to evaluate collection similarity, and consequently the outcome may vary. We are working on combining distance metrics derived from different visual scheme so that the the overall similarity between image collections can be better modelled.

4 Conclusion

In this work we present some empirical results in profiling and comparison of image collections using SOMs. The prototype system COVIC, based on hierarchical SOMs generated on low-level visual features, is implemented as an interface for users to browse through image collections, conduct image search, and also assess the similarity between image collections.

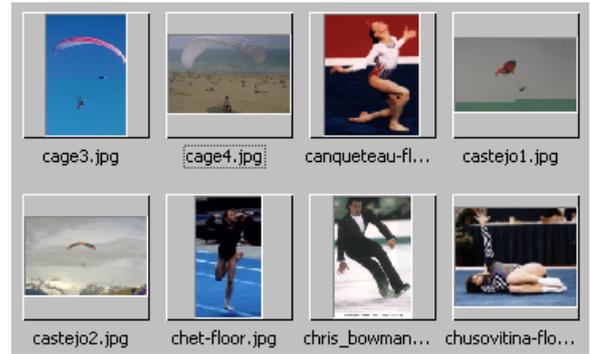
On the other hand, the computational approach is not optimal. SOM basically is not incremental and has poor plasticity in continuous on-line learning.



(a)



(b)



(c)

Figure 5: Thumbnails from image collections: (a)Views, (b)Vehicles, and (c) Sports.

Its probability density modelling is not perfect and it is subject to rigid topology setting that may fit poorly to new data during on-line learning, for which the situation is even worse in HSOM. Noticeably there have been some solutions with growing network models that aim at exact probability modelling, e.g., (Tino & Nabhey 2002). It would be interesting to investigate how they scale for multimedia asset management problems. On the other hand, with the introduction of probabilistic modelling algorithms, other metrics defined in probability distribution space can be explored.

While CBIR provides a useful approach for multimedia information retrieval, it is still of rather limited capability. It has become more and more widely accepted that there is a semantic gap for CBIR researchers to overcome, in order to achieve more effective image data storage, retrieval and understanding (Smeulders et al. 2000). This may require joint effort in pattern recognition, machine learning, and knowledge engineering. We believe the semantic gap has the same implication for multimedia

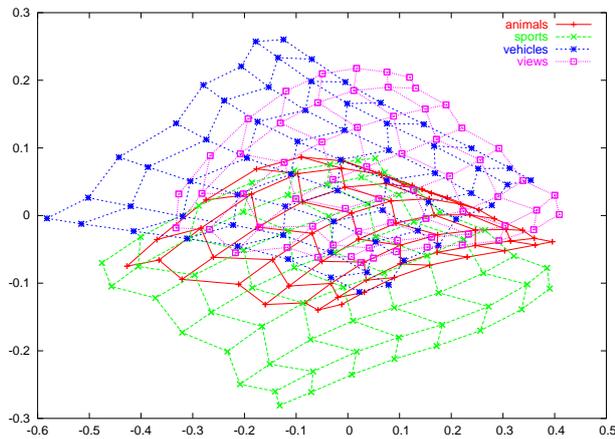


Figure 6: Four collection profile maps visualised by 2-D PCA.

asset management and it has to be addressed in the future.

Acknowledgment: This work is supported by University of Otago Research Grant 200200621.

References

Buijs, J. & Lew, M. (1999), Learning visual concepts, in 'Proc. ACM Multimedia 99', pp. 5–7.

Carson, C., Thomas, M., Belongie, S. & et al. (1999), Blobworld: A system for region-based image indexing and retrieval, in 'Proc. Int. Conf. Visual Inf. Sys.', pp. 509–516.

Corridoni, J., Del Bimbo, A. & Pala, P. (1999), 'Image retrieval by color semantics', *Multimedia Systems* **7**, 175–183.

Deng, D. (2003), Content-based image collection profiling and comparison via self-organised maps, in 'IEEE Conf. on Hybrid Intelligent Systems, to appear'.

Haykin, S. (1999), *Neural Networks: A Comprehensive Foundation*, second edn, Prentice Hall.

Kaski, S. & Lagus, K. (1996), Comparing self-organizing maps, in J. Vorbruggen & B. Sendhoff, eds, 'Proceedings of ICANN96 International Conference on Artificial Neural Networks', Vol. 1112 of *Lecture Notes in Computer Science*, Springer, Berlin, pp. 809 – 814.

Kohonen, T. (1997), *Self-organizing Maps*, second edn, Springer-Verlag.

Kohonen, T., Hynninen, J., Kangas, J. & Laaksonen, J. (1996), SOM_PAK: The self-organizing map program package, Technical Report A31, Helsinki University of Technology, Laboratory of Computer and Information Science.

Laaksonen, J., Koskela, M. & Oja, E. (1999), Content-based image retrieval using self-organizing maps, in 'Visual Information and Information Systems', pp. 541–548.

Lampinen, J. (1992), On clustering properties of hierarchical self-organizing maps, in I. Aleksander & J. Taylor, eds, 'Artificial Neural Networks, 2', Vol. II, North-Holland, Amsterdam, Netherlands, pp. 1219–1222.

Naud, A. & Duch, W. (2000), Interactive data exploration using mds mapping, in '5th Conference on Neural Networks and Soft Computing', pp. 255–260.

Nürnberger, A. & Klose, A. (2002), Improving clustering and visualization of multimedia data using interactive user feedback, in 'Proceedings of IPMU 2002', pp. 993–999.

Pass, G. & Zabih, R. (1999), 'Comparing images using joint histograms', *Multimedia Systems* **7**(3), 234–240.

Rauber, A. & Merkl, D. (1999), The somlib digital library system, in 'Proc. of European Conference on Digital Libraries', pp. 323–342.

Ritter, H. (1991), 'Asymptotic level density for a class of vector quantization processes', *IEEE Trans. Neural Networks* **2**, 173–175.

Rubner, Y., Tomasi, C. & Guibas, L. (1998), A metric for distributions with applications to image databases, in 'Proc. of IEEE ICCV', pp. 59–66.

Sammon, W. (1969), 'A nonlinear mapping for data analysis', *IEEE Transactions on Computers* **5**, 401–409.

Smeulders, A., Worring, M., Santini, S., Gupta, A. & Jain, R. (2000), 'Content-based image retrieval at the end of the early years', *IEEE Transaction on Pattern Analysis and Machine Intelligence* **22**(12), 1349–1380.

Smith, J. & Chang, S. (1996), Visualseek: a fully automated content-based image query system, in 'Proc. of ACM Multimedia 96', pp. 87–98.

Tino, P. & Nabhey, I. (2002), 'Hierarchical gtm: constructing localized non-linear projection manifolds in a principled way', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(5), 639–656.

Turner, M. (1986), 'Texture discrimination by gabor functions', *Biological Cybernetics* **55**, 71–82.