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: $d_{cheb}(x,y) = max_{1 \leq i \leq n}\{|x_i - y_i|\}$

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.

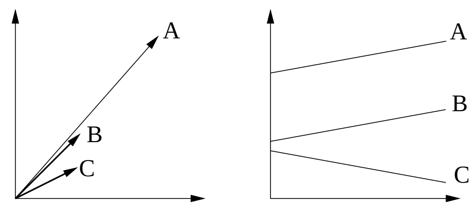


Fig. 2. The choice of distance influences data analysis. On the left are three vectors A, B, C and on the right three time series A, B, C. In both cases, B is close (far) to C (A) for Euclidean distance, but B is close (far) to A (C) for angular distance.

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

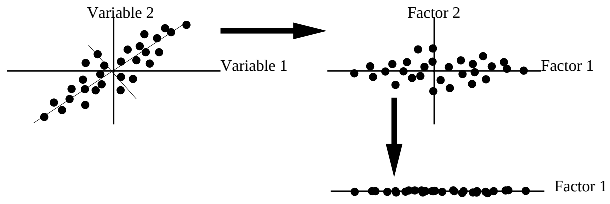


Fig. 3. Principal component analysis.

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.

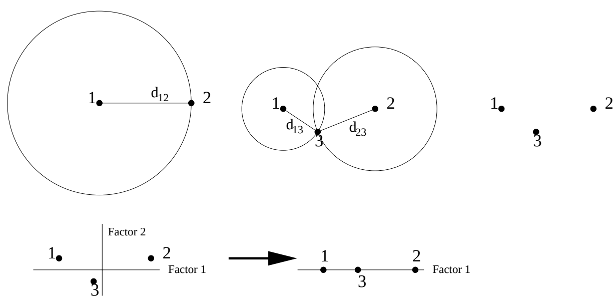


Fig. 4. Multi-dimensional scaling with principal co-ordinate analysis (PCoA).

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

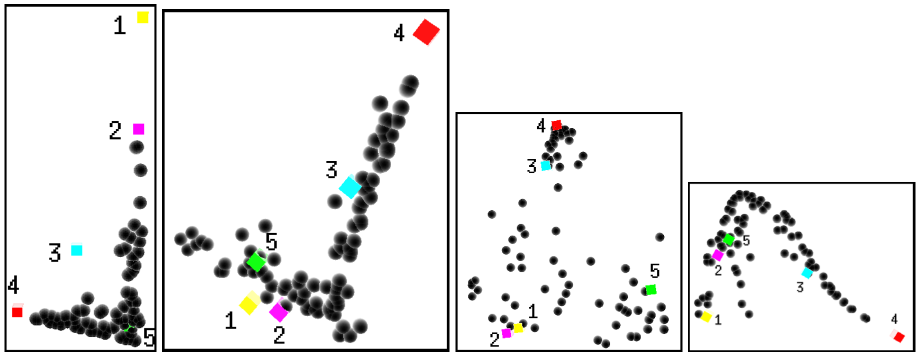


Fig. 5. 90,000 calls made by 100 customers. Caller profiles by destination of calls. These are screenshots of 3D VRML worlds. From top left to bottom right: (1) direct visualization by applying PCA, (2-4) indirect visualization by calculation of a distance followed by matrix transformation (PCoA); Distances 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

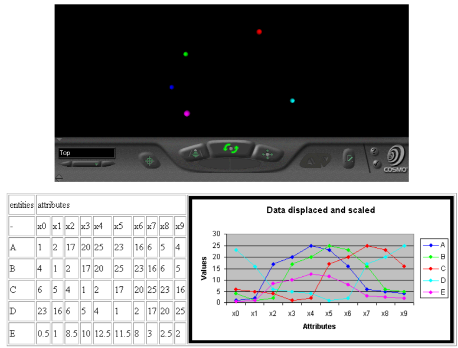
Do our visualizations actually work? This question was asked at a recent conference [11] and statistics from conference papers given that showed less than 10% had carried out evaluation. Informal testing in the early stages was indicated to be beneficial and our feasibility test [2] is of this nature. 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.



entities	attributes									
-	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9
A	1	2	17	20	25	23	16	6	5	4
B	4	1	2	17	20	25	23	16	6	5
C	6	5	4	1	2	17	20	25	23	16
D	23	16	6	5	4	1	2	17	20	25
E	0.5	1	8.5	10	12.5	11.5	8	3	2.5	2

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

$$X = \begin{array}{lrrrr} A & 1 & 0 & 0 & 0 \\ B & 0 & 1 & 0 & 0 \\ C & 2 & 1 & 1 & 0 \\ D & 1 & -1 & 2 & 1 \\ E & 1 & 1 & 1 & 1 \\ F & 0 & 0 & 0 & 1 \\ G & 2 & 0 & 0 & 1 \\ H & 2 & 0 & 1 & 2 \end{array}$$

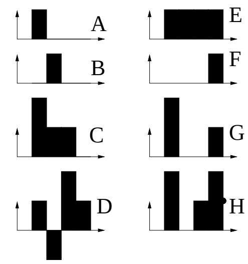


Fig. 7. Multivariate data for eight entities.

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 $X'|Y$

$$\begin{array}{lrrrr|rr} A & 1 & 0 & 0 & 0 & 0 & 3 \\ B & 0 & 1 & 0 & 0 & 0 & 2 \\ C & 2 & 1 & 1 & 0 & 2 & 0 \\ D & 1 & -1 & 2 & 1 & 4 & 2 \end{array}$$

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

$$\begin{array}{lrrrr|rr} A & 1 & 0 & 0 & 0 & 0 & 3 \\ B & 0 & 1 & 0 & 0 & 0 & 2 \\ C & 0 & 0 & 1 & 0 & 2 & -8 \\ D & 0 & 0 & 0 & 1 & 0 & 17 \end{array}$$

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.

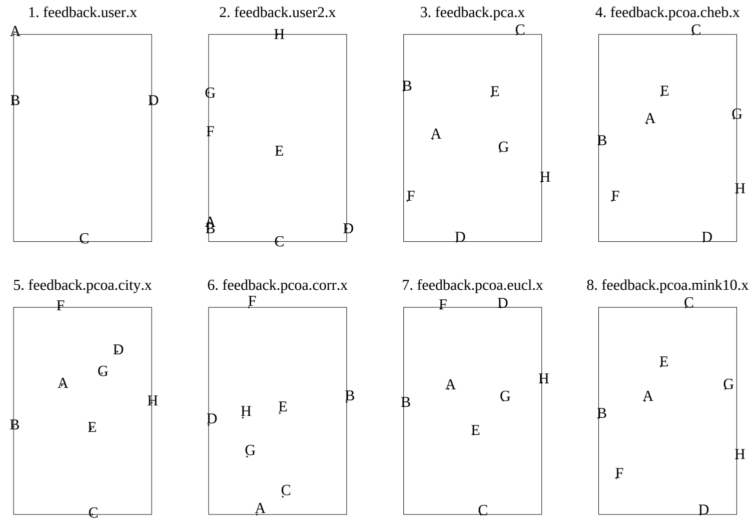


Fig. 8. 1. Four entities placed by a user. 2. Generalisation of these placements and application to all entities. 3-8. Scatterplot of the eight entities using PCA and PCoA with Chebychev, City or Manhattan (Minkowski with $\lambda = 1$), angular, Euclidean and Minkowski ($\lambda = 10$) distance.

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.