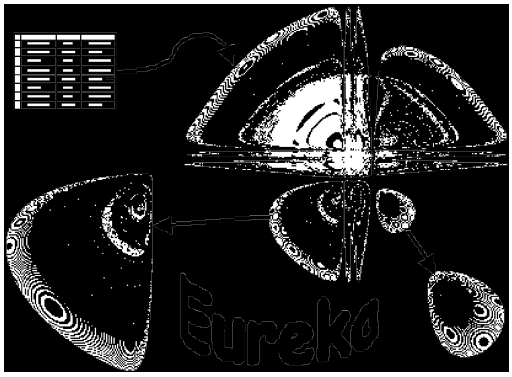


Proceedings
International Workshop on Visual Data Mining

Edited by Simeon J. Simoff, Monique Noirhomme-Fraiture and Michael H. Böhlen

---

in conjunction with ECML/PKDD2001 - 2th European Conference on Machine Learning (ECML'01) and 5th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'01), 3-7 September, 2001, Freiburg, Germany

---

Proceedings of the International Workshop on Visual Data Mining - VDM@PKDD'2001,
in conjunction with ECML/PKDD2001 - 2th European Conference on Machine Learning
(ECML'01) and 5th European Conference on Principles and Practice of Knowledge
Discovery in Databases (PKDD'01), 3-7 September, 2001, Freiburg, Germany
S. J. Simoff, M. Noirhomme-Fraiture and M. H. Böhlen (eds)

Workshop Web Site: http://www-staff.it.uts.edu.au/~simeon/vdm_pkdd2001/

Foreword

John W. Tukey, who made unparalleled contributions to statistics and to science in general, during his long career at Bell Labs and Princeton University, emphasized that seeing may be believing or disbelieving, but above all, data analysis involves visual, as well as statistical, understanding. Perhaps the most famous and certainly one of the oldest visual explanations in mathematics is the visual proof of the Pythagorean theorem. This proof is unusual in its brevity and its complete appropriateness to the problem. Pictures and diagrams are also used in non-geometrical parts of mathematics, mostly for psychological reasons: harnessing our ability to reason "visually" with the elements of a diagram in order to assist our more purely logical or analytical thought processes. Thus, visual reasoning approach to the area of data mining and machine learning promises to overcome some of the difficulties experienced in the comprehension of the information encoded in data sets and the models derived by other quantitative data mining methods.

Visual data mining is an emerging area in explorative and intelligent data analysis and mining which is based on the integration of concepts from computer graphics, visualisation metaphors and methods, information and scientific data visualisation, visual perception, cognitive psychology, diagrammatic reasoning, 3D virtual reality systems and recently, from affective computing and collaborative virtual environments. Visual data mining is a collection of interactive reflective methods that support exploration of data sets by dynamically adjusting parameters to see how they affect the information being presented. It offers the machine learning and data mining community powerful means of analysis that can assist in uncovering patterns and trends that are likely to be missed with other non-visual methods. Data mining projects are often performed using other non-visual paradigms, such as quantitative statistical methods, rule induction, unsupervised neural network modeling, combined sometimes with genetic programming methods. However, many of these approaches require that the data is analysed in a hypothesis testing mode in which one have a priori notions about what the important results will be before the analysis actually begins. Results obtained with these methods tend to describe overall group trends, generalised differences, as well as broad categorizations. Although clearly useful for many purposes, global trends are not the most interesting or actionable outcomes in many cases. Visualisation methods allow to discover overall trends in a data set while also affording an opportunity to discover smaller hidden patterns that can often be just as important within an application. Visualising the correlations and associations among data objects can quickly reveal patterns and trends implicit in the data, thus increasing the chances of making discoveries out of the information encoded in the data set.

Since visual analysis is such a different technique, it is an extremely significant topic for the growing area of data mining and knowledge discovery research and development. When data mining routinely uses data visualisation to help manage diverse, intricate, and complex data sets, the emphasis of machine learning research has been on symbolic methods (whether analytical or quantitative numerical). Visualisation has proven to be reliable, easy to learn, and extremely cost effective. Additionally, visualistion provides a natural method for integrating multiple data sets and has been used across a number of disciplines including commercial research, forensic accounting, and

throughout the investigative community. Visual data mining expands this research with developing systematic methodologies for visual reasoning and discovery. Visual data mining techniques offer the luxury of being able to make observations without preconception. During the visual analysis one can discover patterns of interest, based on boundary violations, frequency of occurrence and all sorts of data interdependencies. The intimate involvement of the human in the twists and turns of the analysis leads to deeper understanding of the data set – sometimes much deeper, than the one that can be achieved with non-visual methods, like neural network training. A major problem in the development of visual data mining methods is that in absolute sense there is no one type of visualisation model that is better than another. The visualisation method of choice will usually be determined by such factors as appropriateness for the application domain, scope of the data sets, how data is mapped onto visualisation schemata, how the visualisation schemata utilises cognitive "strengths" of human information processing and how it overcomes its cognitive limitations.

The co-occurrence of PKDD and ECML offers unique and perhaps the best opportunity for discussing the latest developments in visual data mining and how machine learning and knowledge discovery methodologies can benefit from it and build on it. The main goal of this international workshop (dated after the Visual Data Mining workshop at 2001 KDD Conference in San Francisco) is to provide a forum for presentation and discussion of the newest both mature and greenhouse ideas, research and developments in visual data mining and supporting disciplines, and to identify the short- and long-term research directions in the field and preferences of the potential end users. This volume contains the papers selected for presentation at the workshop. The papers are of particular interest to the broad community of researchers in cognitive technologies in computing. They are grouped in the following streams:

- Methodologies for Visual Data Mining
- Applications of Visual Data Mining
- Support for Visual Data Mining

We would like to thank all those, who supported our efforts on all stages - from the development and submission of the workshop proposal to ECML/PKDD Selection Committee through to the preparation of the final program and proceedings. Each paper was reviewed by two referees drawn from the international program committee. The papers have been reviewed under tight time constraints. Special thanks and acknowledgements go to them for the final quality of selected papers depends on their efforts.

Simeon J. Simoff
Monique Noirhomme-Fraiture
Michael H. Böhlen

Workshop co-Chairs
July 2001

Workshop Chairs

- Simeon J. Simoff
- Monique Noirhomme-Fraiture
- Michael H. Böhlen

Program Committee

- James L. Alty, Loughborough University, UK
- Heinz-Dieter Boecker, GMD National Research Center for Information Technology, Germany
- Chaomei Chen, Brunel University, UK
- Di Cook, Iowa State University, USA
- John Debenham, University of Technology Sydney, Australia
- Alberto Del Bimbo, Universitá degli Studi di Firenze, Italy
- Edwin Diday, Université Paris IX - Dauphine, France
- Chitra Dorai, IBM Thomas J. Watson Research, USA
- Alex Duffy, University of Strathclyde, UK
- Erik Granum, Aalborg University, Denmark
- Georges Hebrail, EDF R&D, France
- Maolin Huang, University of Technology Sydney, Australia
- Alfred Inselberg, Multidimensional Graph Ltd, Israel
- Daniel A. Keim, University of Konstanz, Germany
- Carlo Lauro, University of Naples, Italy
- Torsten Möller, Simon Fraser University, Canada
- Bruce Thomas, University of South Australia, Australia
- Carl H. Smith, University of Maryland, USA
- Masaki Suwa, Chukyo University, Japan
- Osmar R. Zaïane, University of Alberta, Canada
- Ahmed Zighed, Universite Lumiere Lyon, France

Preliminary Program for VDM@PKDD2001 Workshop

Tuesday, 4 September, 2001, Freiburg, Germany

9:00 – 9:15	**Opening and Welcome**
9:15 – 10:15	**Session 1 – Methodologies**
9:15 – 9:45	*Using Nested Surfaces to Detect Structures in Databases* Artüras Mažeika, Michael Böhlen and Peer Mylov
9:45 – 10:15	*Methods for Visual Mining of Data in Virtual Reality* Henrik R. Nagel, Erik Granum and Peter Musaeus
10:15 – 10:45	**Coffee Break**
10:45 – 12:30	**Session 2 – Applications**
10:45 – 11:15	*Visual Data Mining Using a Constellation Graph* Tokihiko Niwa, Kenji Fujikawa, Kazuyoshi Tanaka, and Mayumi Oyama
11:15 – 11:45	*Mining Travel Data with a Visualiser* Colin Ho, Ben Azvine
11:45 – 12:15	*Customer Data Mining and Visualization by Generative Topographic Mapping Methods* Jinsan Yang and Byoung-Tak Zhang
12:15 – 12:30	*The 3DVDM Group in Aalborg University (Denmark)* Erik Granum
12:30 – 14:15	**Lunch**
14:15 – 15:45	**Session 3 - Support**
14:15 – 14:45	*Supporting Data Analysis Through Visualizations Paolo Buono*, Maria Francesca Costabile, Francesca A. Lisi
14:45 – 15:15	*Introducing Signature Exploration: a Means to Aid the Comprehension and Choice of Visualization Algorithms* Penny Noy and Michael Schroeder
15:15 – 15:45	*Demonstration of Multimedia Support for Visual Data Mining* Monique Noirhomme-Fraiture
15:45 – 16:15	**Coffee Break**
16:15 – 17:30	**Session 4 – Discussion**
16:15 – 16:35	*Towards the Development of Environments for Designing Visualisation Support for Visual Data Mining* Simeon J. Simoff
16:35 – 17:30	Discussion Panel and Closing

Table of Contents

Using Nested Surfaces to Detect Structures in Databases

Artūras Mažeika[1,2] arturas@cs.auc.dk
Michael Böhlen[1] boehlen@cs.auc.dk
Peer Mylov[2] mylov@hum.auc.dk

[1]Department of Computer Science, Aalborg University, Fredrik Bajers Vej 7E, 9220 Aalborg, Denmark
[2]Institute of Communication, Aalborg University, Niels Jernes Vej 14, 9220 Aalborg, Denmark

Abstract. We define, compute, and evaluate *nested surfaces* for the purpose of visual data mining. Nested surfaces enclose the data at various density levels, and make it possible to equalize the more and less pronounced structures in the data. This facilitates the detection of multiple structures, which is important for data mining where the less obvious relationships are often the most interesting ones. The experimental results illustrate that surfaces are fairly robust with respect to the number of observations, easy to perceive, and intuitive to interpret. We give a topology-based definition of nested surfaces and establish a relationship to the density of the data. Several algorithms are given that compute surface grids and surface contours, respectively.

1 Introduction

Visual data mining exploits the human perceptual faculties to detect interesting relationships in the data. To support the detection of relationships it is important to visualize data in a form that is easy understandable to humans. It is common to use scatter plots for this purpose [3]. Employing scatter plots is intuitive as each observation is faithfully displayed. Scatter plots have successfully been used for detecting relationships in two dimensions. For higher dimensions scatter plots are combined with grand tour methods. A grand tour displays a smooth rotation of two dimensional projections that eventually covers the entire high dimensional search space.

Scatter plots hit limitations if the dataset is big, noisy, or if it contains multiple structures. With lots of data the amount of displayed objects makes it difficult to detect any structure at all. Noise easily blurs the picture and can make it impossible to find interesting relationships. Finally, with multiple structures it often happens that one structure is more pronounced than another. In this situation the less pronounced structures easily get lost. For the purpose of data mining this is particularly bad as it is usually the less obvious relationships that are the most interesting ones. Surfaces equalize the more and less pronounced structures and thus support the detection of less obvious relationships.

In this paper we explore the potential of nested surfaces to analyze data sets. Nested surfaces enclose the data at varying densities. Humans are used to perceive surface information and to abstract surfaces from individual observations. This greatly simplifies the interpretation of the data. Nested surfaces do not suffer if the amount of data is big, and the nesting supports the detection of multiple structures. We provide a topology-based definition of surfaces and prove that the boundary $\partial C_\alpha = \partial\{(x,y,z) : f(x,y,z) \geq \alpha\}$ is a surface if the density function f has a continuous derivative. This

provides the basis for an algorithm that computes nested level surfaces. Given a density estimation, which has continuos derivative, and a density level α we give algorithms that compute surface grids and surface contours, respectively. The described methods have been implemented and integrated into the 3D Visual Data Mining (3DVDM) System. The 3DVDM System is used for explorative data analyses in a 6-sided Cave, a 180° Panorama, and on regular computer monitors. It can be downloaded from `http://www.cs.auc.dk/3DVDM` and runs on SGI and PC/Linux computers.

The nested surface method works well with continuous datasets that contain multiple structures. We expect that the method will also work fine for categorical data. In this case, the ordering of dimensions and other parameters may be significant and can yield different visual results.

Usually, high number of observations overloads scatter plots. In contrast, nested surfaces produce nice results. The visualization is not affected by a high number of observations and it is continuously improving as the number of observations increases.

The computation of nested surfaces for the purpose of data mining has only received scant attention. Mostly surfaces have been investigated in connection with advanced visualization techniques, such as rendering, lighting, transparency, or stereoscopy [5, 8, 9, 13]. These approaches focus on methods and data structures related to visualization aspects. Our goal is the computation of the defining structure of surface that emphasize the structure of the data.

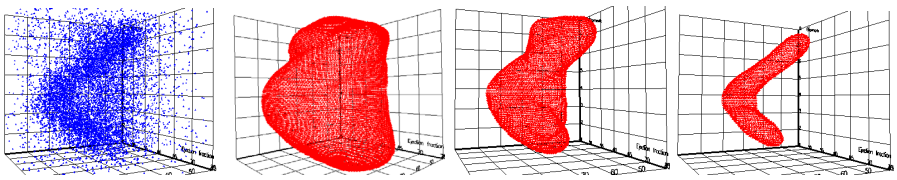
The paper proceeds as follows. We motivate our method in Section 2. Section 3 provides background material about probability density functions (PDFs), kernel estimations, clustering, and outliers. Section 4 defines surfaces. Section 5 gives algorithms for computing PDF estimates, level grid surfaces, and level contour surfaces. The algorithms are evaluated in Section 6. Section 7 discusses experimental results. Section 8 summarizes the paper and points to future work.

2 Motivation

Scatter plots are used to find structures in data. These structures are usually described as an accumulation of points. Scatter plots are good in getting a first impression of the data set, but they have a number of disadvantages. It is hard to understand very dense regions since data points hide each other. On the other hand it is also difficult to investigate sparse regions since data points in sparse areas are easily perceived as noise.

To illustrate our method we use the Spiral–Line data set presented in Figure 1(a). The data set consist of a vertical line in the middle (4'000 points), a spiral curve around the line (4'000 points) and uniformly distributed noise (2'000 points). The data points around the spiral curve form the most dense and notable region. Since the data points around the vertical line have a higher spreadness it is easy to treat it is as noise and not pay attention to it.

Figures 1(b) to 1(d) present the surfaces for different density levels α. Figure 1(b) shows the surface for the lowest density level. This Figure can be used for the detection of outliers. Figures 1(c) and 1(d) show surfaces for higher density levels. Together with Figure 1(b) they emphasize the structure of the data set. Note that the surfaces in Fig-



(a) Dataset (b) Surface for $\alpha = 1/10m$ (c) Surface for $\alpha = 5/10m$ (d) Surface for $\alpha = 9/10m$

Fig. 1: Spiral–Line DB and Associated Surfaces. m denotes the maximum density in DB

ure 1(c) clearly identify the vertical line *and* the spiral (the quality is much better on the monitor, see also Figures 5 and 6).

In contrast to scatter plot visualizations, surfaces do not deteroriate if the amount of observation increases. Nested surfaces are often easier to interpret than the raw data. Moreover multiple nested surfaces at different density levels facilitate the analysis of the data at different levels of detail. This gives the ability to explore the internal structure of data regions.

3 Preliminaries

3.1 Probability Density Function

Throughout, we assume that the data has been normalized to the three-dimensional unit cube, i.e., each coordinate falls into the [0,1] interval.

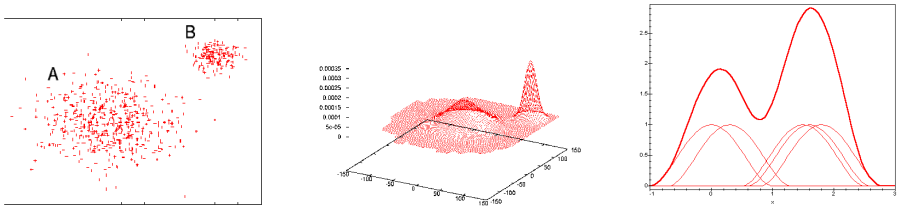
Definition 1. *(Probability density function) Let X be a 3 dimensional random vector with distribution function F. A 3-dimensional real value function f with*

$$F(x, y, z) = \int_{\infty}^{x} \int_{\infty}^{y} \int_{\infty}^{z} f(t, s, q)\, dt\, ds\, dq$$

is a probability density function (PDF).

Figure 2(a) shows a dataset with two clusters: A and B. The corresponding PDF is shown in Figure 2(b). The PDF shows the density of the dataset. Since the density of region A is lower than the density of region B the PDF value for region B is higher than for region A. The PDF also shows that region A is more spread than region B.

In general, we have to estimate the PDF because we are dealing with random datasets for which the true PDF is unknown. Different PDF estimates were proposed in the literature, with the kernel method being one of the most general ones [4, 2, 12, 11, 6]. The essence of the kernel method is that each observation increases the chances of having another observation nearby. Therefore, we draw a symmetric kernel with an area equal to 1 around each observation. Adding all kernels (cf. Figure 2(c)) yields an estimate for the PDF. To control the influence of one observation on the overall estimation a smoothing parameter h is introduced. The kernel estimate [10] for a a set of



(a) Dataset D (b) PDF for Dataset D (c) Kernel Addition

Fig. 2: Dataset and Corresponding PDF

observations, $(X_i, Y_i, Z_i), i = 1, \ldots, n$, at point (x, y, z) is defined as follows:

$$\hat{f}_K(x, y, z) = \frac{1}{nh^3} \sum_{k=1}^{n} K\Big(\frac{x - X_i}{h}, \frac{y - Y_i}{h}, \frac{z - Z_i}{h}\Big), \tag{1}$$

where K is a function (kernel) with $K \geq 0$, $\int K = 1$, and $K(x) = K(-x)$.

Various kernels K have been proposed in the statistical literature. Examples include square wave or Gaussian functions. It has been shown [12] that the accuracy of the estimation depends mostly on the smoothing parameter h and less on the choice of the kernel K. Parzen [2] showed that the smoothing parameter

$$h = h_{opt} = c(K, \sigma_1, \sigma_2, \sigma_3)/n^{-1/7} \tag{2}$$

minimizes the mean integrated square error (MISE):

$$\text{MISE} = \mathbf{E} \iiint (\hat{f}(x, y, z) - f(x, y, z))^2 \, dx \, dy \, dz. \tag{3}$$

c is constant for a given dataset and depends on the variance $(\sigma_1^2, \sigma_2^2, \sigma_3^2)$ of the random vector (X_1, X_2, X_3) and the kernel function K.

We assume that the estimation is implemented as a three dimensional data cube with g cells for each dimension. The choice of appropriate values for the size of the data set sample, n, and the number of grid lines, g, is described in detail in [1].

3.2 Clusters and Outliers

Density functions are also used to define clusters and outliers. Clusters and outliers are important characteristics of a dataset, and they are often used for data analysis. In the next section we will establish a relationship between clusters and surfaces. Let D be a set of 3 dimensional points, and let $(\mathbf{x}, \mathbf{x}^*] = \{\mathbf{x}t + \mathbf{x}^*(1 - t), t \in (0, 1]\}$ be an interval in the three-dimensional space.

Definition 2. *(Cluster) A cluster for a set of local maxima M of the density function f and threshold ξ is the subset*

$$C = \{\mathbf{x} \in D \mid \forall \mathbf{x}^* \in M \wedge \forall \mathbf{y} \in (\mathbf{x}, \mathbf{x}^*] : f(\mathbf{y}) \geq \xi\}.$$

Definition 3. *(Outliers) The points $O \subseteq D$ are* outliers *iff for all local maxima $\mathbf{x}^*$ of the density function f*

$$O = \{\mathbf{x} \in D \mid \forall \mathbf{y} \in (\mathbf{x}, \mathbf{x}^*] : f(\mathbf{y}) < \xi\}.$$

Thus, a cluster is a set that contains PDF center (maxima) points together with all surrounding points that "exceed noise level ξ".

4 Surface Definition

We use a topological approach to define a surface. Intuitively, a surface is a set of points iff the neighbourhood of any point is similar to a two–dimensional open disk. To define the resemblance with an open disk we use a homeomorphic (one-to-one, continuous inverse) function.

Definition 4. *(Elementary surface) Let f be a function that maps an open disc D^2 to a set of points S. S is an* elementary surface *iff f is homeomorphic.*

Definition 5. *(Surface) A* surface *is a connected set of points iff the neighbourhood of any point of the surface is an elementary surface.*

Next, we establish a relationship between the border of a cluster and a surface. A border is a set of points: $\partial C = [C] \backslash C^\circ$ where $[C]$ contains the limit points of C and C° contains the inner points of C. We show that ∂C is a surface by giving a parametrisation function that maps a disk D^2 into ∂C.

Theorem 1. *(Implicit function theorem) Suppose $f : R^n \times R^m \to R^m$ is differentiable in an open set around (u, v) and $f(u, v) = 0$. Let M be the $m \times m$ matrix given by*

$$M = \left(\frac{\partial f_i(u)}{\partial x_{n+j}}\right) \quad 1 \leq i, j \leq m.$$

If $\det M \neq 0$ then there is an open set $U \subset R^n$ that contains u and an open set V that contains v, such that for each $r \in U$ there exists $s \in V$ and $f(r, s) = 0$. If we define $g : U \to V$ as $g(r) = s$, then g is differentiable.

The implicit function theorem is a classical result and ensures the existence of a cluster boundary parametrisation. A proof can be found for example in [7].

Lemma 1. *Let f be a probability density function which has continuous derivative ($f \in C^1$), and C be a cluster for a set of maxima M and level noise ξ. Let grad $f(x) \neq 0$, $x \in \partial C$. Then ∂C is a surface.*

Proof. Notice that $\partial C = \{x \in D : f(x) = \xi\}$. Let $(a, b, c) \in \partial C$. Since grad$f \neq 0$ there is at least one coordinate x_i such that $\partial f / \partial x_i \neq 0$ at point (a, b, c). Then the implicit function theorem with $m = 1$ and $v = x_i$ proofs the lemma.

5 Algorithms

This section gives algorithms to compute nested surfaces: `Surface_GridPoints` and `Surface_GridLines`. Starting from the raw data, the first step is the estimation of the PDF. We scan the (sample of the) data set twice to estimate the PDF. The first scan is used to calculate the estimation parameters (cf. Equation (2)). The second scan is used to compute the actual PDF estimation. We use the Epanechninikov kernel [2], which is equal to 0 outside the area $t_1^2 + t_2^2 + t_3^2 \leq 1$. Thus, only observations that fall into the area $\{(t_1, t_2, t_3) : (t_1 - x)^2 + (t_2 - y)^2 + (t_3 - z)^2 \leq h^2\}$ influence the estimated PDF value at point (x, y, z).

Algorithm: PDF_Estimation

Input:
  Database with n observations: $(X[i], Y[i], Z[i]),\ \ i = 1, \ldots, n$
  Number of grid points in each dimension: g

Output:
  Data cube with PDF values on grid points: PDF

Body:
  1. Initialize PDF
  2. Calculate estimation parameters according to Equation (2)
  3. FOR $i = 1$ TO n DO
     3.1. Determine the set of PDF points $\mathcal{A}_g$ that are
          influenced by the data point $(X[i], Y[i], Z[i])$
     3.2. FOR EACH $(k, l, m) \in \mathcal{A}_g$ DO

$$PDF[k, l, m] = PDF[k, l, m] + K(\tfrac{k - X_i}{h}, \tfrac{l - Y_i}{h}, \tfrac{m - Z_i}{h})$$

The `Surface_GridPoints` algorithm calculates the border $B = \partial\{(x, y, z) : f(x, y, z) \geq \alpha\}$. The basic idea of the algorithm is to scan the PDF and compare each value against its neighbours: if the value is greater than α and there exists a point in the neighborhood that is less than α then the value is added to B.

Algorithm: Surface_GridPoints

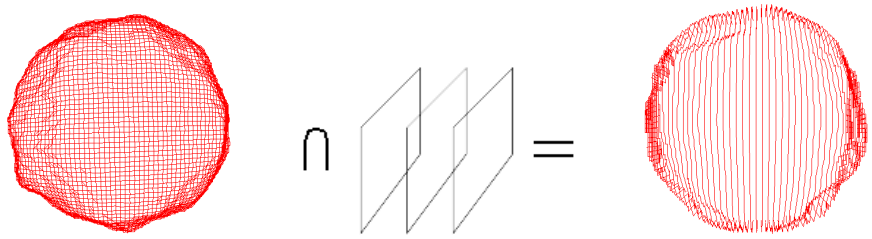
Input:
  Number of grid lines per dimension: g
  Data cube with PDF grid point values: PDF
  Density level: α

Output:
  Surface grid: B

Body:
  1. FUNCTION IsBorderPoint(PDF,i,j,k)
  2.   RETURN ($PDF[i, j, k] \geq \alpha$) AND ($PDF[i', j', k'] < \alpha$)
                for some $(i', j', k') \in (i + h_1, j + h_2, k + h_3)$
                where $h_1, h_2, h_3 = -1, 0, 1,\ \ |h_1| + |h_2| + |h_3| = 1$
  3. END FUNCTION

```
4. B = ∅
5. FOR i, j, k = 1 TO g DO
   5.1 IF IsBorderPoint(PDF, i, j, k) THEN B = B ∪ PDF[i, j, k]
```

The `Surface_GridLines` algorithm extends the `Surface_GridPoints` algorithm. The main idea of the algorithm is to draw contour curves on the surface. These curves, in turn, are calculated by intersecting a surface with cutting planes parallel to the XY, ZY, and ZX planes (cf. Figure 3).



(a) Surface (b) Intersecting planes (c) Vertical Grid-Lines-Surface

Fig. 3: Grid-Line-Surface

The idea of plane curve's calculation is presented in Figure 3. We scan the PDF values with a condition $i = i_0$ for ZY planes, $j = j_0$ for ZX planes, and $k = k_0$ for XY planes.

Figure 4(a) shows a cutting plane. Border points are shown as filled circles, inner cluster points as plus signs, and outer cluster points are not shown in the picture. The algorithm connects the border points to form a polygon curve.

(a) Connecting border points (b) Drawing directions (c) `Plane_Curve` method's output. Not valid directions are drawn in grey

Fig. 4: Curve computation in intercepting plane

For each PDF border point we are looking for PDF border points in the directions presented in Figure 4(b). Note, that we scan PDF from left to right, from bottom to top. Therefore, there is no need to draw lines to the bottom and to the left.

We make vertical and horizontal connections between border points. For diagonal we make additional checks. We do not draw diagonal line if there are two lines in its neighborhood (cf. Figure 4(c)). With this we avoid squares with crossing diagonals inside. The Individual steps of the ZY plain curve calculation are presented in the ZY_Plane_Curve algorithm.

Algorithm: ZY_Plane_Curve

Input:
  ZY plane number: i_0
  Data cube with PDF grid point values: PDF

Output:
  polygonal contour line on ZY plane at level i_0: $C = C_{i_0}^{ZY}$

Body:
  1. $C = \emptyset$, $i = i_0$
  2. FOR $j, k = 1$ TO g DO
     2.1 IF IsBorderPoint(PDF,i,j,k) THEN
       IF IsBorderPoint(PDF,i,j+1,k) THEN
         $C = C \cup \text{line}(i, j, k,\ \ i, j+1, k)$
       IF IsBorderPoint(PDF,i,j,k+1) THEN
         $C = C \cup \text{line}(i, j, k,\ \ i, j, k+1)$
       IF IsBorderPoint(PDF,i,j-1,k+1) AND
          ¬IsBorderPoint(PDF,i,j-1,k) AND
          ¬IsBorderPoint(PDF,i,j,k+1) THEN
            $C = C \cup \text{line}(i, j, k,\ \ i, j-1, k+1)$
       IF IsBorderPoint(PDF,i,j+1,k+1) AND
          ¬IsBorderPoint(PDF,i,j+1,k) AND
          ¬IsBorderPoint(PDF,i,j,k+1) THEN
            $C = C \cup \text{line}(i, j, k,\ \ i, j-1, k+1)$

Algorithm: Surface_GridLines

Input:
  Data cube with PDF grid point values: PDF

Output:
  Contour lines on the surface: C

Body:
  1. $C = \emptyset$
  2. FOR $i = 1$ TO g DO $C = C \cup \text{ZY\_PlaneCurve}(PDF, i)$
  3. FOR $j = 1$ TO g DO $C = C \cup \text{ZX\_PlaneCurve}(PDF, j)$
  4. FOR $k = 1$ TO g DO $C = C \cup \text{XY\_PlaneCurve}(PDF, k)$

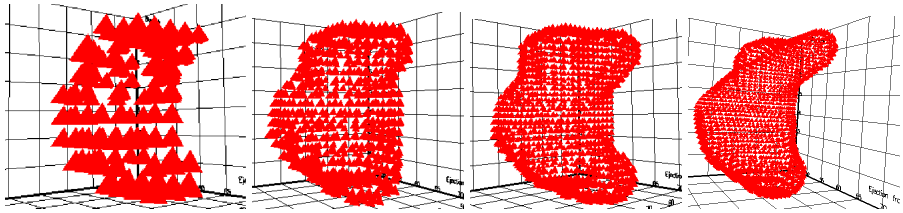
Note, that in order to include a 3D picture into the paper we have to project it into 2D. We use the Surface_GridPoints method to illustrate surfaces on a 2D devices while we use the Surface_GridLines method to illustrate surfaces in immersive 3D environments.

6 Evaluation

This section evaluates the quality of the algorithms numerically and visually. The experiments were calculated on the Pentium III 1GHz PC computer with 512MB of main memory running GNU/Linux OS with the gcc compiler.

6.1 Quality of the Surfaces

First, we evaluate the surface quality with respect to the number of grid lines. We use the three-dimensional scatter plot in Figure 1(a) and a single level surface for $\alpha = 1/10m$. In order to get a fair visual comparison of the influence of g on the quality of the surface we let the size of tetrahedra depend on g. It is chosen so that the tetrahedras visually are always near each other. Figures 5(a) and 5(b) show that $g = 10$ and $g = 20$ are not enough for a nice surface. There are too few tetrahedras at the ends of the spiral curve. As g reaches 30 the picture becomes detailed enough.



(a) $g = 10$ (b) $g = 20$ (c) $g = 30$ (d) $g = 50$

Fig. 5: Cluster Surface for $\alpha = 1/10m$ for Varying Values of g

To quantitatively measure the quality of the surfaces we use Equation (4). It quantifies the average error we make at any point (i, j).

$$AE_S = \frac{1}{g^2 \max_{x,y,z} \hat{f}_g(x,y,z)} \sum_{i,j=1}^{g} |\hat{s}_g(i,j) - s(i,j)|, \tag{4}$$

s is the parametrisation function that maps the open unit disk to $\partial C_\alpha = \partial\{(x,y,z) : f(x,y,z) \geq \alpha\}$. Since s is usually unknown we replace it with $\hat{s}_{\bar{g}}$ with a large value for $\bar{g}$:

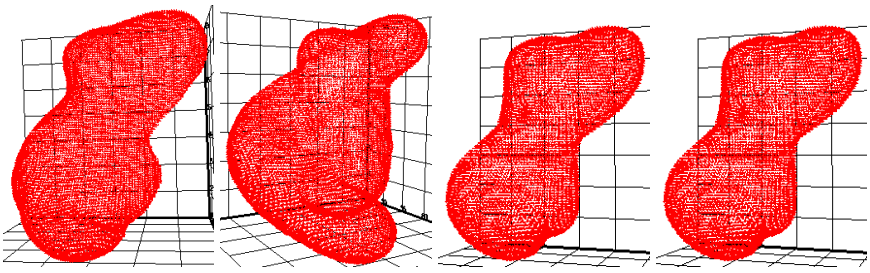
$$EAE_S = \frac{1}{g^2 \max_{x,y,z} \hat{f}_g(x,y,z)} \sum_{i,j=1}^{g} |\hat{s}_g(i,j) - s_{\bar{g}}(i,j)|, \tag{5}$$

Table 1 gives the numbers for EAE_S with $\bar{g} = 100$. The result shows that the error is very low. It is below 1% if the number of grid lines is greater than 30.

α	$g = 10$	$g = 30$	$g = 50$
1/10m	0.0289	0.0083	0.0045
5/10m	0.0249	0.0071	0.0038
9/10m	0.0069	0.0011	0.0005

Table 1: The EAE_S Error for Different Number of Grid Lines

Figure 6 presents the the impact of the size of the database sample on the surface quality. The figures show that $n = 10'000$ is sufficient for a nice surface. Note that Figure 6(b) is shown from a different prospective. This perspective emphasizes the unevenness of the vertical line.



(a) $n = 1'000$ (b) $n = 5'000$ (c) $n = 10'000$ (d) $n = 100'000$

Fig. 6: Cluster Surface for $\alpha = 1/10m$ and Varying Values of n

6.2 Space and Time Complexities

With the number of dimensions fixed at three the number of grid lines g and the number of observations n has the biggest impact on the computation time. We use the dataset from Figure 1(a) to measure the time to compute the surfaces.

Table 2 shows the times in seconds to calculate the surfaces from the raw data. A detailed analysis of the runtime reveals that the vast amount of the time is spent for the PDF estimation. Less than 1 second is needed to calculate a surface. Thus, to improve the performance it is possible to pre-compute and store PDFs. Table 3 shows that the size of the PDF is small and not usually relevant when compared to the size of the original database.

n	$g = 10$	$g = 30$	$g = 50$
1'000	<1	2	9
5'000	<1	6	24
10'000	<1	8	34
100'000	3	37	130
1'000'000	24	164	547

Table 2: Computation Time for Different Number of Grid Lines and Data Points

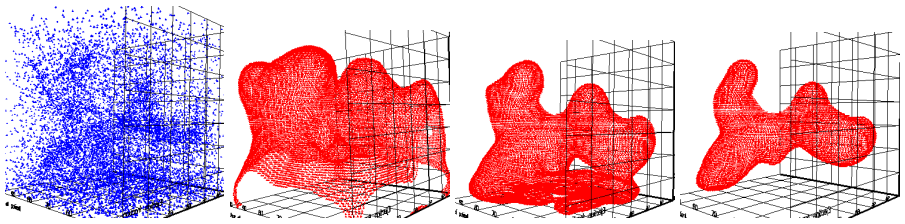
g	10	30	50	100
Size	4	108	500	4'000

Table 3: Size of PDF in KB

7 Experiments

This section illustrates our methods on an artificial data set (cf. Figure 7(a)). We show nested surface grids and offer an interpretation. Note that the visual information in the printed images is somewhat limited as three dimensions have to be projected into two. Also nested surfaces have to be shown in figures side-by-side. The reader may download and install the 3DVDM system to experiment with the surfaces.

The data set contains three data structures: 1) points, which are spread around randomly generated polygonal line, 2) a 3D structure defined in terms of a simulated random variable: (uniform(0,1), uniform(0,1), and 3) uniform noise in the data set.



(a) Dataset (b) $\alpha = 1/10m$ (c) $\alpha = 3/10m$ (d) $\alpha = 5/10m$

Fig. 7: Artificial Dataset

It is hard to understand the structure from the scatter plot (cf. Figure 7(a)). The nested surfaces in Figures 7(b) to 7(d) emphasize and clarify the structure.

8 Summary and Future Work

In this paper we defined and evaluated nested surfaces for the purpose of visual data mining. Since humans perceive surfaces much easier that individual observations. This approach to mining data gives an ability to investigate the structure of the data data more easily than the scatter plot of the data itself. In addition, surfaces clarify very dense and sparse (and the combination of both) regions of the data set. That gives an ability to detect arbitrary shaped structures in a data set.

The surface calculation is based on an estimated PDF which makes our method independent of the data. The PDF estimation is implemented as a three-dimensional cube. We presented empirical results that show that the space and time complexity is reasonable. It is possible to compute surfaces on the fly during data explorations. Real time interaction can be achieved by precomputing and storing small density estimates.

In the future we will refine our methods to find curves and 2-D structures in a data set. It would also be interesting to experiment with the display of visually advanced surfaces that use transparency, light and shading.

Acknowledgments

This work is supported in part by the Danish research council through grant 5051-01-004. We greatly appreciate the comments of the 3DVDM project members and our partners from Nykredit. We thank the VRCN for the opportunity to work with immersive visualizations.

References

1. A. Mažeika, M. Böhlen, P. Mylov. Density Surfaces for Immersive Explorative Data Analyses. To appear in SIGKDD Workshop on Visual Data Mining, 2001.
2. B.W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, 1986.
3. D. A. Keim and H.-P. Kriegel. Visualization Techniques for Mining Large Databases: A Comparison. *Transactions on Knowledge and Data Engineering, Special Issue on Data Mining*, 8(6):923–938, 1996.
4. D. W. Scot. *Multivariate Density Estimation*. Wiley & Sons, New York, 1992.
5. E. J. Wegman, Q. Luo. Visualizing Densities. Technical Report Report No. 100, Center for Computational Statistics, 1994.
6. G. C. van den Eijkel, J. C. A. Van der Lubbe, E. Backer. A Modulated Parzen-Windows Approach for Probability Density Estimation. In *IDA*, 1997.
7. G. E. Bredon. *Topology and Geometry*. Springer–Verlag, 1995.
8. H. Shen and C. Johnson. Sweeping Simplicies: A Fast Isosurface Extraction Algorithm for Unstructured Grids, 1995.
9. J. Wilhelms and A. Van Gelder. Octrees for Faster Isosurface Generation. *ACM Transactions on Graphics*, 11(3):201–227, 1992.
10. L.Devroy, L.Gyorfi. *Nonparametric Density Estimation*. Jon Wiley & Sons, 1984.
11. M. Farmen, J.S. Marron. An Assesment of Finite Sample Performace of Adaptive Methods inDensity Estimation. In *Computational Statistics and Data Analysis*, 1998.
12. M.C. Jones M.P. Wand. *Kernel Smoothing*. Chapman & Hall, 1985.
13. W. Lorensen and H. Cline. Marchine cubes: A high resolution 3d surface construction algorithm, 1987.

Methods for Visual Mining of Data in Virtual Reality

Henrik R. Nagel, Erik Granum, and Peter Musaeus

Lab. of Computer Vision and Media Technology, Aalborg University, Denmark
{hrn, eg, petermus}@cvmt.dk

Abstract. Recent advances in technology have made it possible to use 3-D Virtual Reality for Visual Data Mining. This paper presents a modular system architecture with a series of tools for explorative analysis of large data sets in Virtual Reality. A 3-D Scatter Plot tool is extended to become an "Object Property Space", where data records are visualized as objects with as many statistical variables as possible represented as object properties like shape, color, etc. A working hypothesis is that the free and real-time navigation of the observer in the immersive virtual space will support the chances of finding interesting data structures and relationships. The system is now ready to be used for experiments to validate the hypothesis.

Keywords: data exploration, visualization, perception

1 Introduction

Visual Data Mining traditionally uses 2-D graphics or very simple 3-D graphics to visualize results on ordinary monitors. Real-time interaction is only used to a limited extent. One of the reasons for this has been the lack of adequate hardware for visualizing complex graphics. However, during the last years graphics cards have doubled their speed every half year. Together with advances in supercomputers, user interface technology, etc., this has made it possible to view large and complex visualizations in immersive 3-D Virtual Reality (VR), with real-time interaction. Thus today it is possible to design new methods for visualizing the often very large and complex databases. These methods enable analysts to perceive data also from the inside out, thereby, hopefully, adding extra opportunities to recognizing patterns, clusters, etc.

Some statisticians have experimented with extending the methods traditionally used in statistical data exploration, to work in VR. An example of this is XGobi/VRGobi [12] with a VR version of "The Grand Tour" [2, 4, 18]. These methods were originally designed for ordinary workstations with standard 2-D monitors and are often used to analyze relative small data sets. It may therefore be possible to improve upon these methods using the enhanced visualization, processing, and interaction facilities available for VR today. Except for systems like TIDE (Tele-Immersive Data Explorer) [15] and, DIVE-ON (Data mining in an Immersed Virtual Environment Over a Network) [1], little scientific research

has been done on how best to use VR in visual data mining. TIDE focuses on the use of collaboration and resource sharing to mine large data sets in VR, while DIVE-ON focuses on interaction with a Virtual Data Warehouse over a network.

In the autumn of 1999 a new Virtual Reality center now called "VR Media Lab" was inaugurated at Aalborg University in Denmark. Among its facilities are a 3-D Power Wall, a 160 degree Panorama, a 6-sided cubic CAVE, and a 16 processor SGI Onyx2 with 6 graphics pipes. A research project called "3-D Visual Data Mining" (3DVDM) was also initiated to study how VR may be used in Visual Data Mining. The project group consists of persons with expertise within the scientific fields: databases, statistics, perceptual psychology, and visualization.

This paper presents work of the visualization group of the 3DVDM project. It contains a discussion of system architecture, data exploration tools, and experience with using natural human perceptual skills for mining data in VR.

1.1 Motivation and Tasks

With the new high-end VR technology, it is possible to let users perceive visual worlds from inside 3-D immersive environments. Users are able to recognize whether objects appear large and far away or small and close by. The scenes look realistic, since close objects stand out in space in front of the users. It gives possibilities for exploiting natural human perceptual skills in finding unspecified patterns in a visual 3-D representation of data.

Tracking of a user's position, orientation, and pointing gestures, allows the computer to calculate the position and orientation for the right visualization. This means that the user in real-time can move around objects or clusters of objects close to him, and see them from all sides. Virtual reality thus allows a whole range of new possibilities for expressing and inspecting information.

Visual Data Mining projects aim at allowing analysts to explore large and complex data sets. To investigate the possibilities available in VR, our tasks are:

1. Developing a modular system architecture suitable for empirical studies in a research context.
2. Investigating state-of-the-art software methodologies for optimally exploiting VR technology, allowing as many degrees of freedom as possible.
3. Developing a visual language suitable for expressing information in VR. This requires investigations on how to artificially generate perceptual sensations in VR, which optimally exploit the excellent faculties of human perception, for analysis of information content and structure of data.
4. Developing new VR data exploration methods that make use of our findings.

2 The Visualization System

3DVDM is also the name of our software system for exploring large databases in VR. The aim of the software system is to accommodate new methods for data

exploration, that makes full use of supercomputing, VR visualization, real-time interaction, human perceptual skills, etc.

In figure 1 is shown the general approach adopted in this research project for visualizing representations of data from databases.

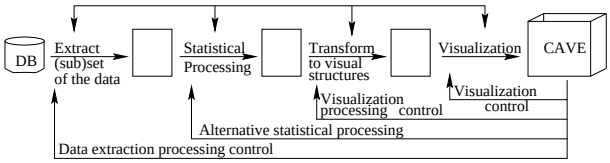


Fig. 1. Data Flow and Interaction Patterns

The system contains different data processing modules in a pipeline with the possibility of feedback from the user to each module.

First database technology extracts a relevant subset of the data in a database, and produces an easily accessible internal database, which is passed on for statistical processing. Data is then transformed into an equivalent symbolic graphical representation. This data format is independent of specific hardware and software requirements. Last step is to transform this data to polygons, which are rendered in a 3-D space.

The following sections contain a description of the project in each of the 3 areas: system architecture data exploration tools, and visual perception

2.1 Design Goals

It is expected that several different transformations of data will be invented during the project's lifetime. The system architecture has been designed in a highly modular way to allow easy addition and substitution of functionality by specifying a set of formal interfaces between the modules. Existing modules are therefore completely compatible with modules designed months or years later, provided that all modules use the same interface. New interfaces can easily be added to the system, if needed.

The primary goals for the 3DVDM system, were:

1. Highly modular programming
2. Automatic handling of data flow between modules
3. Automatic handling of process flow between modules
4. Structural flexibility to implement most kinds of modules.
5. Interface rigidity to ensure compatibility between modules

2.2 System Architecture

The approach taken was to design an object-oriented framework, with graphs and nodes as the basic structuring mechanism. To create an application a number

of nodes are connected in a graph, so that the data generated as output of one node is sent to the input of another node. Each node acts as a tool that contains the implementation of a method devised by participants in the project.

The nodes can be divided into source nodes, mapper nodes, and sink nodes. A source node receives no input, but produces data that is subsequently processed by other nodes. Data is represented internally as objects. Mapper nodes receive an input data object, and produce an output data object. This could be a node that filters data according to user defined criteria. A sink node is a node that receives input data object, but does not produce any output data object. These nodes produce the result of the program by other means, such as visualizing graphics in a VR arena.

Graphs transport data between nodes and execute the nodes in the correct order. They could be considered the skeleton of the program. In the original design of the system architecture, 3 kinds of graphs were included:

- **Sequential graphs** execute their nodes in sequential order, starting with the source nodes in the graph, and ending with the sink nodes.
- **Thread graphs** execute their nodes in parallel using e.g. POSIX threads. Data objects are passed from one node to another using shared memory. This kind of graph is particularly useful for allowing parallel execution on a supercomputer.
- **Process graphs** execute their nodes in parallel processes, using popular message passing protocols such as Message Passing Interface (MPI), or Parallel Virtual Machine (PVM). These kinds of graphs are particularly useful when executing programs on clusters of PC's or workstations.

In the current version of the software only sequential graphs are implemented.

2.3 Data Preparation

One of the main differences between ordinary statistical data exploration and data mining is the amount of data being analyzed. While it is normal in the former to work with databases containing only a few 100 records, the latter concerns much larger databases. It is thus necessary either to use a data exploration tool, which can handle large amounts of data, or perform analysis on a sample of the data. The system supports both methods by defining a parameter called "Coarseness". If its value is n, where $n \geq 1$, then 1/n part of the data is extracted. The extracted data is stored in an easily accessible internal database. Random sampling is not yet supported.

The internal database may be passed through a filter to extract a subset of the data using user-defined rules such as $Age \geq 20$ and $Age \leq 80$.

After this, selected statistical variables are mapped into visual properties like: position, color, shape, size, and pose. This allows subsequent data exploration tools to perform a visual analysis of the data based on analyst's choice of variables.

2.4 VR Visualization

Visualization is done by creating lists of object-vectors. To each visual object corresponds an object-vector. Each object vector contains only the minimum necessary information about a visual object. Object-vectors have the advantage of being easy to process, and independent of special software and hardware requirements.

For the rendering part, SGI's OpenGL Performer has been chosen as the basic 3-D graphics toolkit in the 3DVDM system. OpenGL Performer exists on SGI's Irix based computers, and on Linux. It provides real-time 3-D graphics, with automatic multi-processing in the low-level parts of the rendering system.

Performer stores the data that define virtual worlds in scene graphs. A scene graph includes low-level descriptions of object geometry and their appearance, as well as higher-level spatial information, such as positions and object transformations. The object vectors are transformed to polygon data, and stored in a Performer scene graph.

This scene graph can be rendered by Performer itself, in which case one can view the visualizations on ordinary monitors. However, one can also choose the combination of OpenGL Performer and VRCO's CAVELib VR toolkit, in which case the visualizations are done in 3-D virtual reality arenas. The latter gives analysts the benefit of being immersed in data, and thus being able to study local phenomena with a higher degree of detail from any viewpoint and in the context of the full data set.

3 Perception of Visual Cues

The task in Visual Data Mining is to extract and analyze as much interesting information as possible. This entails encoding and processing of visual stimuli by the human perceptual system.

What guidelines can inform our construction of data exploration tools in order to facilitate visual data exploration? One guideline is that VR-displays for visual data exploration are constructed with perceptual cues, which pop-up pre-attentively in order that the encoding happens quickly and reliably [16, 17]. Even though visual data mining is mainly about exploring data, one prerequisite for such exploration can in some cases be that data is read accurately from the VR-displays. Here we can be informed by guidelines concerning perception of 2-D graphs or traditional displays, such as dashboards [6, 11, 3].

As mentioned in section 2.3 the data preparation in the 3DVDM-system is performed in such a way that a single geometric shape represents an observation in a data set. The parameters in the static object property space are: Position, pose, size, shape, color and texture. In the following we will briefly present some thoughts on the potential use of these perceptual parameters in terms of mapping statistical variables.

Position Position is a fundamental parameter, which determines the relative position of objects. Position can be a strong pop-up cue (e.g. close objects tend

to be grouped together according to the gestalt law of proximity). Furthermore, stereoscopic depth - the distance to the object from an observer - is a pop-up cue. When the task is merely to read off data in displays such as in a 2-D histogram, position has been found to be the most efficient way to map data. With regards to a 3-D Scatter Plot, the perceptual system can discriminate fine changes in position and position can be used to map three continuous statistical variables.

Pose The pose - or spatial orientation - of an object is often perceived to be upright in relations to the observer's interpretation of vertical and horizontal in the visual space. Vertical or horizontal visual stimuli are perceived more efficiently than tilted visual stimuli [9]. Co-linearity of line structures is a pop-up phenomenon. Pose can theoretically be used to map up to two continuous variables.

The use of pose requires coordination with the use of the shape property, as orientation characteristics should be maintained for all shape-variations used.

Size The size of an object is potentially important to consider since a large object stands out from a population of smaller ones, and groupings of data points (in 2-D) which occupy less area is perceived as figure whereas regions with bigger area is perceived as ground. In traditional displays, objects should have no more than three different sizes in order to be efficiently encoded. In VR colored objects should not be too small, e.g. if the color-difference is in the yellow-blue direction the smallest size should be larger than half a grade of the visual angle.

In the 3DVDM system we have so far found it useful to have "frozen" object size to be constant by some parameter (e.g. volume). This reserves size for use by the observer for depth perception. Ambiguities regarding statistical information and distance to object should thus be avoided.

Shape Symmetric shapes are often thought to be encoded and processed more efficiently than non-symmetric shapes [7]. The contour of the shape can have pop-up qualities since the length - and width - of a line and curvature are pop-up phenomena. Furthermore, shapes with added marks work as pop-up stimuli. For instance a dot added to a square in a population of squares will make that particular square pop-up. The perceptual system clearly differentiates between topologically different objects, such as a ring with a hole and a ring without a hole, but not between topologically equivalent, such as triangle and square, square and circle, or triangle and circle [5]. Up to 15 different shapes can be distinguished in traditional displays, but no more than five different shapes should be used. This guideline seems to apply for all the 3DVDM data exploration tools, but the project only use 3 shapes so far. In order for the human perceptual system to notice a difference, distortions in the length of a shape, as measured horizontally or vertically, are not perceived if the distortion is less than 1.4% of the original length. Given the limits in number of shapes that can be efficiently encoded, shape can be used to map one categorical variable.

The data exploration tools developed so far use object shape as one or more visual properties to be varied parametrically or via fixed categories (like cube, tetrahedron etc.). Similar shapes tend to be grouped together (gestalt law of similarity).

Color The color (both hue and saturation) of objects can act as particularly strong pop-up feature. E.g. in a cluttered visual space with heterogeneous objects, the observer detects specific objects faster knowing in advance the color of the object as opposed to its size or shape. The pop-up effect is enhanced when objects are colored with a black rim on a white background or a white rim on a black background. Perceiving visual stimuli in VR, more than 6 colors are easily confused. Generally it is advised against using color as a continuous variable for two reasons: First due to limits in the human perceptual system in distinguishing accurately between hue, saturation, or brightness. Second due to the fact it does not unequivocally make sense to say that one hue is more or less than another (e.g. green is not more or less than red). In the 3DVDM system color can in fact be used to map continuous variables.

Texture Texture can e.g. be defined according to granularity, orientation and pattern [19]. The texture of an object aids the observer in determining the object's pose and shape. Texture can theoretically be used to map one or more continuous variable.

Dynamic Object Properties In the current system blinking is also an option as a dynamic object property. Blinking is particularly appropriate for drawing attention to alert signals, but has been shown to tire the observer [7] and must, therefore, be used with caution.

Spatial Distance Metric We are currently developing design rules for constructing VR-visualizations for visual exploration by adopting experience from perceptual psychology [10]. First we establish a spatial distance metric on the basis of maximal object size as the basic spatial unit. This conveniently allows a scaling of the visualization to refer to perceptually relevant measures. We also evaluate all object properties in terms of distance range within which variations of the visualization of the properties can be distinguished for individual objects.

We suggest initially the following upper bounds for the range of three variables:

Texture 25 spatial distance units
Shape 50 spatial distance units
Color 100 spatial distance units

The potential importance of these different upper bounds is that the perceptual grouping and the perceptual pop-up phenomenon on the basis of these properties are correspondingly bounded spatially, relative to the observer. Hence

a statistical variable encoded as texture properties will only work informatively in a relatively close neighbourhood as defined by the three variables currently used as positional variables. If the figures above hold, color may offer a potential for perceptual structuring in a neighbourhood up to 16 times larger. A more thorough investigation into these relationships is required, as they reveal some important relationships for the mapping between statistical variables and object properties.

4 Results

We have designed and implemented a software system that can be used for conducting experimental research on new methods for Visual Data Mining in VR using, e.g. the object property space.

4.1 Data Exploration Tools

We have until now designed and implemented the following data exploration tools:

3-D Histogram The 3-D histogram tool divides the space into cubes in a coordinate system with 3 axes. 4 variables from the data set are used for the visualization. 3 variables are mapped to position. The average value of a 4th variable over the records that fall inside a cube, is mapped to the color attribute of the cube as shown in figure 2.

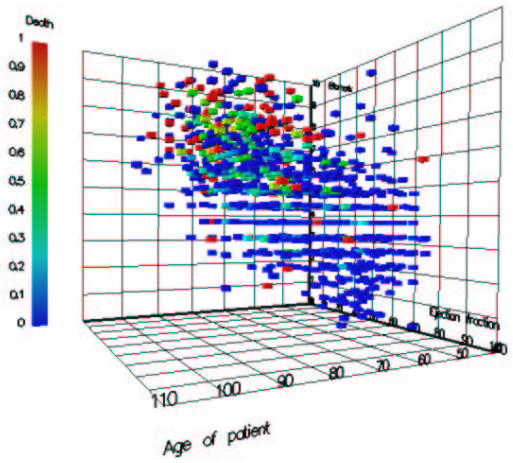


Fig. 2. Visualizing 4 statistical variables.

The size of each cube is here used as an additional dimension for showing the number of counts in them as illustrated in figure 3.

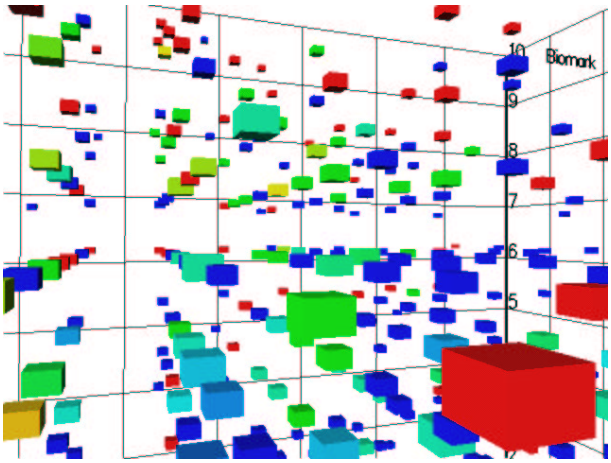


Fig. 3. Count is mapped to the size of each cube.

This tool is substantially more useful than an ordinary 2-D histogram, since it utilizes 3-D space more effectively. One can take a closer look at a subset of the data by "flying" into the middle of the data to study a phenomena there. These kinds of visualizations have also been explored in, e.g. DIVE-ON [1].

3-D Scatter Plot The 3-D Scatter Plot tool maps each data record as a data point in a 3-D coordinate system of continuous variables, see figure 4.

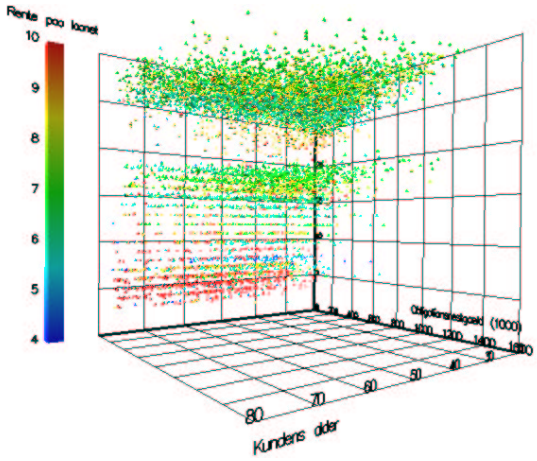


Fig. 4. Scatter Plot

A data point is illustrated as an object with the minimal number of surface polygons, a tetrahedron with 4 surfaces. To maintain the smooth real time response our system can handle up to 80.000 polygons, which allows it to visualize about 20.000 data points simultaneously and still have smooth visualization when changing viewpoint. The data points may be colored to visualize one more variable.

Initially the 3-D Scatter Plot just gives us spatial resolution compared to the 3-D Histogram. But this higher resolution allows further exploration of the navigation facility in virtual space. One may have a close look at a local configuration of data points and smoothly change to alternative viewing direction and/or viewing distance, and hence gradually obtain a global view and observe local configurations in the large context.

Another possibility is to calculate a surface map of the data. A (very) simple "Kernel density estimate" function is used for the calculations as shown in figure 5. More work has to be done to make this a useful tool.

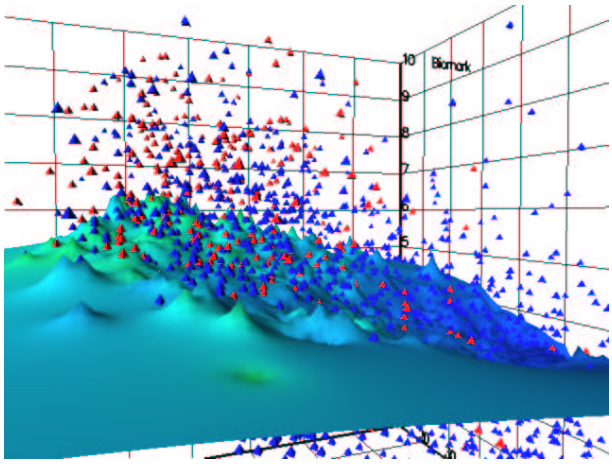


Fig. 5. Scatter Plot with surface.

It is also possible to "highlight" some of the visual objects. This is done by specifying an expression exactly as when choosing a set of filters. The highlighted records are visualized as visual objects blinking between their original color and white.

3-D Object Property Space In an attempt to map a larger number of statistical variables into the visualized world we have extended the 3-D Scatter Plot tool to become the "Object Property Space". The objects visualizing the data points may be given various visual properties that may illustrate other variables as exemplified in figure 6.

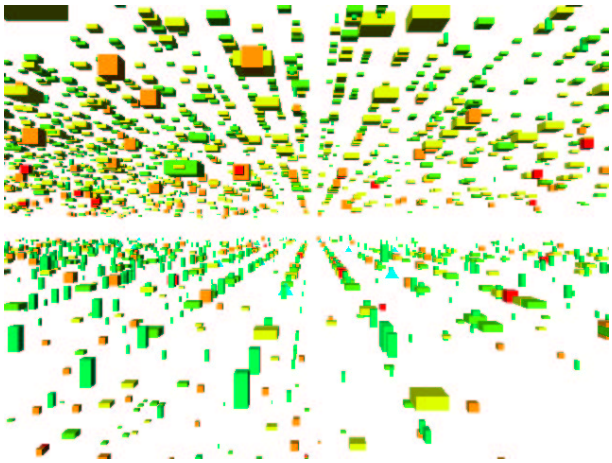


Fig. 6. A look inside Object Property Space, using position, color, shape, and size to represent statistical variables.

In principle a large range of possibilities emerge when taking this approach. Visually perceivable object characteristics are form, size, surface texture, and/or color and object orientation. These are all static properties and we may add animations that make objects vibrate or rotate with different amplitude, frequency and phase.

Taking this line of thinking further one may also let one of the variables drive a temporal development of the visual space. In a simple version a time series of "snapshots" may be visualized and in the more advanced version, for which methods still need to be developed, a continuous temporal development may be visualized. At this stage of the project we have implemented the use of object form, size, orientation and surface color as well as the snapshot series.

3-D Scatter Plot Matrix The 3-D Scatter Plot Matrix tool allows one to view multiple, small 3-D Scatter Plots simultaneously, making it possible to obtain an overview of a data set with alternative combinations of variables used for spatial position, see figure 7.

To maintain smooth real time interaction, fewer data points are used in each of the small Scatter Plot. Since the coordinate systems are relatively small, the tetrahedra are only shown in white. The highlight function also works in this plot. The highlighted tetrahedra are shown in magenta.

This tool is useful, when deciding which variables to use for the bare spatial distribution of data points (objects) in a full Scatter Plot.

3-D Scatter Plot Tour The 3-D Scatter Plot Tour tool is equivalent to the 3-D Scatter Plot Matrix tool, except that it only shows one coordinate system at the time. All possible unique combinations of the selected variables are shown

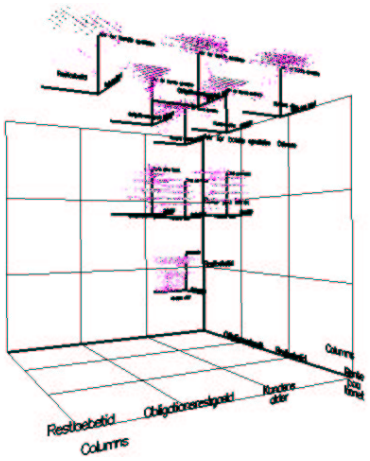


Fig. 7. Scatter Plot Matrix.

as positional variables in snapshots of 5 seconds each. It is possible to pause the animation at any given snapshot and navigate around before next combination is visualized.

The main advantage of this visualization in comparison to the "3-D Scatter Plot Matrix" is that much more data points can be used in each Scatter Plot. This makes it possible to add more visual cues to each data point.

4.2 Use and Performance

The 3DVDM system uses GNU's General Public License and is publicly available on the Internet [13]. It has been downloaded by users from international research institutions and bug-reports have been received. So far, however, it has not been used to any large extent by the public.

The system has automatic installations scripts for both SGI Irix computers, and Linux computers and it has a tool for loading data. However, configuration of CAVELib has to be done individually for each VR arena to be used.

Parameters are specified through a user-friendly X-Windows menu. The system also allows real-time interaction through VR input devices.

Flexibility One of the main advantages of the system is the ease with which it is possible to extend the system with new modules. Developers can concentrate on working with their modules, without interfering with the work of other developers.

Performance Loading of data from databases is done with MySQL, and is considered to be relative slow compared with other applications.

Currently only one processor in a multiprocessor computer is used for statistical processing.

However, Performer and CAVELib use multiple processes for rendering the 3-D graphics on monitors, as well as in VR arenas. On our Onyx2 supercomputer this allows visualization of 100.000 tetrahedra simultaneously. If smooth real-time interaction is required then the limit is about 20.000 tetrahedra simultaneously.

Immersiveness Our preliminary experiences indicate that in particular the "object property space", where the analyst is immersed in data, gives possibilities of navigating around to find "new" interesting structures and relationships in data. Even the Panorama arena provides well for the immersive experience, and it has the advantage of allowing a larger group of more than 10 people to take part in and discuss the same visual analysis.

5 Discussion and Future Work

As of today, the data handling part of the system performs poorly compared with state-of-the-art, and it is insufficient for very large data sets. An interesting solution to this problem would be to implement parallel data extraction into the system.

The system should use multiprocessing to:

1. Increase performance.
2. Allow different time-consuming visualizations, e.g. snapshots, to be calculated simultaneously.
3. Make it possible for a user of the system to obtain response to a complicated question in real-time.
4. Experiment with dynamic object properties.

3-D menus, buttons, etc. are highly desirerable, as well as audio aspects of VR.

It should also be possible to map, e.g. the position of the user in real-time directly to a process performing a statistical calculation based on this information. This would also make it possible for a statistical process to control the users' relative position dynamically in real-time.

Dynamic particle systems have been known since 1983 [14]. An interesting experiment would be to investigate how useful such systems are for "object property space" visualizations concerning, e.g. dynamic object properties.

Visual perception is often made in reference to a solid ground. In particular we suggest that such a perceptually "solid ground" for depth perception in the nearest neighbourhood provides a good reference for exploiting object pose as an informative property. Thus the free navigation and browsing in the 3-D world may provide the opportunity for seeing the data (objects) from arbitrary viewpoints and from arbitrary view-directions. This might aid the analyst in finding

visual events of potential interest, and it is in particular important for exploiting the information encoded in the pose-parameter. Seeing along their longitudinal versus perpendicular view could give cues about perceptual grouping hinting at clusters and structures.

We plan to carry out experiments on perceptual tasks in relation to the mentioned objects properties. Hereby we hope to test our working hypothesis that visual data mining is facilitated in immersive virtual environments.

6 Conclusion

This paper presented work in progress, with emphasis on a description of an approach to Visual Data Mining in 3-D immersive environments. The project aims at investigating and hopefully verifying that current immersive visualization technology in combination with the human perceptual capabilities provide for a new scope of explorative data analysis.

A flexible, maintainable system architecture was presented, as well as several methods for exploring data in VR. Our first findings concerning the enconding of a larger number of statistical variables and the use of human perceptual skills in the field of Visual Data Mining in VR are promising.

Acknowledgements

We gratefully acknowledge the support to the 3DVDM project from the Danish Research Councils, grant no. 9900103.

References

[1] A. Ammoura. Dive-on: From databases to virtual reality. *ACM Crossroads Database Special Edition*, 7(3), 2001.

[2] D. Asimov. The grand tour: A tool for viewing multidimensional data. *SIAM. Journal of Science and Statistical Computing.*, 6:128–143, 1985. (original paper, 2-D grand tour).

[3] K.R. Boff and J.E. Lincoln. *Engineering data compendium: human perception and performance.* Ohio: Harry G. Armstrong Aerospace Medial Research Laboratory, 1988.

[4] A. Buja and D. Asimov. Grand tour methods: an outline. In *Computing Science and Statistics: Proceedings of the Seventeenth Symposium on the Interface*, pages 63–67, 1985. (2-D grand tour).

[5] L. Chenk. Topological structure in visual perception. *Science*, 218(4573):699–700, 1982.

[6] R.E. Christ. Review and analysis of color coding research for visual displays. *Human factors*, 17:542–570, 1975.

[7] E.G. Davis and R.W. Swezey. Human factors guidelines in computer graphics: A case study. *Man Machine Studies*, 18(2):113–133, 1983.

[8] George Eckel. *IRIS Performer Getting Started Guide.* Silicon Graphics, Inc., 1997.

[9] A. Friedman and D.L. Hall. The importance of being upright: Use of environmental and viewer-centered reference frames in shape discriminations of novel three-dimensional objects. *Memory and Cognition.*, 24(3):285–295, 1996.

[10] E. Granum and P. Musaeus. Constructing virtual worlds for visual explorers. In L. Qvortrup, editor, *Virtual Space Construction: The Spatiality of Virtual Inhabited 3D Worlds*. Springer, Berlin, 2001. (In press).

[11] E.J. McCormick and M.S. Sanders. *Human factors in engineering and design. (5th ed.)*. Mcgraw-Hill Book Company., 1983.

[12] L. Nelson, D. Cook, and C. Cruz-Neira. Xgobi vs the c2: Results of an experiment comparing data visualization in a 3-d immersive virtual reality environment with a 2-d workstation display. *Computational Statistics*, 14:39–51, 1999.

[13] The 3DVDM project group. http://www.cs.auc.dk/3DVDM/.

[14] W. T. Reeves. Particle systems - a technique for modeling a class of fuzzy objects. In *Proc. of SIGGRAPH '83*, 1983.

[15] N. Sawant, C. Scharver, J. Leigh, A. Johnson, G. Reinhart, E. Creel, S. Batchu, S. Bailey, and R. Grossman. The tele-immersive data explorer: A distributed architecture for collaborative interactive visualization of large data-sets. *Proceedings of 4th International Immersive Projection Technology Workshop, Ames, Iowa*, June 2000.

[16] A. Treisman. Perceptual grouping and attention in visual search for features and for objects. *Experimental Psychology: Human Perception and Performance.*, 8(2):194–214, 1982.

[17] C. Ware. *Information Visualization: Perception for Design.* Morgan Kaufmann Interactive Technologies Series, 2000.

[18] E. J. Wegman. The grand tour in k-dimensions. In *Computing Science and Statistics: Proceedings of the 22nd Symposium on the Interface*, pages 127–136, 1991. (general k-dimensional grand tour).

[19] L. Wilkinson. *The Grammar of Graphics.* Springer, NY, 1999.

Visual Data Mining Using a Constellation Graph

Tokihiko Niwa*, Kenji Fujikawa**,
Kazuyoshi Tanaka**, and Mayumi Oyama***
* Kwansei Gakuin High School
Uegahara 1-1-155, Nishinomiya, 662-8501 Japan
**Web System Support Center, Hitachi Systems and Services, Ltd.
***Kwansei Gakuin University, Center for Information & Media Studies
Uegahara 1-1-155, Nishinomiya, 662-8501 Japan
niwa@kwansei.ac.jp, ke-fujikawa@hitachi-system.co.jp,
kzy-tanaka@hitachi-system.co.jp, oyama@kwansei.ac.jp

Abstract. Software was developed to enable visual data mining of multivariate data using a constellation graph. Two applications of the new software are presented. A constellation graph is an effective way to display multivariate data on to a two dimensional plane. The value of the classification variable used just as it is or it classifies it into some levels. Then, the cluster on each level is made to be able to be identified using the color or the symbol. The operator can change the weight of the variable interactively and do the visual mining by seeing the change of the star on the graph. Moreover, our program computes the weight of the variable to separate each cluster. It is possible to presume the data, which cluster it belongs to. In addition, using a computer mouse, the operator can select an area of the graph to examine data in detail.

1 Introduction

The aim of this research is shown below:

1. It displays multivariate data using the constellation graph on the 2-dimension plane.

2. By changing weighting factor to each variable, being interactive from 0 to 1, the position and the movement of the star on the constellation graph can be examined.

3. Choosing one classification variable and classifying it into the level to distinguish between each level using the color and the symbol on the graph.

4. The condition and the change of the cluster, which has a value with the same level on the constellation graph, can be examined by changing the weight.

5. Calculating the weighting factor so that the within-cluster variance has the minimum value and inter-cluster has the maximum value.

6. Cutting off the cluster and the necessary part from the graph and it preserves them in the file and they can be used for another analysis.

7. By displaying the data that a classification variable is not proved in the same sample data on the constellation graph, the level can be presumed.

Visual data mining using a constellation graph isn't suitable for most data processing applications, due to the limitations of the graphical display. However, since a constellation graph is an effective way to display multivariate data onto a two dimensional plane. The program that we developed allows the operator to interactively change the weighting factors of the variable. As these values are changed, the display is automatically updated so that the operator can immediately see how changes affect the clustering of the data.

The weighting factor that is given to each variable determines the length of the vector for each variable drawn on the constellation graph. If it is not possible to position a star in a desired position, the operator must reconsider the data in question to determine whether there is a more effective way to mine the data. When a cluster of stars becomes very cluttered, it is possible to select that area for further investigation using a computer mouse. The data from this area is then saved in a new file that can be investigated separately. This has proven to be a very effective technique for visual data mining. Another methods to display multivariate data have been discussed [1] [2]. However, there were problems when the number of the variable increased, and it was not possible

to change the weighting factors. Especially, in case of the parallel graph, the pattern is different when changing the order of the display of the variable but the constellation graph isn't related with the order of the variable because it is a vector. And the number of the variable can increase the capacity of the computer as far as it permits.

2 The Constellation Graph

This chapter shows how to construct the constellation graph and how to calculate the optimum weighting factors so that the within-cluster variance has the minimum value and the inter-cluster variance has the maximum value on the constellation graph.

3.1 2.1 Constructing a Constellation Graph

A constellation graph uses multivariate table-type data. The output of this data is the observed value of a star on the half pie chart of the constellation graph. The stellar position is determined by an angle that is dependent on the variable and by a vector that is dependent on the weight of each variable. A connected graph is made and a star is displayed at the end of the graph. This is called a "constellation graph" as it is made up several stars, each representing a portion of the data. Constellation graphs were originally proposed in 1977 [3]. The means by which a general constellation graph is constructed is described in the Appendix.

3.2 Constructing a Constellation Graph with Weighted Variable

Having chosen a classification variable, the values of the weighting factors to be used are then based on either "what to read in the data" or "how to read the data". The cluster means the category value or the repartition value of the classification variable. To sort the data one can:

1. Sort by gathering clusters of related data, or
2. Sort using regression curves that were made by the clusters

Our program uses the first method. To accomplish this, we must first choose an average mark as a basing point and then determine the value of the average mark.

The classification variable n is divided into kinds of clusters. Next, we consider the terminal coordinates of the connection vector of the variable that consist of j kinds of variables, here equations (14) and (15) on Appendix are used.

$$\left(\xi_{ij}, r_{ij}\right)\left(i = 1,2,\cdots, m_j \quad j = 1,2,\cdots, n \quad N = \sum_{k=1}^{n} m_k \right) \qquad (\mathbf{1})$$

This distinguishes similar clusters of data, while at the same time ensuring that there is very little overlap between clusters. Equations (2) and (3) are used to determine the average of each cluster.

$$\overline{C_j} = \frac{1}{m_j} \sum_{i=1+m_{j-1}}^{m_j} r_{ij} \cos\xi_{ij} \quad , \quad \overline{S_j} = \frac{1}{m_j} \sum_{i=1+m_{j-1}}^{m_j} r_{ij} \sin\xi_{ij} \qquad (j = 1,2,\cdots, n \quad m_0 = 0) \qquad (\mathbf{2})$$

$$\overline{\xi_j} = \tan^{-1} \frac{\overline{S_j}}{\overline{C_j}} \qquad (j = 1,2,\cdots, n), \qquad \overline{R_j} = \sqrt{\overline{C_j^2} + \overline{S_j^2}} \qquad (j = 1,2,\cdots, n) \qquad (\mathbf{3})$$

The whole average mark is determined using equations (4) and (5).

$$\overline{C} = \frac{1}{N} \sum_{j=1}^{n} \sum_{i=1+m_{j-1}}^{m_j} r_{ij} \cos\xi_{ij}, \qquad \overline{S} = \frac{1}{N} \sum_{j=1}^{n} \sum_{i=1+m_{j-1}}^{m_j} r_{ij} \sin\xi_{ij} \qquad (j = 1,2,\cdots, n \quad m_0 = 0) \qquad (\mathbf{4})$$

$$\overline{\xi} = \tan^{-1} \frac{\overline{S}}{\overline{C}} \qquad (j = 1,2,\cdots, n), \qquad \overline{R} = \sqrt{\overline{C}^2 + \overline{S}^2} \qquad (j = 1,2,\cdots, n) \qquad (\mathbf{5})$$

Merdia [4] and Fisher [5] defined this average using circle coordinates. Our program uses this for the value of the average mark. Based on this average mark, it then calculates each data point and a vector to the average mark at the top of the circumference in the shooting shadow. The radius is from the center of the segment that links the starting point and an average mark with this circumference. Then, it sums the data in every cluster and the distance at which there was a shooting shadow using the average mark. In addition, it sums the distance between all the data and the average mark using equation (6).

$$Var_j = \sum_{i=1+m_{j-1}}^{m_j} \overline{R_j}(\xi_i - \overline{\xi_j}) \quad (j = 1,2,\cdots,n) \qquad Var = \sum_{j=1}^{n} \sum_{i=1+m_{j-1}}^{m_j} \overline{R}(\xi_i - \overline{\xi}) \qquad \textbf{(6)}$$

with $m_0 = 0$.

Now, we want to scatter the big clusters rather than the small ones. In other words, we must decrease the value of Var_j and increase the value of Var. To do this, we define a value of J that is determined by equation (7). To determine the best weighting factors to cluster every cluster, J should be minimized.

$$J = \sum_{j=1}^{n} \frac{Var_j}{m_j} \cdot \frac{N}{Var} \qquad \textbf{(7)}$$

3 Visual Data Mining using a Constellation Graph

The following example illustrates the visual data mining technique using a constellation graph. The example data that we have chosen are the familiar Iris data, which include five variables (the kind of iris, petal length, petal width, sepal length, and sepal width). There are three kinds of iris. The kind of iris is used as the dependent value and is colored with three different colors on the graph. The other values are used as the variables.

The constellation graph visual data mining system has two display forms. One is a data table view and the other is a graph view.

3.3 The Data Table View

The data table view is shown Figure 1. The input data can be loaded into the table from the keyboard or from a file saved in CSV format.

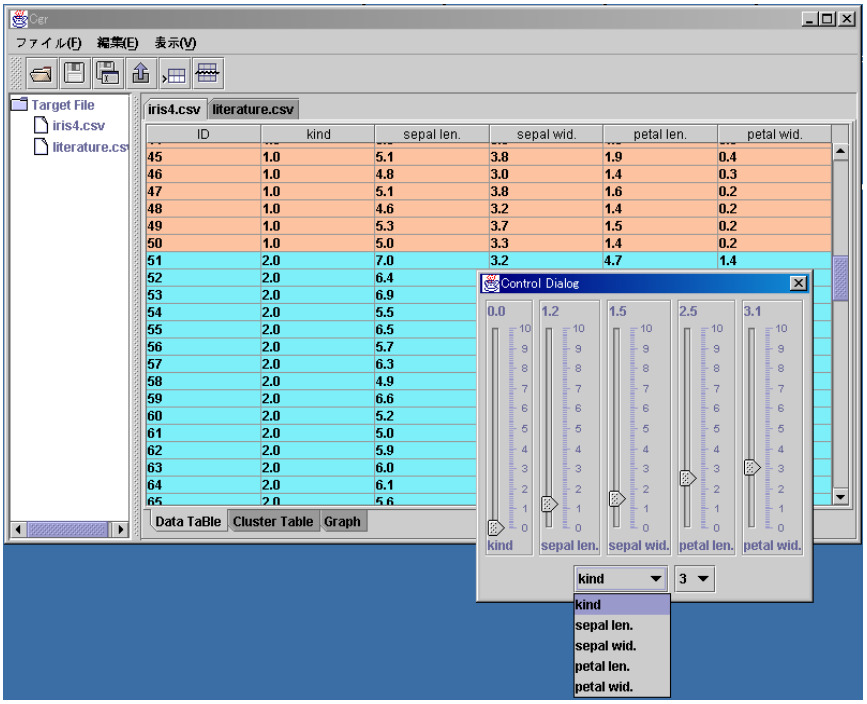


Fig. 1. Each line in the table is a variable. The variable at the left end of the line assigns the ID number of the data, which shows the number of the observation. Each row consists of the values of each observation.

Once the data input has been completed, the control dialog that displays the weight of each variable is displayed. Using the slide bars on the control dialog display and the mouse, the operator can change the weight of each variable. One variable is selected as the classification variable. Once the classification variable is chosen, the data table is sorted based on the range of the other variables. Observations with different values of the classification variable are displayed in different colors. For example, in Figure 1, the classification variable is “kind”, and three different colors are used to show the three

different kinds of iris. The operator can change the choice of the classification variable at any time. In the graph view, the color of the star changes to show the different values of the classification variable.

3.4 The Graph View

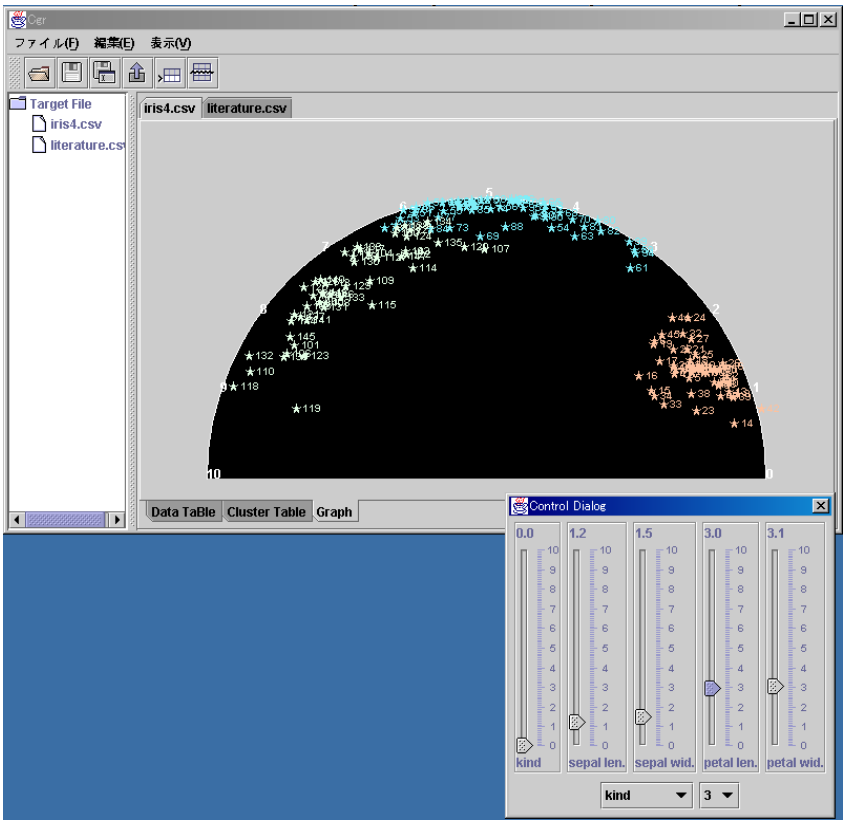


Fig. 2. The graph view (petal length=1.2, petal width=1.5, sepal length=3.0, sepal width=3.1)

The constellation graph is shown in Figure 2. The control dialog display can be used to change the values of the variables in the same way as used for the data table display. The graph display updates automatically in real-time as the value of a variable is changed. The different values of the classification variable are shown as different colors on the graph. Since the weighting factor of the classification variable is fixed at zero, it is not used in calculating the stellar position on the graph. Figure 3 shows the graph view changing the weighting factor of variable.

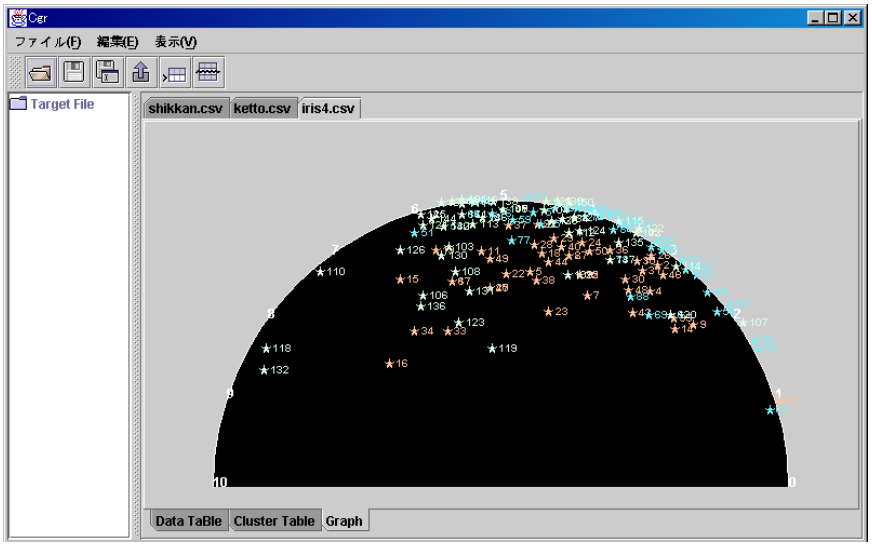


Fig. 3. The graph view (petal length=1.2, petal width=1.5, sepal length=0.0, sepal width=0.0)

3.5 Extracting Data from the Constellation Graph

As shown in Figure 4, using the graph view, the operator can use the mouse to choose a range of data designated by a polygon. This selected range can be deleted or saved in a separate file for later analysis. Data extracted and saved from a graph are stored in CSV format. Therefore, the data file can be easily used as the input for other statistical analysis and rule discovery tools.

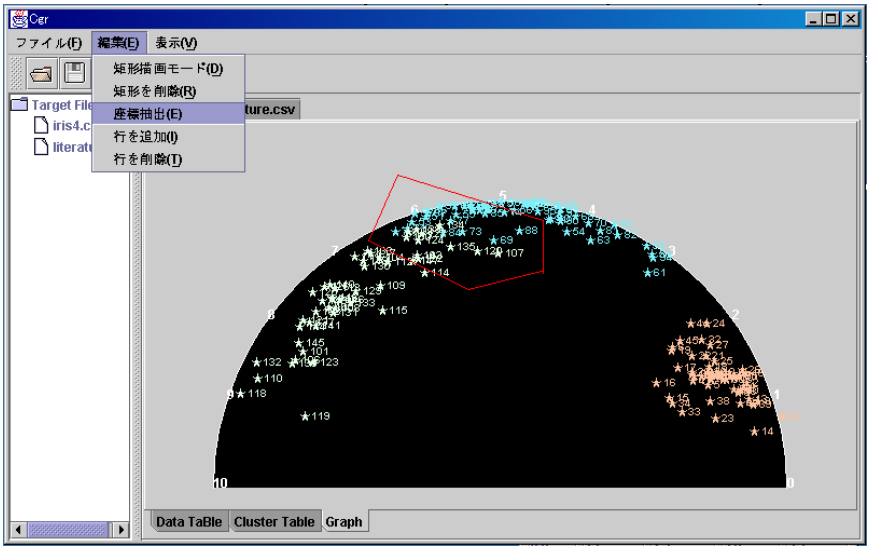


Fig. 4. Extracting data from the constellation.

4 Visual Data Mining using a Constellation Graph

To illustrate data mining using a constellation graph, two examples are discussed in the following sections.

3.6 An Example that Identified the Characteristics of Three Japanese Writers

We analyzed the characteristics of the works of three famous Japanese writers, Yasushi Inoue, Yukio Mishima and Atsushi Nakajima [6].

In Japanese, several commas are used in a sentence. As our input data, we used data that examined the frequency of use of six kinds of tokens in front of the comma to determine the characteristic of the artist. The six tokens were “to”, “wa”, “de”, “toki”, “ato”, and “e”. We examined twenty-one different documents. Eight of the documents were written by Mishima, four by Inoue, and the remaining nine by Nakajima. There were six variables in the data. The classification variable was the number representing the writer. Figure 5 shows the constellation graph when no weighting factors have been assigned. Figure 6 shows the graph when the weighting factors have been included. Table 1 gives the values of the weighting factors.

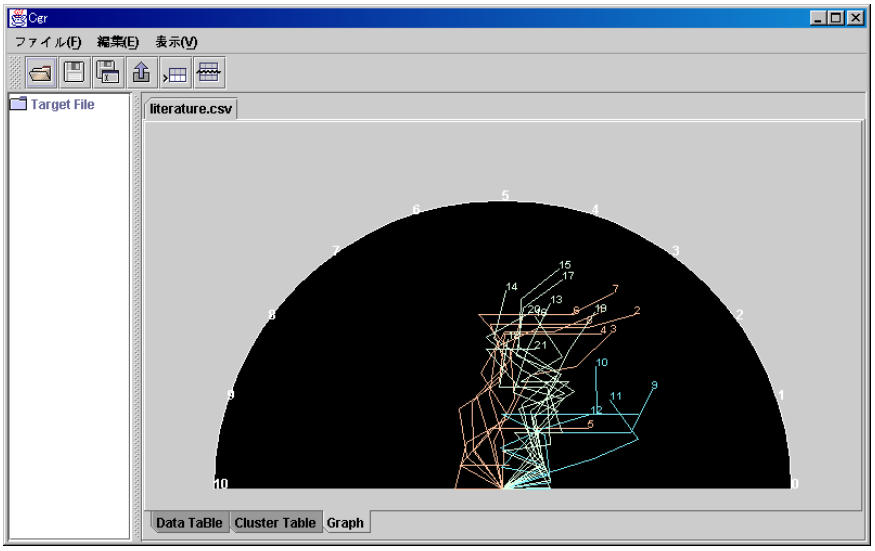


Fig. 5. The graph using the unit weighting factors.

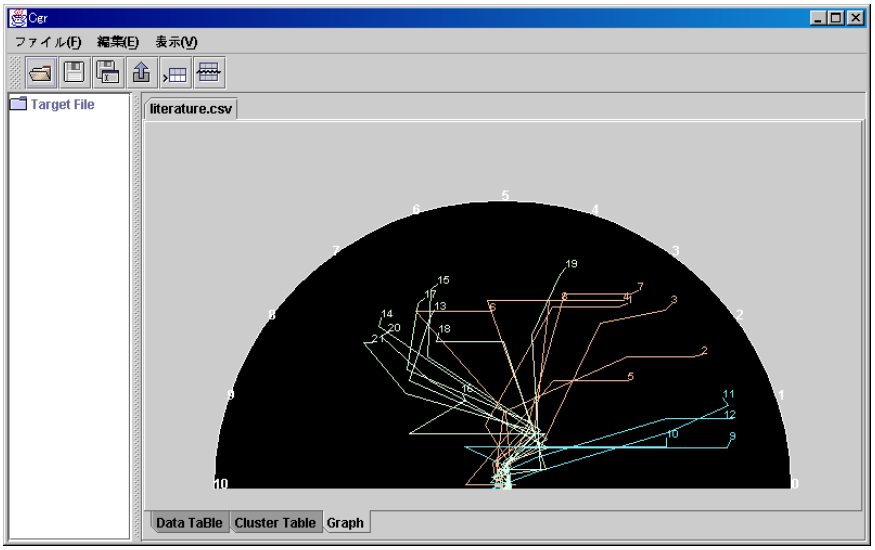


Fig. 6. The graph after incorporated the optimum weighting factors. All weighting factors are calculated so that the within-cluster variance has the minimum value and the inter-cluster variance has the maximum value.

Table 1. The weight table showing the characteristics of the writers

Writer	Weight "to"	Weight "wa"	Weight "de"	Weight "toki"	Weight "ato"	Weight "e"	The min. of J
Inoue	2.3	0.0	0.0	0.0	7.7	0.0	0.04
Mishima	0.7	4.3	1.3	0.7	3.0	0.0	0.01
Nakajima	3.7	3.7	0.0	2.3	0.3	0.3	0.13
Divides all clusters	0.3	0.7	1.7	4.7	2.3	0.3	0.85

The characteristic of the three writers can be determined from the constellation graph and the values of the weighting factors. The weight on the Table1 shows the characteristics of each writer and the value that divides all clusters on the graph. For example, if we need to distinguish only about writer "Inoue" on the graph, it is good to incorporate the weight value "to"= 2.3 and "ato"= 7.7 (the slide bar is defined from 0 to 10 on the graph). It is possible to use for the judgment of the literary works of the author not to understand, too.

3.7 An Example Analyzing Diabetes Diagnosis Data

Our next example shows the results that were obtained using diabetes checkup data of 145 people who were not overweight [7]. The data consist of five items X0 to X4:

X0: the relative weight.

X1: the blood sugar when the person became hungry.

X2: the area under the serum sugar curve when the person was given sugar when he/she became hungry; the value was recorded for three hours at 30-min intervals.

X3: the area under the serum insulin curve when after the sugar challenge recorded for three hours at 30-min intervals.

X4: the serum sugar level at equilibrium state after an intravenous injection of insulin and sugar.

The classification variable had three possible values:

1. The person is normal,
2. The person has chemical diabetes, and
3. The person has clinical diabetes.

Figure 7 shows the constellation graph incorporating no weighting factors. There is no discernable difference between the three different diagnoses. Figure 8 shows the graph when the weighting factors are included. Table 2 gives the values of the weighting factors.

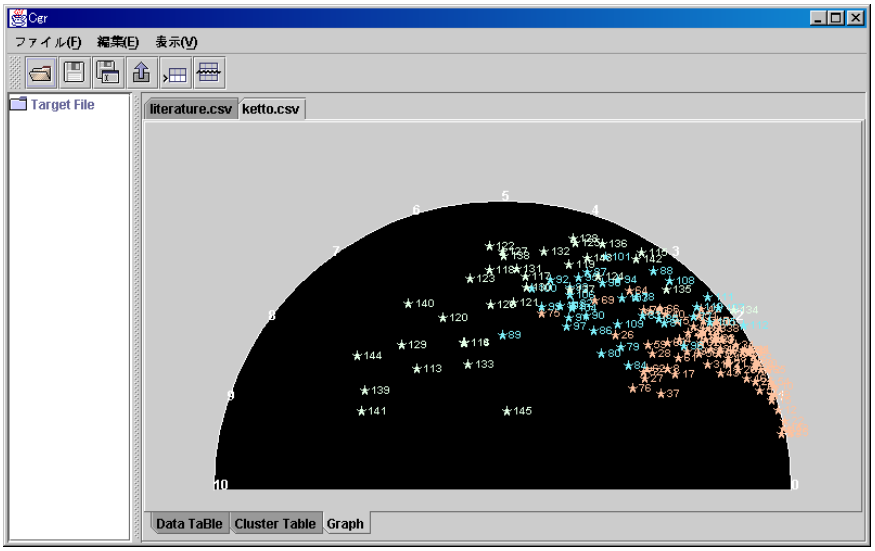


Fig. 7. The graph using the unit weighting factors.

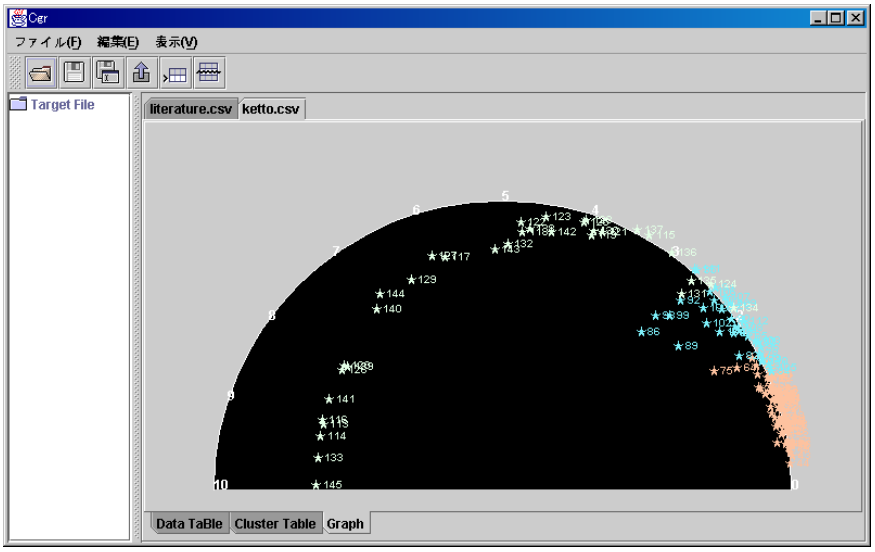


Fig. 8. The graph after incorporated the optimum weighting factors. All weighting factors are calculated so that the within-cluster variance has the minimum value and the inter-cluster variance has the maximum value.

Table 2: The weighting factors, showing the kind of the diabetes diagnosis

Daibetes	Weight X0	Weight X1	Weight X2	Weight X3	Weight X4	The min. of J
Normal	0.0	4.7	5.3	0.0	0.0	0.16
Chemical diabetes	1.0	5.7	3.3	0.0	0.0	0.23
Clinical diabetes	0.0	0.0	4.7	0.0	5.3	0.45
Divides all clusters	0.0	0.0	8.3	1.7	0.0	1.61

5 Results and Discussion

A method of constructing a constellation graph using weighting factors for the variable was presented. In addition, we have shown how an operator can select areas of the constellation graph for deletion or separate analysis at a later time. Two examples showing how our program can be used to effectively data mine a constellation graph were discussed. The most advantageous feature of our program is the provision of a means by which the operator can dynamically change the weighting factors and have these changes reflected instantly in the graph. This significantly enhances the operator's ability to cluster the data for data mining. However, visual data mining using our program is not suitable for large amounts of data due to limitations of the graphical display.

When the classification variable is categorical data, such as in our diabetes example, it is time-consuming to assign a unique value to each category when the number of categories is large or is not easily identified. Our future work will address this problem by proposing a new method for handling categorical data.

References

1. http://www.kdnuggets.com/software/visualization.html
2. Hiromi KATO: Visual Multi-Dimensional Analysis (Visualizing OLAP), IPSJ Magazine Vol.41 No.4 Apr.2000
3. Wakimoyo K., and Taguri M. : Constellation graphical method for representing multi-dimensional data. Ann. Inst. Statist.Math.30 (1997) 97-104
4. Mardia, K., and Jupp, P.: Directional Statistics John. Wiley & Sons Ltd. (1999)
5. Fisher, N.: Statistical analysis of circular data. Cambridge University Press. (1993)
6. Oyama, M., and Okada, T.: Extraction of Text Style Characteristics by Knowledge Discovery Method. K.G.Studies in Computer Science, vol.11(1996)23-36
7. Andrews, F., and Herzberg, M.: Data A Collection of Problems from Many Fields for the Student and research Worker. Springer(1985)

Appendix

Let p be the number of variables and i be the number of observations.

$$(x_{i1}, x_{i2}, x_{i3}, \ldots, x_{ip}), \qquad i = 1,2,\ldots,n \tag{8}$$

Next, $f_1, f_2, \cdots, f_p$ are given as the real number functions.

$$\theta_{ij} = f_j(x_{ij}), \qquad i = 1,2,\cdots,n, \qquad j = 1,2,\cdots,p \tag{9}$$

x_{ij} is changed into an angle using condition (10).

$$0 \le f_j(x_{ij}) \le \pi \qquad f_j, \quad j = 1,2,\cdots,p \tag{10}$$

If data increases continuously, equation (11) is used.

$$f_j(x_{ij}) = \frac{x_{ij} - x_j^{(1)}}{x_j^{(2)} - x_j^{(1)}}\pi \tag{11}$$

$x_j^{(1)}, x_j^{(2)}$ are determined using (12).

$$x_j^{(1)} = \min_{1 \le i \le n} x_{ij}, \qquad x_j^{(2)} = \max_{1 \le i \le n} x_{ij}; \qquad (j = 1,2,\cdots,p) \tag{12}$$

Picture a semicircle with radius 1 and mark it with degrees.

Mark a vector $(\cos\theta_{ij}, \sin\theta_{ij})$ corresponding to x_{ij}.

Next, multiply the vector by the weight, w_j, assigned to this vector. The vector is determined by equation (13).

$$\vec{x}_i = \sum_{j=1}^{p} (w_j \cos_{ij}, w_j \sin\theta_{ij}) \quad j = 1,2,\cdots,p \qquad (\mathbf{13})$$

Now connect these vectors.

Here, $\sum_{j=1}^{p} w_j = 1$

ξ_i: the angle of $\vec{x}_i$ and the x-axis are determined using equation (14).

$$\arg\left(\vec{x}_i\right) = \xi_i = \tan^{-1}\left(\frac{\sum_{j=1}^{p} w_j \sin\theta_{ij}}{\sum_{j=1}^{p} w_j \cos\theta_{ij}}\right) \qquad (\mathbf{14})$$

$\left|\vec{x}_i\right| = r_i$ is determined by equation (15).

$$\left|\vec{x}\right| = r_i = \sqrt{(\sum_{j=1}^{p} w_j \sin\theta_{ij})^2 + (\sum_{j=1}^{p} w_j \cos\theta_{ij})^2} \qquad (\mathbf{15})$$

Picture a star at the termination of $\vec{x}_i$.

By repeating the above steps for all x_{ij}, the constellation graph is constructed.

Mining Travel Data with a Visualiser

Colin Ho
BT Asia-Pacific
hoc@hongkong.btap.bt.com

Ben Azvine
BTexact Technologies
ben.azvine@bt.com

Abstract: The rapid advances of mapping technology have provided new challenges to the KDD communities in exploiting its potential for visual data mining on spatial data. This paper discusses the issue of integrating this technology with data mining methods to provide an interactive environment for exploratory data analysis. We demonstrate our ideas by building highly interactive interfaces that integrate these tools together to enable collaboration between human experts and the system in the process of mining travel data. More specifically, we describe a travel visualiser and the role it plays in the knowledge discovery process.

1 Introduction

Visual data mining has emerged as one of the most popular and powerful techniques to discover hidden patterns in large volumes of data. This is not surprising as visual pattern recognition skills far exceed our ability to comprehend collections of texts and numbers. This sole benefit usually results in active user participation in the process of knowledge discovery, which in turn facilitates the development of better algorithms and processes for data mining in revealing interesting, hidden patterns and relevant anomalies in the data.

Over the past two decades, we have seen rapid advances in the development of visualisation methods and tools for exploration of large amounts of spatial data (Andrienko and Andrienko1999, Kraak and MacEachren 1999). Many digital data generated today has embedded geographical information like coordinates (latitude and longitude) and postal codes. This information can be fully exploited to provide a highly interactive environment to explore and present dynamic spatial data.

This paper discusses the issue of integrating spatial visualisation and data mining tools with the database. The goal of this integration is to provide a highly interactive tool that facilitates both the process of uncovering patterns and relationships in large, complex data and providing explanation of those patterns and relationships.

To address this issue of bringing these technologies together we need to carefully design an interface that will provide an environment for exploratory visual data analysis. However it would be difficult to build an interface that will fit all types of database as each has its own unique characteristics. Our approach is to develop domain-specific manipulation tools that integrates data mining methods and visualisation tools that enable human and machine to work together in the process of pattern discovery, and integrating databases with visualisation to query for relevant information to enable thinking, hypothesis generation, and problem solving.

Applications using this technology have been used to solve a wide range of business-critical problems, including detecting telephone calling fraud, estimating the traffic flows in cities, and managing resources in a tightly controlled environment. In this paper, we illustrates our ideas by describing the use of this technique to improve the estimation of travel times based on information gathered by BT field engineers.

In the next section we describe the application domain and the rationale behind the work. Section 3 describes the travel visualiser which uses the mapping technology to display travel patterns that let the user quickly see unusual patterns, Section 4 covers the role of the visualisation system in the knowledge discovery process. The last section contains conclusions and recommendations.

2 Travel Time Estimation

Any organisation with a large mobile workforce needs to ensure efficient utilisation of its resources as they move between tasks distributed over a large geographical area. BT employs around 20000 engineers in the UK who provide services for business and resident customers such as network maintenance, line provision and fault repairs. In order to manage its resources efficiently and effectively, BT uses a sophisticated dynamic scheduling system to build proposed sequence of work for field engineers. This system is typically developed to schedule tasks and activities for field engineers in accordance with predetermined rules governing cost, travel and business targets. The scheduler has the ability to modify the sequence in real-time to accommodate the dynamics of resource availability if new high priority tasks appear on the system.

A typical schedule for a field engineer contains a sequence of time windows for travel and task. To generate accurate schedules the system must have accurate estimates for time taken between tasks referred to in this paper as "travel time" and estimates for task duration. By using visual data-mining techniques we have implemented a system that improves the accuracy of "travel time" estimates by 30% compared to the previous system. Under the old "travel time" estimation system many engineers mainly due to underestimation of "travel time" were not able to arrive on-time for their next task resulting in knock-on effect and inefficient schedules. This was evidenced by the our preliminary analysis that for some end-of-day tours field engineers travelled from region *A* to region *B* and then back to a location close to region *A*, causing a criss-crossing effect on the overall schedule. This deficiency points to an underlying problem of providing accurate estimates for "travel times" between jobs.

The system collects event logs on the activities undertaken by a field engineer. Typically, this information comes from a system that monitors the workflow from the moment it issues a task to a field engineer till the task is completed. These event logs are also recorded and stored in a central database. A typical event log includes the field engineer information, the time a new task is issued, the location and region code of the destination site, the arrival time on site, the time the engineer accepts to carry on the task, and the time the task is completed.

Note that the "travel time" is calculated as the difference between the time a task is issued and the arrival time on site. Specifically, a travel time includes the time required to leave the current site, walk to the car-park, start the car, drive to the destination site, park the car, and arrive at the door-step of the next customer. In most cases a large proportion of the travel time is driving from one site to another and the car-park is within the premises of the site. Unfortunately this may not be the case in the urban area like London or city centres where the engineers may take a substantial amount of time to find a car-park and to walk from the car-park to the door-step of the customer.

Travel time is typically treated as a fixed overhead when scheduling jobs, and is extremely difficult to quantify, because factors such as road conditions, weather, vehicle type, route disruption, driving behaviour, traffic peak periods, etc. all contribute to journey times, making it difficult to prescribe an expected inter-job

journey time. More specifically, given a site location *A* and the destination location *B*, we want to predict the time, under a typical condition, it will take for a field engineer to travel from site *A* to *B*.

Our own experiences in driving tells us that a collection of our past journey experiences, taking into consideration the above factors and discarding those abnormal travels where unusual events like accident occurred, gives a good prediction of travel times. Hence we define a theory to predict travel time as follow:

> Definition 1: If *m* journeys take *n* minutes to travel from site *A* to *B*, then it is likely that it would take *n* ±15 minutes to travel from *A* to *B*.

From this theory, we know that it is crucial to carefully select the typical journeys to build a prediction model. For example, we should include journeys where there are road works that last a period of time, but should exclude those where an accident has occurred. We use the visual data mining approach to solve this problem as it would provide a useful tool to identify unusual travels.

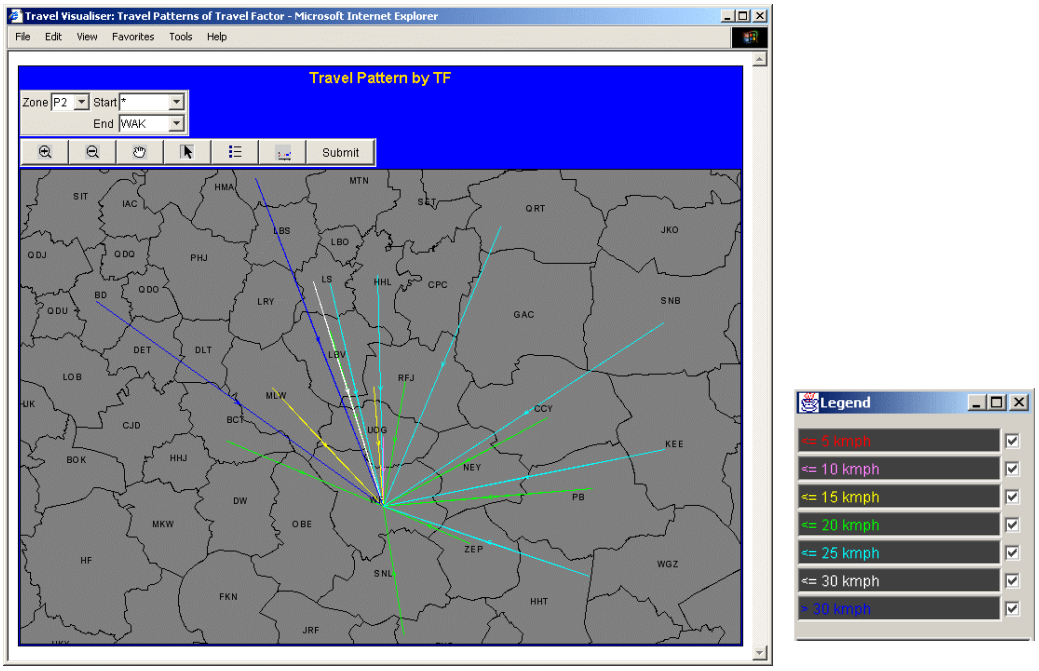


Figure 1: A visualiser displays travel patterns based on estimation model. It shows all travels ending in the region "WAK". The legend on the right shows the colour-codes representing each travel.

3 The Travel Visualiser

Existing visualisation tools, like scatter plot and histogram, are powerful visual aids to show the relationship between two attributes in a data. Over the past few years, we have seen a new generation of computerised visualisation tools is represented including MineSet™, a data mining software from Silicon Graphics [Brunk et al., 1997] and XGobi [Swayne et al., 1998]. The advanced visualising models of MineSet™ include Scatter, Map, Tree, and Evidence Visualiser. Although these tools are useful in many ways, they do not meet our needs in visualising travels, where geographical information is one important source of knowledge.

In order to help users actively explore and interpret data of their interest, we have designed a tightly coupled interface of an interactive system that provides the visualisation of travel patterns with facilities for geographical information, scatter plot, colour-coding, direct data querying, data drill-down, identifying hot-spots, and travel explanation. The visualiser is also integrated to other travel visualisation tools like AutoRoute. In this section, we briefly describe the use of each of these facilities.

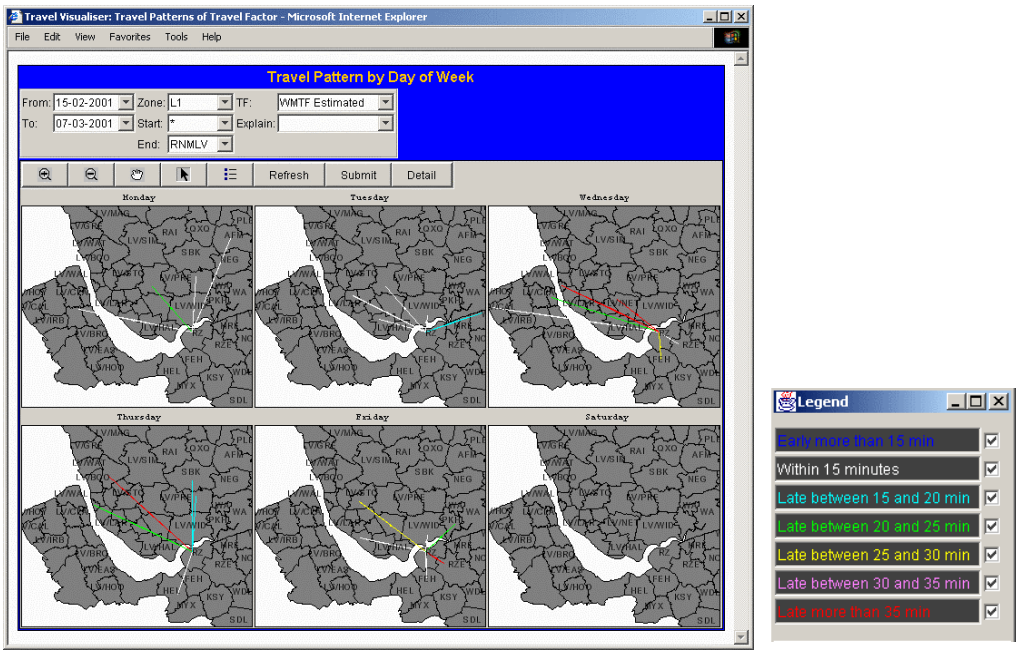


Figure 2: A visualiser displays travel patterns based on day of week. It shows all travels recorded for the period from 15 Feb to 7th March that starts from all regions and ends in "RNMLV". Each map represents all travels done for the day, for example 19/2, 26/2, and 5/3 are Monday. Each coloured line represents the performance of a travel based on the setting specified in the legend, shown on the right, for example a red line shows that the travel is late for more than 35 minutes.

We use the geographical components in the data to display the results on a map. Maps are more informative than simple charts and graphs, and can be interpreted more quickly and easily than spreadsheets or 2-D graphs. Each journey is represented on a map by a line using the X-Y co-ordinates of the start and end locations and an arrow to show the direction of the travel. This simple plot also implicitly reveals the distance between the two points. The performance of each travel is categorised into groups based on the speed of travel, for example <= 5kmph, <= 10kmph, etc. We then define the legend by assigning different colour codes for each of these categories. What makes it all come together is a visual display of the travels on the map. Figure 1 shows a visualiser displaying travel patterns generated by an estimation model. The user can select the scatter plot button to display the relationship between speed and distance.

3.1 Mapping Technology

We include the facility to provide basic geographical information of the region, which is an important source of knowledge, by integrating the mapping technology in the visualiser. The map is based on the same mapping technology used in products such

as MapInfo Professional and Microsoft Map. It adds powerful mapping capabilities to the visualiser as it can display information in a format that is easy for everyone to understand.

Figure 2 shows a visualiser that displays travel patterns based on days of week. By viewing the map, we can immediately know that a field engineer would need to make a detour to reach the destination because a river or canal separates the two regions, thus prolonging travel times. However, we might not be able to draw such a conclusion if the visualiser does not contain the geographical information. This feature lets the users see patterns and relationships in the mass of information quickly and easily without having to pore over the database.

The map was designed such that each region represents an area code of the telephone numbers. It gives an abstract view of the region and omits geographical details such as roads and highways so as not to overwhelm the user with too much information. This is consistent in practice as field engineers usually have local knowledge of the regions and may use different approach routes to the same destination.

3.2 Direct Data Querying

A fundamental function of a visualiser is to allow the user to interact with the data directly for additional information. Users can click on the map to select a journey and the system extracts and displays all information relating to the travel from the database. This facility is essential to exploratory data analysis as the user can generate and verify hypothesis about a travel pattern by performing data drill-down, which we will explain in the next section. For example, a user may want to request information for a peculiar travel so that he can verify it against travel patterns by period of day to see if other travels have the same behaviour.

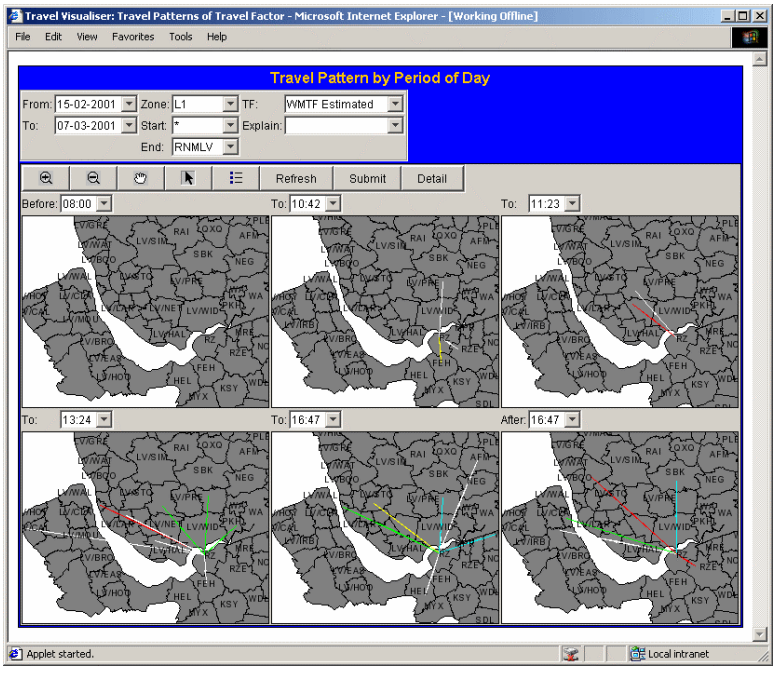


Figure 3: A visualiser displays travel patterns based on periods of day from 15 Feb to 7th March that starts from all regions and ends in "RNMLV". Each map represents all travels done for the period, for example the first map shows travels done before 8a.m. while the second map from 8a.m. to 10:42a.m.

3.3 Data Drill-Down

In order not to overwhelm the user with too much information on a single map, we provide multiple views of travel patterns including evaluation of estimation model (see Figure 1), days of week (see Figure 2), periods of day (see Figure 3), and driving behaviour of engineers (see Figure 4). Each view has been designed to be used by different user groups with distinct requirement for what analysis should be done and how data should be displayed.

This is done by providing numerous filtering utilities in each view. For example, an abstract view can be displayed by selecting all travels from all start and end regions, whereas a filtered view can be obtained by selecting a specific start or end region as shown in Figure 2. The flexibility to select a date range would allow a user to view travel patterns by days, weeks or months. This is particularly useful when it is used to monitor the trend of travels in a region where a road work begins or ends. An operation staff may also use the visualiser to aid in decision making when assigning a high-priority task to an engineer who has a faster approach route to the site.

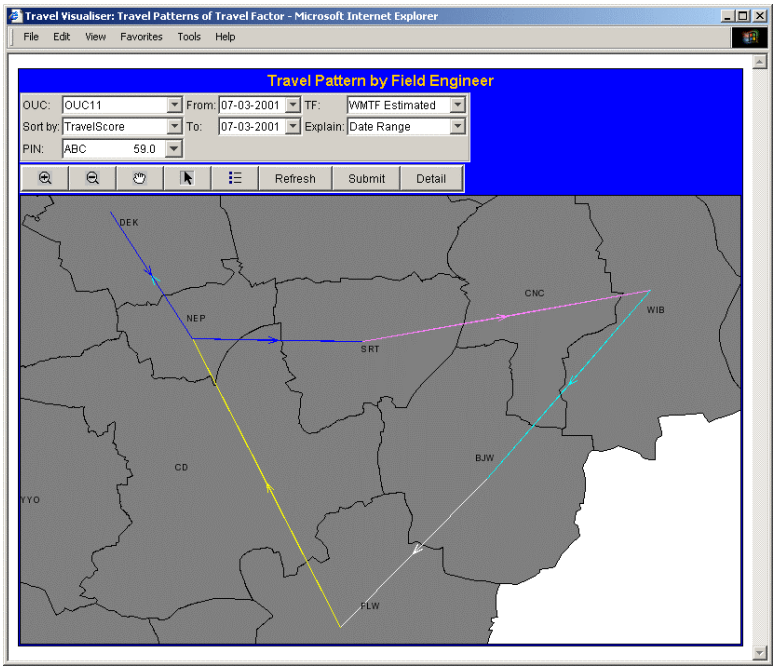


Figure 4: A visualiser displays the end-of-day tour of a field engineer, ABC, on 7th March 2001. It also displays the travel behaviour as compared to the overall engineers. In this case, ABC has a travel score of 59%, where 100% represents the best travel ranking.

3.4 Identify Hot-Spots

The ability to display hot-spots is an important issue in many visualisation systems. To address this issue, we provide facilities to allow the user to switch on travel hot-spots. The system also suggests period hot-spots intelligently.

A legend window, as shown in Figure 2, shows the representation of the colour-codes on travels. The user can select the travels they want to see on the map. For example if the user want to identify hot-spots, which represent travels that are late for more than 35 minutes, then all colour-codes, except red, in the legend are

unselected. This flexibility allows the user to quickly identify travel hot-spots and performs further investigation by using one or more drill-down views.

Another feature of the travel visualiser (see Figure 2) is to automatically identify period hot-spots and display travels according to their respective periods. The system first reads the time-related attributes, like start and arrival time, and performs a supervised discretization (Fayyad and Irani, 1993) using the travel categories as the teacher. This feature shows the relationship between time-related attributes and travel and the user can easily identify on-peak and off-peak periods in a region.

3.5 Travel Explanation

One of the most fundamental issue in knowledge discovery is the ability to distinguish the similarities or differences between a particular record against the numerous patterns discovered from the mass of data. For example in the case of travel patterns, the user identifies a peculiarity on a particular travel and would like to know which other travels share the same behaviour. This would allow the user to gain deeper insight into the underlying patterns in the data.

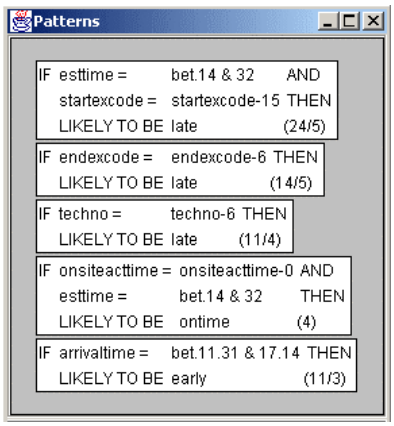


Figure 5: The system offers 5 likely explanations with respect to the travel selected. Each box is a rule whose conditions match the values of the travel. The bracket at the end of each rule represent (total classified / incorrectly classified).

We address this issue by designing an explanation module and integrating it into the visualiser. The process of generating an explanation model including data selection, transformation, attribute selection, model building, and rules matching. This module has been designed to be used by different user groups. The user first decides the scope of data to build the model by selecting the values provided. For example in Figure 4, the user selects the date range to scope the explanation model, which in this case all travels recorded on the date, 7th March, will be used to build the model. Alternatively, the user, who is a team leader, may want to scope the data to his own OUC group. An explanation model will then be build based on the values selected.

The system then extracts the data from the data warehouse. The data then undergoes a series of transformation process including global discretization of all numerical attributes (Fayyad and Irani, 1993), global grouping of attribute values (Ho and Scott, 2000), and removal of irrelevant attributes. The system then use the transformed data to generate an explanation model which is in the form of production rules (Quinlan 1993).

When the user selects a travel on the map, the system will pass through all rules in the model and display those rules whose conditions match the values of the

travel. Figure 5 shows a typical explanation window displaying the patterns. The user may treat this as new hypotheses and investigate them using one or more drill-down views. Hence this module augments the data drill-down facilities in the system.

3.6 Integration to other tools

Another auxiliary feature of the visualiser is the integration of other commercial travel visualisation products like AutoRoute 2001, which provide driving direction from address-to-address. The user can select a travel on the map and request AutoRoute to suggest a detailed route. This facility would further facilitate the patterns searching process as the data analyst would have a rough idea of what route an engineer might take. It can also be used as a coaching tool for field engineers who do not have domain knowledge of the region.

4 Knowledge Discovery Process

We have described the travel visualiser and highlighted some of its features. As can be seen, this visualiser has been designed as a tool to be used for mining travel patterns. In this section, we will briefly describe the role it plays in each stage of the knowledge discovery process.

4.1 Data pre-processing: Cleaning and Transformations

As many data miners will testify with us that a large proportion of the knowledge discovery time is taken up within this stage. Visual data mining attempts to shorten this time by providing opportunities for closer interaction between the data, domain experts, and data miners. Hence, it is important that the visualiser is designed in such a way that domain experts, at different levels, could easily gain insights on the data.

From Definition 1 we know that it is crucial that we accurately select m journeys to simulate "normal" travels, which include routes that have road works over a period of time, peak periods. However, we should exclude "abnormal" travels, including those that do not meet the legal travel requirements, for example, speed at 100 miles per hour, and travels that were held up by an accident. Such events are difficult to spot as there is no travel description recorded in the data.

By using the visualiser, the domain experts could easily point out travels that are classified as "abnormal". For example field engineers are required to sign on to the network to register their start-of-work for the day. The system keeps a record of the default sign-on location for each engineer and travel times are calculated from this location to the first task location. However, the visualiser shows that a large proportion of these travels are either late or very early, and one domain expert explains that an engineer may not be in the default location as registered in the system when they signed on to the network. Such a revelation prompts us to remove all first job travels from the training data. With such active user participation and the aid of the visualiser, we are able to implement rules to remove unreliable data from the training set.

After the data cleaning process, the data are transformed, reformatted, and the visualiser is used to verify the correctness of the new data. One such transformation is the discretization of time-related variables like start and arrival time. As shown in Figure 2, we could use the visualiser to evaluate the effectiveness of such data transformation.

4.2 Data Mining and Evaluation

This step can be viewed as the automated application of data mining algorithms to build a predictive model that fit to the data (for example, a regression tree, a linear function, a set of fuzzy rules, etc.). The predictive model is then subjected to critical evaluations on the quality of the output, but unfortunately this step is often ignored and usually limited only to the statistical evaluation.

The travel visualiser, integrated with the explanation module, is a useful tool for evaluating the model generated by the data mining algorithms. As can be seen in Figure 5, the value of the *esttime* is estimated by the output model. This offers some indications to the quality of the model. The user can use one of more views to verify their evaluations and provide feedback to the data miner. This may suggest a need to further clean or transform the data, forming a feedback loop to the KDD process. This cycle is repeated until the user is confident with the results.

4.3 Deployment

The travel trends change from day to day, and there is a need to update the estimation model to match the latest condition on the road. We automate the overall KDD process from data selection to model construction so that the model can be used continuously in real-time for estimating travels.

The visualiser has the facility to allow the user to select the type of model for estimation, i.e. current and recommended. When the system activates the model update process, the current and recommended will produce the same travel estimations. As more travels are added to the system, the recommended model will gradually be different from the current. The user can use such facility to help them access the models and decide when to perform the next model update. This flexibility increases the confidence of the user in the overall system.

5 Conclusions and Future Work

We have demonstrated that visual data mining offers many benefits to the KDD process. We have carefully designed a visualiser which has complete integration to the database and other "off-the-shelf" visualisation tools.

The use of the mapping technology in the visualiser has added another dimension to visual data mining. By providing multiple maps in a single view, the user could easily recognise complex dependencies between many attributes. This encourages active user participation as they could apply their perceptual abilities to gain insights to the underlying patterns of the large data sets. By using the visualiser as a tool for dialogues between users and developers, it speeds up the overall KDD process. When a visualiser is designed to be used at every stage of the KDD process, it increases the quality of the output model. This is evidence in our project as the accuracy of our proposed system significantly outperforms the existing system by thirty percent, which in turn causes the scheduler to produce a higher quality tours of work. This also improves customer service as the company could allocate a smaller waiting time window for the customer.

Researchers should look beyond scatter plots in designing a visualisation tool. The integration of the mapping technology in a visualiser opens up many new challenges in visual data mining, particularly when there are geographical components in the data. Data visualisation is often restricted to general-purpose tools provided by most commercial data mining packages. Many such tools are usually catered for scientists but not domain experts. In many situations, developing an

application specific visualiser would significantly improve the environment for the data exploration process.

Acknowledgement

We wish to acknowledge Richard Maxwell from BT Retail and Ted Lawson from BT Wholesale for their help and guidance during the trial and implementation stages of the system.

References

G. L. Andrienko & N. V. Andrienko. Interactive Maps for Visual Data Exploration, in *International Journal Geographic Information Science*, 13 (4), pp. 355-374,1999.

C, Brunk, J. Kelly, and R. Kohavi. Mineset: An Integrated System for Data Mining, In *Proc. KDD-1997: The Third International Conference on Knowledge Discovery and Data Mining*, Menlo Park, CA, 1997.

U. M. Fayyad and K. B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proc. IJCAI-1993: Thirteenth International Joint Conference on Artificial Intelligence*, Morgan Kaufmann Publishers, Los Altos, CA, pp. 1022-1027, 1993.

K. M. Ho, and P. D. Scott. Reducing Decision Tree Fragmentation Through Attribute Value Grouping: A Comparative Study, in *Intelligent Data Analysis Journal*, 4(1), pp.1-20, 2000.

M. J. Kraak and A. MacEachren. Visualization for exploration of spatial data. In *International Journal of Geographical Information Science*, 13(4): pp. 285-287, 1999.

J. R. Quinlan. Programs for Machine Learning. Morgan Kaufmann Publishers, Los Altos, CA, 1993.

D. F. Swayne, D. Cook, and A. Buja. `XGobi: Interactive Dynamic Data Visualization in the X Window System. *In Journal of Computational and Graphical Statistics*, **7**(1), pp. 113-130, 1998.

Customer Data Mining and Visualization by Generative Topographic Mapping Methods

Jinsan Yang and **Byoung-Tak Zhang**
Artificial Intelligence Lab (SCAI)
School of Computer Science and Engineering
Seoul National University
Seoul 151-742, Korea
{jsyang, btzhang}@scai.snu.ac.kr

Abstract

Understanding various characteristics of potential customers is important in the web business with respect to economy and efficiency. When analyzing a large data set on the customers, the structure of data can be highly complicated due to the correlations and redundancy of the observed data. In that case a meaningful insight of data can be discovered by applying a latent variable model to the observed data. Generative topographic mapping (GTM) [2] is a latent graphical model which can simplify the data structure by projecting a high dimensional data onto a lower dimensional space of intrinsic features. When the latent space is a plane, we can visualize the data set in the latent plane. We applied GTM methods in analyzing the web customer data and compared their relative merits on the clustering and visualization with other known method like self-organizing map (SOM) [10] or principle component projections (PCA). When applied to a KDD data set, GTM demonstrated improved visualizations due to its probabilistic and nonlinear mapping.

KEYWORDS: visualization, generative topographic mapping (GTM), self-organizing map (SOM), clustering, web data mining

1 Introduction

The complexity and amount of data in commercial and scientific domain grow explosively by the advance of data collecting methods and computer technologies and the development of internet has changed patterns and paradigm of business in an unprecedented way The application of data mining techniques to web data and related customer information is important with respect to economy and efficiency. For efficient analysis of such data, the understanding of structure and characteristics of data is essential. The complex nature of data can

be expressed through various models. The model from biological origin is the neural networks which is inspired from extraordinary capabilities of biological systems (via ensembles of neurons) in learning a complex task. By training the neurons in the artificial neural networks the case with several input features can be classified with (supervised) or without (unsupervised) knowing the target [8]. A graphical approach for expressing the data structure can be done by using Bayesian belief networks which show graphically the inter-relations of the various features with local conditional probabilities and log-likelihood of fitness [14]. More direct and intuitive methods of expressing the data structure are the visualization of data through dimension reduction from data space into the visible 2D or 3D space. One of traditional methods is PCA by projecting the data into a lower subspace through minimizing errors. PCA intrinsically assumes that the given data structure can be modeled by linear or flat planes and can show meaningful data structure when this is the case. For the general nonlinear case, SOM is used for visualization. SOM can reflect the structure of the data [10] by expressing the cluster centers in the 2D plane topographically. By selecting the winning node in the plane for each data point, the high dimensional data can be visualized. GTM [2] is a more flexible way of data visualization using a generative model based on their posterior likelihood. GTM can be thought as a nonlinear PCA [7]. Like SOM, GTM maps high dimensional data into a visible 2D plane using predetermined grid. By selecting each grid point in probabilistic way, GTM can project each data point over the entire plane allowing better visualizations. GTM can be connected with SOM by regarding the latent vector as a neuron and the basis function as a connecting strength between neurons [9]. For the clustering of data, we projected the data into the latent space and performed the clustering analysis using k-means clustering algorithm. For the case of SOM, [16] has used the node set as a projection of data for the clustering. But due to the limitation of nodes, the projection is limited to the given nodes (compare Figure 4 and 5 in Section 3).
We will explain more details about GTM in Section 2 and apply them in analyzing the KDD 2000 data of web customers. In section 3, the results of analyzing KDD data under several feature selections are discussed. In section 4, some conclusions and future works are discussed.

2 Visualizing complex data by generative topographic mapping

In the analysis of high dimensional data, there are several extensively used dimension reduction methods. The latent variable model is one of the methodologies by assuming hidden variables and finding the relationships between observed data and hidden variables. GTM can be regarded as a nonlinear generalization of factor analysis to model the nonlinear mappings between latent space and data space. The complexity of data can be taken as a reflection of the intrinsic features or factors and if we can express the data in terms of this intrinsic

features, the data complexity can be greatly reduced allowing correct and easy analysis. GTM assigns each data point to a set of grids based on a probabilistic model (soft clustering) while in SOM the data point is assigned to the closest node or neuron to the data point (hard clustering). Since in GTM, each grid can assume a posterior probability of taking a data point, the clustering can be expressed over the whole latent space.
SOM is an unsupervised neural network algorithm inspired from the biological phenomenon of human brain. When the external images are perceived in the sensory cortex of brain, part of the neurons are stimulated to respond for the incoming spatial images. Similarly SOM maps each high dimensional data point to a 2 dimensional array of nodes preserving topologies of the data structure. By updating reference vectors repeatedly the data structure is reflected in the nodes of the plane. The expression of data structure by SOM is limited on the given node set by winner-take-all selection method and the relationship between data and node set is ambiguous. GTM allows more flexible expression by adopting soft clustering through responsibility of each data point. GTM is a nonlinear PCA for a set of basis functions and much more flexible than PCA when the relationship between feature and latent variable is not linear.
The basic assumption of GTM is through a generative model which defines a relationship between data space and latent space [2]. For $t \in R^D$ (a data space), $x \in R^L$ (a latent space) with noise e and a parameter matrix W, the form of generative model of a non-linear mapping y becomes

$$t = y(x, W) + e$$

where $y(x, W)$ is a product of basis function and weight vector for each observed data. The data point is assigned according to its posterior probability or *responsibility*. The responsibility of assigning the n-th data point to the k-th grid point is

$$r_{kn} = p(x_k|t_n, W) = \frac{p(t_n|x_k, W)p(x_k)}{p(t_n|W)}$$

To avoid computational difficulties in calculating the denominator, the distribution of x in the latent space is assumed to be a grid.

$$p(x) = \frac{1}{K}\sum_{k}^{K}\delta(x - x_k)$$

Under appropriate settings, it can be shown that this grid vector in the latent space corresponds to a neuron of the SOM and the corresponding basis function corresponds to a binding strength between data and neuron [9].

The basis functions usually consist in three different forms corresponding to bias, linear (polynomial) trends and nonlinearity of the data

$$\{1, x, \Phi(\mu; \sigma^2)\}$$

where Φ is a Gaussian kernel.
Given the specified grid and a set of basis functions, the data can be modeled

iteratively using EM algorithm by updating the parameter matrix W (M-step) and assigning each data point according to its responsibility (E-step). After modeling the data structure, the data can be projected into the latent space according to the posterior probabilities of grid points for visualization. We analyzed a real web data using topographical mapping methods and compared their visualization aspects in great detail in the next section.

selection criteria	Discriminant analysis	Decision trees	naive Bayes
selected feature sets	v229, v240, v304 v368, v283, v396 v394, v80	v234, v237, v240 v243, v245, v304 v324, v368, v374 v412	v18, v108, v229 v369, v417, v451 v452, v457

Table 1: The composition of feature sets from the three different selection criteria

3 Web customer data mining and visualization

KDD Cup 2000 data (Question 3) is a record of the web customers who have visited an internet company, *Gazelle.com* which sells leg care/wear items during the period of Jan. 30, 2000 $\sim$ March 30, 2000. Understanding these customers can save lots of money and time and provide a useful directions for future marketing and saling of this company. The primary concern of the company is to analyze the characteristics between heavy spenders who spend over $12 and light spenders who spend less than $12. The data has 426 features over 1700 cases with a target variable for indication of heavy/light spenders. The features are various measurements of categorical, discrete and continuous characteristics. Examples of features include residence area (categorical), age (discrete), income (discrete), rate of discounted items (continuous) and so on.

3.1 Feature selection

Since there are over 400 various measurements of features in the data set, appropriate feature selection for the purpose of data analysis is necessary before applying the latent variable model. Feature selection process [3] is summarized in the following four steps: (1) generation of feature set (2) evaluation by specific criteria (3) stopping conditions (4) validation by test set.
In each step of the selection process an appropriate evaluation method has to be assumed. Distance measure and information gain are two typical selection criteria for the measure of discrimination of clusters. The distance measure is used in evaluating the feature set by measuring the distance between clusters while the information gain is used by measuring the prior and posterior entropy for the feature set. The distance measure used for instance in the discriminant

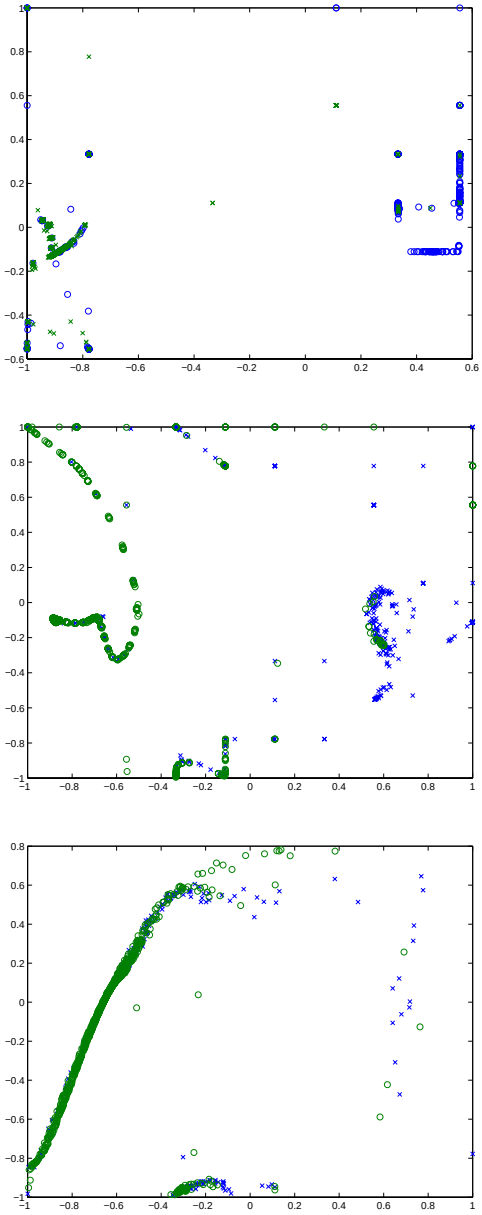


Figure 1: Three GTM plots of KDD data by the feature set selected from discriminant analysis (above), decision tree (middle) and naive Bayes criteria (below) .(o: light spender, x: heavy spender)

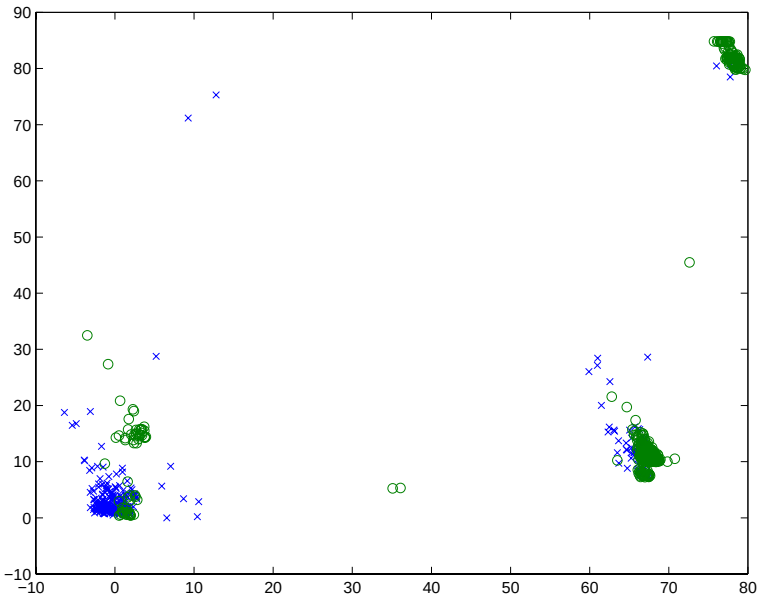


Figure 2: The PCA plot of KDD data ('x': heavy spenders 'o': light spenders

analysis is the Mahalanobis distance that is a generalization of Euclidian distance. The densities of each class can be assumed as multi-variate normal or can be estimated by non-parametric density estimations based on kernels or k-nearest neighbor methods [6].
For the selection method, distance measure uses stepwise selections to avoid the redundancy in the feature set. The selection of attributes in the discriminant analysis can be controlled by adjusting the threshold values of selection criteria. On the other hand the decision tree algorithm [12] is utilizing the information gain and the features in the pruned trees can be selected. Other than above two, naive Bayes classifiers [11] select features by calculating the posterior probability of feature selection assuming the conditional independence of features given the target value.
Table 1 shows different sets of features in the KDD data which are selected by each selection method mentioned in the previous section. The discount rate in ordered items (v229,v234 $\sim$ v237), the weight of items (v368, v369) are common for all three methods and the minimum order shipping amount (v304) is common for the first two selections. The third selection contains several interesting features: the geographic location (v18), products purchased on Monday (v108), number of lotion, men, children product views (v417, v451, v452) and average time spent for each page view (v457). The features about men's sports collections (v283), house value (v80), number of free gift (v394), vender (v396) views are in the first selection set. Order line amount (v243, v245), number

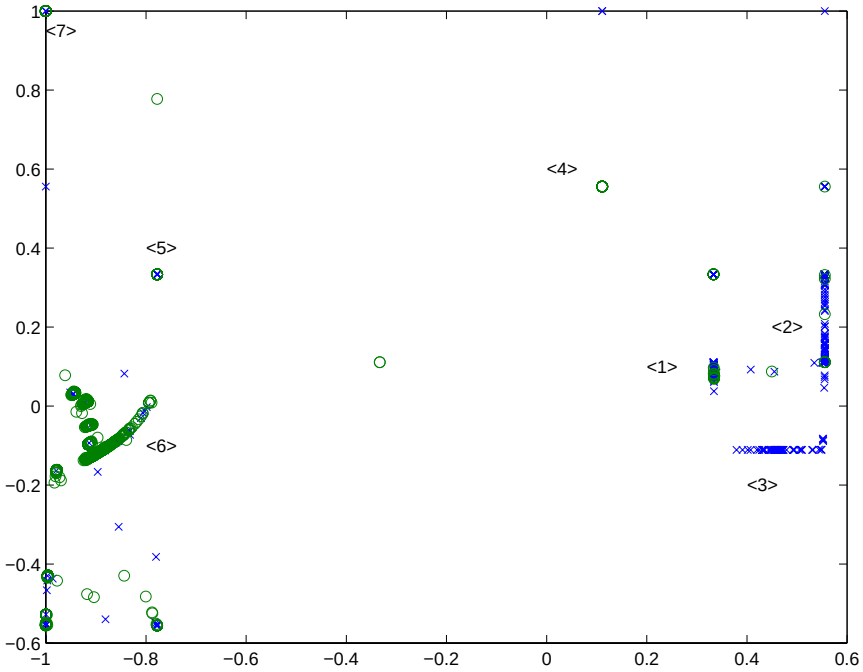


Figure 3: The GTM plot of KDD data with 7 clusters. 'x': heavy spender, 'o': light spender

of leg care, replenishment stock, main template views (v324, v412, v374) are among the second set.
In Figure 1, the plots of three methods applied to the KDD data are compared with respect to the clustering and visualization. The first plot shows a cluster of light spender and three groups of heavy spenders. The second plot has two clusters of light spenders and a cluster of mixed spenders. The third one shows two clusters of light spenders mixed with heavy spenders. The performance of each selection method depends on the nature of data and it is not easy to see which method is better than the others with respect to the visualization.

3.2 Data mining and visualization

We selected 8 variables by parametric discriminant analysis with 75.1 % of canonical correlation rate and used them for the analysis of KDD data. In PCA plot (Figure 2), the clustering of heavy/light spenders is not so evident. Especially in one cluster (the bottom left cluster in Figure 2) they are heavily mixed indicating non-linear trend of KDD data.

Such ambiguities are greatly resolved in GTM plot (Figure 3) where the clusters are divided and reshaped into 7 clusters. The heavy spenders are divided

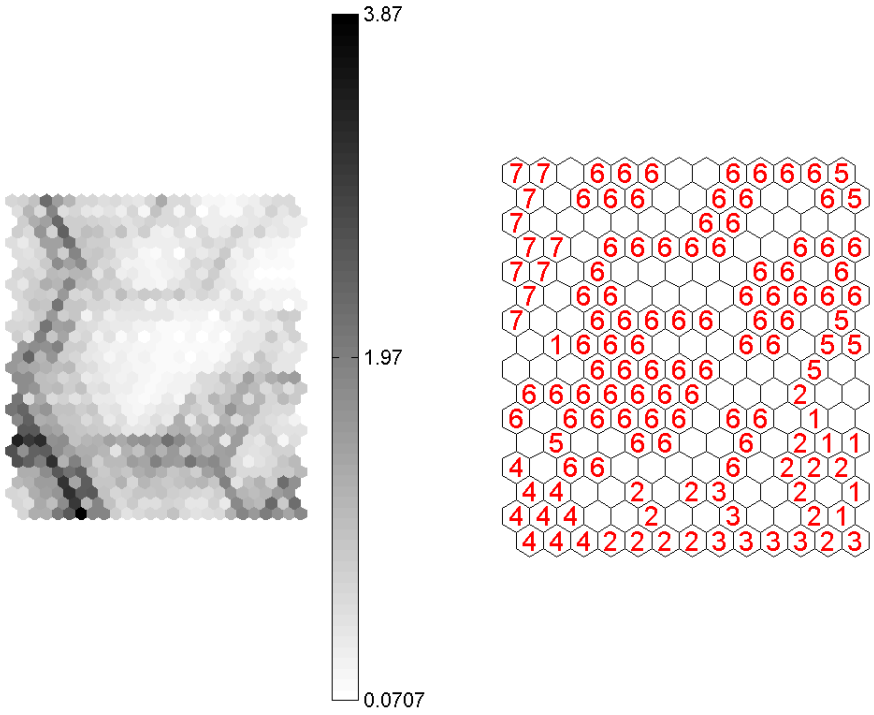


Figure 4: The SOM of KDD data marked with 7 clusters of GTM. The darker color indicates longer distance between neurons

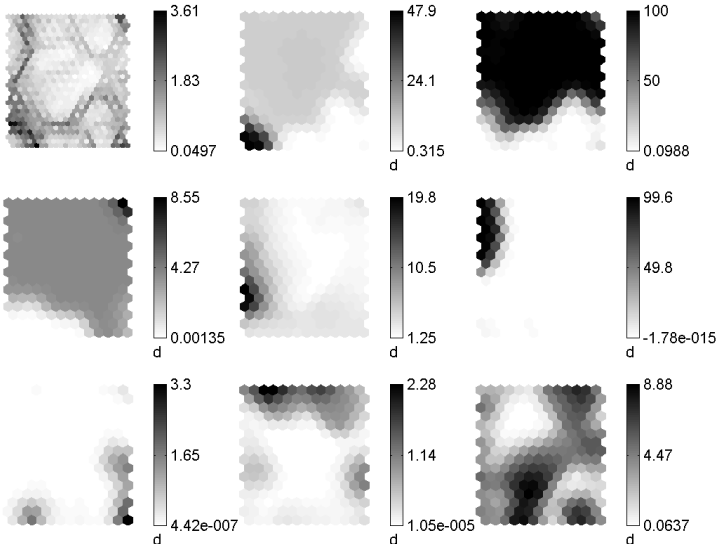


Figure 5: The SOM for the whole set and for each variables: v229,v240,v368 ,v283,v396 and v80

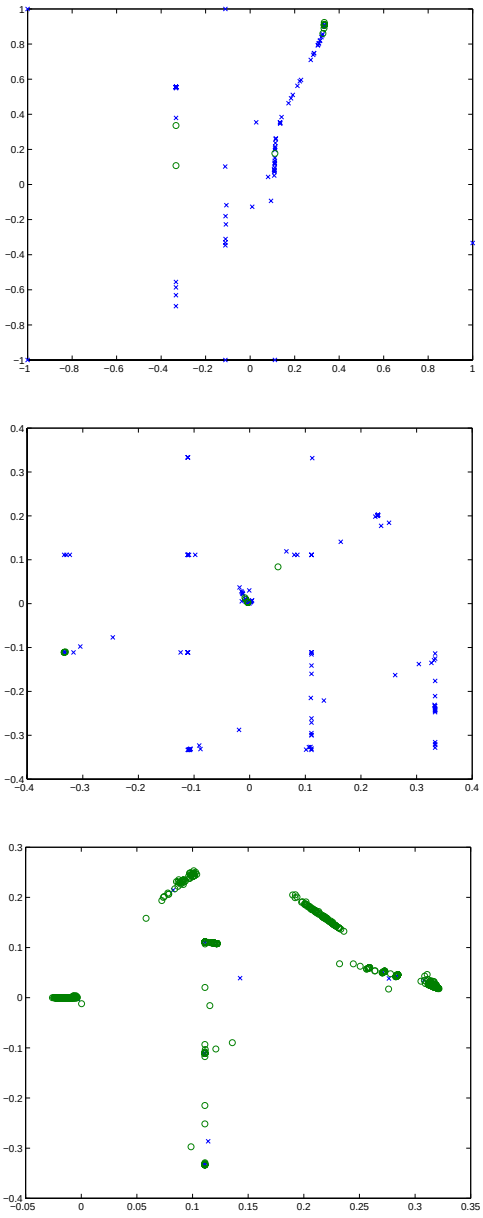


Figure 6: The hierarchical GTM plot of KDD data. The three plots are for clusters 1, 2 and 6 each from top to the bottom

cluster	1	2	3	4
# heavy /light spenders	76/9	141/10	62/0	0/38
wear frequently (·/overall%)	hosiery (34.2%/17.6%)	hosiery (29.6%/17.6%)	hosiery (38.3%/17.6%)	trouser socks (24.3%/15.2%)
hear from (·/overall%)	friend/family (6.6%/33.4%)	friend/family (9.9%/33.4%)	other (40.3%/29.4%)	other (54.1%/29.4%)
residence area (·/overall%)	west (27.6%/15.2%)	west (27.6%/15.2%)	east (53.2%/39.5%)	east (94.6%/39.5%)

Table 2: The composition of seven clusters and their characteristics

cluster	5	6	7
# heavy /light spenders	3/46	89/1297	3/117
wear frequently (·/overall%)	.	casual socks (34.3%/27.8%)	athletic socks (47.9%/17.0%)
hear from (·/overall%)	friend/family (46.7%/33.4%)		
residence area (·/overall%)	middle (26.7%/15.9%)	west (49.9%/15.2%)	

Table 3: The composition of seven clusters and their characteristics (cont'd from Table 2.)

into cluster #1 (mixed with 11.84% of light spenders). #2 (mixed with 7.09% of light spenders) and #3 (not mixed) and the light spenders are divided into clusters #4 (not mixed), #5 (mixed with 6.12 % of heavy spenders), #6 (mixed with 6.86% of heavy spenders) and #7 (mixed with 2.56% of heavy spenders). Each cluster shows its own characteristics (Table 2 and 3). One notable result is the formation of cluster #4 which is exactly the group of people who have more than 40% of discounted items in their ordering.

Facts from the KDD data indicate that customers who know the company by *hearing from friend/family* are light spenders ($8.80), by *other way* are heavy spenders ($32.19) and customers wearing *athletic and casual socks* are light spenders. Clusters #1 $\sim$ #3 reflect the first two facts and clusters #6 and #7 reflect the third fact (Tables 2 and 3).

In Figure 4, the analysis of KDD data by SOM is proved in U-Matrix (unified distance matrix) with labels of clusters and component-wise SOM plots for each feature are also provided. The U-matrix in the SOM visualizes the relative distance between the neurons by different tones of coloring scales (in Figure 4, darker color represent larger distances between neurons as indicated in the middle scale bar)

There appear about $7 \sim 9$ clusters in the SOM plot divided by dark boundary of the scaled grey level. Clusters #7, #4, #3 and #6 can be identified easily whereas clusters #1,#2 and #5 are expressed as two clusters each. In component-wise SOM (Figure 5), v229 (the rate of discounted item in order) and v240 (rate of friend promotion in order) highlight clusters #1∼#3, clusters #4 and #5 whereas v304 (minimum shipping amount) highlights #4.
For KDD data, PCA does not show the structure of the data since the complexity of data goes beyond linearity. In SOM, several clusters are visualized but there still remains some ambiguity since the expression is limited up to the node set. Much flexibility is allowed in GTM since data points can be expressed over the whole latent plane expressing each cluster more compactly. For instance, cluster #4 (lower left corner) in SOM plot (Figure 4) is expressed in GTM as a point reducing ambiguity greatly.
To understand the characteristics of data of complex features, we visualized hierachically the structure of data into the latent plane by GTM in Figure 3 for clusters 1,2 and 6. Depending on the purpose of analysis, each cluster can be further visualized.

4 Conclusions and discussions

We have used GTM for mining a real-life web data. Applied to the KDD Cup 2000 data, the results were compared with those of PCA and SOM. GTM showed a meaningful cluster structure and provided a clear underlying structure of clusters. Since GTM relies on a generative model for the visualization, features are assumed to take continuous measures and categorical variables are not considered in the formation of modeling. Missing values are treated as taking another value (= 0) in this analysis. If they are approximated properly, a more informative results can be expected. Automatic visualization system based on GTM can be developed with proper parameter selection plans as a future application. When the structure of data is changing dynamically through time, the corresponding visualization process would be much more complicated. Appropriate methods are waiting for the future development.

Acknowledgement

This work was supported by BK21-IT Program, Brain Science and Engineering Project and IITA.

References

[1] Ansari,S., Kohavi R., Mason, L. and Zhang, Z.(2000). Integrating e-commerce and data mining, *Technical report* Blue Martini Software, CA.

[2] Bishop, C.M., Svenson, M. and William, C.K.I. (1998) GTM: The generative topographic mapping. *Neural Computation*, 10(1).

[3] Dash, M. and Liu, H. (1997) Feature selection for classification. *Intelligent Data Analysis*, Vol.1, no. 3.

[4] Famili, A. and Bruha, I. (2000) Workshop on post -processing in machine learning and data mining: interpretation, visualization, integration and related topics. *The sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*

[5] Fayyad, U., Shapiro, P., Smyth, P., Uthurusamy R. (1996) eds. *Advances in Knowledge discovery and data mining.* CA, AAAI Press.

[6] Hand, D.J. (1981) *Discrimination and classification* New York: John Wiley & Sons.

[7] Hastie, T. and Stuetzle, W (1989) Principle curves. *Journal of the American Statistical Association, 84(406): pp. 512-516*

[8] Haykin, S. (1999) *Neural Networks: A Comprehensive Foundation.* Upper Saddle River, N.J., Prentice Hall. 2nd ed.

[9] Kiviluoto K. and Oja E. (1997) S-Map: A network with a simple self-organizing algorithm for generative topographic mappings. NIPS Vol.10, pp.549-555.

[10] Kohonen, T. (1990) The self-organizing map. *Proceedongs of the IEEE 78(9): pp. 1464-1480*

[11] Kontkanen, P., Myllymaki, P., Silander, T. and Tirri, H. (1998) BAYDA: Software for Bayesian classification and feature selection. CA, AAAI Proceedings.

[12] Quinlan J. (1986) Induction of decision trees. Machine Learning, 1:pp. 81-106.

[13] Rao, C.R. (1964) The use and interpretation of principle component analysis in applied research. *Sankhya A, 26, pp. 329-358.*

[14] Spiegelhalter, D.J., David, A.P., Lauritzen, S.L., and Cowell, R.G. (1993) Bayesian analysis in expert systems. *Statistical Science , 8, pp. 219-247.*

[15] Vesanto J. (1999) SOM-based data visualization methods. *Intelligent Data Analysis, Vol. 3, no. 2.*

[16] Vesanto J. and Alhoniemi E. (2000) Clustering of the Self-Organizing Map. *Transc. on Neural Networks, Vol.11, No.3, pp.586-601.*

Supporting Data Analysis Through Visualizations

Paolo Buono, Maria Francesca Costabile, Francesca A. Lisi

Dipartimento di Informatica, Università di Bari, via Orabona 4, 70125 Bari, Italy
{buono, costabile, lisi}@di.uniba.it

Abstract. Information visualization is a process that transforms information into a visual form, thus enabling the user to observe it. By graphically presenting data, the user may discover new and useful properties, their correlations, and also detect possible deviations from the expected values. In this paper, after discussing some ideas about possible fruitful use of visualization for data mining, we present a visualization module we are developing in the context of a project funded by the European Union. The project aims at offering on-line innovative services to support the business processes of trade fairs, both real and/or Web-based virtual fair. This module generates data visualizations on the WWW, which are exploited to facilitate human-computer interaction, to allow easy access to the stored data, and to present the retrieved information in appropriate ways, thus helping users in their data analysis activities.

1 Introduction and Motivation

Visual representation have the capability of shifting load from the user's cognitive system to the perceptual system. Indeed, information needs to be visualized in an information space in order to be retrieved by users. This visualization can either be carried out by the users in their own mind, in which case it is essentially the users' conceptualisation of that information, or it could be accomplished by the system, in which case the visualization is generated on the display screen. The latter is actually called *information visualization*, and is defined as "a process of transforming information into a visual form enabling the user to observe information" [1]. Recent research has proved that a suitable visualization can reduce the time to get information, and to make sense out of it. In the field of information systems, visualizations have a wide range of applications: they can be used for visualizing various types of meta-information, as well as queries and retrieved results. Moreover, by allowing dynamic user control of the visual information through direct manipulation principles, it is possible to traverse large information spaces and facilitate comprehension with reduced anxiety. In a few tenths of a second, humans can recognize features in mega-pixel displays, identify patterns and exceptions, recall related images. The use of proximity coding, colour coding, size coding, animated presentation, and user-controlled selections enable users to explore large information spaces rapidly and with fun.

Because of these characteristics, we believe that information visualization techniques may be very fruitful for data mining [2]. They may especially support post-processing activities necessary to fully understand the results of a data mining application [3]. Information visualisation can be also considered in some way a data mining technique itself, and the one that involves more interaction between users and system. Indeed, graphically presenting data may allow user to discover new and useful properties, their correlations and also detect possible deviations from the expected values. Use of colour may highlight data aggregations, use of animation may allow to quickly go through multiple levels of details (see for example [4]). A further advantage of this type of techniques is that users do not need to know what kind of phenomena they should observe in order to discover anything interesting or unusual. We are confident that useful techniques can be derived from the information visualization area to support the work performed by machine learning and data mining communities for analyzing huge amount of data.
In this paper, we describe how some visualization techniques can be advantageously applied in FAIRWIS (trade FAIR Web-based Information Services), a project funded by the European Union, which aims at offering on-line innovative services to support the business processes of trade fairs, both real and/or Web-based virtual fair. More specifically, the paper is organized as follows. In Section 2, we briefly survey some information visualization prototypes that primarily inspired our work. We then discuss some information visualization guidelines in Section 3. Section 4 describes the FAIRWIS project, while Section 5 illustrates the visualization techniques that are exploited to allow easy access to the stored data, and to present the retrieved information in appropriate ways, thus helping users in their data analysis activities.

2 Information Visualization Prototypes

We are all familiar with direct manipulation interfaces; their success testify the power of using the computer in a more visual manner. Direct manipulation is based on some fundamental concepts, such as the visualization of actions and objects of interest, the use of fast, incremental and reversible actions, and the immediate visualization of the result. Visual displays give the possibility of showing relationships by proximity, containment, connected lines, color coding, etc. Highlighting techniques, like blinking, brightening, reverse video can be used to focus the attention to specific items among thousands of items. Rapid selection can be performed by pointing to a visual display.
By visually presenting information, we exploit the potentiality of visual perception of human beings. Visual presentations are particularly useful since they allow users to activate perceptual procedures to quickly obtain the desired results. Such procedures substitute the logical inferences the user should perform without a visual presentation.
Exploring large multi-attribute databases is greatly facilitated by presenting information visually. Among different visualization techniques of databases proposed in the literature, Ahlberg and Shneiderman have proposed starfield displays [5], that plot items from a database as small selectable spots (either points

or small 2D figures) using two of the ordinal attributes of the data as the variables along the display axes. The shown information can be filtered by changing the range of displayed values on either axes. If this is done incrementally and smoothly, the result is zooming in and out on the starfield display, and the user can track the motion of the spots without getting disoriented by sudden, large changes in context.
The values of other attributes of the database can also be varied by the user through appropriate widgets that allow performing dynamic queries [6]. This is a very interesting visual query formulation technique (see [7] for a classification of such techniques), based on range selection, i.e. it allows a search conditioned by a given range on multi-key data sets. The query is formulated through direct manipulation of graphical widgets, such as buttons, sliders, and scrollable lists, with one widget being used for every key. The user can either indicate a range of numerical values (with a range slider), or a sequence of names alphabetically ordered (with an alpha slider). Given a query, a new query is easily formulated by moving the position of a slider with a mouse; this is supposed to give a sense of power but also of fun to the user, who is challenged to try other queries and see how the result is modified.
Higher usability is ensured if the query results fit on a single screen and are displayed quickly, i.e. within a second [8]. Moreover, input and output data are of the same type and may even coincide. As a consequence, dynamic query applications typically encode multi-attribute database items as dots or colored polygons on a starfield display.
A first application of dynamic queries is shown in [6] and refers to a real-estate database. There are sliders for location, number of bedrooms, and price of houses in the Washington, D. C. area. The user moves these sliders to find appropriate houses. Retrieved ones are indicated by bright points on a Washington, D. C. map shown on the screen. Another interesting application that combines dynamic queries and starfield displays is FilmFinder [5]; it allows information about movies to be retrieved by providing names of actors, actresses, or movie directors through Alphasliders, or values of other attributes through appropriate range sliders and buttons. The user can select some values by using a slider, and this first choice determines the set of values that can be selected with the remaining widgets. For example, if the user has selected a specific movie director, only names of actors and actresses who worked with that director can be selected next. This strategy is called tight coupling and it is aimed at preventing users from specifying null sets. In other words, query widgets and their related query formulation mechanisms are designed to interact with each other to avoid empty query results; this is achieved by restricting users to specify query criteria that lead to non-empty results. A tightly coupled query is then a series of filters selecting a subset of a database. For each new filter that is set, users can only select values of the remaining filters that let through at least one database object still existing after the last filter.
Dynamic queries are also called direct-manipulation queries, since they are based on the same fundamental concepts of direct manipulation illustrated above. One of the big advantages of such interaction technique is that it allows focusing the attention on the task users have to perform. Objects of interest are all displayed so that actions occur in the high level semantic domain. Each command is a comprehensible action in the problem domain whose effect is immediately visible; this relieves the user from the burden of decomposing tasks into syntactically complex sequences, thus reducing user load in problem solving. The sliders are a

good metaphor for the operation of entering a value for a field in the query: changing the value is done by a physical action instead of entering the value by a keyboard. Such action is easily reversible by moving the drag box, if the obtained results are not what users expected. No action is illegal, hence error messages are not needed. More references to work on dynamic queries can be found in [9].

At Xerox PARC in the last ten years a group of researchers has developed several information visualizations, with the aim of helping the users understand and process the information stored into the system [10, 11, 12, 13]. They have created the "information workspaces", i.e. computer environments in which the information is moved from the original source, such as networked databases, and where several tools are at disposal of users for browsing and manipulating the information. One of the main characteristics of such workspaces is that they offer graphical representations of information that facilitate rapid perception of the overall patterns. Moreover, they use 3D and/or distortion techniques to show some portion of the information at a greater level of detail, but keeping it within a larger context. These are usually called fisheye techniques [14], or alternatively focus + context, that better gives the idea of showing an area of interest (the focus) quite large and with detail, while the other areas are shown successively smaller and in less detail. Such an approach is very effective when applied to documents, and also to graphs [15]. It achieves a smooth integration of local detail and global context. It has more advantages of other approaches to filter information, such as 1) zooming or 2) the use of two or more views, one of the entire structure and the other of a zoomed portion; the former approach shows local details but looses the overall structure, the latter requires extra screen space and forces the viewer to mentally integrate the views. In the focus + context approach, it is effective to provide animated transitions when changing the focus, so that the user remain oriented across dynamic changes of the display avoiding unnecessary cognitive load. The Perspective Wall [10] provides a good example. For other techniques developed at Xerox PARC see [11].

Numerous prototypes have been proposed for information visualization. The ones mentioned above are among those providing novel ideas that have inspired our work. A very good reference for a survey of information visualization techniques is [16].

3 Supported Tasks in Information Visualization

There are many visual design guidelines. A central principle for information visualization might be summarized in the Shneiderman's Visual Information Seeking *Mantra "Overview* first, *zoom* and *filter*, then *details on demand*" [17]. The overview allows the user to grasp the content of the application and its distribution across the different attributes. Providing an overview is particularly useful in WWW interfaces for information systems that give users direct access to the content and interconnections within an information domain. WWW navigation should be stimulating and attractive for the users; unfortunately, due to the large amount of accessible information, the search of some detailed information can often become a long and complex activity for the user. One of the main problem is the

difficulty users have in generating their mental model of the system they are interacting with; it can be difficult for them to grasp the kind of information stored and the modality for managing it. Such a problem is particularly serious since WWW interfaces are mostly used by occasional users, who are not willing to perform an in-depth study, but need to easily grasp the kind of information they can have and want to get it quickly.
Zooming is another interesting task, since users typically have an interest in some portion of a collection, and they need tools to enable them to control the zoom focus and the zoom factor. A satisfying way to zoom in is to point to a location and to issue a zooming command. Smooth zooming helps users to preserve their sense of position and context. Another popular approach for keeping the context while zooming some areas of interest is the fisheye strategy [14]; the fisheye distortion magnifies one or more areas of the display.
Users may filter out uninteresting items, so that they can quickly focus on item of interest. Dynamic queries applied to the items in the collection constitute one of the key ideas in information visualization [5]. Sliders, buttons, or other control widgets coupled to rapid display update are used for the filter task.
We can select an item or a group of items to get details. Once we have obtained a few dozen of items, it should be easy to browse the details about the group or individual items. The usual approach is to simply click on an item to get a pop-up window with values of each attribute. In Spotfire [18], the details-on-demand window can contain text with links to further information.
Besides the four tasks explicitly mentioned in the Shneiderman's Mantra, three other tasks are very useful in information visualization, namely *relate*, *history*, *extract*. Referring to the first, users can view relationship among items. In the FilmFinder details-on-demand window [5] users could select an attribute, such as the film's director, and cause the director Alphaslider to be reset to the director name, thereby displaying only films by that director. The Table Lens emphasizes finding correlations among pairs of numerical attributes [11].
We can keep a history of actions to support undo, reply, and progressive refinement. Information exploration is inherently a process with many steps, thus keeping the history of actions and allowing users to retrace their steps is important.
Once the users have obtained the item or the set of items they desire, it would be useful for them to be able to extract that set and to store into a file in a format that would facilitate further uses, such as sending by e-mail, printing, inserting into a presentation package. As an alternative to saving the result set, they might want to save the settings for the control widgets.

4 Data Analysis in Trade Fairs: the FAIRWIS Project

FAIRWIS (Trade FAIR Web-based Information Services) is an on going project at the University of Bari, funded by EU this project aims to offering on-line innovative services to support the business processes of real trade fairs as well as providing information services to a great number of exhibitors organised in a Web-based virtual fair. FAIRWIS has a real time connection with an underlying database to guarantee coherence of data and up-to-date status.

Traditionally, information media for supporting trade fair events is paper-based: booklets, flyers, maps, etc. are the means used to exchange information. In recent years, some Web-based information sites have been made available, providing information both on trade fair events and on companies participating in these fairs. However, these data are not organised in an integrated, homogeneous and comprehensive way, since are usually presented in a rigid pre-designed company oriented style. Moreover, currently available Web sites exploit static data that it is difficult to update and to put on-line in an appropriate format.
Presenting data on the WWW in a convincing and understandable way requires a lot of work when data change dynamically; in particular it is difficult to modify the graphical layout without disorienting the users. We describe a module of the FAIRWIS system, devoted to the generation of data visualizations that are exploited to facilitate human-computer interaction and to allow easy access to the stored data. By presenting the retrieved information in appropriate ways, specific categories of user to whom FAIRWIS is primarily addressed, namely fair organisers, exhibitors, and professional visitors (people who visit the fair for business reasons and not only for fun), will get a valuable help in the different phases of the decision making processes they may undergo to improve their own business.
FAIRWIS aims to both support real trade fairs and offer on-line innovative services regarding a virtually unlimited number of companies, products and events. The whole concept of trade fairs is transferred into an electronic form, and visualisation techniques, including virtual reality, are used in order to provide “reality” feelings to the users of trade fair information systems. The project does not aim at substituting, but at enhancing the existing traditional approach of getting people together.
The software module here described presents a WWW interface that allows users to easily retrieve information useful for their marketing activities. The aim of the FAIRWIS marketing component is to manage and improve interactive relationships among the FAIRWIS users. More specifically, fair organisers and company managers may forecast their company activity on the basis of history data available in the database. To this purpose, they need techniques that enable them to discover specific trends of the stored data. To provide support to the users in this process, we have developed a prototype that exploits an information visualisation technique known as query previews [19]. It is presented in the next section.

5 Analyzing FAIRWIS Data Through Query Previews

Within the FAIRWIS project we examined hundreds of Web sites related to trade fair events and we discovered that most of them do not provide any mechanism for analysing the data they may possibly store in underlying databases. A web site that makes some kind of data analysis is CIBUS, a fair event of Fiera di Parma trade fair, which stores in a database data of the exhibiting companies, and makes them available for the final user on the WWW [20]. The users can make query by choosing the filter criteria of several attributes, thus retrieving the companies that fit their needs.
As shown in Figure 1, the user may choose among the attributes such as the

company sector (*settore di produzione*), the product type (*tipologia*), the company income (*classe di fatturato*), the number of employees (*numero dipendenti*) and so on. When the user has selected some of these attributes, he or she clicks on the button "*inizia ricerca*" to retrieve the companies of interest. This kind of interface has some drawbacks. It does not provide the user any hints about the actual content of the database, for example the user doesn't know if there are exhibitors whose company sector is *carni congelate* or if there are exhibitors coming from a certain town. As a consequence, after the user has spent time for inputting several attribute values, very likely he or she gets as result an empty dataset, as shown in Figure 2, where the message in Italian says that the search did not find any company.

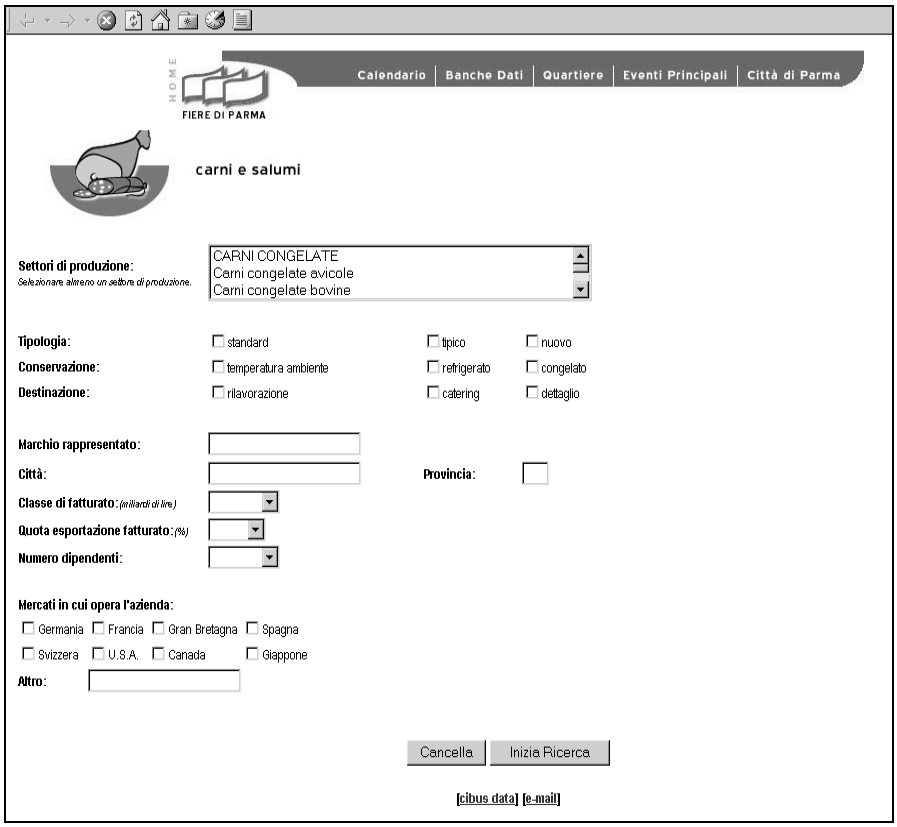


Fig. 1. A screenshot of CIBUS: the user can input values of several attributes to find the companies of interest.

Fig. 2. The result of a query in CIBUS often gives an empty data set.

To overcome these problems, in FAIRWIS we designed a visualization tool that provides users a rapid overview of the information stored in the fair database in order to support the users' data analysis. This overview shows the data distribution along some major attributes. Then, we use dynamic queries and query previews to support efficient query formulation [19]. As we said in Section 2, dynamic query user interfaces apply the principles of direct manipulation and imply: 1) visual representation of the query and of the results; 2) rapid, incremental and reversible control of the query; 3) selection by pointing (no typing); 4) immediate and continuous feedback.

Query preview interfaces provide the possibility of easily getting preliminary information about data interesting for the user, making visible the problems or gaps in the metadata that are undetectable with traditional form fill-in interfaces. In this way, the user may rapidly eliminate undesired datasets and also preview the size of the result set to avoid the so-called zero-hit queries, i.e., queries that provide an empty set as result.

In order to see how a query preview interface works, let us refer to the FAIRWIS prototype. Let us suppose the organiser of a fair on agriculture wants to perform a segmentation of the exhibitors of the last edition of the fair. In order to help the users (in this case fair organisers) in their analysis, we allow them to some major attributes for generating a data overview and then perform some query previews.

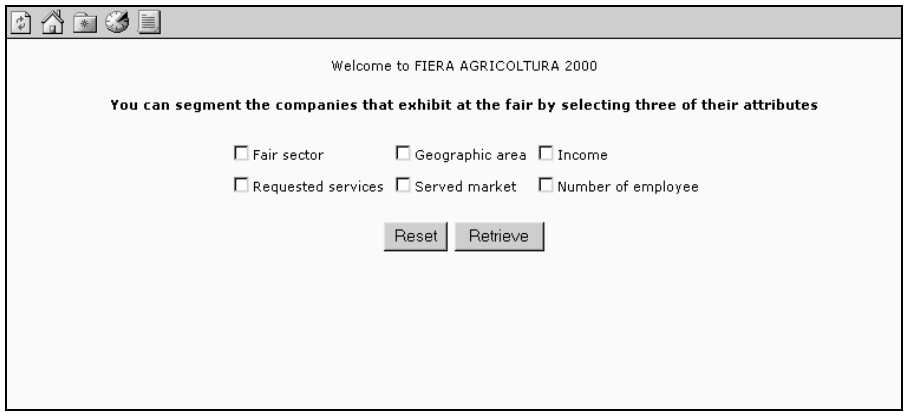


Fig. 3. A first Web page in which a user is invited to select three major attributes.

In Figure 3, the user can select some attributes (three in this prototype) of interest. Let us to suppose that the user selects the attributes *Fair sector*, *Geographic Area*, *Requested services*. The resulting overview is shown in Figure 4. In the overview, data distribution is displayed along these attributes.

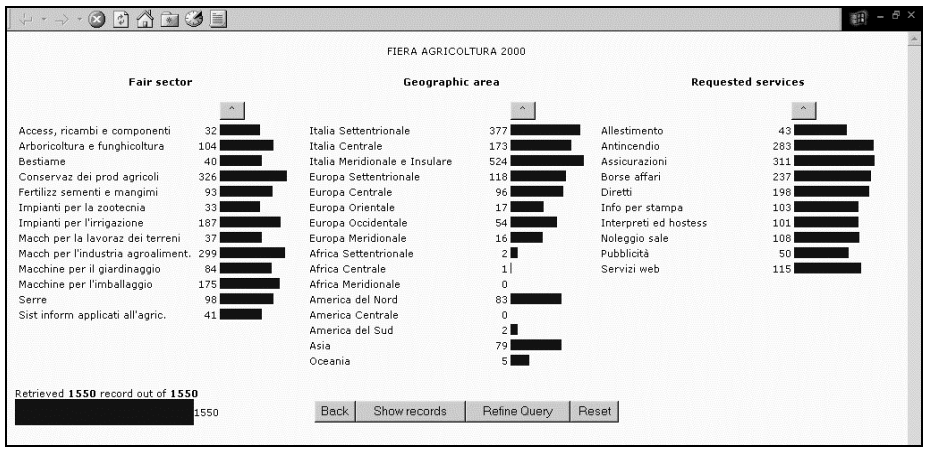


Fig. 4. A Query Preview interface for Fiera Agricoltura 2000.

The user immediately gets a lot of information from the overview: for example, no company comes from South Africa (*Africa meridionale*) so that it is useless to perform any query with *Africa meridionale* as *Geographic Area* since it will return a empty data set. The user also sees that 96 exhibitors come from Central Europe (*Europa centrale*), and so on.
The interface allows the user to perform previews of data, for example by clicking on the value *Conservaz dei prod agricoli*, only the records with this attribute value

will be selected and the number of retrieved data is updated consequently, as shown in Figure 5. We also provided the possibility of sorting the shown elements. The user can sort the values in numerical order by clicking on the icon on top of the numerical value of the retrieved record.
If the query preview shows too many records, it would not be useful to visualize all of them. In this case, the system allows query refinement by clicking on the button "Refine query". The query refinement phase supports dynamic queries over other relevant attributes of the database. In this way, the user can get a reduced set more meaningful for his or her interests. Once we get the list of the retrieved records, details on a specific record can be obtained by clicking on an element of the list, and a window with all available information on that specific company appears on the screen.

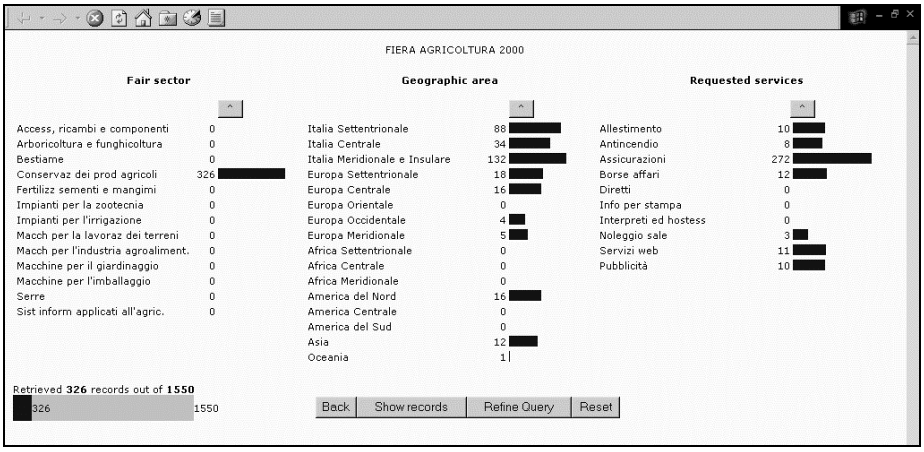


Fig. 5. Results of a query preview. The user has selected the value *Conservaz dei prod agricoli* for the attribute *Fair sector* and only the 326 companies exhibiting in that sector are retrieved.

6 Conclusions

The prototype presented here is part of the on-going project FAIRWIS carried out with other partners in the European Union. FAIRWIS primarily addresses the needs of professional users in the fair context, namely fair organisers, exhibitors, and visitors attending the fair for business reasons.
We have shown some functionalities of a data analysis engine module that allows users to easily retrieve information useful for their marketing activities. The module exploits information visualization techniques that are capable to give to the users useful hints on meaningful patterns of data. As Klösgen says, "knowledge discovery in databases can be divided into paradigms such as search, visualization, navigation, and low level strategies for searching and evaluating patterns. Visualization gives a

feeling for the contents of the data and presents findings" [21]. Our work is in accordance with this perspective.
Other tools are under development in order to allow the FAIRWIS users to manage their interactive relationships, and to improve their company activity by letting them to directly take full advantage of the data available in the system database.

Acknowledgement

We are grateful to Marco di Fonzo for his help in implementing the prototype. The support of European Commission through grant FAIRWIS IST-1999-12641 is acknowledged.

References

1. Card, S., Eick, S. G., and Gershon, N. (1997) "Information visualization", CHI97 Tutorial Notes, Atlanta, GA, March 1997, pp. 22-27.
2. Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (Eds). Advances in Knowledge Discovery and Data Mining. AAAI Press/The MIT Press, 1996.
3. B. Moxon, "*Defining Data Mining*", DBMS online, 1996, http://www.dbmsmag.com/9608d53.html
4. Silicon Graphics, "*MineSet*", http://www.sgi.com/software/mineset, 2001.
5. Ahlberg, C., Shneiderman, B. "Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays", Proceedings ACM CHI'94: Human factors in computing systems, pp. 313-317.
6. Shneiderman, B., Williamson, C., Ahlberg, C. "Dynamic Queries: Database Searching by Direct Manipulation", Proceedings ACM CHI'92: Human Factors in computing Systems, pp. 669-670.
7. Catarci, T., Costabile, M. F., Levialdi, S., Batini, C. "Visual Query Systems for Databases: a Survey", Journal of Visual Languages and Computing, Vol. 8, 1997, pp. 215-260.
8. Ahlberg, C., Williamson, C., Shneiderman, B. "Dynamic Queries for Information Exploration: An Implementation and Evaluation", Proceedings ACM CHI'92: Human Factors in computing Systems, pp. 619-626.
9. Shneiderman, B. "Dynamic Queries for Visual Information Seeking", IEEE Software, 11, 1994, pp. 70-77.
10. Robertson, G.G., Card S.K., Mackinlay, J.D., "Information Visualization Using 3D Interactive Animation", Communications of the ACM, 36 (4): 7-71.
11. Rao, R., Pedersen, J.O., Hearst, M.A., Mackinlay, J.D., Card, S.K., Masinster, L., Halvorsen, P.-K., and Robertson, G.G. (1995) "Rich interaction in the digital library", Communications of the ACM, 38 (4): 29-39.
12. Card S. K., G.G. Robertson, W. York "The WebBook and the Web Forager: An Information Workspace for the World-Wide Web", Proceedings CHI'96, April 13-18, 1996, pp. 111-117.
13. Card S.K., "Visualizing Retrieved Information: A Survey", IEEE Computer Graphics and Applications, March 1996, pp. 63-67.
14. Furnas, G.V., "Generalized Fisheye Views", Proceedings CHI'86 Conference: Human Factors in Computing Systems, ACM Press, New York, pp. 16-23.

15. Sarkar M., Brown M.H., " Graphical Fisheye Views", Communications of the ACM, 37(12): 73-84.
16. S. Card, J. MacKinlay, B. Shneiderman. Readings in Information Visualization. Morgan Kaufmann Publisher Inc., 1999.
17. Shneiderman, B. (1996) The eyes have it: A task by data type taxonomy for information visualization, *Proc. 1996 IEEE Symposium on Visual Languages*, Boulder, Colorado, September 1996, IEEE Computer Society Press, Los Alamitos, California, pp. 336-343.
18. C. Ahlberg, IVEE Development AB, "*Spotfire*", http://www.ivee.com, 2001.
19. Shneiderman B., C. Plaisant, K. Doan, T. Bruns, "Interface and Data Architecture for Query Preview in Networked Information Systems", *ACM Transaction on Information System*, vol. 17, No.3, 320-341, July 1999.
20. http://www.fiere.parma.it/CibusData/irpc0.htm
21. Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (Eds). Advances in Knowledge Discovery and Data Mining. AAAI Press/The MIT Press, 1996, pp. 249-271.

Introducing Signature Exploration: a means to aid the comprehension and choice of visualization algorithms

Penny Noy and Michael Schroeder

Department of Computing, City University
London EC1V 0HB, UK
{p.a.noy,msch}@soi.city.ac.uk

Abstract. Visualization is increasing its role in the search for meaning in complex data. Two important issues are: a) choice of visualization method; b) comprehension of resultant visualizations.

This paper proposes a new concept, **signature exploration**, which is defined as the exploration of the behaviour of a visualization algorithm by means of the use of specially constructed data sets. Five types of constructed data are suggested; generic, constructed, query, landmark and feedback.

Two examples of signature exploration are described - a feasibility test and a feedback application. The feasibility test results indicate the value of the concept and the desire of users for an interactive interface for the entry and exploration of datasets containing specified features. The feedback application involves user placement of a subset of entities, which enables comparison with the various placements for different algorithms, as well as arrangement of other entities not in the original subset.

1 Introduction

Many people recognize the increasing importance of visualization in examining the mass of data that surrounds us - identifying the value of existing visualization techniques, as well as the need to explore the new possibilities offered by technological advances to extend the role and range of types of visualization. Researchers in the field of cognition and perception urge us to take advantage of the substantial work completed in this area [18], while others point to the difficulty of applying this work [7].

In looking at information visualization from a data mining perspective, a complex data set must be transformed to a level at which it can be displayed, as constrained by the medium. A general loss of comprehensibility usually results. A 276 dimensional matrix of data for 100 entities can be reduced to a neat scatterplot (see figure 5), but can sense be made of the resulting patterns?

Thus we consider the loss of meaning associated with visualization transformations of complex data. The ideas presented here have developed out of initial visualizations of data involving dimension reduction using the tool Space Explorer [13,15,14] and in the context of ongoing work to address the problem of visualizing complex data. Space Explorer is a visualization application for multivariate and proximity data. In an initial investigation it was shown how one data set can be displayed in a number of different

ways, producing different clusters and outliers (see again figure 5). This is a well known problem, but one for which general solutions are not evident.

Intuitively the user wants initially to construct sets of data (rather than starting with a large unknown one) and see what the visualization algorithm does with these test sets. We take a data set that we feel we *know* and see what it looks like in the visualization. We think this will help us in two ways, firstly to get a concrete feel for how the algorithm or tool behaves, secondly to better understand the result obtained with a large unknown data set. It may be possible to see what algorithm best suits the particular data and the type of questions we seek to answer about it. At the same time, as the user wants to intuitively learn and understand the algorithm of the application, the application should be able to learn the user's intuition of the data.

From such ideas and experience we propose the use of constructed data sets as a general design feature of visualization tools to aid comprehension and presentation of complex data. We call this **signature exploration** (please see acknowledgement at the end of the paper).

2 Introducing signature exploration

We define **signature exploration** as the exploration of the behaviour of a **visualization algorithm** by means of the visualization of specially constructed data sets. In this way **known** data sets are visualized for the user as concrete examples of the behaviour of the algorithm. The visualization result, the pattern produced, is the *signature* of the algorithm for that data. Different algorithms will produce their own corresponding signatures for the data set. The data set may be one of a set of standard types provided, or any set constructed by the user. Thus the signature of the algorithm is explored for sets of known data. By *known* is meant that the user has a sense of knowing the data, in a concrete but not necessarily precisely defined way. By *visualization algorithm* is meant any application, tool or algorithm that produces a visual representation of data.

The purpose of signature exploration may be solely to understand the behaviour of a visualization algorithm, or the comparison of different algorithms so as to make an appropriate selection or classification. The modification of the original dataset, or the visualization algorithm, by providing feedback data (see item 5 below) is also considered to be a means of exploring the signature of the algorithm.

As an initial suggestion we outline five types of constructed data for exploration.

1. **Generic**: characteristic sets of data that illustrate the various behaviours of metrics and visualizations.

   The idea of generic data sets is to provide the user with a range of data sets showing specific features, so that they can form a more concrete impression of the behaviour of the algorithm and to assist in the comparison of behaviours of different algorithms. The extent to which generic data sets can be identified, that are illustrative in this respect, is unclear at this stage of our work. The usefulness of the data sets is considered on two levels. On a familiarization level the provision of example sets containing, for instance, identical entities or data sets containing no structure

(random values), together with a variety of examples of phase shift [1] and scaling of shapes across variables is suggested. Such basic presentations of data are useful because, in our experience it is not always immediately obvious how they will be displayed - due to dimension reduction or unfamiliar presentation (eg hierarchical axes, parallel axes), but they also serve to focus the user upon developing their understanding of the behaviour of the algorithm(s). The other level of usefulness is the more challenging question of which algorithms map specific features (phase shifted patterns, for instance) to clusters. It is not clear whether progress may be made on this issue, but the provision of a data construction and manipulation interface within the visualization application will assist.

That the visualization application designer seek appropriate data sets, to illustrate the behaviour of the algorithms they employ, and make these available to the user in an interactive interface, may prove to be a desirable design requirement for visualization systems.

Some initial examples of generic type data sets are as follows:-

- Data is of two or more groups of identical entities (identical in the sense that their data table entries are identical).
- Data is of two approximately equal sized groups. The first group contains identical entities; the second group varies one variable in equal steps away from the value in the first.
- As in previous set, but with the variation of a different variable for each member of the second group
- Data contains entities with variables as follows: overall 'shape' the same, magnitude altered; shape not the same, magnitudes as before; increase number of shapes and grups.
- Use of mathematical functions to specify data.

2. **Constructed**: static and simulated. By *static* is meant the direct specification of a matrix, *simulated* refers to a matrix derived from a log of events of a set of entities with specified behaviours.

   This type of data is constructed by the user. Static constructions are matrices specified by the user which can then be visualized. The variable values for each entity may be entered individually or according to a formula, or representing a scaling or phase shifting of values of another entity. They are static in the sense that they are an instance of creation by the user, as opposed to simulated constructions which are the result of data produced by a simulation of entity behaviours. Perhaps the user looks at their own real-world data set of interest and hypothesizes about the entity behaviours that would produce such data. On a complex level this would result in system simulations and possible prediction models. In simpler terms it is an invitation to the user to think about the data in a different way and derive questions and hypotheses which can then be examined. It thus extends the question - 'if my data looked like this what would the visualization look like?' to 'if my data was produced by these behaviours what would the visualization look like?'.

   Preliminary work has indicated the usefulness of taking supplied, generic type, data sets as a starting point and then giving provision for interactively changing shapes

[1] If variables are considered to be a time series, irrespective of whether they actually are, then a displacement of a pattern can be described as a phase shift

or values. Thus the starting point for the constructed data type may be a generic data set.

3. **Query** (based on a data set under consideration): a subset of the data, which may be directly selected by the user (either from the visualization itself, or by querying the original dataset) or automatically derived (eg outliers, extremities).
   A cluster in a visualization of a user data set may be highlighted, or an outlier, or the extremities of a pattern and this form the constructed data for manipulation. Alternatively the data set may be queried in an SQL type query to create a subset. Some of these techniques are well known and widely used. Here the purpose is to explore the behaviour of the visualization algorithm. Although ultimately the discovery of knowledge in the data remains the goal, it is indirectly so.
4. **Landmark**: one or more entries, which may be the result of queries or static constructions, to add to (or highlight within) the data set under consideration.
   Landmark and query overlap as concepts. In the landmark use of constructed data, entities are placed to provide landmarks in the user's mind as well as in the visualization. The entities may be invented, static constructions, or identified by a query.
5. **Feedback**: the user arranges a set of entity-representatives (or clusters of entities) on the screen for which data is also provided. The system uses the user layout information for the display of subsequent data by, for example, weighting the given attributes or selecting the algorithm that provides the closest layout to the user defined one.
   Concepts of similarity may be very subjective, as is clear in the case of comparison of image and video data. However, to some extent many comparisons have a subjective aspect, if only from the point of view of the user's particular enquiry or perspective. The user may also be unable to articulate, or even be aware of, relevant domain knowledge that they have. The feedback idea comes from a reversal of the process known-data-to-visualization. The user is asked to position a number of entities on the screen such that the distances between them represent their similarity (or measure of connectedness) according to the user's perception. It is assumed that there is multivariate data also available for these entities, so that the system can derive a layout that provides a mapping between the two (which may necessarily be approximate) and thus provide a means of displaying unknown data according to the user's classification. This is considered to be signature exploration by signature modification, although the simplest application would select the algorithm which gave the layout closest to that specified by the user.

3 Algorithm description

Space Explorer [13,15,14] implements a number of algorithms for visualizing multivariate and proximity data. Proximity data specifies a *distance* between entities which may be a direct measurement or derived from the multivariate data. Figure 1 shows some of the ways in which multivariate data can be converted into distance data. Coordinates for a scatterplot are derived such that distances between objects reflect their relationship. To this end, there are two techniques used here: Principal Component

Analysis, which is applicable to multivariate data, and multidimensional scaling (see e.g. [5,3,8,9,19]), which is applicable to similarity data.

1. Let $x,y \in R^n$ be vectors. Then $(x,y) = \sum_{i=1}^{n} x_i y_i$ is called scalar product and $||x|| = \sqrt{\sum_{i=1}^{n} x_i^2}$ Euclidean norm.
2. Direction cosines: $d_{cos}(x,y) = cos\theta = \frac{(x,y)}{||x||||y||}$ and angle (angular distance): $d_{angle}(x,y) = acos(d_{cos}(x,y))$
3. Euclidean distance: $d_e(x,y) = ||x-y|| = \sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}$
4. Minkowski distance: $d_m(x,y) = (\sum_{i=1}^{n} |x_i-y_i|^{\lambda})^{\frac{1}{\lambda}}, \lambda \in R$
5. Chebychev distance: $d_{cheb}(x,y) = max_{1 \le i \le n}\{|x_i-y_i|\}$

Fig. 1. Different distance definitions for vectors.

The choice of an appropriate distance is the most crucial step in the visualization process, and can affect drastically the quality of the result. Figure 2 illustrates this with two simple examples.

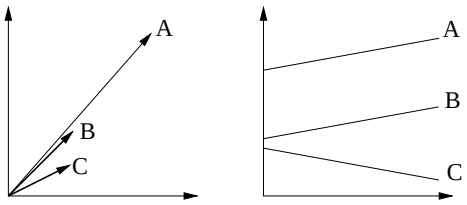


Fig. 2. The choice of distance influences data analysis. On the left are three vectors A,B,C and on the right three time series A,B,C. In both cases, B is close (far) to C (A) for Euclidean distance, but B is close (far) to A (C) for angular distance.

3.1 Principal component analysis

A multivariate table with m columns can be seen as a mapping of objects into an m-dimensional space. Graphical representations, e.g scatter plots, are however restricted to 1, 2, or 3 spatial dimensions and up to 8 dimensions (colour, shape, orientation, surface texture, motion coding and blink coding) overall [18]. The problem is thus to reduce the multivariate table to the most representative dimensions. This is the purpose of Principal Component Analysis (PCA), which transforms the m variables into m factors, each factor being a linear combination of variables (see Figure 3). The m factors are ordered by importance: the first factor explains as much as possible of the differences among objects, the second factor as much as possible of what cannot be explained by

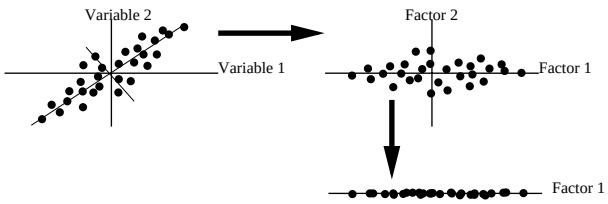


Fig. 3. Principal component analysis.

the first factor, and so on. A simple two-dimensional example is shown in Figure 3. The example can be generalized to *m* initial variables, where the 2 or 3 most important factors are displayed in a 2D or 3D space.

3.2 Multidimensional scaling: from distances to coordinates

The idea of finding points in space which satisfy some given distances dates back to the late 1960s [12], and is referred to as multidimensional scaling [5,3,8,9,19]. There are various approaches to the problem of multidimensional scaling, the one used here is Principal Co-ordinate Analysis.

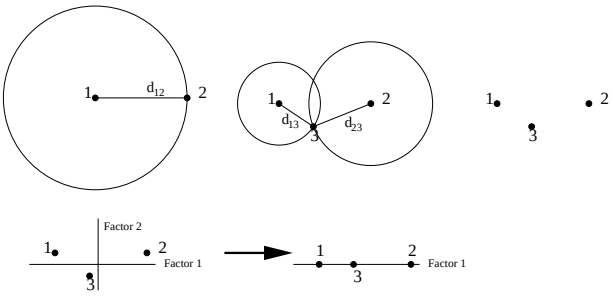


Fig. 4. Multi-dimensional scaling with principal co-ordinate analysis (PCoA).

Principal Co-ordinate Analysis In the principal co-ordinate analysis (PCoA) approach [6], a set of equations that relates distances to coordinates is solved. In this case, distances are known and the coordinates are the variables to be determined. Figure 4 illustrates the construction of a solution for three points in a 2D space, according to predefined distances. Intuitively, the positions can be constructed with a compass. For placing the three nodes in Figure 4 we start by placing the first node randomly and drawing a circle with radius d_{12} around it. Place node 2 anywhere on this circle. Then obtain node 3 by drawing a circle with radius d_{13} around node 1 and drawing a circle

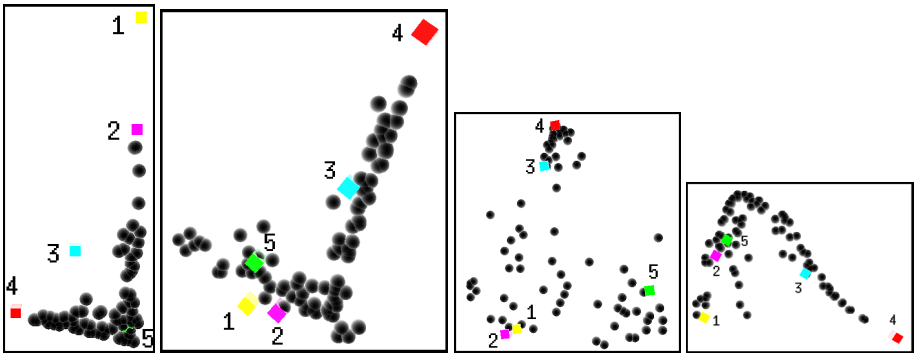


Fig. 5. 90,000 calls made by 100 customers. Caller profiles by destination of calls. These are screenshots of 3D VRML worlds. From top left to bottom right: (1) direct visualization by applying PCA, (2-4) indirect visualization by calculation of a distance followed by matrix transformation (PCoA); Distances in particular: (2) Euclidean, (3) angular distance, (4)Minkowski distance with $\lambda = 10$ The customers represented as squares and labelled with a number can be traced through the different visualization outcomes.

with radius d_{23} around node 2. Node 3 is placed at the intersection of circles d_{13} and d_{23}. This method determines if there is a solution to the problem and constructs it. If we abstract from this geometric construction, it turns out that we actually solve quadratic equations. And instead of solving them iteratively, we can solve them simultaneously. This gives us a possibly higher-dimensional solution, which we then reduce as we did for PCA. The possibly $n-1$ dimensional exact solution can then be reduced to an approximate 2D or 3D solution by selecting the first two or three axes, respectively.

4 The problem of algorithm choice

The identification of valid clusters in the data, for data simplification or prediction, is the goal of classification (as defined by Gordon [5]). Since different algorithms produce different clusterings, the question is how to make a valid choice, if one exists. An example of the different clustering obtained for a set of data is shown in figure 5. This is a set of British Telecom data of 90,000 calls made by 100 customers from one particular area. The data set was cleansed, by BT, of all private information. Thus the originating and destination exchange references were available, but not the complete originating and destination phone numbers. These visualizations use the destination local exchange reference in a data table, such that entry x_{ij} is the number of calls made by customer i to location j. Notice the clusters and outliers differ as do the pattern shapes [15].

5 Illustrations of Signature Exploration

Contemporary visualization systems contain many elements for assisting the user's exploration of the data (general references: [17,2,1]. Special features such as brushing

and for context and focus control (eg the semantic lens, hyperbolic browsers) have been developed. Querying of data with conventional database query language and dynamic querying within the visualization itself (eg Attribute Explorer [16]) are much used. Visual selection and reordering of that data are also employed, for example in the context of a colour map (eg GenExplore [4]) or directly from a datatable. These features promote the exploration of both the data *and*, intrinsically, the algorithm. Signature exploration focuses not on the data itself, but on the algorithm's behaviour, not as an end in itself, but as a process within and adjacent to that of exploring the dataset. The many techniques available to assist exploration of the dataset, instanced above, fall within the scope of this concept.

In beginning the work to assess the value of signature exploration, we have started with the constructed data types *generic* and *feedback*, since these appear the least provided for in current visualization systems: a preliminary feasibility test was set up to look at generic data; an initial example of user layout to provide feedback was developed to illustrate the concept. These are described below. It should be stressed that these descriptions are included solely for illustrative purposes and are not intended to represent a validation of the concepts.

5.1 Signature exploration feasibility test

Do our visualizations actually work? This question was asked at a recent conference [11] and statistics from conference papers given that showed less than 10% had carried out evaluation. Informal testing in the early stages was indicated to be beneficial and our feasibility test [2] is of this nature. Twelve participants were briefed about the domain of our work and then given a series of web pages to examine in combination with a paper questionnaire. The test first illustrated the problem by displaying the second of the visualizations of the call data set of figure 5, which also gave the user the opportunity to familiarize themselves with navigating in 3D. They were asked to note any conclusions they were able to draw at this stage from the pattern of the data. A series of 3D visualizations of simple datasets followed (using the same algorithm - Euclidean distance calculation followed by layout with Principle Co-ordinate Analysis using Space Explorer). The data tables were shown, together with the data shown as time series. Figure 6 is an example web page from the test. Most of the questions were to guide the exploration of the material. The key questions at the end were:

1. Do you think these explorations of constructed data sets have increased your understanding of the behaviour of the visualization algorithm? Results: Yes (5) No (3) Not sure/not much (4)
2. Do you think that an interface which allowed you to construct your own data, either from scratch or to modify given ones, would be useful? Results: Yes (10) No (2)

Although this was an informal test, it indicated that users would like an interface that allowed them to enter and explore their own example data sets or use the ones

[2] The web pages and questionnaire are online at www.soi.city.ac.uk/homes/dk707/webTest/main.html and Instructions_and_questionnaire2.doc

supplied as starting points for manipulation. Also that an interactive exploration of the way data values affect the visualization could enhance the user's understanding of the algorithms used and assist in the appropriate choice of metric if relevant.

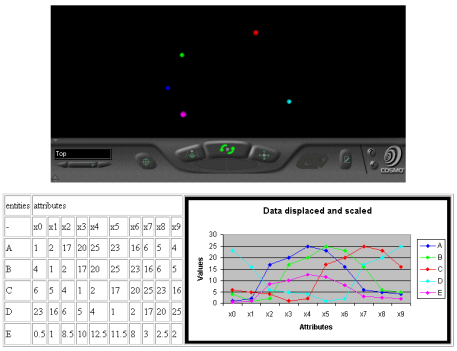


Fig. 6. An example webpage from the website feasibility test

6 Feedback

The initial inspiration for signature exploration, and for the particular aspect of modification of the algorithm by user classification, came from work on dynamic querying of image libraries (eg [10]). Starting from a particular image, users query the library for similar images. Since the selection and weighting of feature lists for images is such a complex and subjective task, the user is also invited to choose a selection of images and give these to the application to arrange in terms of similarity and provide insight into the behaviour of the algorithm (an example of our signature exploration). However, it would be useful to start from the user layout of entities (images in this case) and modify the algorithm to reflect the user's concept of similarity. Hence signature modification using feedback data.

In this example, the user positions four objects (on the basis of perceived similarity). Each object also possesses a set of attributes and, by solving the linear equations (attribute set / x,y co-ordinate set), a mapping from the attribute values to the x,y co-ordinates is obtained. Members from a larger group from which the four objects are drawn can now be positioned to reflect the user's similarity measure. The layout can also be compared to those obtained by a variety of algorithms, so that the one that is the least different can be chosen.

6.1 Algorithm

Given multivariate data $X \in R^{n,m}, n > m$, where n is the number of entities and m the number of attributes and a subset $X' \in R^{m,m}$ of $m < n$ rows of X. Furthermore let us

assume that the user specified coordinates for the selected m entities, i.e. $Y \in R^{m,2}$ is given. Then solve the linear equation $X'|Y$, i.e. convert $X'|Y$ to $I|Y'$, where I is the identity matrix and $Y' \in R^{m,2}$. Then compute $C = XY' \in R^{n,2}$, which contains the x and y-coordinates for the n entities in its columns.

6.2 Example

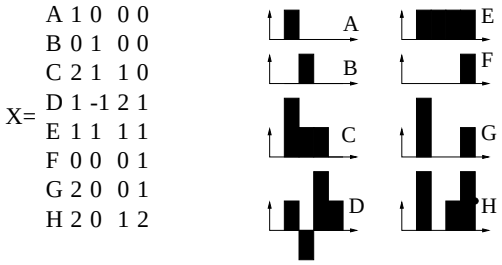


Fig. 7. Multivariate data for eight entities.

Consider the following example: There are eight entities $A - H$ and multivariate data $X \in R^{8,4}$ shown in Figure 7.

The user knows about the four entities $A - D$ and draws a layout of them as shown in Figure 8.1. According to the above algorithm we have $X'|Y$

A	1	0	0	0	3
B	0	1	0	0	2
C	2	1	1	2	0
D	1	-1	2	1	4	2

Deducing the third and fourth rows from the first and second we get $I|Y'$ as

A	1	0	0	0	0	3
B	0	1	0	0	0	2
C	0	0	1	0	2	-8
D	0	0	0	1	0	17

Computing the final coordinates XY', which are generalised from the subset $A - D$ and applied to all entities $A - H$, the layout is as shown in 8.2. Compare these user defined and generalised distances to the methods mentioned earlier. Considering only the entities $A - D$, which the user knows about, it is striking that they place A and C far away from each other (Fig. 8.1), whereas all others put them closer to each other, in particular correlation (Fig 8.6). Now consider the entities the user did not know about. The generalisation of the user distances places e.g. F close to A, B, which is done by the Minkowski distance, but not at all by correlation.

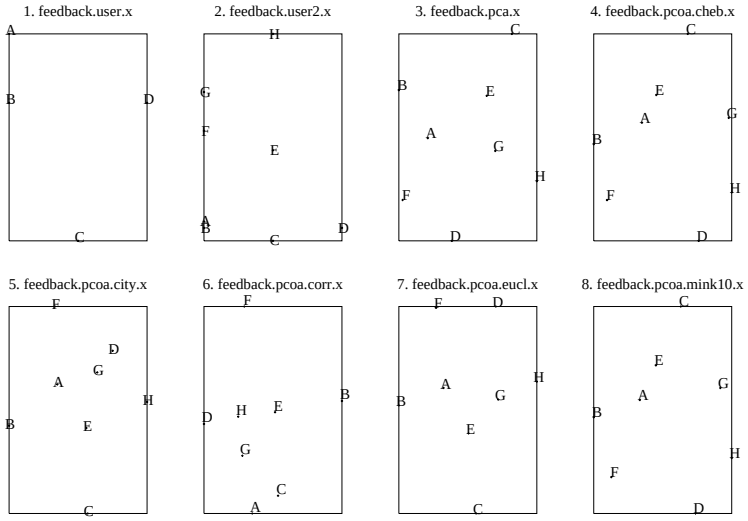


Fig. 8. 1. Four entities placed by a user. 2. Generalisation of these placements and application to all entities. 3-8. Scatterplot of the eight entities using PCA and PCoA with Chebychev, City or Manhattan (Minkowski with $\lambda = 1$), angular, Euclidean and Minkowski ($\lambda = 10$) distance.

7 Conclusions and future work

This paper has elaborated a concept, signature exploration, which reframes and extends existing work to assist users of visualization systems in their search for meaning in data, from the point of view of understanding visualizations as well as appropriate choice of visualization algorithm or application. An initial five categories of constructed data with which to explore are suggested. The results of an initial test of the concept and of one particular application employing user positioned data are described.

Although the principle is conceived as a general one - that it should (if validated) become a general visualization design requirement - the paper focuses on multivariate and proximity data layout using a particular tool. A weakness of this work is that the successful demonstration of the concept becomes linked with success in developing understanding of algorithms that involve, for instance, dimension reduction of complex data - an area known to be challenging. Whilst it is desirable to create tools that assist users in viewing complex data, it is important that we assess the concept of signature exploration in a more general context, that is, with a range of visualization applications, so that it will not fail because we fail to fully explain cluster shapes in dimension reduction scatter plots.

The initial test of the concept of signature exploration gave favourable results in terms of increasing understanding of visualization algorithms and thus of resulting patterns in the data. It strongly indicated the usefulness of developing an interface for entering data values and patterns of values to explore visualization algorithm behaviour, both

in the search for data sets that reveal an algorithm's behaviour and as a direct means of exploring the algorithm. The user specified layout example indicates the usefulness of capturing the user's domain knowledge for comparison and prediction and shows the possibility of the application's algorithm being modified accordingly.

It is intended that each of the constructed data types be explored in detail. Automatic algorithm choice is desirable, but the appropriate algorithm choice is expected to follow from the questions the user wants to answer of the dataset, at least in some cases, and there still remains the comprehension of the visualization itself, so that it is likely that an interface to a data construction engine of some kind would be valuable.

It is hoped that the framing of this scenario as signature exploration will focus energy upon these aspects of visualization system design - producing meaningful choice and increased comprehension.

7.1 Acknowledgements

We sincerely thank Professor Robert Spence, Emeritus Professor of Information Engineering, Department of Electrical and Electronic Engineering, Imperial College, London, for suggesting the names **signature exploration** and **landmark** during discussions on this topic.

This work is supported by the EPSRC and British Telecom (CASE studentship - award number 99803052). In particular we would like to thank Robert Ghanea-Hercock and Paul Coker of The Future Technologies Group, Adastral Park, Ipswich, for their assistance in provision of the calldata data set.

References

1. Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman. *Readings in Information Visualization: Using Vision To Think*. Morgan Kaufmann, 1999.
2. Chaomei Chen. *Information Visualisation and Virtual Environments*. Springer, 1999.
3. B. S. Everitt. *Graphical techniques for multivariate data*. Heinemann Educational Books, 1978.
4. GeneExplore. http://www.applied-maths.com/ge/ge.htm.
5. A. D. Gordon. *Classification*. Chapman and Hall/CRC, 1999.
6. J. C. Gower. Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika*, 53:325–38, 1966.
7. I. Herman, G. Melancon, and M.S. Marshall. Graph visualization and navigation in information visualisation: a survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24–43, 2000. http://www.cwi.nl/InfoVisu/Survey/StarGraphVisuInInfoVis.pdf.
8. J. Kruskal. The relationship between multidimensional scaling and clustering. *Classification and clustering*, 1977.
9. K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate analysis*. Academic Press, 1979.
10. Pearl Pu and Zorn Pecenovic. Dynamic overview techniques for image retrieval. In *Data Visualization 2000: Proceedings of the Joint Eurographics and IEEE TCVG Symposium on Visualization, Amsterdam, The Netherlands*. Springer, 2000.
11. George Robertson. Leveraging human capabilities in information perceptualization. Keynote speech at IEEE International Conference on Information Visualization IV2000 London July 19-21 2000, 2000.
12. J. W. Sammon. A nonlinear mapping for data analysis. *IEEE Transactions on Computers*, C(18):401–409, 1969.

13. Michael Schroeder. Using singular value decomposition to visualise relations within multi-agent systems. In *Proceedings of the third Conference on Autonomous Agents*, Seattle, USA, 1999. ACM Press.
14. Michael Schroeder, David Gilbert, Jacques van Helden, and Penny Noy. Approaches to visualisation in bioinformatics: from dendrograms to space explorer. *Accepted for Information Sciences: An International Journal*, 2001.
15. Michael Schroeder and Penny Noy. Multi-agent visualization based on multivariate data. In *Proceedings of Autonomous Agents2001*, Montreal, Canada, 2001. ACM press.
16. R. Spence and L. Tweedie. The attribute explorer: information synthesis via exploration. *Interacting with Computers*, 1998.
17. Robert Spence. *Information Visualization*. Addison-Wesley, 2001.
18. Colin Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann, 2000.
19. Andrew Webb. *Statistical pattern recognition*. Arnold, 1999.

Towards the Development of Environments for Designing Visualisation Support for Visual Data Mining

Simeon J. Simoff

Faculty of Information Technology
University of Technology, Sydney
NSW 2007, Australia
simeon@it.uts.edu.au

Abstract. The design of consistent information visualisation models is a key component in the development of visual data mining methods. However, it is a challenging activity to find the methods, techniques and corresponding tools that are suitable for specific visual mining task, or a particular type of data. The comparison of visualisation techniques across different designs is not a trivial problem. This paper discusses the issues connected with the development of consistent approach to formal development, evaluation and comparison of visualisation methods. Proposed formal approach is illustrated with examples of development of a visualisation model for data from the area of team collaboration in virtual environments and evaluation of visualisation schemes for the results of text analysis. The papers concludes with the discussion of the limitations of the proposed approach and future directions of the research and development of proposed approach.

Keywords: data visualisation, visual data mining, metaphor representation, virtual environments, formal methods, affective computing

1. Handcrafting and creativity in the design of visualisation techniques

Visual data mining[1] (Michalski et al., 1999) is an approach to explorative data analysis and knowledge discovery that is built on the extensive use of visual computing (Gross, 1994, Nielson et al., 1997). The basic assumption is that large and normally incomprehensible amounts of data can be reduced to a form that can be understand and interpreted by a human through the use of visualisation techniques based on a particular metaphor or a combination of several metaphors (preferably, but not necessarily preserving the consistency of their combination). The popularity of digital terrain models, based on the geographical framework (Hetzler et al., 1998a, Hetzler et al., 1998b) and CAD-based architectural models of cities, mapping the metaphor of urban design to information visualisation, has demonstrated that multi-dimensional visualisation can provide a superior means for exploring large data-sets, communicating model results to others and sharing the model (Brown, 1998). Some recent developments are extending visual data mining with algorithmic animation techniques (Meisalo et al., 1998), multimedia support (Noirhomme-Fraiture, 2000) and incorporation of virtual reality immersive representations, aiming at involving wider range of human "input" channels in the mining and discovery processes.

The design of visualisation models for visual data mining, in broad sense, is the formal definition of the rules for translation of data into graphics. Generally, the visualisation of large volumes of abstract data, known as 'information visualisation' is closely related but sometimes contrasted, to scientific visualisation, which is concerned with the visualisation of

[1] Visual data mining spans from 2D visualisations into the use of a virtual reality systems, for example, see http://www.cs.auc.dk/3DVDM/about.html.

(numerical) data used to form concrete representations (Nielson et al., 1997). The frequently used expression "the art of visualisation" appropriately describes the state of research and development in that field. *Currently, it is a challenging activity for information designers to find out the methods, techniques and corresponding tools available to visualise a particular type of information*.

The *comparison of visualisation techniques across different designs* is not a trivial problem (Chen, 1999). Partially, current situation is explained by the lack of generic criteria to access the value of visualisation models. This is a research challenge, since most people develop their own criteria for what makes a good visual representation. The design of visualisation schemata is an area, dominated by individual points of views, which has resulted in a considerable variety of ad hoc techniques (Chen, 1999a). A recent example is the visualisation of association rules proposed in (Hofmann et al., 2000), where rule predicates are associated with shaded rectangular areas in attempt to convey visually some quantitative, qualitative and even semantic information about the rule patterns.

Another integral part of information visualisation is *the evaluation of how well humans understand visualised information* as part of their cognitive tasks and intellectual activities, the efficiency of information compression and level of cognitive overload. (Crapo et al., 2000) has investigated some aspects of developing visualisation schemata from cognitive point of view. Different cognitive styles in human activities, especially in stimulating insights and creativity can play key role in the success of one or another method. For example, the circle of linked histograms visualisation mechanism of Daisy[2] offers a simple visualisation of relationships of different groupings of data records, which has the potential variety of emergent geometries that can lead to some insights about more complex relationships in the data.

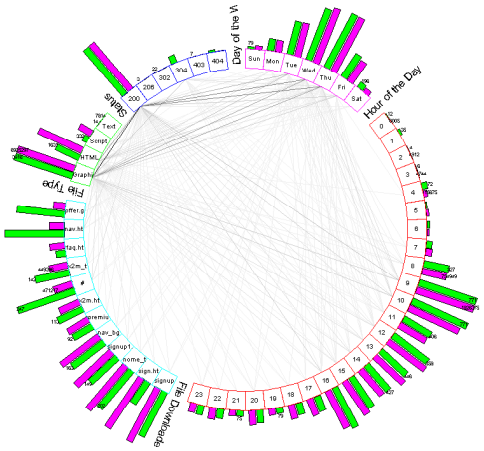


Figure 1. The visualisation schemata of Daisy Analysis visual data mining tool

Consequently, with the increasing attention towards development of interactive visual data mining methods the development of more systematic approach towards the design of visualisation techniques is getting on the "to do" list of the data mining research.

1.1 The special needs of collaborative and immersive data mining

The success of a visual data mining method depends on the development of an adequate computational model of selected metaphor. This is especially important in the context of communicating and sharing of discovered information, and in the context of the emerging methods of computer-mediated collaborative data mining. This new visual data mining methodology is based on the assumption that individuals usually may respond with

[2] http://www.daisy2000.com/

different interpretations of the same information visualisation (Snowdon et al., 1995). A central issue is the communicative role of abstract information visualisation components in collaborative environments for visual data mining. In fact, "miners" can become part of the visualisation. For example, in virtual worlds this can happen through their avatars[3]. In a virtual world, a collaborative perspective is inevitable, thus a shared understanding of the underlying mapping between the semantic space and the visualisation scheme becomes a necessary condition in the development of these environments. The results of CMCDM are heavily influenced by the adequate formalisation of the metaphors that are used to construct the virtual environment, i.e. the visualisation schemata can influence the behavior of collaborators, the way they interact with each other, the way that they reflect on the changes and manipulations of visualisations, and, consequently, their creativity in the discovery process.

1.2 Challenges towards the development of environments for design of visualisations

Currently, it is a challenging task for designers of visual data mining environments to find the strategies, methods and corresponding tools to visualise a particular type of information. Mapping characteristics of data into a visual representation in virtual worlds is one promising way to make the discovery of encoded relations in this data possible. The model of semantically organised place for visual data exploration can be useful for the development of computer support for visual information querying and retrieval (Del Bimbo, 1999, Gong, 1998) in collaborative information filtering. The development of a representational and computational model of selected metaphor(s) for data visualisation will assist the design of virtual environments, dedicated to visual data exploration. This paper argues for the development of more formal approach towards the design and evaluation of visualisation techniques.

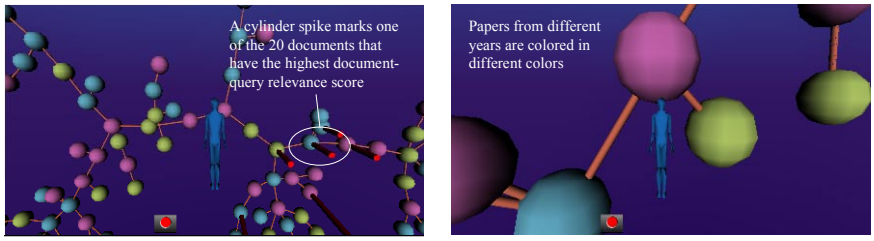
2. Background to a formal approach for the design of visualisation techniques

The formal approach presented in this paper is based on the concept of *semantic visualisation* defined as a visualisation method, which establishes and preserves the semantic link between form and function in the context of the visualisation metaphor. Establishing a connection between form and functionality is not a trivial part of the visualisation design process. In a similar way, selecting the appropriate form for representing data graphically, whether the data consists of numbers or text, is not a straightforward procedure as numbers and text descriptions do not have a natural visual representation. On the other hand, how data are represented visually has a powerful effect on how the structure and hidden semantics in the data is perceived and understood. An example of a virtual world, which attempts to visualise an abstract semantic space, is shown in Figure 2. The visualisation of the semantic space of the domain of human-computer interaction is automatically constructed from a collection of papers from three consecutive ACM CHI conference proceedings (Chen, 1999). The overall landscape is designed according to the theory of cognitive maps. In such a world, there is a variety of possibilities for data exploration. For example, topic areas of papers are represented by coloured spheres. If a cluster of spheres includes every colour but one, this suggests that the particular topic area, represented by the missing coloured sphere, has not been addressed by the papers during that year. However, without the background knowledge of the semantics of coloured spheres, selected information visualisation scheme does not provide cues for understanding and interpreting the environment landscape. It is not clear how the metaphor of a "landscape" has been formalised and represented, what are the elements of the landscape. Associatively, this visualisation is closer with the visualisation of molecular

[3] 3D representations of people in virtual worlds. Avatar is an ancient Sanskrit term meaning ‘a god’s embodiment on the Earth’ (Damer, 1998).

structures. Consequently, will an expert in molecular chemistry be more efficient in discovery specific relations between the visualised entities, based on her/his knowledge of possible associations between particular link configurations and molecular properties.

Semantic visualisation is considered in the context of two derivatives of visualisation - *visibilisation* and *visistraction* (Choras and Steinmann, 1995). Visibilisation is visualisation focusing on the presentation and interpretation which complies with rigorous mapping from physical reality. By contrast, visistraction is the visualisation of abstract concepts and phenomena, which do not have a direct physical interpretation or analogy. Visibilisation has the potential to bring key insights, by emphasising aspects that were unseen before. The dynamic visualisation of the heat transfer during the design of the heat-dissipating tiles cover of the underside of the space-shuttle is an early example of the application of visibilisation (Gore, 1981). Visistracton can give a graphic depiction of intuition regarding objects and relationships. The 4D simulation of data flow is an example of visistraction, which provides insights impossible without it. In a case-base reasoning system, visistraction techniques can be used to trace the change of relationships between different concepts with the addition of new cases.



a. b.

Figure 2. An example of virtual world visualising abstract semantic structure

Both kinds of semantic visualisation play important role in visual data mining. However, semantic visualisation remains a hand-crafted methodology, where each case is considered separately. This paper presents a consistent approach to semantic visualisation based on a cognitive model of metaphors, metaphor formalisation and evaluation. We illustrate the application of this approach with examples from visistraction of communication and collaboration data. Further, the paper is organised as follows. Section 3 presents the Form-Semantics-Function framework for construction and evaluation of visualisation techniques, Section 4 presents an example of the application of the framework towards the construction of a visualisation technique for identifying patterns of team collaboration, Section 5 presents an example of the application of the framework for evaluation and comparison of two visualisation models. The paper concludes with the issues for building visualization models that support collaborative data mining.

3. Form-Semantics-Function: a formal approach towards constructing and evaluating visualisation techniques.

The Form-Semantic-Function (FSF) approach includes the following steps: *metaphor analysis*; *metaphor formalisation*; and *metaphor evaluation*. Through the use of metaphor, people express the concepts in one domain in terms of another domain (Lakoff and Johnson, 1980, Lakoff, 1993). The closest analogy is VIRGILIO (L'Abbate and Hemmje, 1998), where the authors proposed a formal approach for constructing metaphors for visual information retrieval. The FSF framework develops further the formal approach towards constructing and evaluating visualisation techniques, approaching the metaphor in an innovative way.

3.1 Metaphor analysis

During metaphor analysis, the content of the metaphor is established. In the use of metaphor in cognitive linguistics, the terms *source* and *target*[4] refer to the conceptual spaces connected by the metaphor. The target is the conceptual space that is being described, and the source is the space that is being used to describe the target. In this mapping the structure of the source domain is projected onto the target domain in a way that is consistent with inherent target domain structure (Lakoff, 1993, Turner, 1994). In the context of semantic visualisation, the consistent use of metaphor is expected to bring an understanding of a relatively abstract and unstructured domain in terms of more concrete and structured visual elements through the visualisation schemata.

An extension of the source-target mapping, proposed by (Turner and Fauconnier, 1995) includes the notion of generic space and blend space. Generic space contains the skeletal structure that applies to both source and target spaces. The blend space often includes structure not projected to it from either space, namely emergent structure on its own. The ideas and inspirations developed in the blend space can lead to modification of the initial input spaces and change the knowledge about those spaces, i.e. to change and evolve the metaphor. The process is called conceptual blending - it is the essence in the development of semantic visualisation techniques.

In presented approach, the form-semantics-function categorisation of the objects being visualised, is combined with the (Turner and Fauconnier, 1995) model. The form of an object can express the semantics of that object, that is, the form can communicate implicit meaning understood through our experiences with that form. From the form in the source space we can connect to a function in the target space via the semantics of the form. The model of the metaphor analysis is shown in Figure 3.

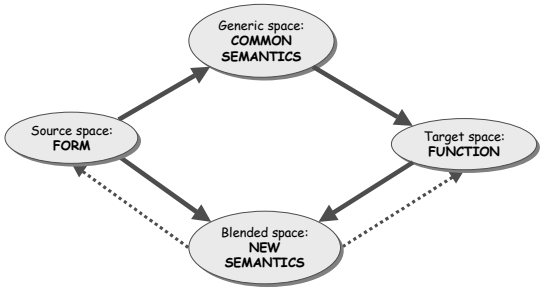


Figure 3. A model for metaphor analysis for constructing semantic visualisation schemata

The term *visual form* refers to the geometry, color, texture, brightness, contrast and other visual attributes that characterise and influence the visual perception of an object. Thus, the source space is the space of 2D and 3D shapes and the attributes of their visual representation. "Functions" (the generalisations of patterns, discovered in data) are described using concepts from the subject domain. Therefore the target space includes such concepts associated with the domain functions. This space is constructed from the domain vocabulary. The actual transfer of semantics has two components - the common semantics, which is carried by notions that are valid in both domains and what is referred as new semantics - the blend, which establishes the unique characteristics revealed by the correspondence between

[4] In the research literature the target is variously referred to as the primary system or the topic, and the source is often called the secondary system or the vehicle.

the form metaphor and functional characteristics of that form. The schema illustrates how metaphorical inferences produce parallel knowledge structures.

3.2 Metaphor formalisation

The common perception of the word "formalisation" is connected with the derivation of some formulas and equations that describe the phenomenon in analytical form. In this case, formalisation is used to describe a series of steps that ensure the correctness of the development of the representation of the metaphor. Metaphor formalisation in the design of semantic visualisation schemes includes the following basic steps:

- *Identification of the source and target spaces of the metaphor* - the class of forms and the class of features or functions that these forms will represent;
- *Conceptual decomposition of the source and target spaces* produces the set of concepts that describe both sides of the metaphor mapping. As a rule, metaphorical mappings do not occur isolated from one another. They are sometimes organized in hierarchical structures, in which `lower' mappings in the hierarchy inherit the structures of the `higher' mappings. In other words, this means that visualisation schemes, which use metaphor are expected to preserve the hierarchical structures of the data that they display. In visistraction, these are the geometric characteristics of the forms from the source space, and other form attributes like colours, line thickness, shading, etc. and the set of functions and features in the target space associated with these attributes and variations;
- *Identifying the dimensions of the metaphor* along which the metaphor operates. These dimensions constitute the common semantics. In visistraction this can be for instance key properties of the form, like symmetry and balance with respect to the center of gravity, that transfer semantics to the corresponding functional elements in the target domain;
- *Establishing semantic links, relations and transformations* between the concepts in both spaces, creating a resemblance between the forms in the source domain and the functions in the target domain.

3.3 Metaphor evaluation

In spite of the large number of papers describing the use of the metaphor in the design of computer interfaces and virtual environments, there is a lack of formal evaluation methods. In the FSF framework metaphor evaluation is tailored following the (Anderson et al., 1994) model, illustrated in Figure 4.

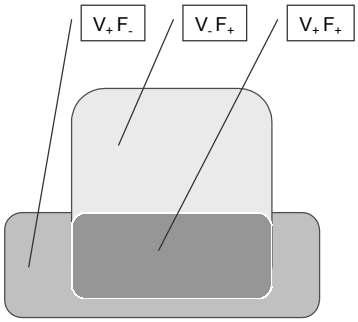


Figure 4. Model for evaluating metaphor mapping (based on (Anderson et al., 1994))

The "V" and "F" are labels for visualisation and function features, respectively. The "VF" label with indices denotes numbers of features, namely:

- V_+F_+ - function features that are mapped to the visualisation schema;

- V_F_+ - function features that are not supported by the visualisation schema;
- $V_+F\_$ - features in the visualisation schema, not mapped to the functional features.

The ratio $\frac{V\_F_+}{V_+F_+}$ provides an estimate of the quality of the metaphor used for the visualisation - the smaller the better.

4. Application of the Form-Semantics-Function approach

The elements of the Form-Semantics-Function approach are illustrated in the examples of constructing visualisation schema for visual data mining and simple evaluation of two text data visualisation techniques. The first example illustrates the metaphor analysis and formalisation stages for the creation of visualisation form and mapping it to the functional features. In the second example the evaluation of the two visualisation schemata is based on the same set of functional features.

4.1 Constructing visualisation schema for visual data mining

The goal of the development of this visualisation schema is to produce a simple but appealing visual representation of asynchronous collaboration data so that researchers will be able to identify patterns in team collaboration. Asynchronous communication is an intrinsic part of computer-mediated teamwork. Among the various models and tools supporting this communication mode (Maher et al., 2000), perhaps the most popular in teamwork are bulletin (discussion) boards. These boards support multi-thread discussion, automatically archiving communication content. One way to identify patterns in team collaboration is via content analysis of team communications. However, it is difficult to automate such analysis, therefore, especially in large scale projects, monitoring and analysis of collaboration data can become a cumbersome task.

In the research in virtual design studios (Maher et al., 1997, Maher et al., 2000) there have been identified two extremes (labeled as "Problem comprehension" and "Problem division") in team collaboration, illustrated in Figure 5. In "Problem comprehension" collaborative mode the resultant project output - a product, solution to a problem, etc., is a product of a continued attempt to construct and maintain a shared conception and understanding of the problem. In other words each of the participants is developing own view over the whole problem and the shared conception is established during the collaborative process via intensive information exchange.

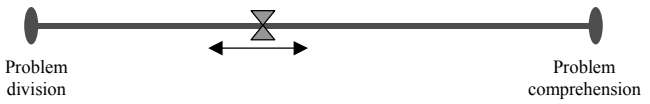


Figure 5. Two extremes in team collaboration

In "Problem division" mode the problem is divided among the participants in a way where each person is responsible for a particular portion of the investigation of the problem. Thus, it does not necessarily require the creation of a single shared conception and understanding of the problem. The two modes of collaboration are two extreme cases. In general, the real case depends on the complexity of the problem.

A key assumption in mining and analysis of collaboration data is that this two extreme styles should be some how reflected in the communication of the teams. Thus, different patterns in team communication on the bulletin board will reflect different collaboration modes. Figure 6 illustrates a fragment of a team bulletin board.

Course: Computer Based Design | **Bulletin Board:** Team 2 | **Venue:** Virtual Design Studio	
4 Lighting etc - **Derek** *08:46:10 10/16/97* (1)	(M_{14})
• Re: Lighting etc - **Sophie Collins** *10:43:18 10/17/97* (0)	(M_{24})
3 Seating - **Derek Raithby** *15:18:57 10/14/97* (2)	(M_{13})
• Re: Seating - **marky** *17:22:56 10/14/97* (1)	(M_{23})
• Re: Seating - **Sophie Collins** *09:03:27 10/15/97* (0)	(M_{33})
2 Product Research. - **Derek Raithby** *14:37:43 10/14/97* (1)	(M_{12})
• Re: Product Research. - **mark** *17:20:16 10/14/97* (0)	(M_{22})
1 Another idea - **Sophie Collins** *14:24:18 10/14/97* (1)	(M_{11})
• Re: another idea - **Derek Raithby** *14:40:00 10/14/97* (0)	(M_{21})

Figure 6. Bulletin board fragment with task-related messages, presented as indention graph

The messages on the board are grouped in threads. (Berthold et al., 1997, Berthold et al., 1998) propose a threefold split of the thread structure of e-mail messages in discussion archives in order to explore the interactive threads. It included (i) reference-depth: how many references were found in a sequence before this message; (ii) reference-width: how many references were found, which referred to this message; and (iii) reference-height: how many references were found in a sequence after this message. In addition to the threefold split, (Sudweeks and Simoff, 2000) included the time variable explicitly. Figure 7 shows the formal representation of the bulletin board fragment in Figure 6.

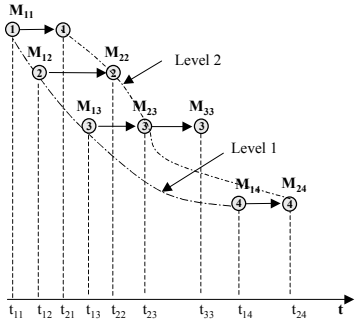


Figure 7. Formal representation of the thread structure in the fragment presented in Figure 6.

4.1.1 Metaphor analysis

Figure 8 shows the Form-Semantics-Function mapping at the metaphor formalisation stage as a particular case of the (Turner and Fauconnier, 1995) model applied to the visualisation of communication utterances data. The source space in this case is the space of 2D geometric shapes, rectangles in particular. The target space includes the concepts associated with the functions that are found in the analysis of a collaborative design session.

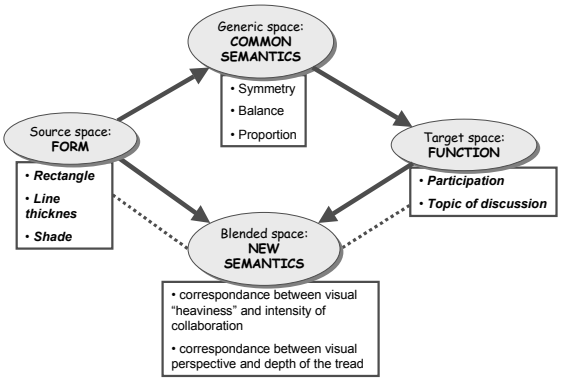


Figure 8. The Form-Semantics-Function mapping for the source space of nested rectangles and the target space of bulletin board discussion threads.

4.1.2 Metaphor formalisation

Below is a brief illustration of the metaphor formalisation in this example.

- *Identification of the source and target spaces of the metaphor* - rectangles are the forms that will be used and the class of features or functions that these forms will represent are the messages on a bulletin board;
- *Conceptual decomposition of the source and target spaces* leads to the notion of nested rectangles, whose centers of gravity coincide, with possible variation of the thickness of their contour lines and the background color. Each rectangle corresponds to a message within a thread. Rectangle that corresponds to a message at a level (n + 1) is placed within the rectangle that corresponds to a message at level n. Messages at the same level are indicated by a one step increase of the thickness of the contour line of the corresponding rectangle. Thus, a group of nested rectangles can represent several threads in a bulletin board discussion;
- *Identifying the dimensions of the metaphor* - visual balance and the "depth" or "perspective" of the nested rectangles are the dimension of the metaphor, transferring via the visual appearance the semantics of different communication styles;
- *Establishing semantic links, relations and transformations* - this is connected with the identification of typical form configurations that correspond to typical patterns of collaboration. For example, Figure 9 illustrates two different fragments A and B (each of one thread). Figure 10 illustrates the visualisation of this fragments according to the developed visualisation schema.

The visualisation schema has been used extensively in communication analysis. Figure 11 illustrates communication patterns corresponding to different collaboration styles. An additional content analysis of communication confirmed the correct identification of collaboration patterns.

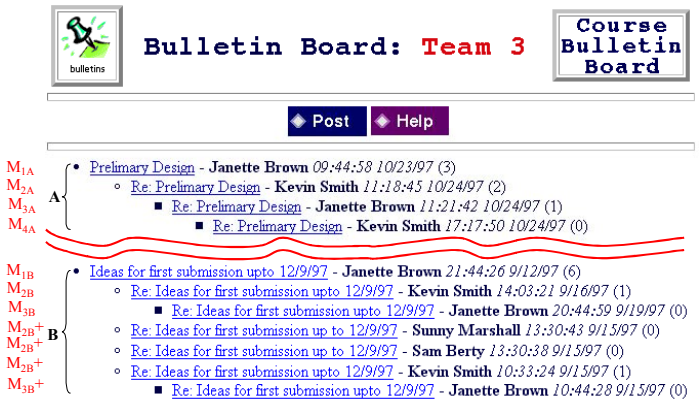


Figure 9. Bulletin board fragment with task-related messages, presented as indention graph

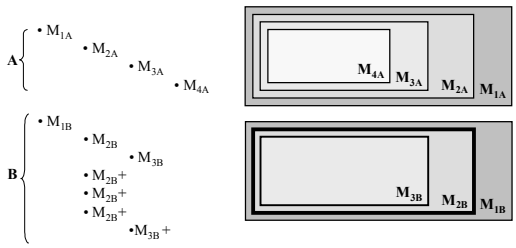
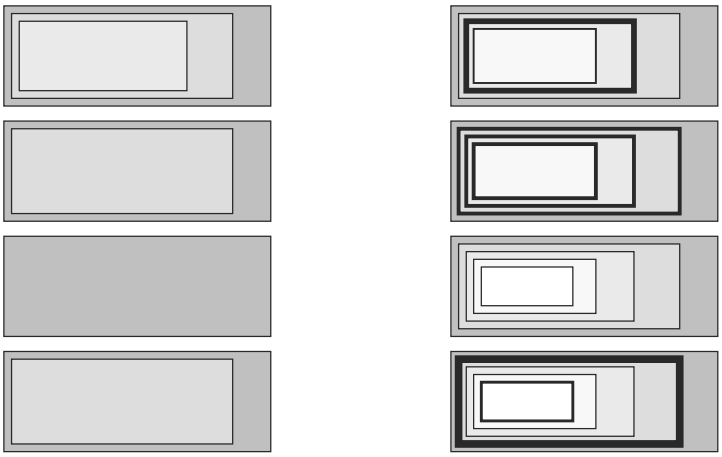


Figure 10. Visualisation of fragments A and B in Figure 9.



a. Communication patterns, typical for "problem division" collaboration

b. Communication patterns, typical for "problem comprehension" collaboration

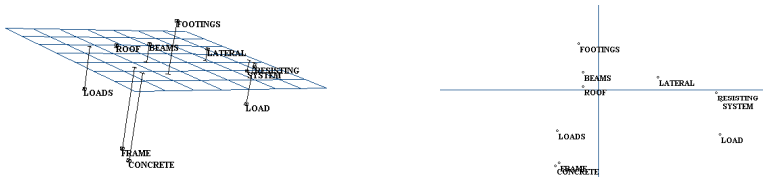
Figure 11. Communication patterns, corresponding to different collaboration styles.

4.2 Evaluation and comparison of two visualisation schemata

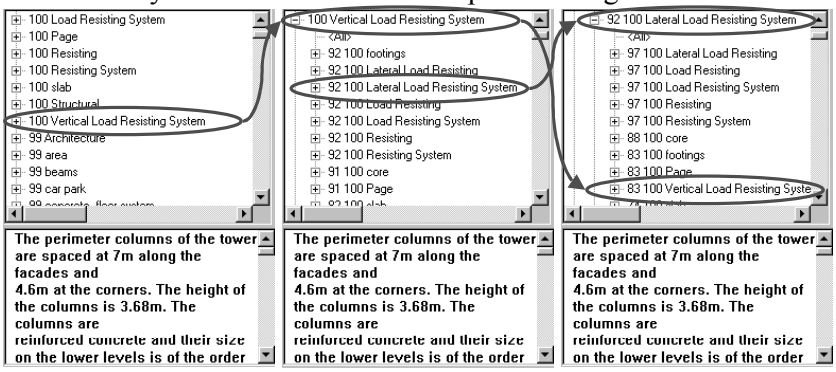
We illustrate the idea by evaluating examples of semantic visualisation of textual data and objects in virtual environments. The role of visistraction in concept relationship analyis is to assist the discovery of the relationship between concepts, as reflected in the available text data. The analysis uses word frequencies, their co-occurence and other statistics, and cluster analysis procedures. We investigate two visual metaphors - "Euclidian space" and "Tree", which provide a mapping from the numerical statistics and cluster analysis data into the target space of concepts and relations between them. The visualisation features for both metaphors and the function features of the target space are shown in Table 1. Examples of the two visualisation metaphors are shown in Figure 12.

Table 1. Visualisation and function features

Visualisation features of Euclidian space metaphor	**Visualisation features of tree metaphor**	**Function features**
- point	- nodes	- simple/complex concept
- alphanumeric single-word point labels	- alphanumeric multi-word node labels	- subject key word
- axes	- signs "+" and "-"	- hierarchical relationship
- plane	- branches	- context link
- color	- numeric labels for branches	- link strength
- line segment		- synonymy
		- hyponymy



a. 10 most frequent words in the description of lateral load resisiting system in one of the wide-span building cases[5]



b. Part of a semantic net[6] for the term "load resisting system".

Figure 12. Visualisation of the results of cluster analysis performed over a text data set[7]

4.2.1 Metaphor evaluation

The first scheme[8] maps the source domain of Euclidean space (coordinates of points in 2D/3D space) to the target domain of word statistics. The blending semantics is that the degree, to which the terms are related to each other, can be perceived visually from the distance between the corresponding data points - the closer the points the tighter is the relationship between the words. The second scheme[9] maps the source domain of the topology of linked nodes to the same target domain of words statistics. This mapping generates one of the possible visualisations of semantic networks. This visualisation includes nodes with single- and multiple-word labels, numeric values of each link between terms and the weight of the term among the other terms in the tree. The results of the comparison between the two metaphors are presented in Table 2 and

Table 3.

The Euclidean space metaphor has a poor performance for visualisation of concept relationships. What is the meaning of such closeness - it is difficult to make a steady judgement about what the relation is and whether we deal with simple (one word) or complex (more than one word) terms. The distance to the surface, proportional to the frequency of the words, can convey the message that a word is a key word. However, there is no feature in the visualisation, which shows context links between words, the strength of this links and other relations between words.

[5] This visualisation is used in TerraVision, which is part of the CATPAC system for text analysis by Provalis Research Co.
[6] This is the visualisation of semantic networks in TextAnalyst by Megaputer Intelligence, Inc.
[7] The source text comes from the SAM (Structure and Materials) case library available at http://www.arch.usyd.edu.au/kcdc/caut/
[8] The schema is used in the CATPAC qualitative analysis package by Terra Research Inc.
[9] The schema is used in TextAnalyst by Megaputer Intelligence (see www.megaputer.com).

Table 2. Visualisation support for function features in Euclidean space and tree metaphors

Function features	**Support by the Euclidean space metaphor**	**Support by the Tree metaphor**
Simple/complex concept	-	+
Subject key word	+	+
Hierarchical relationship	-	+
Context link	-	+
Link strength	-	+
Synonymy	-	-
Hyponymy	-	-

Table 3. Comparison of in Euclidean space and tree metaphors

	Euclidean space metaphor	**Tree metaphor**
V_+F_+	1	5
V_-F_+	6	2
$\frac{V_-F_+}{V_+F_+}$	6	0.4

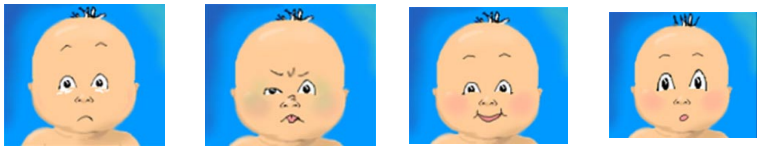
5. Limitations of proposed approach and future directions

The Form-Semantics-Function framework resented in this work is an attempt to develop a formal approach towards the use of metaphors in constructing consistent visualisation schemes. In its current development, it has at least the following limitations:

- to preserve consistency for each change in the visualisation schemata the analysis/formalisation pass has to be conducted again;
- the evaluation part of the framework does not include the analysis of the cognitive overload from the point of information design.

Currently the FSF framework is further developed in the following directions:

- *supporting visualisation schemes based on affective computing models*, which aim at communicating specific emotions for specific patterns in the data. The idea is illustrated by using the emotional expression of a human face to represent four data patterns in Figure 2 (a-d). It is based on the assumption that the observer can perceive any change in the emotional expression of the face, i.e. in one or more of its features, when representing a different pattern in data, accurately.



a. b. c. d.

Figure 13. Emotion-based agents acting as visualisation support in multi-agent data mining environment (the actual faces are adapted from (Ventura, 2000))

- *supporting collaborative visual data mining in virtual worlds*. The different perceptions of a visualisation model in a such environment may increase the gap between individuals as they interact with it in a data exploration session. However, individual differences may lead to a potential variety of discovered patterns and insights in the visualised information across participants.

In this context the future research within the FSF framework will be focused on exploring:

- whether people attach special meanings to abstract visualisation objects;
- what are the design criteria towards visualisation objects, engaged in visual data exploration, that people can effectively construct and communicate knowledge in visual data mining environments;
- what are the necessary cues that should be supported in semantically organised virtual environments.

Acknowledgements

This research has been supported by the University of Technology Sydney and an ATN Research Grant.

References

Anderson, B., Smyth, M., Knott, R. P., Bergan, M., Bergan, J. and Alty, J. L. (1994). Minimising conceptual baggage: Making choices about metaphor. In Cocton, G., Draper, S. and Weir, G. (eds), *People and computers IX* Cambridge University Press, Cambridge, pp. 179-194.

Berthold, M. R., Sudweeks, F., Newton, S. and Coyne, R. (1997). Clustering on the Net: Applying an autoassociative neural network to computer-mediated discussions., *Journal of Computer Mediated Communication,* **2**(4), . Available: http://www.ascusc.org/jcmc/vol2/issue4/bert-hold.html.

Berthold, M. R., Sudweeks, F., Newton, S. and Coyne, R. (1998). It makes sense: Using an autoassociative neural network to explore typicality in computer mediated discussions. In Sudweeks, F., McLaughlin, M. and Rafaeli, S. (eds), *Network and Netplay: Virtual Groups on the Internet* AAAI/MIT Press, Menlo Park, CA, pp. 191-220. Available: http://www.ascusc.org/jcmc/vol2/issue4/bert-hold.html.

Brown, I. M. (1998). A 3D user interface for visualisation of Web-based data-sets, *Proceedings of the 6th international symposium on Advances in geographic information systems*, , , pp. 100-105.

Chen, C. (1999). *Information Visualisation and Virtual Environments,* Springer-Verlag, London.

Choras, D. N. and Steinmann, H. (1995). *Virtual reality: Practical applications in business and industry,* Prentice Hall, Upper Saddle River, New Jersey.

Crapo, A. W., Waisel, L. B., Wallace, W. A. and Willemain, T. R. (2000). Visualization and the process of modeling: A cognitive-theoretic approach, *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD-2000*, The Association for Computing Machinery, Boston, MA, USA, pp. 218-226.

Damer, B. (1998). *Avatars,* Peachpit Press, an imprint of Addison Wesley Longman.

Del Bimbo, A. (1999). *Visual information retrieval,* Morgan Kaufmann Publishers, Inc., San Francisco, CA.

Gong, Y. (1998). *Intelligent image databases: Towards advanced image retrieval,* Kluwer Academic Publishers, Boston, MA.

Gore, R. (1981). When the space shuttle finally flies, *National Geographic,* **159**, 317-347.

Gross, M. (1994). *Visual computing: The integration of computer graphics, visual perception and imaging,* Springer-Verlag, Heidelberg.

Hetzler, B., Harris, W. M., Havre, S. and Whitney, P. (1998a). Visualising the full spectrum of document relationships, *Proceedings of the Fifth International ISKO Conference:*

Structures and Relations in the Knowledge Organisation, , Lille, France. Available: http://multimedia.pnl.gov:2080/infoviz/isko.pdf.

Hetzler, B., Whitney, P., Martucci, L. and Thomas, J. (1998b). Multi-faceted insight through interoperable visual information analysis paradigms, *Proceedings of IEEE Information Visualization 98*, IEEE, . Available: http://multimedia.pnl.gov:2080/infoviz/ieee98.pdf.

Hofmann, H., Siebes, A. P. J. M. and Wilhelm, A. F. X. (2000). Visualizing association rules with interactive mosaic plots, *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD-2000*, The Association for Computing Machinery, Boston, MA, USA, pp. 227-235.

L'Abbate, M. and Hemmje, M. (1998). VIRGILIO - The metaphor definition tool. (ed.)^(eds), GMD Report, Darmstadt.

Lakoff, G. (1993). The contemorary theory of metaphor. In Ortony, A. (ed.) *Metaphor and thought* Cambridge University Press, Cambridge, pp. 202-251.

Lakoff, G. and Johnson, M. (1980). *Metaphors we live by,* University of Chicago Press, Chicago, IL.

Maher, M. L., J, S. S. and A, C. (1997). Potentials and limitations of Virtual Design Studio, *Interactive Construction On-line,* **January, a1**, . Available: http://www.inconstruction.com.

Maher, M. L., Simoff, S. J. and Cicognani, A. (2000). *Understanding virtual design studios,* Springer-Verlag, London, UK.

Meisalo, V., Sutinen, E., Tarhio, J. and Teraumlsvirta, T. (1998). Combining algorithmic and creative problem solving on the Web, *Proceedings Teleteaching '98/IFIP World Computer Congress 1998*, Austrian Computer Society, , pp. 715-724.

Michalski, R. S., Bratko, I. and Kubat, M. (Eds) (1999) *Machine learning and data mining,* John Wiley and Sons, Chichester, England.

Nielson, G. M., Hagen, H. and Muller, H. (1997). *Scientific visualization : overviews, methodologies, and techniques,* IEEE Computer Society,, Los Alamitos, CA.

Noirhomme-Fraiture, M. (2000). Multimedia Support for Complex Multidimensional Data Mining, *Proceedings of the First International Workshop on Multimedia Data Mining (MDM/KDD'2000), in conjunction with Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD 2000*, ACM Press, Boston, MA, pp. 54-59.

Snowdon, D. N., Greenhalgh, C. M. and Benford, S. D. (1995). What You See is Not What I See: Subjectivity in virtual environments, *Proceedings Framework for Immersive Virtual Environments - FIVE'95*, , London, UK.

Sudweeks, F. and Simoff, S. (2000). Complementary explorative data analysis: The reconciliation of quantitative and qualitative principles. In Jones, S. (ed.) *Doing Internet Research*, Vol. 29-55 Sage Publications, .

Turner, M. (1994). Design for a theory of meaning. In Overton, W. and Palermo, D. (eds), *The Nature and Ontogenesis of Meaning* Lawrence Erlbaum Associates, , pp. 91-107.

Turner, M. and Fauconnier, G. (1995). Conceptual integration and formal expression, *Journal of Metaphor and Symbolic Activity,* **10**(3), 183-204.

Ventura, R.M. (2000). *Emotion-Based Agents*, MSc Thesis, Lisbon Polytechnics, Lisbon.