Enhancing Comprehension of Complex Data Visualizations: Framework and Techniques Based on Signature Exploration

Penelope Ann Noy

Submission for the Degree of Doctor of Philosophy

Department of Computing

August 2005

Abstract

This thesis presents a framework and set of readily applicable techniques for enhancing comprehension of complex data visualizations. Central to the work has been the definition and exploration of a new concept, **signature exploration**.

Visualization is being used increasingly to help make sense of large sets of data and information. Abstractions of complex data can be performed to reduce the dimensions to 2 or 3 for display. Novel or established representations can be used that allow direct mapping of greater numbers of attributes, and of a variety of data structures. There is an ever expanding set of visualization tools available. Two questions face the user: how to choose appropriate displays and how to understand the resultant graphic. This thesis examines how to support the user's comprehension in this context.

The work makes the following three main contributions to enhancing comprehension of complex data visualizations: the definition and application of signature exploration, a concept describing the exploration of visualization behaviour using specially constructed data; the proposal of a framework for the design of visualization systems for increased comprehension; the introduction of two new forms of interaction - which are here described as *visual data tracking* and *feature fingerprinting*.

The central theme for the exploration presented in this work is the notion that a user wants to take data that is *known* in some way, put this into the visualization process and assess the resultant visual depiction. This intuitive desire has been captured in the definition of the concept, *signature exploration*. Signature exploration describes the exploration of the behaviour of visual representations using specially constructed datasets that contain features of interest. The datasets are used to explore the signatures of different visual representations and mathematical transformations. The thesis defines and illustrates signature exploration, with five proposedapproaches: generic dataset provision; user-construction of data; querying; insertion of landmarks; elicitation and application of feedback data. These applications of signature exploration, together with analysis of the comprehension challenges presented by different aspects of visualization, and established work to support user comprehension, form the basis of the framework for increased user comprehension.

Example software has been developed within the context of a visualization application that employs a number of visualization algorithms to generate graphics for multivariate or proximity data. Principal Components Analysis, Principal Coordinates Analysis and distance metrics of various kinds are the algorithms used. An additional interface is given to the user, to perform signature exploration. The work has resulted in the specification of a set of techniques that developers can readily apply. Two new interaction forms are described: visual data tracking - bi-directional brushing and linking between representations also allowing change of position or value; feature fingerprinting - synthetic additions to real-world datasets to provide the user with calibration of the visual depiction.

Contents

List of Figures

List of Tables

Acknowledgements The work described in this thesis was funded by the Engineering and Physical Science Research Council and by BTExact (CASE studentship - award number 99803052). I am also grateful to these bodies for a 4 month extension that I received to help deal with the setback, in the second year of my work, of a serious fire at City University which destroyed my office, together with my paper collection and many notes. To the individuals at City, the EPSRC and BT who made this extension possible, my sincere thanks.

I am endebted to BTExact, the School of Informatics Research Committee, my supervisor, Michael Schroeder, AgentLink and the Society for the Study of Artificial Intelligence and the Simulation of Behaviour for providing funds for me to attend a number of international conferences and workshops.

When I began my PhD, I had no idea how many people would eventually have had a hand in bringing it to fruition. To all my colleagues at City University, in the Future Technologies group at BTExact, and so many people met at events in the UK and around the world with whom I have exchanged ideas, let me extend my thanks. In particular:

- My main supervisor, Michael Schroeder, for his enduring support and assistance during this work - especially for his good humour and optimism.
- To my colleagues Alex, Eddy, Jacek, Panos, Reinhold, Rodrigo and Tshiamo, and others in the Autonomous and Intelligent Systems group for their encouragement.
- To members of the Future Technologies Group at BTExact, particularly Robert Ghanea-Hercock, my company supervisor, for their interest and provision of data.
- To members of the International Cartographic Association Commission on Visualization and Virtual Environments, particularly Jason Dykes for his encouragement and insight.
- To Bob Spence for suggesting the words *signature exploration* and *landmark* and encouraging me to go ahead with tackling this topic.
- To Geraint Wiggins, who supervised me in the last stages of the work and helped me retain perspective.

Finally, enormous thanks to my husband, Mike, and my children, James, Simon and Rebecca, who have always encouraged me in this venture and never doubted my ability to succeed. Most especially to Mike for his immeasurable support, interest in all my ideas and belief in the importance of visualization.

This work is partly documented by the following papers:

- Noy, P. and Schroeder, M. Advancing Profile Use in Agent Societies. In Andrea Omicini, Paolo Petta, Jeremy Pitt (Eds.): Engineering Societies in the Agents World IV, 4th International Workshop, ESAW 2003, London, UK, October 29-31, 2003, Revised Selected and Invited Papers. *Lecture Notes in Artificial Intelligence 3071* Springer 2004, ISBN 3-540-22231-6, pp 360-375.

- Andrienko, G., Andrienko, N., Dykes, J., Gahegan, M., Mountain, D., Noy, P., Roberts, J., Rodgers, P. and Theus, M. Creating Instruments for Ideation: Software Approaches to Geovisualization. In MacEachren, A., Kraak, M.-J. and Dykes, J. (Eds.) *Exploring Geovisualization* (International Cartographic Association, Commission on Visualization and Virtual Environments). Amsterdam: Elseviers. Jan 2004.

- Noy, P. Signature Exploration, a Means to Improve Comprehension of Complex Visualization Processes: Issues and Opportunities. In MacEachren, A., Kraak, M.-J. and Dykes, J. (Eds.) *Exploring Geovisualization* (International Cartographic Association, Commission on Visualization and Virtual Environments). Amsterdam: Elseviers. Jan 2004.

- Noy, P. and Schroeder, M. Approximate Profile Utilization for Finding Like Minds and Personalization in Socio-Cognitive Grids. In *Proceedings of 1st International Workshop on Socio-Cognitive Grids: The Net as a Universal Human Resource*, in conjunction with Tales of the Disappearing Computer, Santorini, Greece. June 2003.

- Noy, P. and Schroeder, M. Aspects of Human Centred Design in the Role of Visualization Systems: Signature Exploration and the Use of Generic Datasets. In Gustav Lundberg, editor, *Proceedings of the Conference on Distributed Decision Making and Man-Machine Cooperation, Human Centred Processes 2003*, Luxembourg. May 2003.

- Noy, P. and Schroeder, M. Defining Like-minded Agents with the Aid of Visualization (Poster), In *Proc. of First International Joint Conference on Autonomous Agents and Multiagent Systems*, Bologna, Italy. ACM press. 2002.

- Noy, P. and Schroeder, M. Defining Like-minded Agents with the Aid of Visualization, In *ECML/PKDD Visual Data Mining Workshop Proceedings*, Helsinki, Finland. 2002.

- Noy, P. and Schroeder, M. Introducing Signature Exploration: a Means to Aid the Comprehension and Choice of Visualization Algorithms. In *ECML-PKDD01 Visual Data Mining Workshop*, Freiburg, Germany. 2001.

- Noy, P. and Schroeder, M. Mobile Agents for Distributed Processing. In *Infrastructure for Agents, Multi-Agent Systems, and Scalable Multi-Agent Systems* (T. Wagner, O. Rana, Eds.), pp. 263-5, Lecture Notes in Computer Science. Springer. 2001.

- Schroeder, M., Gilbert, D., van Helden, J. and Noy, P. Approaches to Visualisation in Bioinformatics: from Dendrograms to Space Explorer, *Information Sciences* 139 : 19-57. Elsevier. 2001.

- Schroeder, M. and Noy, P. Multi-agent Visualization Based on Multivariate Data, In *Proceedings of Autonomous Agents2001*. Montreal, Canada, pp. 85-91. ACM press. 2001.

Chapter 1

Introduction

1.1 Introduction

Vast quantities of data are accumulating; visualization can assist in making sense of this data. Visualization processes are increasing in their variety and complexity (Card et al. 1999) and are being presented to a wider range of users due to increases in the power of desktop computers (Spence 2001). In the field of visual datamining there is a desire to more closely integrate visualization and datamining processes (Shneiderman 2002), as well as develop the exploitation of the human visual system's pattern recognition abilities (Keim 2001). Developers have discovered many techniques for increasing the usefulness of visualization systems and allowing the user to manipulate and view their data in different ways (Card et al. 1999). Perception and cognition researchers have performed many experiments to examine how users perceive and draw conclusions from visual depictions (Ware 2000a).

According to one definition of visualization:

> "Visualization: The use of computer supported, interactive, visual representations of data to amplify cognition." *Readings in Information Visualization* (Card et al. 1999)

So visualization amplifies cognition, but only if the user can make sense of the results. Hence comprehension is key. Also, the current context of increasing variety and complexity of visualization systems, the inclusion of datamining techniques and the wider range of users, provide additional impetus for increasing the development of techniques to help the user make sense of the visual depictions and underlying computational processes.

This thesis explores a concept to increase comprehension of visualization methods. It specifies a framework and set of techniques to provide a guide for the creation of visual depictions and systems that enhance comprehension. The concept, *signature exploration*, is based upon the user's intuitive

desire to put familiar data into a visualization process, see what patterns result and to understand what effect the process has on the data. The concept is also inspired by work with querying of image libraries and by the use of *operational fingerprinting* in pyrolysis mass spectrometry. These two examples use known data in different ways to aid comprehension of the visualization.

The different ways that complex data can be visualized are examined to determine the problems that require support for comprehension. Existing visualization techniques, such as those relating to interaction, do support the user in various ways. The literature study presented with this work identifies these comprehension supporting techniques, thus contextualizing and reframing this work.

Five approaches for applying signature exploration have been explored: generic dataset provision; user-construction of data; querying; insertion of landmarks; elicitation and application of feedback data. Generic dataset provision provides datasets that are illustrative of particular dataset features for the user to see how well a feature is shown by a particular visualization method. User-construction of data provides an interface for the user to construct and alter their own data and thus design their own features of interest. Querying - visual or non-visual - helps users orientate themselves by seeing whether the data appears where they expect it to in the depiction. Landmark insertion is the highlighting of parts of the visualization, or of additional synthetic data, to orientate the user within the visualization. Elicitation of feedback data enables the comparison of the user's arrangement of the data with that of the system's. It also allows what is important to the user to be captured by the system and this can be applied to the system to modify its behaviour.

Example interfaces for the five approaches have been developed by extending an existing tool for visualization, *Space Explorer* (Schroeder et al. 2001). Space Explorer contains a number of clustering and visualization algorithms. The investigation revealed a number of obstacles which suggest further requirements for visualization designers, primarily with regard to accuracy. Experience of the examination of the five approaches and the study of existing techniques have been gathered together in a framework to guide the design of visualization systems as well as individual graphic depictions. Specific readily applicable techniques are proposed, including two new ones: *visual data tracking* and *feature fingerprinting*.

The background to this work was the consideration of visualizing complex systems, particularly multi-agent systems. Data for such systems typically originates as a log of events for a set of entities. An example real world dataset is used here of a set of 100 customers making telephone calls. A specific application area within the agent field has also been used, concerning the comparison of interest profiles.

The remainder of this chapter expands the visualization background motivation, describes the objectives and criteria for success of the work and gives an overview of the thesis.

1.2 Background

Increasingly large amounts and varied types of data are being stored. Web log files, e-commerce files, genome data, supermarket loyalty card data are examples. Because it is easier to store large quantities of data, much data is stored simply because it can be, rather than being the result of the design of an experiment, for example. On the other hand, enormous amounts of data are stored for legal reasons, such as the recording of call centre calls. The capability of data storage is also inspiring projects such as The Personal Image Memory Bank, containing video and sound data from every waking moment of a person's life (Ware 2000a), which illustrate the change in scope of data capture being undertaken. As more datasets become accessible through the World Wide Web ('the Web'), distribution creates composite datasets of much greater size. The Web itself can be viewed as one immense data repository, reminiscent of Gibson's concept of *cyberspace*, first introduced two decades ago (note that this is the only reference to the word in the book):

> "Cyberspace. A consensual hallucination experienced daily by billions of legitimate operators, in every nation, by children being taught mathematical concepts ... A graphical representation of data abstracted from the banks of every computer in the human system. Unthinkable complexity. Lines of light ranged in the non-space of the mind, clusters and constellations of data. Like city lights, receding ..." – *Neuromancer* (Gibson 1984)

The development of the *semantic web* based upon a new form of Web content that is meaningful to computers will, according to Berners-Lee et al. (2001) '... unleash a revolution of new possibilities'. One of the new possibilities is that different data sets have the potential to become linked via the naming of variables and entities. This now brings us the reality of Gibson's '... data abstracted from the banks of every computer in the human system. Unthinkable complexity.' In a sense we are becoming data bound, rather than processor, bandwidth or memory bound: for how are we to make sense of such masses of data? Furthermore, the problem of finding meaning in datasets does not just apply to extremely large datasets. Relatively small datasets may also be difficult to conceptualize - for instance a problem arises as soon as more than a very few dimensions are involved. Thus the problem of complexity covers a very wide range of datasets from the relatively small to the massive. Extremely large datasets are outside the scope of this thesis. For practical purposes this work has used smaller datasets, though many of the issues relating to comprehension remain the same.

How to make sense of this data? Visualization in such forms as ordinary graphs and bar charts has long assisted human endeavour. With the developments in computer processing power and graphics of the last ten to twenty years have come new ways - many visualization systems and novel representations, some developing the exploration of data in virtual worlds, inspired by Gibson's concept of cyberspace. Dimension reduction involving matrix transformation enables an abstraction of multi-

dimensional data to be plotted. Other novel ways of displaying many variables have been developed. At the same time, visualizations are being used by a more general user and in a greater number of applications, as well as in demanding specialist areas. A greater emphasis upon the goal of exploration, the data mining objective, is also evident. The integration of visualization mechanisms with data mining techniques, sometimes referred to as visual data mining, is being undertaken and provides the user with additional ways of exploring their data.

How to make sense of these visualizations? The complexity and inherent high dimensionality leads to visualizations that are hard to understand, clustering and other procedures may be carried out from a number of different points of view and thus produce different results, which lead to different conclusions. These conclusions may or may not be valid. Our attempt to portray complexity leads to loss of comprehension; there is a trade-off between understanding and number of dimensions represented. Thomas Green's work on cognitive dimensions seeks to address this general issue (Green 2000).

How to visualize complex data and systems? A particular application area for this work has been that of visualizing multi-agent systems. The possibility of a single whole system view is tantalizing, yet impossible, due to the many facets and high degree of interconnection involved. Thinking about social systems is illustrative, the general assumption is that no one view is the 'correct' one. Many views exist, from and of, many aspects. How then to produce computer visualizations of such systems and sets of data deriving from them? Increasingly, computer applications for visualization are presented as interactive processes that allow the user to create different visualizations and use visual querying. This approach avoids presenting a static view of the system, but makes the multi-facetedness implicit rather than explicit. So we can add to the complexity of a particular representation from abstraction or novelty, that of the complexity of the system from which the data is derived and the viewing of different aspects of the data interactively.

According to Colin Ware 'The human visual system is a pattern seeker of enormous power and subtlety' (Ware 2000a) and a substantial amount of information about the way we see has been collected by vision researchers. However, much of this work has yet to be taken advantage of and used in applications (Spence 2001; Ware 2000a). So the further exploitation of the human visual system as a pattern seeker may yield additional benefits in dealing with large amounts of information. Thus the potential of developing further new forms of representation and interaction for this purpose is indicated.

In this thesis the word *user* generally means the person that is using the visualization application. However, there are four sub-classes of users for the purposes of this work. This is determined by the person's level of background knowledge or capability (described here as expert or general) and their

familiarity with the particular visualization method being used[1].

1. Expert user familiar with the particular visualization method.
2. Expert user unfamiliar with the particular visualization method.
3. General user familiar with the particular visualization method.
4. General user unfamiliar with the particular visualization method.

Examples of expert users are professional scientists and financial analysts i.e. those with a mathematical background. General users lack the background to understand the technique fully where complex transformations are used. It is often still valid for general users to use such visualizations, because it is not always necessary for them to have complete understanding. In these discussions it is assumed that a user is an appropriate user for a particular visualization method, i.e. that they have the potential capability to use it effectively. This work is aimed primarily at users (2) and (4), who may be described as 'inexperienced', but it is also assumed that users (1) and (3) can benefit from aspects of this work.

Requirements of users of visualization systems are varied: the two main types of task are searching for a specific piece of information and exploratory data analysis (Card et al. 1999). However, there are other purposes of visualization. For instance, visualization of a multi-agent system is used to conceptualize the operation of the system to viewers, as well as to provide a means of monitoring the system. Another, different example of visualization is in network monitoring, where visualization is used so that network misuse and intrusion can be detected (Erbacher and Frincke 2000). Despite the varied requirements of users, for the purposes of this thesis, it is assumed that the underlying requirement of users is to 'understand the visual depiction, since this is a pre-requisite upon which task success relies. In the example used as illustration throughout the work, the user is assumed to be exploring a multi-dimensional data table.

Preliminary work in this project examined various visual representations of a variety of datasets: a call data log, gene expression data and web site access log files. For validation and familiarization purposes, specially constructed data sets were used. These specially constructed datasets were quite simple, containing small numbers of entities (< 10) and small numbers of attributes, with features such as identical or similar entities and randomly assigned attribute values. Aside from for validation of the software, these were useful as concrete introductory examples of the behaviour of the representation. For instance, identical or very similar entities that are shown in the same place in the display are not apparent to the human viewer. These simple examples indicated the potential for using known data to illustrate different representations.

[1] However, this description of users is distinct from users of the framework described in Chapter 12,which are designers of visualization systems, though expert end-users can use the framework to assess the design.

Further inspiration for the use of constructed data comes from work on dynamic querying of image libraries (e.g. Chang and Fu (1980), Pu and Pecenovic (2000)). 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 maybe invited to choose a selection of images and give these to the application to arrange in terms of similarity (Pu and Pecenovic 2000). This process can provide insight into the behaviour of the algorithm that is choosing similar images from the database. Another related concept is that of *fingerprinting*, a technique where an unknown pattern is compared to a set of known ones. In pyrolysis mass spectrometry, the inclusion of a known outlier and reference organism in the dataset guides the user in the interpretation of visual depictions of multidimensional data in a process known as *operational fingerprinting* (Meuzelaar et al. 1982).

As a generalization of this problem, a set of call data has been used for illustration. 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 visualisations use the destination local exchange reference in a data table of the form shown in Table 1.1, such that entry x_{ij} is the number of calls made by customer i to location j.

	destination 1	destination 2	.....
Customer 1	0	3	.....
Customer 2	1	47	.....
...	...		

Table 1.1: Call data customer/destination matrix used for visualization

An example of the problem of comprehension is shown in Figure 1.1, a screen shot of a three-dimensional VRML world. This is a visualization of the call data whose dimensions have been reduced from 276 destinations to 3, for the 100 entities. This dimension reduction can be achieved by using one of a number of methods, in this case, Principle Components Analysis has been used (more detail about these methods will be found in Chapter 3 on page 36). The problem is that it is hard to know what conclusions can be drawn from this shape. In fact, by rotating the virtual world, the shape is, to a human, reminiscent of the leg and webbed foot of a duck, perhaps. Is the only conclusion, that two customers close in this representation are similar? If so, in what way are they similar? The inexperienced user may have no idea as to the answers to these questions, the experienced user will have a better idea, but will need to interact with the data in order to test their hypotheses.

Summarizing the problem, the direct result of the complexity or novelty of representations is that an inexperienced user's initial reaction to a graphic may be 'What does this mean?' Thus, users need methods and tools that help them understand the necessarily abstract representations required to depict complex data. This problem is more marked in visualization systems that present views of

Figure 1.1: 90,000 telephone calls made by 100 customers. Caller profiles by destination of calls. Dimensions (which are the destinations) have been reduced from 276 to 3 by applying Principal Components Analysis. This is a screenshot of a 3D VRML world. Rotating the world shows the shape of the cluster to be similar to a duck's leg and webbed foot.

complex systems, for example multi-agent systems. The user needs to know the characteristics of particular visual representations. Intuitively the user wants to take known data and put this into the visualization process to see what happens. It is this intuition and the experience indicated above, that has led to the proposal of a new concept, *signature exploration*. Signature exploration uses datasets that are known in some way, to explore the behaviour, or signatures, of the different visualization techniques. It is the exploration of this technique which is the basis of the work presented in this thesis.

This thesis uses the terms perception, cognition and comprehension in regard to the human response to visual depictions of data. The following paragraphs discuss how these words are used.

The word perceive (and thus perception) is used in everyday speech to indicate also a cognition and comprehension. Consider the following dictionary entries[2]:

- Perceive:
    1. a. To attain awareness or understanding of. b. To regard as being such
    2. : to become aware of through the senses
- Perception:
    1. a. A result of perceiving. b. A mental image.
    2. a. Awareness of the elements of environment through physical sensation. b. Physical sensation interpreted in the light of experience.
    3. a. Quick, acute, and intuitive cognition. b. A capacity for comprehension.

In this discussion, perception is taken to be 'becoming aware of through the senses'.

The dictionary definition of cognition is as follows:

- Cognition:
    1. The mental process of knowing, including aspects such as awareness, perception, reasoning, and judgement.
    2. That which comes to be known, as through perception, reasoning, or intuition; knowledge.

Thus this shows also an overlap, in usage, with the word perception. This thesis uses the meaning 'the act or process of knowing including awareness, perception, reasoning and judgement'. The discipline of Cognitive Science has the goal of 'understanding the mind and its operation' (Thorgard 1996) - the objective of this work is not to understand how the mind works when the human is viewing

[2]Merriam-Webster's Collegiate Dictionary: http://www.yourdictionary.com.

complex data visualizations, but to ensure that the mind has the correct starting information with which to reason. In this sense we take 'awareness' as the key attribute: making the user aware of the appropriate aspects. Thus we do not examine how the person draws conclusions, but whether (assuming they *do* have the skills to use the information) they have access to the information they need about the representation. This overlaps with the areas of perception and cognition, but does not focus upon them.

The thesis title uses the word 'comprehension' in the general dictionary sense:

- Comprehension:

    1. a. The act or fact of grasping the meaning, nature, or importance of; understanding. b. The knowledge that is acquired in this way.
    2. Capacity to include.

'Comprehension is enhanced' (in the words of the thesis title) by determining what information the user needs and making sure that it is presented to them, or that the discovery of such information is facilitated. The 'information' of interest is how features are represented, how well they are represented, what features are not represented and so on. Such information is a prerequisite to comprehension. This corresponds to 'read fact' and 'read pattern' of a knowledge crystallization task as described by Card et al. (1999).

1.3 Objectives

The overall objective of this work was to examine the usefulness of signature exploration in increasing the comprehension and choice of visual displays of complex data. The underlying aim is to increase comprehension of such displays, so that an analysis of obstacles to comprehension and relevant current techniques was also undertaken. Whilst it is defined more precisely in this thesis, in essence, the term *signature exploration* is used as a convenient phrase to describe the general use of specially constructed datasets to reveal the behaviour of the transformations and representations involved in visual displays.

1.3.1 Research Approach

The approach taken for this work was as follows:

- Analyze aspects of the visualization process to determine obstacles to comprehension.
- Analyze experience of users and potential users of complex data visualization systems:

  - Biologists: protein data (structural and functional). Partly documented in the paper *Approaches to visualisation in bioinformatics: from dendrograms to Space Explorer* (Schroeder et al. 2001).
  - Computer scientists: multi-agent system data. Carried out during an industrial placement with BTExact's Future Technologies Group.
  - Managers: business data. Views from business users in discussions following presentations on data visualization as part of the Department of Trade and Industry's Software Outreach programme: 'Visual Data Mining: Theory and Applications' (25th April 2002) and 'Visual Data Mining and its Application in Biology' (12th March 2003) presented jointly with Michael Schroeder. Also as a result of discussions with the voice and data recording company, Eyretel.

- Analyze different purposes of visualization of complex data.
- Implement initial prototypes of biological, multi-agent system and web site usage data in 3D using VRML.
- Develop concept of signature exploration and means of its application based on discussions with users, experience of initial prototypes and academic peer review (submitting and presenting papers at workshops in visual datamining (Noy and Schroeder 2001, 2002a)).
- Conduct feasibility study.
- Examine relevant literature for each of the proposed means of application and implement illustrations of each approach.
- Review obstacles.
- Summarize findings in a framework identifying obstacles to comprehension and techniques to assist.
- Evaluate framework by applying to existing visualization applications.

The reader should be aware that this is a large problem area and, since the scope of the work was so wide, the question of how to focus the research arose. It was clear that the entire duration of the PhD could be spent investigating a specific technique or aspect. However, restricting the study to one aspect would have meant that promising results in other areas were not uncovered and that the value of the overall concept could not be assessed - the project would become a very different one. Thus is was decided to maintain the meta-level, concept-level approach to assess the value over the whole area, to identify areas of most promise and to maintain the wide scope of the remit. Therefore

the study of each of the five signature exploration approaches was limited to the use of illustrative applications. It would have been impossible to produce a finished application containing all the proposed techniques, so that the work was approached as a series of experiments, some of which required new interfaces to be created within the application, some of which required programmes to be written and investigations carried out separately from the application.

The work takes place in the context of the ongoing development of an existing tool containing clustering and visualization algorithms, *Space Explorer*. New functionality and interfaces were added to enable the proposed techniques to be illustrated - the application providing a test bed for the experiments. Thus the focus of the work was the exploration of the overall aim, not the production of a prototype application.

1.3.2 Hypotheses

To provide focus for the examination of signature exploration, the following hypotheses were used:

1. The application of the concept, signature exploration, aids the comprehension of visualizations of complex data.
2. The application of signature exploration aids the choice of display of complex data.
3. The application of signature exploration can form the basis of a framework for the design of visualization systems for increased comprehension.
4. The application of signature exploration will lead to the development or specification, or both, of a suite of techniques for aiding comprehension.

1.4 Criteria for Success

The goals of the previous section can be broken down into a number of criteria for success:

1. Defining a concept for applying constructed data, signature exploration.
2. Specifying a set of techniques for the application of constructed data.
3. Identifying problem areas and obstacles.
4. Reframing existing techniques.
5. Implementing examples of the different techniques.
6. Developing a framework for the design of visualization systems for increased comprehension.
7. Specifying a set of techniques for aiding comprehension of visual depictions.

Chapter 13 evaluates the research presented in this thesis against these criteria.

1.5 Structure of Thesis

This thesis is organized into thirteen chapters and an appendix. This first chapter has given an overview of the work of the thesis and provided some background as rationale and motivation. Objectives have been discussed, hypotheses posed and criteria for success set. This description of the thesis structure concludes the chapter.

The remaining chapters divide roughly into three parts: firstly, a discussion of data formats, layout and morphologies for visualization, leading to a review of open questions for the field; secondly, the definition and investigation of signature exploration; and thirdly, the proposal of a framework and set of practical techniques together with evaluation and conclusions.

The content of each chapter is as follows:

Chapter 2 examines the different types and structures of data for visualization. It suggests that it is useful to consider the starting point for the data to be a set of entities and a log of events from which various types and structures of data may be derived. Obstacles to comprehension arising from these elements are identified.

Chapter 3 examines the different ways that data can be transformed into a form that can be displayed. It shows that there is an interchangeability between graph layout and proximity data on the one hand and multivariate data on the other. General issues, such as time complexity and scalability are discussed, as well as the number of dimensions available for direct mapping and the way that human perception impacts upon the potential and limits of visual depiction. Obstacles to comprehension arising from these elements are identified.

Chapter 4 is a survey of visualization morphologies, loosely grouped as direct table mapping (with or without dimension reduction) and tree representations. Navigation and interaction are discussed, as they become problematic with increased size. A summary lists the ways that the representations and abstractions present challenges to understanding for the user.

Chapter 5 revisits the open questions of the field in light of the breadth of developments and possibilities indicated in the previous three chapters. It outlines five questions and concludes that the support of the user's comprehension of visual depictions is a key issue which relates to, and serves, the others. Two of the open questions, regarding composite tools and visualizing complex systems, form the basis for a design for a complex system viewer whose ongoing development is the context of this work.

Chapter 6 defines the signature exploration concept and describes five proposed techniques. Each technique involves the production or provision of specially constructed data containing a feature or features of interest.

Chapters 7 to 10 examine the five techniques for applying signature exploration (two are combined in Chapter 9).

Chapter 11 examines the obstacle presented in the form of accuracy. It returns to the literature for relevant work. An empirical study is described which explores accuracy of layout and describes two developments that resulted from this study - a new layout algorithm and a form of profile exchange for agents that is conveniently lightweight and private.

Chapter 12 draws together the results of the work into a framework for the design of visualization systems for increased comprehension and puts forward a list of readily applicable techniques.

The last chapter, Chapter 13, discusses the contributions made by the research, evaluates the work against the criteria for success of this introduction, and outlines future work.

The appendix gives an example of the application of the framework to a different visualization scenario in which the tool Attribute Explorer (Spence and Tweedie 1998) is used.

Chapter 2

Data Types and Structures

2.1 Introduction

Since the starting point for visualization is data, the type of data to be visualized and the data structures involved need to be examined. It will be shown (partly in this chapter and partly in the next) that structures and types can be, to some extent, transformed one to another. The fluidity between structures leads to the need for a base structure that other structures can be considered to derive from. Such a structure is also useful because the data, in its raw form, allows the derivation of different structures. The discussion of possible base structures is also relevant to the design of generic visualization tools, (leaving aside the discussion of the extent to which generic tools are possible or desirable), since a generic application needs a base data structure to start from.

Aside from the data selected for visualization, there is associated implicit data, for instance through the naming of the objects or variables. Various forms of such associated data are described as metadata. Metadata is included in this discussion of the chosen starting data, because it is becoming more common, and desirable, to link in data from other sources during the visual interaction. This linking of other data is described in the *relate* type of the task taxonomy of Shneiderman (1996), though the emphasis of the author is the linking of general information, rather than other specific datasets. The selection of subsets of the data attributes and the generation of new attributes are issues of similar relevance, since the resulting visual depiction is highly dependent upon the choice of attributes.

This chapter describes types and structures of data, suggests a base structure for complex data and discusses various forms of metadata, together with selection and standardization. Comprehension issues arising out of this discussion of data types and structures are noted. The content of this chapter forms the basis from which to examine, in the following two chapters, layout and morphologies for visualization and to further examine aspects which present comprehension difficulties for the user.

2.2 Types

In general data is considered to relate to a number of *objects* or *entities*. These objects are generally described by a set of variables, or attributes, and between the objects there may be explicit (rather than derived) inter-object relationships. The terms *variable* and *attribute* are used interchangeably here. For instance, telephone customers may be considered to be the objects, described by properties, such as whether they are a business customer or not, what tariff band they have chosen etc. These customers may call one another, an example of an explicit relationship. What constitutes an object is not examined in detail here, except to mention two points. Firstly, 'object' covers a wide range of conceptual entities, for example Wagner (2003) allows an entity (object) to be an agent, an event, an action, a claim, a commitment, or an ordinary object. Secondly, variables and objects are interchangeable, since, where a relation exists between two sets of objects, the inverse exists also. For instance, in the call data introduced on page 6, the objects are customers. However, the data can also be looked at from the point of view of the destinations as objects, which allows clustering to be carried out to examine the similarity of the destinations, rather than the customers.

2.2.1 Variable Types

Variable types divide into two groups: *qualitative* and *quantitative*. Qualitative variables are variables that can be placed into distinct categories, according to some characteristic or attribute. They can be divided as follows:

- *Binary*: a binary variable can take one of two states, for example, 'true' or 'false'.
- *Multistate*: these are variables for which there are more than two states or categories. If the categories are ordered, the categorical variable is described as *ordinal*. If not ordered, the categorical variable is described as *nominal*.

Quantitative variables are numerical in nature and can be ordered or ranked. They can be divided into two groups, *discrete* or *continuous*. Discrete variables can be assigned values such as 0, 1 , 2, and are said to be countable. Continuous variables can assume all values between any two specific values and are obtained by measuring.

Often datasets contain different types of data. The examples used in this thesis are confined to those using a single variable type, quantitative, for convenience. Quantitative data can be transformed into qualitative data, with associated loss of information, which makes this transformation one-way only.

2.2.2 Data Types

Card et al. (1999) remind us that the most common way of using space in visual displays is to mirror the physical world. Where there are spatial variables that can be mapped directly to the spatial substrate of a visual structure, the data are described as *physical*. Examples given are molecules, medical images, brain structure, meteorology, space exploration and astrophysics. The visualization of such data is the most direct in mirroring the physical world. Leaving aside physical data, Card et al. (1999) consider four ways that space is used to encode abstract data: 1D, 2D, 3D; multiple dimensions > 3; trees; networks (Figure 2.1). *1D, 2D, 3D* refers to visualizations that encode information by positioning marks on orthogonal axes. *Multiple dimensions* > 3 concerns the harder problem of data that cannot be visualized with three orthogonal axes. *Trees* and *Networks* relate to visualizations that use connection and enclosure to encode relations between objects. By enclosure is meant the techniques of enclosing a level of a hierarchy within a symbol representing a higher level as in *treemaps* (see Section 4.4.5).

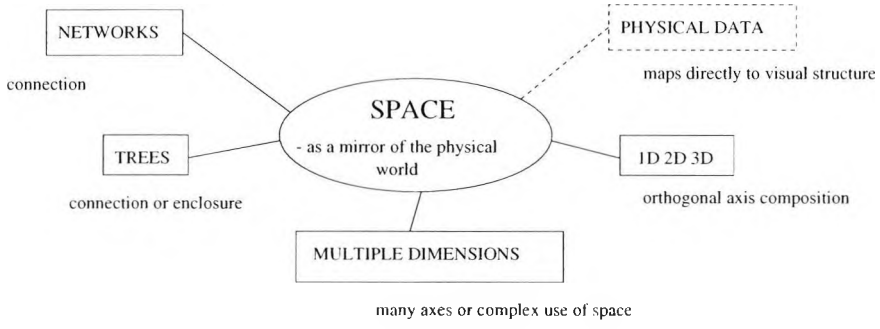


Figure 2.1: The way space is used to encode data in visual structures according to Card et al. (1999). Leaving aside physical data, this proposes four different ways to encode abstract data.

Though this provides us with a taxonomy of encodings for abstract data, this also is a taxonomy of data types in the general sense of 'types', and so provides a way of thinking about the different types of data for visualization. Indeed, adding *temporal* gives Shneiderman's information visualization taxonomy by data type (Shneiderman 1996) as in Figure 2.2, though this taxonomy does not consider dimension reduction or discuss the aspect of transformation between types.

Data without physical reference are often described as *abstract* data. Generally speaking, the field of *information visualization* considers only the visualization of abstract data, as in the definition given by Card, Mackinlay and Shneiderman in the introductory chapter in their book 'Readings in Information Visualization' (Card et al. 1999):

> "Information Visualization: The use of computer-supported, interactive, visual representations of abstract data to amplify cognition."

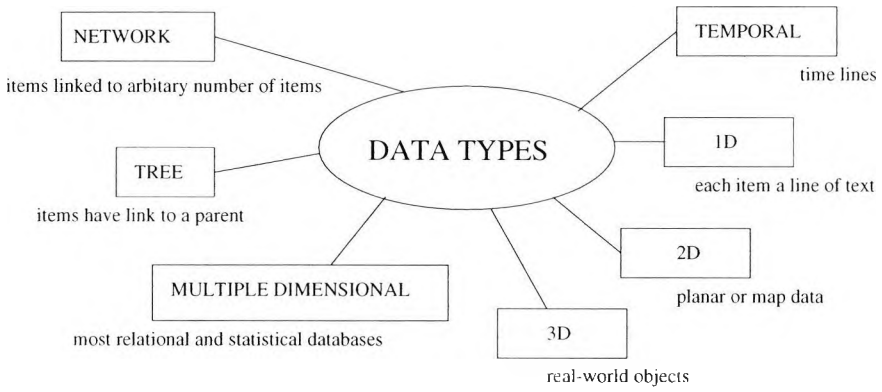


Figure 2.2: Data types for information visualization, according to Shneiderman (1996)

Elsewhere in the text (Card et al. 1999, p. 7) they describe abstract data as that which does not have any 'obvious spatial mapping ... nonphysically based'. Spence (2001), in his book 'Information Visualization', does not define the term explicitly, but says 'the need to display the physical thing is not important ... and is often entirely irrelevant ... in information visualization' (p4). This is a narrowing of the meaning of 'information' as it is generally used. This narrow interpretation of the word 'information' as established in the field of information visualization is followed within this thesis and the data used for illustration is abstract in the sense of being nonphysically based, though the application is more general.

The use of the word 'types' here is distinct from how that word is used in programming where a datatype is a name or label for a set of values and some operations which can be performed on that set of values[1]. The notions of scope, lifespan, and initiation are also not directly relevant. One can find parallels for scope, lifespan and initiation, in that types can be transformed and thus have initiation, scope and lifespan in the context of a data exploration. Type safety is also a relevant concept, though with a different meaning, since a particular visualization method is suitable only for particular types of data. However, these terms are not normally used in the discussion of visualization systems.

2.3 Data Structures

Some surveys take a graph (see Definition 1 below) as the central data structure from which others may be derived (e.g. Herman et al. (2000)). This thesis takes the two main ways of expressing the overall structure of the data as *graph form* and *table form*. The more general consideration of the originating structure of the data, and of time and events, are examined in Section 2.4. To a certain extent graph and table forms are interchangeable, i.e. some data can be expressed equivalently in both. The following definitions cover:

[1] Wikipedia http://en.wikipedia.org/wiki/Datatype

- Graph form: *graph* (Definition 1), *connected graph* (Definition 2), *tree* (Definition 3)
- Table form: *multivariate data* (Definition 4), *proximity data*, (Definition 5.)

Definition 1 *Graph, Digraph*

*A **graph** G is a pair (V, E), where V is a finite non-empty set of elements called **vertices** or **nodes** and E is a finite set of distinct unordered pairs of distinct elements of V called **edges**. The edges may have values, called **weights**, associated with them. If directions are imposed on the edges of a graph, interpreting the edges as ordered rather than unordered pairs of vertices, the corresponding structure is called a **directed graph** or **digraph**.*

Definition 2 *Path, connected graph.*

*A **path** from a vertex u in G to a vertex v in G is an alternating sequence of vertices and edges,*

$$v_1, e_1, v_2, e_2, \ldots, e_{k-1}, v_k$$

*where $v_1 = u$, $v_k = v$, all the vertices and edges in the sequence are distinct, and successive vertices v_i and v_{i+1} are endpoints of the intermediate edge e_i. A graph G (Definition 1) is **connected** if there is a path joining each pair of vertices of G.*

Definition 3 *Cycle, tree, rooted tree.*

*If the definition of path, above, is relaxed to allow the first and last vertices (only) to coincide, the resulting closed path is called a **cycle**. A connected graph (Definition 2) which contains no circuits, or cycles, is called a **tree**. A vertex v of a digraph G is called a **root** if there are directed paths from v to every other vertex in G. A **rooted tree** is a tree in which a distinguished vertex v may be thus identified*

Definition 4 *Multivariate data.*

***Multivariate data** for a set of n objects is an $n \times p$, objects $\times$ variables matrix, whose (i, k)th element provides a value or category for the kth variable describing the ith object ($i = 1, \ldots, n; k = 1, \ldots, p$).*

An example of multivariate data for three objects is shown in Table 2.1. The variable attribute values may represent a time series. The resulting matrix is referred to as a *pattern* matrix.

	x1	x2	x3	x4
a	3	2	0	5
b	0	7	0	4
c	3	4	1	6

Table 2.1: Multivariate data example: attributes $x1$ to $x4$ for entities a, b and c.

Definition 5 *Proximity data.*

Proximity data *for a set of n objects is a symmetric $n \times n$ matrix, also called a* ***dissimilarity*** *or* ***distance*** *matrix D, whose (i, j)th element provides a measure of the dissimilarity, d_{ij}, between the ith and jth objects $(i, j = 1, \ldots, n)$. The following conditions are required:*

$$d_{ij} \geq 0$$

$$d_{ii} = 0$$

$$d_{ij} = d_{ji}$$

D is said to be metric if it satisfies the triangle inequality -

$$d_{ij} \leq d_{ik} + d_{kj}$$

for all triples of objects (i, j, k).

D is said to be ***Euclidean*** *if there exists a configuration of points in Euclidean space $P_i (i = 1, \ldots, n)$ with the distance between P_i and P_j equal to d_{ij}.*

If D is Euclidean it is also metric, but the converse does not hold. Consider four points, three of which are located at the vertices of an equilateral triangle and the fourth is the centre of the triangle. Now slightly reduce the distance between the fourth point and all other points, D is still metric, but not Euclidean. However, most layout methods (see Section 3.4) will still operate if the proximity data is not Euclidean. An example of proximity data is given in Table 2.2. Similar matrices can be used to describe relationships between entities that are not symmetric. Consider, for example, a messaging system: total messages exchanged may be represented (symmetric), or the messages sent from entity a to entity b recorded separately from messages sent b to a (asymmetric) (Schroeder and Noy 2001).

Multivariate data can be transformed into proximity data by the use of metrics which result in a set of dissimilarities or distances between objects. Distance measures are discussed in the following chapter in Section 3.4.2.

Card et al. (1999) give a reference model for visualization which moves from raw data, to data tables, to visual structures, to views. The raw data is described as being in 'idiosyncratic formats' such as spreadsheets or the text of novels. Notice that the raw data may already be a data table, since

	a	b	c
a	0	2	3
b	2	0	4
c	3	4	0

Table 2.2: Proximity data example: shows 'distance' between entities a, b and c.

it must be in some kind of format. The data tables are relations of cases (the entities or objects of our discussion) by variables plus metadata. The metadata in this case is the data that describes the relations in the form of labels of the rows and columns of the data table (various types of metadata are discussed in Section 2.5). Card et al. consider that a table of proximity data (which they call a two-way table) is not a data table according to their definition. They produce a version of the data table that satisfies their definition, but this is not as direct an expression of the data. Whilst the notion of data table as defined by Card et al. satisfies a useful reference model, for this thesis the emphasis is upon how we can conveniently think of the data. In earlier work visualizing data and considering a generic approach, tabular forms (of both proximity and pattern matrices) and graph forms suggested themselves as the two main ways. There is also a close correspondence between these forms and the structural forms that can be expressed as visual structures in visualization applications. The graph structure has also been studied in mathematics and continues to be (see for example, Matoušek (2002)). Thus, for this analysis, both table and graph forms are used.

2.4 Base Structure for Complex Data

What is meant by *complex data*? This thesis uses the expression in a general sense to mean data that is complicated, that is not simple enough to visualize directly. This covers a wide range of datasets from the relatively small to the massive as discussed in Section 1.2. Such data includes that from complex systems.

The New England Complex Systems Institute (http://necsi.org/guide/whatis.html) defines complex systems as follows:

> Complex Systems is a new field of science studying how parts of a system give rise to the collective behaviors of the system, and how the system interacts with its environment. Social systems formed (in part) out of people, the brain formed out of neurons, molecules formed out of atoms, the weather formed out of air flows are all examples of complex systems. The field of complex systems cuts across all traditional disciplines of science, as well as engineering, management, and medicine. It focuses on certain questions about parts, wholes and relationships. These questions are relevant to all traditional fields.

Briefly:

> A complex system has multiple interacting components whose collective behavior cannot be simply inferred from the behavior of components.

This emergence of a collective behaviour from interacting components is often termed *emergent* behaviour, or *emergence*. Whilst the study of complex systems is the study of this emergent behaviour, all data is recorded as part of a complex system, since all entities must exist in the world as part of complex systems. Also data from complex systems is not confined to the emergent component. For these two reasons, (that complex systems are often considered to be special, rather than general, and only relate to the emergent component), the term *complex data* is used here, rather than *data from complex systems*. However, it is assumed that complex data encompasses data from complex systems.

In complex data, data tables and graph structures (hierarchies and a variety of networks) may coexist, thus it is desirable to make the starting point more general. This general structure may be the conceptual or actual originating data structure and is assumed to comprise entities with properties and relationships between the entities as introduced at the beginning of Section 2.2. The properties and relationships change over time. The entities exist in an environment, which is also changing. Actions and events may be considered separately or as the profile or *history* of the entities, or as entities themselves. Systems are considered that exist in time, so there is a sequence of events. This view of the system suggests starting with a set of entities, together with a log of events, and deriving other data structures as required, as indicated in Figure 2.3. Whilst the actual originating data may consist of details of properties of a set of entities and a log of events (Definition 6), the data may be stored in various ways. It may be that all the data is put into a database that can be queried, or stored in some other form. Thus the conceptual base structure does not imply the form that the data is stored in, nor the actual originating data structure. It is a means to generalize the starting point for the visualization process.

Definition 6 *Log.*

A ***log*** *is a record of events concerning a set of entities. Each entry in the log gives event information in the form Time, Entity-name, Action-name, Action-parameters.*

The time entry in the log is optional, but the entries in the log are assumed to be a time-ordered sequence, whether or not the time is specified. The action may be a change in a property value or it may involve an interaction with another entity, for example in sending a message. The creation of the entity itself can be considered as an event and therefore that there are no initial properties or entities.

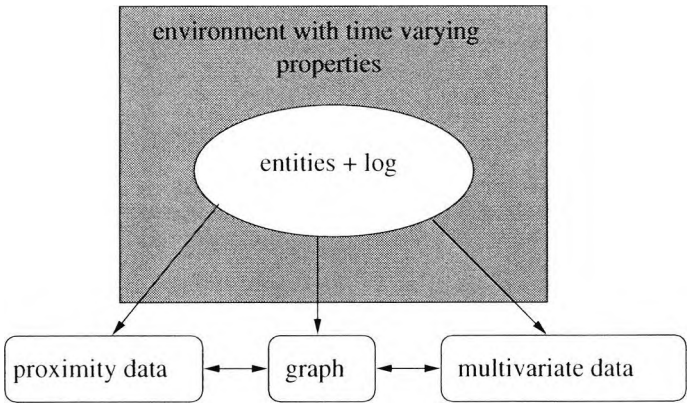


Figure 2.3: Possible data structures and origin for complex data.

2.5 Metadata

A set of data for analysis or visualization normally has an associated body of other information that does not appear explicitly in the data. This information can be described as meta-information or metadata. Metadata literally means data about data. It has also been described as 'descriptive information about data' (Card et al. 1999). Less commonly it is used to refer to transformed data (Tweedie 1997). It can also include the ordering of entries in the data table (Card et al. 1999, p. 18). The different types of metadata can be divided into those regarding structures that can be *derived* and those regarding *linked sets* of information. The most elementary forms of metadata are relationships between values of certain variables, or boundaries (upper and lower) for these values. Another simple form is that associated with variable names. For instance, the variable name *Numbers* implies that interpolation is possible, if the numbers are heights above sea level, for instance, and that interpolation is not possible, if the numbers represent, say, car accidents. This example is of implicit metadata in the variable name (more information is combined with the variable name), but the variable name *Numbers of Car Accidents* would indicate this explicitly, since the use of natural numbers is implied. The other way that variable names can provide metadata is through providing linked information. For instance a destination of a telephone call *Birmingham* (a proper noun) can link to other data about Birmingham, or the variable name *town* (a common noun) can link to information about what a town is, how it may relate to the other variables and so on. The importance of this type of metadata is increasing with databases linked via the Web, and the development of the semantic web, as described in Chapter 1 on page 3. This type of metadata may also be seen as data about the current objects that is not being used and thus is a form of selection, which is discussed in the next section. As more datasets become available via the Web, and with the more widespread development and use of agreed ontologies, it will become the case that a dataset that one is viewing is only a portion of the data about the objects or attributes that are available in real time. In effect, an application can use the web

as its database. MacEachren and Kraak (2001), for geovisualization, suggest new kinds of maps 'no longer conceived of as simply graphic representations of geographic space, but as dynamic portals to interconnected, distributed, geospatial data resources.' This *dynamic portal* to resources applies to many other kinds of data, not only that including geospatial referencing.

Metadata is particularly extensive for certain types of data, such as textual data, for instance where word counts (instances of words in a particular document) are involved. Metadata for word counts from textual data includes all of semantics, syntax and stylometry. Stylometry is the measure of style, which is assumed to contain distinctive and quantifiable elements, which are not consciously controlled by the author, but which mark the style of an author.

Is it always useful to consider the entire set of available metadata? Lebart et al. (1997) give an example in the area of image analysis concerning histograms of wavelengths of the colours of a Rembrandt, based on the colours of individual pixels. Only a fraction of the information contained in the original image is contained in these histograms, but it is possible that the shape of the histogram could distinguish a Rembrandt from a Rubens or Van Dyck. In many information retrieval tasks key words only are used, so that the document is reduced to a reduced set of words with no order or syntax. These examples, of a small amount of the available data being usefully employed, are repeated in many classification applications (see additional discussion in the next section).

2.6 Selection and Standardization

The question of whether to use available metadata, is similar to the question of whether to use all immediately available data, for instance, all data in a data table. Sometimes it is clear what variables should be used to describe objects. However, often variables have to be selected from many possibilities; it is selection *prior* to the visualization process that is meant here, as distinct from selection *during* the process, such as by selecting a subset using highlighting. The process of selection is often not straightforward. The pattern recognition literature describes the appropriate specification of variables as feature extraction. It is tempting to include a large number of variables to avoid excluding anything relevant, but the addition of irrelevant variables can mask underlying structure (see for example Gordon 1999, p. 24).

Having determined appropriate variables, there is then the question of standardizing or differentially weighting them. One aspect to the standardization is that two variables can have very different variability across the dataset. It may or may not be desirable to retain this variability. Standardization may also be with respect to the dataset under consideration or with respect to a population from which the samples are drawn. In the case of quantitative variables, standardization can be made by dividing by their standard deviation or by the range of values they take in the data set. The idea of standardization lies within the larger problem of the differential weighting of variables.

Thus the exploration of complex data is tempered by the knowledge that for many purposes a small subset only of the data is required. One goal of visualization applications must be to facilitate further selection and manipulation of the data by the user - the modification and recreation of the feature list. Whilst the importance of this area is acknowledged, the exploration described in this work confines itself to the visualization of data largely 'as given'.

2.7 Comprehension Challenges

This section notes specific points from the descriptions and discussions above that introduce difficulty in comprehension or require special facilities. The difficulties can be summarized as arising from the interchangeability of objects and attributes, the fluidity between structures, the variety of types and structures of data to be visualized, the availability of large amounts of metadata of varying types, and the general impact of selection and standardization. These problems are described in more detail in Table 2.3, together with suggested visualization facilities to assist. The suggested facilities fall into two groups: those needed to make the user aware of significant aspects of the visual representation (for instance: the data/representation mapping, the sensitivity of representations to selection and normalization, the different ways the data can be viewed); those needed to allow transformations (between types and structures) and selections.

An additional challenge to comprehension of multivariate data lies in the difficulty of conceptualizing high dimensional spaces. How are we to grasp the concept of, say, a 20-dimensional structure? This is a central question of the book 'Multidimensional Man' (Atkin 1981). Atkin argues that 'hypervolume' is a 'lazy' term, since it does not specify the number of dimensions, and that it is misleading, since it implies 'something like a volume only more so' (p72). The treatment of dimensions greater than three as extensions of the Euclidean system for dimensions 1, 2 and 3 is very convenient (the metrics introduced in the next chapter in Section 3.4.2 do this), but in fact a variety of mathematical effects can be observed when one increases the dimensionality of the data space, some of which are non-intuitive (Böhm et al. 2001). Specifically, important parameters such as volume and area depend exponentially on the number of dimensions of the data space and most of the volume of a hypercube of (d) dimensions is very close to the $(d-1)$ surface of the cube. Atkin suggests that another way of looking at the dimensionality of the data is to consider the dimensionality of individual objects in the set, excluding variables with a value of zero. Thus, for the calldata set, each telephone customer will no longer be considered to possess 276-dimensional data, but 10- or 15-dimensional data, say, depending upon how many destinations they call. These dimensions are renamed *q-values*. *Q-analysis* involves the examination of how these sets of q-values intersect and the implications for the extent and nature of connection between the objects (Atkin 1981).

Section	Obstacle	Required Facilities
2.2 Types	What constitutes an object? Ordinary objects, but also events, actions etc.	Convey the nature of different kinds of objects.
	Object/attribute interchangeability.	Allow interchange of rows and columns.
2.2.1 Variable Types	Different variable types, some can be transformed into others.	Allow transformation of variable types. Make user aware of different types.
2.2.2 Data Types	Virtual world automatically viewed as 'mirror of the physical world', which may be misleading. How is space used to encode data?	Methods to reveal the data/representation mapping.
	Different types of data: 1D, 2D, 3D, temporal, multiple dimensions, tree, network.	Make user aware of different types.
2.3 Data Structures	Different structures sometimes equivalent.	Allow expression in graph and table forms.
2.4 Base Structure for Complex Data	Many structures derivable from one dataset, especially for complex data, e.g. log. Difficult for user to derive and view these structures.	Allow interaction to form and view different structures. Support for multiple view applications.
2.5 Metadata	Vast amount of metadata available, including resources of the Web, choice and manipulation difficult.	Allow advanced selection, transformation and linking. Support for multiple view applications.
2.6 Selection and Standardization	Visual representation very sensitive to selection and normalization.	Provide facilities to carry these out easily. Make the user aware of the sensitivity.
This section	Intrinsic difficulty of conceptualizing dimensions > 3. Mathematically this space has special characteristics and its behaviour is sometimes unintuitive.	Support comprehension of encoding and transformation. Use other concepts of dimensionality such as in Q-analysis.

Table 2.3: Obstacles to comprehension of data visualization from types, structures, metadata and selection, with facilities required to assist.

2.8 Summary

This chapter began by considering the starting point for visualization, the different types and structures of data. It has shown that the word *type* means two different things in this context, here the terms *attribute* or *variable type* and *data type* are used to make the distinction. Variable types are divided into *qualitative* and *quantitative*, according to whether the data are in categories or are numerical and the result of measurement. Qualitative data are divided into *nominal* and *ordinal*. Data types for information visualization are abstract in the sense that physically based data are excluded; they are divided into temporal, network, tree, 1D, 2D, 3D, and multiple dimensions. Structures can be divided into those based upon a data table and those based upon a graph structure with nodes and edges. Some of these structures are equivalent, for instance, a graph may be expressed as a data table of proximity data.

The need for an underlying base data structure, from which graphs and data tables are considered to derive, has been outlined. A log of events relating to a set of items or entities is suggested as a suitable base structure. Such a structure is useful for the visualization of complex data, particularly that relating to complex systems such as multi-agent systems, and to provide a generic starting point for visualization systems.

The issue of associated metadata has been discussed, noting that there is an ever increasing availability of metadata. The importance of selection and standardization have also been stressed, as they have a high impact upon the resultant graphic and require the development of facilitating interfaces for visualization applications.

Comprehension challenges arising from this discussion of data have been noted and suggested facilities fall into two groups, those needed to provide extra functionality (e.g. to allow linking of metadata) and those to make users aware of important characteristics of the visual representation (such as the data/representation mapping).

Chapter 3

Viewing the Data - Layout

3.1 Introduction

Having examined selection, types and structures of data in the previous chapter, this chapter considers how to create a visual representation of the data on a computer screen. Some of these ideas relate also to the production of visual representations for pages in books and other essentially static artefacts, but the primary focus here is the use of computers, since this medium has significant extra capabilities in the form of interaction and computational power. For instance to change viewpoints in real time in complex 3D structures.

Putting the data onto the screen essentially means finding positions (on the screen) and forms (of the symbol used on the screen) to represent the data in 2D (or simulated 3D). It emerges that there are ways of increasing these 2 (or 3) dimensions available for direct representation, for example by using attributes of the symbol used on the screen to represent the object, such as, say, its colour and shape. The number of these dimensions available for direct mapping of attributes is discussed in this chapter. Also examined are issues concerning practicalities (perception, time complexity, predictability and scalability) and aesthetics.

And so to the actual layout of data on the screen. The term *layout* is used to mean the set of positions on the screen of a set of symbols (often points) that represent the objects in the dataset under consideration. Though the discussion is extended here to include the mapping of dimensions to characteristics such as colour and shape, that do not primarily concern position. This allows the consideration of the direct mapping of higher dimensions, which is extended in the following chapter on visualization morphologies. The chapter concludes with a summary and the noting of areas of comprehension difficulty arising from the layout phase.

3.2 Number of Dimensions Available for Visualization

How many dimensions in the data can be mapped to the visual representation? Starting from the three dimensions provided by linear perspective, 'space and time' is a way of providing four dimensions (use the fourth variable as time). Jacques Bertin in 'La Semiologie Graphique' (1967, translated in 1983) (Bertin 1983) identified eight primary visual variables: size, value, texture, colour, orientation, shape and the two dimensions of the plane. The first six were described as *retinal variables*, the last two as *planar dimensions*. Making a similar distinction, Benedikt (1991) describes dimensions that map to a point as *intrinsic* and those that map to attributes such as colour as *extrinsic*. MacEachren divided *colour* into *hue* and *saturation* and introduced four dynamic variables (MacEachren 1994): order, duration, rate of change, and phase or synchronicity. Ware (2000a) identifies the following categories: spatial position, colour, shape, orientation, surface texture, motion coding and blink coding. These categories give about 17 possible dimensions. Ware does not include all of MacEachren's dynamic variables, so this total could be increased, but, at any one time, not all of these dimensions are available, since some are interdependent (texture - colour, blink - motion). Also the relevance of some visual variables is limited since the number of resolvable steps available, the *granularity* of the visual variable, may be small. Ware considers that eight dimensions are the maximum that can be mapped clearly.

3.3 Layout Issues

This section concerns the various aspects that arise in discussion of layout techniques (see e.g. Herman et al. (2000))[1]:

- perception for design: how must visual representations take account of how we perceive graphics on the screen?
- planarity and aesthetics: how the layout needs to be arranged for a pleasing and effective result?
- time complexity: how long does the layout take?
- predictability; will the layout always be the same for the same data?
- scalability: can a particular layout technique be used for large datasets?
- usability and evaluation: how usable and effective is the layout?

[1] Some of these issues apply equally to the morphology aspect discussed in the next chapter, but they are introduced here for convenience.

3.3.1 Perception for Design

What of the human viewer? The human visual system can be thought of as an extremely wide bandwidth input channel (from the computer system to the human). The challenge is to make use of this and the potential pattern processing power of the brain that lies behind it. In concrete terms there are aims such as simultaneous viewing of large datasets and identification of clusters, relationships and patterns, as well as the finding of a narrow set of items in a large collection. These two areas can be described as *browse* and *known-item search*.

Whilst, in reality, the flat screen only provides us with one or two dimensions for display, we can simulate a third in the usual way, using the geometry of linear perspective. In fact the real world provides many different types of information about 3D space. These are described as depth cues and a considerable body of research concerns itself with how the human visual system processes these depth cues (Ware 2000a, p.274). A general theory of space perception would indicate which cues were most applicable in which situations and how the different depth cues interact. Unfortunately this has not yet been developed and the work is difficult because of the task dependency of space perception.

3.3.2 Planarity and Aesthetics

Where objects in the dataset are represented by objects on the screen and links between the objects shown as connecting lines, that is, a graph structure is being shown, planarity means the absence of crossovers of the connecting lines (Definition 7).

Definition 7 *Planar graph. A* ***planar graph*** *is a graph which can be embedded in the plane in such a way that no two edges intersect geometrically except at a vertex to which they are both incident.*

A system of aesthetics exists which gives a number of requirements that the layout, particularly of graphs, should satisfy such as:

- Planarity of the graph.
- Arranging similar groupings in different parts of the graph in the same way.

In the second category specific requirements are included e.g. certain nodes having to appear on the left, others on the right and so on.

Aesthetics covers the concept of 'what is a good graphic?' Perceptual psychologists, statisticians and graphic designers (see e.g. Bertin (1983), Cleveland (1993), Tufte (1983, 1990)) offer guidance for static presentation of data, but dynamic displays are considered to take us beyond current wisdom (Shneiderman 1996).

3.3.3 Time Complexity

Time complexity is the way in which the number of steps required by an algorithm varies with the size of the problem it is solving. Time complexity is normally expressed as an order of magnitude, e.g. $O(N^2)$ means that if the size of the problem, N, doubles then the algorithm will take four times as many steps to complete.

For systems requiring real-time interaction, visualization updates must be done in very short time intervals so that the user is unaware of any delay. Thus the time complexity of algorithms is important for interaction. Where the product of the visualization process is a single, static visualization, more time will be available to achieve the layout.

3.3.4 Predictability

What is meant by predictability is that two different runs of one algorithm, involving the same or similar graphs or data tables, should not lead to substantially different visual representations. Lack of predictability may or may not be a problem, depending upon the application.

3.3.5 Scalability

Scalability is a measure of how well a solution to some problem will work when the size of the problem increases. For visualization this has a number of aspects:

- How to get so many elements, or their representations, on the screen. This is determined by a number of factors, depending upon the visualization representation used. The obvious limit for any kind of visualization is the resolution of current displays which is currently around one to three million pixels, e.g. for 19inch displays with a resolution of 1024x1280 pixels about 1.3 million pixels.

- Can the layout be determined in real time? Is interaction, to produce query results or new views, feasible? There are also related issues concerning the practicality of searching and storing large quantities of data and the highlighting of entities in such datasets. These issues are also time complexity issues.

- Can the user orientate themselves so as to have an overview as well as being able to focus on detail? Navigation problems are exacerbated as dataset size increases. Techniques to address this problem, described as providing *focus + context* are examined in Section 4.5.1.

searching and storing large quantities, quickly highlighting entities

3.3.6 Usability - Evaluation

The meaning of usability for information visualization depends upon the purpose of the application. It includes attractiveness, meaning, flexibility, navigation and interaction, scalability. Evaluation of usability is acknowledged to be difficult (Herman et al. 2000) and statistics from IEEE Information Visualization conference proceedings papers show that less than 10% evaluated their systems (Robertson 2000).

3.4 Layout Types and Algorithms

For convenience, layout types are examined in two groups: layout of graphs and layout of multivariate data. Figure 3.1 shows an overview of layout methods by these groups. For both categories a degree of transformation may be required and this aspect is included in this description.

3.4.1 Graph Layout and Proximity Data

Graph layout divides into two main areas:

- Where the requirement is for a pleasing and appropriate layout of a structure where the edge weight (Definition 1) is not specified.
- Where the edge weight is specified and is the basis for the position of the node, this corresponds to proximity data (note also that proximity data can be derived from multivariate data).

These areas reflect the level of constraint on the layout as shown in Figure 3.1. The absence of weights for the links in the graph means that there is a free choice of position from the point of view of relative strength of connection (although the layout is still constrained by showing links and planarity requirements). A fully connected, weighted graph corresponds to proximity data.

Trees, spanning trees, minimal spanning trees, hierarchical clustering, force directed systems, hyperbolic layout and Principle Coordinates Analysis (PCoA) (i.e. all the methods shown in the *graph* box in Figure 3.1) are described in the following sections. Briefly they are as follows:

- Trees. Simpler to visualize in one's mind, hierarchical information in the form of trees are also simplest to lay out.
- Spanning trees can be used to reduce the number of links that are shown and make it easier to provide a layout for weighted or unweighted connected graphs.
- Minimal spanning trees provide a layout based upon minimizing the sum of the weights of a spanning tree.

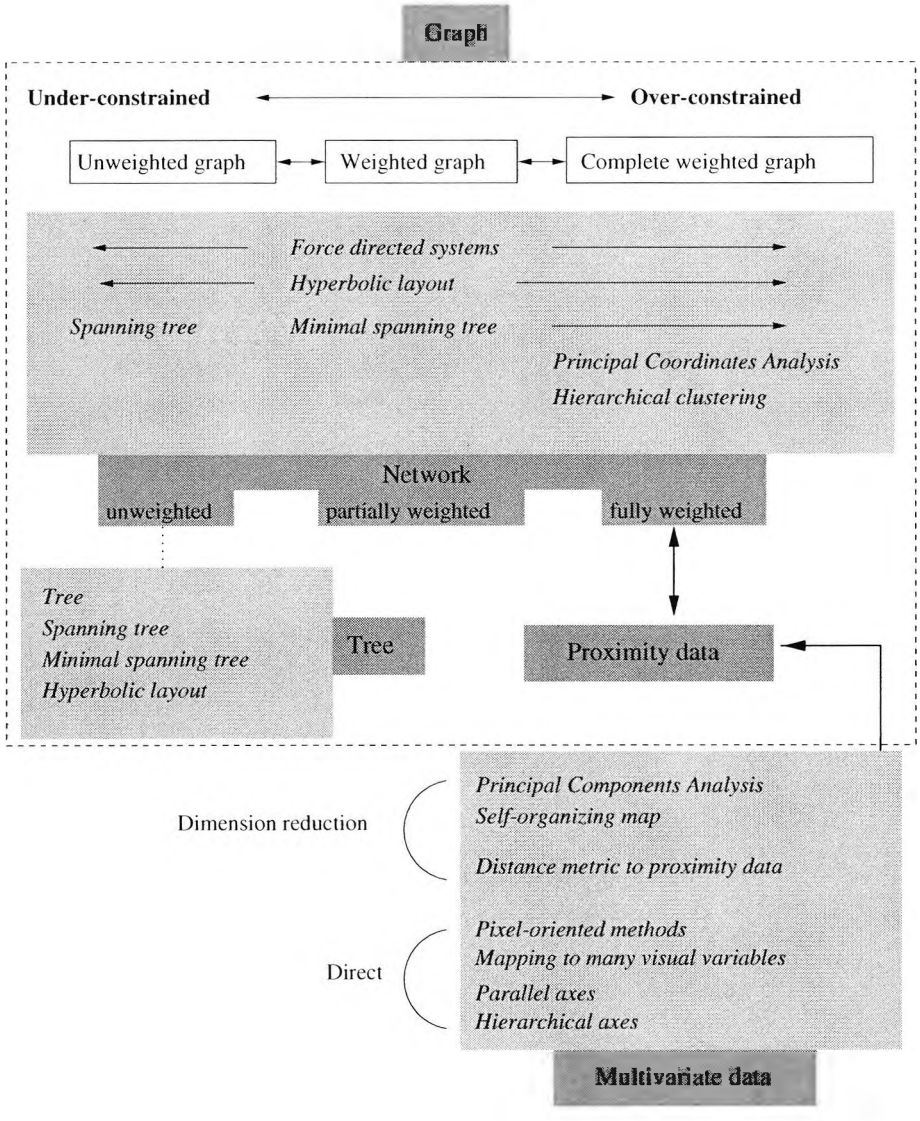


Figure 3.1: Overview of layout methods by data type. The two main boxes show methods for 'graph' layout (within the dotted line) and 'multivariate data' layout. Multivariate data layout methods divide into those involving dimensions reduction and those directly mapping the variables to the graphic. Graph layout is constrained by the presence or absence of weights for the edges, and by its completeness. This corresponds to layout for networks. Trees lie at the incomplete end of the graph constraints. Proximity data derived from multivariate data correspond to a complete weighted graph.

- Hierarchical clustering generates tree structures from proximity data.
- Hyperbolic layout is a special layout that allows large hierarchies and networks to be viewed with a detailed focal area. This is an example of a special method of distorting the layout which allows greater numbers of objects to appear on the screen while maintaining a detailed focal area (focus + context methods, see Section 4.5.1).
- Force directed systems and PCoA are methods of finding layouts for proximity data. Force directed systems are heuristics based on a physical analogy, PCoA involves matrix transformation: both approaches give a 1, 2 or 3 dimensional layout that satisfies or approximates the proximity data. The class of techniques that analyze a matrix of distances or dissimilarities in order to produce a representation of the data points in a reduced-dimension space is described as *multidimensional scaling* (Kruskal 1977; Webb 1999). Minimal spanning trees can also be used, although it is sometimes suggested that this is then not referred to as multidimensional scaling, because links are shown (Chen 1999, p.45). The first stage of the matrix transformation, PCoA, of a dissimilarity matrix, results in a pattern matrix which can be processed from scratch (although this may not always be valid (Kruskal 1977)).

Tree

In one sense this is the easiest to represent, as there is a clear way to proceed - start at the root (for rooted trees) and lay each level of the tree out beneath. These layout types usually have the lowest time complexity which is $O(N)$, where N is the number of nodes. Although, as with any type of structure, there are problems in laying out and navigating trees of great size. The classical layout algorithms are all predictable. An example is the layout shown in Figure 4.14 using the Reingold and Tilford algorithm.

Spanning Trees

Definition 8 *Spanning tree. If G is a connected graph then a **spanning tree** in G is a connected spanning subgraph (Definition 9), containing no circuits.*

Definition 9 *Subgraph. A **subgraph** of a graph $G = (V(G), E(G))$ is a graph $H = (V(H). E(H))$ such that $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. If $V(H) = V(G)$ then H is called a **spanning subgraph**.*

Since tree layout algorithms have the lowest complexity and are simplest to implement, the tree structure can be used to deal with a general connected graph. A spanning tree is extracted and additional edges added. One approach for edge extraction visits nodes of the graph in a breadth first

search and collects edges to form a tree. The choice of the starting node (which becomes the 'root' of the tree), as one whose distance to all other nodes is minimal, can improve the result. Using a spanning tree layout can also gain predictability of the layout (Herman et al. 2000). Spanning trees are also used to solve the problem of too many edges in a graph layout. In general the problem of overcrowding of links can be avoided by not showing any and relying on spatial configuration to show relationships - or redundant links can be removed in advance by algorithms such as Pathfinder Network Scaling (Chen 1999, p.45) and minimal spanning trees (see next section).

Minimal Spanning Trees

A graph often contains redundancy in that there can be multiple paths between two vertices. This redundancy may be desirable, for example to offer alternative routes in the case of breakdown or overloading of an edge (road, connection, phone line) in a network. However, we often require the 'cheapest' sub-network that connects the vertices of a given graph. The total cost or weight of a tree is the sum of the weights of the edges in the tree. A minimum spanning tree of a weighted graph G is the spanning tree (Definition 8) of G whose edges sum to minimum weight. There can be more than one minimum spanning tree in a graph, (consider a graph with identical weight edges). Thus spanning trees can be considered as a method for reducing the number of links shown in graph layout, or as an alternative method of dealing with proximity data.

Hierarchical Clustering

Hierarchical clustering procedures are often used for summarising data structure. The result is a nested set of partitions represented by a special tree diagram or *dendrogram*. A hierarchical tree structure can be formed from a minimum spanning tree (corresponding to single-link hierarchical clustering (Webb 1999, p.278)).

Force-directed Systems

Force directed algorithms (di Battista et al. 1999, chapter 10) draw graphs using a physical analogy. The graph is viewed as a system of bodies with forces acting upon the bodies. A position is sought such that the sum of forces on each body is zero. An example is *spring embedding* (Quinn and Breuer 1979), which considers nodes as mutually repulsive charges and the edges as springs that attract connected nodes. When the spring energy in the entire system reaches the global minimum a solution has been reached. However, the system can be stopped at any point and the layout may be acceptable at the stage reached.

Hyperbolic Layout

The hyperbolic layout (Lamm et al. 1996) provides graph visualization which allows large hierarchies and networks to be viewed while maintaining a detailed focus. Mainly used for trees, hyperbolic layout produces a distorted view of a tree resembling fish-eye distortion (described in the next section). Three examples of implementations are shown in Figure 4.21 on page 62. An important feature of hyperbolic geometry is that the length of a line segment is defined as a function of the position of the points with respect to the perimeter of a containing disc and so segments become exponentially smaller when approaching the perimeter.

Principal Coordinates Analysis

Principal Coordinates Analysis (PCoA), starting with a dissimilarity matrix, first finds a pattern matrix which satisfies the distances, then transforms this into its principal components so that the two or three most important factors can be displayed in 2D or 3D space (Gordon 1999, p. 149) in a similar fashion to Principal Components Analysis (see next section). PCoA is sometimes referred to as 'classical scaling', as it was originally known.

3.4.2 Multivariate Data

There are four main groups of methods for viewing multivariate data, excluding the selection of subgroups of data (e.g. taking selections of 2 or 3 dimensions to view directly in a scatterplot). These are:

- The matrix can be viewed directly as a colour map. Colour maps and other such methods, described as *pixel-oriented* visualization techniques, are illustrated in the next chapter, in Section 4.3.1.
- Selections of attributes can be represented by using the three Euclidean dimensions in addition to time and mappings to colour, shape etc., as described in Section 3.2 and illustrated in Section 4.3.
- Special visualization forms have been developed to handle the direct display of multiple attributes, these include hierarchical axes (Mihalisin et al. 1991) and parallel axes (Inselberg 1997), illustrated in Section 4.3.3.
- Dimension reduction can be achieved directly (by e.g., Principal Component Analysis or Self Organizing Map) or indirectly (by first deriving a proximity matrix based on a defined distance metric). Principal Components Analysis, Self Organizing Maps and distance metrics are described below.

The first three groups of methods are *direct* in the sense that the data values in the dataset map directly to the visualization, without being transformed in a dimension reduction process. Thus, the methods for multivariate data divide into *direct* and *dimension reduction* methods as indicated in Figure 3.1. Only the dimension reduction methods are described in this chapter, because these methods give rise to positions on the screen for the objects. The direct methods are illustrated in the next chapter as visualization morphologies. The aspects of layout and form cannot be considered entirely separately since a form is required in order to embody a layout. However, they are considered separately here for convenience, to simplify the presentation of many methods.

Principal Components Analysis

Principal Components Analysis (PCA) is a means by which a multivariate data table, giving attribute values for a set of entities, may be transformed into a table of factor values, the factors being ordered by importance. The two or three most important factors, the principal components, can then be displayed in 2D or 3D space. A mathematical technique, Singular Value Decomposition (SVD), is used for the matrix transformation and PCA is sometimes known by this term (Chen 1999, p. 30). The process can be conveniently described as replacing the original matrix with a truncated SVD matrix. In the area of information retrieval systems PCA is known as Latent Semantic Indexing (LSI) (Chen 1999). In LSI the multivariate data table is a large term-by-document matrix in which each element is the number of occurrences of a term in a document.

Self Organizing Map

The self organizing map algorithm (Kohonen 1997) is an unsupervised neural net that can be used as a method of dimension reduction for visualization. It automatically organizes entities onto a two-dimensional grid so that related entities appear close to each other. Although it is described as an unsupervised neural net, there are similarities between the self-organizing map method and other data analytic methods such as k-means type clustering (Gordon 1999, p. 170).

Distance Metrics and the Derivation of Proximity Matrices

The multivariate data table can be considered as a vector space and different measures used to define a similarity between the vectors. There are a variety of measures (see e.g. Gordon 1999; Webb 1999) including those based on the Euclidean distance and the angle between vectors described below. Measures are usually presented for comparing objects that are described by a single type of variable, though these can be combined so that measures for data containing variables of different types can be constructed. This discussion restricts itself to quantitative data type for brevity.

Let x_{ik} denote the value that the kth quantitative variable takes for the ith object ($i = 1, \ldots, n; \quad k =$

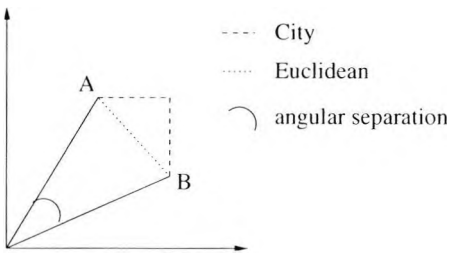


Figure 3.2: Distance measure examples for two-dimensional data. Three ways of measuring the 'difference' between entities A and B: Euclidean distance, City distance and angular separation.

$1, \ldots, p$). The Minkowski metric defines a family of dissimilarity measures, indexed by the parameter λ.

Minkowski distance

$$d_{ij} = (\sum_{k=1}^{p} w_k^{\lambda} |x_{ik} - x_{jk}|^{\lambda})^{1/\lambda} \quad (\lambda \geq 1) \tag{3.1}$$

where $w_i (k = 1, \ldots, p)$ are non-negative weights associated with the variables, allowing standardization and weighting of the original variables. Values of λ of 1 and 2 give the two commonly used metrics of this family: *City-block* and *Euclidean*. These are illustrated in Figure 3.2 for two-dimensional data.

City-block distance, also known as the *Manhattan* or *box-car* or *absolute value* distance, would be suitable for finding the distances between points in a city consisting of a grid of intersecting roads, hence its name. It is less complicated to compute than the Euclidean distance, but yields similar results and therefore may be used instead of Euclidean distance if speed is an issue.

City-block distance

$$d_{ij} = \sum_{k=1}^{p} w_k |x_{ik} - x_{jk}| \tag{3.2}$$

The Euclidean distance has the property of giving greater emphasis to larger differences on a single variable.

Euclidean distance

$$d_{ij} = (\sum_{k=1}^{p} w_k^2 (x_{ik} - x_{jk})^2)^{1/2} \tag{3.3}$$

The choice of an appropriate value for the parameter λ of the Minkowski metric depends on the amount of emphasis you would like to give the larger differences. As λ tends to infinity, the metric tends to the *Chebyshev* or *maximum value* distance. This measure is also used in time critical situations.

Chebyshev distance

$$d_{ij} = max_{k=1}^{p} w_k |x_{ik} - x_{jk}| \tag{3.4}$$

These measures can be standardized, for instance so that d_{ij} is bounded by 1. If $w_k = (p\mathcal{R}_k)^{-1}$, where $\mathcal{R}_k$ denotes the range of values taken by the kth variable. One could also consider $w_k = p(\max_{k=1}^{p}(\mathcal{R}_k))$, which preserves the quantitative comparison between objects. Also consider not the range, but (assuming it is relevant to consider the possible minimum value then take $w_k = p(\max_{i=1}^{n}(x_{ik}))$ or $w_k = p(\max_{k=1}^{p}(\max_{i=1}^{n}(x_{ik}))$. For example:-

	Destination1	Destination2	Destination3
customer1	5	1	3
customer2	4	1	5

Without weighting this gives 1.29, with weighting using the range: 0.816, using the maximum value: 0.086.

Sometimes it is the relative magnitudes of the different variables that is of interest - the behaviour across the variables rather than the absolute values. Put another way, the variables describing the object define a vector with p components and interest is in the comparison of the directions of the vectors. In the following metric the cosine of the angle between the vectors is used, as illustrated in Figure 3.2. Since values are between -1 and 1, the measure can be transformed to take values between 0 and 1 by defining $s'_{ij} = (1 + s_{ij})/2$.

Angular separation

$$s_{ij} = \frac{\sum_{k=1}^{p} x_{ik}x_{jk}}{(\sum_{k=1}^{p} x_{ik}^2 \sum_{l=1}^{p} x_{jl}^2)^{1/2}} \tag{3.5}$$

For the previous example this metric gives a value for s' of 0.0465.

Standardization is used because variables often show different variability across the dataset. Standardization is generally from the point of view of a particular dataset. For example, for quantitative variables, each variable value can be divided by the range of values they take in the dataset, as indicated in the discussion of weights above. One detailed study (Milligan and Cooper 1988) indicated that dividing by the range outperformed other methods investigated. However, as with many issues relating to suitability of methods for datasets, the fact that something works well for many datasets does not indicate that it will for a particular dataset under consideration. Standardization can be considered to be subsumed by the general issue of weighting. The choice of variables to include can also be seen as an extreme case of weighting, where a zero weight is applied.

Whatever choice of metric is applied to calculate pairwise distances for all pairs of objects in the set, the result is a proximity matrix which can be treated as in any of the methods described in the previous section.

The choice of metric depends on the application, and the literature indicates the difficulty, yet desirability, of choosing appropriately. According to Webb (1999):

> "It is not possible to make recommendations, and studies in this area have been largely empirical, but the method you choose should be one that you believe will capture the essential differences between objects."

To illustrate the different distance metrics and the variety of visual depictions their use results in, Figure 3.3 shows the calldata set (introduced in Section 1.2 on page 6) after different methods of dimension reduction. These pictures highlight the difficulty of deciding upon an appropriate dimension reduction method (Schroeder and Noy 2001).

3.5 Comprehension Challenges

This section notes specific points from the discussion above that introduce difficulty in comprehension, as did the corresponding section in the previous chapter. The difficulties can be summarized as arising from the different ways that the same set of data can be represented, from special features of layout choices, from high levels of abstraction and from the impact of layout choice upon interactivity. These aspects are expanded in the next paragraph and shown by section in Table 3.1.

The different ways that data can be represented arise from the choice afforded for the mapping from data to visual variables, any lack of predictability, scalability (different ways of dealing with it) and the different layout algorithms available. Special features of the visual representation include the impact of the different ways of mapping from data to the different visual variables, of the impact of depth cues, scaling and dimension reduction. One of these features is of particular importance, the introduction of high levels of abstraction. Abstraction may be increased in dealing with time complexity, for instance by using a simpler distance measure, or introduced for scaling (reducing the number of objects) or dimension reduction (reducing the number of variables). Anything which reduces interactivity has the potential to reduce comprehension since it is by the interactive process that much understanding is gained. One can consider interaction on two levels here, the interaction overall, to examine different views and selections (of variables, of methods) of data, and interaction within a specific graphic, which does not produce a different display type or selection of the data. Paradoxically, a more pleasing layout can have the impact of reducing interaction on the selection level, since the user is not prompted to change the representation to get a better view. At the same time a specific graphic can increase interactivity by providing interaction features that are inviting.

3.6 Summary

The dimensions available for direct mapping from the dataset cover the following *visual variables*: spatial position, colour, shape, orientation, surface texture, motion coding and blink coding. This

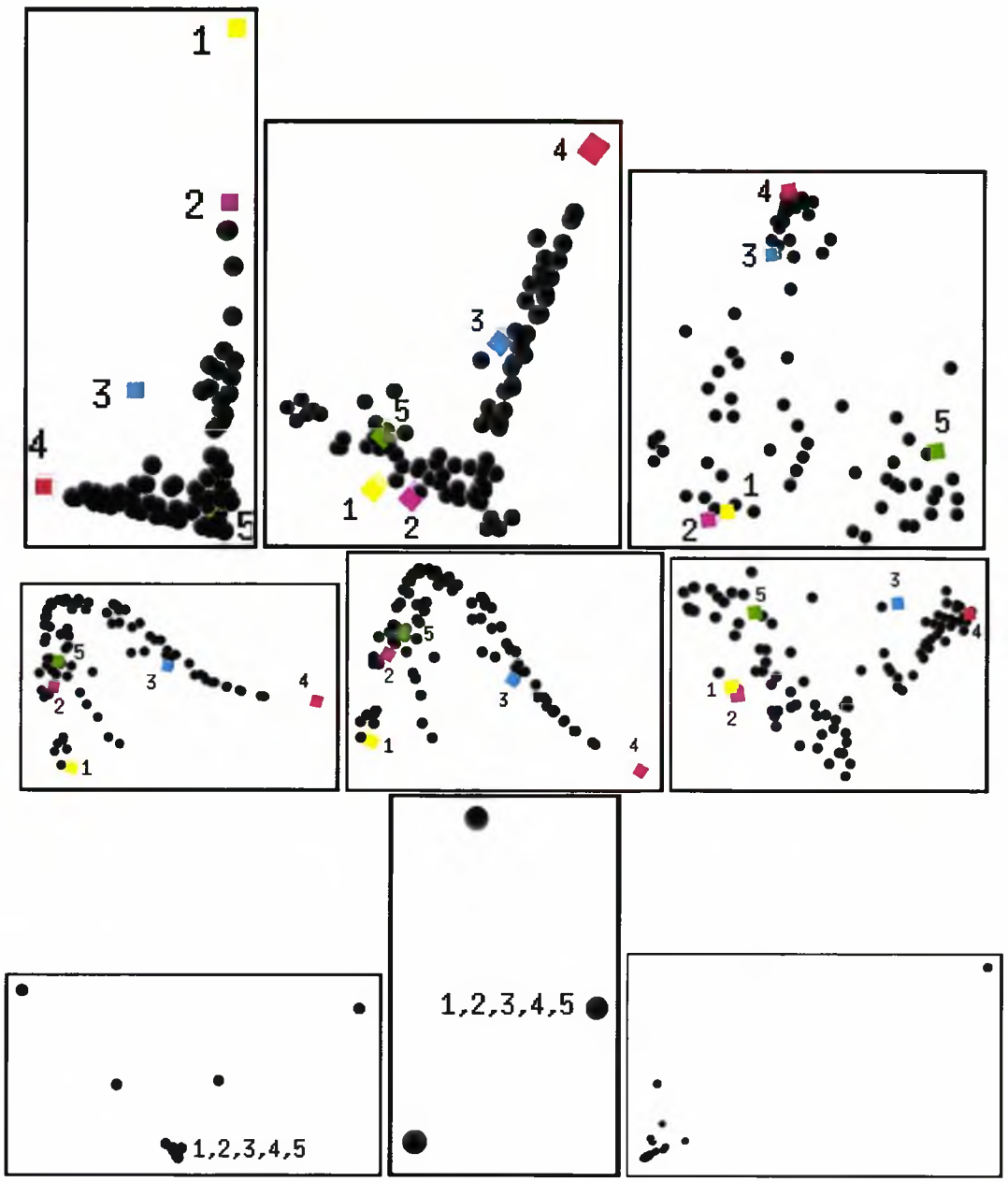


Figure 3.3: 90,000 calls made by 100 customers. Pictures are numbered (1) to (9) from top left to bottom right in horizontal rows. First eight pictures: caller profiles by destination of calls. Last picture: destination profiles by callers. (1) Direct visualisation by applying principal component analysis, in contrast to (2-8) indirect visualisation by calculation of a distance followed by matrix transformation using Singular Value Decomposition; Distances in particular: (2) Euclidean, (3) correlation, (4-8) Minkowski distance with (4) $\lambda = 100$ (5) $\lambda = 10$ (6) $\lambda = 1$ (City distance) (7) $\lambda = 0.1$ (8) $\lambda = 0.01$. The callers represented as squares and labelled with a number can be traced through the different visualisation outcomes (1) to (8), to see how their placement differs (Schroeder and Noy 2001).

Section	Obstacle	Required Facilities
3.2 Number of Dimensions Available for Visualization	Different possibilities, how to choose. Designer can have expertise, but cannot assume the user has.	Suggest appropriate choices, provide guidance.
	Different views possible.	Increase interactivity to make user aware.
3.3.1 Perception	Depth cues can help/hinder.	Suggest suitable forms.
3.3.2 Aesthetics	Pleasing results can both increase and decrease interactivity.	Retain types that promote interaction.
3.3.3 Time Complexity	Time constraints, including for real-time interaction, means greater use of approximation.	Make user aware of higher approximation.
3.3.4 Predictability	Lack of predictability leads to different layouts for same data.	Make user aware of different layout possibilities. Allow saving or 'bookmarking' of layouts.
3.3.5 Scalability	Larger amounts of data can be clustered and clusters shown as points.	Make user aware of clustering method and its features.
	Different methods of clustering.	Make user aware of different clustering possibilities.
3.4 Layout Types and Algorithms	Different possibilities for the same dataset.	Make user aware that there are different possibilities.
	Mathematical transformation in dimension reduction is an abstraction.	Abstraction level indicators desirable.
	How to choose the metric - guidelines difficult?	Increase interactivity so user can easily experiment.

Table 3.1: Obstacles to comprehension of data visualization from layout issues and types, with facilities required to assist.

provides a total of about seventeen, though far fewer can be of practical use at any one time. There are a number of issues to be taken into account when designing visualization applications. These include *perception*, *planarity*, *aesthetics*, *time complexity*, *predictability*, *scalability* and *usability*.

Layout types can be divided into those for the layout of graphs (which includes proximity data) and those for multivariate data. *Trees* are relatively easy to represent, have low time complexity and are predictable, though scaling is a problem. *Spanning trees* can be derived from general connected graphs to make layout easier. *Minimal spanning trees* may be required for intrinsic reasons, or as an alternative method of dealing with proximity data or to form the basis of hierarchical clustering. *Hierarchical clustering* is used for summarizing data structure. *Force-directed systems* are used for proximity data and are based upon the physical analogy of a system of bodies with forces acting upon them. An example is *spring embedding*. *Hyperbolic layout* allows large hierarchies to be viewed while maintaining a detailed focus. *Principle Coordinates Analysis* transforms proximity data into a pattern matrix, which can then be truncated to display only the most important factors.

Multivariate data can be viewed directly by special displays that map more than three variables or by the use of dimension reduction. Special displays include *colour maps*, *hierarchical axes*, *parallel axes* and the mapping of dimensions to *visual variables other than position*. Dimension reduction methods include *Principal Components Analysis*, *Self-organizing maps* and the derivation of proximity matrices by the use of *distance measures*. Principal Components Analysis first transforms the matrix via Singular Value Decomposition, which allows the matrix to be truncated and only the most important factors displayed. The self-organizing map algorithm is an unsupervised neural net that organizes objects into a 2D grid on the basis of their multivariate data attributes. Multivariate data can be considered as a vector space and distances between pairs of vectors calculated. This produces a proximity matrix which can then be displayed. *Euclidean*, *City-block*, *Chebyshev* and *angular separation* are commonly used distance measures.

The layout of data provides challenges to comprehension from the different ways the same data can be represented, the special characteristics of layout methods, the high levels of abstraction sometimes involved and the impact of layout choice upon interactivity.

Chapter 4

Viewing the Data - Morphologies

4.1 Introduction

This chapter presents visualizations according to their appearance. As indicated in the previous two chapters, visualizations can be grouped in various ways according to underlying dimensionality of the data, structure of the data (or derived structure), layout algorithm, dimensionality of representation etc. Chi (2000) provides an overview of information visualization taxonomies. Most of these use a data-centric point of view for classifying techniques. Shneiderman (1996) at first proposed seven types: 1-, 2-, 3- dimensional data, temporal and multi-dimensional data, and tree and network data (as described in Section 2.2.2). This has been extended in the OLIVE (On-line Library of Information Visualization Environments http://otal.umd.edu/Olive/) taxonomy into eight types by including *workspace* as a category. However, here these aspects of classification are set to one side, as a more general view, based upon multivariate and proximity data forms, continues the discussion of the previous two chapters. Interaction techniques also result in particular visualization morphologies[1]. Thus a range of visualization forms are described and illustrated, loosely grouped as:

- direct data table mapping (with or without dimension reduction) - here described as matrix view
- tree representations
- forms addressing the size issue with regard to navigation and interaction

The normal well-known 2D forms such as bar charts and graphs are not included. These are very powerful visualizations in their own right as is demonstrated by their extensive use. Some of the

[1] Morphology is a branch of biology that deals with form and structure without consideration of function. The word is used here in a similar way to focus upon the structure and form of visual representations, rather than visualization techniques as such.

visualizations described here are extensions of them. The chapter includes a section on the challenges to comprehension raised by navigating and interacting with these morphologies.

4.2 Morphology Issues

In discussing visualization morphologies, the issues described in the previous chapter concerning layout (Section 3.3) apply and expand in scope. Thus, the issues *perception for design*, *planarity and aesthetics*, *time complexity*, *predictability*, *scalability* (number of data items - both objects and dimensions), and *usability and evaluation* now expand to include:

- use of colour
- 2D versus 3D
- ordering of the variables (though this is an issue directly concerning the layout, it becomes more relevant in the discussion of particular forms)
- navigation and interaction

Navigation and interaction are discussed in Section 4.5 after the sections describing morphologies. The use of colour, 2D versus 3D and variable ordering are considered in the next three sections. Relevant aspects from the overall list of issues are included when considering the strengths and weaknesses of the visualization morphologies in the subsequent sections. In as much as these strengths and weaknesses are known. For a number of reasons it is currently rarely possible to say 'This is the best way of visualizing these data': some visualization forms are relatively new; evaluation is difficult and highly task and domain dependent; different methods may present or emphasize different (legitimate) aspects of the data. Thus, illustrations of the 'rich palette of available techniques' (Spence 2001, p. 69) are presented here (Sections 4.3 and 4.4), with indications of the characteristics of each individual technique, rather than a strict classification.

4.2.1 Use of Colour

A set of numerical values can be mapped to a colour or grey scale. The advantage of colour scales over grey scales is that the number of just noticeable differences (JNDs) is larger (Ware 2000a). Maximizing the number of JNDs and finding a resulting scale that is intuitive for the domain are the main tasks in choosing a path through the colour space (Herman and Levkowitz (1992) as cited in Keim and Kriegel (1994)). In their experiments with different colour mappings, Keim and Kriegel (1994), found that the coloration had a high impact on the intuitivity of the system. The user sometimes connected good answers with light colours and bad answers with dark colours, or was accustomed to

green colours for good answers and red colours for bad answers. Thus colour has a different effect according to the sequence of colours chosen. More detail on this topic can be found in Ware (2000a).

The way we perceive colours is influenced by many factors. For instance, small patches of light give different results from those of large patches. In general, we are much more sensitive to differences between large patches of colour. People also vary in how they perceive colour, with about 10% of the male population and about 1% of the female population suffering from some form of colour vision deficiency.

Colour is very good for nominal information coding, which requires a non ordered, easily recognizable code giving a set of classes. The number of classes cannot be large, since only a small number (estimates vary between 5 and 10) of these colours can be rapidly perceived.

Sequences of colours used for representing continuously varying map values are widely used, though Bertin (1983) considered that colour was best used to symbolize quantitative differences. Geographers use sequences to display height above sea level: lowlands are green, corresponding to vegetation, and the scale moves up through brown and then to white for the peaks of mountains. The most common coding scheme used by physicists approximates to the physical spectrum. However, it is not a perceptual scale. If users were given a set of blocks painted in corresponding colours and asked to place them in order, they would be unable to do so. If the blocks were shades of grey, they would be able to do this. However, many perceptually orderable colour sequences are possible, obtained by using less hues.

4.2.2 2D Versus 3D

Two-dimensional visual structures are the most common, but this is because three dimensional work has been limited, until recently, to expensive computers used for specialist areas such as in exploratory research or movie production. 3D visualization design poses additional problems because of the six degrees of freedom of movement afforded the user, and the relevance of factors such as shading, lighting and the role of the ground plane (in enhancing the user's perception of 3D). Occlusion is also a problem, for instance foreground objects can hide distant objects. Difference in perception of depth in comparison to height and width is also evident. An ongoing problem is how to render text in 3D, since abstract data often involves textual values. Three dimensional fonts often have to be larger to be readable. It is an open question under what conditions 3D is better than 2D for information visualization (Card et al. 1999).

4.2.3 Ordering

An important property of some axes, usually nominal, is that they can be permuted, thus producing new visual patterns. In the case of a pattern matrix, both the attributes and the objects may be

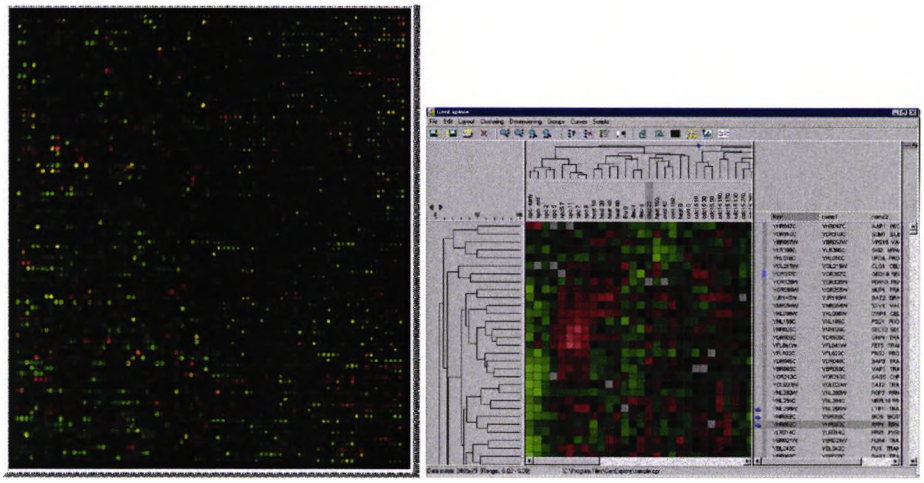


Figure 4.1: Gene expression data examples: deRisi et al. (1997) - full yeast genome (left); sample data matrix (right) using the GeneMaths application, created by Applied Maths (http://www.applied-maths.com/). This is micro-array gene expression data described in Eisen et al. (1998) and available at www.pnas.org.

reorderable. Some visual representations are dependent upon the initial (arbitrary) ordering.

4.3 Matrix View

It is not possible to describe all methods for multivariate data here. Notable omissions are *hierarchical axes*, *hyperslice* and *dimensional stacking* details of which, with others, may be found in the survey paper Wong and Bergeron (1997).

4.3.1 Colour Maps

One step from the showing of a data table as an actual alphanumeric table is to replace the values of variables by colours. The result is as if one is looking at the table, but with the values replaced by colours. Users instantly form a general impression of the way values are distributed. However, the result is highly dependent upon the ordering of both the objects and the variables. Examples are shown in Figure 4.1. More complex colour maps can be formed using spiral and other arrangements of attributes or to show the relevance of items to a database query. More generally these are described as pixel-oriented visualization techniques (Keim 2000).

4.3.2 Mapping Onto Objects

Another system for mapping the data table directly is shown in Figure 4.2. Each data item, with its associated data, is mapped onto the surface of an object. In this example, the *Perspective Wall*

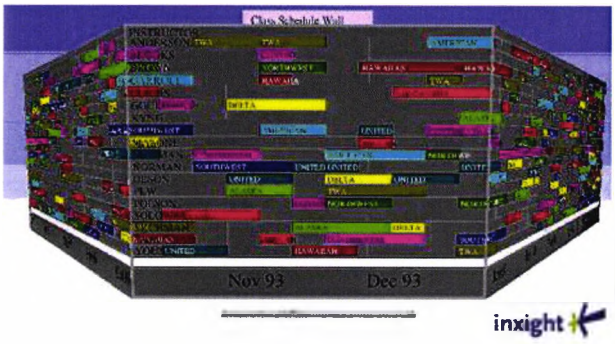


Figure 4.2: A perspective wall using Xerox's Information Visualizer (http://www.parc.xerox.com/istl/projects/uir/projects/InformationVisualization.html).

(Mackinlay et al. 1991), data is mapped onto a wall. Sections are viewed one at a time and the neighbouring sections moved to by the wall smoothly rotating. This facilitates the browsing of a larger number of data items than can be contained by a 2D representation (at least a three-fold improvement) and provides an area of focus, whilst showing the neighbouring context (see Section 4.5.1 for more discussion of focus-and-context methods). The user can adjust the ratio of detail and context.

4.3.3 Parallel Coordinate Plots

If the axes of multidimensional space are arranged parallel to each other, they are described as parallel coordinates and a view of data presented on these axes as a *parallel coordinate plot* (Inselberg 1997). The axes are organized as uniformly spaced lines, either horizontally or vertically. In this way a data element in an n-dimensional space is mapped to a polyline that traverses across all of the horizontal or vertical axes. An example is shown in Figure 4.3. General trends can be seen, even if many graphs are plotted at the same time. However, the ordering of the attributes is important, whilst often the ordering of attributes is arbitrary. The ordering is important because it affects one's ability to interpret the plots, since different orderings can lead to more or less *overwriting* of graphs.

4.3.4 Glyphs

A glyph (Ribarsky et al. 1994) is a graphical object designed to convey multiple data values represented by colour, shape, movement and so on, as described in the section on available visual variables in the previous chapter, Section 3.2. Figure 4.4 visualizes a large amount of data collected about web sites. The attributes size, shape, etc., reflect such things as number of pages, number of links to and from each site. For instance, the *height* of the glyph reflects how many links go to it, i.e. how 'visible' it is to the outside world. The number of attributes that can be mapped to a glyph is limited (though greater than 3, see Section 3.2) and the resulting representation is highly dependent upon the

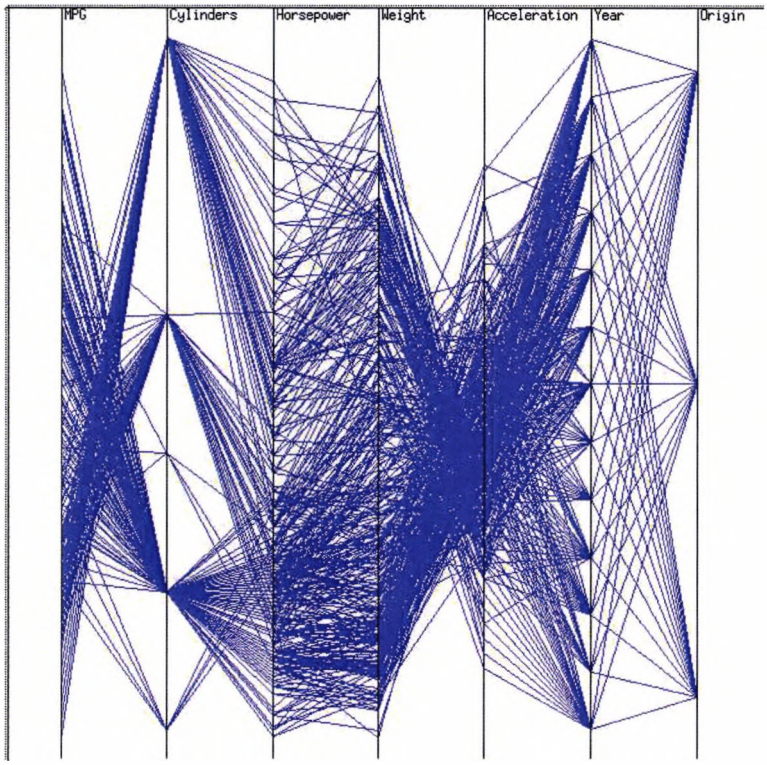


Figure 4.3: Parallel coordinate plot of car data. Miles per gallon of 38 1978-79 model cars is given, together with data such as weight, number of cylinders etc.

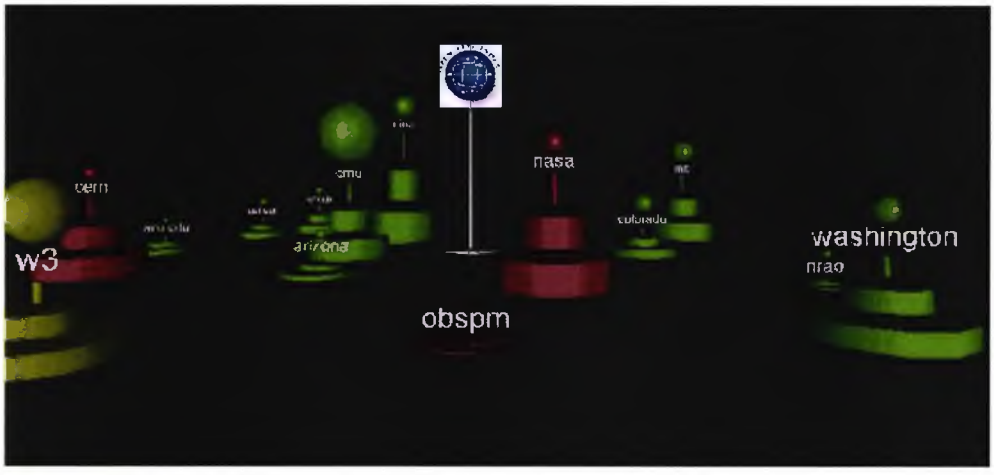


Figure 4.4: Website data (Bray 1996).

choice of shapes, colours etc., so that one data table can be mapped to many different glyph worlds. The positioning of the glyphs in the plane can be arbitrary, a customized ordering based upon domain considerations, or can be the result of a distance measurement based on all or a subset of the attributes (as described in Section 3.4.2).

4.3.5 Star Plots

A star is a variation on the general glyph, where each data value is represented by a line segment radiating out from a central point. An example star plot (Fienberg 1979) is shown in Figure 4.5. These produce a texture field when large numbers are displayed. Without the ends of the line segments being joined up, the glyph is called a *whisker*, whence *whisker plot*. Star and whisker plots succeed in displaying high-dimensional data without dimension reduction. However, the order of attributes has an impact on the resulting overall shape and thus on how the data are presented. Starplots are also difficult to compare to each other as it is difficult to quantify the differences.

4.3.6 Information Landscape

An information landscape uses 2D to define a plane and the third dimension is reserved for specific data to be visualized. With the plane as a basis of the visualisation, users can fly over the landscape and observe the data in the third dimension. The website data visualization of Figure 4.4 is a general information landscape. A tree structure (see also Section 4.4.7) is shown in Figure 4.19. The surface plot and cityscape, below, are specific types of information landscape. Such landscapes can present large quantities of data, but suffer from the general problems of 3D, particularly relating to navigation controls.

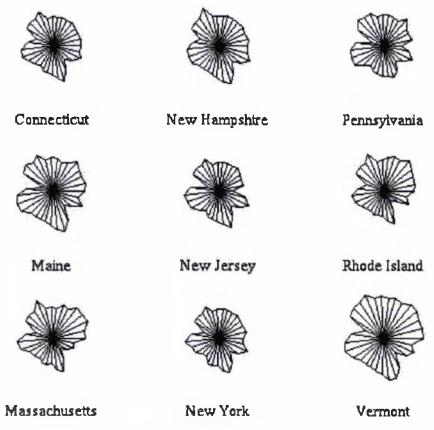


Figure 4.5: A star plot (Fienberg 1979).

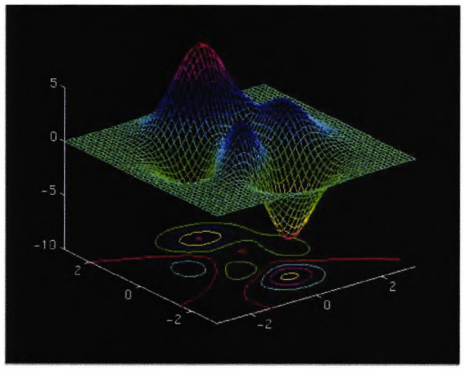


Figure 4.6: A surface plot using Matlab (http://usg.cas.utk.edu/public/matlab/mesh.html).

4.3.7 Surface Plot

A direct extension of the familiar 2D graph, surface plots are constructed by plotting data triples onto the three co-ordinate axes directly. The points are then netted into a surface or mesh as in Figure 4.6. Trends and irregularities can easily be seen, but the number of attributes mappable is limited to 3. Colour can be used redundantly to emphasize the shape, or to encode a further attribute.

4.3.8 Cityscape

A vision of a modern city with skyscrapers extends the bar chart into 3D. Cityscapes are similar to surface plots, but with 3D bars rather than a surface. In Figure 4.7 a fourth dimension is provided by use of colour. As with surface plots, trends and irregularities can be easily seen, though discontinuities can make occlusion more of a problem and the number of attributes mappable is limited to 4.

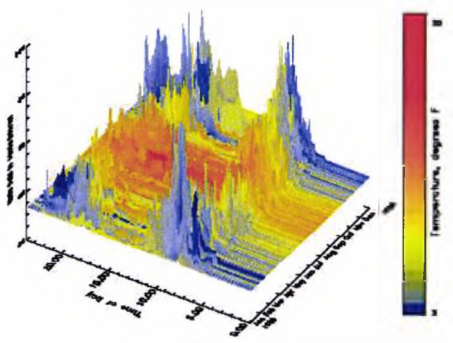


Figure 4.7: A Cityscape from Visualnumerics' JWave system providing four mapping dimensions with the use of colour (http://jwave.vni.com/classes/jwave_demos/bi/Coffee.html).

4.3.9 Scatterplot

The scatterplot is the basis of many current visualizations which then add colour or glyphs. In its 3D form, this is a visualization form that enters the spatial metaphor in the guise of 'galaxies'. A scatterplot makes use of the extrinsic visualisation approach in that it maps data objects to a point such that the distances between objects reflect their relationships. 3D examples are shown in Figure 4.8. In extending to three dimensions the scatterplot suffers from the general problems of 3D. Labelling in large scatterplots is also difficult. Overplotting, where many objects occupy the same location, is also a problem, though colouring the density of co-located points can be used to highlight this issue (see e.g. Haslett et al. (1990)). High dimensional data can be visualized with a scatterplot if dimension reduction is first undertaken. However, difficulties are then created in interpreting the data which has been transformed by the dimension reduction process, involving information loss. A matrix of scatterplots, presenting multiple adjacent scatterplots, can be used in order to show plots of all pairs of variables. The aim is to visually link features in one part of the matrix with features in others, though this may be difficult. Overall, scatterplots are simple and familiar to users, can give a good overview and depict basic structure, but may also deceive users who are unaware of characteristics such as overplotting and dimension reduction abstraction error (unless there is provision to show this information).

4.3.10 Daisy Chart

The Daisy (Data Analysis Interactively) chart shows data as a circular figure with attributes arranged around the circle. The data items in an n-dimensional space map to polylines connecting the corresponding points around the circumference as in Figure 4.9. A Daisy Chart is designed to show the maximum amount of information about a database in a single chart. The chart is designed to be

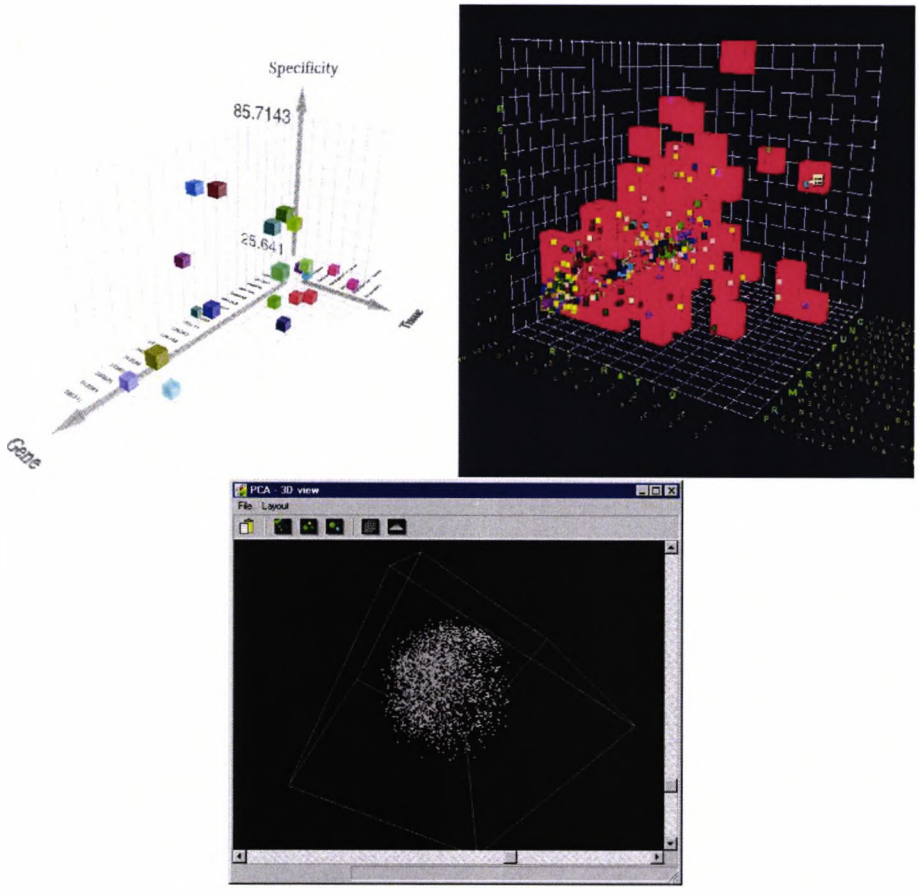


Figure 4.8: Scatterplots. Clockwise from top left: the first is created by Silicon Graphics' Mineset (http://www.sgi.com/chembio/resources/mineset); the second by United Information Systems' Generic Visualisation Architecture (http://www.unitedis.com/gva) and the third by GeneExplore (now part of the GeneMaths application http://www.applied-maths.com/).

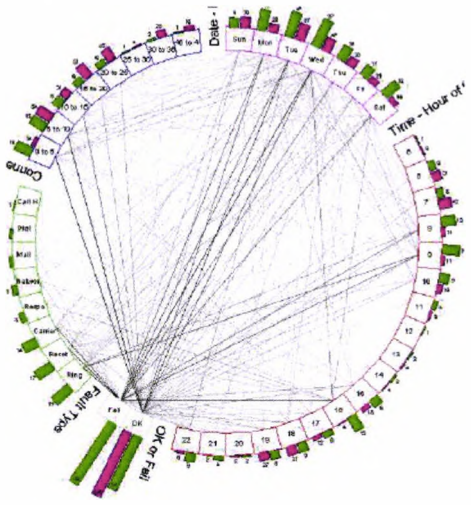


Figure 4.9: Daisy chart (http://www.daisy.co.uk/daisy.html).

directly linked to an underlying database, so that further information can be displayed. The chart also allows zooming to reveal greater detail. The charts are thus flexible and easy to understand, though the number of attributes are limited by division of the circumference of the circle and the resulting graphic depends upon the ordering of the attributes.

4.3.11 Geographical Representation

It has been said that geographical layout provides the most natural organizational principle (Chen 1999, p.106) and it is important because estimates suggest 80% of all digital data generated today includes geospatial referencing (geographic coordinates, addresses, postal codes, etc.) (MacEachren and Kraak 2001). The xy coordinates may represent actual geographical locations or non-geographical data can be presented using the map as a metaphor. Figures 4.10 to 4.12 are examples of visualizations of internet traffic. Figure 4.13 (left) shows a self-organizing map (see Sections 4.3.12 (below) and 3.4.2) of a million documents. In order to present non-geographical data as a map, dimension reduction methods need to be used. This dimension reduction leads to different implications of location in the map - adjacent points may or may not be as near as they are shown. The issue of location implication for different types of maps is returned to in Section 11.4.

4.3.12 Self-organizing Map

Figure 4.13 shows a self-organising map (described in Section 3.4.2) for a million web pages and a self-organizing map of gene expression data. Self-organizing maps of document collections enable users to effectively browse large collections, but the great amount of abstraction necessary to reduce the dimensions means that the location validity is low.

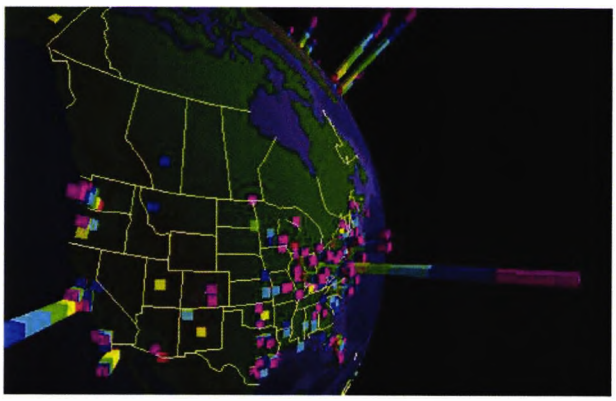


Figure 4.10: Mapping on to the globe. A geographical visualization of Web traffic from researchers at the National Center for Supercomputing Applications. The height of a bar indicates the number of bytes, or number of requests relative to other sites. The colour bands represent the distribution of document types, domain classes, or time intervals between successive requests. This view is for August 22, 1995 at 6 a.m. (Lamm et al. 1996).

Figure 4.11: Billion-byte inbound traffic to the National Science Foundation Network (NSFNET) backbone (a wide-area network that formed the core of the internet in the early 1990's) for September 1991. The traffic volume range is depicted from purple (zero bytes) to white (100 billion bytes). This is a single frame from an animation produced by Donna Cox and Robert Patterson (screen shots and animation at http://archive.ncsa.uiuc.edu/SCMS/DigLib/text/technology/Visualization-Study-NSFNET-Cox.htm).

Figure 4.12: Traffic flow on the internet. Internet traffic flows between fifty countries, as measured by the NSFNET backbone, in the first week in February 1993 (Becker et al. 1995; Eick 1996).

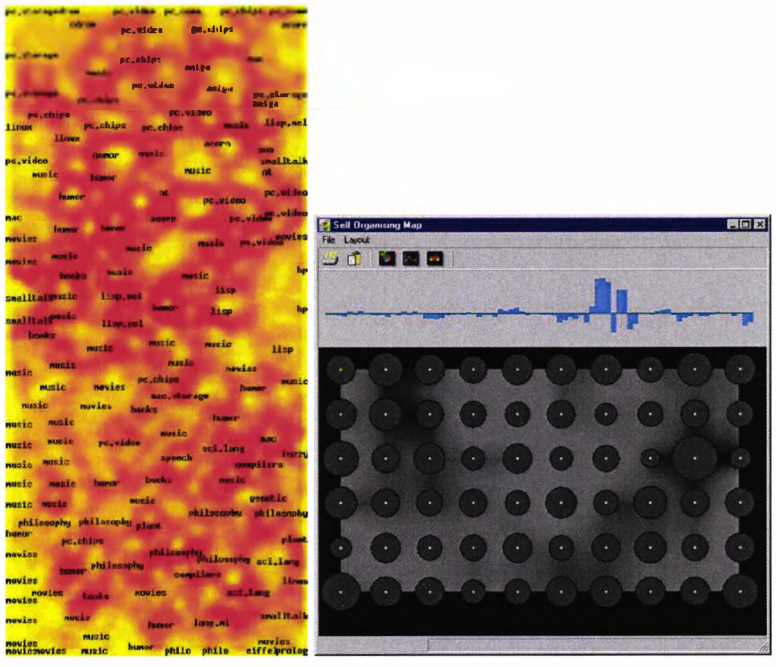


Figure 4.13: Self-organising maps. A collection of a million web pages from Websom (http://websom.hut.fi/websom/) (left), and gene expression data using GeneExplore (now GeneMaths http://www.applied-maths.com/) (right).

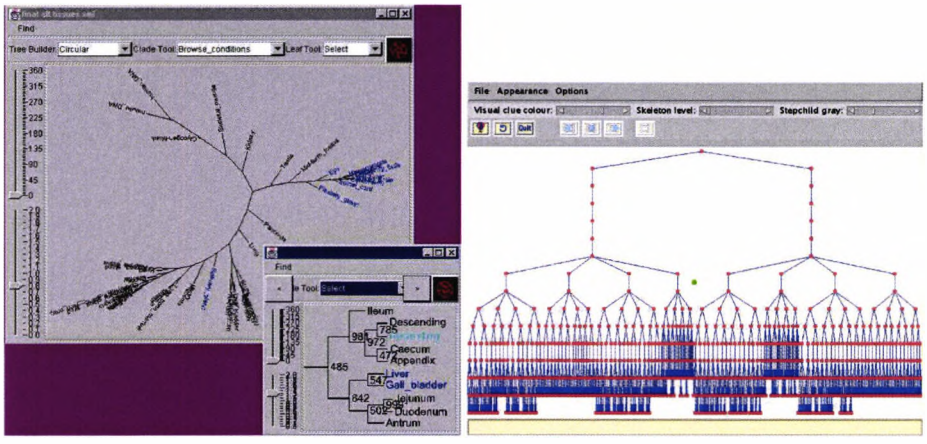


Figure 4.14: Tree layout: An unrooted and rooted dendrogram using the Expression browser (http://www.sanger.ac.uk/Users/mrp/java/ExpressionBrowser/TheTutorial.html) (left) and a tree constructed with the Latour tree visualization toolkit (http://www.cwi.nl/InfoVisu/Past.html) (right).

4.4 Trees and Networks

There are many layouts and forms of trees and networks; the following is a selection. Networks can also be considered from the matrix point of view (whether or not their links are shown), as can proximity data as discussed in the previous chapter (Section 3.4.1). In general there may be many different ways of displaying the same tree or network structures. Branches in a tree may be swapped, creating equivalent trees, yet it is hard to tell whether such isomorphisms are indeed equivalent. Another result of this is that the direct distance between nodes is not directly related to their intended distance. The amount of objects that can be displayed in trees and networks is smaller than for pattern matrices due to the necessity of showing connections between objects, though this is overcome to some extent by treemaps (Section 4.4.5) and information cubes (Section 4.4.6). Thus trees and networks do not scale up well for large amounts of data.

4.4.1 Classical Layouts

Many classical layouts have been developed for trees (Herman et al. 2000), where the nodes are positioned below their parents, together with a variety of 3D versions. These algorithms are all deterministic, but may depend on the order of input data. Usually classical tree layouts position children nodes below their parents (see Figure 4.14). Tree layouts can be adapted to produce the tree left-to-right and on a grid layout.

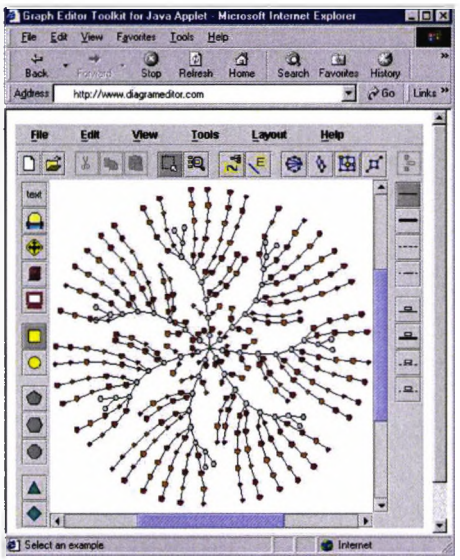


Figure 4.15: Circular tree using the Tom Sawyer graphing toolkit (http://www.tomsawyer.com/products.html).

4.4.2 Dendrograms

A dendrogram is a hierarchical tree that is a nested set of partitions produced by one of a number of hierarchical clustering procedures as indicated on page 34. Dendrograms may be rooted or unrooted as depicted in Figure 4.14. In the rooted version, the difference in height between parent and children indicates the similarity of the two children and it is this characteristic that distinguishes the dendrogram from an ordinary tree.

4.4.3 Circular Trees

Instead of placing a tree's root at the top and leaves below, branches can radiate from the root into all directions as in Figure 4.15.

4.4.4 Cone Trees

Cone trees (Robertson et al. 1991) are three dimensional extensions to 2D tree structures. The root is placed at the top (or side) and is made the apex of a cone. The tree or branch may be rotated to bring other information into view (see Figure 4.16). Cone trees have been used to visualize entire Unix file systems and as browsers for organizational structures. The arrangement allows much larger trees to be displayed than would fit on the screen using 2D layout. The cone tree arrangement is particularly suitable for many hierarchies encountered in real applications which tend to be broad and shallow. However, it can be difficult to navigate and to find desired information. Also the number of levels

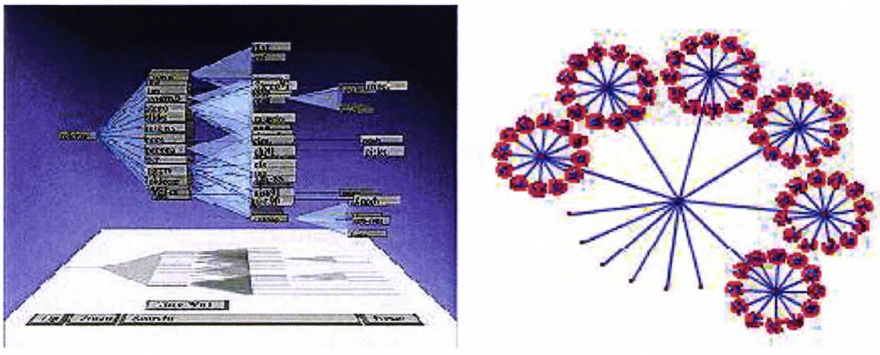


Figure 4.16: A cone tree (Robertson et al. 1991) (left) and a balloon view (Herman et al. 2000) (right).

that can effectively be displayed is limited to about 10. Occlusion is also a problem, though the body of each cone is shaded transparently, so that the cone can be easily seen without blocking the view of cones behind it. Cone trees can be projected to 2D, which is referred to as a balloon view (see Figure 4.16).

4.4.5 Treemaps

In treemaps (Johnson and Shneiderman 1991) the hierarchical structure is mapped to rectangle size and colour (Figure 4.17). The entire screen is used to represent the tree root and its children. Each child of the root is given a horizontal or vertical strip of size proportional to the overall size of its descendants. The process repeats, alternating horizontal and vertical divisions as the hierarchy is descended, so that the rectangle for a group of objects is filled with the rectangles representing its members. Thus treemaps are useful for providing overviews of single attribute hierarchical structures. They are also useful for trees larger than a couple of hundred elements when the usual node-and-link diagrams are inadequate. This is because node-and-link diagrams use most of the pixels of the display space as background. Treemaps use the full display space.

Note that the size of the individual rectangles is significant. For example, when representing a file system hierarchy, this size could be proportional to the size of the respective file. This is useful, but has a drawback relating to how users perceive the areas. For instance, the doubling of an attribute's value, and hence area, may not be accurately perceived as a doubling by the user. Also the standard treemap method often gives thin, elongated rectangles. An extension to the method has been developed, *squarified treemaps*, to address this problem by approximating the rectangles to squares (Bruls et al. 2000). Another problem is that the structure of the tree is not always evident, the user must pay much attention to distinguishing which box belongs to which level. The worst case is a balanced tree, where each parent has the same number of children and each leaf has the same size - here the treemap degenerates into a regular grid.

Figure 4.17: A tree map. This example is a screen shot of the online Marketmap application (http://www.smartmoney.com/marketmap/) which visualizes real-time stock market information.

4.4.6 Information Cube

The information cube (Rekimoto and Green 1993) (see Figure 4.18) is a 3D version of a two-dimensional design which uses nested boxes to represent hierarchical information. Like the treemap, the information cube addresses the problem that the screen space is too limited to display a large tree structure in the usual way. The designers also set out to address the problems encountered with the cone tree and treemap, when the tree is balanced or when the nesting is deep. The outermost cube corresponds to the top level data, while the next level data are represented as cubes in the outermost cube and so on. Each cube is rendered in semi-transparent colour, so that inside the cubes can be seen. The cubes can also contain arbitrary objects. The system displays using either a conventional or head-mounted display. In either case selection and rotation of the cube by hand is achieved using a DataGlove. A DataGlove is a glove equipped with sensors that feed spatial and tactile data to a computer, allowing the wearer to manipulate and explore environments in virtual reality. The information cube allows balanced and large tree structures to be effectively visualized, though one study concluded that it suffers from lack of global context, and that users find navigation and compare tasks very difficult (Wiss and Carr 1999).

4.4.7 Information Landscape - Tree

General information landscapes have been introduced above (Section 4.3.6). A particular sub-category uses the plane to draw a tree structure. Figure 4.19 shows such an information landscape tree, the result of a database query. This is a representation of 180 million hits, hierarchically arranged.

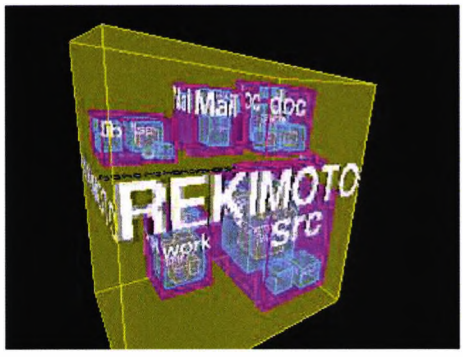


Figure 4.18: An information cube (Rekimoto and Green 1993).

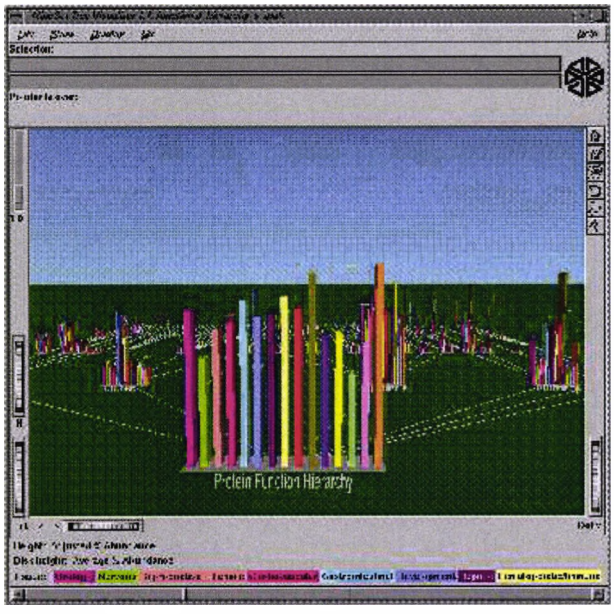


Figure 4.19: An information landscape tree created with MineSet (http://www.sgi.com/chembio/resources/mineset). The tree uses the plane and information at nodes makes use of the third dimension.

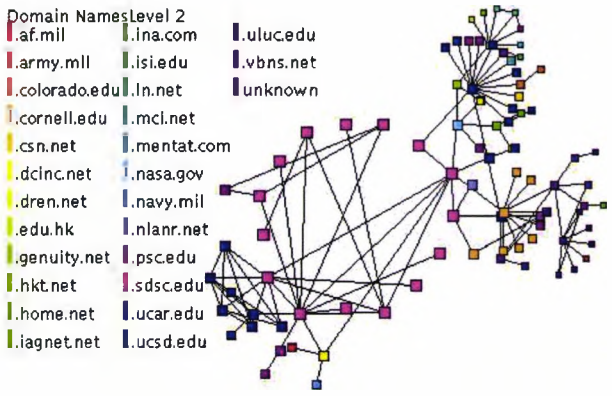


Figure 4.20: Network visualization using Caida's tool Otter (http://www.caida.org/tools/visualization/otter/), a general-purpose network visualization tool.

4.4.8 Network Data Visualization

Figure 4.20 shows an example from a tool for visualizing arbitrary network data, i.e. data that can be expressed as a set of nodes, links or paths. It was developed to handle visualization tasks for a wide variety of internet data, including data sets on topology, workload, performance, and routing.

4.5 The Size Issue - Navigation and Interaction

As soon as the number of entities reaches more than a few dozen, size becomes a problem. Whilst there are technical issues concerning rendering and layout efficiency, these are not considered here. The issue addressed here is that, from the user's point of view, there is a need to maintain overall orientation, while being able to zoom in to see details. This is described as providing *focus and context*, usually expressed as *focus+context*. The easiest and commonest way to deal with this is with opening of secondary and subsequent windows, but there are also special systems that distort the representation in some manner, so as to simultaneously give focus and context. These special systems are described in more detail in the next section.

The user's orientation is enhanced by a number of special interaction techniques; examples are *brushing* and the *semantic lens*. Brushing is the general term for linking items of data between views, usually using colour (Cleveland and McGill 1984; MacDonald 1990). The semantic lens is a method of connecting, by brushing, a group of data, delineated by the 'lens', to daughter windows that show an enlarged view of the lens. The lenses are semantic in the sense that they can display different information and properties about the underlying data points (Semantic Lenses http://industry.ebi.ac.uk/~alan/SemanticLenses/index.html).

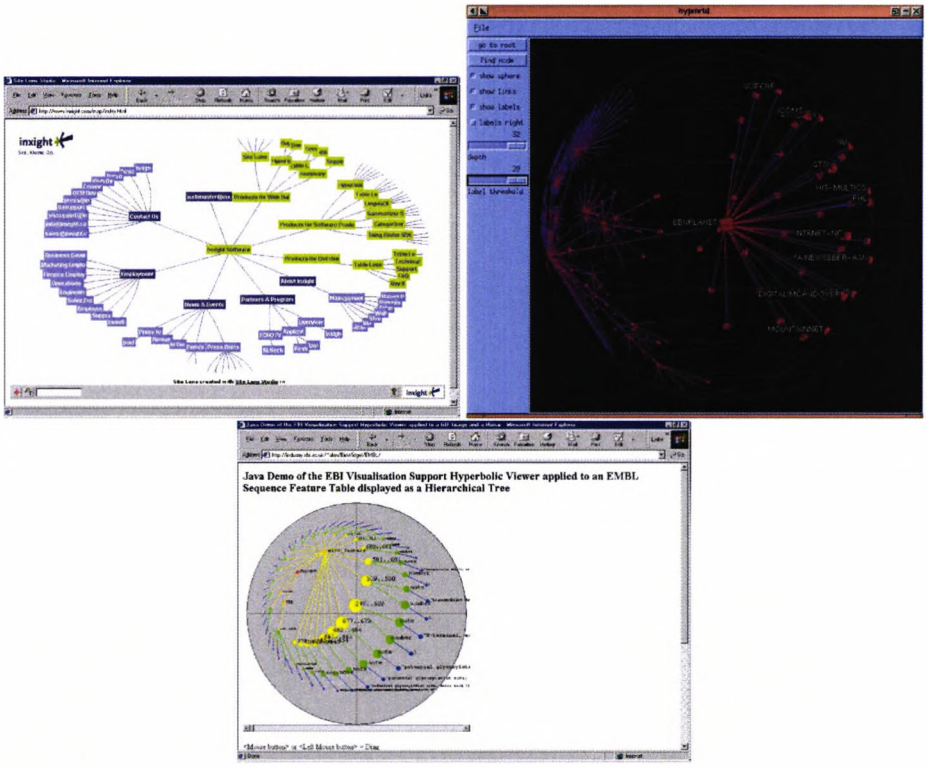


Figure 4.21: Hyperbolic layout (clockwise from top left) from Inxight's SiteLens (http://www.insight.com), Caida's Skitter (http://www.caida.org), and the Visualisation Support Hyperbolic Viewer (http://industry.ebi.ac.uk/~alan/BioWidget/EMBL/).

4.5.1 Focus+Context

Focus+context is the maintenance of overall orientation in a visualization (context), whilst being able to zoom in at the same time (focus). The most direct way of providing this is by opening up a new window to display the detail, or providing a portion of the display to show where your detailed view is in the overall representation. However, the term focus+context is sometimes restricted to those involving a single window and providing both focus and context within the same display using special methods (Card et al. 1999); these methods include hyperbolic layout, fish-eye lens, mapping onto objects and clustering. These are described in the following sections.

Hyperbolic Views

Figure 4.21 gives examples of hyperbolic trees. The trees are mapped onto a circle or sphere using hyperbolic, instead of Euclidean, geometry (Lamping and Rao 1996). The effect of this method is that distances decrease exponentially as the circumference is approached. The user can drag the nodes to change the focus of attention. Such a viewer addresses scalability, as the objects are bigger the closer they are to the focus.

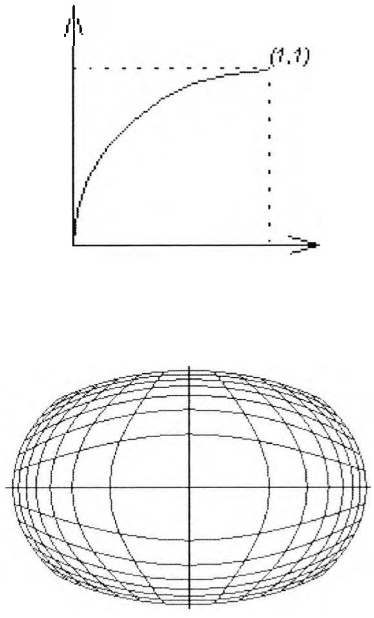


Figure 4.22: Distortion function for fish-eye effect (top). Fish-eye effect of distortion function on grid around origin (bottom). The function takes the distance in the layout, x, usually from the centre as in this case, and distorts it with a function such as this one ($h(x)$ described in the text) so that smaller values of x are increased.

Fish-eye Distortion

Produced to mimic the effect of a very wide angle fish-eye lens, the information in a part of the screen employing fish-eye distortion is shown in greater detail, whilst still displaying the whole, by the use of a distortion function (Furnas 1981). A focal point is chosen, where the information is required to be shown in greater detail, and the distance of points to this focus is then distorted by a function.

A simple distortion function (original citation Sarkar and Brown (1992), as cited in Herman et al. (2000)) is

$$h(x) = \frac{d+1}{d+\frac{1}{x}}$$

This is plotted in Figure 4.22 for distortion factor, $d = 4$.

Mapping onto Objects

Mapping-onto-objects is another way of providing context-and-focus. The globe (Figure 4.10), sphere and perspective wall (Figure 4.2) are examples of this.

Clustering

Clustering may be structure-based, i.e. non-specific, or content-based, i.e. specific and requiring domain knowledge. Ideas about clustering relate to interaction in the desire to filter and search. Clustering reduces the number of elements being viewed and potentially improves clarity and performance. Performance improvements are gained for rendering and layout. Clustering may be approached by repeatedly deriving subgraphs using content- or structure-based hierarchical clustering. Force-directed methods may be used to identify visually apparent clusters. Clusters may be grouped to form 'super nodes' and new edges induced. These super nodes are sometimes referred to as glyphs (as distinct from glyphs previously described in Section 4.3.4) and the new graph as a compound graph.

4.6 Comprehension Challenges

In terms of user comprehension, there are a number of general categories of problem that these morphologies present:

- Unfamiliar, possibly complex, visual forms. Examples are colour maps, spiral colour maps, parallel coordinate plots and Daisy plots. For example, this may be where a colour scale replaces a numerical scale providing a different perception of the value scale or in the use of a polyline to represent an object. The user needs to become familiar with the application and appreciate relevant differences in perception from the use of colour scales etc.

- Existence of equivalent representations. This may be due to arbitrary ordering of attributes, as in colour maps and parallel coordinate plots, or isomorphism, such as in dendrograms following hierarchical clustering. Different representations based on selections of different visual variables result in different types of glyphs. The user needs to be made aware of the equivalent representations.

- Ambiguity in meaning of spatial component. Is it obvious what the distance between objects means? The layout positions may be chosen to create a pleasing effect, or represent measures of actual or approximate distances or dissimilarities. The user needs to be aware of the meaning of the spatial component.

- Transformations of the data may have been applied. Dimension reduction provides distances between objects. Semantic or structural clustering produces new objects. Transformations are often abstractions - approximations which need to be revealed to inexperienced users.

- Multiple windows. Focus+context techniques and other interaction techniques often result in a large number of windows for the user to control and navigate between. We need to know more

about the problems associated with multiple windows. This is an active research area, 2003 seeing the first international conference entitled 'Coordinated & Multiple Views in Exploratory Visualization' (Roberts 2003).

4.7 Summary

This chapter has described some of the wide range of visualization morphologies available. Colour maps, mapping onto objects, parallel coordinate plots, glyphs, star plots, information landscapes, surface plots, cityscapes, scatterplots, Daisy charts, geographical representations and self-organizing maps are ways of viewing multivariate data that have been described. Some methods are limited in the number of attributes or objects that they can present. Classical trees, circular trees, cone trees, treemaps, information cubes, tree representation as information landscapes, and network data visualizations have been illustrated to show the variety of ways that trees and networks are visualized.

Focus+context methods are used to see areas in detail as well as being able to provide overall orientation to the user. An obvious way to do this is by using child windows. Some special methods provide visual distortions: examples are hyperbolic views and fish-eye distortion. Other methods map data onto 3D objects or present the results of semantic- or structure-based clustering. A key technique is the linking of windows via brushing, so that items highlighted in one window are also indicated in the others. This forms the basis of overview and detail window pairs, as well as special techniques such as the use of semantic lenses.

Key challenges to the user's comprehension arise from: unfamiliarity, especially for new types of visual representations and inexperienced users; the existence of equivalent, in some cases isomorphic, representations; ambiguity in meaning of the spatial component; whether or not dimension reduction transformations have been applied; the use of multiple viewing windows. In general these are all issues that one would like users to be somehow made aware of. There are also aspects whose impact needs further study, such as the use of multiple windows.

Chapter 5

Open Questions for Information Visualization

5.1 Introduction

The last three chapters have presented a detailed survey of the field and obstacles to comprehension have been identified (Tables 2.3 and 3.1 and Section 4.6). This chapter returns to the thesis objective concerning the usefulness of the proposed technique signature exploration and its value for increasing the comprehension and choice of visual displays of complex data (set out in Section 1.3). Now this objective can be reconsidered in relation to the wider open questions and areas of active development in information visualization. This discussion provides further evidence for the importance of the general thrust of the work - *enhancing comprehension of complex data visualizations* - and points to the specific means of achieving this proposed - the exploration of the *signatures* of the transformations and visualizations.

The question 'How can the user's comprehension of the transformations and representations of visual depictions be supported?' is the motivation for the techniques defined and developed in this thesis. The need for understanding is key, since, without this, 'amplification of cognition' cannot result from visualization (as indicated in the introduction (page 1)). However, there are also specific motivations for comprehension support arising from particular requirements of current developments in information visualization, which are discussed here. The previous chapters on data structure, visualization purpose, layout and morphologies have indicated the breadth of developments in information visualization. Looking at this detail shows many areas that offer scope for research relating to new and improved forms of interaction and visual representation. Whilst these contributions are and continue to be valuable, the number of forms already developed point to a number of general issues

relating to *design*, *combination* and *use*. The last decade or so has produced an explosion of forms (interaction and representation), some of which have been described in the previous chapter, now it becomes more important (or at least equally important) to examine their combination and use, and how to design such systems effectively (Spence 2001, Preface). For instance, a goal for exploratory data analysis is the presentation to the user of systems that represent a convenient *shelf of visualization tools* with which they can explore their data. Here four active areas of information visualization are described by the following open questions:

1. How can designers and users of visualization systems be made aware of the impact of cognition and perception issues?
2. How can techniques and tools be combined effectively?
3. How can the visual exploration of complex data and systems be supported?
4. How can datamining techniques be integrated with visualization?

Together with the comprehension question above, these five questions relate closely to the research agenda for geovisualization developed by the International Cartographic Association which organized international teams to address four themes: *representation* of geospatial information, *integration* of visual with computational methods of knowledge construction, *interface design* for geovisualization environments, and *cognitive/usability* aspects of geovisualization (MacEachren and Kraak 2001).

The five questions are examined in the body of this chapter. The discussion of the comprehension question reviews the obstacles to comprehension of the previous chapters. The conclusion introduces the idea of the exploration of *signatures* of transformations and representations, in preparation for the detailed specification of signature exploration in the following chapter.

5.2 Cognition and Perception Issue Awareness

Ware has set out to bring together important aspects of the large body of work in this area in his recent book, in which he says:

> "There is a gold mine of information about how we see, to be found in more than a century of work by vision researchers." – *Information Visualization: Perception for Design* (Ware 2000a)

However, it transpires that this goldmine, like some of its real counterparts, may be very difficult to mine, so that incorporating this body of knowledge is not easy. The information often appears anecdotally, such as 'motion assists 3D perception' (Ware 2000a, p.282). Also the observations are

highly task dependent and many ideas have been developed in the context of specific experiments (Ware 2000b). Nevertheless, authors indicate considerable scope for application of the results so far obtained (Herman et al. 2000; Ware 2000a).

Section 3.3.1 on page 29 has briefly discussed work relating to perception and understanding of visual representations. Other sections on colour, how we perceive 3D, differences in mapping to the different shapes etc. show the kind of characteristics that designers and users need to be aware of and the potential for deception in visual depictions. Thus, the task of information visualization designers is to apply this knowledge in their applications, both to take advantage of the human visual system and to avoid unknowingly misleading the user. It is not only the designer that needs to be aware of these issues. At times, the user is choosing, for example, a colour scale or an attribute to map to a shape, and may also need more information to understand the implications of their choice. Such information could be given in textual form: 'The colourscale you have chosen has the following characteristics ...'; or as datasets that illustrate good and bad effects.

5.3 Effective Technique and Tool Combination

There is a general trend towards combining greater numbers of tools within a single application, but also the need to integrate statistical tools within the visualization application (Keim 2001; Unwin 2000). Besides many tools to combine, there are many types of users, specialisms, tasks and data types. The many instruments (tools, techniques) that are available for data exploration (which may or may not result in a visualization), often require high levels of specialism to operate - the user may need to be a domain expert, may need to acquire expertise in the use of the tool, or they may be an expert in the use of a particular technique or group of techniques such as dimension reduction algorithms. How can a single user operate a tool that combines difficult techniques? The many techniques available are also not all relevant to all tasks, so there is an appropriateness in the consideration of what techniques should be combined.

The means of combination relates to the issues of code reuse, the large number of possible components and the constant emergence of new hardware and software technologies. A need arising from the move to compose visualization applications from a number of visualization techniques is the ability to efficiently program such tools and interfaces.

Researchers need to be able to compare their results; greater ease of use of other techniques would assist this. Though a similar result may be achievable by other means, such as the establishment of benchmark datasets. There are drawbacks to the pursuit of common components and standards, particularly the time taken to decide standards and the increased effort required by module developers to meet the requirements.

A vision for a future system for the viewing of complex data includes a wide range of layout

methods and visualization morphologies as described in the previous two chapters. The design in Figure 5.1 is an example of such a system using similarity metrics, animation, level-of-detail, a 3D editor and a set of visualization morphologies. The user can create different views of the object group in an editor. This avoids a search for a 'correct' view, encouraging different views to be created. Selection and transformations of the data must be deliberately chosen, the base structure being that of a log of events as outlined in Section 2.4. Using animation, clustering and drill-down, the data can be rearranged and viewed in other ways.

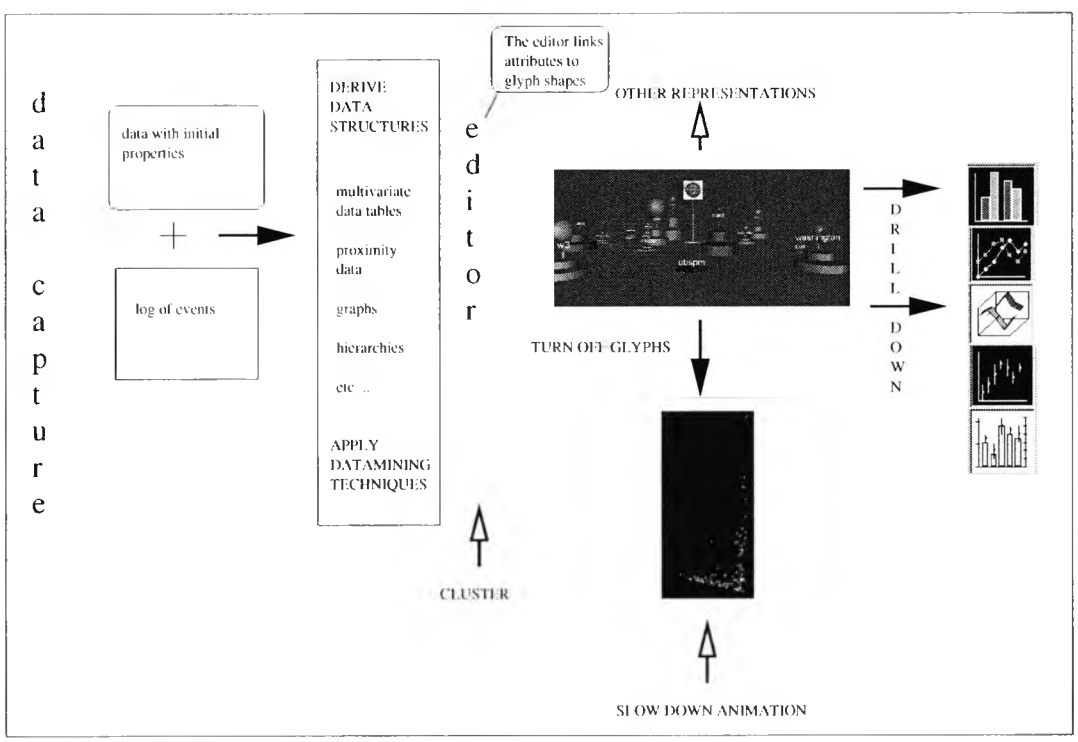


Figure 5.1: Technique combination for viewing complex data based upon an animated glyph world with editor for linking attributes and datamining techniques to derive different data structures.

An information landscape or a scatterplot is the central visual representation in Figure 5.1. The user decides what features they are interested in and maps these to dimensions, shapes etc. of the glyphs. The layout can be based upon a specification, (eg geographic, domain, arbitrary), or upon the derivation of a distance metric using a user-selected feature group or the whole dataset. It can be static or dynamic. The glyphs can be animated with the log data. One possibility is to watch an animated production of a visualization as data is collected. The speed of animation can be controlled to show different levels of detail. Glyphs can be turned on and off (when off, the object is a simple sphere or cube). Instead of the information landscape, other visual representations can be selected from the range of possibilities indicated in the last chapter.

There are a number of systems that feed data into 3D worlds (e.g. Russo Dos Santos et al. (2000)),

and many other types of systems have been developed that combine different kinds of methods (for example the Xmdv tool (Rundensteiner et al. 2002), XGobi (Swayne et al. 1998) and VisDB (Keim and Kriegel 1994), but no one system as yet combines this wide range of techniques. However, the trend is for greater tool combination and also to integrate datamining techniques with visualization (see Section 5.5 below).

5.4 Systems for Visualization of Complex Data

The starting point for visualization is often considered to be a specific table, tree or graph structure. However, much data has a more general origin such as a log of events, as described in Section 2.4, from which different structures may be derived. In viewing this data, the user needs to be able to create the data structures that are of interest to them (for instance, in creating a relevant structure from a log of events, as in the creation of the customer/destination call matrix from the log of individual calls) and view them in different ways. Thus the user is given flexibility in creating custom views, for instance using different visualization forms and different glyph attributes, clustering algorithms, distance metrics etc., as well as animation. The process of the user creating data structures and views also preserves in the mind of the user the multi-faceted nature of complex data and systems. Such a visualization application thus combines the ability to create data structures from a log of events (if this is indeed the starting point) with the combination of information visualization and clustering techniques as illustrated in Figure 5.1.

5.5 Integration of Datamining and Information Visualization Tools

The domain of *knowledge discovery in databases* develops methods that find useful structure in large volumes of data and seeks to explain this structure. The variety of techniques used for this purpose, association rule derivation, classification, clustering etc., are generally referred to as datamining techniques. Visual datamining integrates visual with computational methods in this context as well as developing the human visual system's pattern recognition abilities in general (Keim 2001). According to MacEachren and Kraak (2001):

> "Fundamental advances in our approach to (and success at) knowledge construction from geospatial data are most likely if we can integrate the advantages of computational and visual approaches. The goal of this integration is visually enabled knowledge construction tools that facilitate both the process of uncovering patterns and relationships in complex data and subsequent explanation of those patterns and relationships."

Human involvement can be before the datamining step, to display initial data and focus on or narrow the search space to make it relevant. It can be during the datamining step, *tightly integrated visualization*, to display intermediate results and direct the search, allowing domain knowledge to be applied. It can be after the datamining step, *subsequent visualization*, to display the result. These different points at which human involvement can take place are identified by Keim (2001). Subsequent visualization examples include displaying results after association rule derivation, classification, clustering and text mining. Quite a few examples of subsequent visualization exist (for a listing see Keim and Ankerst (2001)). Tightly integrated visualization is a more recent area of development (Andrienko et al. (2001); Keim and Ankerst (2001); MacEachren et al. (1999); Shneiderman (2002)). Potentially, the integration of these techniques increases the complexity that the user faces. Where techniques are also combined, the user needs to become expert in each technique.

5.6 Support for Comprehension of Visual Depictions

The last three chapters have revealed a number of obstacles to comprehension of visual depictions. Section 2.7 and Table 2.3 examine those related to data types and structures. Section 3.5 and Table 3.1 consider those arising from layout mechanisms and Section 4.6 looks at those associated with particular morphologies. These three sets overlap, since data structure, data layout and visual morphologies are aspects of a whole in the sense that all three together affect any one visual depiction that the human views. Key requirements are summarized in Table 5.1 and example graphics given in Figure 5.2.

How to address these obstacles? Additional functionality and development of composite systems are useful. The added functionality may enable users to interact with the visual depictions and the underlying data more easily, thereby understanding more about their nature. Composite systems allow the user to view their data in different ways and thus allow and encourage methods to be contrasted. However, a large number of the issues require the user to be made aware of various characteristics of the visualization process. Dykes (1997) introduces the idea of exploring these characteristics. This awareness can be considered to involve *revealing* the characteristics of data, data structure and layout, and visual form to the user. Intuitively, one wants to put a known, familiar set of data into a visualization process and see what happens. If the dataset is specially constructed to illustrate a certain feature of interest, one may then have a concrete example of the behaviour of a particular visual depiction. This idea is the basis of *signature exploration* described fully in the next chapter. The key obstacles to comprehension that may be addressed by this are indicated in Table 5.1.

Difficulties in comprehension are exacerbated by the involvement of a much wider range of users, both within and outside the scientific community:

Comprehension Issues	**Might constructed data be useful?**
Difficulty of conceptualization of high dimensional spaces.	Yes (indirectly)
Interchangeability of objects and attributes.	No
Equivalence of data structures.	No
Transformation between types and structures of data.	No
Existence and implications of large amounts of metadata.	No
Impact of selection and standardization.	Partly
Different ways the same data can be represented.	Yes
Special features of layout choices, including the retinal variables.	Yes
High levels of abstraction.	Yes
Impact of layout choice upon interactivity.	No
Characteristics of unfamiliar forms.	Yes
Equivalent representations, e.g. ordering of attributes, mapping of glyphs.	Partly
Ambiguity of spatial component.	Yes
Transformations.	Yes
Multiple windows.	Possibly
Context and focus techniques.	Possibly

Table 5.1: Key issues that present obstacles to comprehension and how they may be addressed by the use of specially constructed datasets. By *specially constructed data* is meant data that contains a particular feature, known to the user. This data is then put into a visualization process, so that the user has a concrete example of the behaviour of the process. This notion forms the basis of *signature exploration* described in detail in the next chapter.

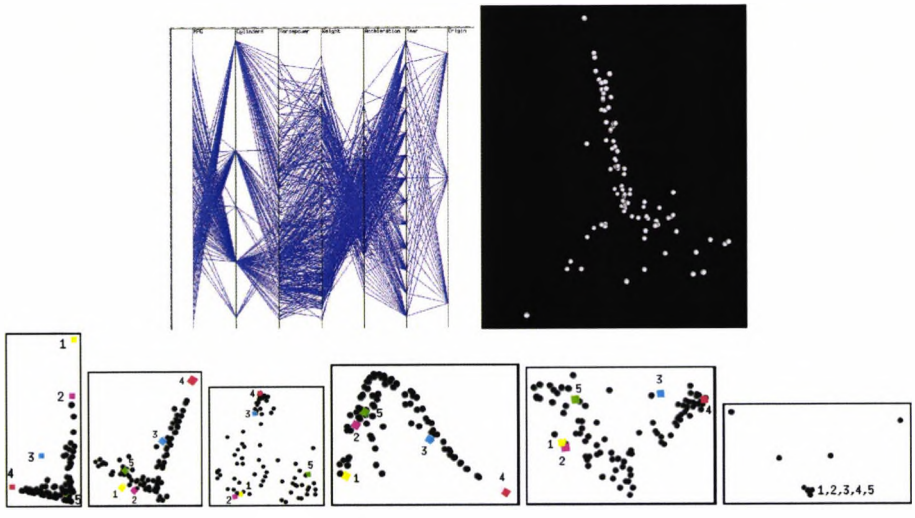


Figure 5.2: Examples of graphics involving comprehension difficulty: (top left) parallel coordinate plot (from Figure 4.3) - How to assist users with unfamiliar representations? (top right) dimension reduction of the calldata set (from Figure 1.1) - How to make sense of the resultant patterns? (bottom set) 6 different possibilities for dimension reduction (from Figure 3.3) - How to choose an appropriate method?

> "Earlier studies of data visualization mainly had the professional scientist as the target customer, concentrated on the examination of static presentations of data and were closely associated with statistical analysis. By contrast, and as a result of technological progress, the benefits of information visualization are now available to a much wider range of customers ranging from supermarket managers to fraud investigators." *Information Visualization* (Spence 2001, Preface)

This means that the user does not always have a statistician at their elbow, neither is one themselves. The visualization tool itself must guide their choice of algorithm etc., if any guiding is to be done. For the specialist with statistical knowledge, mechanisms to increase comprehension may also prove valuable since they may reduce the time taken to find information. Greater insight may also result, particularly where complex combined techniques are involved.

5.7 Summary and Conclusion

This chapter proposes five areas of information visualization that constitute open questions, though these issues are not restricted to the field of information visualization.

1. How can designers and users of visualization systems be made aware of the impact of cognition and perception issues? Users, as well as designers, need to be aware of issues such as how we perceive colour and 3D.

2. How can tools be combined effectively? Technique combination is desirable to give users access to a variety of techniques for visualization, representation and mathematical transformations (such as clustering), but produces systems that are harder to build and harder for the user to understand.

3. How can the visual exploration of complex data and systems be supported? Complex data and system viewing emphasizes the need for composite tools and user comprehension.

4. How can datamining techniques be integrated with visualization? The field of visual datamining is exploring and emphasizing ways in which visualization can be integrated with datamining.

5. How can the user's comprehension of the transformations and representations of visual depictions be supported? Dimension reduction algorithms and novel visual methods give results that are either hard to understand or with which the user is unfamiliar, leading to difficulty in making appropriate choices and conclusions for a wider range of users.

The last of these questions - how to support the user's comprehension of the transformations and representations of visual depictions - is a key issue in relation to the others. How are the other open questions addressed by improving comprehension? Concerning the first: increasing comprehension includes making people aware of the nature or characteristics, the strengths and weaknesses, of particular visual representations, such as 3D representation or shading and colouring. Relating to the second and third questions: combining techniques and looking at multi-faceted data and complex systems, requires increased comprehension support, because of the higher levels of complexity involved. Relating to the fourth question: the integration of datamining and visualization puts greater demands upon the user's understanding, whilst seeking to enhance it.

Examining the need to support the user's comprehension prompts the intuitive desire to put familiar data into the process to obtain a concrete example of the behaviour for reference. Taking this idea further suggests the use of specially constructed datasets that contain particular features in order to reveal the pattern, or *signature*, that the visualization process gives for that particular feature. This idea forms the basis of *signature exploration*, which is introduced and defined in the following chapter. Examination of the key obstacles to comprehension identified in the previous three chapters indicates that this approach will assist with many of these obstacles.

In general terms, the issue is that the many techniques and transformations used in information visualization have different strengths and weaknesses. The problem is how to get the best out of them. To do this the user needs to understand them intuitively, both singly and in combination. What is needed is a framework or methodology to guide the design of visualization systems for the principled analysis and visualization of complex data.

Chapter 6

Signature Exploration: Definition and Proposed Techniques

The data explosion and the possibilities of the human visual system for viewing data have been illustrated in the previous chapters. Visualization has been examined in terms of data types, issues, layout and morphologies. Issues that raise obstacles to comprehension have been identified. A particular concern has been the thread of complexity, how to deal with multi-facetedness and dimension reduction. Open questions have been examined which centre around comprehension, or are served by it. It has been emphasized that the trends of tool combination, on the one hand, and integration of visualization in datamining processes, on the other, underlie the importance of considering comprehension of visualization methods at this time. The intuitive desire to feed **known** data into the visualization process (to see what happens) has been introduced, as suggested by work with image libraries and the use of fingerprinting in science. This chapter now defines the concept, *signature exploration*, developed from this idea. Five techniques for applying signature exploration are described.

6.1 Introduction

Following the description and discussion of visualization methods for complex data that has been undertaken in the previous chapters, we can summarize the problem and indicate the main aspects that need comprehension support. Complex data, with many observations and many attributes for each observation, are often impossible to visualize in their entirety - thus displays typically depict subsets or abstractions of the data. There are two reasons for this. On the one hand, a single large data table poses a problem because of its high dimensionality. On the other hand, some datasets provide sets of different types of tables that cannot be simultaneously viewed - it may be that, in its raw form,

Mathematical Transformation	**Graphical Representation**
Conceptualizing high dimensional spaces.*	Different ways the same data can be represented.*
Equivalence of data structures.*	Characteristics of unfamiliar forms.
Transformation between types and structures of data.*	Equivalent representations, e.g. ordering of attributes, mapping of glyphs.*
Existence and implications of large amounts of metadata.*	Multiple windows.†
Impact of selection and standardization.*	Focus+context techniques.†
Special features of layout choices.*	Special features of layout choices, including the retinal variables (excluding transformations).*
Ambiguity of spatial component.	Ambiguity of spatial component.
High levels of abstraction.*	Impact of layout choice upon interactivity.†
Interchangeability of objects and attributes.*	
Transformations.	
* Relevant also to the issue of seeing the visual depiction as 'one view of many'. † Secondary representation: interaction with representations or combination of representations.	

Table 6.1: Categorization of key comprehension issues relating to mathematical transformation and graphical representation. Also indicated are those issues relevant to seeing the visual depiction as 'one view of many'. The table categorizes the issues from Table 5.1.

such a dataset is a log of events reflecting both attributes and interactions or that the data are complex linked datasets, such as those increasingly provided via the Web. There are many techniques available for the visualization of complex data, however there exist many obstacles to comprehension of the resultant graphics. Overall, to increase the user's understanding, there are three main aspects of the process that the user needs to appreciate the implications of:

- mathematical transformation
- graphical representation
- each depiction as 'one view of many'

That is to say, the obstacles to comprehension identified in Sections 2.7, 3.5 and 4.6, and summarized in Table 5.1 fall broadly into the categories *mathematical transformation* and *graphical representation* as shown in Table 6.1, and the appreciation of the depiction as 'one view of many' is an issue which relates to issues from both.

To the user, mathematical transformations and graphical representation are seen in combination. Mathematical transformation of the data is a data-to-data operation and often involves a significant abstraction of the original data and therefore a **loss**. Graphical representation is the production of a graphical depiction of the data (a mapping from the data to one or more graphical objects). Whilst representation always follows transformation in order to produce a graphic, the representation may

be simple, as in the case of a dot representing a point, or complex as in the sequence of lines in a parallel coordinates plot (Inselberg 1997) (Section 4.3.3) or ziggurats in a glyph world (Ribarsky et al. 1994) (Section 4.3.4). Many novel forms of representation have been proposed to enlarge the number of attributes, or overall matrix entries, that can be directly visualized (Card et al. 1999; Chen 1999; Spence 2001) as have been illustrated in the previous chapters. It has been emphasized that new representations bring with them the problem that they are unfamiliar to users. Thus for both transformations and representations, tools and techniques are required to characterize the behaviour of the visualization process to the user, a process which may involve the loss, distortion or hiding of information.

The extent to which applications show views of the data as 'one view of many' varies. Shneiderman (1996) defines the steps of visual information seeking as: start with an overview of the dataset, zoom in on items of interest and filter out uninteresting items, then provide details on demand. This he describes as the *visual information seeking mantra*. It indicates an interactive process, but does not imply interactivity at a particular level. Consider the case of providing overviews of database contents for querying. Here, visualization may not be a particularly interactive process at the overview level (though contain many elements of interactivity overall) - the user may have one look at the overview then go on to create queries or zoom in to look at detail. The user only wants to get a general idea about the overall content. Arguably, this means that the precise representation does not matter, but a static presentation of the data overview has the potential to be misleading. If visualization is presented as an interactive process in which the user looks at many different views (Monmonier 1991a), at a particular level, then the user is not misled into thinking there is one correct view, though they may still struggle to understand the meaning of the different representations. In geovisualization, the inevitability of misleading the viewer in a map, is well known (Monmonier 1991b) and a number of strategies have been proposed to avoid what is described as the one-map solution (Monmonier 1991a). As maps of non-spatial data are used more frequently (Fabrikant and Skupin 2004), this expertise needs to be applied more widely.

The two main goals from the designer's perspective can be expressed thus:

- Avoid the implication that there is one correct view where this is inappropriate.
- Provide the means for enhancing the user's understanding of the transformation and representation, especially where abstraction and complexity or novelty of visual depiction are involved.

The second of these is the focus of this work. To some extent it serves the first in that increasing the user's understanding of visualization methods for high-dimensional data also increases their appreciation that such data are many-faceted. However, examination of the general problem of how to avoid a fixed view of complex data is not the main aim here, though it is addressed indirectly.

The direct result of the complexity or novelty of representations is that a user's initial reaction to a graphic may be 'What does this mean?' Thus, users need methods and tools that help them understand the necessarily abstract representations required to depict complex data. Intuitively the user wants to take *known* data and put this into the visualization process to see what happens. This idea comes from work in image libraries and the application of fingerprinting as described on page 6; it was discussed with visualization system designers and researchers at a number of conferences and workshops, and with users (biologists associated with our research group, industry participants in various seminars and workshops). Thus, in these discussions users say that they want to know how specific data, that they are familiar with, are shown in the graphic and designers/researchers want users to know that a certain pattern in the visual depiction indicates a certain feature in the data. It is this intuition that led to the proposal of the concept *signature exploration* - to use datasets that are *known* in some way, to explore the behaviour, or signatures, of the different visualization techniques. This illustration of the behaviour of the process can be described as helping to *appreciate the computational element*. The *computational element* may be taken to cover both visual and non-visual elements of the application and becomes of greater importance as composite tools of greater complexity are created. One way of appreciating the computational element is to provide an animation of the algorithm itself (see the Complete Collection of Algorithm Animations site: http://www.cs.hope.edu/ alganim/ccaa/), but this illustrates how the algorithm works, rather than its effect upon data, i.e. how features in the data map to patterns in the visual representation.

The next section, Section 6.2, defines the concept signature exploration. The examination of signature exploration has inspired a number of techniques relating to the construction of datasets containing features of interest and these are described in Section 6.3. In Section 6.4 it is demonstrated that these approaches build upon existing work which provides interaction mechanisms for visualization. The chapter concludes with a summary.

6.2 Definition

I have proposed the use of constructed data in a process called signature exploration (Noy and Schroeder 2001; Noy Noy) to assist with the difficulty of understanding abstractions or novel representations of high dimensional data, and the corresponding problem to the user of choosing between different visual depictions. This problem area covers a wide variety of visualization situations, wherever there is difficulty in interpreting the resultant graphic.

Definition 10 *Signature Exploration.* ***Signature exploration*** *is defined as the exploration of the behaviour of a* ***visualization method*** *by means of the visualization of specially constructed datasets, which contain, or are representative of, particular features of interest. In this way* ***known*** *datasets*

are visualized for the user as concrete examples of the behaviour of the method. The visualization result, the pattern produced, is the ***signature*** *of the method for that data. Different methods will produce their own corresponding signatures for the dataset, enabling comparison. The dataset may be one of a set of standard types provided, or any set constructed by the user. Thus the signature of the method is explored for sets of known data. The term* ***signature*** *is used in the sense of a distinctive mark, or pattern, indicating identity of a dataset feature for a particular visualization method. By* ***known*** *is meant that the user has a sense of knowing the data, that it contains or represents a certain feature, in a concrete, but not necessarily precisely defined, way.* ***Visualization method*** *is used here to mean any application, tool or algorithm that produces a visual representation of data.*

Why use the term *signature*? As indicated in the definition, it is in the sense of a 'distinctive mark', 'characteristic' or 'indicator', that the term signature is used here. Signature derives from the Latin, `signatus`, past participle of `signare`, *to mark*, from `signum`, *sign* (American Heritage Dictionary of the English Language 2000). One of its meanings is given as:

> *A distinctive mark, characteristic or sound indicating identity.*

Also, there is an equivalence between the words *signature* and *sign* since to sign means to write one's signature. Sign has the meaning:

> *Something that suggests the presence or existence of a fact, condition or quality. An indicator.*

Thus *signature* in signature exploration has the meaning:

> *A distinctive mark or pattern indicating identity of a dataset feature for a particular visualization method.*

In theory, from the above definition, all instantiations of the visualization of data are signatures. If, however, the set of all possible datasets is reduced to a set of representative datasets which are representative of features of interest, the scope is reduced. The input data is reduced to a subset that represent particular features, but this is not a closed set. The ability to create further datasets is important for two reasons: firstly, it engages the user in the interaction process, bringing their domain knowledge to the fore; secondly, the likelihood of universally agreeing a set of datasets, that appropriately represent all possible features of interest, is low. Thus armed with a set of characteristic input datasets and the ability to create more, the user may be able to make a series of formulations of the form - 'for this visualization method, this pattern means the presence of feature x in the dataset'.

The characterization of *knowing* as 'a sense of knowing' (see Definition 10) can be expressed as the user having an understanding of the characteristics and structure of the dataset being used as an exemplar, though one would have to then say what is meant by 'understanding', what their level of

statistical expertise is, say. However, it is not that the user is assessed as 'knowing' by an external authority, but that the user assesses themselves as knowing in some way. This allows the meaning to encompass a variety of users, for instance, novices, classification experts, statisticians.

The goals of signature exploration are:

- To increase the user's understanding of the behaviour of a particular visualization method for complex data.
- To enable the user to compare different visualization methods for the purpose of choice or classification with respect to the visual evidence of specific dataset features.

These goals have the underlying objectives:

- To develop a set of techniques for aiding comprehension.
- To develop a framework for the design of visualization systems for increased comprehension.

These two goals and two objectives correspond to the objectives of the work given on page 11 as four hypotheses of the benefits of signature exploration.

6.3 Proposed Techniques

Five techniques for signature exploration have been proposed and investigated. Each technique involves the production or provision of constructed data containing a feature or features of interest:

1. Generic dataset provision
2. User-construction of data
3. Querying
4. Insertion of landmarks
5. Elicitation and application of feedback data

These techniques are introduced and defined here; investigations of each technique are presented in following chapters.

6.3.1 Generic Dataset Provision

Definition 11 *For **generic dataset provision**, characteristic sets of data that illustrate the various behaviours of metrics and visualizations are provided within the visualization application.*

	apples	pears	bananas	oranges	...
customer1	1	2	3	4	...
customer2	2	2	2	2	...
customer3	3	10	20	30	...
...	...	...	...	...	...

Table 6.2: Example: to show the behaviour of different metrics. Euclidean metric (followed by multidimensional scaling) groups customers 1 and 2, angular separation (and multidimensional scaling) groups customers 1 and 3.

Consider, for example, the data in Table 6.2. This data can be used to illustrate the difference between the Euclidean and angular separation metrics (Section 3.4.2 on page 36). Applying the Euclidean measure (followed by multidimensional scaling) groups customers 1 and 2, whilst angular distance (and multidimensional scaling) groups customers 1 and 3.

The purpose of generic dataset provision is to provide the user with a range of datasets showing specific features, so that they can form a more concrete impression of the behaviour of the visualization method, and to assist them in the comparison of behaviours of different algorithms. There are many issues relating to the choice and specification of these generic datasets, which are the subject of ongoing work. However, in order to conduct an initial test, a number of representative datasets have been used.

There are numerous on-line repositories of datasets, predominantly containing real-world datasets, (detail of some of these can be found in the next chapter, in Sections 7.2.1 and 7.2.2), and a lot is known about many of these, from their use, for instance in machine learning and statistics. However, these are known in the sense that they have been used in numerous studies, rather than that they contain a specific feature with which to illustrate the behaviour of graphical representations to the user. Thus, this study considers synthetic datasets for the generic dataset type and, for the time being leaves aside real-world datasets.

6.3.2 User-Construction of Data

Definition 12 ***User-construction of data*** *is the construction of data by the user -* ***static*** *or* ***simulated*** *construction. By* static *is meant the direct specification of a data table, i.e. the user enters data directly into the table.* Simulated *refers to a data table derived from a simulation whose model is provided by the user as specified behaviours.*

In the customer/fruit scenario above, a user, perhaps the store manager, creates specific data values for their typical customer groups and sees how they are shown by the visual representation. For instance customer_type1 buys apples and pears and nothing else, customer_type2 buys oranges and bananas and nothing else, customer_type3 only buys apples. On the other hand, the store manager

may, from their experience, believe that there are certain behaviours such as: the customer who regularly buys certain fruits once a week; the customer who lives locally and enters at random and buys a small amount chosen randomly; the customer who comes in twice a week and buys what is on special offer, and so on. From these behaviours a simulation can be built to generate data which can then be represented visually.

Thus, for user-construction of data, the user may create the data from scratch, transform an existing dataset that they are exploring or start with one of the generic datasets provided. Static constructions are matrices specified by the user, which can then be visualized. The variable values for each entry may be entered individually, generated according to a formula, or represent 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. In relation to the use of data constructed via simulation, perhaps the user looks at their own real-world dataset 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 that 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 were produced by these behaviours what would the visualization look like?' It can be useful to use software agents to model the entities in the data, for instance the customers in the calldata example, then agent simulation can be used to test hypotheses for patterns seen in the data, based upon a set of behaviours given to the entities. An example is the use of an agent-based characterization to explain regularities in web surfing (Liu 2002).

6.3.3 Querying

Definition 13 ***Querying****, based on a real dataset that the user wants to analyze, results in a subset of the data being directly selected by the user (either from the visualization itself, or by querying the original dataset) or automatically derived (e.g. outliers or extremities).*

Examples of directly querying the fruit/customer data are: Which customers buy more than 10 apples? Which customer has bought most items? An example of visual querying is the user highlighting a customer on the edge of the display and viewing the customer's data.

In querying, a cluster in a visualization of a dataset under consideration may be highlighted, or an outlier, or the extremities of a pattern to form the constructed data for subsequent manipulation. Alternatively the dataset 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 graphical display. For instance, highlighting a group will allow the user to answer the question

'Has the visualization method placed these objects as I expected it to?' or 'On what basis is the visualization method placing these objects together?'. Similarly the answer to a query of a database of houses for sale, such as: 'Which houses have five bedrooms and a garden?' allows the user to question whether the placement of these houses is expected or not.

6.3.4 Insertion of Landmarks

Definition 14 *The addition and/or highlighting of one or more entities, within a dataset under consideration, to provide points of orientation, is described as* ***the insertion of landmarks****.*

In the customer/fruit example, assuming a set of real data collected about the customers, the user adds three customers: one who buys 10 apples only, one 10 pears only and one who buys nothing. The user can then see which customers are closest to their constructed customers.

Landmark and query overlap as concepts, whilst query relates to an action for which there is an answer, the insertion of a landmark adds a point or group for the purpose of orientation. Thus the highlighted entities may be left as landmarks in the display, but new entities may also be invented. An example, in the case of looking at houses available on the market, could be to highlight the most expensive house (highlighting one of the members of the dataset) or include one's ideal house (including a new entity).

6.3.5 Elicitation and Application of Feedback Data

Definition 15 *For the* ***elicitation*** *of feedback data, the user arranges a set of objects that are known to them on the screen. Real-world data is also available for these objects. The objects are known to the user in the sense that the user has a personal view of some (or all) of their qualities and can arrange the objects on the screen according to their own perceived sense of similarity between objects. The system* ***applies*** *this feedback data by using the proximities 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.*

In the customer/fruit example, if it is enforced that customers 1,2 and 3 are grouped together by the store manager deciding the buying behaviour of these three is similar (by arranging them close together on the screen or by some other means), the visualization application then derives that 5 and 6 (say) are also similar, but 7, 8 and 9 are not.

The initial inspiration for signature exploration, and especially for this feedback technique, came from work on dynamic querying of image libraries (e.g. Chang and Fu (1980); Pu and Pecenovic (2000)), as described in the introduction to this work on page 6. Here the user may choose a selection of images and then see how the application arranges them in terms of similarity, so they can better

understand how the application responds when asked to return, from the database, images similar to a particular image. This is an example of signature exploration, since the user is given insight into the behaviour of the algorithm by seeing how some known data is arranged. However, further development suggests itself: 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. This can be regarded as signature modification using feedback data.

Concepts of similarity may be very subjective, as is particularly clear in the case of comparison of image data. To some extent many comparisons have a subjective aspect, either from the point of view of the user's particular inquiry or from their perspective. The user may also be unable to articulate, or even be aware of, relevant domain knowledge that they have. In feedback exploration, the user is asked to position a number of familiar objects 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 objects, so that the system can derive 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. The simplest application would select the algorithm which gave the layout closest to that specified by the user.

6.4 Relationship to Existing Work

Apart from the work already referenced in the use of image libraries and spectrometry (page 6), contemporary visualization systems contain many elements for assisting the user's exploration of the data (as illustrated in Chapters 3 and 4; general references: Card et al. (1999); Chen (1999); Spence (2001)). Such features include: brushing (Cleveland and McGill 1984; MacDonald 1990) and the use of multiple linked views (Roberts 2004); for context and focus control (Furnas 1981; Lamm et al. 1996; Rao and Card 1994); querying of data with conventional database query language and dynamic querying within the visualization itself (e.g. Attribute Explorer (Spence 2001)); visual selection and reordering of that data, for example in the context of a colour map (e.g. Ankerst (2001)) or directly from a data table. These features promote the exploration of both the data and, intrinsically, the visualization method. Signature exploration focuses not on the data itself, but on the visualization method'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 datasets, indicated above, fall within the scope of the signature exploration concept, so that, whilst the examination of signature exploration has suggested new techniques and a framework for the increased comprehension of complex data visualization, it is also a reframing of much existing work.

6.5 Implementation

Example interfaces for the five approaches have been developed by extending an existing tool for visualization, the *Space Explorer* visualization environment (Schroeder and Noy 2001; Schroeder et al. 2001) and by other testing. Space Explorer contains a number of clustering and visualization algorithms implemented in Java and the Virtual Reality Modelling Language (VRML). VRML is a mark-up language which allows one to specify 3D worlds, which can then be displayed and explored in any web-browser with a VRML plug-in. In the original application, users can load multivariate or proximity matrices. PCA, PCoA, spring embedding, distance metrics of various kinds and hierarchical clustering algorithms with different linkage methods can be selected as appropriate. One, two or three dimensional output can be requested, the 3D representations are presented within an interactive VRML world. A set of prototype interfaces for this application was developed which provides facilities for signature exploration as menu items; these are described in the next four chapters (query and landmark are combined). The investigations described in these following chapters are partly presented within these interfaces, dynamically (upon request) and partly as a result of manually changing the data (this will be made clear in the relevant chapter descriptions).

The original application was rewritten using the Java Swing[1] library, because a spreadsheet window was required and this would be easier to implement with Swing, as well as having the benefit of additional functionality. The architecture of the original application was changed to provide an initial componentization including a template for the addition of new display types as they were developed. Significant additions or changes to the original application are[2]:

- Updating to Swing of the functions for multivariate data and the interface. Only the functions required for this work were included (PCA and distance metrics followed by PCoA).
- Change from a form to a menu-based application.
- Spreadsheet application - to enable the viewing, highlighting, changing and entering of numerical data directly into a table.
- Bar chart implementation.
- Display windows chosen by the user one by one, rather than the selection being hard coded and all appearing at the end of the data transformations.
- Extending brushing and linking to allow connection between the display window and the data table, in both directions, i.e. so that changes in the table window result in changes in the display

[1]The Swing components are part of the Java Foundation Classes (JFC) which encompasses a group of features to help the building of graphical user interfaces. The JFC was first announced in 1997 - updating the AWT (Abstract Windowing Toolkit).

[2]Some of these modifications will become clearer after descriptions in subsequent chapters.

window and vice versa (where this is appropriate, that is where the mapping is direct and does not involve dimension reduction).

- Feedback interface in the scatterplot display for multivariate data, to allow users to move the entities to create their own arrangement and capture the distance measurements. (This is a generalization of the 'backward' brushing and linking between scatterplot and multivariate data table).
- Partial componentization to facilitate addition of new displays.

Note that it was beyond the scope of this project to produce a complete, finished application. The extensions and modifications above were made in order to carry out the experiments to examine the techniques of signature exploration, limited to the examination of *illustrative* applications for each technique, to maximise the benefit as mentioned in the introduction (page 10). Thus the resulting application operates a subset of the overall functionality of the original application, whilst also making additions.

6.6 Summary

This chapter begins by summarizing the problem of visualizing complex data and indicates the three main aspects that need comprehension support: mathematical transformation; graphical representation; each depiction as 'one view of many'. These three aspects are a categorization of the key issues that present obstacles to comprehension identified in the previous four chapters. Mathematical transformation and graphical representation are seen as one by the user. The information seeking mantra of 'overview, zoom, then details-on-demand', encourages an interactive process. However, there is still a need to prevent a static one-view solution to visualizing complex data, particularly for the overview phase. Thus the two main goals for the designer who wishes to optimize the user's comprehension are: avoid the one-view graphic; enhance the user's understanding of mathematical transformation and graphical representation. The second of these is the focus of this work, though it serves the first in part.

The proposal of signature exploration is based upon the intuitive desire to take known data and put it into the visualization process to see what happens - to use datasets that are known in some way to explore the behaviours, or signatures, of the different visualization techniques. In this way the user can gain insight into how features in the data map to patterns in the visual representation. Signature exploration is defined as the exploration of the behaviour of a visualization method by means of the visualization of specially constructed datasets, which contain, or are representative of, particular features of interest. *Signature* is used to mean a distinctive mark or pattern indicating a dataset feature for a particular visualization method.

Five techniques involving the production or provision of constructed data containing a feature or features of interest are presented: generic dataset provision; user-construction of data; querying; insertion of landmarks; elicitation and application of feedback data. For generic dataset provision, characteristic sets of data that illustrate the various behaviours of metrics and visualizations are provided within the visualization application. User construction of data involves the user creating their own data from scratch or by transforming an existing dataset, or by a simulation based upon a user-specified set of behaviours. Querying relates to a real dataset under consideration; it results in a subset of the data being selected by visual or SQL-type query, so that the user can examine the way this subset is depicted by the visualization process. Landmarks are inserted for orientation within the graphic; they may be selected from within the dataset under consideration or be synthetic additions to the dataset. The elicitation and application of feedback data captures and applies the user's knowledge of a subset of entities from a dataset under consideration.

This work builds upon work of the last ten to twenty years that contains many elements for assisting the user's exploration of data, such as brushing, for focus+context control, dynamic querying and visual selection and reordering of data. Signature exploration reframes this work by placing it within the context of understanding the visualization process itself, rather than the data, and by establishing further techniques to this end, together with a framework for the design of complex data visualization applications that optimize user comprehension.

The exploration of the concept of signature exploration has taken place within the ongoing development of an existing visualization environment, Space Explorer. The application was updated and reorganized to provide a template for new displays and additional functionality including: a spreadsheet, bar chart display, brushing and linking between windows (including the datatable window) and a feedback interface for the dimension reduced scatterplot. Facilities for signature exploration are included as menu items.

Chapter 7

Generic Dataset Provision

7.1 Introduction

The previous chapter has defined generic dataset provision as the provision within the visualization application of *characteristic sets of data that illustrate the various behaviours of metrics and visualizations* (Definition 11 on page 80). This is to provide the user with a range of datasets showing specific features to demonstrate how these features appear in the pattern in the resultant graphic, how the features map to patterns in the graphic, with the aim of saying (if possible), 'If I see pattern x in the graphic, it means that the dataset contains feature y.' There are two scenarios: to examine a single visualization method using several datasets; to compare different visualization methods using the same dataset. These are illustrated in Figure 7.1.

Immediately a number of problems present themselves. There are so many datasets, how is one to choose representative ones? Will the result (how the feature maps to a particular pattern) hold only for that *specific* dataset or can the result be generalized? How can the *feature* in the dataset be specified or classified. can it be measured? How can the resultant pattern be specified? In order to measure how well the feature is shown in the graphic, can the resultant pattern be measured? Regarding dimension reduction situations, the information loss, or abstraction, involved indicates that more than one dataset will result in the same graphic; what are the implications of this?

Despite the difficulty of answering these questions (which are returned to in subsequent sections of this chapter), we can procure or devise datasets that show general features such as:

- 'Structureless' datasets based on pseudo-random variables with, for instance, uniform and Gaussian distributions.
- Clusters of various numbers and types, alone or placed within the datasets of the first item.
- Datasets containing specific *entity-entity* structures, i.e. structures that concern relationships

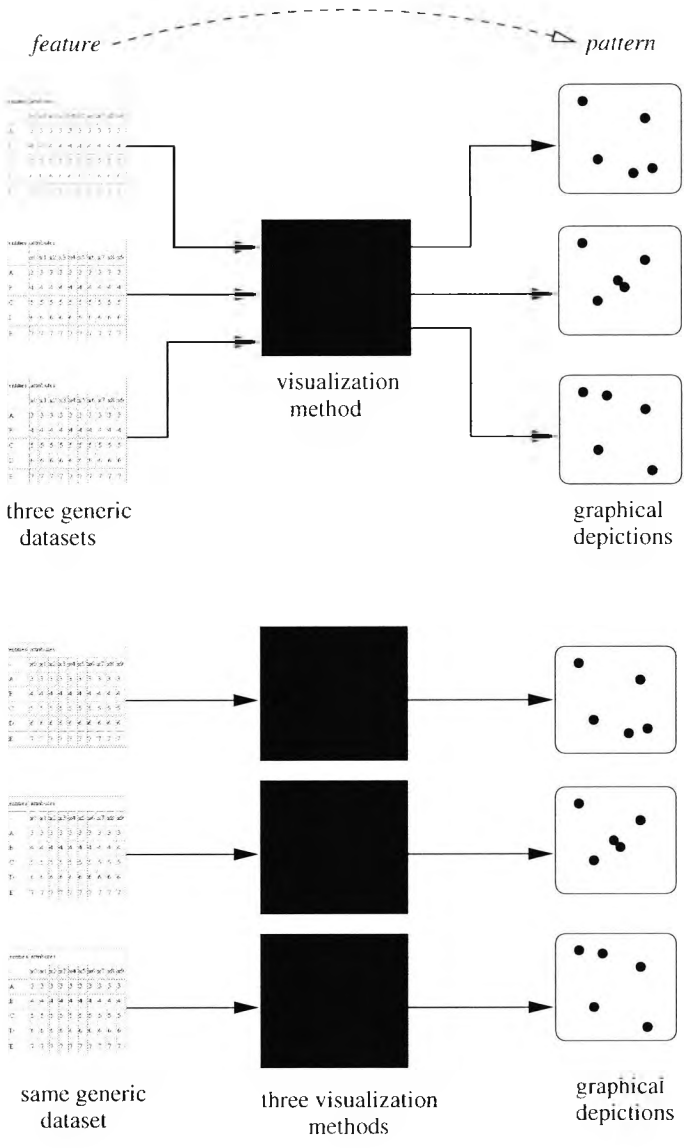


Figure 7.1: Generic dataset provision. Two scenarios: examining a single visualization method using several datasets (top); comparing different visualization methods using the same dataset (bottom). The visualization method is treated as a black box with the aim of seeing how features in the dataset, which can be seen directly in the table form, map to patterns in the graphic. Some other representation for the dataset may be used, depending upon the type of data, for instance, for time series data, line plots may be more appropriate. Line plots are used in addition to tables in the feasibility test in Section 7.3.1.

between the entities, such as[1]:

- scaling (amplitude, phase and frequency types)
- showing patterns of interest (such as overall behaviour across variables)

- Datasets that contain examples of correlation between variables[2].

Though it may not be possible to quantify these features in all cases, nor to specify them, formally, in mathematical terms, it can be said that such features exist. With such data we can then:

- See how different visualization methods display these data.
- Give a subjective assessment of how well a feature is mapped.

On an intuitive level one might consider the following: as a designer creates a new graphical depiction, they supply to the user within the application, where feasible, the test datasets with which they explored and validated their representation, which also reveal any artefacts. An artefact in this situation being an apparent pattern in the data that resulted from the visualization method itself, as distinct from an artefact of the data collection method. The sense of pragmatism in which the designer chooses representative datasets with which to test their application can also guide our approach to the choice of representative generic datasets.

Again, (as mentioned in the previous chapter on page 78), note that one approach to understanding the visualization method would be through understanding the process in the black box itself, perhaps by algorithm animation, but this does not directly address the question of how features in the dataset map to patterns in the graphic. Also, particularly in the case of complex transformations, it may still fall short of allowing the user to predict patterns in the representation.

Generic dataset provision entails establishing a set of datasets that contain features of interest and can be described as generic in the sense that they are representative of these features. This concept is close to the idea of *benchmark* datasets, but not quite the same. The purpose of benchmark datasets is to establish a recognized set of datasets for assessing (or measuring, if appropriate) the characteristics of a particular visualization method, primarily for the *designer's* benefit. Whereas generic datasets are proposed for the benefit of the user. Thus generic datasets, if identified, may be useful as benchmark datasets, but the aim in this work is not, specifically, to establish benchmark datasets. The significance and potential of the findings of this work vis-à-vis benchmark datasets is examined in the concluding chapter of this thesis (Chapter 13).

In examining the possibilities and application of generic datasets, the approach taken here is to first look at how various aspects of the issue appear in the literature, and, based upon this, then

[1] Examples of these are given in subsequent sections of this chapter.
[2] Correlation becomes an entity-entity relationship if objects and attributes are swapped.

create representative datasets. A feasibility test was carried out to see whether this approach would assist with the understanding of the *duck's leg and webbed foot* pattern from the calldata (page 7) (corresponding to the first scenario in Figure 7.1). Also undertaken was an examination of the choice of dimension reduction method for an application in the software agent domain, to identify *like-minded* agents (corresponding to the second scenario in Figure 7.1). This led to the proposal of a new technique for use of profile data which is described in Section 11.5. Thus the following four sections cover: relevant literature; specification of datasets and description of tests; results and conclusions.

7.2 Literature Background

The literature has been examined from the point of view of finding suggestions for datasets to use, as well as creating datasets that contain specified features, such as outliers or relationships between entities. Books of collections of datasets are available, as well as on-line repositories. The classification literature examines types of datasets, including suitability of clustering methods, validation of clusters and null models (of data which lack structure in some sense). The information visualization literature reports work to identify test datasets for evaluation of information visualization systems. These areas are described below.

7.2.1 Collections of Datasets

In the introduction to 'Small Datasets'[3] by Hand et al. (1994), the use of synthetic data is criticised:

> 'If data purporting to come from some real domain are invented... there is the risk of misleading - it is in fact quite difficult to create realistic artificial data sets unless one is very familiar with the application area.'

The authors indicate how hard it can be to find real datasets and this is part of the rationale for the book. Nevertheless, this book does contain a couple of synthetic datasets, including Anscombe's correlation data - four synthetic two-dimensional datasets with correlation coefficients and regression lines the same, but very different scatter diagrams (Anscombe's data are discussed in Tufte (1983, p. 13)). The question of the validity of the use of synthetic data would appear to be an issue relating to the nature of the feature of interest. An outlier can certainly be successfully synthesized for the purpose of seeing whether a particular visualization method reveals it. At the other extreme, data from a complex system exhibiting emergent behaviour cannot, in general, be synthesized[4]. In a sense, emergent behaviour is a meta-level feature compared to that of an outlier. Thus it would

[3]The two collections of datasets examined here are recommended by Webb (1999, p. 381) for assessing statistical pattern recognition methods.

[4]For this reason we used the calldata to represent an agent system...see later description. Data showing emergent behaviour can be synthesized when the mechanism is known, as in cellular automata, but arguably these may then be considered synthetic.

appear valid to conclude that, in choosing representative datasets, synthetic and real-world datasets are both appropriate, depending upon the feature of interest.

Another book of datasets, 'Data: a collection of problems from many fields for the student and research worker' (Andrews and Herzberg 1985), shows a broad collection of sets of data from a large number of situations and points out that in many cases, different forms of statistical analysis lead to different conclusions, thus facing the reader with the challenge of finding an appropriate analysis.

In examining both collections of data, no obvious candidates for generic datasets were found, apart from those exhibiting a certain number of clusters, such as the well-known *Iris* dataset. The Iris dataset (Fisher (1936) as described and reproduced in Andrews and Herzberg (1985)) is used widely to illustrate new representations. The dataset contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other two; the latter are not linearly separable from each other. This is valuable in the situation where the user knows this dataset and has seen many other representations of it with which to compare. For the inexperienced user, however, and from the point of view of a search for datasets containing specific features, it is not an obvious starting point. It is possible that the lack of suitable datasets in these collections arises because the datasets available reflect the desire for datasets with which to test different statistical methods, rather than for illustrating and testing the behaviour of graphical representations to the user[5].

7.2.2 On-line Data Sources

There are numerous on-line repositories of datasets. Some of the datasets that are available have been used for numerous studies and a lot is known about them. Like the books of datasets (some of which appear in the electronic sources), they contain almost exclusively real-world datasets, so cannot provide examples for the generic type under examination here. Some datasets are also (unlike the datasets in the two books above) too large for the display in a table. Indication of content of some of these on-line repositories of datasets is shown in table 7.1. These repositories were examined (the UCI Machine Learning Repository, by examining the descriptions of all the datasets, the other repositories by scanning a selection), for suitable datasets to use here. However, the view formed by examining the sources in books, described in the previous section, was confirmed, that there were no obvious candidates for generic dataset provision[6].

[5] In retrospect, this still seems odd, perhaps the concept of a feature was too narrow. At any rate, the datasets did not appear to contain the features that were of interest at that time.

[6] More extensive examination of available datasets may lead to a different conclusion. Also, further experimentation with candidate generic datasets may reveal the importance of features that are illustrated by some of these datasets, though such features were not considered here.

Repository Name and Web Address	Comment
UCI Machine Learning Repository www.ics.uci.edu/ mlearn/MLRepository.html	Over 70 databases that are used by the machine learning community for the empirical analysis of machine learning algorithms. Recommended by Webb (1999).
UCI Knowledge Discovery in Databases Archive kdd.ics.uci.edu/	Large datasets: wide variety of data types, analysis tasks, and application areas to enable researchers in knowledge discovery and data mining to scale existing and future data analysis algorithms to very large and complex datasets.
Biz/ed: for students and educators in business and economics www.bized.ac.uk/dataserv/datahome.htm	Hosts both original and mirrored datasets for economics, business and finance.
Data and Story Library lib.stat.cmu.edu/DASL/	Datafiles and 'stories' that illustrate the use of basic statistics methods.
Journal of Statistics Education - Data Resources www.amstat.org/publications/jse/jse_data_archive.html	Data for use in teaching statistics.
UCLA Statistics Data Sets www.stat.ucla.edu/data/	Includes the sets of datasets from the two books described above (Andrews and Herzberg (1985); Hand et al. (1994) and others for courses and from other books.

Table 7.1: On-line repositories of datasets. A list containing these and other data sources on the web is given by CTI Statistics at www.stats.gla.ac.uk/cti/links_stats/data.html.

7.2.3 Admissibility Criteria and Clustering Validity

Fisher and Van Ness (1971) and Van Ness (1973) suggest an admissibility procedure for clustering algorithms. Their starting point is that it is usually impossible to determine a 'best' clustering procedure. They suggest the formulation of properties which any reasonable procedure should satisfy and call a procedure satisfying them *admissible*. The aim is to eliminate obviously bad clustering algorithms, but is not an attempt to specify the best method. Fisher and Van Ness describe this as follows:

> 'Let A denote some property which should be satisfied by any reasonable procedure either in general or when used in a special application. Any procedure which satisfies A is called A-admissible.'

The properties that are used relate to three different aspects: the resultant clusters; the structure of the data; the consistency of the result when changes are made to the data. For instance, if the resulting clusters have convex hulls which do not intersect, (i.e. clusters do not cut through one another), the clustering procedure is described as *convex-admissible*. The properties relating to structure cover *well-structured (exact tree)*, i.e. having an exact tree structure, and *well-structured (k-group)* if there exists a clustering $C_1, \ldots, C_k$ such that all within-cluster distances are smaller than

all between-cluster distances. The properties relating to changes to the data (transformations and additions) include duplications of single entities in a cluster (*point proportion admissible*), duplications of all entities in one cluster (*cluster proportion admissible*) and removal of all entities in one cluster (*cluster omission admissible*).

This work is not directly applicable for generic dataset provision as it concerns the behaviour of clustering algorithms, but the concept of admissibility and the specification of properties, together with formal definitions, constitute an overall approach that is useful here.

There are many references to the difficulty of choosing methods for classification and clustering (for example, Gordon (1999); Webb (1999)). The process of clustering has been described as a system for generating hypotheses (Williams and Dale (1965) as cited in Sneath (1997)) and this description is also apt for visualization, regardless of whether clustering algorithms are employed. Nevertheless, in practice the user will not wish to validate *all* hypotheses about clusters or visualization, so that it is useful to have 'guidelines to what can be accepted with confidence' (Sneath 1997).

However, an example of the difficulty of using guidelines is in assessing the validity of PCA by examining the eigenvalue tail-off (Section 3.4.2). If the eigenvalues tail off fast (within 3 or 4), the truncation of the matrix is considered valid. A guideline is to say that, if three quarters of the information is present (by truncation at a particular point), this indicates validity. It is often the case that the eigenvalues do not tail off and this is taken as an indication of lack of structure in the data. However, examples have been observed where a satisfactory structure of the data was obtained even if the eigenvalues indicated otherwise (Lebart et al. 1997, p. 57). An intuitive explanation of this is that 'noise' is present in the data. We can recreate this by first creating an evenly distributed random variable space and putting within this Gaussian clouds (Keim et al. 1995).

Procedures for evaluating the results of a clustering algorithm are described as cluster validation and tests are defined to measure:

- the complete absence of class structure (the null hypothesis, a statement of random structure of a dataset)
- the validity of an individual cluster
- the validity of a partition
- the validity of a hierarchical classification

This absence of class structure is an immediate candidate for generic data, since this will show how the representation appears with a dataset lacking structure and will, at the same time, reveal any artefacts that are the creation of the representation itself. The validity of clusters[7], partitions and

[7] In the case of clusters there is a further issue concerning the number of clusters requested, since many clustering algorithms require this to be decided by the user. Thus the validity of the number of requested clusters needs to be examined as well.

hierarchies is measured numerically by various methods; these measures may help the user to assess the visual depiction where clustering has been used, but for the purpose of this work, for simplicity, it is assumed that this assessment is done visually, at least partially. Thus the integration of cluster validation information with the visualization is not examined in this work, though such things as isolation and cohesion of clusters would be useful measures to put figures on what we can see.

7.2.4 Null Models

A number of null models for absence of structure have been described for pattern matrices, dissimilarity matrices and tree diagrams (See e.g. Gordon 1999). The Poisson model assumes that objects can be represented by points that are uniformly distributed in some region, A, of p-dimensional space, either the unit p-dimensional hypercube or hypersphere or the convex hull of these. Two things are identified in this description - the region within which the points are located (i.e. the overall 'shape' of the region A) and their distribution within this space. In the unimodal model the joint distribution of the variables describing the objects is unimodal, usually a multivariate normal distribution with identity covariance matrix[8] has been specified. In the random permutation model, the entries in each of the columns of the pattern matrix are permuted, ignoring correlation between variables.

7.2.5 Datasets for Evaluating Visualization Systems

A workshop, entitled *Perceptual Issues in Visualization* (Grinstein and Levkowitz 1995), was held at the IEEE Visualization '93 conference. The first of its kind, this workshop included a subgroup examining the topic *Test Data Sets*. An approach suggested by this group is to use datasets that are created by combining *noise* with *embedded stimuli* (Keim et al. 1995). The noise is equivalent to a random dataset and the embedded stimuli are clusters within this as introduced in Section 7.2.3. Such artificial datasets are used to test whether the embedded stimuli can be observed in visualizations. Specifying datasets that contain specific (or absence of) correlation between variables is also included, though the authors concede that this is a difficult problem and only tackle two variates.

Turton et al. (2000) presents synthetic data generators, including the spatial element and time, for testing space-time and more complex hyperspace geographical analysis tools (not visualization tools). Keogh and Pazzani (1999) describe different 'global distortions' in time series data that are relevant to users, these are: offset translation; amplitude scaling; linear drift and discontinuities. These are described in detail in Chapter 10 as they relate to the use of feedback data[9].

[8]The covariance matrix shows the covariance between variables, in addition to the variance of individual variables. The unity covariance matrix has zero values for covariance between variables, which means that there is no linear correlation between them.

[9]Since completing this work, Keogh has published further work that is relevant on the need for time series data mining benchmarks (Keogh and Kasetty 2002).

7.2.6 Statistical Specification

It would be useful to be able to measure the features that are of interest to us in the dataset and measure the amount that the feature is revealed in the representation. Can statistical measures help? One can consider statistical measures to be, in themselves, features of interest, but statistical measurements can be the same for very different sets of data as illustrated by Anscome's data (Anscombe 1973), discussed in Section 7.2.1, for which reason Tufte says:

> "Graphics reveal data." - *The Visual Display of Quantative Information* (Tufte 1983)

On the other hand, if transformation including abstraction is involved in the visual representation, this characteristic of revelation is somewhat lost and must be compensated for by the use of a combination of measures (statistical and other) and special visualizations (such as of error, see Section 11.3), so that the relationship between visual representation and statistics (as well as other measures) is somewhat circular. Note also that statistical measures are abstractions in themselves, involving another layer of representation for the user to understand. The issue of how to *statistify* visualization applications is stressed by Unwin (2001), but is not the main concern of this work, which is (at this stage) concerned with revealing the behaviour of visualization methods with respect to features that are known in a more concrete sense, i.e. by direct examination of the data in the form of a table or direct visualization such as a line plot.

7.2.7 Visual Languages

Since it is difficult to mathematically specify, both qualitatively (to some extent) and quantitatively, the interesting features in datasets, it is possible to consider the use of a visual language (for three relevant papers see Bottoni et al. (1998); Narayanan and Hübscher (1998); Wang and Zeevat (1998) in Marriott and Meyer (1998)) for the user to express, for instance, their estimation of similarity between entities. This may be merely an inadequate placeholder for a superior, more rigorous, formal analysis, yet-to-be developed, but it may also have benefits in terms of engaging the user in examination of the algorithm behaviour and thus increasing their understanding. Intuitively it appears that much comprehension of a particular visualization system could be obtained by extensive study of the algorithms used and any mathematics involved, as well as studying many problems with the tool. However, the first obstacle here is to obtain the willingness or eagerness to carry this out. The possibility of aspects of the interface *engaging* the user in this process (which they may otherwise not engage in at all) is important. Thus, such a visual language need not be proved important only in its efficacy with respect to representing the desired dataset features, it can have value in engaging the user in a process. Arguably it is risky to add another level of abstraction at this point, but the fact is that the desired features do need to be measured in some way, if one is to move beyond the

classification purely as presence/absence.

7.3 Choice and Specification of Datasets

The previous section has shown that collections of published datasets do not contain suitable datasets for the mapping of specific features in the data to patterns in the visual depiction, except for examples of clusters of various numbers and types, so that it is necessary to construct synthetic datasets. The following categories for synthetic generic types were identified:

- Null modes - lacking structure, or minimal structure in some sense (necessary to be able to identify what the lack of structure looks like for various methods).
- Clusters within noise (different types of data) (corresponding to the 'noise with embedded stimuli' of Keim et al. (1995)).
- Containing specific features including:
  - Overplotting due to identical or closely similar entities (i.e. features identified as problematic for particular visualization methods).
  - Outliers (of interest in most fields).
- Inter-element features (of particular relevance where dimension reduction clusterings are examined) including:
  - scaling
  - phase shift or modulation
  - amplitude modulation
  - frequency modulation

The type of data has been restricted to multivariate data tables, from which proximity data may or may not be derived. These restrictions are from two points of view: for simplicity and to provide focus.

In general it is considered that datasets can be *known* on three levels:

1. Where the actual data values are known precisely (indicating the dataset size is small).
2. Where some statistical features about the dataset are known (which allows the dataset to be large), but note Anscombe's problem (see Section 7.2.6).

3. Where the knowledge is intuitive, *tacit* knowledge. Tacit knowledge is loosely defined as knowledge that is not written down, but in people's minds. A tighter definition specifies knowledge that is in the mind, but that the person is unaware of[10]: such knowledge is of two types according to how it was acquired - compiled knowledge and implicit learning. Compiled knowledge is derived from non-tacit knowledge as in the learnt skills of driving and typing. Knowledge derived from implicit learning has been learnt unconsciously and is hard to get at[11]. Elicitation techniques are used to collect tacit knowledge in requirements elicitation and usability assessment (of websites for instance). Tacit knowledge and elicitation techniques are returned to in Chapter 10.

This investigation begins with the first in this list, small datasets, to satisfy the requirement of concretely knowing the data. Datasets lacking structure and those containing specific features are examined. The small size makes the creation of a random dataset problematic, because a small portion of a random dataset will always show some apparent structure, for instance areas of less density, and thus a degree of clustering is implied. The datasets are examined with the various algorithms in Space Explorer - to see if a distinct mapping of features is indicated or refuted for the different algorithms i.e. whether an informal *feature-admissible* classification can be made. This is illustrated in the following two sections describing respectively: a feasibility test; the integration of generic datasets within the user interface. The feasibility test involves users exploring the behaviour of a single algorithm with different datasets, the integration of datasets within the user interface allows the comparison of behaviours of different algorithms. The elicitation and use of datasets based on intuitive knowledge is left to the investigation of constructed and feedback types of signature exploration (chapters 8 and 10), since such data cannot, by definition, be generic.

7.3.1 Feasibility Test

Do our visualizations actually work? This question was asked in a keynote speech at a recent conference (Robertson 2000) and statistics from conference papers given that showed less than 10% had carried out evaluation. Informal testing in the early stages was recommended and our feasibility test 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 web pages contained embedded VRML 3D representations of data that the participants could manipulate. The test first illustrated the problem by displaying the visualization of the call data of Figure 1.1 on page 7, 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

[10]The tighter definition is used in the HCI community and the looser one in the field of knowledge management.
[11]Descriptions taken from the ACRE categorisation of memory and communication types (Maiden and Rugg 1996)

3D visualizations of simple datasets followed (using the same algorithm - Euclidean distance calculation followed by layout with PCoA using Space Explorer). The data tables were shown, together with the data displayed as time series line plots. Figure 7.2 shows the feasibility test website[12]. Figures 7.3 to 7.8 show the six subsequent webpages of the test comprising one page of further details and five example generic datasets. The test took each participant about 1 hour to complete, including discussion.

The datasets illustrate constant, linear and complex pattern shapes across the variables and include scaling and displacement features. These datasets were chosen to cover the inter-element features scaling, phase shift and amplitude modulation as these are features of interest in comparing entities in a dimension reduction situation such as the one here. These answer the question 'promimity represents similarity, but similarity in what sense?' Phase shift is relevant in time series data, such as in examining biological or financial data, where a pattern of behaviour over a particular time period is repeated at a later time point by another entity. Phase shift also has meaning for non-time series data where all variables are of the same type (as in the calldata where all variables are destinations); here it corresponds to similar behaviour across the variables, but irrespective of particular variables. The particular examples of the data for each example were chosen so that the user could get a good impression of the data from the line plots.

Most of the questions in the questionnaire were to guide the exploration of the material. The key questions at the end were:

1. Do you think these explorations of constructed datasets 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 datasets or use the ones 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.

Observations from the feasibility test are shown in table 7.2. Each visualization showed a recognizable pattern relating to the feature in the illustrative dataset. The orthogonal, or approximately orthogonal, relationship between the loci of scaling and means, is interesting, because it provides indicators to the meaning of direction in the space, see Figures 7.9 and 7.10. Also the approximately circular nature of the phase shifted pattern. These, however, were not enough in themselves to assist the user to interpret the webbed foot problem of the test.

[12]The questionnaires filled in by participants were unfortunately destroyed in the serious fire that occurred at City University in the spring of 2001, which destroyed paperwork in our office.

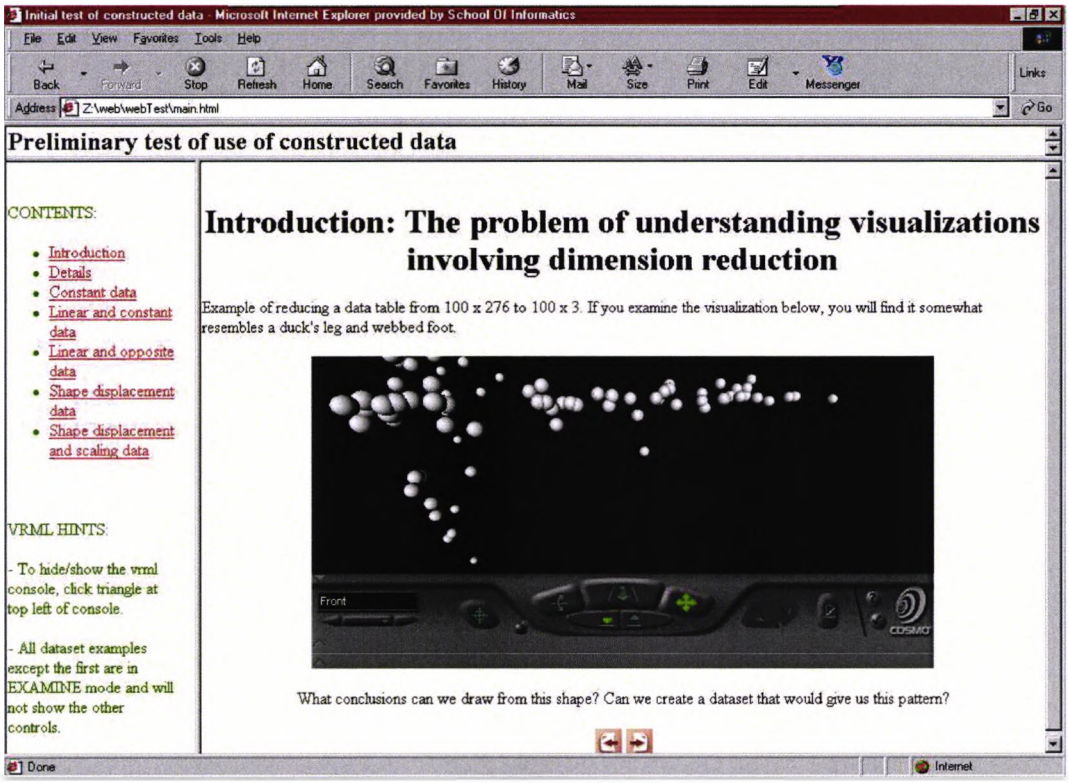
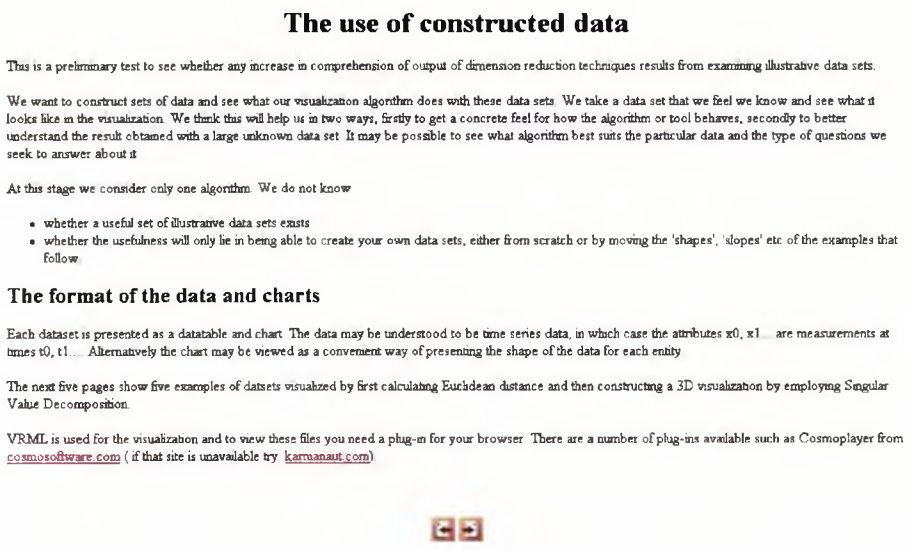


Figure 7.2: Website for feasibility test: opening page.

7.3.2 Adding Generic Datasets to the User Interface

A dataset for seven entities was created, with each entity having seven attributes, using Java's uniformly distributed pseudo-random number generator (see for example Weiss (2002, Chapter 9.) for a discussion of randomization). Generic datasets are, by definition, domain-independent, but the data here is taken to be a dataset of seven agents with different levels of interest in seven subjects $a, b, \ldots g$ as shown in Table 7.3. This data is shown here because it was used in work applying generic dataset provision to the selection of metrics for measuring similarity between agent profiles, which is documented in papers presented to the agent and visual datamining communities, defining like-minded agents with the aid of visualization (Noy and Schroeder 2002b,a). Three other entities were added with attributes scaled with respect to a reference entity in the set. The scaling was done in different ways to give three datasets, see Tables 7.4, 7.5 and 7.6. These were arbitrarily labelled Scaling1, Scaling2 and Scaling3 in the Space Explorer interface to emphasize that they are examples chosen from many possibilities.

The datasets were integrated within the Swing version of Space Explorer so that selecting them would give the user separate windows containing: the data table in a spreadsheet and a stacked bar chart of the data (as opposed to a line plot of the data that was used for the time series data in the



The use of constructed data

This is a preliminary test to see whether any increase in comprehension of output of dimension reduction techniques results from examining illustrative data sets.

We want to construct sets of data and see what our visualization algorithm does with these data 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 this stage we consider only one algorithm. We do not know

- whether a useful set of illustrative data sets exists
- whether the usefulness will only be in being able to create your own data sets, either from scratch or by moving the 'shapes', 'slopes' etc of the examples that follow.

The format of the data and charts

Each dataset is presented as a datatable and chart. The data may be understood to be time series data, in which case the attributes x0, x1... are measurements at times t0, t1.... Alternatively the chart may be viewed as a convenient way of presenting the shape of the data for each entity.

The next five pages show five examples of datsets visualized by first calculating Euclidean distance and then constructing a 3D visualization by employing Singular Value Decomposition.

VRML is used for the visualization and to view these files you need a plug-in for your browser. There are a number of plug-ins available such as Cosmoplayer from cosmosoftware.com (if that site is unavailable try karmanaut.com).

Figure 7.3: The details webpage of the website feasibility test.

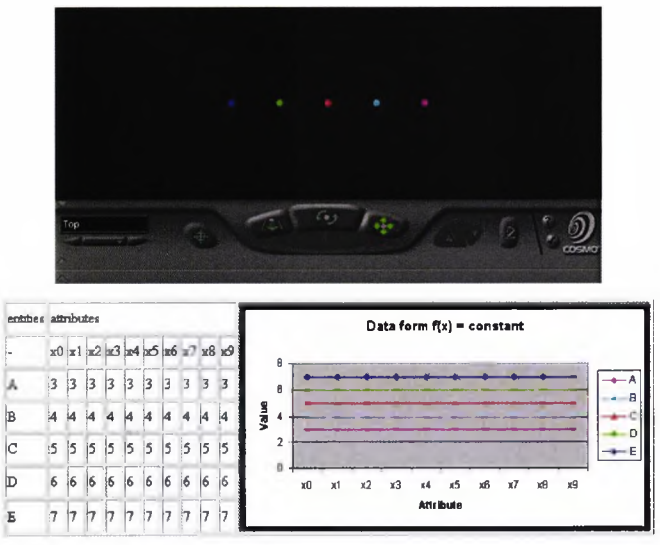


Figure 7.4: The first dataset webpage of the feasibility test shows constant data.

The data is of the form f(x) = kx + c : equivalent mean constant values are shown too ie D is the mean of B, C is the mean of A

entities	attributes									
-	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9
A	1	2	3	4	5	6	7	8	9	10
B	10	11	12	13	14	15	16	17	18	19
C	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5
D	14.5	14.5	14.5	14.5	14.5	14.5	14.5	14.5	14.5	14.5
E	2	3	4	5	6	7	8	9	10	11

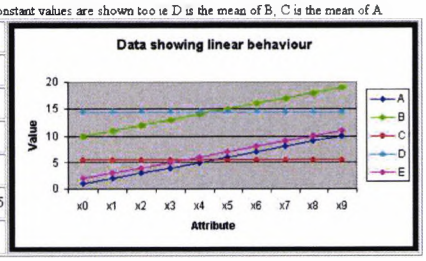


Figure 7.5: The second dataset webpage of the feasibility test shows linear data.

The data is of the form f(x) = kx + c : equivalent mean constant values are shown too ie D is the mean of B, C is the mean of A. A,B,C and D have the same values as the previous set. E is the reverse of A.

entities	attributes									
-	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9
A	1	2	3	4	5	6	7	8	9	10
B	10	11	12	13	14	15	16	17	18	19
C	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5
D	14.5	14.5	14.5	14.5	14.5	14.5	14.5	14.5	14.5	14.5
E	10	9	8	7	6	5	4	3	2	1

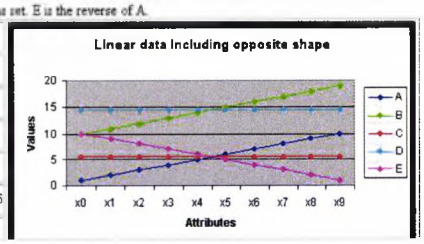


Figure 7.6: The third dataset webpage of the feasibility test shows linear data including reversed gradient.

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	20	25	23	16	6	5	4	1	2	17

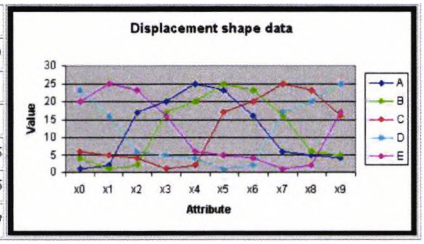


Figure 7.7: The fourth dataset webpage of the feasibility test shows complex shape with displacement.

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

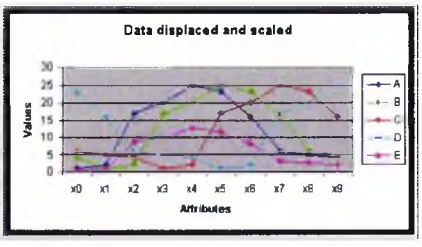


Figure 7.8: The fifth dataset webpage of the feasibility test shows shape displacement and scaling.

Relevant Figure	Data Description	Visualization Description (after Euclidean distance and PCoA)	Feature Admissible?
Figure 7.4	Constant data with scaling (equally spaced).	Entities in a line, equally spaced.	Yes, scaled output implies equality.
Figure 7.5	Scaled linear behaviour across variables with mean.	Scaled in a line, means are orthogonal. See Figure 7.9	Yes.
Figure 7.6	As previous entry with negative gradient.	Extends line in other direction. See Figure 7.9.	Yes.
Figure 7.7	Gaussian shape across variables, with phase shift.	Entities move in curve, approximately in a circle.	Yes, approximately.
Figure 7.8	Phase shift and scaled Gaussian.	*Scaled* moves out from the approximate plane of the *phase shifted*, but less than orthogonal. See Figure 7.10.	Yes, maybe.

Table 7.2: Descriptions of 3D representations using Euclidean distance followed by PCoA of the five datasets used in the feasibility test.

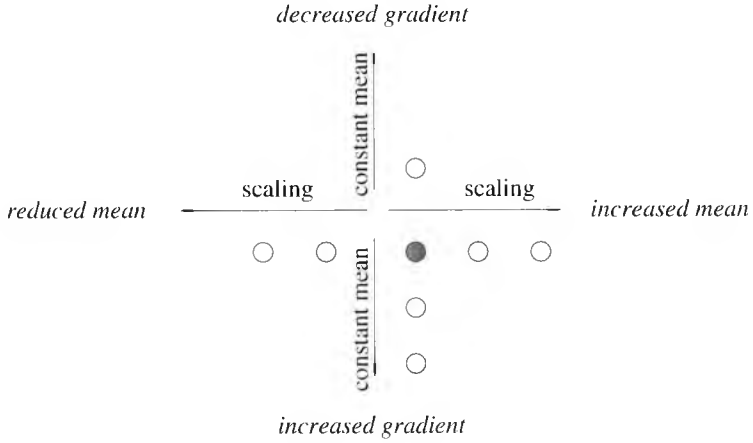


Figure 7.9: Orthogonal pattern obtained with change of slope (constant mean) and scaling with respect to the corner reference entity. See Figures 7.5 and 7.6, and Table 7.2.

	a	b	c	d	e	f	g
Agent1	9	3	4	6	5	5	5
Agent2	1	10	10	1	7	2	0
Agent3	4	1	6	8	0	5	7
Agent4	2	7	8	4	0	2	0
Agent5	3	6	4	7	1	10	6
Agent6	1	7	6	5	0	2	0
Agent7	8	1	7	1	2	5	9

Table 7.3: Random dataset for seven agents showing interest levels between 0 and 10 in seven subjects $a, \ldots g$.

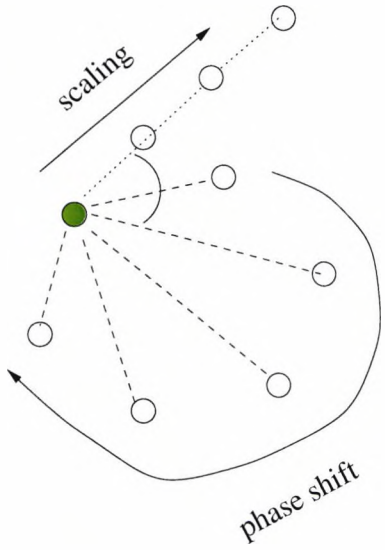


Figure 7.10: Pattern obtained for phase shift and scaling with respect to the corner reference entity (time series data - Gaussian shape).

	a	b	c	d	e	f	g
Agent1	9	3	4	6	5	5	5
Agenta	8	2	3	5	4	4	4
Agentb	7	1	2	4	3	3	3
Agentc	6	0	1	3	2	2	2
Agent2	1	10	10	1	7	2	0
Agent3	4	1	6	8	0	5	7
Agent4	2	7	8	4	0	2	0
Agent5	3	6	4	7	1	10	6
Agent6	1	7	6	5	0	2	0
Agent7	8	1	7	1	2	5	9

Table 7.4: Scaling1 dataset. This dataset has three entities (Agenta, Agentb and Agentc) added to the random dataset of Table 7.3, with values scaled with respect to Agent1. The scaling is by subtraction of 1, 2 or 3 from the values of Agent1.

	a	b	c	d	e	f	g
Agent1	5	5	5	5	5	5	5
Agenta	4	4	4	4	4	4	4
Agentb	3	3	3	3	3	3	3
Agentc	2	2	2	2	2	2	2
Agent2	1	10	10	1	7	2	0
Agent3	4	1	6	8	0	5	7
Agent4	2	7	8	4	0	2	0
Agent5	3	6	4	7	1	10	6
Agent6	1	7	6	5	0	2	0
Agent7	8	1	7	1	2	5	9

Table 7.5: Scaling2 dataset. This dataset is as Scaling1, but Agent1's values are constant across variables.

	a	b	c	d	e	f	g
Agent1	9	3	4	6	5	5	5
Agenta	1	7	6	4	5	5	5
Agentb	7	1	2	4	3	3	3
Agentc	9	0	0	9	3	8	10
Agent2	1	10	10	1	7	2	0
Agent3	4	1	6	8	0	5	7
Agent4	2	7	8	4	0	2	0
Agent5	3	6	4	7	1	10	6
Agent6	1	7	6	5	0	2	0
Agent7	8	1	7	1	2	5	9

Table 7.6: Scaling3 dataset. This dataset shows inversion (between 1 and a, and 2 and c) and scaling (b values are two away from 1's).

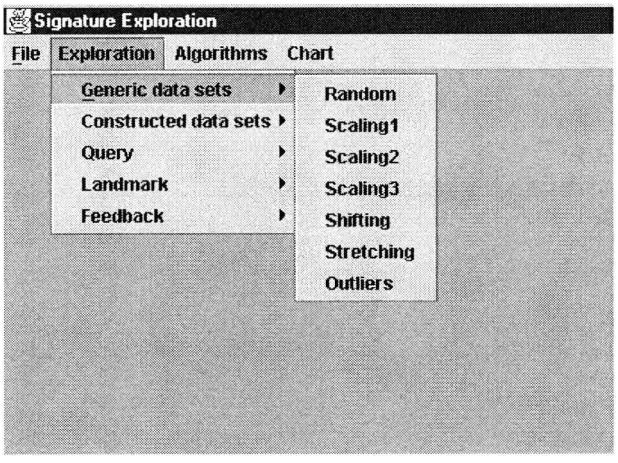


Figure 7.11: Opened menus showing the selection of generic datasets available.

feasibility test). From then the user could request different dimension reductions and view the results. The opened menu for generic datasets is shown in Figure 7.11 and a set of windows for one selection is shown in Figure 7.12. There were two aspects to this part of the experiment

- To illustrate in the interface of Space Explorer how generic datasets are intended to be used.
- To conduct informal testing by observing the results and in using the interface in the context of choosing metrics for measuring similarity between agent profiles.

An example of results obtained in comparing metrics is given in Table 7.7. In each case identical entities were hidden in the graphic due to overlap. Each method clearly showed the outlier. Entities with scaled attributes resulted in patterns obtained by the different algorithms that were similar for the Minkowski-based ones (Euclidean and City) and different for the angular separation (Figure 7.13). The scaling result can be examined in two ways. If scaled entities are shown in an identical position, as they are for angular separation, this has succeeded in the similarity sense, that is to say, angular

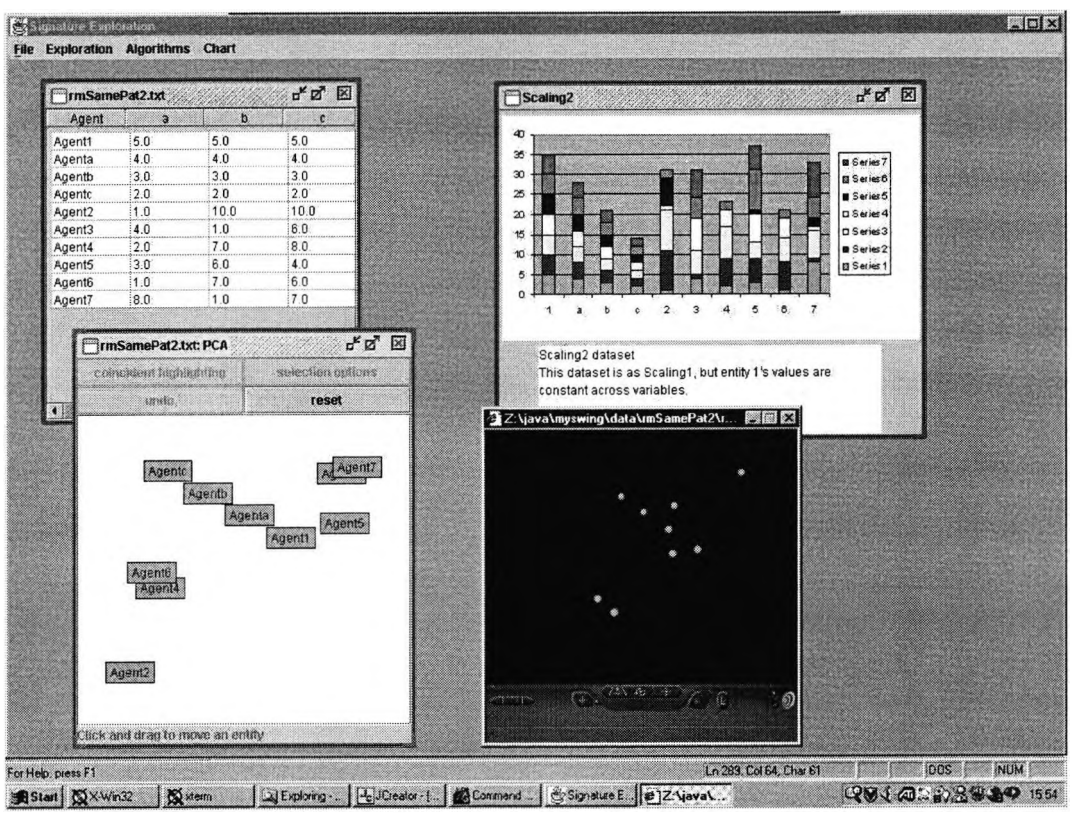


Figure 7.12: Set of windows for the scaling2 generic dataset with PCA applied: spreadsheet, stacked bar chart, 2D and 3D scatterplots.

Dataset	Metric + PCoA		
	City	Euclidean	Angular Separation
Identical entities	Not shown in any		
Outlier	Shown in all		
Random	Always shows apparent clustering		
Scaling within random. Screenshots in Figure 7.13.	Close and in a straight line, one further away.	Closer and in a straight line, equidistant.	All in same place.

Table 7.7: Example user observations for comparison of three algorithms using datasets including identical, outlier and scaled (with respect to a reference entity) entities.

separation has resulted in identifying the entities as very similar compared to the others. However, from the visual point of view, this is hidden by overplotting, so that City and Euclidean show it better, with Euclidean also reflecting the regularity of the differences by the entities being equidistant.

7.4 Conclusion

This section first presents conclusions from the feasibility test and adding generic datasets to the user interface, then discusses the problems of accuracy and evaluation, ending with a summary of conclusions.

7.4.1 Feasibility Test Conclusions

There are two aspects to the feasibility test: How useful does generic dataset provision appear from this test? What are the results of using these specific datasets to understand the visualization method used?

The feasibility test indicated that users may find the use of generic data helpful in understanding dimension reduction algorithms, and that they are interested in viewing the original data in a table and being able to construct and modify data in the table. However, the sample was biased in that the participants were all colleagues (2 lecturers and the rest PhD students) who were interested in the work. Another criticism voiced has been that users always desire greater interactivity, so that the question relating to this in the questionnaire was bound to elicit positive answers. Since this was an informal test, more criticisms can be made relating to the construction of the questionnaire and the website. Nevertheless, the test confirmed the overall feasibility of the proposal. At the same time it underlined the difficulty of the task of understanding visual depictions involving dimension reduction, since, though some of the participants felt that they had moved toward an understanding of the behaviour of the visualization method, none could explain the shape of the calldata depiction, despite the fact that several participants were familiar with the dimension reduction algorithm being

used.

The results from examining the set of visual depictions suggested that we may be able to assist in orientating the user in spaces where direction has no obvious meaning. Specifically, the observed orthogonal, or approximately orthogonal patterns obtained (for scaling and means, phase shifting and scaling: the general form of the patterns as in Figures 7.9 and 7.10) are interesting, because they suggest the possibility of orientating oneself in the space of the scatterplot, with respect to a particular entity. Thus the scatterplot xy (or xyz) axes, though not possessing intrinsic meaning (i.e. not representing a variable or quality in themselves), may have meaning with respect to an entity in the scatterplot. However, will these observed orthogonal patterns remain, if the entities lie within a larger dataset? Will this depend upon whether they are localized to (in the neighbourhood of) the reference entity? To test this, entities with corresponding similarities need to be created and visualized with respect to a highlighted entity in a synthetic or real-world dataset. For this process we propose the term *feature fingerprinting* after Meuzelaar's operational fingerprinting (see page 6), to put the feature into a dataset under consideration. For instance, in the case of the webbed foot pattern, to see if the column represents means or scaling or neither. Thus we need to check whether the shape is preserved, which it might be concluded would not be due to dimension reduction approximation. Having some indication of the meaning of direction may then help us understand the cluster shapes. Another issue is to what extent this is dependent upon the metric used?

Describing the visual depictions in this test also showed the value of the feature admissibility approach, if combined with a descriptive comment about the pattern that resulted from the feature. So that one can say 'this visualization method *does* or *does not* reveal feature x' and, if it does, what pattern arises from the feature.

7.4.2 User Interface Generic Datasets Conclusions

The comparison of algorithms and the implementation of the generic dataset option in Space Explorer enabled the user to gain a concrete view of the difference in behaviour of several visualization algorithms. In particular, this led to observations about City and Euclidean: City is not an intuitive measure compared to Euclidean (consider the measure in 2D, it makes much more sense to go directly from one point to the other, rather than going along and down or up), but the visualization indicated that the results of City were similar to Euclidean, which is important where ease of use is concerned (City is simpler to compute and entities can be added without recomputing). Angular separation is also shown to be appropriate where absolute values are not relevant. These results can be inferred from the definitions, but they are made concrete by these visualizations. There is also the issue about the extent to which we consider here the visualization algorithm as a black box. In one sense it is desirable to consider it as a black box, since in use (by at least a proportion of users) it will

be.

7.4.3 Problem: Accuracy

As has been raised earlier, distance measures can usually only be satisfied approximately when finding a visual depiction and generally this applies to any process involving dimension reduction, since no layout in a lower dimensional space can satisfy completely the original data. This lack of accuracy in depiction raises two questions relating to the investigations of this chapter:

- Will the patterns be retained when placed in large datasets and in datasets of different types?
- How can we view the validity (and accuracy in terms of how the original distance measures are retained) when the eigenvalues do not tail off fast (as is common for many real-world datasets)?

One might think that accuracy is an important issue where dimension reduction is involved and that it needs at all stages to be illustrated, but dimension reduction methods, such as PCA, are also valid in that they show underlying structure, so that the low representation of data may not be so serious (as previously mentioned on page 94). Nevertheless, it would appear to be important to consider the option of showing accuracy when visualizing, so that the user is invited to consider (and, if appropriate, negate) its significance. This can be done by quoting eigenvalue contributions and average and variance of error in distances, but of interest is also to explore visual means that may give different information (such as highlighting entities that have errors in their distances above a certain threshold). There appears to be a significant lack of such considerations in the field of information visualization. Comments such as that data 'may not always fit into low-dimension spaces comfortably' (Chen 1999, p. 31) appear, but otherwise visualizations of high dimensional data, specifically after self-organizing map, correspondence analysis, minimum spanning tree, PCA, spring-embedding and so on, are routinely presented without any comment as to accuracy. It may be argued that they are providing a rough overview, so the accuracy does not matter, but these pictures may be also considered to mislead the viewer, in that they *appear* to be precise. Chapter 11 examines the question of accuracy in more detail.

7.4.4 Problem: Evaluation

Measuring whether an increase in comprehension has resulted is difficult. In our case we have either asked the participant of the feasibility test for their subjective assessment or examined the visual depictions to see whether the feature in the dataset is immediately apparent in the visual depiction. The latter approach means that a feature-admissible classification after Fisher (Section 7.2.3) can be followed. However, there is still the difficulty of generalizing; it can only be stated that the feature is shown for that particular dataset. Also, for another dataset, the feature may be obscured by

overplotting, as is the case for the 'scaling1' dataset (Table 7.4) displayed after PCA. In some visualization evaluations users are given entities to find in a particular representation (such as a directory containing particular information) which tests how easy a particular representation is to navigate and orientate oneself within, but such tests are not directly applicable to this situation.

7.4.5 Conclusion Summary

Overall, it can be said that some users increased their understanding (according to their subjective assessment) and that we were able to classify the visualization methods as revealing, or not, certain features in the datasets; with the proviso that we cannot guarantee the result for different datasets containing the features due to characteristics such as overplotting and the constraints of dimension reduction. This method may prove more useful for comparison of direct methods (colourmap, parallel coordinates etc.) not involving dimension reduction, since the representation is not so complex. The work should be continued to examine the behaviour of different visualization methods, different datasets and an expansion of the concept of feature-admissibility.

The question of type of dataset, i.e. what feature is represented, has been discussed in this chapter, but the precise values used to create the feature, the number of dimensions and so on, have not been the focus of the work. It is very difficult to propose the use of one set of values rather than another, so that the datasets used in this chapter are examples taken from many possibilities. In these examples data that was convenient for other reasons was chosen, and so they are not presented as definitive representatives.

There are two indications of usefulness in a wider context:

- For evaluation, towards the establishment of a set, or sets of datasets for comparison, evaluation and general appraisal of different visualization algorithms (which contributes to the establishment of benchmark datasets for visualization).

- For placing *feature fingerprints* in a complex dataset, to help make sense of the shape of clusters, and to orientate the user. A feature fingerprint results from the placement of a small localized group of entities, created with specific similarities to a particular entity, or group of entities, selected within a visual depiction, so that the fingerprint of the feature can be seen in the visual depiction.

It was decided to examine the latter of these, feature fingerprinting, in the use of user constructed data, which is described in the next chapter.

7.5 Summary

In choosing datasets to illustrate the behaviours of metrics and visualizations, referred to here as generic, many questions arise due, in the main, to the difficulty of quantifying features in the dataset and patterns in the graphic, as well as generalizing results from one dataset to many. However, features of interest can be suggested and ways of producing datasets to contain them, broadly grouped as those that are structureless and those containing various types of embedded structures.

The clustering and visualization literature was first examined for relevant work. The datasets in two books of datasets and a number of online datasets were examined, but no obvious candidates for generic dataset provision were found, other than examples of different numbers and orientations of clusters. The literature contains a description of a method for classifying clustering algorithms which proposes an admissibility procedure. This procedure is suitable for adapting to visualization methods. Many references refer to the difficulty of choosing methods for classification and clustering and the same can be said for visualization, nevertheless guidelines are desirable. A number of null models for absence of structure have been described in the literature. For pattern matrices there are two aspects: the 'shape' of the region of data and the distribution of entities within it. Work in information visualization has proposed embedded clusters within noise to test whether the embedded stimuli can be observed in visualizations. Statistical measures are useful to give information about the data. Further integration of statistical measures with information visualization is desirable (they are absent from many applications), but not the main focus here. Visual languages show potential for assisting the user in devising measures for the feature in question, where mathematical measures are not available, and increase the engagement of the user in the process.

The general categories identified for generic dataset provision are: null modes; clusters within noise; specific features (overplotting, outliers); inter-element features, such as scaling. This work is restricted to multivariate data tables. The data is considered to be *known* on 3 levels: by knowing the actual data values (implying small datasets); by knowing some statistical features (including large datasets); where tacit knowledge, knowledge that is not written down, is involved. This work focusses on small datasets. Some larger datasets are used in testing accuracy of different layouts in Section 11.5.2. Tacit knowledge is returned to in Chapter 10.

Two scenarios for the application of generic datasets are considered here: examining a single visualization method using several datasets; comparing different visualization methods using a single dataset. Correspondingly, two pieces of work are described: a feasibility test and the integration of generic datasets within the user interface.

The feasibility test involved 12 participants examining a series of visual representations (using a single visualization method) of example datasets, to see whether their understanding of a dimension reduction visual depiction (the calldata representation introduced in Chapter 1) was increased. Some

participants considered that the test increased their understanding, but they were not able to increase their explanation of the calldata pattern. Most of the participants wanted to be able to construct and manipulate data in the data table themselves, so that they could experiment further. The results from examining the set of visual depictions in this test suggested a means to assist in orientating the user in spaces where direction has no obvious meaning. This followed from the observed orthogonal or nearly orthogonal relationship between scaling and means (or phase shifting and means). From this I have proposed a new approach, *feature fingerprinting* where data representing a feature is added to a real-world dataset under consideration (this is examined in the next chapter). Feature fingerprinting will also be useful for illustrating the behaviour of visualization methods in general. The visual depictions in the test were examined for feature-admissibility (was the feature clearly shown in the depiction, or not?). Thus one could say whether the visualization method revealed the feature, and, if it did, what pattern resulted in the depiction.

A second examination of generic dataset provision aimed firstly, to illustrate how generic datasets could be integrated within a visualization application and, secondly, to use the datasets for assistance in choosing between metrics. Space Explorer was modified to contain spreadsheet and stacked bar chart provision, together with example generic datasets. This enables the user to choose a dataset, display the numerical values in the data and view a simple direct representation (in the stacked bar chart), then choose different dimension reduction methods to produce scatterplots in 2D and 3D. The interface was then used in the task of choosing metrics for an agent application measuring similarity of interests between agents. The features examined in the interface were: identical entities; outlier; lack of structure (randomness); scaling.

The creation and addition of a feature to an existing dataset, with values based upon some relationship to the values of a particular entity, feature fingerprinting, raises important questions concerning dimension reduction applications. Will the patterns be the same in different datasets? How can the validity be viewed if a high level of abstraction is involved?

The work described in this chapter has shown that the exploration of generic datasets increases comprehension and assists metric choice to a certain extent. A feature admissible classification has been outlined. Further development of the interface, investigation of datasets (many datasets containing many different features) and the feature-admissible procedure are recommended. In a wider context, the work has relevance to the establishment of benchmark datasets and for the development of a new technique, feature fingerprinting.

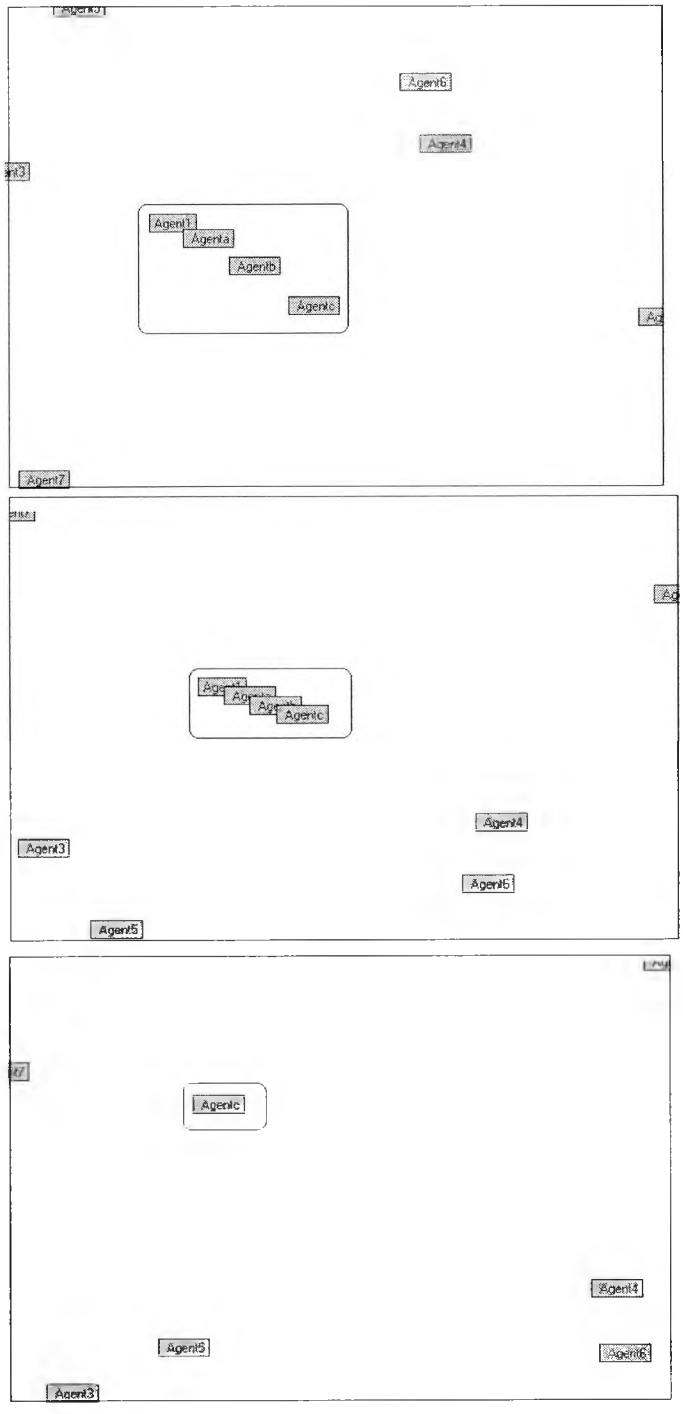


Figure 7.13: City (top), Euclidean (middle) and Angular Separation (bottom) measures followed by layout using Principal Coordinates Analysis. Agents a, b and c have scaled interest distribution of Agent1. In the angular separation, Agents a,b,c and 1 are located in the same position.

Chapter 8

User Construction of Data

8.1 Introduction

In the approach to signature exploration employing user construction of data, (Definition 12 on page 81), the user is given the ability to construct datasets in a variety of ways, all of which allow the user to *know*, in some sense, the data they then put into the visualization method. Two approaches for user construction are i) to provide a generic set of datasets for the user to start with, and ii) to provide tools for the user to create data from scratch. Both of these are desirable, since it is very difficult to agree upon a specific example of, for instance, 'outlier' or '3 clusters', nor to close the set of possible interest features. There is also some indication that users will benefit from the combination of these two approaches, as demonstrated by the feasibility test (see page 112) described in the previous chapter. So that generic datasets can be provided to give illustrative examples which the user can then explore by adding to and modifying the data. This is aside from the general question of the importance of engaging the user in an active, interactive sequence of events, for which interaction with the originating data (i.e. by modifying original data values, querying or adding landmark values) is beneficial.

In general, for construction of data, the user can create the data from scratch, transform an existing dataset that they are exploring, or start with one of the generic datasets provided. User-construction means:

- The creation of data showing features of interest to the user (*static* constructions).
- The creation of data as a result of simulation based upon sets of behaviours specified by the user (*simulated* constructions).

Static constructions are matrices specified by the user, which can then be visualized. They are static in the sense that they are an instance of creation by the user, as opposed to simulated construc-

tions, which are the result of data produced by a simulation of entity behaviours as described in Section 6.3.2.

This chapter describes a number of ways that can be used for the static construction of data:[1]

- direct entry or change within a data table
- interest feature specification
- generating data via a visual representation
- using synthetic data generators

These methods are illustrated in implementations of

- multiple windows linked to the data table by brushing (direct entry of data in a data table; generating data via a visual representation)
- synthetic agent profile data creation based on the user's sense of similarity (interest feature specification)
- dataset generation based upon Gaussian 'clouds' for examination of accuracy of dimension reduction algorithms

The last implementation is not described here, but is used in the following papers,

- "Approximate Profile Utilization for Finding Like Minds and Personalization in Socio-Cognitive Grids" (Noy and Schroeder 2003)
- "Advancing Profile Use in Agent Societies" (Noy and Schroeder 2004)

In these papers data is generated in order to test the accuracy of different layouts, including a new layout method which performs iteratively and can be used by software agents to measure their similarity to one another, without needing to reveal the details of their original profile or to use a trusted third party.

8.2 Methods

8.2.1 Direct Entry

For direct entry or change of data, the user enters data directly into a spreadsheet application and has the facility to link the data to chosen visual representations. The user enters data manually into a table so that they know the actual data values. They may start with a dataset and modify the

[1] This is not an exhaustive list, for instance, it excludes creation of data by specifying functions, such as, for entity y's time series data, $y = f(t)$.

values (including the creation of new entries) or begin with a completely blank spreadsheet. The spreadsheet application is directly linked to any visual displays chosen by the user, so that the data can be viewed as it is entered or changed, as illustrated in the description of a direct entry interface below (Section 8.3).

As an aside, it is noted that providing the user with the facility for viewing the actual data values is, in itself, important for user comprehension and also for user confidence, apart from its role here in applying signature exploration. Statisticians, in particular, express a desire to view the actual data, but there are also groups of users who are comfortable with the data in its raw form and much less sure of the value of any visualization except that of the most traditional kind. Thus starting with their *known view* of the data, they can move to new views of the data. This *known view of the data* is the data table, or a visual representation that that the user is very familiar with. Now *known* relates to the visualization method, rather than the data (though the table representation can be considered to be a *view* in this context). Thus, understanding of new (to the user) visualization methods is increased by comparing an unfamiliar output with other outputs with which the user is more familiar.

There are three aspects of visualization system design that this discussion suggests:

- include spreadsheet application for presentation and modification of the data
- include linking between data in the spreadsheet and any visualization windows
- include *familiar* representations of the data

8.2.2 Interest Feature Specification

Users have features that are of interest to them concerning their domain and type of data. User construction of data provides the means for the user to create datasets containing such features of interest - this process is described here as interest feature specification. In specifying such interest features, the first task is to identify the features of interest, then to consider how to measure the degree to which an object, or group of objects, demonstrate the feature. For instance, in the case of time series data, the user may be interested in the absolute average value overall, or in the existence of a phase-shifted pattern between objects. In order to explore the different representations of such elements, data can be produced which reflects the whole range of possibilities, i.e. showing varying amounts of the feature. A visualization method can then be used to see how well clustered the similar objects are and how well shown the feature of interest is. This approach helps the user to identify and quantify their interest features.

Interest feature specification is an extension of the direct entry described above, in that the user identifies feature(s) of interest to them and then creates data covering a range of levels of this feature. In clustering applications users are interested in how similar two objects are. Thus data can be created

showing different amounts of similarity to a reference object based upon the user's ad hoc creation of similarity measures, capturing their sense of what makes two objects similar. A useful application for this is in the case of choosing a similarity metric and this is illustrated below in the example to find suitable metrics for agent profile data (Section 8.4).

8.2.3 Generating New Data Via the Visual Representation

Creating data by altering or generating a visual representation is also a way of exploring the visual representation itself, but is not valid where mathematical abstractions of the data have occurred, such as those involving dimension reduction. Such transformations are not one-to-one functions, but many-to-one, so that a new point in the visualization represents any one of a number of data table entries. Consider, however, direct plot methods such as: bar chart, line plot, parallel plot, colourmap etc. In each of these the user could change the position or colour (as appropriate) of an element and the data value in the data table could be changed automatically. In this way the user can directly, without ambiguity, find the answer to the question: 'If the visual display looked like this, what would the data look like?'. This facility means that users can view the changes in individual objects' entry rows in the data table as they alter the visual representation, by clicking and dragging points and lines with the mouse. In this way data is created by interaction with the visual representation. *Sketching* could also be used in this way to generate data. In this way, for instance, for time series data the *shape* across time can be scaled, frequency- or phase-shifted, to create new sets of object values (as in the last example of the feasibility test Figure 7.8 on page 103). An implementation of the visual generation of data using a bar chart is described in Section 8.3.

8.2.4 Synthetic Data Generators

Although the previous three methods generate synthetic data, they do not do so by using statistical models and they focus, instead, upon individual values. Though knowledge of individual values leads to a certain clarity, especially in relation to small datasets, (remember Tufte's 'Graphics reveal data' and Anscombe's quartet of datasets with the same summary statistics referred to on page 96), it is difficult to generalize about abstractions upon this basis. Such abstractions can be highly sensitive to the precise circumstances within which a data feature arises. For instance, if a specific pattern is obtained with a particular small set of data, there is no guarantee that such a pattern will appear when that set of data occurs within a larger one. However, it can be hypothesized that the relative scale of the pattern to the rest of the dataset is a determining factor, i.e. when the pattern represents only a local 'disturbance' it will not be much altered. This hypothesis is tested in Chapter 9 in the context of adding a small constructed set of data to an existing dataset.

The discussion between large, statistically specified synthetic datasets and small, manually con-

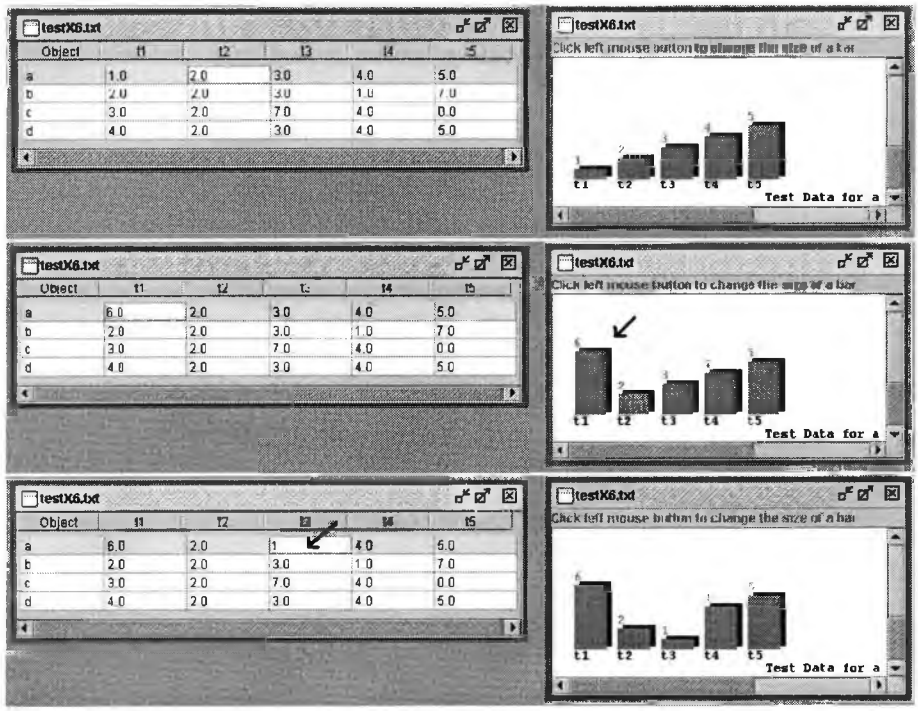


Figure 8.1: A sequence of 3 pairs of spreadsheet and bar chart windows showing linkage: top row of spreadsheet selected (top); mouse click in bar chart changes height of first bar and change is reflected in data table (middle); table value alteration is reflected in bar chart (bottom).

structed ones, has been raised in the previous chapter (Section 7.3). The addition of synthetic data generation complements the provision for small scale data construction and allows the user to experiment with cluster generation of different types and to address issues of scale. Some combination might also prove useful where a small user-constructed dataset is embedded in a larger statistically specified one. Thus expanding the idea of *noise* with *embedded stimuli* (see Section 7.2.5).

8.3 Illustration of Direct Entry Interface

This section illustrates linking the data table by brushing to other visual representations, thus generating data via the visual representation. Figure 8.1 shows two linked windows for spreadsheet and bar chart presentations of the data. These windows are linked so that when the user clicks in the bar chart they can change the height of the bar and the data table entry in the spreadsheet is correspondingly changed and vice versa. The user can thus enter or change each row as desired and view the resultant visualizations of the whole dataset. This implementation illustrates construction of data by direct entry and via a visual representation.

	A	B	C	D	E
agent1	9	3	4	6	5
agent2	1	10	10	7	2
agent3	4	1	6	8	0
agent4	2	7	4	8	0
agent5	3	6	4	7	1

Table 8.1: Data showing interest level in 5 subjects for 5 agents.

8.4 Application to Agent Profiles

This section illustrates interest feature specification to examine the behaviour of different metrics for an application involving agent profiles. This material is covered in the following papers: Noy and Schroeder (2002b), Noy and Schroeder (2002a) and Noy (Noy).

The agent profiles here can be of a task or of a user and 'agent' can be either a software or human entity. Consider comparison of agents based upon the similarity of their interests. An example dataset, concerning five agents is shown in table 8.1. These agents have five possible interest areas $(A, \ldots, E)$ and interest level values in the range 0 to 10.

In order to examine the way different metrics behave in relation to this data, and to enable the user to explore their sense of similarity and the type of features they are interested in, three steps are suggested: some features of interest are decided; a measure is created with which to generate test datasets; visual and/or numerical evaluation is performed. This three-step process is repeated iteratively as appropriate. For the agent data the process is illustrated as follows:

Step 1 - decide features of interest Considering agents with different interest levels in a set of subject areas, possible features of interest are: overlap of interest; intensity of interest; joint disinterest; similar pattern of interest (irrespective of subject). One or more of these elements can form the basis of a classification system which gives numerical values to differences between a pair of agents' interests or a binary variable, providing a similar/not similar partition. This ad hoc classification can be compared with those provided by other metrics.

In a search for a metric, the user's measure may be found suitable to use instead of the metrics under consideration. Examples can be found of the use of simple metrics in multi-agent systems (see e.g. Faratin et al. (2000); Foner (1995), which respectively use: a measure which examines all differences and uses the largest one; a similarity classification based upon the existence of a single joint interest). However, here the purpose may be to choose a metric, but it may also be to examine the behaviour of a particular metric, in which case a comparison to the user-constructed measure is valuable. Another aspect to this situation is the existence of large quantities of complex, multivariate profiling data and the desire to make use of it in a more sophisticated manner. Thus the search is for

more subtle measures, something that reflects the multidimensional nature of the available data. This corresponds to the scope that lies between the two questions: 'Are you interested in sport?' and 'Are you like me?'

Also, if you are interested in sport, it may be valuable to know if you are a specialist or a generalist and in general terms what level your interest is on. Thus other similarity measures act as discriminators in this situation, in which case, final choice of overall similarity measure consists of additions of different similarity metrics (including any results of specific queries).

The use of visualizations of data for pairs of agents can assist in the specification of features of interest.

Step 2 - create a measure for the features to generate test data sets Suppose that the user chooses to examine agent similarity based upon overlaps of three or more interests of high intensity. To illustrate, a data set was created to produce examples covering the range of possibilities of overlap extent and intensity with respect to a reference agent's interests. The reference agent was randomly assigned levels of interest in six subjects and data for a number of other agents created that covered a range of possibilities of overlap intensity (one third, two thirds and the same as, the reference agent's level of interest in that subject) and number of joint interests (1 to 6 with the reference agent). This allows the metrics to be examined to see how they cluster the group of similarities with number of overlap subjects $>= 3$ and intensity of overlap $>= 2/3$.

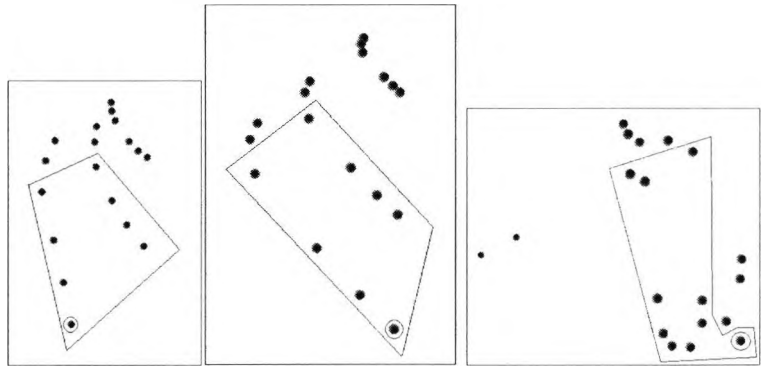


Figure 8.2: A user-specified constructed dataset of agent profiles, displayed with three different distance metrics - (left to right) City, Euclidean and Angular Separation - followed by PCoA layout. The areas indicated contain agents with 3 or more joint interests with intensity of overlap greater than 66% with respect to the circled, reference agent. The linear groupings reflect the way that the data were constructed (see text).

Step 3 - evaluate visually and numerically The visual evaluation consists of visualizing the constructed data set and observing how well clustered the group of interest is. However, since the layout of such visualizations is an approximation (in order to satisfy the distances), and the observations

not themselves measurements, evaluation by visualization is inexact. On the other hand, numerical evaluation, based on measuring differences between the estimated differences and the differences arrived at by the metric under consideration, is precise, but relies on the ability of the user to define or estimate similarities between the data entities. Since the user is using rough ideas to get a feel for the behaviour of the metrics, it is appropriate to look at the results visually as part of an iterative process. For this example the numerical calculation can be done by using an intuitive points system according to number and intensity of topics of joint interest between pairs of agents. By awarding points for the similarity of each pair of agents, a proximity matrix can be derived which can then be compared numerically to those obtained by using the different metrics.

Figure 8.2 shows PCoA layout with City, Euclidean and Angular Separation differences, the reference agent is circled and the agents that are in the user's group of interest (according to their criteria in step 2 of greater than 66% joint interest in 3 or more subjects) are indicated. The linear groupings shown in some of the screenshots of Figure 8.2 correspond to the three levels of interest overlap and the number of joint interests. These linear groupings thus reflect the method of creation of the dataset. The point of interest to the user is where the entities, which they classify as similar, lie in the different displays. These entities are outlined in Figure 8.2. How well are they clustered by the different methods? The outlines traced by the points in the City and Euclidean plots correspond closely to the classification system and the group of interest is well clustered in that all the agents shown in the proximity of the reference agent are considered by the user's classification to be similar to the reference agent. The Angular Separation plot does not cluster so well, placing three agents near to the reference agent that are not considered by the user to be similar to the reference agent. The layout of the angular separation distances is actually a screenshot of a 3D representation as the layout was particularly inaccurate and needed the extra dimension to improve it (the first two eigenvalues accounted for only 38% of the variance in the data and the first three for only 48%). The inaccuracy of this layout highlights the difficulty of using visualization to assess similarity.

8.5 Summary and Conclusions

A number of ways for the user to construct data have been outlined in this chapter: direct entry or change within a data table; interest feature specification; generating data via a visual representation; using synthetic data generators. Three implementations illustrate these methods (excluding the synthetic data generator, though an example is used in several associated papers). An interface showing direct entry and linkage to visualization via brushing has been demonstrated, which also illustrates the generation of data via a visual representation. An example of data construction based upon the user's features of interest and concepts of similarity has been examined and shown to assist with

metric choice in an application involving agent profiles. [2]

Bi-directional brushing of data between the data table and visual representation, (i.e. *forwards*, from the data table to the visual representation, and *backwards*, from the visual representation to the data table) that also allows the values in the data or display (colours, lines, points) to be changed, introduces a new level of interaction to visual applications. This can be thought of as *visual data tracking*, an extension to brushing that allows changing as well as highlighting of data and specifically includes a data table view. This facility will encourage interaction and hypothesis formation as well as giving concrete, dynamic illustrations of the behaviour of the visualization method. Existing applications go some way to providing this functionality, however, they do not change parameter values by interacting directly with the graphic. For instance 'DataDesk' (Velleman 1992), has graphical sliders which control and automatically update displays so that the user can observe their analyses through a continuous range of parameter values. As the slider is moved through new values, all plots and tables connected to the slider update instantly, displaying real-time animation.

In assisting metric choice, value is demonstrated of the specification of a dataset containing entities which demonstrate a feature over the range of possibilities. The data creation in this example was done manually after determining the features of interest. Making the data creation possible within the interface is desirable, as is the inclusion of data generation facilities in general. Since the visual depiction of the result of applying metrics is necessarily an approximation, the conclusions drawn need to be verified with numerical calculations before deciding in favour of a particular metric. The impact of this inaccuracy of layout needs further investigation in relation to conclusions from this type of experiment.

The issue of size of dataset is important, as in the previous chapter on generic dataset provision. The question of validity of layout and whether the result can be generalized also apply. Again, the work suggests that providing the ability to construct data *within* a real-world dataset would be illuminating; that is, to construct data for synthetic objects and see where they appear in the visual depiction.

[2]The elicitation of feedback data examined in Chapter 10 is another method of utilizing the user's estimation of similarity between entities, though based upon a direct estimation of similarity between entities with existing data, rather than upon the user constructing synthetic data.

Chapter 9

Querying and the Insertion of Landmarks

9.1 Introduction

This chapter examines the different means available for *querying* (Definition 13 on page 82) in the visualization process as part of signature exploration. An illustration of visual querying of the calldata is given. The closely related concept of *insertion of landmarks* (Definition 14 on page 83) is also described and illustrated.

The last two chapters have indicated the desirability of placing a group of constructed data entities within an existing dataset, to assist orientation within the visual representation. This may be orientation in the sense of direction, as is the case when dimension reduction is involved, or in the more general sense of understanding the meaning of other features, as well as direction, such as colour and shape. The set of added data may be a generic set, illustrating a particular general feature, or a user-constructed set, illustrating a feature of interest to the user. This approach, for which I propose the name, *feature fingerprinting*, is a form of landmark insertion and is also illustrated in this chapter.

9.2 Querying

In querying, a cluster in a visual representation of a dataset under consideration can be highlighted, or an outlier, or the extremities of a pattern. This is visual querying. Alternatively the dataset can be queried using a conventional query language. A subset can also be obtained and visualized by the use of dynamic querying, where selections are made using sliders. These subsets of data, obtained visually or by directly querying the dataset, can be considered to be the constructed data for signature

exploration. Some of these techniques are well known[1], though not necessarily widely used. There is also a difference of focus here, since the aim is to reveal the behaviour of the visual representation. The next three sections discuss these three approaches:

- Use of conventional query language.
- Dynamic querying.
- Visual querying.

9.2.1 Use of Conventional Query Language

With a conventional query language, one can formulate queries at the command line to directly query the data and obtain subsets satisfying the criteria of the SELECT statement. This treats the data table as a database. There are many disadvantages in using this approach. Spence (2001, p. 71) lists seven, though he is considering the use of command line query *instead of*, rather than *as well as* visualization. He includes: having to learn the language; errors not tolerated; too few or two many hits; no indication of how the query may be reformulated to make it more successful; significant time delay between formulation and result; useful contextual data hidden; difficulty of user building mental model. The impact of these disadvantages is lessened if command line querying is used in conjunction with visualization, but the user still needs to be familiar with the query language etc.

9.2.2 Dynamic Queries

The general problems associated with command line querying are mostly solved by the use of dynamic querying (Williamson and Shneiderman 1992), where the selection of ranges within variables is made using sliders. Used alone, this technique loses context, since it returns the requested subset, which is then viewed separately. Nevertheless, it is proving popular, see, for example the Spotfire display (Ahlberg and Shneiderman 1994). The use of this technique within the overall dataset, is demonstrated in Attribute Explorer (Spence and Tweedie 1998) which thus provides contextual information. This contextual information shows not only the whole of the data as well as the subset returned by the query expressed by the slider positions, but also indicates the objects that failed the selection by only 1,2 or 3 criteria. This can be described as *sensitivity information*, i.e. how sensitive the visual depiction is to changes in the slider. *Insensitivity* can be indicated in the sliders to show that moving the slider in a portion of its range will have no effect, if that condition has been produced by the settings of the other sliders.

[1]The conclusion one comes to is that applications are needed that incorporate *all* the functionality that researchers are developing, though this makes applications more and more complex to build. However, as has been noted in Chapter 5, whilst some of this functionality makes applications easier to use, especially in hypothesis exploration, some elements produce complications of their own, for instance those using new dimension reduction algorithms or involving multiple windows.

9.2.3 Visual Queries

Selecting areas of the visual display, or sets of individual objects within it, by moving the mouse over or clicking an object of interest, to view more information about the objects, is an example of visually querying a graphic. The focus+context techniques, and those described in Section 11.3 on visual depictions of accuracy, also provide more information about a selected group. Brushing can be seen as a form of querying (the *answer* is in the form of another visual depiction), as can, in general terms, all interaction with the visualization.

9.2.4 Issues

Within signature exploration, the use of querying is to examine the behaviour of the visualization method, rather than investigating a dataset. However, no evidence suggests that this makes a difference to the techniques required, apart from indicating the desirability of expanding a subset of techniques to query the behaviour of the visualization method itself, such as in showing accuracy. As introduced in Section 6.3.3, which defined querying for signature exploration, the aim here is to answer the question 'Has the visualization method placed these objects as I expected it to?' or 'On what basis is the visualization method placing these objects together?' In general, querying for signature exploration returns a new visualization, or gives information about a subset, or highlights the subset within the visualization as usual.

How to measure the success of querying to increase comprehension of the visualization method or the dataset? An initial reaction is that the use of querying of the dataset will *obviously* aid comprehension of both a dataset under study and the visualization method itself, since the user obtains more information about one or other of them, or both. However, how can the impact of querying be measured? One approach suggested is to examine the *interaction response time* (Spence 2001). The interaction response time can be thought of as the delay between formulation of a new query (or hypothesis) and delivery of its result. An important aim of developments in the field of information visualization is to make querying and hypothesis formulation easier, and one expression is to reduce the interaction response time. *Responsive interaction* has been expressed as when 'an effect occurs within less than about 0.1s of its cause' (Spence (2001, p. 71)) and this can be taken as a base measurement for more complex 'causes', such as in hypothesis formulation of the kind involved in user-construction of data, where a new dataset is created to see how the system behaves. A general expression of the techniques needed in this area is that tools are required for *interrogation* of the visualization *and* the data, and to make this interrogation easier. However, there still remains the question of whether users will *like* these tools and *want* to use them, aspects that are much more difficult to evaluate.

9.3 Illustration of Query

An illustration of directly querying the graphic is shown in Figure 9.1. The calldata dataset is visualized with Euclidean distance followed by PCoA for 3 dimensions. The extremities are highlighted and the data for these individual customers viewed as bar charts (impulse charts in gnuplot[2]). A chart showing all customers' bar charts superimposed is used for comparison. It is difficult to view the bar charts because of the number of destinations, but is just possible in this case, because of the sparsity of the dataset. These visualizations give insight into the behaviour of the dimension reduction algorithm: the outliers are customers with the high numbers of calls to a particular destination (the screenshot in Figure 9.1 does not show this well). Rotating the 3D representation also suggests that these main destinations form axes around which the other customers are clustered.

However, are the single destination peaks the highest overall? A different kind of query is required now, for example, to select the customers making more than 50 calls to a single destination. Are there customers with more than one such destination? Are the selected extremity customers in the 3D scatterplot the ones with the largest peaks? The results of this query are shown in Figure 9.2, which colours the 28 customers in this category, using different colours according to the destination of their calls. This figure shows that the selected extremity customers all fall into the category of making above 50 calls to a single destination and confirms that these main destinations tend to form axes. However, the question of whether the selected extremity customers are the ones with the largest peaks (and no others) is only partially answered since the exact values of the peaks are not shown. The peak heights are given in Table 9.1 showing that all the destination 'E' peaks are contained in the column of the scatterplot, so that the row2 customer, which is an extremity, actually has a low peak (55 calls to destination 'G'). The table also shows that only one customer has two destinations to which it makes more than 50 calls. In this case it is clear that neighbours to the extremities have a large number of calls to the same destination. In other situations a query to examine a certain number of neighbours, or area around an entity, is needed - to see whether they share a peak or why else it is that they are close to each other.

This example illustrates that a variety of query types are needed and that these types need to be integrated within the application and with one another. It will be useful to further develop query constructs to support hypothesis testing. Two types of construct are required - one for the display and one for the source of the data. These correspond to the use of conventional query language and visual queries of the previous sections. (Dynamic queries being a form of querying the data source and visualizing the result.) For example, where the 3D calldata dimension reduction display is concerned, one needs to ask 'the closest point to', 'the largest x-value', 'the densest region', etc.

[2]Gnuplot (correctly spelled 'gnuplot' - i.e. with a lower case 'g') is a command-driven function plotting program which can also plot data.

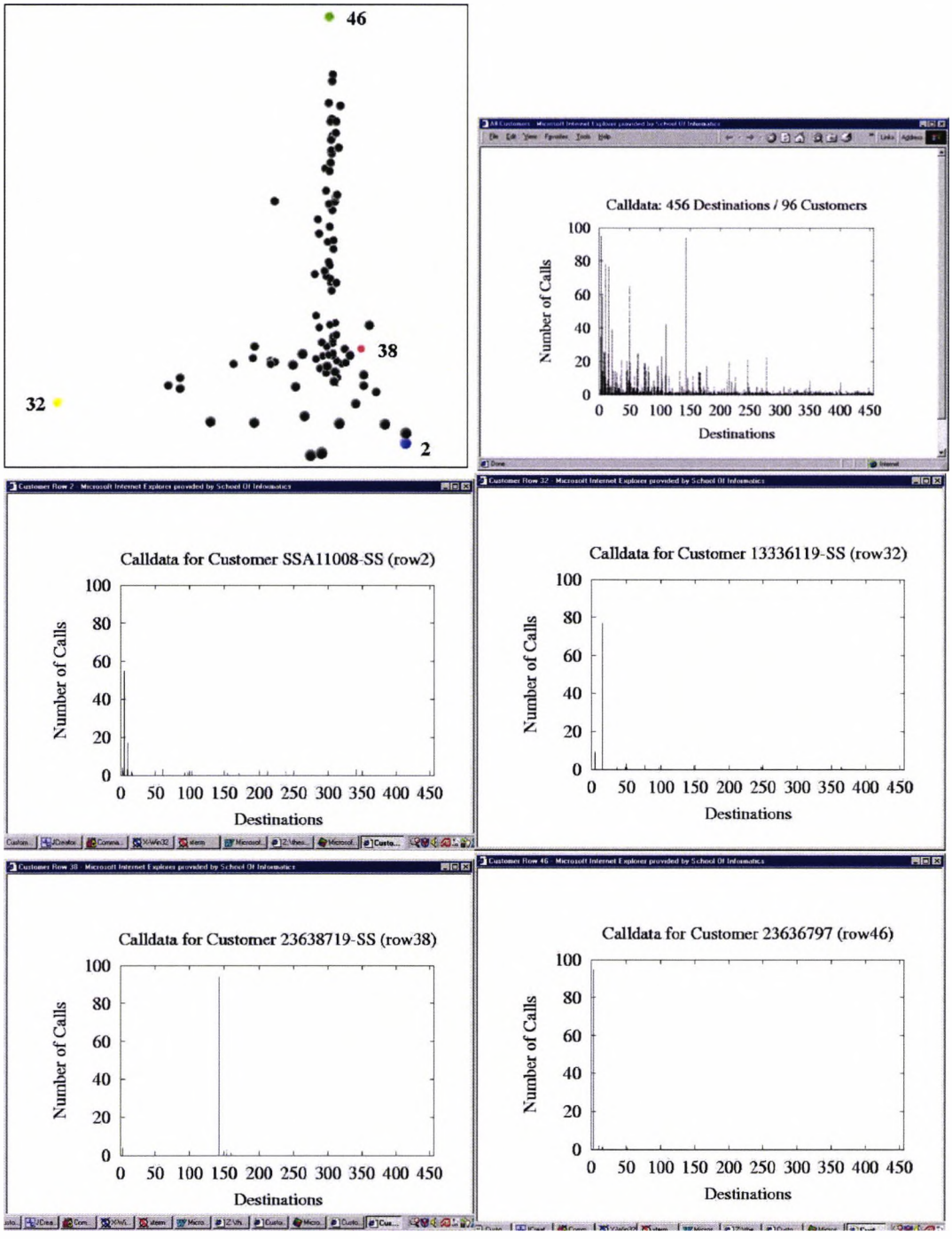


Figure 9.1: Example of direct visual querying of the calldata set - selection of the extremities. First the bar chart showing all customer calls superimposed (top right) is examined. It shows that there are a few peaks of high numbers of calls to a destination, but that the majority of values are low. The scatterplot produced using the Euclidean distance measure followed by PCoA for 3 dimensions (top left) shows the duck's leg and webbed foot shape. The numbers 2, 32, 38 and 46 correspond to the row in the data table in which the customer appears (row 38 customer is the furthest sphere to the back of the screenshot). Charts of these selected extremities (middle and bottom) show that the highlighted customers correspond to peaks for single destinations.

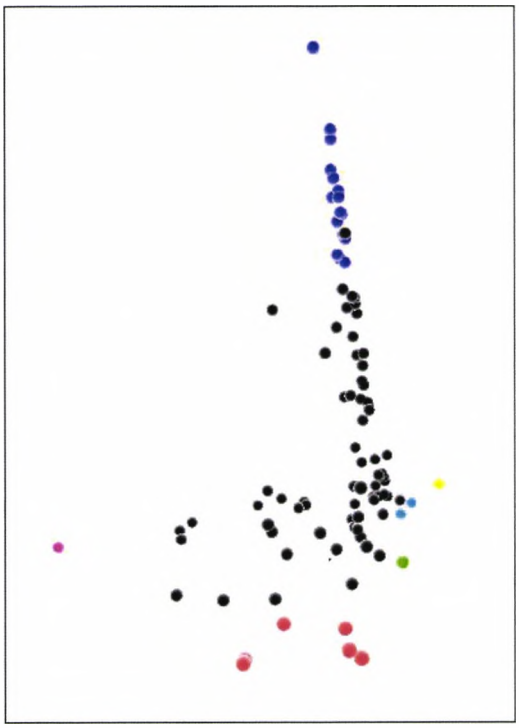


Figure 9.2: Example of visualizing the result of a query to obtain more information about the extremities. The data is queried to obtain the customers that make more than 50 calls to a single destination. The result returns 28 customers involving 6 destinations, only one customer makes more than 50 calls to more than one destination (see Table 9.1). These customers are highlighted in the 3D scatterplot, with different colours for the different destinations (key is in Table 9.1).

Number of Calls Made	Customer Row Number	Destination	Highlighted Extremity?
52	33	G	
52	39	E	
52	89	G	
53	50	E	
55	2	G	yes
55	3	G	
56	74	E	
58	28	G	
58	31	E	
58	62	AZ	
58	83	E	
59	13	G	
59	69	E	
61	71	AZ	
62	49	E	
62	53	E	
63	25	E	
67	16	E	
67	57	E	
67	90	E	
71	75	E	
72	15	E	
77	32	R	yes
78	14	L (also 60 calls to E)	
78	85	E	
79	4	E	
94	38	EO	yes
95	46	E	yes

Table 9.1: Customers making more than 50 calls to a single destination: details of call numbers, destinations and whether the customer was highlighted in the scatterplot of Figure 9.1. There are 28 customers, 6 destinations. 17 customers make calls to E, 6 to G, 2 to AZ, 1 each to L, R and EO. Colours in Figure 9.2 are E: red; G: blue; L: green; R: pink; EO: turquoise; AZ: yellow.

The source in this case is a time series, relevant questions are the average number of calls per destination, the largest number of calls per destination etc. Thus the *source* or *target* data can be identified: also *aggregates*, being those involving the whole of the source or target data (e.g. average, largest), and those involving a *single point* (e.g. closest to a point). Typically one would expect to start with formulating aggregate queries on the source data and then single queries on the target.

9.4 Insertion of Landmarks

Landmark and query overlap as concepts, since an entity or group of entities in a visual depiction can be highlighted and their data examined (visual querying), then left highlighted in the graphic to provide orientation (providing a landmark). Whilst query relates to an action for which there is

an answer, the insertion of a landmark adds a point or group for the purpose of orientation. Thus, highlighted entities resulting from a query can be left as landmarks in the display, but new entities can also be invented. For example highlighting the most expensive house (query result) or including one's ideal house (synthetic addition) in a dataset of houses for sale.

The incorporation of synthetic data into a visual depiction of data is suggested here as a new general element for visualization applications (having noted its desirability in the previous two chapters). This mixing of real-world and synthetic data in visualizations is similar to *operational fingerprinting* described in the introduction (page 6), where a known or *standard* substance is included in the dataset, so that its pattern will help guide the user in interpreting the results, specifically in relation to establishing the closeness of any unknown isolate to a known organism. The addition of data embodying a specific feature, within the data for visualization is here described as *feature fingerprinting*.

9.5 Illustration of Landmark: Feature Fingerprinting

The examples in this section illustrate the augmentation of a real-world dataset by the addition of synthetic data to orientate the user in the visual representation. Figure 9.3 shows an example of the addition of customers with data similar to a reference entity, in this case the row 46 customer of Figure 9.1. Euclidean distance and PCoA for 3D are used as before. The data is formed by adding or subtracting one unit from the selected reference entity's values. This is an example of feature fingerprinting with a 'scaling' feature with respect to a chosen entity. An example test dataset is shown in Table 9.2 and the new file containing 6 extra entities is shown in Table 9.3. Another set of entities is formed by shifting the reference entity's values to the left or right; this is illustrated with a similar test dataset in Table 9.4. The data additions for the calldata dataset were produced in a similar manner, but with slight variation: only positive values were allowed; zeros (in the reference entity's values) were treated in two ways a) by allowing the additions (Figure 9.3 top right) and b) by leaving the zeros unchanged to preserve the characteristic sparsity of the data (Figure 9.3 middle left). This 'shifting' of values provides an important feature in a dataset of this type - one where the overall behaviour of the entity across the variables is similar: in this case the calling behaviour is the same, but to different destinations. The shifting mechanism here is a quick way of producing such similar patterns. In time series data this corresponds to amplitude modulation. The visual representation after adding shifted entities to the calldata dataset is shown in Figure 9.4. This provides an example of feature fingerprinting with a 'shifting' feature with respect to a reference entity.

The results suggest that the reference entity, R, in these examples, contains the largest value or values of the column in the 3D representation. As the values of R are increased to create additional entities, the column of the representation is extended, though at a slight angle to that of the column. When the values of R are reduced to create additions, the new entities are placed towards the rest of

a	1	2	3
b	4	5	6
c	7	8	9

Table 9.2: Test dataset.

a	1	2	3
b	4	5	6
b+1	5	6	7
b+2	6	7	8
b+3	7	8	9
b-1	3	4	5
b-2	2	3	4
b-3	1	2	3
c	7	8	9

Table 9.3: Test dataset showing six additional entries using **b** as the reference entity.

a	1	2	3	4	5	6
b	7	8	9	10	11	12
b+1	8	9	10	11	12	13
b+2	9	10	11	12	13	14
b+3	10	11	12	13	14	15
b-1	6	7	8	9	10	11
b-2	5	6	7	8	9	10
b-3	4	5	6	7	8	9
b:shift1left	8	9	10	11	12	7
b:shift1right	12	7	8	9	10	11
b:shift2left	9	10	11	12	7	8
b:shift2right	11	12	7	8	9	10
b:shift3left	10	11	12	7	8	9
b:shift3right	10	11	12	7	8	9
c	13	14	15	16	17	18

Table 9.4: Test dataset showing twelve additional entries using **b** as the reference entity. Addition, subtraction and shifting of the reference entities values are included. The entity name indicates how the data has been constructed, e.g. *b:shift1right* means that the new entity has the values of **b** shifted one destination to the right.

the column and in line with it. This change in direction of the line is because negative numbers are disallowed. The entities with shifted values have positions far away from R. This shows that similarity in behaviour of calling, irrespective of destination, is not reflected in this particular similarity measure. The idea that extremities of the 3D pattern represent large values for particular destinations (from the visual querying example in Figure 9.1) is now qualified, since some of the added entries (SR1, SR3 and SL2) are very close together when they were expected to become new extremities. This suggests that another factor in the situation is the number of members of the dataset that share all or any of the reference entity's destinations. The desirability of the ability to answer with ease such questions as 'Which caller makes calls to the same destinations as the reference caller?' is again underlined, i.e. the ability to use database query language, or an interface providing the equivalent, within the application.

9.6 Conclusion

The illustrations of query and insertion of landmarks show that the two concepts overlap closely. Though landmark includes the addition of synthetic data, it is otherwise the highlighting of query hits in the visualization.

The *mixing* of real-world and synthetic data has been demonstrated to help orientate the user in a dimension reduction visual depiction. This work indicates the potential for a general visual fingerprinting, *feature fingerprinting* technique, which provides the interaction mechanisms and interface required for the user to place selected data of known qualities or similarity to a reference entity. Further work with a variety of datasets and visualization methods, as well as the development of suitable interfaces, is required to examine the value of this approach.

The previous two chapters have raised questions about the placement of new data within existing datasets for orientation. Section 7.4.1 asks: Will the observed orthogonal patterns remain, if the entities lie within a larger dataset? Will this depend upon whether they are localized to (in the neighbourhood of) the reference entity? To what extent is this dependent upon the metric used? Section 8.2.4 hypothesizes again, that the relative scale of the pattern to the rest of the dataset is a determining factor. In relation to these questions, this illustration, for the visualization method used, supports the validity of the pattern being repeated when introduced into a small area around the reference entity. More work needs to be done to examine under what conditions this will hold for other methods, datasets and sizes and types of introduced pattern.

This work has underlined the importance of providing a range of hypothesis support tools within the interface. The interaction response time can be used, as a measure of the time it takes to test a hypothesis. Comprehensive command line, dynamic query and visual query facilities are necessary, and require further development to facilitate hypothesis generation. For instance, to test the hypoth-

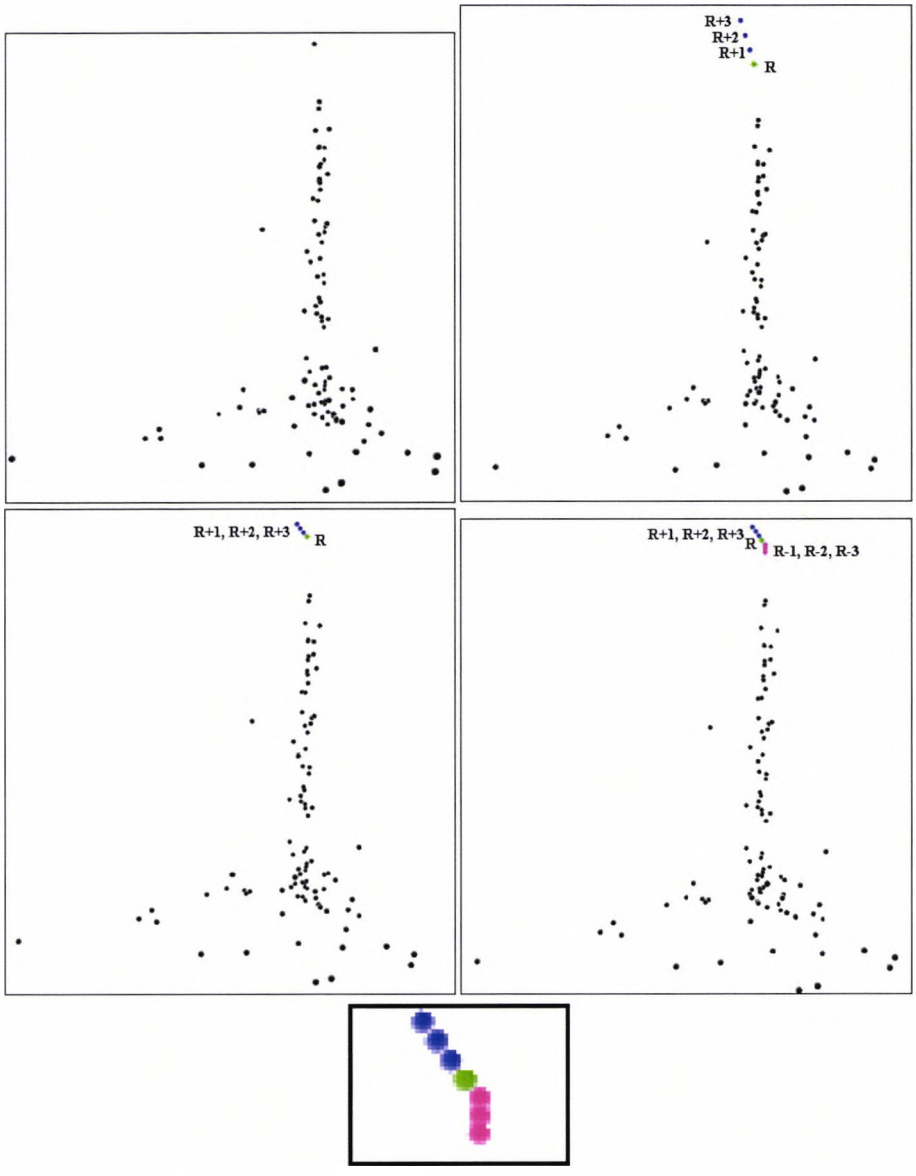


Figure 9.3: Feature fingerprinting with scaling feature - addition of constructed data scaled with respect to a reference entity. Top left: calldata dataset with Euclidean distance followed by PCoA for 3 dimensions. Top right: calldata dataset with a customer selected as a reference entity (**R**) and 3 new customers added to the dataset by increasing R's values by 1 (**R+1**), 2 (**R+2**) and 3 (**R+3**) respectively. Middle left: as top right, but R's zero values are not incremented, to maintain the characteristic sparsity of the data. Middle right: as middle left with 3 extra customers whose data is obtained by decreasing R's values by 1 (**R-1**), 2 (**R-2**) and 3 (**R-3**) respectively. Bottom: enlargement of middle right additions.

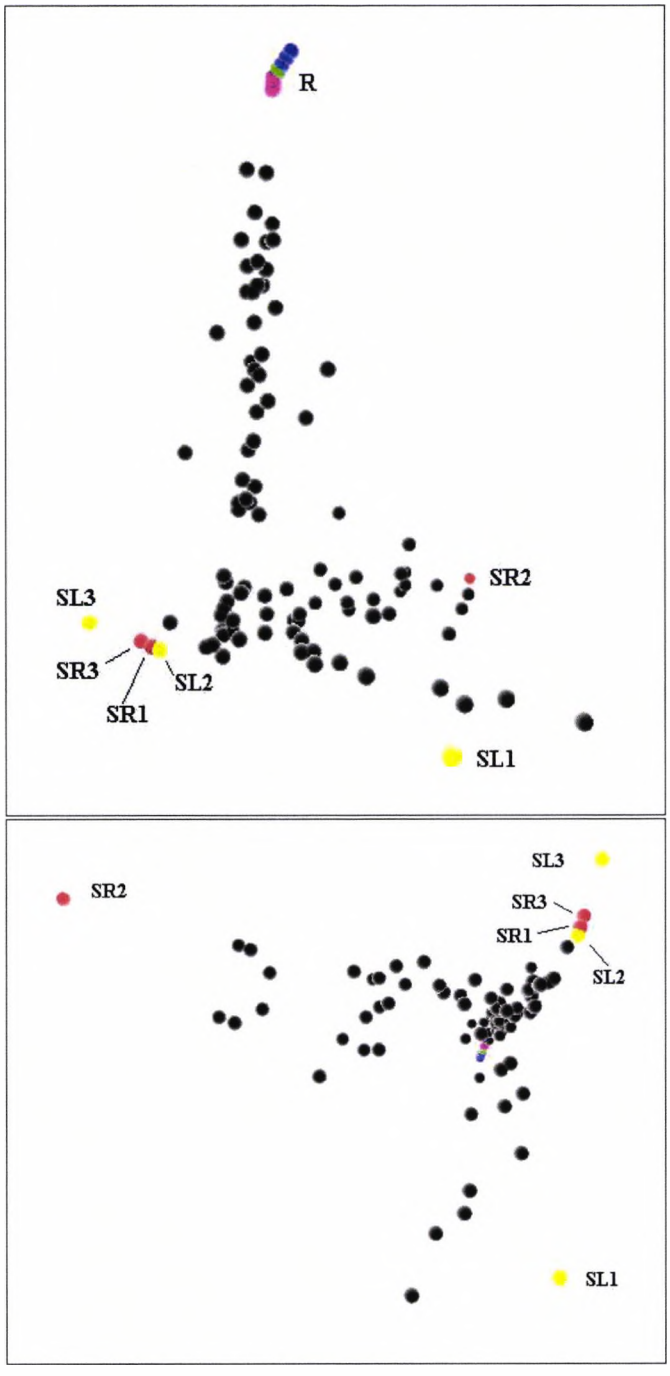


Figure 9.4: Feature fingerprinting with 'shifting' feature - addition of constructed data shifted with respect to a reference entity. Calldata dataset with Euclidean distance followed by PCoA for 3 dimensions and additional customers formed by adding (and subtracting) with respect to reference customer **R** as shown in the previous figure (9.3). Additional customers are added by shifting R's values 1, 2 or 3 columns (destinations) to the right or left. These customers are labelled **SL1, SL2, SL3** and **SR1, SR2, SR3**. Both pictures are of the 3D representation, from two different angles, the lower one is a shot from underneath with respect to the upper one. From the bottom picture, it can be seen that SL1 and SR2 are far away from the main group - which is not shown very well in the top 3D shot.

esis 'all the customers in the column of the visual representation make calls to the same destination as my reference customer', one needs to be able to query the dataset 'select all customers who have values > 0 to the same destinations as my reference customer'. This query needs to be able to be formulated and tested *easily*, otherwise the user will not proceed with testing the hypothesis in this way. Though for smaller datasets the problem is simpler because bar charts of individual customers' data can be viewed easily. Each of these various queries and constructions takes the user one step *into* the representation, which can otherwise appear as an incomprehensible mass whose only conveyed information is the proximity of objects, a pattern of colours etc.

9.7 Summary

This chapter has described different ways of querying in the context of visualization. Data can be queried using a conventional query language, but there are significant difficulties in the use of this, notably having to learn the language and obtaining too few or too many hits. Sliders that select ranges for the data to be displayed can be used in the interface, providing a facility described as dynamic querying. Visual querying is also possible, for instance highlighting an entity and finding information about it. Other forms of visual querying are focus+context techniques and brushing one or more entities to view them in another display.

Querying for signature exploration requires the same techniques as usual querying, but includes a subset that relate to the behaviour of the visualization method itself, such as in showing accuracy. To evaluate the usefulness of querying, particularly in relation to the time required for hypothesis generation and testing, the interaction response time is a useful measure.

An illustration of visual querying is given using the calldata set. Highlighting the entities at the extremities of the pattern and viewing impulse charts of the data shows that these entities are the customers with the highest number of calls to a particular destination.

Landmark and query overlap as concepts. Essentially, query finds an answer to a question, landmark provides points within a visual depiction for orientation.

The addition of synthetic data to a real-world dataset under consideration is proposed here as a general element to include in visualization applications. The facility to place specific features within the visualization, proposed here as *feature fingerprinting*, is also recommended.

Examples of feature fingerprinting using a scaling feature (added entities having values scaled with respect to a reference entity) and a shifting feature (added entities having values shifted with respect to a reference entity) are given using the calldata. The examples show that the visualization method considers scaled entities to be similar, but not shifted entities, thus providing the user with insight into the behaviour of the algorithm, as well as the particular dataset. The work underlies the importance, yet difficulty, of providing a range of hypothesis support tools within the interface.

Chapter 10

Elicitation and Application of Feedback Data

10.1 Introduction - Compare, Capture, Modify

The elicitation and application of feedback data for signature exploration is defined on page 83 (Definition 15) as follows:

> *For the **elicitation** of feedback data, the user arranges a set of objects that are known to them on the screen. Real-world data is also available for these objects. The objects are known to the user in the sense that the user has a personal view of some (or all) of their qualities and can arrange the objects on the screen according to their own perceived sense of similarity between objects. The system **applies** this feedback data by using the proximities 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.*

This technique differs from the other four (generic dataset provision, user-construction of data, querying and insertion of landmarks) in that it uses data known by the user (or entities known by the user) to modify the behaviour of the visualization method. The modification of the dataset or visualization method is considered to be another form of signature exploration.

Signature exploration is a way for the user to *compare* their own view of the data with that presented by the system and thus increase their understanding of the representation; the use of feedback provides a means for these two views of the data to interact. The user's sense of similarity is *captured* and used to *modify* the visual representation. The user's sense of similarity between objects may be

captured, or the importance they place upon different aspects. Thus *compare, capture, modify* completes an interaction circle between computer and user: the user compares their sense of similarity with the computer's; the user's sense of similarity is captured by the computer; the computer modifies the visualization method. This process is illustrated in Figure 10.1.

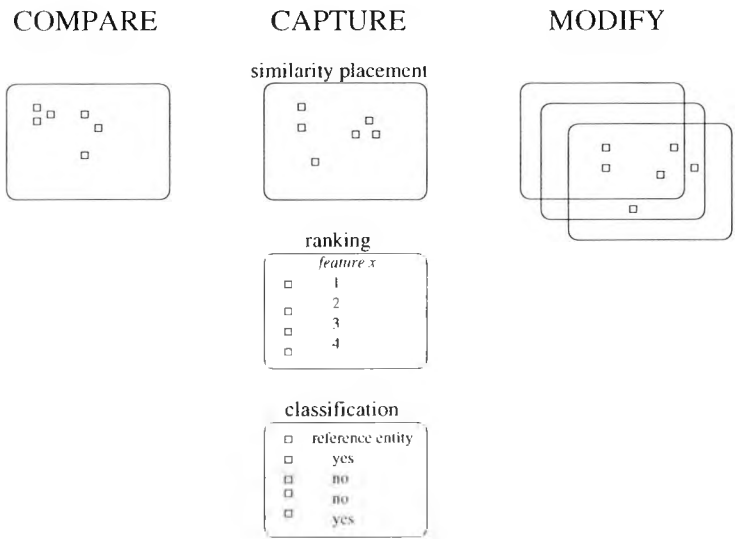


Figure 10.1: The *compare, capture, modify* process illustrates feedback from system to user and user to system. Left: The user gives known data to the application to see how the system places their data. Middle: The user arranges entities on screen by referring to the actual data or their own intuition. Tacit knowledge can be elicited from the user by ranking or classifying with respect to a reference entity or a feature. Right: The system modifies the algorithm (using for example, least squares multiple regression) and replots or presents different possibilities. New data can then be plotted based upon these results.

As an example of how feedback would be used in practise, consider a biologist with high dimensional data relating to a number of proteins. In this set of proteins there is a group that the biologist is very familiar with and can arrange or rate in terms of their similarity. The elicitation and feedback interface will capture their measures of similarity for these proteins and then provide a layout of the whole dataset based upon their assessment of the group they are familiar with, or a predetermined layout that corresponds most closely to this.

The following two sections look at the capture of the user's sense of similarity and the layout modification possibilities, examining some relevant literature (Sections 10.1.1 and 10.1.2). An example illustrates the mechanism of capturing and using the user sense of similarity (Section 10.2). This is followed by a description of the interface for elicitation and feedback that has been implemented in Space Explorer (Section 10.3).

10.1.1 Capture

When this technique was introduced on page 83, the work on the dynamic querying of image libraries was referenced and indicated as the inspiration for this technique (and for signature exploration in general). It was also noted that concepts of similarity are often very subjective. This subjectivity results from the specific query and the user's particular perspective etc. Also, as the user interacts with the system, their sense of similarity often changes, so that the measure should, ideally, be continuously learned (Picard 1995). This change may be brought about by the exploration process and the need to ask different questions or the realization by the user that new features, of which they were previously unaware, are of interest to them.

Is the Euclidean Measure Intuitive?

Keogh and Pazzani (1999), in relation to time series data, claim that there is little evidence that Euclidean distance maps onto human intuition of similarity, though the Euclidean measure, or some approximation or extension thereof, is widely used for time series data. In certain situations the reverse is true. Examples where this is the case are shown in Figure 10.2; this figure shows four datasets each consisting of three time series datasets. In each case the upper two sets appear visually similar, whilst the lower two are similar by the Euclidean measure. Four general features or *global distortions* can be identified (Keogh and Pazzani 1999) that correspond to the differences in the patterns that retain the visual similarity, yet alter the Euclidean measure.

Figure 10.2 contains examples of the *global distortions*, which are illustrated in Figure 10.3. They are: offset translation; amplitude scaling; linear drift and discontinuities. The authors give examples of people for whom different aspects would be important, for instance offset translation would be very important for a doctor looking at patient temperatures, but may not be for a stock market analyst. In each case the shapes result in an arbitrarily large dissimilarity because of a particular feature, here described as a distortion, with respect to a particular shape. Offset translation results where two similar, or possibly identical, shapes are separated in the y-axis. Offset translation can be removed by normalizing the data so that they have the same mean, but this removes information which may be required by some users. Amplitude scaling results where two shapes are similar but one of them has been 'stretched' or 'compressed' in the y-axis with respect to the other. (See Agrawal et al. (1995) for a model that deals with noise, scaling and offset translation). An example of linear drift is the sales of ice cream in two cities with similar climates and populations. The shapes of the time series could be quite similar, but if one city experienced population growth, this would be reflected by an underlying upwards trend in the sales shape. Discontinuities are typically sensor calibration artefacts and can be treated by detection and smoothing or translation. This list is not exhaustive, for instance, phase and frequency scaling could be added. Discontinuities are not meaningful for non time series data and

the other descriptions do not have the same implications as important relationships to discover. Thus a different set of global distortions must be described for different types of data.

Using the concept of global distortions, the following elements are needed in the interface:

- enhancement of user perception of global distortions and how they are treated by the system
- modelling user preference of sensitivity to global distortions
- provision of feedback to the system

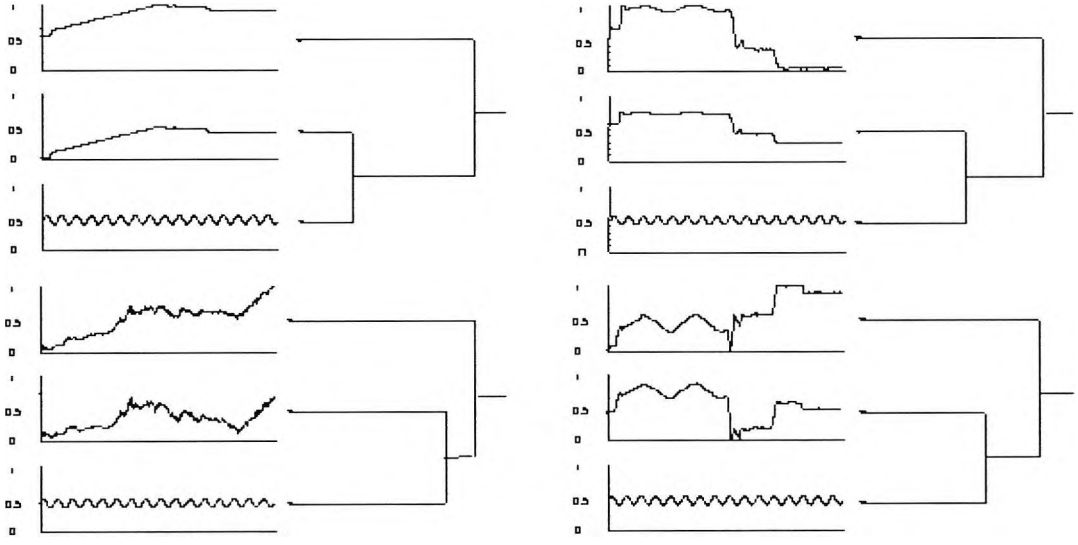


Figure 10.2: The Euclidean distance measure can produce unintuitive clustering of time series data. Clockwise from the top left, unintuitive clustering caused by offset translation, amplitude scaling, discontinuities and linear trends. Although the top two time series appear most similar in all four cases, Euclidean distance indicates that the bottom two are most similar. Figure reproduced from Keogh and Pazzani (1999), ©1999 ACM, Inc. Included here by permission.

Modelling the User's Sensitivity to the Global Distortions

Keogh and Pazzani (1999) also introduce a profile that encodes the user's subjective notion of similarity in a domain, based on the user's *sensitivity* to the global distortions. Users provide a query sequence which may be drawn by the user on the screen, or a sequence from the database. The system ranks all sequences in the database with respect to this query sequence. The best n sequences are shown to the user. The user ranks these and the query sequence is modified in a process analogous to the use of relevance feedback in text retrieval systems. Further improvement in the relevance of the returned sequences can be obtained by embedding the user's global distortion sensitivity profile.

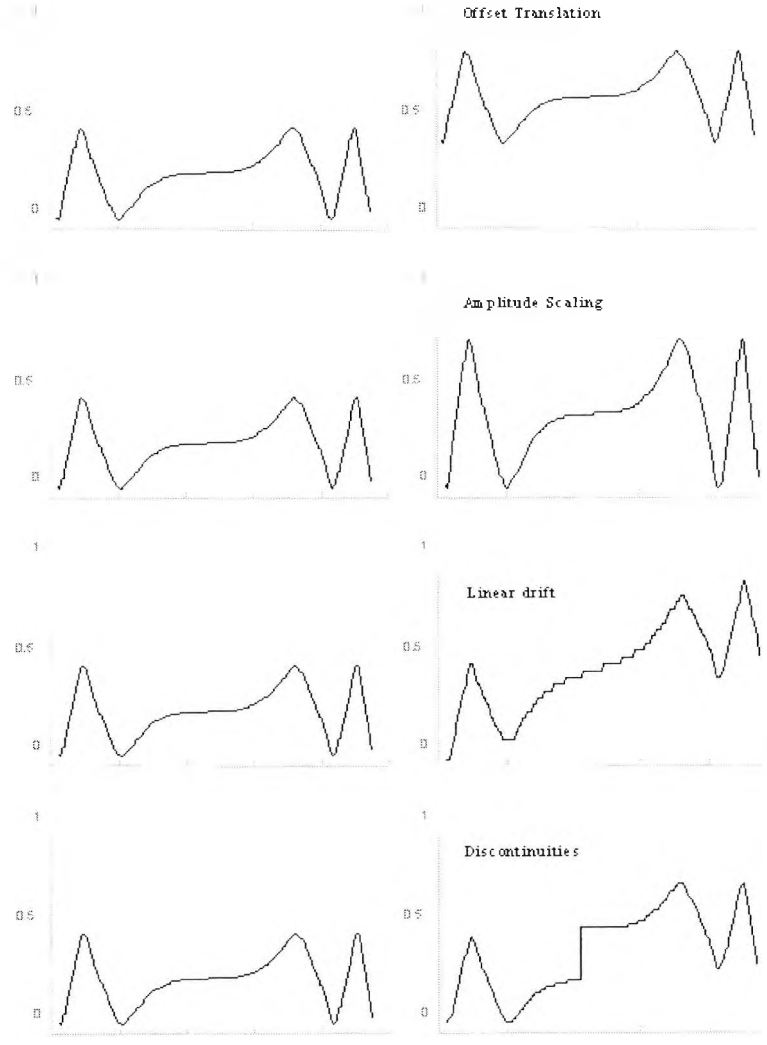


Figure 10.3: The four global distortions described in Section 10.1.1. Figure reproduced from Keogh and Pazzani (1999), ©1999 ACM, Inc. Included here by permission.

Thus the *subjective distance* is measured between two time series Q and S by first shifting, rescaling, retrending and removing relative discontinuities from S to produce a new sequence S'. The distance between Q and S', $DS(Q, S')$, is now measured and the subjective distance is defined as:

$$sub\_dis(Q, S) \equiv DS(Q, S') \times TransformationPenalty(S, S')$$

Where $TransformationPenalty(S, S')$ is the *cost* of converting S into S'. This cost depends on the user's profile, their subjective judgement of the desirability of the four distortions in a particular domain. The cost is calculated from the user's 'preference' distribution for each distortion. The preference distribution is inferred from the user's estimates of similarity between sequences. Note, the Euclidean distance measure remains the measure in use, though modified.

Capturing the User's Sense of Similarity

A variety of interfaces can be envisaged to enable the capture of the user's sense of similarity as indicated in the middle column of Figure 10.1. On the meta-level, where entities are compared to one another in overall terms, the user could be asked to arrange the entities on the screen with the distances between pairs reflecting how similar they consider the pair to be. Here there are two problems: users may lack the knowledge to do this directly and the accuracy of the layout may be coarse, since it is likely that only a rough clustering of the entities would be achieved by the user. For the user who cannot proceed directly, the entities could be ranked with respect to one another (as in the example in the previous paragraph), or simply classified as *similar* or *not similar*. It may be impossible to consider similarity on this general level, in which case ranking and classification can be made with respect to the possession or level of possession of a feature of interest. A list of features of interest, with levels for each entity, generates multivariate data that can be compared to the data that is provided with the entities. This procedure also provides the user with a means of exploring their own view of the entities' qualities and understand which factors are important to them.

Knowledge that the user is unable to articulate, or even be aware that they have, is described as *tacit* knowledge (as introduced on page 98). A variety of techniques have been devised to elicit requirements from tacit knowledge and these provide suggestions for the elicitation of domain knowledge and sense of similarity in visual applications. So that, in the situation where the user is asked to arrange or rank entities, but is unable to, similar techniques can assist. One such method is *card sort* (Rugg et al. 1992); in essence, the card sort process, normally undertaken manually, presents a series of cards to stakeholders for them to sort into groups. Each card has the name of some domain entity written or depicted upon it. The user then says what the criterion was for the sorting, and what the groups were. Repetitions can be made with the user choosing another criterion. When the user exhausts criteria that they can think of, a diadic sort can assist - here two cards are chosen at random

and the user is asked the main single difference between them. The process continues until no further criteria emerge. This method finds attributes that matter to the respondents and suggests an order of importance.

10.1.2 Modify

The user having placed a number of on-screen movable icons or labels representing familiar objects, such that the distances between them represent their similarity, or a measure of the user's sense of similarity having been captured by an elicitation method, the task now is to use this information to modify the behaviour of the measures employed by the system as indicated in Figure 10.1. It is assumed that there is corresponding multivariate or distance data associated with the familiar objects (which may or may not have been the basis of the user's distance arrangement) and that this data is taken from a larger dataset which includes objects with which the user is unfamiliar. The system modifies its representation of the data to incorporate the user's sense of similarity and displays the full dataset by this new means. An intermediate stage can be included where the user is asked to rate various modified representations of the familiar objects.

The scenario of a large dataset containing a small subset of objects with which the user is familiar, capturing the user's distance measure for the known objects and using it to modify the layout of the whole dataset, is assumed here for the general case, though the technique can have other contexts, for instance:

- Understanding the behaviour of metrics, choosing a metric for a specific application: by comparing the system's sense of similarity with theirs, does it coincide? The system can show them what modifications are needed to a selected metric, or indicate which metric is closest to their view. In this case the user's preferences are not applied to the larger dataset.
- Increasing engagement of the user - with the application by exploring meaning, with the data by making the user more aware of their own domain knowledge and preferences.
- Querying of time series and image databases - to improve accuracy (from the user's point of view) of response.

What methods can be used to modify the behaviour of the visualization algorithms? To some extent this depends on the nature of the data, for instance time series data allows the visual ranking of segments according to similarity of shape, whereas the lack of ordering of non time series data makes the shape less meaningful. The most direct option is to choose the layout that is closest to the user defined one, which also provides the user with information that they may be interested in. The method of providing a cost term which qualifies the distance term, based upon the user's tolerance of global distortions, as described in Section 10.1.1, was developed to improve the quality of query returns for

X=

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

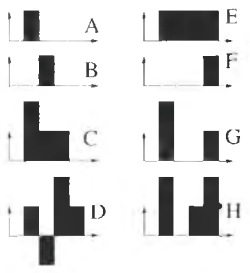


Figure 10.4: Multivariate data for eight entities.

time series databases. This method may be applicable more widely, assuming that a corresponding cost profile, (the degree to which the user tolerates the global distortions), can be built and the method can be generalized to provide a layout of objects rather than a ranking of similarities with respect to a single object. Other possible methods involve solving simultaneous equations, using multiple linear or monotone least squares regression (Gordon 1990, 1999; Kruskal 1964a,b; Sokal and Rohlf 1980), or using learning algorithms such as neural nets and genetic algorithms. The use of simultaneous equations, multiple linear regression and neural nets are described in the following sections.

10.2 Illustration Using Simultaneous Equations

In this example, (which appears in Noy and Schroeder (2001)), 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.

10.2.1 Algorithm

Given multivariate data $X \in \mathcal{R}^{n,m}, n > m$, where n is the number of entities and m the number of attributes and a subset $X' \in \mathcal{R}^{m,m}$ of $m < n$ rows of X. Furthermore let us assume that the user specified co-ordinates for the selected m entities, i.e. $Y \in \mathcal{R}^{m,2}$ is given. Then solve the linear equation $X'|Y$, i.e. convert $X'|Y$ to $I|Y'$, where I is the identity matrix and $Y' \in \mathcal{R}^{m,2}$. Then compute $C = XY' \in \mathcal{R}^{n,2}$, which contains the x and y-co-ordinates for the n entities in its columns.

10.2.2 Example

Consider the following example: There are eight entities $A - H$ and multivariate data $X \in \mathcal{R}^{8,4}$ shown in Figure 10.4.

The user knows about the four entities $A - D$ and draws a layout of them as shown in Figure 10.5.1. According to the above algorithm we have $X'|Y$

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

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 co-ordinates XY', which are generalised from the subset $A - D$ and applied to all entities $A - H$, the layout is as shown in 10.5.2. 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. 10.5.1), whereas all others put them closer to each other, in particular correlation (Fig 10.5.6). Now consider the entities the user did not know about. The generalisation of the user distances places e.g. F near to A, B, which is done by the Minkowski distance family, but not at all by correlation. On the other hand, the user's initial placement of ABCD is generalized by essentially separating ABCD and EFGH and none of the metrics separates these two sets, though City distance and correlation do to some extent, correlation is the best (from the visual examination). The mean square distance error is often used to compare layouts, this is the mean of the sum of the squares of the differences between distances between all pairs of entities in the one layout and the other, i.e. for two layouts, a and b, of n entities:

$$\varepsilon_{ab} = 2/n(n-1) \sum (\delta_{aij} - \delta_{bij})^2$$

where δ_{aij} is the distance between the entities i and j in layout a. However, the mean square distance error here gives, for instance, a better value for PCA (1.6) than correlation (3.2), which conflicts with the visually observed separation of the two groups, thus indicating that this way of measuring which of the six metrics comes closest to the user's layout might not be the best.

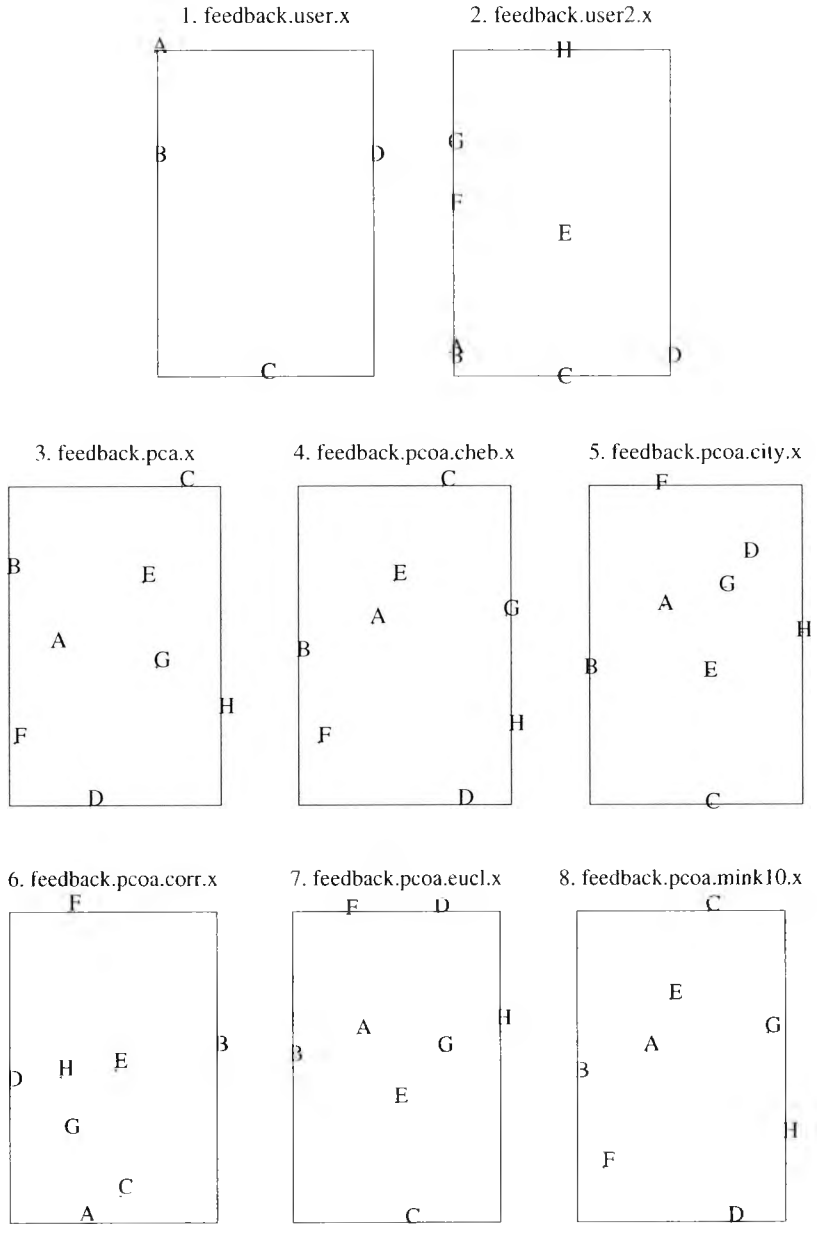


Figure 10.5: 1. Four entities placed by a user. 2. Generalisation of these placements and application to all entities. 3-8. Scatterplot of the eight entities using PCA and PCoA with Chebychev, City or Manhattan (Minkowski with $\lambda = 1$), angular, Euclidean and Minkowski ($\lambda = 10$) distance. From visual comparison of the layouts, the user's initial placement of ABCD (1) is generalized by essentially separating ABCD and EFGH (2). None of the metrics (3-8) separate these two groups, though City distance (5) and correlation (6) do to some extent, correlation the most. On the other hand, the commonly used mean square distance error gives a lower value for PCA (3) than for correlation, indicating that PCA is closer to the user's layout.

10.3 Interface Illustration Using Multiple Linear Regression

The interface is extended so that the scatterplot entities can be moved around in the display and the distances between them captured. An example is given in Figure 10.6, a subset of the Iris dataset (introduced on page 92) is displayed and the user then moves the squares on the screen to the arrangement that they think reflects the objects' similarities. Here it is assumed that the user is familiar with these 8 instances. Figure 10.6 shows the view before rearrangement by the user on the left and after arrangement on the right. When the user is satisfied with their arrangement, they click the 'capture distances' button and the distances are calculated and stored.

Multiple least squares linear regression is used to derive weights for the attributes as follows. The following information is available:

$$\Delta = (\delta_{ij})$$

where δ_{ij} is a measure of dissimilarity between the i-th and j-th objects, captured from the arrangement on screen of the objects;

$$\mathsf{D}(w) = (d_{ij}(w))$$

is the distance matrix based upon the values in the multivariate data table where d_{ij} is the Euclidean distance between the i-th and j-th objects, w are weights associated with the variables (see Equation 3.3 on page 37).

The aim is to find values for the weights, $w = (w_1, w_2, \ldots, w_m)$, (for m attributes), such that $D(w)$ is a good approximation to Δ, i.e.

$$\delta_{ij} \approx c d_{ij}(w), \quad c > 0$$

The linear regression model can be used here:

$$y_t = f_t(\beta) + u_t, \quad t = 1, \ldots, n$$

- y_t is the t^{th} observation on the dependent variable, which is a random variable
- β is a vector of unknown parameters
- $f_t(\beta)$ is a regression function which determines the mean value of y_t conditional on a specified set of independent (explanatory) variables x_t and on β. This function varies from observation to observation as x_t varies
- u_t is an error term.

Thus

$$y = X\beta + u$$

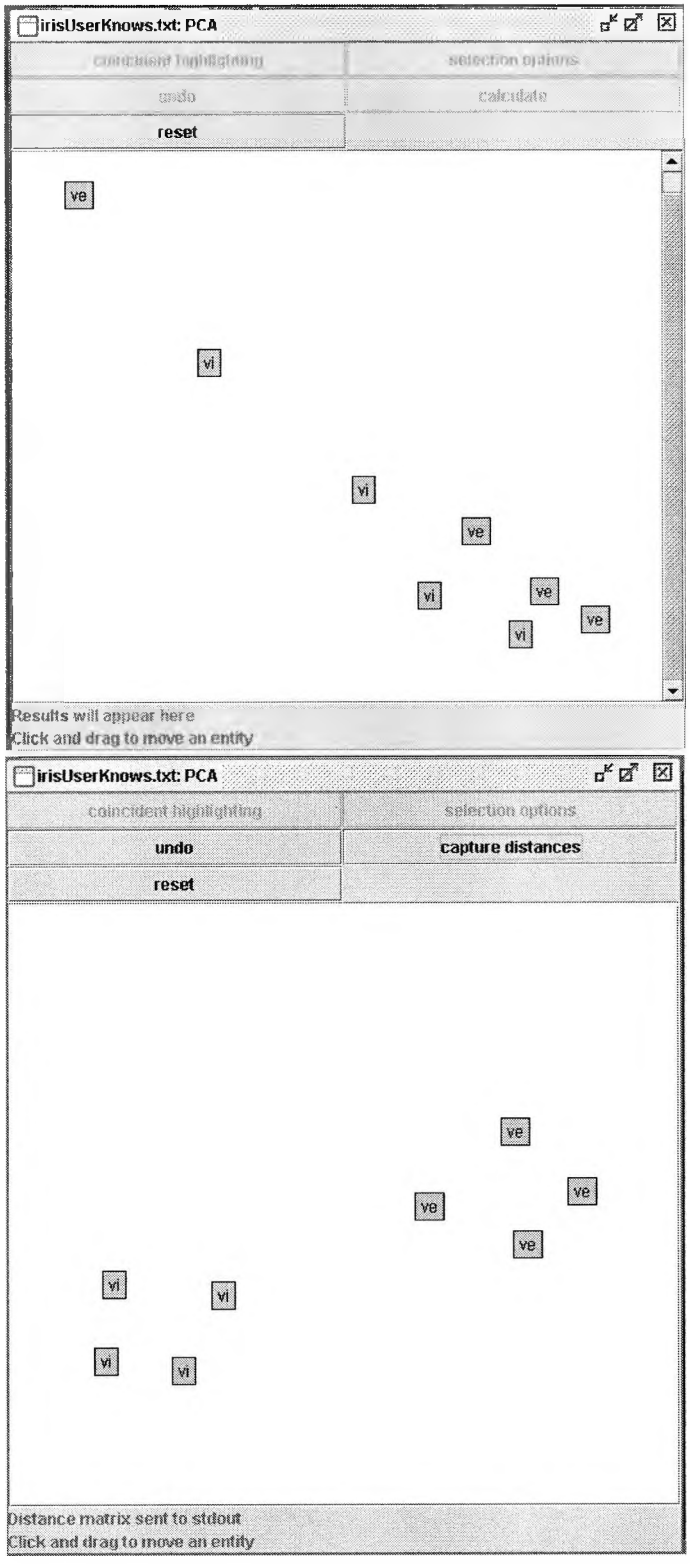


Figure 10.6: Capturing user similarities between entities on screen using a subset of the Iris dataset. Top: before user arrangement. The user can click and drag the entities to wherever they want them to be. When they are satisfied with their arrangement, they click the 'capture distances' button for the distances to be calculated and stored (bottom).

Weight Estimates			
w_1 sepal-length	w_2 sepal-width	w_3 petal-length	w_4 petal-width
0.26	0.19	0.18	0.37

Table 10.1: Optimal weights of the four variables of the Iris data using linear regression.

where

$$X = \begin{bmatrix} x_{11} & x_{21} & \dots & x_{k1} \\ x_{21} & x_{22} & \dots & x_{k2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{ln} & x_{2n} & \dots & x_{kn} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

for n observations and k independent variables. Ordinary least squares estimation of β is:

$$\hat{\beta} = (X'X)^{-1}X'y$$

Thus for this case:

$$\vec{w} = (X'X)^{-1}X'\delta$$

and the values x_{kn} are the squared differences between attribute k values for the n^{th} pair of objects.

The implementation in the interface of SpaceExplorer uses the java matrix library, JAMA[1], for this calculation. The weight values obtained from the distances captured on screen are shown in Table 10.1. Note that 0.25 is the optimal obtained value, since there are four variables. Once the weight values have been estimated from the squared distances entered by the user, they can be used to display the larger dataset as shown in Figure 10.7. This figure shows that the overlapping classes are not moved appreciably further apart, though the user arrangement of Figure 10.1 clearly does. These overlapping classes are known to be difficult to separate, for instance, even the use of support vector machines cannot classify them without errors (Jong et al. 2003). However, there are other difficulties in this elicitation and application of feedback: the user's arrangement of entities may be highly approximate; the relationship between the data and the user's layout distances may be non-linear; the user may be using (tacit) knowledge they have which does not correspond to the data, (i.e., in this case other than sepal and petal lengths and widths), and could contradict it.

10.4 Using Learning Algorithms

Supervised learning relates to the classification problem, where the learner is required to learn, or approximate the behaviour of, a function which maps a vector into one of several classes by looking

[1] http://math.nist.gov/javanumerics/jama/

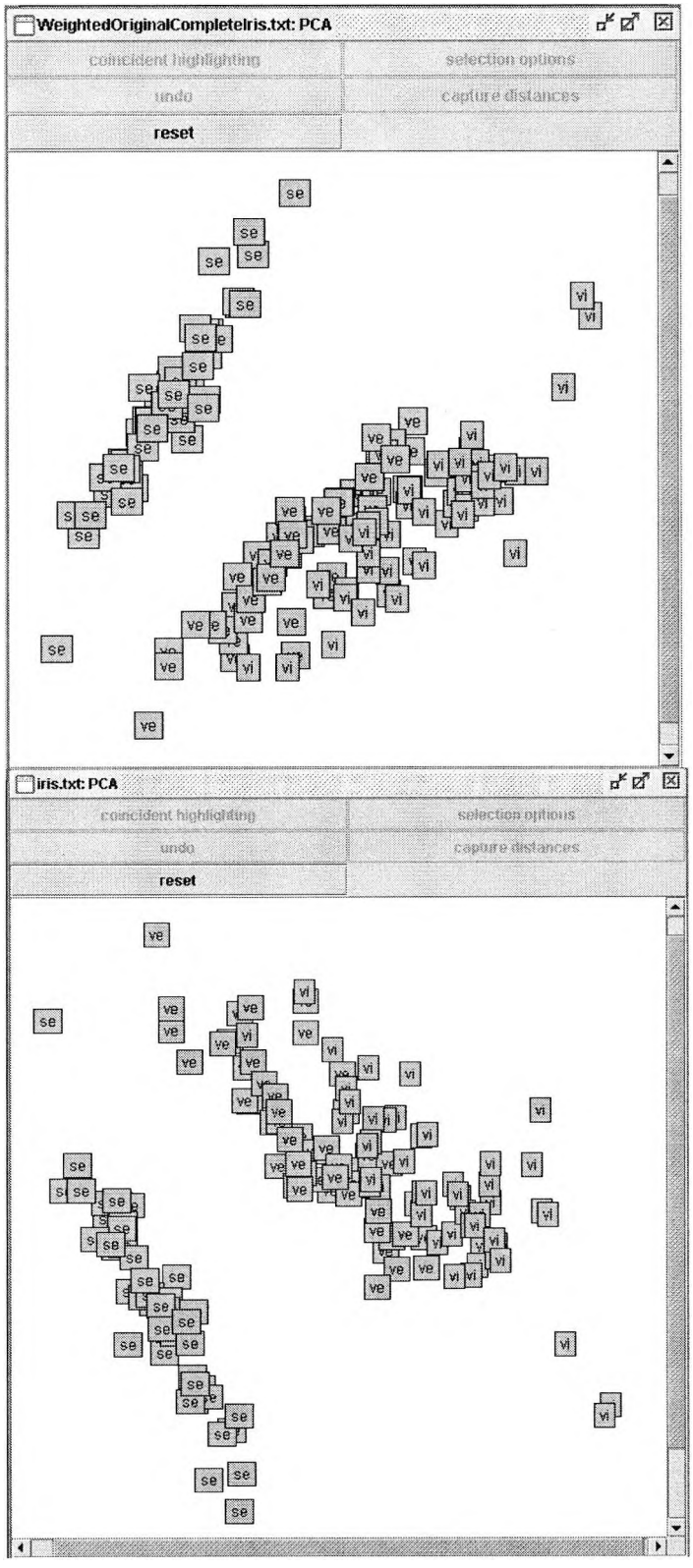


Figure 10.7: 2D layout (after PCA) of the Iris dataset (top). 2D layout of the Iris dataset after attributes have been weighted with weights in Table 10.1 (bottom).

at input-output examples of the function. The output can be a continuous value or can predict a class.

10.4.1 Neural Nets

The Stuttgart Neural Net Simulator was used to examine the usefulness of neural nets for learning the distances provided by the user moving the subset of entities on the screen (as described in the previous section). This can be downloaded from http://www-ra.informatik.uni-tuebingen.de/SNNS/. A version with a Java interface is available, and this was used here.

A neural network (strictly an *artificial* neural network) is an interconnected group of nodes, inspired by the neurons in the human brain (Bishop 1995). The nodes, also described as *units*, have directed weighted links between them. There are different types of neural networks, here a typical single hidden layer *feedforward* network with *back propagation* has been used. When the network is in operation, a value is applied to each input node. Each node passes this value to the connections leading out of it, the value being multiplied by a weight. The next layer nodes each receive a value that is the sum of the values from connections to it. Each node then performs a computation on the value, often a *sigmoid* function is used. The process is continued for the next layer. The sigmoid curve introduces non-linearity into the computation. Back propagation is the name given to a type of learning technique where the output values are compared with the correct answer and an error function computed. The error is then fed back to through the network, adjusting the weights. As the process is repeated, the network usually converges to a state where the error is small.

There are three types of unit (node): input unit, hidden unit and output unit. In the example described in the previous section, the test data consists of the distances between all pairs in the group 8 entities taken from the Iris dataset and arranged on the screen by the user. Each Iris example has four values for petal width and length, and sepal width and length. This investigation considers two ways of applying the neural net model. In the first case 8 inputs (pairs of vectors), a hidden layer with various numbers of units and 6 outputs (the range of values for the distances divided into 6 categories). In the second case, the same number of inputs and hidden units, but only one output unit. In the first case the sigmoid function is used for the output units, in the second case, the identity function is used. In both cases the sigmoid function is used for the hidden units.

Since there are only 8 entities concerned, there are only $n/2(n-1) = 28$ (the number of pairwise distances) training examples. In general there is a problem of developing a network that performs well on examples not used as training examples, particularly, as in this case, for very limited numbers of training examples. The network may *overfit* the training data. A simple heuristic, *early stopping*, is where the training set is split into a new training set and a validation set. After each sweep through the new training set, the network is evaluated on the validation set. The network with the best performance on the validation set is then used for actual testing. Examination of the error graphs for these

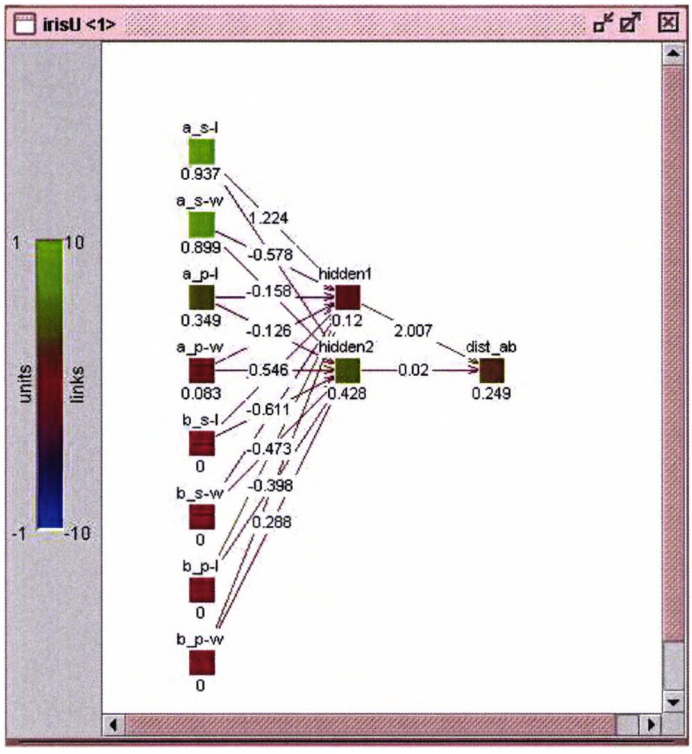


Figure 10.8: The single hidden layer feedforward net used to learn the distances obtained from the user arrangement of 8 iris examples on the screen (shown in Figure 10.6. This is a screenshot of one of the windows in the Java interface for the Stuttgart Neural Network Simulator. The inputs are the four values (s-l: sepal length, s-w: sepal width, p-l: petal length, p-w: petal width) for patterns **a** and **b**. For example, **a_s-l** is the sepal length for **a**. The output, **dist_ab**, is the distance between **a** and **b** in the user arrangement.

two datasets usually shows that, beyond a certain point, as the accuracy for the training set improves, the accuracy for the validation set worsens. *Cross-validation* is a more complex form of validation where the training set is divided into k subsets of approximately equal size. The net is trained k times, each time leaving out one of the subsets from training, but using the omitted subset to compute the error. For the first example early stopping was used, but gave an error rate of only 25% (per cent of correct outputs for the validation dataset) for 2 hidden layers. Using 7-fold cross-validation shows that the values obtained are consistent, though the error increases to 50% in some cases (this is 2 out of 4 correct, since there are only 4 patterns in one 7-fold cross-validation set). The error on the training set was typically around 80%. Increasing the number of hidden layers to 3 improved the accuracy on the training datasets to 100%, but decreased that for the validation ones, i.e. overtraining. Since the error was so high for these nets, it was decided to try the single output (with the identity function for the output). This gave a better result: 100% for the training data and 82% for the validation data (for a typical cross-validation pair), so this was the configuration used for the final net. This net was used to generate distances for the whole of the Iris dataset, by presenting all pairs of vectors as inputs.

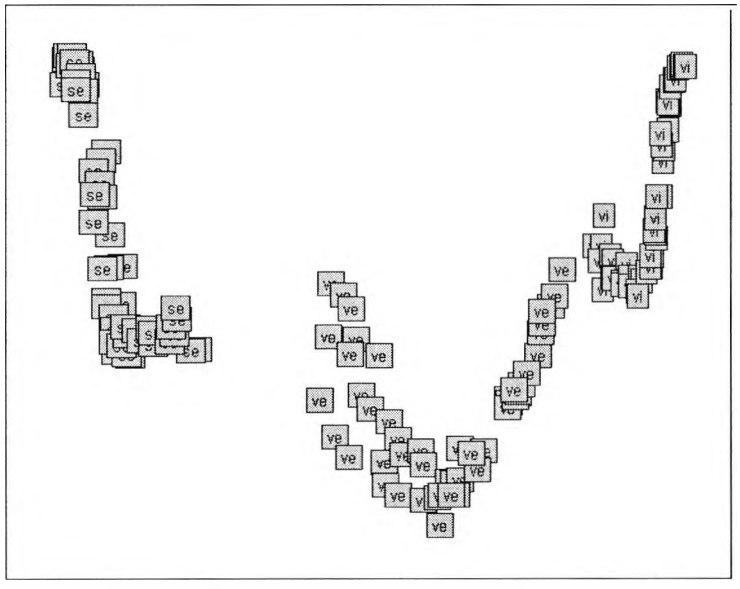


Figure 10.9: The layout of the Iris dataset using a neural network trained on the user distance data for a subset of 8 examples (shown in Figure 10.6. This shows that the species Iris-versicolor (ve) and Iris-virginica (vi) are now in separate clusters. Compare this to the original shown in Figure 10.7 (top), where these two species overlap.

The final net used is shown in Figure 10.8 and the results of display with PCoA are shown in Figure 10.9. This shows that the species Iris-versicolor (ve) and Iris-virginica (vi) are now in separate clusters, which follows the user arrangement. This is an improvement over multiple least squares linear regression, which is to be expected because of the non-linear nature of the problem. However, the use of neural nets for cases with such small training sets can be problematic and this result may not hold in general. Also, the training of the net was done off-line, whereas the multiple least squares linear regression can be done automatically.

10.4.2 Genetic Algorithms

Genetic algorithms (Rawlins 1991) are inspired by Darwin's theory of evolution. An evolutionary process is used to find a best, or *fittest*, solution. In each cell of a living organism there is the same set of chromosomes. Chromosomes are strings of DNA and serve as a model for the organism. A chromosome consists of blocks of DNA called genes. Each gene encodes a particular protein. Basically, it can be said that each gene encodes a trait, for example colour of eyes. A complete set of genetic material (i.e. all chromosomes) is called a *genome*.

During reproduction, *recombination* (or *crossover*) first occurs - here genes from parents combine. *Mutation*, where the elements of DNA are changed by copying error (or other means), can also occur. The fitness of an organism is measured by success of the organism.

Evolutionary algorithms operate on a population of potential solutions applying this principle of

survival of the fittest to produce better and better approximations to a solution. At each generation, a new set of approximations is created by the process of selecting individuals (or genomes) according to their level of fitness and breeding them together using operators such as recombination and crossover.

Code was written to develop a model and thus discover weights for the variables using genetic algorithms. Various polynomial relationships were examined (for simplicity, using only squared terms, then adding cubed terms and so on) and weights determined to minimize the utility (maximizing the fitness). The utility being the sum of the squared differences between the distance obtained by computing with the model (and selected weights) and the distance given by the user in their screen layout. The general form of the polynomials used was

$$\delta_{ij} = (x_{i1}^n\omega_1^n + x_{i1}^{n-1}\omega_1^{n-1} + \ldots + x_{i1}\omega_1) + (x_{i2}^n\omega_2^n + x_{i2}^{n-1}\omega_2^{n-1} + \ldots + x_{i2}\omega_2) + \ldots \\ +(x_{j4}^n\omega_8^n + x_{j4}^{n-1}\omega_8^{n-1} + \ldots + x_{j4}\omega_8)$$

Where δ_{ij} is the distance between entities i and j as arranged by the user, x_{i1} is the value of the variable x_1 for the entity i, w_1 is the weight for the variable x_1 that needs to be determined, and n is the order of the polynomial that is to be used. There are four variables for each of the entities, i, j, and thus 8 weights to determine. Results are given here for orders 2, 3 and 5. Various numbers of generations were used to see whether overfitting was a problem.

The four main stages of the approach adopted are described below:

- Selection. This is the process by which genomes are selected for reproduction. Tournament selection has been used for this evaluation: here a number of individuals are chosen randomly from the population and the best individual from this group is selected to be a parent. The parameter for tournament selection is the tournament size *Tour* and takes values from 2 to the number of individuals in the population. A value of Tour of 3 has been used here.

- Recombination. There are numerous methods of recombination available, the method used here is called discrete recombination. Discrete recombination works by looking at the individual values within each genome. A random number from the set 1, 2 is assigned, the numbers 1 and 2 being the potential parents. Offspring take the genome from the assigned parent for that value.

- Mutation. Offspring values are mutated by the addition of small random values, with low probability. The probability of mutating a variable is set to be inversely proportional to the number of variables. Different results have been reported for the optimal mutation rate and the mutation rate can be varied through the generations. However, a mutation rate of $1/n$ has been shown to produce good results for a range of test functions, though varying the mutation rate has been shown to produce only an insignificant improvement. In these examples a mutation

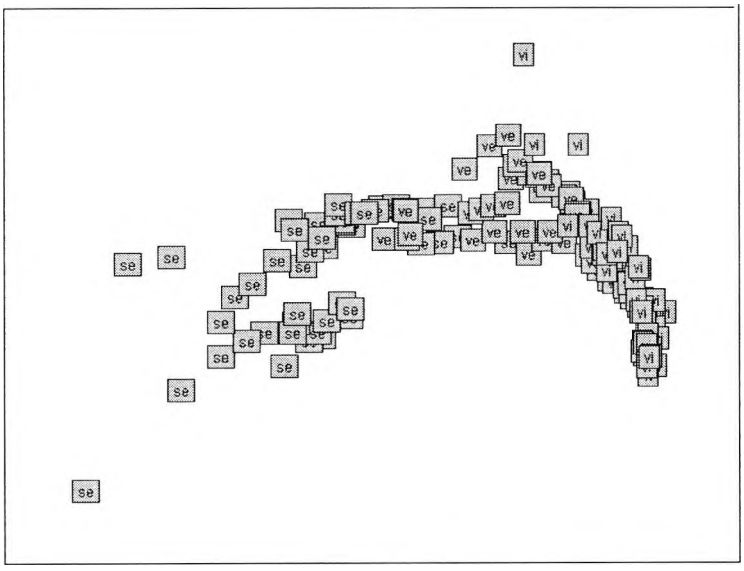


Figure 10.10: The layout of the Iris dataset using weights derived using a genetic algorithm: using polynomial relationship of order 2. Genetic algorithm run for 700 generations, utility 15.

probability factor is used and the value of 0.01% found to be effective.

- Reinsertion. Reinsertion is where the newly produced children are reinserted into the population. Here a reinsertion process is used called *elitist reinsertion*. This is where less offspring than parents are produced and then used to replace the worst parents. The same number of offspring as parents are produced, and then sorted by fitness, subsequently taking a percentage of the most fit from both sets. The percentage used in this example is 50.

Results of displaying the whole of the Iris dataset with the weights derived for polynomials of order 2, 3, 5 (only using terms of order 5) and 5 (using terms of all orders) are shown in Figures 10.10 to 10.13. New distances are calculated including the weights, then a display is found using PCoA. The utility achieved in each case varied, and the assessment of the result was visual. None of the examples separate the three clusters and the separation of the 'se' cluster from the other two, that is quite clear after PCA, has been lost. This problem with the 'se' group is to be expected since the training data did not include any examples from this group. The clustering appears slightly better in the higher order pictures, despite the fact that the utility is higher. This indicates that it would be useful to explore the use of utility measures based upon how well the clustering was achieved, rather than a comparison with each individual distance, though such measures are difficult to specify. However, in the proposed feedback scenario, a classification would not normally be available, so that this kind of utility measure would not be possible. In order to use genetic algorithms within the feedback interface, the model used needs to be evolved, as well as the weights, and the utility function needs to be explored to see why the better arrangements have a higher utility.

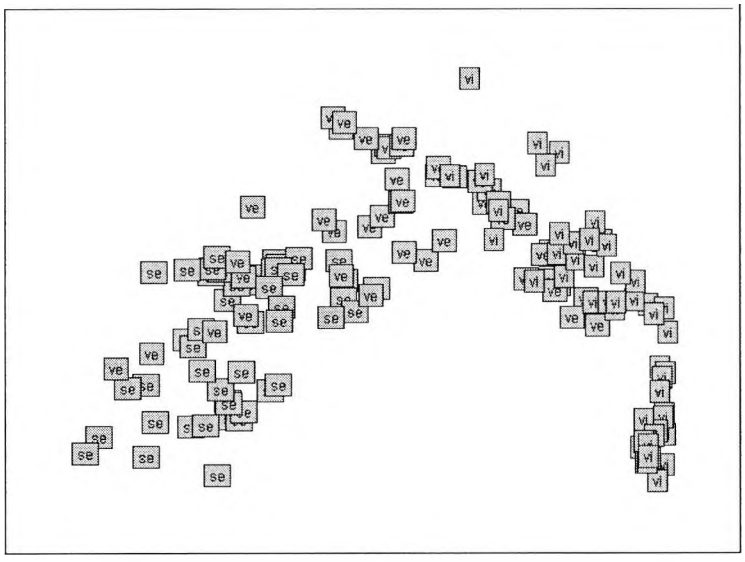


Figure 10.11: The layout of the Iris dataset using weights derived using a genetic algorithm: using polynomial relationship of order 3. Genetic algorithm run for 1,000 generations, utility 21.38 (no improvement for 10,000 generations).

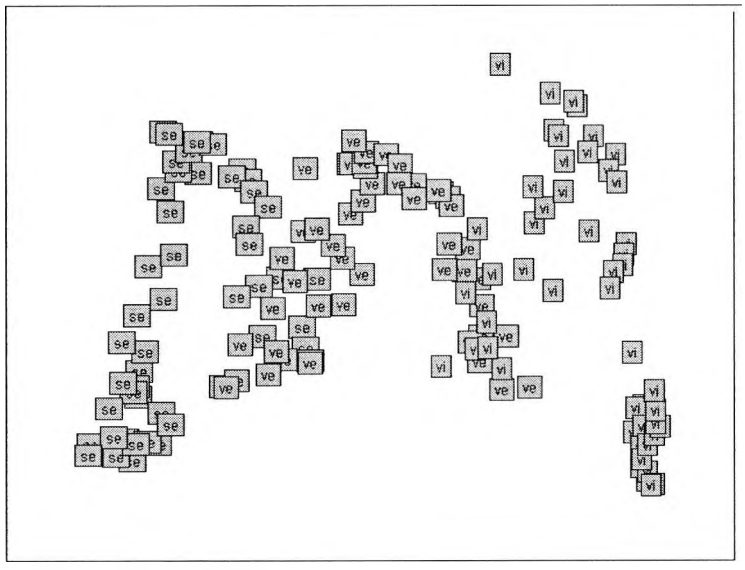


Figure 10.12: The layout of the Iris dataset using weights derived using a genetic algorithm: using polynomial relationship of order 5, including only the order 5 terms. Genetic algorithm run for 500 generations, utility 30.5.

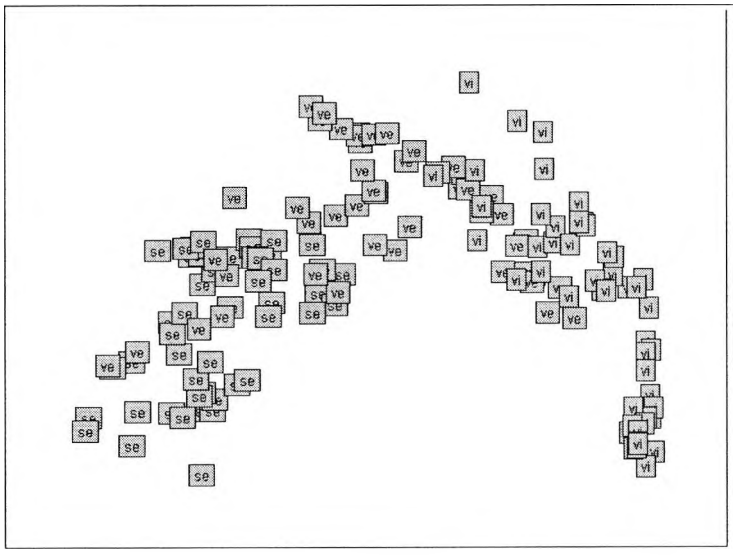


Figure 10.13: The layout of the Iris dataset using weights derived using a genetic algorithm: using polynomial relationship of order 5 (all terms). Genetic algorithm run for 500 generations, utility 22.75.

10.5 Summary and Conclusions

The elicitation and application of feedback data for signature exploration has the three aspects: compare, capture and modify. The user compares their sense of similarity (between entities) with the computer's; the user's sense of similarity is captured by the computer; the computer modifies the visualization to take account of the user's sense of similarity and to display a larger number of entities.

Relevant work in the dynamic querying of image libraries emphasizes the subjective nature of concepts of similarity. Researchers involved in the querying of time series data question the intuitivity of the Euclidean measure, though this measure is widely used. Their work shows that, where offset translation, amplitude scaling, linear drift and discontinuities (termed the 'global distortions') are involved, the Euclidean measure gives an unintuitive result. These global distortions also correspond to aspects about the data which the user may wish to disregard. Thus the user's sensitivity to these global distortions can be modelled and used to calculate the subjective distance between two sequences.

Capturing the user's sense of similarity can be approached in a variety of ways: by relevance feedback; by giving the user a variety of examples to classify, or rank, with respect to a reference entity; by the user arranging the entities on the screen where the distances correspond to dissimilarity; by the use of an elicitation technique similar to those often employed in the elicitation of requirements (for example 'card sort').

There are also a variety of methods for modifying the behaviour of the visualization method: the technique that provides results closest to the user's arrangement can be ascertained; the metric can

be modified based upon the user's sensitivity to global distortions; weights for the attributes can be found (e.g. using multiple least squares linear or monotone regression, and learning algorithms).

Two illustrations are given for the application of feedback data based upon the user arranging known entities taken from a larger dataset. In the first illustration, the user positions objects on the basis of perceived similarity. A mapping from the attribute values to the x,y co-ordinates is then obtained by solving the linear equations. This mapping can then be applied to the larger group of objects. In the second illustration, implemented within the SpaceExplorer interface, the system captures a set of distances between objects and uses them to calculate weightings for the individual attributes using multiple least squares linear regression. The user's sense of similarity is captured in an interface which allows the user to move objects around the screen, by clicking and dragging. When the user is satisfied with the arrangement, such that the distances between objects reflect their (the user's) view of the objects' similarity, the distances are captured and used to calculate attribute weights. An example using the Iris dataset is described, and weights calculated using multiple least squares linear regression, neural nets and genetic algorithms. The best results were obtained with the use of neural nets.

There were several problems encountered:

- Difficulty for users to accurately quantify similarity. Thus the distances obtained from the user must be regarded as quite approximate. Better results may be obtained by applying elicitation techniques, such as card sort, or modelling the user's sensitivity to global distortions, or using ranking and classification methods. These methods would capture the user's knowledge and perspective more accurately and, dependent upon the technique, in more detail.

- The application of multiple least squares linear regression and the learning algorithms is limited in scale. This method can only be used where the number of objects (whose similarity is estimated by the user) is greater than or equal to the number of attributes. Therefore, this method could not be used, for instance, on the calldata set, where the complete data table has a higher number of destinations than customers. In any case, a larger number of destinations than about 20 would be difficult for the user to arrange (the arrowhead example used 14). Thus the other methods involving ranking and comparison of individual attributes, as well as direct ones modelling user preferences, should be used as well. Similar restrictions apply to the use of learning algorithms, where, though possible, results lose statistical validity.

- Complete mismatch of data and layout. The user could be arranging the data based upon information other than that in the dataset being used.

- Impracticality of learning algorithms within the interface. The genetic algorithm and neural net applications, as they have been used in these experiments, cannot be used in real time within

the interface. The extent to which these methods could be used *unsupervised* has not been established - this is a serious restriction upon their use.

Nevertheless, aside from these problems, this kind of feedback interface is useful for several other reasons: for engaging the user in the process of developing understanding of their *own* sense of similarity, not just that of the application; to help the user identify which attributes, if any, are not of interest to them; to help users determine their own weights directly.

Can expressions of similarity other than proximity, such as colour or glyph height, be captured? If colour or glyph height is used to denote similarity, then the user could be given the opportunity to change the colour, or height, rather than the position. However, such quantities are rarely used after dimension reduction, because they are equivalent to reducing the dimensions to one. In the case of colour, there is also the problem that colour scales are perceived in a non-linear way (even when carefully constructed). Nevertheless, it would prove easier for the user to express their own sense of similarity in this way, though they also need to be able to arrange the elements in groups, where some distances (or colour differences) are precise and others not. Thus a more expressive interface is needed, which can make these distinctions, whether colour, height or distance is used.

Will the elicitation and application of feedback data work with visualizations that do not use dimension reduction, such as parallel coordinate plots? The application relates to similarity, so it is only relevant where there is dimension reduction producing a layout where proximity of position relates to similarity (or where similarity is mapped to some other quantity such as colour, as discussed in the previous paragraph). Forms of elicitation (without subsequent application) could be applied to find out what is important to users, where this was relevant. For instance, to help users choose attributes to display, or ordering of attributes, or for personalization.

The elicitation and application of feedback has been used in the querying of image and time series databases, but not as a means of exploring the behaviour of visualization applications. This work has illustrated that the technique could be used more generally for visualization, and that there is considerable scope to expand it for both visualization and querying. Expansion in terms of different kinds of interfaces for capturing the user's sense of similarity or preferences and of different ways of modifying the behaviour of the application, based upon these captured quantities.

Chapter 11

Obstacle: Accuracy of Depiction

11.1 Introduction

An area of difficulty that has arisen in drawing conclusions from this work and in determining future directions concerns accuracy:

- In reducing dimensions an abstraction error occurs, so that layouts are necessarily approximate. What are the implications of this from the point of view of interpreting patterns in the display? How can these errors be shown to the user?

This chapter is a discussion of this area, examining relevant literature. From the point of view of signature exploration the corresponding question is:

- How can abstraction error be demonstrated to users? (An aspect of revealing the behaviour of the visualization method.)

The discussion includes an empirical examination of error for an application involving agent interest profiles which leads to the proposal of a new layout algorithm and a mechanism for agents to exchange approximate profiles without revealing their detailed profiles or involving a third party.

11.2 Error and its Sources in the Visualization Process

The three words *accuracy*, *error* and *abstraction* relate to the loss of information resulting from the process of taking data from the real world and finding a form of visual representation for it. We can distinguish these terms in the following way: accuracy measures the discrepancy from a modelled or assumed value; error measures discrepancy from the true value (Buttenfield and Beard 1994); abstraction is a more general term, which may be used to apply to any loss of information. This loss

of information is inevitable because of the discretization of measurement, both from the technical and human conceptual points of view as expressed by Goodchild et al. (1994):

> "By definition, reality is continuous, while the observation of reality is discrete. Technology discretises measurement, as for example in satellite image 'snapshots' taken at regular intervals in comprehensive scanning paths. Perception also occurs in discrete 'chunks', is selective and easily masked or distracted."

The loss of information is also entailed in the abstraction of the application of the representation to the data. In the case of dimension reduction this representational loss (which may be expressed as a mathematical error term - a truncation of a matrix) may be very large. However, any representation has the potential to provide a loss of data or introduce a distortion (for instance by using a colour scale, which is not perceived linearly, to represent a linear scale), notwithstanding that such effects may also reveal facts about the data. Effects such as overplotting can be considered to introduce error, since a loss of information arises in the representation.

The different contributory factors affecting the accuracy of a visual representation arise at different stages of the visualization process as illustrated in Figure 11.1: the accuracy of the original data (1); the accuracy of the transformation from data to visual depiction (2 and 3); the accuracy of the interpretation (4). This figure emphasizes the transformational aspect of the visualization process. A cartographic exploration of this transformational aspect is given in Tobler (1979) where it is said that 'the entire process of making, and using, a map can be viewed as a sequence of transformations'. The discussion in this chapter is concerned primarily with the accuracy of the transformation from data to visual depiction (2 and 3) and indirectly with the accuracy of interpretation (4) from the point of view of providing the user with necessary information about 2 and 3, rather than the mechanisms of interpretation (i.e. of cognition) themselves. This focus is because the aim of this work is to increase the user's understanding of the behaviour of the visualization method. However, in considering how to convey information about error to the user, it is useful to look at examples of visualization of general data uncertainty from the literature to see if they can be used more widely. There is much work on this topic in geovisualization which discusses *data validity*, *uncertainty* and *quality*. Validity as a term is used to imply testable elements (Goodchild et al. 1994), data quality in geovisualization is precisely defined. More references to the geovisualization literature are made in the following sections.

Examples of error sources for colourmaps, parallel coordinate plots and scatterplots are shown in Figure 11.2.

It should be noted that loss of information is not, of itself, undesirable. The issue is whether the resulting abstraction is appreciated by the user, who can than assess its validity. Consider, a synthetic dataset containing several clusters; now add randomly distributed noise. A dimension reduction

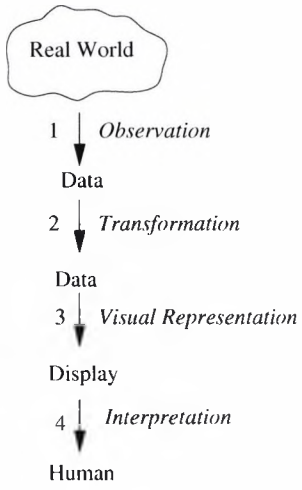


Figure 11.1: Sources of error in visualization: 1. Observation: from real-world to data. 2. Transformation: from data to data (in preparation for visualization). 3. Visual representation: from data to display. 4. Interpretation: from display to human.

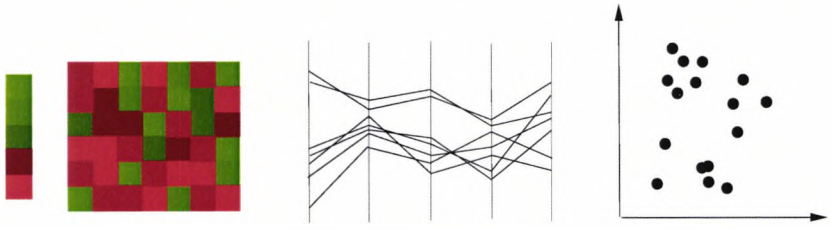


Figure 11.2: Examples of error in visual depictions: colourmap (left)- visual representation error in non-linear perception of colour scale (data to display); parallel coordinate plot (middle) - possible interpretation error (display to human) and visual representation error due to overplotting (data to display); scatterplot after dimension reduction (right) - transformation error due to dimension reduction (data to data), visual representation error due to overplotting.

process may result in information loss, but if this is the removal of the noise, the result is meaningful. The process of abstraction is a fundamental principle of 'how an information processing organism or machine reduces the otherwise unmanageable glut of information' (Card et al. 1999, p. 11):

> "[T]here appears to be a general *Principle of Selective Omission of Information* at work in all biological information processing systems. The sensory organs simplify and organize their inputs, supplying the higher processing centers with aggregated forms of information which, to a considerable extent, predetermine the patterned structures that the higher centers can detect. The higher centers in their turn reduce the quantity of information which will be processed at later stages by further organization of the partly processed information into more abstract and universal forms." (Resnikoff 1987, p. 19) (As cited in Card et al. (1999, p. 11))

11.3 Accuracy Estimators and Depiction

In assessing the accuracy of a layout we can give an overall measure which is an average of the errors, but we can also ask specific questions, such as: 'If I draw a radius of 1 unit around this point, how many points will be included in this circle that should not be there, and how many are outside this area that should be inside it?' Those points included inside the circle that should not be there are described as *false alarms* or *errors of commission* or Type I errors (in statistics) and those that are outside, that should be inside, are described as *false dismissals* or *errors of omission* or Type II errors. This issue is relevant in indexing large databases of time series data (for instance) where the user wants to ask the question, 'How many entities are within a distance of x units to this one?' This is an example of similarity search, useful for exploring time series databases, but also important in clustering and classification (Böhm et al. 2001; Keogh et al. 2000). Queries of this kind are difficult on large datasets, so indexing methods are used that reduce the dimensionality of the dataset. However, the condition that there are no false dismissals needs to be preserved, though false alarms are allowed. False alarms can be removed in a post-processing stage, though it may still be desirable for them to be relatively infrequent. Similarly, in a visualization, the user may want to draw a circle around a point and know the number of false alarms/false dismissals within that area.

In most visual depictions of high dimensional data, there is no indication within the visualization of the level of abstraction that has occurred. This applies to self-organizing maps, as well as scatterplots following the application of dimension reduction algorithms, though scatterplots are often accompanied by eigenvalue spectra, which give some indication of the loss of information in the data-data transformation. The following sections discuss relevant work that is in the literature of the mathematics, statistics, geovisualization and pattern recognition fields. This work was found

during the literature study and in discussions with researchers at agent, information visualization and datamining conferences.

11.3.1 Geometry of Graphs

The study of the error involved in embedding high dimensional data in lower dimensional spaces is to be found in an area of mathematics known as the geometry of graphs (Matoušek 2002). This field yields a general result for all finite graphs mapping into a host space with a guaranteed small distortion. Bourgain (1985) (as cited in Matoušek (2002) and Linial et al. (1995)) has shown that every n-point metric can be embedded in an $O(\log n)$ dimensions Euclidean space with $O(\log n)$ distortion. Better embeddings have been demonstrated for particular families of graphs. It may be possible to apply these results to the error involved in dimension reduction of certain features in the dataset and to provide formal descriptions of the mapping of features into lower dimensional spaces.

11.3.2 Statistical Information

Summary statistics can be provided in textual form to convey basic information about the distribution of variables and allow a basic check of data quality. Though our general assumption is that the dataset under consideration is of high quality, (i.e. it does not contain observation errors, or missing observations), clearly the user should check that this is really the case for their dataset. To encourage this, and not assume that users will automatically do this as good practice, the means to make these checks should be available within the application. In the case of outliers, they may or may not be due to errors as they may be the tail of a distribution. Summary statistics may be calculated for each variable individually and for each class, or group (if there are any identified). Using the sample mean and standard deviation are methods commonly used for giving an indication of the location and spread of the data. Table 11.1 is an example of this process for the first two variables of the calldata set. The complete table of summary statistics for multivariate data is unwieldy, since there are several columns for each variable. However, visually scanning them (or plots of the values if there are too many to scan), and creating summary statistics of the summary statistics themselves, can help deal with the large number of attributes. For example, in examining the calldata dataset, one can quickly establish whether there are any negative values, (in this case 'no'), then what the minimum values look like (all zero's), then the maxima (mostly very low, a few high), then the sum of attribute values (mostly low, only a few high). The plot of the sum is useful and shows that there are very few (out of the 255 destinations) that have many calls made to them. Thus the dataset is shown to be very sparse, containing mostly zeros. Note, this information is *not* available from the scatterplot of the whole dataset, though it would be indicated by a colourmap.

In addition to these summary data, others can be given, such as maximum, minimum, mean and

Destination 1				Destination 2			
	mean	stdev	range		mean	stdev	range
sample	0.5	2.55	0-23	sample	0.6	1.61	0-10

Table 11.1: Example of summary statistics for two destinations in the calldata dataset.

standard deviation of differences between distances calculated between entities in the original dataset and the distances in the transformed dataset. Correlation between variables in the original dataset can be examined, though the correlation coefficient is a measure only of *linear* correlation. Variables may be correlated in very different ways, consider the relation $y = x^2$ for x varying from -1 to 1, this has zero covariance (and correlation coefficient), but x and y are not independent. Non-linear correlation needs to be examined by visually examining plots of all pairs of attributes. Dot and box plots provide summary statistics in a visual form Unwin (2000) gives a description of how they complement textual summary statistics, though are not widely used.

Unwin (2001) has criticized some visual displays of data for not including statistical elements and calls for the *statistification* of visualizations, pointing out that results from big datasets are statistics themselves, that should be statistically justified. Without the user going through an exploratory process, even a simple one as described in the previous paragraph, it is hard to see how this type of information can be conveyed, unless the application is able to carry out such a procedure (or a similar procedure) itself. A related issue is the impact of standardization of variables. For instance, if one has an understanding of the application of PCA, one may know that, without standardization by the range or mean and standard deviation, the entities will line up upon the dimension which has the largest values. Thus, in the case of the calldata, the two destinations that have larger values than the rest will dominate. This is an example of something that the inexpert user cannot be expected to know. One possibility for dealing with this problem is to use generic datasets with different standardization methods to illustrate the behaviour of the visualization algorithm with respect to standardization. On the other hand, the application may be able to make a check and warn the user of the impact of not carrying out standardization. This example indicates the two general ideas for dealing with errors:

- Use generic datasets to illustrate to the user the behaviour of particular methods - what they are good for and what they conceal.
- Include a mode of operation where the user is alerted to such things as the impact of standardization and the concealing of important features in the current view.

11.3.3 Correspondence Analysis and Self-Organizing Maps

In relation to accuracy, for correspondence analysis and self-organizing maps the problem is compounded by the fact that these two methods result in combined spaces of the row-entities and column-

entities. The validity of viewing these spaces concurrently and guidelines as to drawing inferences between entities of row and column types, are issues that are left out of this discussion. Here we consider this, if one map is an abstraction and the distances between entities approximate, then what of two superimposed maps? Such maps are useful for browsing large collections and the attractiveness of being provided with named *features* (the attribute names) in the space is undeniable, but the accuracy issue remains. Again, at the least, the user should be made aware of this, be shown just how much of an abstraction these indicators are. Perhaps these particular forms (correspondence analysis and self-organizing maps) should be presented as an animation sequence of views (to convey the idea that there is no one *correct* view) or the designers should show, in some way, the implication of the juxtaposition and its limits. A simple colouring of all entities possessing a particular variable, for instance, or an additional map of the quantities of that variable possessed by each entity. This colouring would also highlight the intuition that these methods are more suited to a sparse dataset where individual entity dimensionality, in Atkin's sense of the number of attributes that they have non-zero values for, is low, though the overall dataset is high-dimensional (Atkin 1981).

11.3.4 Error Animation

An animated map can be described as one in which at least one map element changes in time. Animations can be changes in the symbolism, as in the flashing of a lightening symbol, or the variable. These changes of symbolism are described as *endochronic*, for animated symbols, and *synchronic*, for temporal variation of a variable (Shepherd 1995). These two animation types provide two ways of showing error using animation. Fisher (1994) has produced a number of animations for visualizing uncertainty in soil maps and reliability in classified remotely sensed images and computer-generated dot maps and elevation models using synchronic animation[1]. One of Fisher's animations relates to spatial information where location is imprecisely known, but general locations are known. In this situation points are selected randomly within the general location and an animation produced by fading out the points and replacing them with new randomly chosen ones. The speed of randomization can be altered by the user. In this way the uncertainty is demonstrated by changing location constantly, whilst the background remains the same. Another example concerns soil classification, where an area is classified as a certain soil type, though other soil types exist within it. In the animation these 'inclusions', as they are termed, appear as animated squares of the colour representing the type included in the area. Another example relates to uncertainty in identifying land cover by remote sensing. The likelihood of a cover type at a pixel being the cover type on the ground is animated. Whilst these methods are all concerned with error or lack of precision of information in the original data, rather than due to the process of visualization, these methods could be applied to the visual depiction of

[1] Demos can be downloaded from http:\\www.geog.le.ac.uk\pff1\Research\Error_Animation\Error_Animation.html

other types of error.

11.3.5 Superimposition of Minimal Spanning Tree

An example of the superimposition of a minimal spanning tree on a scatterplot of data, where dimensions have been reduced by using PCA, is illustrated in figure 11.3. This plot uses a tool called *Spinne* (Bienfait and Gasteiger 1997). This allows the user to identify, for instance, points that appear close to each other, but are, in the higher dimensional space, not really close. It reveals whether there are distortions of this kind everywhere, or only in some regions. For instance, points 26 and 9, connected by a red line, are too close to each other; to recognize this, one must compare the length of the red line with the coloured scale and see whether it comes to the same colour on this scale. The length of the 26-9 line corresponds to the yellow part of the colour scale, indicating that the line is too short.

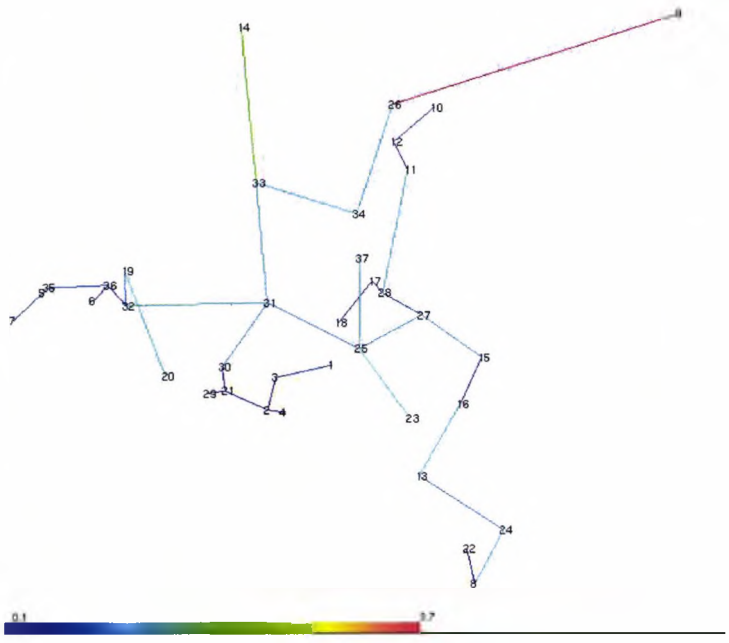


Figure 11.3: Superimposition of coloured minimum spanning tree on a dataset projected from 4 dimensions to 2 using PCA. Coloured scale shows actual Euclidean interdistances calculated in the original space, ranging from 0.1 to 3.7. Large interdistances are represented by red, short ones by a violet-dark blue line.

11.4 Different Types of Maps and Location Implication

The approximation resulting from an abstraction of high dimensional data to something that can be displayed on a 2D screen has cartographic equivalents. Just as adjacent points in a scatterplot

depicting PCA or SOM may or may not be as near as they are shown, traditional maps contain uncertainties, such as the fundamental transformation of locations from a sphere (or geoid[2]) to the plane and the omission of elevation on many planimetric maps (maps without contours). There also may be a number of equally valid representations of location depending upon the use of different metrics, categorization methods (of land use, for instance, an example of which is given in the section above) (Fisher 1994), measures of uncertainty (Ehlschlaeger et al. 1997), contour interpolation methods (Wood 1994) etc. Thus, whilst types of maps may be roughly divided into those mapping spatial and non-spatial data, the range of location implication is wide in both cases. Uncertainty in traditional maps is introduced also because of the quality of the data itself (which has a number of different aspects, see Goodchild et al. (1994)) and necessary procedures to map between levels of detail (such as those that involve interpolation). Visual representations have been used to visualize this uncertainty (Drecki 2002; Ehlschlaeger et al. 1997; Fisher et al. 2004; Fisher 1994; Lucieer and Kraak 2002; MacEachren 2002; Wood 1994) (including the animations described in the previous section).

The issue of accuracy of location highlights an important difference in types of 'maps'. In viewing topographic maps (maps with contours), many users regard location as an absolute quality and not especially approximate. They may not be aware that there are scaling and smoothing inaccuracies for such things as road and river widths and locations, and so on, and that there are problems created by projecting from a non-planar surface onto the plane. Perhaps we are so used to and reliant upon maps that many of us take their representations for granted. However, distortions and omissions do occur as illustrated. Nevertheless, location has a literal sense in these maps. Thematic maps have approximate location as do, for example, other maps such as subway maps. Mathematical transformations, such as PCA, create a map that may have a high degree of arbitrariness in the location. A continuum can be described, of the level of 'absoluteness' of location and relative location meaning. Topographic maps, typically, have a high level of location validity, scatterplots of high dimensional data have a low level. The user has thus to deal with a wide variation in level of accuracy and sense of absoluteness of relative location. Care must be taken that the user assumes the appropriate level of location validity.

11.5 Empirical Examination of Error: New Layout Method and Agent Profile Application

If the visual representations of greatly reduced dimensions that are in use are valid, then other domains can use them. As this work was being used to examine how signature exploration could be used to choose a metric for measuring the similarity of agent interest profiles (as introduced in Section 8.4), the idea arose of giving the information map to an agent to use to guide their interactions

[2]Geoid: the mathematical figure of the earth. Defined as: The equipotential surface of the Earth's gravity field which best fits, in a least squares sense, global mean sea level (National Geodetic Survey, http://www.ngs.noaa.gov/).

with other agents. In turn this led to the proposal that the agent carry their profile as the xy co-ordinates in this space and then exchange only xy co-ordinates with agents when they meet, thus keeping the details of their profile private. In order to avoid a third party calculating all the points in the reduced dimension space, an iterative method was proposed based upon the use of randomly chosen reference vectors[3]. This provides a useful means of comparing profiles without revealing their precise detail and without involvement of a third party. It also provides a form of PCA for layout which has the same time complexity as ordinary PCA, yet allows extra entities to be plotted without recalculating the whole set.

This technique and an empirical investigation are described in the next two sections. The security of the transformation (i.e. whether it is possible to obtain the original details of the profile from exchanging xy co-ordinates) is examined in the paper that describes this work, "Advancing Profile Use in Agent Societies" (Noy and Schroeder 2004). More detail will be found of the method and its motivation in this paper as well as in "Approximate Profile Utilization for Finding Like Minds and Personalization in Socio-Cognitive Grids" (Noy and Schroeder 2003).

11.5.1 Position as Profile

The pictures of information spaces as maps or terrains derived from multivariate data using self-organizing maps or PCA or metrics followed by multidimensional scaling, provide us with a compelling image of a profile or topic space to explore. Though this may be a misleading image, since the data are high dimensional and it is impossible to represent their similarities accurately in 2 or 3D space, nevertheless, as an approximation, such representations are often used in visualization. Suppose we assume the validity of the layout and propose that such a space can be used by software agents who want to find similar agents for collaboration and exchange of information. Thus it is proposed that the agent carries with them their xy (or xyz) co-ordinates in this space and uses them as their profile. When meeting a fellow agent they can ask for the agent's xy co-ordinates and compute the Euclidean distance, say, to calculate their similarity. This would be more efficient than carrying a potentially long profile vector and enable them to use their profile without revealing details or requiring encryption. To illustrate this approximate profile method, consider a small matrix of 7 agents with certain levels of interest (of 0 to 10) in 7 topics. Note that this data could also relate to the specification of tasks, products or information etc.

[3] In higher dimensional spaces there is an interesting observation (Hecht-Nelson 1994) that there exist in a high-dimensional space a much larger number of almost orthogonal vectors than orthogonal vectors, so that vectors having random directions might turn out to be close to orthogonal.

Agent1	9	3	4	6	5	5	5
Agent2	1	10	10	1	7	2	0
Agent3	4	1	6	8	0	5	7
Agent4	2	7	8	4	0	2	0
Agent5	3	6	4	7	1	10	6
Agent6	1	7	6	5	0	2	0
Agent7	8	1	7	1	2	5	9

Suppose Agent1 and Agent2 want to compare their profiles without exchanging them. Here we propose that they are given positions in the plot in 2D produced by reducing the dimensions of this matrix. The position can be derived in two ways, here described as *by base calculation* and *by calculation on-the-fly*.

By base calculation

The agents both have the calculations done at a base point and periodically return for updates. Here the error will be that of the layout itself and the agent would be able to have details of the mean error and variance supplied with its co-ordinates, so that it can take this into account. Figure 11.4 shows the layout after City distance and PCoA of the seven agents of randomly generated data from above. The City distance is first calculated between each pair of agents resulting in a 7x7 symmetric proximity matrix. A two-dimensional layout that approximately satisfies this proximity matrix is then found using PCoA, resulting in the set of xy co-ordinates plotted in Figure 11.4. Thus, if Agent1 meets Agent2 they can compare co-ordinates, ((-12.30, -5.20),(23.27,-8.44)), to calculate the Euclidean distance to give them the distance they are apart in this map.

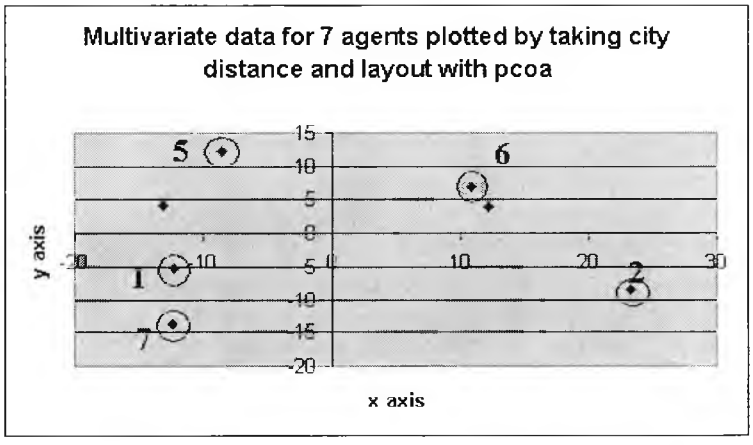


Figure 11.4: Illustration of base plot, the three reference agents (5,6 and 7) and the two of interest in this measurement (1 and 2) are circled.

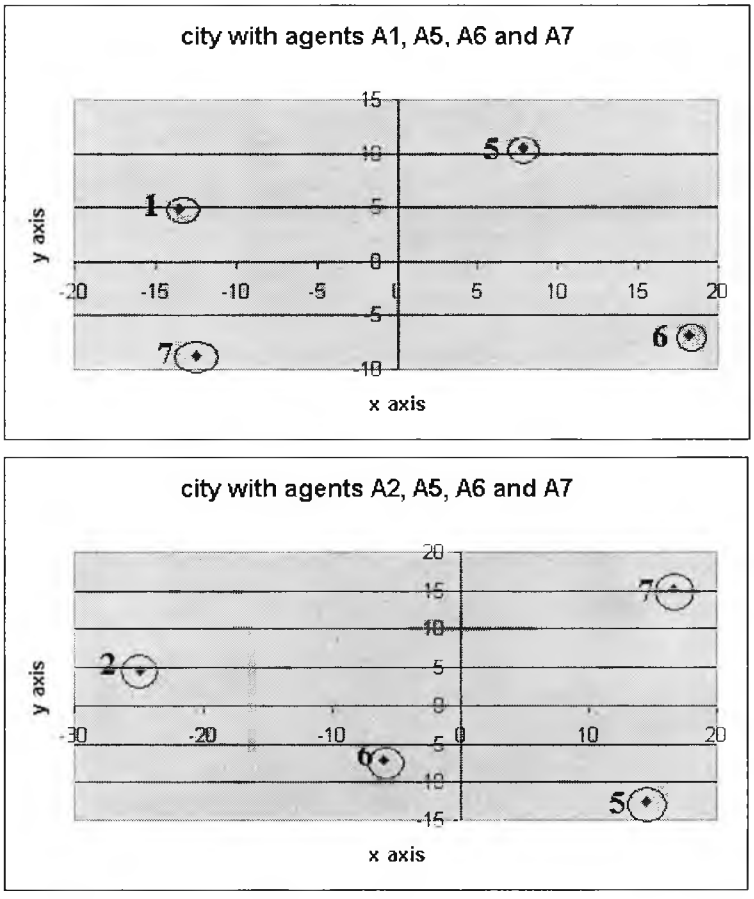


Figure 11.5: Illustration of plots calculated individually by agents 1 (top) and 2 (bottom) with respect to the three reference agents (5,6 and 7) as circled and numbered.

By calculation on the fly

Here the agent calculates its position with respect to a number of reference vectors (either dynamically or at an earlier point in time) and then compares with another agent's position calculated similarly. Using the seven agent random data again, the reference vectors are chosen to be agents 5,6 and 7. Three reference agents are the minimum since only two will create two possible arrangements when agents 1 and 2 overlay their positions. Agents 1 and 2 separately calculate their City distances to the three reference vectors and subsequently lay out these distances with PCoA as shown in Figure 11.5.

They now have xy co-ordinates, but in order to compare them they must be scaled (the Euclidean distance between 5 and 6 is used here), centered (here Agent 5 is placed at 0,0) and finally rotated to bring the agents 5,6 and 7 into position. Now the co-ordinates of the agent's position are in a form that they can use for comparisons. The results of the base calculation and on-the-fly calculation of the difference between agents 1 and 2 are given in the table below. (Since these are normalized with

respect to the distance between agents 5 and 6, a value of 1 would indicate that they were the same distance away from each other as agents 5 and 6 are)

original city dist	base dist	on-the-fly dist
1.64	1.77	1.57
exact	8%err	-4%err

This iterative version of the transformation has the same time complexity as the direct method, since, given n entities with d dimensions, PCA has $O(nd^2)$, whereas, in the case of 3 reference vectors, the calculation is done for 4 entities, iteratively n times, which requires a time n times PCA for $n = 4$ with dimensions, d, which is also $O(nd^2)$. Also the iterative version has transformed the process into one which can allow the addition of entities and the change in attributes of an existing entity, *without re-calculation of the whole set*, which would be necessary with direct PCA. The implication of this is also that the iterative version is less affected by missing or erroneous values, though these will still potentially affect the standardization procedure.

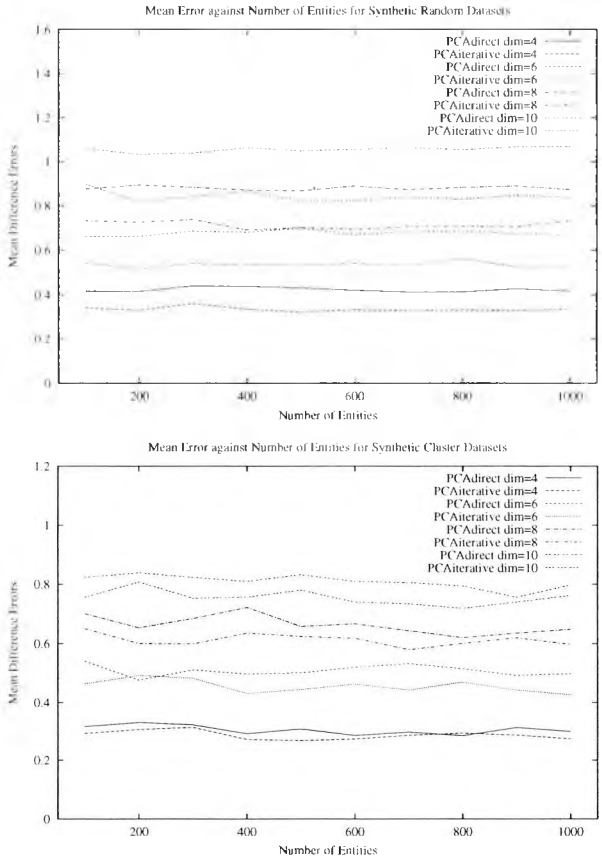


Figure 11.6: Mean Error against Number of Entities for Synthetic Random Datasets (top) and Cluster Datasets (bottom)

11.5.2 Empirical Accuracy Examination

The aim of this series of experiments was to examine the accuracy of using a position-as-profile version of the profile, both using base calculation and on-the-fly calculation. The Euclidean metric was used to give a measure of distance between the original profiles. The corresponding distances were calculated for the transformed profiles, for base calculation and on-the-fly. The distance errors obtained were then calculated and averaged to give an average difference in distance error, d_{avg}, where:

$$d_{ij} = (\sum_{k=1}^{p}(x_{ik} - x_{jk})^2)^{1/2}$$

and

$$d_{avg} = 1/p^2 \sum_{i,j=1}^{p} d_{ij}$$

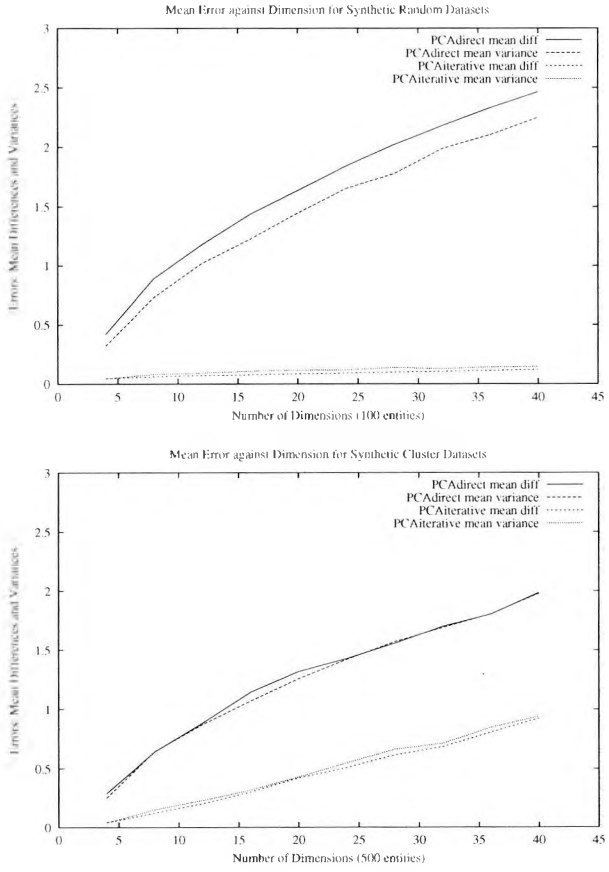


Figure 11.7: Mean Error against Dimension for Synthetic Random Datasets (top) and Cluster Datasets (bottom)

Two different types of synthetic datasets were used - random and clustered. For each of these datasets, the number of entities, n, and the number of dimensions, d, were varied and for each d and n, 30 runs were executed (i.e. 30 datasets of that type and size created), d_{avg} for each taken and the

30 values averaged. In each case, for convenience, 3 reference vectors were chosen from within the datasets and used to align the transformed sub matrices as described in section 11.5.1. n was varied between 100 and 1000. d between 4 and 40. Standardization with individual ranges of variable was used (Gordon 1999). PCA was used for the matrix transformations.

The results given in Figures 11.6 and 11.7 show that the error is largely independent of the number of entities. The error increases with increasing dimensions, but iterative PCA outperforms direct PCA. In the clustered datasets, the variance increases significantly for increasing dimension.

11.6 Summary and Conclusions

This chapter has examined a major obstacle encountered from the outset in this work: accuracy. Accuracy is an issue in dimension reduction situations where a high level of abstraction is involved, but also in other aspects of the visualization process.

Accuracy, error and abstraction are all words that relate to the loss of information resulting from the process of taking data from the real world and finding a form of visual representation for it. The loss of information is inevitable because of the discretization of measurement and the change of form from data to representation. Sources of error are: in observation from real-world to data; in transformation from data to data in preparation for visual depiction; in visual representation, from data to display; in interpretation, from display to human.

In estimating accuracy in dimension reduction applications, overall measures can be given, but it is useful to consider how may points in an area around a point should not be there (false alarms) and how many should be there, but are not (false dismissals). Simple methods to highlight these entities can be employed, though currently, in most visual depictions of high dimensional data, no indication of such error is given.

The analysis of this problem in mathematics is known as the geometry of graphs, but applicable results to formalize the mapping of features to visual depiction and thus provide precise expressions of the error, or distortion, are not generally available. Summary statistics, such as providing mean, standard deviation, maximum value and minimum value for attributes, should be presented to the user within the visualization, as this information is not always revealed by a particular visualization method. Neither do visual depictions always show such basic information as whether the dataset is sparse or that the dimensionality of individual entities is low. Dot and box plots are useful because they show additional statistical information visually.

For correspondence analysis and self-organizing maps, the problem is compounded by the combination of spaces of the row entities and the column entities, both spaces often involving high levels of abstraction. Some means of indicating this to the user is desirable.

Error animation has been used in geovisualization to show data where location is imprecisely

known and where ground cover type is uncertain. Error in dimension reduction scatterplots has been demonstrated by superimposing a minimal spanning tree upon the scatterplot so that pairs of entities that are too far from, or near to, each other can be identified.

A study of how different kinds of maps are perceived is valuable, since users make assumptions about the absoluteness of location in examining either a geographic map or information map. An information map could be a SOM or a scatterplot of data after reducing dimensions; these are seen as maps in that direction has no absolute meaning in terms of level of an attribute possessed, but locality has meaning in relation to the location of other entities in the representation. The field of geovisualization has considerable experience in dealing with uncertainty of different types and some examples of how to convey this to the user in visual form. This experience should be drawn upon, but more techniques are also required to ensure that the user makes the correct assumptions about the meaning of location for information visualization.

Included in this chapter is an application for facilitating agent profile use, which suggested itself from the general accuracy considerations of visual depictions of high dimensional data. The use of a lightweight reduced dimension profile which also keeps the details of the profile private has been demonstrated. This differs from the standard use of metrics in two ways: by the use of the transformed profile in agent/agent (or other) interactions; by using a method of transforming the profile without third party involvement. Tests with random and clustered datasets show that the error involved in the transformation is not affected by increasing number of entities, but does increase sharply for increasing dimensions. The results for calculation *on-the-fly* (using a form of iterative PCA with respect to three random reference vectors) are slightly better than those for *base* calculation (direct PCA) (the mean error is lower, though the variance is increased). In this case, using PCA, the time complexity is not altered, which means that the iterative form of PCA could be used to avoid recalculating the whole set, and be useful where values are missing or erroneous, the entity that possesses this problematic data will be affected, but not the rest. Thus, this iterative form of PCA provides a convenient way of an agent finding their approximate position in an interest space, without third party involvement, but is also a new form of layout.

How does the accuracy question relate to the signature exploration process? If one cannot be sure about the relative location of particular entities in the display, then any conclusions one may make from particular patterns, caused by particular features in the dataset, are suspect. We know that, in general, any conclusions made from a dimension reduction plot should be treated as hypotheses, but does the user know that? Thus, we need some way of communicating this to the user, but also specific techniques for the user to be able to explore this aspect of the visual depiction. This is an aspect of visualizing data that is of importance where users unfamiliar with particular methods are involved.

In a wider sense, all the techniques of signature exploration can be considered to concern error, if error is considered also in its widest form; since examining the behaviour of visualization methods

will reveal distortions (mappings) and omissions as well as measurable losses of information. Though we remind ourselves that the power of abstraction for visualization lies in its ability to provide us with a means to reduce large quantities of data to something manageable.

Chapter 12

Framework for the Design of Visualization Systems for Increased User Comprehension

12.1 Introduction

The work in this chapter draws together all the work of the previous chapters to provide a guide to designers in examining their particular application for sources of difficulty of understanding and to suggest appropriate techniques to address those areas identified. The result is a framework for identifying problem areas and techniques to apply. The analysis in this thesis of problems presenting obstacles to comprehension (from Chapters 2,3,4 and 11) is summarized in the first part of the framework, and the identification of both existing and new techniques for addressing these problems (resulting from the whole of the thesis to this point) makes up the second part of the framework.

The main motivation of this work is that we have many new techniques proposed for information visualization, and more being proposed all the time, but there is less emphasis upon analyzing the ones we already have. Whilst new developments are innovative - for instance, allowing us to deal with greater amounts of data - there is a corresponding need to provide support for the use of these techniques and to improve their presentation and accessibility to the user. (These issues have been explored more fully in Chapter 5.) This situation reflects a more general one in computing - the pursuit of greater functionality at the expense of quality of the interface and of the user's experience in general.

This work assumes that a generic solution does not exist, so that it is inappropriate to specify a set of techniques for *all* visualization systems to address comprehension issues. The main justification

for this assumption is the circular nature of the problem that the added techniques in themselves require understanding on the part of the user, and, even if the added techniques are intuitive in use, presenting a large number of them in the interface creates another problem for the user. Clearly there would also be some issues not encountered, for instance if colour or 3D are not used. Another reason to assume that a generic solution does not exist is that it is likely that different sets of techniques are useful for different situations, that there is not an optimal set. There is a general view (Green 2000) that there is no perfect user interface, notation or representation and that designers must make trade-offs. Thus the framework described here provides a means for the designer to analyze their particular application by breaking down the visualization process into several stages and asking a number of questions about each of these stages, then recommending particular techniques to assist.

Overall, the aim of this framework is to motivate visualization designers to focus more on the issue of user comprehension and to contextualize the work of the community of visualization researchers to this end. The problem area is broad and the development of a solid theory, with empirical evidence, will take time. Nevertheless, it is intended that this broad sweep of the area demonstrates an approach that is immediately applicable. The next two sections describe the two parts of the framework: the aspects that need comprehension support and the techniques to aid comprehension. The following section applies the framework to the calldata scenario. A further detailed example of application of the framework (to the tool Attribute Explorer) is given in Appendix A. The framework is then applied to several other visualization systems and the results evaluated. The final section summarizes the work.

12.2 Finding Aspects That Need Comprehension Support

To identify which aspects are involved in a particular visual depiction, the visualization process is broken down into the stages of the visualization process identified in the previous chapter for sources of error in visualization (Figure 11.1). These stages are real world, raw data, data for layout, display and human. These are covered by the chapters on data, layout and morphologies (Chapters 2, 3 and 4) and the sources of error in Chapter 11. This part of the framework summarizes the conclusions from each of these chapters and so presents the results of the examination of the obstacles to comprehension given in Sections 2.7, 3.5 and 4.6, and summarized in Table 5.1 and Table 6.1 (in categories mathematical transformation and graphical representation).

Here a visual depiction could be a whole system, a particular visual representation type (such as a colourmap), or an investigation of a particular dataset with a particular visual representation. As in the discussion of accuracy in the previous chapter, the issue of the quality of the original data in terms of its validity and accuracy from observation, is included in this framework, though it is not the focus.

The stages used for the framework are:

- Real world:
  - Domain relevance: is domain information important to how the information is visualized or how clustering is carried out? e.g. offset translation may be of no importance.
  - Data collection impact: how does the data collection affect the visualization? Is the sample representative of the population?
  - Measurement error: what is the quality of the data, what errors of measurement exist? e.g. measurement tolerances.

- Raw data:
  - Multiple structures: many structures are derivable from the dataset, for instance if it is a log which supplies information about entities, their characteristics and interactions. It may be difficult for the user to derive and view these structures.
  - Choice of object: what constitutes an object may be variable, for instance an entity or an action could be the visualization object. Entities and attributes can be interchanged.
  - Data and attribute type: there are different types of data and some can be transformed into others e.g. numerical type can be transformed into categorical. If attributes (or entities) have no intrinsic ordering, their order is arbitrary, though metadata may be used to introduce an ordering (e.g. ordering the destinations according to size, in the calldata set).
  - Size: the number of entities or the number of attributes in a multivariate or distance matrix, the number of entries in a log, the number of entities and links in a graph, levels in a hierarchy etc. are all indications of the size of a dataset. In each case, where these structures are large, the user needs support in understanding the way that the data are presented.
  - Associated metadata[1]: data about the data in the dataset, metadata, is available concerning the entities, attributes etc., e.g. location of destinations in the calldata. Vast amounts of metadata are available, including resources on the Web, though linking of the data into visualization applications is currently difficult.

[1] Metadata as associated data, rather than an abstraction of data as described in Chapter 2.

  - Selection and standardization: displays are (obviously) very sensitive to the data that is selected to be viewed. Some methods usually assume the data is normalized, or there are different standardization methods available which affect the display.

- Data for layout:

  - User choice: some applications allow the user to choose how their data will be mapped, for instance, to the shapes in a glyph, or the colour scale for an attribute. Users do not have the expertise of designers and may need guidance.
  - Predictability: the lack of predictability of algorithms leads to different layouts for the same data.
  - Abstraction: mathematical transformation of high dimensional data results in high levels of abstraction. There are different ways of reducing dimensions. Users need to know that there are different possibilities and need support in choosing between different methods. They also need to appreciate that information loss has occurred.

- Display:

  - Unfamiliarity: the visual representation is novel and it is likely that the user has not used it before. The representation may also be complex, for instance, a spiral colour map.
  - Spatial meaning: ambiguity of the meaning of the spatial component, e.g. location accuracy is low in a SOM.
  - Hidden features: the representation conceals certain features in the data whilst revealing others, e.g. overplotting in scatterplots.
  - Multiple windows: the creation of new windows to provide focus+context and different views introduces the problem of how to deal with multiple windows.
  - Mapping complexity: the mapping of the data to elements in the display may be complex or unintuitive.
  - Ordering: equivalent representations may exist due to arbitrary ordering of attributes or isomorphism.

- Human:

  - Expertise: users will have varying levels of expertise e.g. with a representation or a mathematical technique.
  - Perception: how the user perceives elements in the display may be unexpected, e.g. colour scales are not perceived linearly, depth cues can help or hinder.

- Cognition: what the user understands from the visualization process. Potentially this aspect relies upon all the other in this list, but is used here so that particular difficulties may be flagged, e.g. whether the user is/isn't aware that there are multiple valid views of their data.

12.3 Techniques For Increasing the User's Comprehension

Having identified the problem areas in a particular application, the designer needs to identify possible techniques to assist the user in meeting the comprehension challenges presented by the application. This section provides a list of possible techniques based upon (i) the analysis of the literature presented in the earlier chapters of this thesis, (ii) the examination of signature exploration of the middle chapters and (iii) the conclusions from the chapter examining accuracy. The list also looks forward to the remaining chapter of this thesis, the conclusion, which provides further rationale for these techniques and more detail of suggested developments, for instance in providing accuracy visualization and proactivity of the interface.

Note that some of these techniques overlap, for instance *feature demonstration* and *illustrative datasets*, in the sense that they apply to the same problem. Also, some of the techniques are generally applicable. For instance, the *visual tracking*, *query and interaction*, *statistics* and *variety* techniques are all appropriate to use generally for increased comprehension in complex data visualization.

- Feature demonstration: provide facilities for direct demonstration of features of a representation that are characteristic, or that hide or distort. For example, visual or textual indications of accuracy and overplotting. (This is direct demonstration, as compared to the indirect demonstration of behaviour resulting from the use of illustrative datasets.)

- Feature fingerprinting: support user input of synthetic data within a real-world dataset to provide feature fingerprinting. This gives the user a means of *calibrating* the representation.

- Feedback: where there is dimension reduction, include a feedback interface allowing the user to compare their sense of similarity to that of the application, to elicit and capture the user's sense of similarity, and to modify the dimension reduction algorithm.

- Illustrative datasets: supply illustrative datasets for the user to see how specific features appear in the representation.

- Pedagogic mode: provide a pedagogic mode for inexperienced users to alert them to important aspects, for instance, accuracy or information hiding issues. For an example see Plaisant et al. (2003).

- Proactivity: include a level of proactivity on the part of the application, so that it carries out its own checks and analyses of the data and makes recommendations. In general this means that the application should make the user aware of a particular fact. (Though in this sense proactivity applies to a greater or lesser extent to most of the other techniques in this list.)

- Query and interaction: extend query facilities and interaction features in general. Ensure that necessary interaction for engagement is increased (or maintained) by added functionality, such as by using dynamic querying or direct querying of the data with the result highlighted in the depiction. Allow queries to be used as landmarks.

- Selection and standardization: provide the means to apply selection and standardization procedures, including the selection of the 'object' for the purposes of visualization.

- Statistics: provide statistical information about the data in the form of textual and visual formats.

- Variety: ensure that an application provides a great enough variety of visualization methods to show the different aspects of the data, for instance to compensate for an individual method's hiding of information, or to show that there are different, valid, views of the data.

- Visual tracking (bi-directionally linked brushing also allowing change of the data and display): support linked brushing between the data table and the representation - in both directions, unless abstraction is involved. By implication, this includes a data table view. Also enable data values or display values (colours, points, lines) to be changed whilst linked.

12.4 Applying the Framework to the Calldata Visualization

An example of applying the framework is shown in Table 12.1. The problem areas are identified by examining the stages of the visualization process (the first part of the framework) to see which are relevant for the particular application. Specific comments are noted down as to their relevance. The techniques of the second part of the framework are then examined to see which can assist. The example used here is the calldata set in the Space Explorer interface.

Table 12.1: Framework for identifying challenges to user comprehension and identifying solutions, applied to the calldata scenario.

Problem Aspect		Comment	Suggested Approach
Real World			
Domain relevance	√	Overall customer behaviour, independent of actual destinations, is of interest.	Feedback.
Data collection impact	–	Collection details unknown.	
Measurement error	–	Assumed 100% accuracy.	
Raw Data			
Multiple structures	√	Logs provide calling information about times and destinations of calls, customer types.	Selection and standardization. Variety.
Choice of object	√	The object could be the customer or the destination.	Selection and standardization.
Data type	√	Destinations are categories (though could be ordered by distance), entity ordering is arbitrary.	Feature demonstration (ordering).
Dimensionality	√	There are a large number of dimensions (100x276).	Feature demonstration (accuracy). Feature fingerprinting. Feedback. Illustrative datasets. Statistics. Variety.
Associated metadata	–	Data on locations, size of destinations etc., location, type, etc. of customer. This information is not used.	
Selection impact	√	Normalization affects dimension reduction algorithms. Different normalization methods are in use.	Illustrative datasets. Selection and standardization.
Data for Layout			
User making choice	√	The user chooses between different dimension reduction methods.	Illustrative datasets.
Predictability	√	Spring embedding is one of the algorithms in Space Explorer.	Proactivity.

continued on next page

Table 12.1: *continued*

Problem Aspect		Comment	Suggested Approach
Abstraction	√	Dimension reduction is used.	Feature demonstration (accuracy).
Display			
Unfamiliarity	√	The user may not be familiar with navigating in 3D environments.	Illustrative datasets. Pedagogic mode.
Spatial Meaning	√	There is low location accuracy in scatterplots following the dimension reduction.	Feature demonstration (location meaning, accuracy).
Hidden Features	√	Overplotting in scatterplots. Accuracy of layout from dimension reduction.	Feature demonstration (overplotting, accuracy).
Multiple windows	√	Creates multiple windows.	Brushing (unidirectional linking).
Mapping complexity	√	Mapping may be clear, though inexperienced users do not understand the dimension reduction process which creates a space in which direction has no intrinsic meaning.	Illustrative datasets. Query and interaction. Feature demonstration.
Ordering	–	The dimension reduction algorithms operate independently of the order of entities and attributes.	
Human			
Expertise	√	Users are unfamiliar with PCA.	Illustrative datasets. Feature demonstration. Selection and standardization.
Perception	√	3D representations involve depth perception issues.	Feature demonstration. Proactivity.
Cognition	√	The user needs to understand that there is no one correct view of the data etc.	Variety.

After examining the aspects that present comprehension difficulties and identifying relevant categories of techniques for each of these, as in the table above, the required techniques can be grouped

according to the headings in the framework. This produces the following result for the application of the framework to the calldata visualization:

- Feature demonstration. *Accuracy depiction* in the dimension-reduced displays. Include visual colouring of false alarms and dismissals, as well as textual information (such as, of eigenvalues). *Overplotting* could be shown with colouring.
- Feature fingerprinting. Useful in dimension reduction plots.
- Feedback. Users need to understand the sense of similarity of the algorithms. Also the application needs to capture the user's view of what is important.
- Illustrative datasets. Illustrate different kinds of features for users unfamiliar with the algorithms. Also provide the techniques for signature exploration in pedagogic mode.
- Pedagogic mode. This is important because the abstraction level is high and users are not always familiar with navigating in 3D. A signature exploration suite is suggested and a set of alerts, textual or visual. The alerts to cover the following: abstraction level, overplotting, non-predictable layout, standardization, 'no one correct view'.
- Proactivity. Alerts as in pedagogic mode.
- Query and interaction. Extend facilities to further engage the user and reduce the interaction response time (see Section 9.2.4).
- Selection and standardization. Dimension reduction algorithms are sensitive to standardization, so alerts may be appropriate. Selection from the original log would be beneficial, to allow the user to select different data tables and view them easily. Also to link to appropriate metadata. The interface should also allow customer/destination reversal.
- Visual data tracking. Using a data table and linking is useful, but linking back to the data table is not relevant here (except for the bar chart), because of dimension reduction. However, normal brushing is required and a data table view (so that actual values can be scanned).

12.5 Applying the Framework More Widely

The framework is a very general tool in the sense that it considers all possible comprehension issues and all possible techniques to assist comprehension. This means that it can be used to examine various types of application, from those consisting of essentially a single technique, to those providing various techniques and interfaces. Thus it can be used to consider the problems involved with a specific type of dataset and a single technique, such as in the example above, or a specific dataset with

several techniques. The process can be repeated to consider different dataset types, with the same single technique or set of techniques.

There are two types of situation in which the framework can be used: in preparation for the design of a visualization system for a particular purpose - to help the designer identify problems and solutions (for increasing user comprehension); to evaluate existing systems. The examples given in this thesis, in this section, the previous section and in Appendices A and B, show its use for the evaluation of existing systems. Appendix A illustrates the application of the framework to an existing visualization system ('Attribute Explorer') chosen for its particular brushing mechanism. Appendix B contains the reports of a business user in applying the framework to three visualization systems chosen to represent: a widely used system for general data analysis ('Excel'); a single interface employing a connectionist technique ('Daisy'); a research tool for exploratory data analysis ('Ggobi').

12.6 Evaluating the Framework

The framework summarizes the entire work of the thesis in identifying the comprehension difficulties faced by the user and suggesting techniques to assist with these problems (including both new and existing techniques). The main aim of the thesis was the exploration of the concept of signature exploration, so that the framework was the end point rather than the starting point of the work. Thus the thorough development and evaluation of the framework lies outside the scope of this work. The framework, in its current form, is designed for the experienced visualization researcher to use, in conjunction with the material presented in this thesis.

However, the application of the framework to the five example scenarios (three by a business user of visualization systems) demonstrates its general applicability (to visualization systems of different types) and indicates ways in which it could be developed to make it more widely available to visualization system users.

The main issue arising from the business user's application is that this user did not understand some of the issues, nor appreciate the scope and nature of some of the proposed techniques. Nevertheless, the user found the process useful for providing a structured way of looking at the visualization system and suggesting improvements to the design. Thus the main improvement will be gained by making the descriptions of the problem areas and techniques more detailed. Exploring the possibility of suggesting connections between specific problems and techniques is also important, though such connections are likely to emerge from the experience of applying the framework over time.

12.7 Summary

This chapter has presented the conclusions from analyzing the display of complex data for comprehension challenges, as well as from the investigations of the techniques for signature exploration. These conclusions are presented in the form of a framework for identifying problem areas and techniques to assist. The overall aim of the framework is to motivate visualization designers to focus more on the issue of user comprehension, contextualizing, enhancing and extending current work of visualization researchers.

The framework has two parts: aspects that need comprehension support; techniques to aid comprehension. The aspects are broken down into those that occur at various stages of the visualization process as follows: the real-world origin of the data; the raw data; the data in a form suitable for layout; the visual display; the human viewer. At each stage there are a number of issues that may arise:

- Real-world: relevant domain information; data collection impact; measurement error.
- Raw data: multiple structures; choice of object; data and attribute types; size; associated metadata; selection and standardization.
- Data for layout: user choice; predictability; abstraction.
- Display: unfamiliarity; spatial meaning; hidden features; multiple windows; mapping complexity; ordering.
- Human: expertise; perception; cognition.

The categories of techniques for increasing the user's comprehension are: visual data tracking, feature demonstration, feature fingerprinting, feedback, illustrative datasets, pedagogic mode, proactivity, query and interaction, selection and standardization, statistics, variety. Whilst some of these techniques are generally applicable, it is to be expected that the evaluation of a particular scenario against the aspects requiring comprehension support will lead to a particular subset of techniques being considered appropriate. To illustrate the use of the framework, the calldata scenario is analyzed and techniques recommended. Appendices A and B contain further examples of application of the framework. An evaluation of this experience indicates that the framework could be improved by making it more understandable to users and by identifying links between problem aspects and suggested approaches.

Chapter 13

Conclusion

The overall objective of this work was to enhance the comprehension of complex data visualizations, whether this complexity derived from the techniques used to display or preprocess the data, or the data itself in terms of numbers of entities, attributes or complexity of structure. A particular means to achieve this objective was proposed, namely, the application of a set of techniques based on a new concept, signature exploration, designed to concretely demonstrate the behaviour of the visualization method to the user. Thus, the focus of this thesis was a broad examination of the issues involved in understanding complex data visualizations and the application of signature exploration techniques to address these issues.

This chapter summarizes the results and conclusions of the work presented in individual chapters of this thesis, evaluates the work against the criteria for success introduced in Section 1.4, appraises the hypotheses of Section 1.3, presents general conclusions, identifies contributions and suggests future work.

13.1 Summary of Results, Contributions and Conclusions

This section restates results and conclusions from the conclusion and summary sections of individual chapters, but also extends the conclusions in light of the overall work. Contributions are identified, together with their relationship to previous work.

13.1.1 Analyzing the Background Literature to Identify Comprehension Challenges

An analysis of the aspects of data types and structures, data for layout, and display morphologies, presented in Chapters 2, 3 and 4, was carried out in these background chapters to identify elements

likely to present comprehension difficulties. The data types and structures discussion identified that facilities to support comprehension of these aspects fall into two groups: those needed to provide extra functionality for interacting with the data or display, and those to make users aware of important characteristics of the display. Examination of different layouts showed that the challenges arise from the different ways the same data can be represented, the special characteristics of layout methods, the high levels of abstraction sometimes involved, and the impact layout choice has upon interactivity. Key challenges identified by the examination of morphologies were: unfamiliarity for new types of representation and inexperienced users; the existence of equivalent representations; ambiguity in meaning of the spatial component and the use of multiple windows and other methods to provide focus and context.

Contribution: The identification of obstacles to comprehension from all aspects of the visualization process and the corresponding identification of elements in the literature to address these obstacles. Much is known about the design of static presentations, so that users will not be misled by incorrect or misleading representations, and to take account of perceptual issues such as how colour scales are perceived (Bertin 1983; Cleveland 1993; Tufte 1983, 1990; Ware 2000a). At the same time, to assist the user in exploring their data, many different methods of displaying data (such as pixel displays (Keim 2000) and parallel coordinate plots (Inselberg 1997)) and interaction forms (such as brushing (Cleveland and McGill 1984; MacDonald 1990) and dynamic querying (Shneiderman 1994)) have been developed. I have surveyed this work from the point of view of the user's understanding and have identified the issues and relevant techniques from the literature. This work is beneficial because it provides a new perspective upon the design of information visualization systems, a new topic, to enhance the user's understanding of visualization systems. This perspective is much needed in the current context of expanding types of data display, a wider range of users and increased combination and complexity of techniques.

13.1.2 The Rationale for the Work

The rationale for the importance and timeliness of investigating the greater provision of comprehension support for the user, begun in Chapter 1, was extended in Chapter 5, where it was shown that this issue underpins other current open questions for information visualization. Chapter 5 showed that the proposed signature exploration approach addresses many of the obstacles to comprehension identified in the earlier chapters.

Contribution: Identifying the reasons for the importance and timeliness of raising the issue of user comprehension of complex data visualization. Previously, researchers have indicated the scope for greater application of the results from cognitive science and human factors in general and

the need for the development of principles and analyses for information visualization, to develop a human-centred approach (Herman et al. 2000; MacEachren and Kraak 2001; Ware 2000a). I have provided a comprehensive rationale for the importance and timeliness of investigating the issue of user comprehension, based upon what information about the visualization method is a prerequisite for the user, which will provide designers with increased motivation to address this issue.

13.1.3 Signature Exploration

Chapters 6 to 10 defined and applied signature exploration. Chapter 6 examined the obstacles to comprehension further and identified three main categories: mathematical transformation, graphical representation and 'one view of many'. Signature exploration is a means of the user gaining insight into how features in the data map to patterns in the visual representation. It allows the user to use datasets that are known in some way to explore the behaviours, or signatures, of different visualization techniques. It is defined as the exploration of the behaviour of a visualization method by means of the visualization of specially constructed datasets, which contain, or are representative of, particular features of interest. Five techniques involving the production or provision of constructed data were presented: generic dataset provision; user-construction of data; querying; insertion of landmarks; elicitation and application of feedback data. This approach builds on the work of the last ten to twenty years for assisting the user's exploration of data, reframing this work by placing the focus upon the understanding of the visualization process itself. An existing tool, Space Explorer, was extended to implement the illustrations of the proposed techniques in the following four chapters.

Contribution: The proposal of a new concept, signature exploration, and a set of techniques for its application, to aid the user's understanding of the behaviour of visualization methods. (Contributions of the individual approaches are given below.) Many researchers have proposed techniques for assisting users to interact with data depictions, for example, brushing (Cleveland and McGill 1984; MacDonald 1990), and for focus and context control (Furnas 1981; Lamm et al. 1996; Rao and Card 1994). Shneiderman (1996) has proposed the visualization information seeking mantra as a guide to the overall design of a system. I have proposed a new concept that focusses on the illustration of the behaviour of the visualization process to the user. This is useful because attention upon the process itself, the transformation from data to display, the representation, makes explicit the requirement that the user understand the process and seeks concrete ways of ensuring this. The application of signature exploration uses existing interaction techniques, thus extending their use, but also results in new ones.

13.1.4 Generic Dataset Provision

In Chapter 7, the literature was examined for assistance with choice of generic datasets. General categories were identified: null modes, clusters within noise, specific features and inter-entity features. A method of classifying clustering algorithms according to feature admissibility was adapted to visualization methods. Two scenarios were considered: examining a single visualization method using several datasets, and comparing different visualization methods using a single dataset. Two pieces of work were carried out: a feasibility test, which explored a single visualization method, and the integration of generic datasets within the user interface.

The feasibility test was a web-based test which examined a set of generic datasets displayed with a particular dimension reduction method, to see whether this would help the user to understand the meaning of the pattern obtained for the calldata. Some participants in the feasibility test considered their understanding to have been increased, though they could not explain the calldata pattern. Participants expressed the desire for more interaction and the ability to alter the data to see what would happen in the display. The test suggested a new technique, which I have called feature fingerprinting, to place a small reference dataset in a real-world dataset under consideration, to provide orientation to the user within the larger set.

The second examination illustrated the generic dataset provision within a newly developed interface. A menu gives the user a choice of datasets, which they can then view directly as data tables and stacked bar charts, then, after choosing a dimension reduction algorithm, as 2D or 3D scatterplots. This interface was used in the task of choosing a metric for an agent application.

The specification of generic datasets was found to be a difficult task for a number of reasons, particularly the large number of possibilities to choose from and the difficulty of specifying and quantifying features. Feature admissibility was found to be useful for classifying visualization methods according to which features in a dataset are shown in the visual depiction. This will be helpful in the related task of specifying benchmark datasets for evaluation of visualization methods. An additional problem is how to motivate users to use generic datasets, since this is not part of their usual approach.

Contribution: Identification of features for generic datasets and the demonstration of use of generic datasets. Designers sometimes provide example datasets and tutorials with applications, see for example Spence and Tweedie (1998), for Attribute Explorer, Rundensteiner et al. (2002), for the Xmdv tool. I have developed generic dataset provision which extends this element of illustration to provide, within the application, datasets containing example features to show how these features are presented by the visualization process. This provision will help users better understand the strengths and limitations of different representations.

Contribution: Application of feature admissibility for visualization methods. Researchers have classified visualization systems in various ways, according to different data types and processes (see e.g. Card et al. 1999; Chi 2000; Shneiderman 1996). Fisher and Van Ness (1971) and Van Ness (1973) have proposed feature admissibility for classifying clustering procedures. I have applied feature admissibility to classify visualization systems based on their ability to show specific features, which adds to the range of classification systems a means to show the strengths and limitations of visualization systems. This will assist in evaluating visualization methods and in establishing benchmark datasets for this purpose.

Contribution: Feature fingerprinting. Researchers have added reference data, or fingerprints, to high dimensional datasets which are subsequently visualized after dimension reduction (Meuzelaar et al. 1982). Also, the technique of brushing has been developed to highlight a group of entities in a display to show, by linking the data in displays, how these entities appear in another type of display (Cleveland and McGill 1984; MacDonald 1990). I have combined these two elements in a new technique, feature fingerprinting, which enables individual, or patterns of entities illustrating specific features, to be added interactively into a visual depiction. This enables users to orientate themselves in unfamiliar representations, or where the abstraction level is high.

13.1.5 User-construction of Data

A number of ways for the user to construct their own data, to test the behaviour of the visualization method, were outlined: direct entry or change within a data table; interest feature specification; generating data via a visual representation; using synthetic data generators. Three implementations illustrated these methods (excluding the last one which is illustrated in papers (Noy and Schroeder 2003, 2004)). The generation of data from interest feature specification was shown to assist metric choice. Also demonstrated was the linking in both directions between data table and display to allow data and point changes. I have described this as *visual data tracking*. This was useful to give immediate feedback to the user of the effect of changes either in the display or the data table.

The ability to create datasets containing varying amounts of a specific (user specified) feature allows the user to examine and choose between visual representations of features of interest to them. However, in this work, the interest feature specification was done manually; it would be preferable for the interface to be extended so that this could be accomplished more easily. Again, (as with generic data provision), the ability to put such constructed data into a real-world dataset, for orientation, appeared to be desirable.

Contribution: Visual data tracking. Brushing and linking between windows has been used extensively in information visualization applications to assist in the exploration of data displays (Cleveland

and McGill 1984; MacDonald 1990). I have extended this concept to include the linking between windows showing different representations (including the data table) to allow change of position of data point or value. This is a benefit because users can see the effect of changes in one representation, or changes of data values, and the effect this has, which will improve their understanding of the behaviour of the representations.

Contribution: Demonstration of user-construction of data for assisting metric choice. Distance measures are employed for layout in displays (see e.g. Webb 1999), but also to provide a similarity measure in many applications, such as in comparing document or entity specifications. A number of authors indicate the difficulty of choosing an appropriate metric (Gordon 1999; Webb 1999). I have illustrated the user construction of data for assisting metric choice, where the user creates data containing features of interest and examines the behaviour of different metrics visually. This process of examining the behaviour of measures on a general level helps the user determine what features are of importance to them and which metric is best for such features.

13.1.6 Querying and the Insertion of Landmarks

Three established means of querying were described: use of conventional query language, dynamic querying and visual querying. For signature exploration, these established techniques are used, but in order to understand the behaviour of the visualization method. The closely related technique of insertion of landmarks includes the highlighting of query results within a display to provide orientation for the user, but also includes the insertion of synthetic data to fulfil the same function. Where the inclusion of a specific pattern of data (with respect to a particular entity) is introduced, I have described the process as feature fingerprinting. This was shown to provide orientation within the calldata set.

The work highlighted the desirability of general hypothesis support elements in the interface, since these significantly reduce interaction response time and therefore encourage exploration, though such facilities are difficult to provide.

Contribution: The fixing of landmark entities in a display. Information visualization applications use the highlighting of entities in a display using brushing (Cleveland and McGill 1984; MacDonald 1990) or to display the result of a query (Shneiderman 1994). I extend this highlighting of entities to provide ongoing orientation within the display, by fixing the highlighting and describing such highlighted entities as landmarks. This concept of landmark is also expanded to include the insertion of synthetic data within a display. Providing the facility to place landmarks in complex data display helps orientate the user. Feature fingerprinting - the insertion of a synthetic feature within a display, for orientation, described in Section 13.1.4 - is also based upon this concept.

Contribution: The insertion of synthetic data into a display. I have found that the addition of synthetic data to a dataset under consideration is valuable to orientate the user in the graphic. This resulted from a combination of the concept of landmarking and provision of generic datasets. How it relates to existing work and contributes is described in fixing of landmark entities (previous paragraph) and in feature fingerprinting (Section 13.1.4).

13.1.7 Elicitation and Application of Feedback Data

In examining elicitation and application of feedback, the feedback process was considered to have three aspects: compare, capture and modify. The user compares their sense of similarity with the application's, the application captures the user's sense of similarity, the application modifies its behaviour to reflect the user's sense of similarity. Examination of related literature illustrated an issue concerning the use of the Euclidean measure for time series data, which sometimes produces unintuitive results, despite being widely used. The concept of 'global distortions' and modelling the user's sensitivity to these, is one way to apply the user's sense of similarity. The capture of the user's sense of similarity can be approached by relevance feedback or by asking the user to arrange entities on a screen so that their perceived 'distance' or dissimilarity between entities can be captured. Two illustrations of the latter method were given, one of which was implemented within the Space Explorer interface. In the interface, the distances of a subset of a dataset are arranged by the user, then the distances computed and weights for the variables calculated using multiple least squares linear regression.

The effectiveness of the modification needs further investigation, particularly using non-linear methods, though it is likely that this capture will remain approximate, since it is unreasonable to expect the user to specify precise dissimilarities between entities. Better results may be obtained with a ranking system, which provides a more reasonable capture of the user's view, or by modelling the user's sensitivity to distortions.

Contribution: Capture and application of feedback. Feedback from the user has been used in relation to queries, for instance finding similar time series patterns in a database (Keogh et al. 2000). I have extended this work to apply it generally to visualization systems, not only those involving querying. The user's sense of similarity is captured and the layout algorithm for the interface is modified. The advantage is threefold: the user (and the application) is assisted in identifying what aspects about the data are important to them in making comparisons; the application can alter its layout to reflect the user's preferences, where possible; the user can compare their sense of similarity with the application's and where these greatly diverge (even with modification), can reassess the available methods and the choice of data representing the entities.

13.1.8 Obstacle: Accuracy of Depiction

Chapter 11 discussed a major issue in the work, accuracy of depiction in the visual display. The related concepts accuracy, error and abstraction were examined. The different stages of the visualization process at which error could arise were identified, thus indicating the different sources of error. Accuracy estimators and depiction methods from the literature were described, though these are rarely used in practice. The specific problem of different types of maps and location implication was highlighted. This examination of accuracy led to a proposal of a new agent application for lightweight and private profile use, and a new iterative form of Principal Components Analysis (PCA), which allows the position of new entities to be calculated without recalculating the whole set. Overall, the investigation of accuracy issues suggested the need for visual (and other) forms of illustration of error to be generally employed.

Contribution: Highlighting the issue of different types of maps and location implication. Information visualization maps are becoming popular, for instance to provide overviews of document collections (Kohonen et al. 2000). In geovisualization, the inevitability of misleading the viewer in a map is well known (Monmonier 1991a). I have raised the issue of how users make assumptions about different kinds of maps and recommend making the location implication explicit. This will allow more accurate use of information maps.

Contribution: New software agent application for profile use. Profiles are used in many applications, for instance to match documents with queries, requests with specifications. In the software agent field there is a need to use profiles for agents to compare themselves (or their tasks) with others, while at the same time a desire to keep their profiles private. For instance, a matchmaking framework which includes demand and supply profiles (Veit et al. 2001), a system for finding common interests between agents (Foner 1997). These systems require a trusted third party to keep the agents' information private. I have devised a new iterative method of applying Principal Components Analysis (PCA), which allows agents to compare their positions in an interest space without revealing the details of their profile and without involving a third party. This will make it easier for the use of profiles in agent systems to be expanded.

Contribution: New variation of PCA for layout. PCA is often used to provide layouts for the display of high dimensional data (see e.g. Gordon 1999). I have devised an iterative form of PCA (as described in the previous paragraph) which allows new entities to be added without recalculating the whole set. This method is comparable in terms of accuracy and time complexity to the normal form of PCA. This method will be useful in a dynamic situation where entities are being added to a dataset to provide an animation.

13.1.9 Framework

Chapter 12 proposed a framework based upon the work of the earlier chapters, to assess comprehension challenges and suggest appropriate techniques for a particular visualization scenario. The framework assists by identifying problem areas and suggesting techniques that are applicable. Some techniques are simple and straightforward to implement, thus providing readily applicable solutions. Providing a checklist of problem areas is a convenient way of guiding designers and will also prompt them to consider their own ideas in this area. The framework should be developed to examine a number of visualization scenarios. This may enable it to be changed to make more specific recommendations.

Contribution: Specification of a framework for designing visualization systems for greater comprehension. A number of frameworks have been presented in the literature which assess the applicability of visualization systems (Chi 2000; Shneiderman 1996). I have designed a framework for identifying areas presenting comprehension difficulty and techniques to assist. The use of this framework will increase awareness of comprehension issues amongst designers, providing a structured approach that will result in more understandable and accessible systems.

13.2 Scope and Scalability

The work in this thesis has necessarily concerned a subset of possible visualization scenarios, both in terms of the type and structure of the data, and in terms of the kinds of visualization methods used (layout methods, dimension reduction, morphologies etc.). Two important issues are now examined: scope, i.e. how broadly can the work be applied; and scalability of techniques. Scalability concerns the size of the dataset in terms of the number of values in the dataset, the number of attributes and the number of entities. The complexity of the data also comes under the discussion of scope. Since the analysis of comprehension difficulties and possible techniques relates to existing (or proposed) visualization systems, there is an underlying measure of scalability of the visualization system under consideration. Thus the size of dataset that can be considered is dependent, in general, upon this and needs to be examined separately from it. Section 1.2 introduced the discussion of what is meant by complexity and pointed out that the problem of complexity relates to a very wide range of datasets from the relatively small to the massive. Section 3.3 examined the term scalability in the context of visualization systems and noted the limit brought about by the resolution of displays (currently around a million items). Also noted were other aspects of scalability relating to interaction time and practicality of navigation.

This section examines the results, contributions and conclusions from the previous section to

consider these issues.

- Background analysis. This analysis covered a wide variety of data types and structures, layout methods, dimension reduction methods, and display forms. Thus it was broad in scope, but could be made more detailed by considering other visualization forms, and by covering the ground in greater detail.

- Signature Exploration.

  - General concept. Techniques have been examined with a specific scenario requiring dimension reduction and using particular dimension reduction methods. This can be applied to other dimension reduction methods, but the question is whether the techniques can be applied to other visualization methods and types/structures of data? The concept itself - seeking to illustrate the behaviour of the visualization method to the user - is one which is generally applicable to different visualization systems with different data and structures. This follows since examining what can or cannot be seen in a visual depiction is an approach that is valid to apply to all visual depictions, irrespective of type or size of dataset.

  - Generic dataset provision. In general this technique is scalable and broad in scope since it is always appropriate to use illustrative datasets with any visualization system. Specific aspects:

    * Datasets that show particular features. There will always be features that can be shown in a particular visualization method, otherwise such a visualization method would be of no use. Thus, there will be features that can be shown, but not all features will be appropriate to show.

    * Feature admissibility. For the reason identified in the previous point, the classification of visualization systems as feature admissible for a particular feature is valid for all systems. However, there will be different sets of features applicable in different situations, for instance the features identified in this work for data tables will not all be relevant for hierarchical data.

    * Feature fingerprinting. Developed in this work for orientating the user in the dimension reduction scenario, the usefulness of this technique for broader application is unknown. However, it is likely that the addition of known items or patterns to a dataset would be generally applicable, though the choice of relationship to existing items would change. Thus, in the dimension reduction example, similarity of items is of interest, so that features based upon different types of similarity were used. For hierarchical data it would be more relevant to use similarity of branches. The na-

ture of the feature would also be determined by the size of the dataset represented: for instance, if the dimension reduction example used here had many more entities, the features proposed would not be observable and so would themselves need to be scaled. Note, however, that the general validity of the results for dimension reduction situations has not been established, due to the problem of the disturbance created by the additional data.

- User construction of data. It is valid to provide facilities for the user to construct and visualize data for any system, since user construction of data is essentially the creation of synthetic data. The particular means of constructing the data will vary according to the size and type of data required. For instance, the examples here have used direct entry into a data table, as well as statistical specification of clusters. Specific aspects:
  - Visual data tracking. There are limits to the applicability of this: backward linkage of layouts resulting from dimension reduction algorithms are not possible, since the mapping is one-to-many; it requires the moving of data points (or values) so that it can only be used where individual points are discernible, unless aggregation is employed.
  - For assisting metric choice. Here the user does not need to examine behaviour with large datasets.
- Querying. Querying methods are broadly applicable and scalable to the extent that the underlying visualization systems are.
- Landmark insertion. The comments under feature fingerprinting, above, also relate to landmark insertion, which is a subset of feature fingerprinting. Also, landmark insertion will only be appropriate where individual entities are discernible, otherwise aggregation is needed. Otherwise landmark insertion is generally applicable.
- Feedback data. The examples of elicitation and use of feedback data given in this thesis use similarity data elicited from the user to modify a dimension reduction algorithm. Thus, this particular application is restricted to that scenario. In addition, the number of dimensions that can be handled is restricted, since the number of examples for training is restricted. In general, results are considered reliable only when the number of training examples is at least 10 times the number of dimensions.

- Accuracy. The thesis contributions are general in nature on this issue, so apply broadly. The dimensionality of dataset used in the software agent application for profile use affects the error generated by the approximation used. This was established through the experiments given in Section 11.5.2. Thus this technique is scalable for the number of entities, but not the number of

Criterion	Chapter
1. *Defining a concept for applying constructed data, signature exploration*	6
2. *Specifying a set of techniques for the application of constructed data*	6 to 10
3. *Identifying problem areas and obstacles*	2 to 5, 11
4. *Reframing existing techniques*	6 and throughout.
5. *Implementing examples of the different techniques*	7 to 10
6. *Developing a framework for the design of visualization systems for increased comprehension*	12
7. *Specifying a set of techniques for aiding comprehension of visual depictions*	12

Table 13.1: Meeting the criteria set in Chapter 1. The chapters in which the criteria are satisfied are shown. Refer to previous section for details.

dimensions. These comments apply equally to the use of the iterative form of PCA as a layout method.

- Framework. Since the framework is for assessing visualization systems (existing or proposed), scalability is not relevant. From the point of view of scope, the framework was designed to have a broad scope as it is based upon the overall analysis of the visualization process and available techniques. The five examples given here of its use demonstrate its general applicability. However, the scope of the framework would be increased through its development by further use as described in the future work section.

13.3 Evaluation of Criteria for Success and Hypotheses

The details in the previous section show that all the criteria for success, introduced in the introduction, have been met. The relevant chapters are indicated in Table 13.1. There is scope to expand the work under all criteria, but especially for criteria 4 to 7. This is discussed further in the future work section below.

The criteria for success were derived from four hypotheses, introduced in Section 1.3. The first of these hypotheses was:

> *The application of the concept, signature exploration, aids the comprehension of visualizations of complex data.*

This hypothesis was tested by applying the concept, signature exploration, to visualization problems relating to dimension reduction and metric choice. Evidence to suppot this hypothesis lies in the increased comprehension that resulted from the exploration, the understanding of the implications of the overall shape of the clustering of high dimensional data, and the choice of metrics to match user's preferences. Comprehension was increased by means of a number of different techniques as follows:

Technique	How comprehension increased
Generic data.	Understanding how features in the data appear.
User construction.	Enables metric choice and understanding of how features in the data are shown.
Visual data tracking.	Bi-directional linked brushing allows concrete examination as values are changed.
Query and landmark.	Enable the user to orientate themselves in the dataset and test hypotheses.
Feature fingerprinting.	Provides orientation in the calldata dataset.

However, it was not possible to show this to be generally true in a rigorous and systematic manner without altering the scope of the investigations (as explained in the first chapter). It is usual in some areas of research to undertake and present work without the formal expression of hypotheses (see e.g. Knight (2000)), because not all research problems can be examined purely in terms of supporting or rejecting hypotheses with data. The hypotheses were used here to guide the work, but not to impose a purely quantitative examination of the area, which would have restricted the scope. Perhaps some readers will be more comfortable with rephrasing these ideas as objectives and dispensing with hypotheses.

The second hypothesis was:

The application of signature exploration aids the choice of display of complex data.

Support for this hypothesis is in the case of generic data provision and user-construction of data, which have aided the choice of dimension reduction method. Also the application of signature exploration aids the choice of display of complex data *indirectly*, since it leads to systems which better support user comprehension.

Hypotheses 3 and 4 are as follows:

The application of signature exploration will lead to the development or specification, or both, of a suite of techniques for aiding comprehension.

The application of signature exploration can form the basis of a framework for the design of visualization systems for increased comprehension.

Evidence to support these hypotheses is as follows: a framework for the design of visualization systems for increased comprehension has been created; a set of techniques have been partly specified and partly developed. A number of techniques have been specified, but not developed, in that their implementation has not been included in this work, for instance: techniques for user construction of data; greater facilities for querying and the visualization of error; alerts and the general proactivity of

the interface. Techniques that have been implemented, have been developed in one sense, yet afford considerable further development. However, the reason that the techniques have only been 'partly developed' and 'partly specified', a conscious decision on the part of the author, is in order to maintain the broad scope of the work. Thus, to fully specify and develop a single technique, would have taken the whole time available. Likewise, this examination of signature exploration forms the *basis* of a framework, but the framework has potential for expansion in scope and detail. The framework is a summing-up of the work of the thesis, to be used in the context (and understanding by the user) of the work of the thesis. The framework is the ending point, not the starting point of the work of the thesis. It requires considerable further work to develop and evaluate a framework for general use.

13.4 Future Work

This section identifies future work for each area undertaken in this thesis and suggests new directions.

13.4.1 Generic Dataset Provision

This work should be continued to examine the behaviour of different visualization methods, different datasets and an expansion of the concept of feature-admissibility for evaluating visualization methods.

Repeat Experiments with Different Visualization Methods

The area of dimension reduction is the most difficult one to support the user's understanding in. This is due to the often high level of abstraction that is carried out on the data. Thus the experiments presented in this thesis have tackled the most difficult area in which to support user comprehension. It would be useful to carry this series of experiments out for a wide variety of visualization methods, to cover, for instance, colourmaps, parallel-coordinate plots and other connectivity-based tools (such as *Daisy*), as well as those for displaying tree structures. The results will be easier to generalize in these cases, since the representations are not so affected by mathematical transformations and abstractions (approximations and errors in the raw data notwithstanding).

Other Possible Datasets

A number of other types of generic dataset have been suggested by this work, either as a result of the experiments, or because they are used to assess clustering or classification algorithms.

- overlap: this can be considered in different ways - as two clusters overlapping as in the *Iris* dataset, or as in the situation where there is actually a partition, but this may only be apparent in certain visualizations or because the data has been constructed in this way. An example is

one where there are two 'C'-shaped clusters which are hooked together, but do not intersect (Corsini et al. 2002). Another example is a coiled band (in 3D). Such examples are used to examine the ability for clustering algorithms to recognize such partitions (Is the first example considered to be 2 clusters or 1?) or structures (Is the second example shown as a round cluster after reducing to 2 dimensions, or is the band characteristic recognized?).

- noise + clusters or other features: embedding the generic data within noise - does a visualization reveal a 'hidden' cluster or feature?
- scaling and phase shifting relations of different types: and for different types of data (such as trees).
- behaviour similarity (i.e. irrespective of variables): only applicable to datasets where all variables are of the same type, such as the *destinations* in the call data, or values in time series data. Different *behaviour profiles* can be examined. This can be generated by deciding on a set of behaviour shapes, then randomly assigning the variables. For instance a behaviour shape for the call data where a customer makes calls to half the destinations and to the destinations that are called they make the same number of calls.
- entities distributed evenly throughout the space: in a sense this is also a type of null model, since the pattern is uniform.
- accuracy: datasets to highlight accuracy problems.

Use of Admissibility

The use of admissibility criteria to systematically examine and record the behaviour of visualization methods should be expanded. Thus properties include behaviour under transformation, visual appearance of features (such as how the feature is observed, whether by pop-out) as well as types of data (such as *sparse*) and feature (such as *outlier*). Examining a wide range of these properties for a wide range of visualization methods will enable the value of the use of feature admissibility for classification of visualization methods to be developed.

13.4.2 User-construction of Data

Data Creation within the Interface

It is desirable to make data creation possible within the interface, as well as to include general data generation facilities, so that the user can experiment more easily and quickly. In particular, including functionality to transpose and repeat patterns, to add new objects and variables, to enter equations to generate the data, and to 'sketch' patterns (in a line plot) from which data is derived.

Visual Data Tracking

Visual data tracking is a promising technique since it extends the brushing between windows with the ability to *move* data points (or change table values). Testing within different visualization contexts (different data type and visualization method) to measure its effectiveness and task relevance is needed. Reverse tracking cannot be used when dimension reduction is involved, since the transformation is one-to-many (many points in the higher dimensional space map to one in the lower dimensional space - this is discussed further in Noy and Schroeder (2004)). It may be possible to identify a probable range of originating points, but it is not clear how to do this.

13.4.3 Query and the Insertion of Landmarks

Extend Query Facilities

Query facilities should be extended to allow greater freedom of hypothesis generation and testing. However, this is difficult because, on the one hand it requires much programming effort, and on the other hand there are fundamental difficulties in the framing of queries as discussed in the thesis.

Feature Fingerprinting

Feature fingerprinting has been demonstrated in this thesis, but requires development of interfaces within the visualization application for the user to interactively place patterns of constructed data in the display of a larger dataset. It would be useful if a constraint or range could be inserted into the dataset under consideration, so that this could then be seen in the visual depiction. Indeed, if axes (in the originating data) themselves could be mapped, this would provide useful reference structure. However, this is only possible in direct one-to-one mappings that do not involve dimension reduction, since otherwise the data would create too great a distortion in the layout. The addition of points without great distortion can only be accomplished when localized to small areas. In particular, the mapping of the bounds of a (high dimensional) space produces the greatest distortion.

13.4.4 Elicitation and Application of Feedback Data

Capturing the User's View

More types of interface for capturing the user's view of the data, using ranking and other elicitation methods, are needed, to see which have the best result. This is especially so that the natural approximation in this exercise (by the user) can be reflected.

Modifying the Visualization Behaviour

The modification methods used here (least squares linear regression, neural nets and genetic algorithms) need to be investigated for a wide range of datasets (varying number of variables, entities and data types) and different user similarity capture methods (rank and similarity). Different ways of modifying the behaviour of the visualization method need to be explored, particularly to address the problem of statistical validity for small training datasets, and the non-linearity of the mappings.

Feedback for Direct Methods

Whilst the complexity of the mapping in dimension reduction examples has been the inspiration for this use of feedback from the user, the method can be applied to direct methods. For example, the user can rate, classify or arrange the similarity of a number of familiar entities and these values could be used to weight the attributes before displaying the data using a direct method such as a bar chart or colourmap in the same way as in the dimension reduction context. The process is now a means of carrying out a kind of *user normalization* or *selection* of the data, by capturing what is of importance to the user.

13.4.5 Accuracy of Depiction

Identifying Problems

These accuracy problems need to be further documented so that designers and users are fully aware of the issues. Each existing visualization method should be analyzed to identify accuracy problems.

Accuracy Depiction Methods

This work has underlined the importance of promoting the wider use of accuracy depiction methods and of developing more methods of depicting accuracy, both visual and textual. This concept can be extended to include an examination of information that the user ought to know and whether (and how) designers should seek ways to communicate this information to the user, so that the application changes from being essentially passive, to having a more proactive role.

13.4.6 Framework

The framework as it is presented in this thesis represents a summary of the findings of the work of the thesis. In its two halves it covers the work identifying the problem areas for visualization of complex data, and the techniques (existing and ones proposed here) available to address the problems. The main aim of the thesis was not to develop a framework, but to explore the concept of signature exploration. Thus, the current framework is a starting point for the development of a more practical

version, appropriate for general application by designers and users. The main issues that need to be addressed are:

- applicability to visualization without dimension reduction: the dimension reduction context was the main one for this work. The framework needs to be used for different types of visualization methods, to see how the categories of comprehension problem and recommendations may be refined.
- understandability: the framework description assumes a research-level understanding of clustering and visualization issues and techniques.
- linking problem to suggested technique: the structure of the framework can be developed so that suggested techniques are liked to problem areas. These connections can be identified as the framework is applied more widely.
- evaluation: once these three issues above are addressed, the framework should be evaluated with designers and users.

13.4.7 Related Developments

Benchmark Datasets

The establishment of benchmark datasets for evaluating visualization systems would help designers to compare techniques. The publishing of pictures (2 or 3D) of visualizations of such datasets would enable researchers to compare systems without having to possess implementations of all such systems. Some of the types of datasets presented in the generic dataset section of this work are suitable for this purpose. The admissibility system could be used to establish a useful set.

Automation and Proactive Systems

Recent years have seen the growth of datamining and information visualization techniques, and the corresponding need to explore ways of guiding the user in the data exploration process. However, visualization systems remain essentially passive, waiting for the user to load their dataset, then waiting for the user to choose the layout, alter the settings and so on. This thesis has suggested that there may be things the interface **ought** to tell the user about, such as hidden features or errors in the data. However, if the application was truly proactive, it would carry out its own analysis of the data, as a parallel process to the user's selection and manipulation of displays. It would find all things it *could* know about the dataset and possible displays and have this information ready for the user, or make suggestions to the user. In addition, as distributed processing become easier to harness in the

everyday office environment, the application need not be limited by the computing power of a single processor to carry out its investigations.

Thus the work of this thesis points to the need and possibility of a new kind of intelligent visualization, *proactive visualization*, where proactive components communicate knowledge about the data or the display characteristics to the user - conveying to the user, graphically or textually, information about the data that is not apparent in the current display space. This approach turns passive visualization components into proactive ones, which use meta-knowledge about the data, analysis techniques and visualization capabilities to guide the user.

Proactivity could be enhanced by the elicitation from the user of a profile indicating preferences. The application may act proactively by presenting results of a range of background calculations or by indicating limitations of a display. A simple example of the latter is the colouring of an item in a scatterplot to indicate that it represents not one, but many coincident items. A more subtle form is the use of an alphaslider whose upper range (say) is greyed out indicating that movement in this direction will not alter the visual representation (Spence 2001). Background computations, which may or may not have been requested by the user, include summary statistics and the automatic selection of graphical representations or interesting submatrices of multivariate data (e.g. Shneiderman (2002)). It is considered difficult, if not impossible, in general, to algorithmically specify what constitutes structure in data (Ward et al. 1994) and choice of graphic is task dependent. Hence datamining and visualization processes will never be fully automatic, but some aspects can be automated and accuracy of analysis technique and strengths and weaknesses of visualization techniques can be captured. With this meta-knowledge, proactive visualization would seek to provide a better informed and guided approach to visual data exploration.

13.5 General Discussion of Conclusions and Contributions

The three main contributions of this work are: the definition and application of a new concept, signature exploration; the proposal of a framework and set of techniques; the specification of two new interaction mechanisms for visualization systems - visual data tracking and feature fingerprinting. There are also two contributions on a more general level: the motivation and conceptual framework for increasing comprehension support in visualization systems; the motivation of the need for visual and textual indicators of the characteristics, particularly accuracy of layout, of visualization methods. In addition, two contributions in related topics, not directly concerned with the objective of this work, have been made: a new software agent application for profile use; a new iterative method for layout with PCA.

New Topic of Increasing Comprehension The broad motivation of this work was to find a useful approach to enhance the user's comprehension of complex visualizations. From a concept based

upon an intuitive assessment of the situation, a number of techniques were proposed and researched, first examining any related literature and then developing example software. It appears that the focus upon enhancing user comprehension is, in one sense, a new topic, since one does not currently find workshops or conference symposia devoted to this, nor many conference or journal papers. However, much attention has been given by designers over some decades to techniques that support the user's exploration of the data. The concept presented in this thesis is a reframing or contextualization of this work. These techniques, at the same time as revealing more about the data, increase the user's comprehension of the visual depictions. The difference is in the focus of the approach. Here, the characteristics of the visual depiction are explicitly investigated, rather than the implicit exploration which accompanies the visualization of datasets. Thus, the first potential for original contribution of this work is the focus of attention upon the exploration of characteristics of visual depictions in themselves.

This work has provided motivation for the greater importance of providing increased support for comprehension of visualization methods at this time, by discussing the wider range of users, continuing innovation in visualization techniques and the development of complex composite tools. In addition, an examination of current open questions for information visualization has shown that other areas will benefit from improvements in user comprehension.

Signature Exploration and its Application The examination of signature exploration and the set of techniques for applying it, have led to a rich area of possibilities for increasing comprehension (rich in ideas and techniques - promising and original). Generic dataset provision, though requiring more work to determine a useful basic set, has shown promise for illustrating the behaviour of visualization methods. It also suggests itself for benchmarking and general evaluation, though these are slightly different purposes. The work did not set out to evaluate and classify visualization methods. Interfaces that allow user-construction and interaction with the data itself suggest another level of interaction. The ability to construct data additions within real-world datasets under consideration is very promising, since it provides the user with a unique method of orientating themselves in a space which otherwise has a characteristic arbitrariness of direction in space in dimension reduction situations. This may also provide a quick means of orientating oneself, in more general terms, in any unfamiliar or complex visual representation. Data construction interfaces also enable the user to explore their sense of 'what matters' to them in their data and particularly provides them with an environment to explore the application of metric choice. The query and landmark examination highlighted that there are still difficulties in the conceptually simple mechanism of query in a visual interface, and that any extra provision in this area is useful. The highlighting of the results of a query or the addition of synthetic data as landmarks, suggested that calibration was an apt description for the role that these play and that this is an important requirement for the user, apart from the recog-

nition of particular pattern meanings. The elicitation and application of feedback data examination showed that much could be gained from a variety of these techniques and that they had importance, both in terms of the user's understanding of the behaviour of the visualization method and in terms of making the user, and the application, or both, aware of what was of importance to the user. When the user is more aware of what factors in the data are important or unimportant, they are better able to choose appropriate views of their data.

Two new techniques have been specified:

1. Visual data tracking: two-way brushing between data table and visual depiction, or two or more visual depictions, which also allows values and positions to be altered. This enables the user to change the graphic and see how values in the data table alter as a result, and vice versa.

2. Feature fingerprinting: synthetic additions to real-world datasets to provide the user with calibration of the visual depiction.

Framework and Techniques This work introduces a framework to guide the assessment of challenges to comprehension and determine suitable techniques to apply.The challenges to comprehension have been identified by examining each aspect of the visualization process, including data types, transformations and visualization morphologies. This work could be extended to further analyze strengths and weaknesses of particular methods and suggestions for addressing weaknesses. The suggested techniques have arisen from analysis of existing systems and interaction mechanisms, from the application of signature exploration and from examination of the accuracy issue. Whilst there is scope for extending the framework, the work nevertheless contributes a directly applicable approach.

Accuracy Indicators This project has revealed the need for visual or textual indications to alert the users to special characteristics of the representation, such as the existence of hidden data due to overplotting, or the approximate nature of location. In general terms, we have brought great complexity to the screen - the user needs to appreciate the limitations.

Visualization Paradox Visual representations of complex data can be useful and lead to insight, but such visualizations themselves are complex due to the transformations necessary to put the data on the screen. Thus there is a paradox - on the one hand visualization can reveal things, but on the other hand, the visualization process is often complex and hard to understand. There are no short cuts in good science: nothing can replace the experience of the informed user; complexity cannot be replaced by simplicity, without incurring a loss. However, complex techniques are being used by a wider range of users and nothing will halt this. Also, expert users benefit from developments which ease their analysis, particularly in relation to hypothesis generation and testing. This work has

examined what characteristics of the visualization method can, or should, be shown to all users; it has identified methods (new and existing) such as exploratory interfaces, alerts, and intrinsic or intuitive features, which can be made available as required. We would like all visualization to be natural and clear, but, in dealing with complexity of size and structure, this is not possible. We would like all users to be experienced and ready to engage in exhaustive search - again this is unrealistic. On the other hand we would like to be able to automatically generate appropriate visual representations of data - again, this is rarely possible. Thus, we take a pragmatic approach, we seek to know what can be known about the data and the visual representation, and convey this information, or make it available, to the user.

Another Level of Complexity? All interaction mechanisms designed to increase the power and clarity of visualization systems, themselves present a further layer of complexity that the user must deal with. Whether or not this complexity presents a significant difficulty for the user, and, if so, whether this difficulty is justified by the cost, varies according to the particular mechanism. We are presented with another paradox - where interaction mechanisms create more complexity, yet are designed to alleviate complexity. This relates to established techniques such as provision for overview and zoom, or dynamic querying, as well as the techniques described in this work. For instance, the insertion of landmark entities can be easily understood and the benefit easily obtained by the general user. On the other hand, the implication of adding a pattern of entities related by general behaviour to an existing entity, followed by dimension reduction, requires greater understanding, yet has the potential to provide much needed orientation.

Closing Statement The examination of the concept of signature exploration has led to a rich area of techniques and ideas. The work presented in this thesis provides impetus for further work to increase the comprehension of complex data visualizations and moves us towards a comprehensive methodology for the design of systems for enhancing comprehension. Visualization system designers are providing us with increasing numbers of methods of dealing with complex data. Support for comprehension of complex data visualizations will lead to greater knowledge discovery from data and enable us to fully exploit the potential of new and existing displays.

Appendix A

Applying the Framework to Attribute Explorer

An example of applying the framework is shown in Table B.3. The example used here is the visualization of a dataset of information about cars with the tool Attribute Explorer. Attribute Explorer provides the user with a set of interactive linked attribute histograms (Spence and Tweedie 1998). It can be downloaded from http://www.alphaworks.ibm.com/tech/visualexplorer. The tool is intended to complement IBM's DB2 Intelligent Miner for Data or to run as a standalone tool. The latest version includes parallel coordinate plotting which is not covered in this analysis. The car dataset is a sample dataset supplied with the tool.

Attribute Explorer provides three views of the data: the primary view shows the field names, types and statistical information; the charts view shows the histograms for fields selected in the primary view; the details view shows a tabular view of the source data (each row representing a data point). An example screenshot of the linked histograms is given in Figure A.1. Moving the mouse over the segments in the charts highlights the data point under the mouse in all charts and the details view. Beneath each chart is a slider with which the user can apply constraints by selecting and deselecting segments. Each data point is colour-coded to indicate the number of field criteria it fails. The colour of the background and data points can be changed, as well as the segment sizes and the number of charts shown on one screen. By interacting with the selection sliders the effect on data points for each of the attributes can be rapidly assessed.

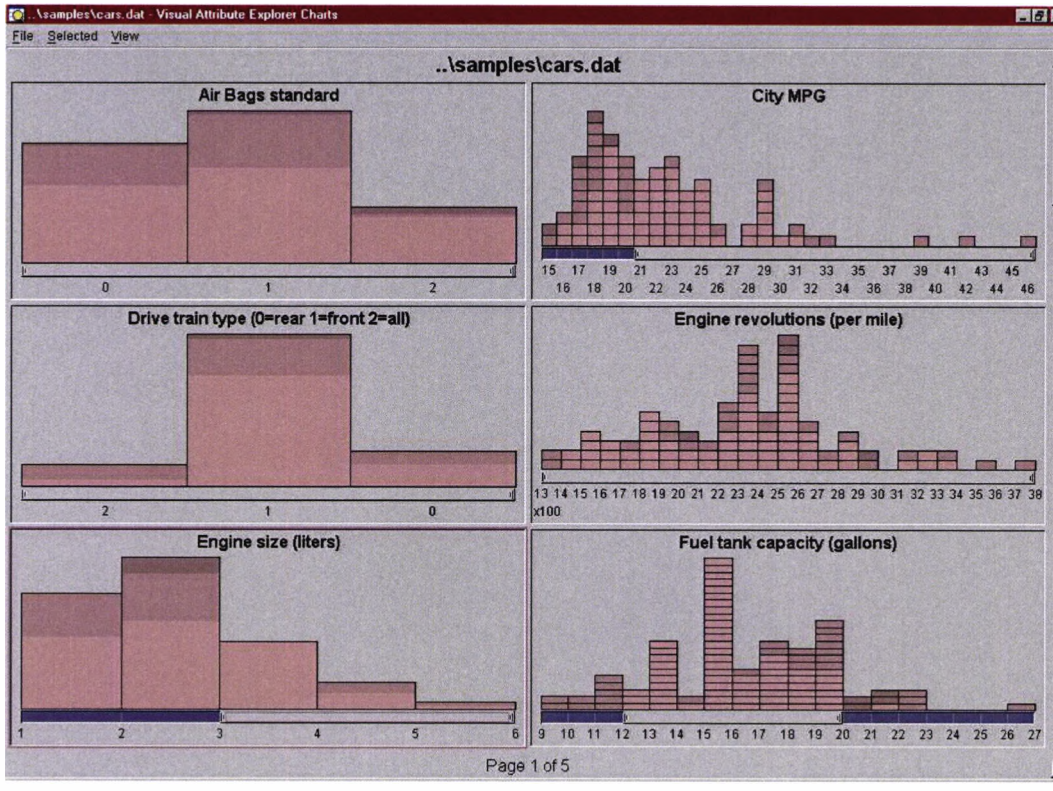


Figure A.1: A screenshot of the Attribute Explorer linked histograms. Data is the sample dataset of car data provided with the application.

Table A.1: Framework for identifying challenges to user comprehension and identifying solutions, applied to the Attribute Explorer.

Problem Aspect		Comment	Suggested Approach
Real World			
Domain relevance	√	The application is to find the most acceptable object or, perhaps a small number of candidate objects worthy of more detailed consideration, given a collection of objects, each described by values associated with a set of attributes (Spence 2001, p. 77). This means that general exploratory data analysis is not the goal. The application for choosing a car implies a wide range of users.	Pedagogic mode (ordinary users). Feature fingerprinting (adding perfect car). Query and interaction (landmarks).
Data collection impact	–	Collection details unknown, the dataset is a sample set with the application. In a real application, information about the dataset should be supplied, e.g. how comprehensive it is.	

continued on next page

Table A.1: *continued*

Problem Aspect		Comment	Suggested Approach
Measurement error	√	Accuracy of measurements unknown. Null values are listed in the summary data view, but there is no way to show this in the display. Here, the existence of null values does not invalidate the visual depiction, but it does invalidate the query (if they are involved - and one doesn't know whether they are or not, until the detailed record is examined).	Feature demonstration (null values, accuracy of original measurements).
Raw Data			
Multiple structures	√	Different segmentation values result in different views. Ordering of categorical data arbitrary. User can select different fields, segmentation values, colours. Highlighted position of a car in a bar on the chart varies according to the constraint applied. However, the user is aware of some of these from the interactivity of the interface.	Pedagogic mode. Proactivity.
Choice of object	√	Each of the charts is a frequency plot of a single attribute, so there is no ambiguity of object. Other views of the data are desirable, e.g. showing scatterplots of pairs of attributes and the application of clustering to complement this, also to provide overview, but, within the application this issue is not relevant.	
Data type	√	Categorical data has arbitrary ordering. Data types which have few categories dominate the display, contributing less information for the space occupied.	
Dimensionality	√	There are 93 records with 26 fields, but dimension reduction is not used. Depiction involves minimal loss of information, but the large number of attributes and instances means it is difficult to get an overview.	Proactivity. Feature fingerprinting (landmark addition). Interaction with the data table is not possible. Could highlight, change and add values. Especially adding values - to put in one's perfect car, an old car etc. - for calibration of the visual display. Feedback (elicit preferences to limit display).
Associated metadata	–	For the purpose of 'finding the most acceptable such object', metadata would be useful, for instance, about the company.	Provide linked metadata.
Selection impact	√	Size of segments, ordering of categorical data (and of record fields) and selection of colours have an impact.	Illustrative datasets. Selection and standardization (allow changing of arbitrary placements).

continued on next page

Table A.1: *continued*

Problem Aspect		Comment	Suggested Approach
Data for Layout			
User making choice	√	Users can choose colours, segment size, colour of mouseover highlight, best, worst.	Illustrative datasets. Pedagogic mode.
Predictability	√	Layouts are predictable in the strict sense. However, the display is predictable in the general sense only for the same segment sizes, colouring etc. How does the application calculate the segments to start with? Also, the behaviour of bars for fields, other than the one where the slider is being moved, shows position of entities moves. Thus, highlighting (by mouseover) shows where that particular entity is in all the charts. However, when that entity fails the selection it moves to become part of the top shaded part of the bar. This is necessary to provide the characteristic effect of the display, but is slightly confusing, since sometimes the entity has a precise place in the display, sometimes it doesn't.	Proactivity (alerts for better segment size, colouring). Illustrative datasets (showing difference of display).
Abstraction	√	Dimension reduction is not employed. Frequency plots result in information loss for larger segment sizes.	Feature demonstration (animation of the results of segment size alteration etc.).
Display			
Unfamiliarity	√	It is likely that the new user has not used this interface before.	Illustrative datasets. Pedagogic mode.
Spatial Meaning	√	Ordering of the charts is fixed, ordering of categories, size of bars (segment size) is not. Also the size of the bars in charts of categorical data with few categories is greater and tends to dominate the view. So reordering is possible and some sizing is approximate and disproportionate. In general, though, spatial meaning is clear.	Feature demonstration.
Hidden Features	√	Aside from the discussion about what could be discovered about the dataset with other display methods, (though this is the effective 'hiding' of features), the segment size hides peaks and valleys. Also the user will not find things if they do not follow a sustained interaction sequence.	Feature demonstration. For overview, maybe the starting view should show all the fields. This forces the user to start with an overview, but also prompts interaction (in the case of a large number of fields). Obviously there is a limit to how many fields can be shown, though.

continued on next page

Table A.1: *continued*

Problem Aspect		Comment	Suggested Approach
Multiple windows	√	Creation of new windows is an issue - the user needs to be able to see both detail and chart displays at once.	Brushing (bidirectional linking - allow selection in details view).
Mapping complexity	–	Mapping is quite clear and reasonably intuitive.	
Ordering	√	Equivalent representations exist due to arbitrary ordering of charts and categorical variables (chart view) and records (details view).	Feature demonstration. Selection and standardization (selection in details view). Pedagogic mode.
Human			
Expertise	√	Users will have varying levels of expertise, re: what cannot be shown; frequency plots; how to use colour; interaction ability (and propensity) in general; ability to construct their own path of experimentation to discover the features of the display etc.	Illustrative datasets. Feature demonstration. Pedagogic mode. Proactivity. Problem for many fields: gradation of colour is too difficult to distinguish, i.e. too many shades. This can be helped by using two colours (for best and worst) rather than a monochrome, but I only discovered this later. it wasn't immediately obvious. Perhaps it would be better if the application suggested it immediately, when there were quite a few fields. Also, colour changing window is difficult to understand how to use and to know what kind of effect to expect.
Perception	–	Colour scales not perceived linearly, but this is not of great importance here since the required granularity is low.	
Cognition	√	Though the mapping appears simple, there are subtleties to discover. As it is, this discovery relies upon the interaction pathway of the user.	Pedagogic mode. Proactivity.

Main Results by Technique Category

- Feature demonstration. Consider ways of showing the effect of null values, the accuracy of the original measurements, the results of segment size alteration (perhaps by animation), ordering, colour changes, what is readily perceived (e.g. an outlier, clusters).
- Illustrative datasets. Does the designer know that certain datasets illustrate particular features

in the display? If so, use them, i.e. seek to embed the designer's experience of use in the application.

- Pedagogic mode. Provide a novice's mode to ensure that all features are covered and understood. This can include such things as animating the effect of segment change and colour change possibilities, or highlighting arbitrary placements.
- Proactivity. Consider alerts, particularly in pedagogic mode, for example, to alert the user to the undesirability of monochrome scales for many fields. Provide 'optimal' segment sizes, colours and layouts, for particular datasets. Begin the sequence with an overview showing all charts.
- Query and interaction. Increase interaction (reduce interaction response time) by allowing removal of charts by mouse click. Increase visual comparison scope by allowing repositioning of charts. Allow landmark insertion.
- Selection and standardization. Allow change of arbitrary placements. Allow selection in the details view. Make provision for the linking of metadata.
- Visual data tracking. Add linking from the data table to the charts, so that the user can find a specific car model. Add provision for making additions to the data table to allow placement of, for instance, one's perfect car.

Appendix B

Framework Application by Business User

This appendix details the experience of a business user evaluating three visualization applications with the framework. This user is a marketing manager with a great deal of experience analyzing business data. This analysis frequently involves visualization of different kinds, principally with Excel. The user is also familiar with more recent developments such as self-organizing maps and tree maps.

The user was given a detailed briefing which covered the description and examples of the application of the framework (as described in Chapter 12 and Appendix A). The possible techniques (both existing ones and new ones proposed in this thesis) were also explained.

The user evaluated three visualization systems, 'Excel', 'Daisy' and 'Ggobi'. These were chosen to represent the differing types of application available, both in terms of the type of visualization techniques used, as well as the contrast between an application using a single technique and one containing a number of techniques. Excel was chosen as a widely available application that is used by a very wide range of users. Daisy was chosen as an example of an individual new type of interface. Ggobi was chosen as an example, well-known in the research community, of a visualization system containing a variety of methods for analyzing high-dimensional data . In each case the evaluation was done for a specific dataset; for Excel a dataset of sales prospects was used, for the other two applications the example datasets supplied with the applications were used.

The user was asked to complete a questionnaire afterwards, to examine

their experience of applying the framework. The questions related to relevance, ease of use, general assessment, and whether or not they were already familiar with the systems evaluated.

B.1 Excel

Excel is an almost ubiquitous tool used by business managers and analysts to visualize complex data sets. This strong position in the market has attracted a large number of specialist 'add in' functionality to compliment an already extensive in-built functionality. Over and above this the application allows users to program custom solutions with Visual Basic.

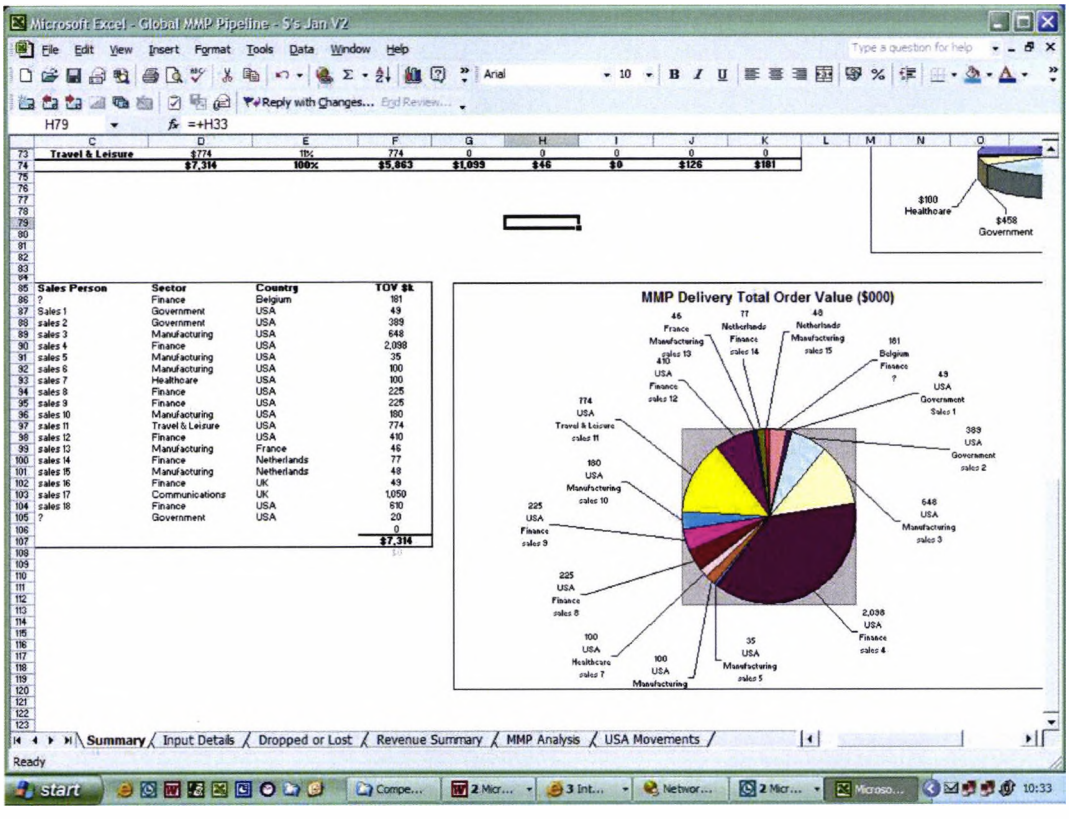


Figure B.1: Excel screenshot: Data and associated pie chart showing the order value breakdown with labelling identifying the country, market sector and sales person.

The spreadsheet format facilitates entry of data and commentary in a grid format that allows a high degree of dimensionality, but relies on the user's experience to choose appropriate tools and formats to add charts and graphs to the display.

The framework was used to evaluate Excel using a dataset of sales prospects; in different countries with differing commercial terms; by market sector; by sales person; by month; won, lost or percentage likelihood of winning; margin; revenue total and broken down by month over a three year period. The primary aim of the model used is to forecast revenue and margin and compare the return against investments in each country or region. The results of applying the framework are shown in Table B.1 and the following report.

Examples of the charts used to analyze the data are shown in Figures B.1 and B.2. Figure B.1

shows a portion of the summary / overview sheet. Figure B.2 shows a histogram from the revenue summary sheet.

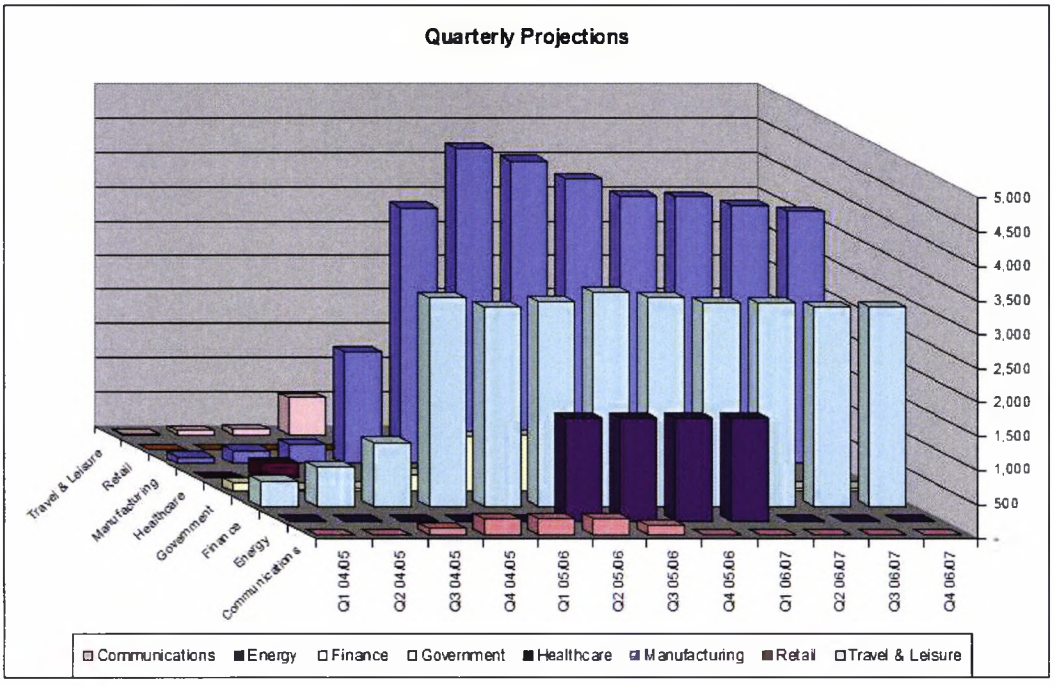


Figure B.2: Excel screenshot: Revenue shown by quarter by market sector.

Table B.1: Framework applied to Excel by Business User.

Problem Aspect		Comment	Suggested Approach
Real World			
Domain relevance	√	Users will require a good understanding of the aim of the visualization being created. The two or three dimensions that are to be compared by graphs and charts need to be chosen against specific criterion or convention.	Illustrative datasets and feature demonstration. Standard models should be available, giving examples of layout for maximum flexibility.
Data collection impact	√	Data collection is done via multiple phone and face to face interviews with sales staff. These are done on a non regular basis, usually as a result of senior management concern over performance.	Regular and routine submission of figures and reports by sales staff would improve the consistency of the data collected.
Measurement error	√	Data input is often affected by subjective issues such as the point in the sales cycle or the concern of sales staff over perception of individual performance.	Link personal recompense to accuracy of forecasting.
Raw Data			

continued on next page

Table B.1: *continued*

Problem Aspect		Comment	Suggested Approach
Multiple structures	√	There are a vast array of potential visualizations, some of which are linked to particular mathematical constructs. The user will, in most cases be guided by convention or recently seen examples.	Illustrative datasets and feature demonstration. Standard models should be available, giving examples of layout for maximum flexibility.
Choice of object	√	There are many choices of object that can be made, from mathematical functions to graphical objects.	Illustrative datasets and feature demonstration. Standard models.
Data type	–	Numerical, categories, regions, personnel and dates. Data is highly structured, with data automatically entered into one part dependant on input in other sections.	
Dimensionality	√	There are 92 records with 65 fields. Dimension reduction is not used. Choice of dimensions to visualize has to be specific and accurately chosen. Failure to do this properly results in confusing and misleading representations. It is very difficult to get an overview of the entirety of the data set.	Pedagogic model. Increased user guidance and use of examples to help choices.
Associated metadata	–	Metadata is created within the visualization as a 'dash board' attempt to give an over view of the data. However assumptions are made as to which dimensions should be compared with each other, usually from convention. Potentially important links can easily be lost.	Variety. Peer level reviews of metadata visualizations will maximise the experience brought to the visualization.
Selection impact	√	Choice of data to display or manipulate is completely user decided. The data models and displays used should be guided by an experienced user of the application.	Pedagogic mode + training required.
Data for Layout			
User making choice	√	Type of chart; colours; textual notation; ordering and scales are all choices for the user.	Pedagogic mode + training and examples required.
Predictability	–	The layouts achieved using the various mathematical and graphical formula are predictable in all cases that the author is aware of.	
Abstraction	–	No reduction of dimensions is available, the abstractions available are all two or three dimensional and do not involve the loss of any data.	
Display			

continued on next page

Table B.1: *continued*

Problem Aspect		Comment	Suggested Approach
Unfamiliarity	√	The high familiarity of this type of visualization (spreadsheet) causes problems of being over confident in the accuracy of the visualization produced. Users often will not question highly unusual results that have been generated by errors	Proactivity, range checks and independent (from the designer) quality testing of models created.
Spatial Meaning	–	Spatial meaning is generally unambiguous.	
Hidden Features	√	Some displays can have over plotted areas that hide important data, this is particularly relevant when three dimensions are being shown in a graph or chart. However it is generally clear to the user that some data is being obscured.	Visual tracking would help user to explore the impact of different values and allow them to 'uncover' the part of the visualization that is hidden.
Multiple windows	√	The large number or dimensions available and the limited dimension visualizations make it inevitable that multiple charts will be created to look at various interdependencies. The application allows for reasonable segmentation by the use of tagged worksheets. Visualizations can be resized to fit onto screen readable or A4 formats.	Pedagogic mode + training and guidance on the structure and layout of the spreadsheet should help with this issue.
Mapping complexity	√	Mapping of data to visualization objects is often ambiguous. This is overcome to some extent by the ability to label different portions of the charts. However this is done automatically and in many cases, where a large number of values are displayed, the chart labels clash and become unreadable. Considerable time is then required from the user to manually re set the positions of the labels.	Query and interaction. For example, roll-over labelling of charts.
Ordering	–	Some visualizations are susceptible to the ordering of data within the sheet; particularly variables that change over time, convention will determine the interpretation of line graphs with time on one axis.	
Human			

continued on next page

Table B.1: *continued*

Problem Aspect		Comment	Suggested Approach
Expertise	√	The level of expertise will have a direct effect on the success of the application. Critical aspects in the design process are: - initial layout design; knowledge of visualization techniques available and the ability to program with Visual Basic.	Training of users; example layouts and encouragement to use resources available on the internet will make a significant difference in the models created by any particular user.
Perception	√	The use of colour is very much at user discretion and is a potential stumbling block unless conventions are understood and maintained.	Proactivity within the application would assist. Agree use of colour with participants and document.
Cognition	√	Many of the visualizations used by designers of Excel spreadsheets are quickly understood by convention. Care must be taken when conventional techniques are used to display unconventional data. Also the generally two dimensionality of the displays can lead to misrepresentation if inappropriate dimensions are compared.	Pedagogic model.

Report for Excel

- Illustrative datasets and feature demonstration.

  The wealth of features and the need to properly structure the data within the spreadsheet can lead an inexperienced user to struggle to layout the data so that the tools within the application can be properly deployed. There are many resources available including data sets and examples of how to use the various features, however specific examples for the type of dataset under investigation are not always easily found and additional templates would be useful for guidance with notes highlighting the most telling visualizations.

- Pedagogic mode.

  Many tutorials are available for Excel and an interactive help function comes with the application. Unfortunately many users ignore these and find the automated help functions irritating, and hence turn them off. A more user engaging help function might overcome this reluctance.

- Variety

  The user should be invited to move beyond the standard visualization such as line graphs and pie charts to attempt to give an overview of the dataset. In the example used for this evaluation some attempts should have been made to look for trends in successful transition from

a sales prospect to successful sale, tracking the rate of success and examining geographical differences.

- Proactivity

  There are a number of features available in Excel for the designer of the model to check for out of range errors and unlikely data input. Unfortunately this is rarely used, this was the case in the evaluation example. An automated scan for the use of these techniques could usefully remind the user of the potential to use these features.

- Visual tracking

  The ability to interact with the visualization, moving line or point with the dataset changing to reflect this is potentially a counter intuitive feature. However the insight this would give to the impact of certain component data values, would be most useful.

- Query and interaction.

  With the often confusing layering of the automated labelling system, an improvement might be made in some cases by such techniques as role over labelling.

B.2 Daisy

The visualization application *Daisy* was introduced in Section 4.3.10. It can be downloaded from http://www.daisy.co.uk/daisy.html. The dataset used to evaluate Daisy, provided with the downloaded application, consists of 350 records of phone calls with the following fields; date; time; extension; number dialled; line used; duration of call.

The visualization consists of nodes arranged in a circle. Each node represents a segment of one of the field values, for example, duration 10 to 15 minutes. The nodes are connected together for each of the records, with the lines graded in shade according to the relative weighting of the record, as set by the user. Figure B.3 shows the effect of weighting by frequency of call, Figure B.4 the effect of weighting by duration of call. The results of applying the framework are shown in Table B.2

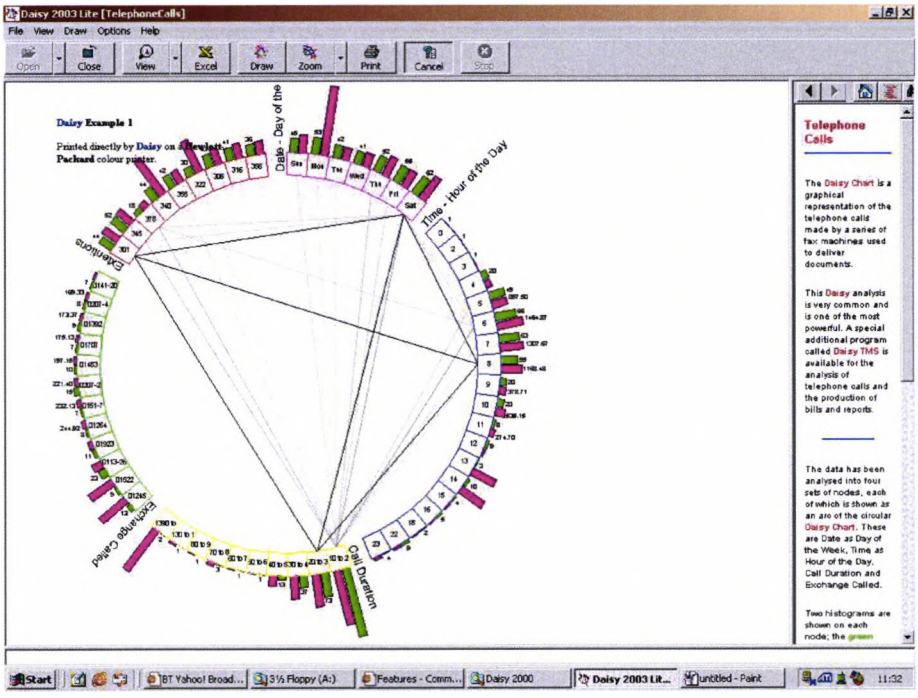


Figure B.3: Frequency of call shown by; telephone number dialled; date; time; extension and duration of call.

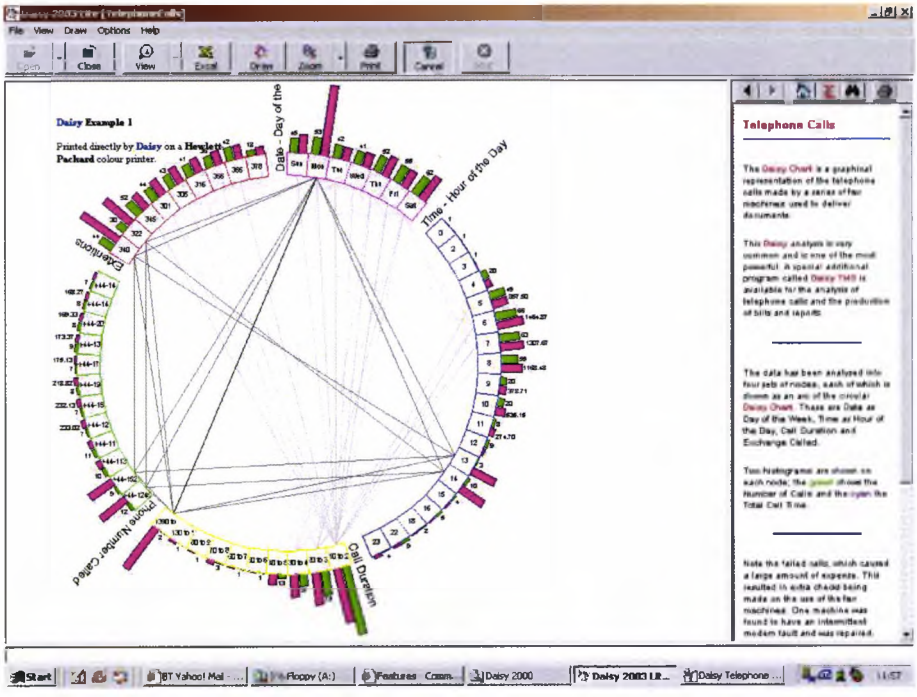


Figure B.4: Daisy screenshot: Duration of call shown by; exchange dialled; date; time; extension and duration of call.

Table B.2: Framework applied to Daisy by Business User.

Problem Aspect		Comment	Suggested Approach
Real World			
Domain relevance	—	Users will not need a great deal of domain knowledge to use and explore data with this visualization. The data does not need any special layout and is set out as a straight forward two dimensional file, either in an Excel spreadsheet or CSV (Comma Separated Values) file.	
Data collection impact	—	The simplicity of the file structure leads to easy automation of input values. The telephony dataset used for evaluation could be created automatically from the onsite PBX.	
Measurement error	—	Given the collection method described above there is no problem with errors occurring.	
Raw Data			
Multiple structures	√	Whilst this is a single structure visualization it does mix the connected node approach with histograms at the edge of the circle. With large data sets this gives a visual reference to see exceptional data sets. The zoom facility allows for rapid focus on areas of interest. Navigation by panning across the visualization is counter intuitive and clumsy.	Use conventional panning method.
Choice of object	—	Choice of object is relatively limited in the single structure approach and allows the new user to quickly understand the relevant choices.	
Data type	—	Data can be text or numerical values. There is no conversion of these values other than standard mathematical formula such as averages and standard deviation. Text can be compared for similarity.	
Dimensionality	√	Specifically designed to identify significant records out of large scale datasets Daisy is well equipped to deal with multidimensional data. The user can gain a good over view of the data and quickly identify exceptional records, or group of records. Dimensions can be reduced manually by eliminating those that are not needed from the visualization.	Feedback. Automation of the process of identifying redundant dimensions and then modifying the visualization should be considered. In diagram 2 above the application could have identified that the phone number fields where not relevant and left them out of the diagram.

continued on next page

Table B.2: *continued*

Problem Aspect		Comment	Suggested Approach
Associated metadata	√	Metadata is displayed in the histograms that sit at the edge of the circle. The nature of this metadata is chosen by the user to focus on the most relevant issues. Starting with the over view of all data the selection of this metadata could be difficult.	Proactivity. Selection of metadata displayed could be automated at the first instance based on the existence of outliers or other standard features of datasets, as selected by the user.
Selection impact	√	Selections are relatively limited; it is primarily around choice of what should or should not be included and the choice of weighting records to be displayed in the interconnections.	Proactivity. Some suggestions based on the structure of the dataset would be helpful.
Data for Layout			
User making choice	√	There is virtually no choice available for the user in the layout of the visualization. Whilst this simplifies the user interface considerably, some advanced options might help with specific identification issues.	Query and interaction. The ability to choose colours for ranges of field values might be helpful. The use of colour gradation in visualization of the interlinks might give additional insight.
Predictability	–	The visualization does not involve any ambiguous conversions or dimension reductions.	
Abstraction	–	The only abstraction used is in the weighting of the links between the representation of the nodes. The author does not have sight of how this weighting has been achieved, but in the examples used the results were clearly correct when compared to the actual data records.	
Display			
Unfamiliarity	–	For new users this visualization will be unfamiliar. However there are clear descriptions of the meaning of each component. After a little experiment the author was able to focus on specific issues within the data set and add / change the display to identify areas of interest.	
Spatial Meaning	–	Once understood the spatial layout is clear and unchanging.	
Hidden Features	√	With large scale datasets labelling is likely to be an issue although this is countered by the ease of use of the zoom facility and the use of roll over labelling. However the interconnections are not labelled or enabled for interrogation.	Query and interaction. Given that the interconnections between the nodes are one of the key characteristics that help to understand the diagram it is surprising that there is no ability to interrogate individual links and understand the weight factors.

continued on next page

Table B.2: *continued*

Problem Aspect		Comment	Suggested Approach
Multiple windows	–	The visualization itself is designed in one window. More windows appear if the user requests detailed information of the records themselves or wishes to modify the items displayed. Due to the lack of interaction between the windows there is little need to see the whole of the visualization whilst making changes, the user has to manually request recalculation and re drawing of the visualization.	
Mapping complexity	√	As the weighting mechanism is unclear to the user the impact of different dataset features such as large standard deviation or a number of significant outliers is not understood.	Feature fingerprinting. The use of known datasets to explore the impact of specific datasets to the gradation of the interconnections would help to interpret the significance of the highlighted connections.
Ordering	–	There is no significance in the ordering of the data within the records.	
Human			
Expertise	–	Expertise will have some impact on the ability of the user to interpret what is being looked at. However the relative simplicity of the interface minimises this impact and the user should quickly be able to effectively use the application.	
Perception	–	Once understood the user is unlikely to misunderstand the visualization. Colours are used only to differentiate differing components and bring little to the understanding of the diagram.	
Cognition	–	This is a very focused application that assists in identification of connections between fields. This focus limits the need to understand different representations and simplifies the task of investigating the data.	

Report for Daisy

- Feedback.

  Automation of the process of identifying redundant dimensions and then modifying the visualization should be considered. In Figure B.4 above, the application could have identified that

the phone number fields were not relevant and left them out of the diagram.

- Proactivity.

  Selection of metadata displayed could be automated in the first instance based on the existence of outliers or other standard features, as selected by the user. Also some suggestions of which fields should be used for weighting, based on the structure of the dataset, would be helpful.

- Query and interaction.

  The ability to choose colours for ranges of field values might be helpful. The use of colour gradation in visualization of the interlinks might give additional insight. Also, given that the interconnections between the nodes are one of the key characteristics that help the user to understand the diagram, it is surprising that there is no ability to interrogate individual links and understand the weight factors.

- Feature fingerprinting.

  The use of known datasets to explore the impact of specific datasets to the gradation of the interconnections would help to interpret the significance of the highlighted connections.

B.3 Ggobi

Ggobi is a multivariate visualization tool developed by the American telephone company, AT&T. It is freely available (http://www.ggobi.org/), open source, and appears to be regularly updated; recent additions such as XML and integration to the statistical analysis software R being examples. It is primarily used by the research community and is often sited as a 'best of breed' visualization tool for users with large and complex data sets that are to be analysed for as yet undiscovered trends or associations.

The data set used for the evaluation was the percentage composition of eight fatty acids found in the lipid fraction of 572 Italian Olive oils collected from 9 different areas. The hypothesis was explored that the oils of individual areas had different signatures and could be identified by the percentages of the fatty acids.

Figures B.5 and B.6 show screen shots examining the percentage of Oleic acid by region and comparison of Oleic acid to Palmitoleic acid, labelled with regions. The evaluation was limited to this task and a general exploration of features and functions. The software package is extensive and clearly designed for users with a reasonable grasp of visualization techniques. It was not possible for the author to explore all of the features or necessarily understand the purpose of all the functions that were identified. Hence the evaluation is based on initial impressions and specific functionality used in the task.

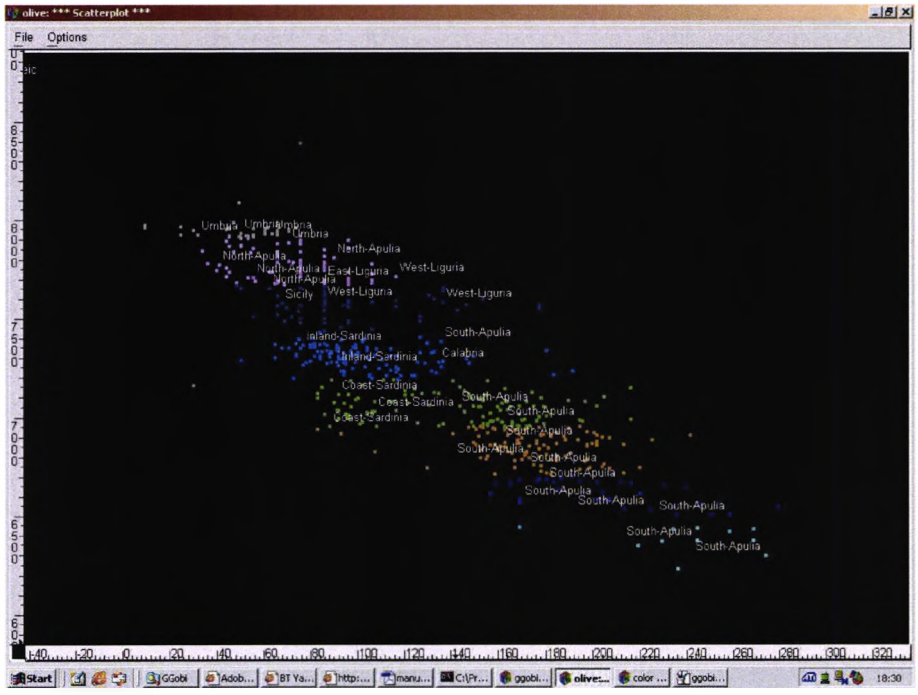


Figure B.5: Ggobi screen shot: Oleic acid by region.

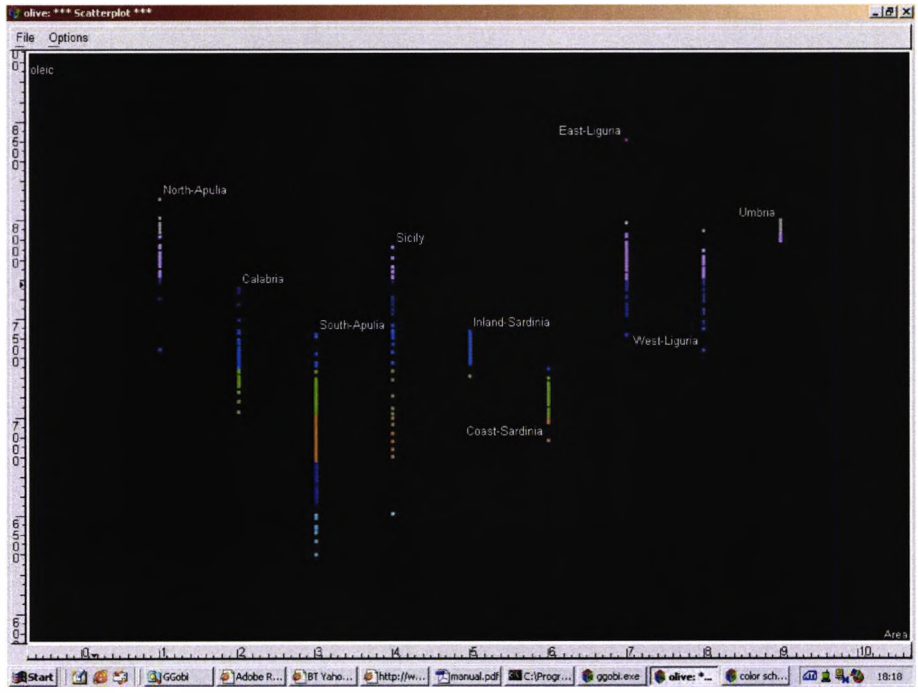


Figure B.6: Ggobi screen shot: % Oleic acid plotted by % Palmitoleic acid - labelled by regions of origin.

Table B.3: Framework applied to Excel by Business User.

Problem Aspect		Comment	Suggested Approach
Real World			
Domain relevance	√	The nature of the dataset was relatively straightforward to understand. However the visualization tool required significant understanding of visualization techniques or significant experience with the tool itself.	Pedagogic mode: Often the labelling of buttons or menu items was insufficient to understand the function. More explanation would help the inexperienced user.
Data collection impact	–	No impact in this case. However an understanding of the data set and the likely outcomes was important.	
Measurement error	–	Not known.	
Raw Data			
Multiple structures	√	Ggobi has a great deal of flexibility and choice in the way a visualization is created. This is ideal for a researcher looking for complex patterns. However for this relatively simple task it was frustrating to have so much choice with little understanding of the relevance of what any particular choice would have.	Pedagogic mode: Given that the tool is designed for researchers, the great choice and flexibility is a strength in Ggobi, however some additional guidance would be helpful. In this particular exercise the way in which colour was assigned to glyphs was not at all clear.
Choice of object	√	Objects and their attributes are often unclear in the visualizations created. Labelling helped although when multiple objects were on screen this became untenable. Sample labelling was then very useful.	Illustrative datasets and feature demonstration. Whilst the dataset used was regarded as an illustrative one, the large size and dimensionality made it difficult to gain comprehension of the visualization. Smaller and simpler datasets would have been a better starting point.
Data type	–	No transformational or reduction functionality was found in this exercise.	
Dimensionality	–	The dataset was large and complex. Ggobi is specifically designed to analyse this type of dataset.	
Associated metadata	–	In this example it was not possible to use or assemble any metadata.	
Selection impact	√	Selection of visualization type, the parameters to be used and the 'physical' manipulation required are all user selectable. There are both standard and novel tools available. No statistical functions where found.	Feature demonstration. Some indication of the significance of certain techniques available would help in user choice.
Data for Layout			
User making choice	√	The user choice is impressive, if confusing.	Illustrative datasets and feature demonstration. Examples of the visualizations that can be produced might inspire users to try unexpected combinations.

continued on next page

Table B.3: *continued*

Problem Aspect		Comment	Suggested Approach
Predictability	–	As there was no transformational or reducing functionality in the software the visualization always showed absolute values.	
Abstraction	–	None available	
Display			
Unfamiliarity	√	The user can choose familiar or unfamiliar displays.	
Spatial Meaning	√	The spatial meaning of the data represented was often unclear. Labelling individual points helped to understand the positioning to some extent but the attribute and values represented by direction and distance did not seemed to be labelled.	Feature fingerprinting. With large datasets there is a limit to the number of labels that can be displayed at any one time. The ability to gain an intuitive understanding of the representation of attribute and value through the use of a known dataset would help.
Hidden Features	√	Overplotting in the scatter plots was evident in the application. However in this exercise it was not a problem.	
Multiple windows	√	The screen quickly became crowded with two or three windows. The main issue was that the visualization had to be reduced in size in order to accommodate the tool control panels, this reduced the ability to understand the visualization.	Proactivity. The application could provide an optimised tiling, dependant on the control panels currently in use.
Mapping complexity	√	The mapping of the data to the points on visualizations was often completely unclear. Sample labelling was the only way to gain any clarification.	Feature fingerprinting. Comment as in special meaning above.
Ordering	–	No issues were found.	
Human			
Expertise	√	Ggobi is clearly meant for users that will spend considerable time learning the tool and have a level of expertise in visualization. The terms used and the brief explanations offered require a background knowledge.	Proactivity. It might be helpful to have a collaborative framework for experienced users to contribute to when new or particularly helpful techniques are found. If these could be categorized and presented to users dependant on context they could choose to try them / contact the contributor.
Perception	√	The user choice for all aspects of the display allow for the creation of potentially confusing displays.	Selection and standardisation. The ability to reset colours and other attributes to a linear or 'normal' progression might be useful after experimentation.

continued on next page

Table B.3: *continued*

Problem Aspect		Comment	Suggested Approach
Cognition	√	Too many variables for an inexperienced user. Whilst there is a manual available the process of learning from this is too distant and ultimately the user must interact with the software to understand it properly and be able to choose relevant displays.	Feature Fingerprinting: With such a flexible tool the ability to use standard and known datasets, perhaps with particular features would give the user insight into the most relevant or insightful visualizations.

Report for Ggobi

- Pedagogic mode:

  Often the labelling of buttons or menu items was insufficient to understand the function. More explanation would help the inexperienced user. Given that the tool is designed for researchers the great choice and flexibility is a strength in Ggobi, however some additional guidance would be helpful. In this particular exercise the way in which colour was assigned to glyphs was not at all clear,

- Illustrative datasets and feature demonstration:

  Whilst the dataset used was regarded as an illustrative one, the large size and dimensionality made it difficult to gain comprehension of the visualization. Smaller and simpler datasets would have been a better starting point. Some indication of the significance of certain techniques available would help in user choice. Examples of the visualizations that can be produced might inspire users to try unexpected combinations.

- Feature fingerprinting.

  With large datasets there is a limit to the number of labels that can be displayed at any one time. The ability to gain an intuitive understanding of the representation of attribute and value through the use of a known dataset would help. With such a flexible tool the ability to use standard and known datasets, perhaps with particular features would give the user insight into the most relevant or insightful visualizations.

- Proactivity.

  The application could provide an optimised tiling, dependant on the control panels currently in use.

  It might be helpful to have a collaborative framework for experienced users to contribute to when new or particularly helpful techniques are found. If these could be categorized and

presented to users dependant on context they could choose to try them / contact the contributor.

- Selection and standardisation.

  The ability to reset colours and other attributes to a linear or 'normal' progression might be useful after experimentation.

B.4 Reviewer's Comments after Using the Framework

The reviewer was asked to record their comments in the following categories: ease of use; time required; insight; clarity.

- Ease of Use. The overall concept of the framework is easily understood. The process of going through the questions, referring to the suggested areas of improvement and finally summarising the comments under type of improvement encourages a broad view initially, focussing down to the specific recommendations as the process finishes. This avoids any tendency to put any unrepresentative emphasis on specific features that immediately strike the user as difficult or confusing.

  The questions do not always apply to the application under review, and in some cases the interpretation would change with different types of visualisation. This made the reviewer somewhat uncertain of the precise meaning of the question. However in the spirit of using the framework as a tool for evaluation this variation did not seem to be critical in catching the strengths and weaknesses of the visualisation.

- Time Required. Using the framework took approximately two and a half hours for each application reviewed. Before this time, was taken to become familiar with the features of the software. The amount of time spent doing this varied, primarily dependant on the amount of prior knowledge and the complexity of the visualisation. In particular the reviewer had detailed knowledge of Excel, spent an hour understanding the Daisywheel software and two and a half hours with Ggobi.

  For a designer these times might well expand considerably as the detailed understanding of the application would require much fuller answers and more thought about the implications of the potential recommendations. Also for applications such as Ggobi, with substantial domain knowledge being a requirement, the reviewer should be a proficient user with some considerable experience.

- Insight. It was particularly illuminating to use the framework to evaluate a well known package, Excel. Clearly an evaluation of such a mature and widely used product is unlikely to

identify many deficits in terms of functions and features. However it was interesting to see that there are a number of considerations thrown up by such an evaluation format that considers the effectiveness of the end result and guides the reviewer to think of specific aspects of the application.

The most difficult application for the reviewer to evaluate was the Ggobi software; the insights achieved were limited to the first impressions of a new user trying to understand the capability and meaning of the many features available. However from the experience of the review of Excel, it is reasonable to extrapolate that the framework would be useful for evaluation by an experienced user.

- Clarity. For the most part the meaning and context of the questions asked were easily understood, however they were occasionally open to interpretation. This was particularly true for the Excel review in which it was occasionally unclear whether the questions applied to the underlying tool or the visualisation that was built on top of this.

Bibliography

Agrawal, R., Lin, K., Sawhney, H. S., and Sim, K. (1995). Fast similarity search in the presence of noise, scaling, and translation in time-series databases. In *Proceedings of 21st VLDB Conference*, Zürich, Switzerland.

Ahlberg, C. and Shneiderman, B. (1994). The alphaslider: a compact and rapid selector. In *Proceedings CHI'94*, pages 365–371. ACM.

American Heritage Dictionary of the English Language (2000).

Andrews, D. F. and Herzberg, A. M. (1985). *Data: A Collection of Problems from many Fields for the Student and Research Worker*. Springer-Verlag.

Andrienko, N., Andrienko, G., Savinov, A., Voss, H., and Wettschereck, D. (2001). Exploratory analysis of spatial data using interactive maps and data mining. *Cartography and Geographic Information Science*, *28*(3), 151–165.

Ankerst, M. (2001). Visual data mining with pixel-oriented visualization techniques. In *ACM SIGKDD Workshop on Visual Data Mining*, San Francisco, CA.

Anscombe, F. J. (1973). Graphs in statistical analysis. *American Statistician*, *27*, 17–21. Feb.

Atkin, R. (1981). *Multidimensional Man*. Penguin Books.

Becker, R. A., Eick, S. G., and Wilks, A. R. (1995). Visualizing network data. *IEEE Transactions on Visualization and Computer Graphics*, *1*(1), 16–28.

Benedikt, M. (1991). *Cyberspace: First Steps*, chapter Cyberspace: Some proposals, pages 119–224. Cambridge, MA: MIT Press.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*, *May*, 34–43.

Bertin, J. (1983). *Semiology of Graphics*. University of Wisconsin Press, Madison. Original French Edition, 1967.

Bienfait, B. and Gasteiger, J. (1997). Checking the projection display of multivariate data with colored graphs. *Journal of Molecular Graphics and Modeling*, *15*(4), 203–215.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press.

Böhm, C., Berchtold, S., and Keim, D. A. (2001). Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. *ACM Computing Surveys (CSUR)*, *33*(3), 322–373.

Bottoni, P., Costabile, M. F., Levialdi, S., and Mussio, P. (1998). Specification of visual languages as means for interaction. In Marriott and Meyer (1998).

Bourgain, J. (1985). On Libschitz embedding of finite metric spaces in Hilbert space. *Israel J. Math.*, *52*, 46–52.

Bray, T. (1996). Measuring the web. In *Fifth International World Wide Web Conference, Paris, France*.

Bruls, M., Huizing, K., and van Wijk, J. J. (2000). Squarified treemaps. In *Data Visualization 2000, Proceedings of the Joint EUROGRAPHICS and IEEE TCVG Symposium on Visualization in Amsterdam, The Netherlands, May 29-31, 2000*, pages 33–42. Springer.

Buttenfield, B. and Beard, K. (1994). Graphical and geographical components of data quality. In Hearnshaw, H. M. and Unwin, D. J. (Eds.), *Visualization in Geographical Information Systems*, pages 141–149. Chichester: Wiley.

Card, S. K., Mackinlay, J. D., and Shneiderman, B. (1999). *Readings in Information Visualization: Using Vision To Think*. Morgan Kaufmann.

Chang, N. S. and Fu, K. S. (1980). Query-by-Pictorial-Example. *IEEE Transactions on Software Engineering*, *SE-6*(6), 519–524.

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

Chi, E. H. (2000). A taxonomy of visualization techniques using the data state reference model. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 69–75.

Cleveland, W. (1993). *Visualizing Data*. Hobart Press, Summit, NJ.

Cleveland, W. S. and McGill, M. E. (1984). *Dynamic Graphics for Statistics*. Pacific Grove, CA: Wadsworth and Brooks/Cole.

Corsini, P., Lazzerini, B., and Marcelloni, F. (2002). Combining supervised and unsupervised learning algorithms for data clustering. In *Unknown*.

deRisi, J., Iyer, V. R., and Brown, P. O. (1997). Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, *278*, 680–686.

di Battista, G., Eades, P., Tamassia, R., and G.Tollis, I. (1999). *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall.

Drecki, I. (2002). Visualisation of uncertainty in geographical data. In Shi, W., Fisher, P. F., and Goodchild, M. F. (Eds.), *Spatial Data Quality*, pages 140–159. London: Taylor & Francis.

Dykes, J. A. (1997). Exploring spatial data representation with dynamic graphics. *Computers and Geosciences*, *23*(4), 345–370.

Ehlschlaeger, C. R., Shortridge, A. M., and Goodchild, M. F. (1997). Visualizing spatial data uncertainty using animation. *Computers & Geosciences*, *23*(4), 387–395.

Eick, S. G. (1996). Aspects of network visualization, special report, computer graphics and visualization. *Computer Graphics and Applications*, *16*(2).

Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. In *Proc. Natl. Acad. Sci.*, volume 95, pages 14863–14868.

Erbacher, R. and Frincke, D. (2000). Visualization in detection of intrusions and misuse in large scale networks. In *Proceedings of the IEEE International Conference on Information Visualization*, London.

Fabrikant, S. I. and Skupin, A. (2004). Cognitively plausible information visualization. In MacEachren, A., Kraak, M.-J., and Dykes, J. (Eds.), *Exploring Geovisualization, International Cartographic Association, Commission on Visualization and Virtual Environments*. Amsterdam: Elseviers.

Faratin, P., Sierra, C., and Jennings, N. R. (2000). Using similarity criteria to make negotiation tradeoffs. In *Proc. of 4th Int. Conf. on Multi-Agent Systems ICMAS-2000*, pages 119–126, Boston, USA. IEEE Computer Society.

Fienberg, S. E. (1979). Graphical methods in statistics. *American Statistician*, *33*(4), 165–178.

Fisher, L. and Van Ness, J. W. (1971). Admissible clustering procedures. *Biometrika*, *58*, 19–104.

Fisher, P., Wood, J., and Cheng, T. (2004). Where is helvellyn? fuzziness of multiscale landscape morphometry. *Transactions of the Institute of British Geographers*, *29*(1), 106–128.

Fisher, P. F. (1994). Animation and sound for the visualization of uncertain spatial information. In Hearnshaw, H. M. and Unwin, D. J. (Eds.), *Visualization in Geographical Information Systems*, pages 181–185. Chichester: Wiley.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, *7*(Part II), 179–188.

Foner, L. (1995). Clustering and information sharing in an ecology of cooperating agents. In *AAAI Spring Symposium '95 on Information Gathering in Distributed, Heterogeneous Environments, Palo Alto*.

Foner, L. (1997). Yenta: a multi-agent, referral-based matchmaking system. In *The First International Conference on Autonomous Agents, Marina del Rey, California*. ACM press.

Furnas, G. W. (1981). The FISHEYE view: A new look at structured files. Technical report, AT&T Bell Laboratories, Murray Hill, NJ.

Gibson, W. (1984). *Neuromancer*. New York: Ace Books.

Goodchild, M., Buttenfield, B., and Wood, J. (1994). Introduction to visualizing data validity. In Hearnshaw, H. M. and Unwin, D. J. (Eds.), *Visualization in Geographical Information Systems*, pages 141–149. Chichester: Wiley.

Gordon, A. D. (1990). Constructing similarity measures. *Journal of Classification*, *7*, 257–269.

Gordon, A. D. (1999). *Classification* (2nd ed.). Chapman and Hall / CRC.

Green, T. R. G. (2000). Instructions and descriptions: Some cognitive aspects of programming and similar activities. In Gesu, V. D., Levialdi, S., and Tarantino, L. (Eds.), *Proceedings of Working Conference on Advanced Visual Interfaces*. Invited paper.

Grinstein, G. and Levkowitz, H. (1995). *Perceptual Issues in Visualization*. Springer.

Hand, D. J., Daly, F., Lunn, A. D., McConway, K. J., and Ostrowski, E. (1994). *A Handbook of Small Data Sets*. Chapman and Hall.

Haslett, J., Wills, G. J., and Unwin, A. R. (1990). SPIDER: An interactive statistical tool for the analysis of spatially distributed data. *International Journal of Geographical Information Systems*, *4*(3), 285–296.

Hecht-Nelson, R. (1994). Context vectors: General purpose approximate meaning representations self-organized from raw data. In Zurada, J. M., Marks, R. J., and Robinson, C. J. (Eds.), *Computational Intelligence: Imitating Life*, pages 43–56. Piscataway, New York: IEEE Press.

Herman, G. T. and Levkowitz, H. (1992). Color scales for image data. *Computer Graphics and Applications*, 72–80.

Herman, I., Melancon, G., and Marshall, M. (2000). Graph visualization and navigation in information visualisation: a survey. *IEEE Transactions on Visualization and Computer Graphics*, *6*(1), 24–43.

Inselberg, A. (1997). Multidimensional detective. In *Proceedings of InfoVis97*, pages 100–107. IEEE Symposium on Information Visualization, IEEE.

Johnson, B. and Shneiderman, B. (1991). Tree-maps: A space-filling approach to the visualization of hierarchical information structures. In *Proceedings of IEEE Visualization '91*, pages 284–291. IEEE.

Jong, K., Marchiori, E., and van der Vaart, A. (2003). Finding clusters using support vector classifiers. In *European Symposium on Artificial Neural Networks, Bruges, Belgium*, pages 223–228.

Keim, D., Bergeron, D., and Pickett, R. (1995). *Perceptual Issues in Visualization*, chapter Test Data Sets for Evaluating Data Visualization Techniques. Springer.

Keim, D. A. (2000). Designing pixel-oriented visualization techniques: Theory and applications. *IEEE Transactions on Visualization and Computer Graphics*, *6*(1).

Keim, D. A. (2001). Visual exploration of large data sets. *Communications of the ACM*, *44*(8), 39–44.

Keim, D. A. and Ankerst, M. (2001). Visual data mining and exploration of large databases. Tutorial at ECML/PKDD01.

Keim, D. A. and Kriegel, H. P. (1994). VisDB: Database exploration using multidimensional visualization. *IEEE Computer Graphics & Applications Journal*, *14*(5), 40–49.

Keogh, E., Chakrabarti, K., Pazzani, M., and Mehrotra, S. (2000). Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and Information Systems*, *3*(3), 263–286.

Keogh, E. and Kasetty, S. (2002). On the need for time series data mining benchmarks: A survey and empirical demonstration. In *8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada*, pages 102–111.

Keogh, E. and Pazzani, M. (1999). Relevance feedback retrieval of time series data. In *22th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 183–190.

Knight, C. (2000). *Virtual Software in Reality*. PhD thesis, Department of Computer Science, Durham, UK.

Kohonen, T. (1997). *Self-organising maps* (2nd ed.). Springer-Verlag.

Kohonen, T., Kaski, S., Lagus, K., Salojärvi, J., Honkela, J., Paatero, V., and Saarela, A. (2000). Self organization of a massive document collection. *IEEE Transactions on Neural Networks, Special Issue on Neural Networks for Data Mining and Knowledge Discovery*, *11*(3), 574–585.

Kruskal, J. B. (1964a). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrica*, *29*(1).

Kruskal, J. B. (1964b). Nonmetric multidimensional scaling: a numerical method. *Psychometrica*, *29*(2).

Kruskal, J. B. (1977). The relationship between multidimensional scaling and clustering. In Van Ryzin, J. (Ed.), *Classification and Clustering*, pages 17–44. New York: Academic Press.

Lamm, S., Reed, D., and Scullin, W. (1996). Real-time geographic visualization of world wide web traffic. In *Fifth International World-Wide Web Conference, May 6-10, 1996, Paris, France*.

Lamping, J. and Rao, R. (1996). The Hyperbolic Browser: A focus + context technique for visualizing large hierarchies. *Journal of Visual Languages and Computing*, *7*(1), 33–55.

Lebart, L., Salem, A., and Berry, L. (1997). *Exploring Textual Data*. Kluwer Academic Publishers.

Linial, N., London, E., and Rabinovich, Y. (1995). The geometry of graphs and some of its algorithmic applications. *Combinatorica*, *15*(2), 215–245.

Liu, J. (2002). Understanding emergent web regularities with information foraging agents. In *Proc. of First International Joint Conference on Autonomous Agents and Multiagent Systems, Bologna, Italy*. ACM press.

Lucieer, A. and Kraak, M. J. (2002). Interactive visualization of a fuzzy classification of remotely-sensed imagery using dynamically linked views to explore uncertainty. In Lowell, G. H. A. K. (Ed.), *5th International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences, July 10-12*, pages 348–356, Melbourne.

MacDonald, L. W. (1990). Using colour effectively in displays for computer-human interface. *Displays*, 129–142. July.

MacEachren, A. M. (1994). Time as a cartographic variable. In Hearnshaw, H. M. and Unwin, D. J. (Eds.), *Visualization in Geographical Information Systems*, pages 115–130. Chichester: Wiley.

MacEachren, A. M. (2002). Visualizing uncertain information. *Cartographic Perspectives*, *13*, 10–19.

MacEachren, A. M. and Kraak, M.-J. (2001). Research challenges in geovisualization. *Cartography and Geographic Information Science*, *28*(1), 3–12.

MacEachren, A. M., Wachowicz, M., Edsall, R., Haug, D., and Masters, R. (1999). Constructing knowledge from multivariate spatiotemporal data: Integrating geographical visualization with knowledge discovery in database methods. *International Journal of Geographical Information Science*.

Mackinlay, J. D., Robertson, G. G., and Card, S. K. (1991). Perspective wall: Detail and context smoothly integrated. In *Proceedings of SIGGH'91*, pages 173–179.

Maiden, N. A. M. and Rugg, G. (1996). ACRE: Selecting methods for requirements acquisition. *Software Engineering*, *11*.

Marriott, K. and Meyer, B. (Eds.). (1998). *Visual Language Theory*. Springer.

Matoušek, J. (2002). *Lectures on Discrete Geometry*. Springer.

Meuzelaar, H. L. C., Haverkamp, J., and Hileman, F. D. (1982). *Pyrolysis Mass Spectrometry of Recent and Fossil Biomaterials*. Amsterdam: Elsevier.

Mihalisin, T., Timlin, J., and Schwegler, J. (1991). Visualizing multivariate functions, data, and distributions. *IEEE Computer Graphics and Applications*, *11*(13), 28–35.

Milligan, G. W. and Cooper, M. C. (1988). A study of standardization of variables in cluster analysis. *Journal of Classification*, *5*, 181–204.

Monmonier, M. (1991a). Ethics and map design: Six strategies for confronting the traditional one-map solution. *Cartographic Perspectives*, *10*, 3–8.

Monmonier, M. (1991b). *How to Lie with Maps*. Chicago, IL: University of Chicago Press.

Narayanan, N. H. and Hübscher, R. (1998). Visual language theory: Towards a human-computer interaction perspective. In Marriott and Meyer (1998).

Noy, P. Signature exploration, a means to improve comprehension of complex visualization processes: Issues and opportunities. In MacEachren, A., Kraak, M.-J., and Dykes, J. (Eds.), *Exploring Geovisualization, International Cartographic Association, Commission on Visualization and Virtual Environments*, page 243. Amsterdam: Elseviers.

Noy, P. and Schroeder, M. (2001). Introducing signature exploration: a means to aid the comprehension and choice of visualization algorithms. In *ECML-PKDD01 Visual Data Mining Workshop*, pages 79–91, Freiburg, Germany.

Noy, P. and Schroeder, M. (2002a). Defining like-minded agents with the aid of visualization. In *ECML/PKDD Workshop Proceedings, Helsinki, Finland*, page 81.

Noy, P. and Schroeder, M. (2002b). Defining like-minded agents with the aid of visualization. In *Proc. of First International Joint Conference on Autonomous Agents and Multiagent Systems, Bologna, Italy*, page 1292. ACM press. Poster.

Noy, P. and Schroeder, M. (2003). Approximate profile utilization for finding like minds and personalization in socio-cognitive grids. In *Proceedings of 1st International Workshop on Socio-Cognitive Grids: The Net as a Universal Human Resource*, page 41.

Noy, P. and Schroeder, M. (2004). Advancing profile use in agent societies. In Omicini, A., Petta, P., and Pitt, J. (Eds.), *Engineering Societies in the Agents World IV, 4th International Workshop, ESAW 2003, London, UK*, page 360. Springer-Verlag: Lecture Notes in Artificial Intelligence 3071.

Picard, R. W. (1995). Computer learning of subjectivity. *ACM Computing Surveys*, *27*(4), 621–623.

Plaisant, C., Kang, H., and Shneiderman, B. (2003). Helping users get started with visual interfaces: Multi-layered interfaces, integrated initial guidance and video demonstrations. In *Proceedings of Human-Computer Interaction International Conference, 2003, Crete*.

Pu, P. and Pecenovic, Z. (2000). 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.

Quinn, N. R. and Breuer, M. A. (1979). A force directed component placement procedure for printed circuit boards. *IEEE Transactions on Circuits and Systems*, *26*(6), 377–388.

Rao, R. and Card, S. K. (1994). The Table Lens: Merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In *Proceedings of CHI'94, ACM Conference on Human Factors in Computing Systems*, pages 318–322 and 481–482, New York.

Rawlins, G. J. E. (1991). *Foundations of Genetic Algorithms*. San Mateo, California, USA: Morgan Kaufmann Publishers.

Rekimoto, J. and Green, M. (1993). The information cube: Using transparency in 3D information visualization. In *Proceedings of the Third Annual Workshop on Information Technologies and Systems*, pages 125–132.

Resnikoff, H. L. (1987). *The Illusion of Reality*. New York: Springer-Verlag.

Ribarsky, W., Ayers, E., Eble, J., and Mukherjea, S. (1994). Glyphmaker: Creating customized visualization of complex data. *IEEE Computer*, *27*(7), 57–64.

Roberts, J. C. (Ed.). (2003). *International Conference on Coordinated and Multiple Views in Exploratory Visualization*. IEEE Computer Society.

Roberts, J. C. (2004). Information visualization - exploration through multiple linked views. In MacEachren, A., Kraak, M.-J., and Dykes, J. (Eds.), *Exploring Geovisualization, International Cartographic Association, Commission on Visualization and Virtual Environments*. Amsterdam: Elseviers.

Robertson, G. (2000). Leveraging human capabilities in information perceptualization. Keynote speech at IEEE International Conference on Information Visualization IV2000 London July 19-21 2000.

Robertson, G. G., Mackinlay, J. D., and Card, S. K. (1991). Cone trees: Animated 3D visualizations of hierarchical information. In *Proceedings of CHI'91, ACM Conference on Human Factors in Computing Systems*, pages 189–194, New York.

Rugg, G., Corbridge, C., Major, N. P., Burton, A. M., and Shadbolt, N. R. (1992). A comparison of sorting techniques in knowledge elicitation. *Knowledge Acquisition*, *4*(3), 270–291.

Rundensteiner, E. A., Ward, M. O., Yang, J., and Doshi, P. R. (2002). Xmdvtool: Visual interactive data exploration and trend discovery of high-dimensional data sets. In *ACM SIGMOD Conference 2002, Wisconsin*. System paper.

Russo Dos Santos, C., Gros, P., Abel, P., Loisel, D., Trichaud, N., and Paris, J. P. (2000). Mapping information onto 3D virtual worlds. In *Proceedings IEEE International Conference on Information Visualisation*.

Sarkar, M. and Brown, M. H. (1992). Graphical fisheye views of graphs. In *Proceedings of CHI'92, ACM Conference on Human Factors in Computing Systems, New York*, pages 83–91.

Schroeder, M., Gilbert, D., van Helden, J., and Noy, P. (2001). Approaches to visualisation in bioinformatics: from dendrograms to Space Explorer. *Information Sciences, 139*, 19–57.

Schroeder, M. and Noy, P. (2001). Multi-agent visualization based on multivariate data. In *Proceedings of Autonomous Agents2001, Montreal, Canada*, pages 85–91. ACM press.

Shepherd, I. D. H. (1995). Putting time on the map: Dynamic displays in data visualization and GIS. In *Innovations in GIS 2*. London: Taylor and Francis.

Shneiderman, B. (1994). Dynamic queries for visual information. *IEEE Software*, *11*(6), 70–77.

Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy of information visualizations. In *Proceedings IEEE Symposium on Visual Languages '96*, Los Alamos, CA.

Shneiderman, B. (2002). Inventing discovery tools: Combining information visualization with data mining. *Information Visualization*, *1*(1), 5–12.

Sneath, P. H. A. (1997). Some statistical problems in numerical taxonomy. *The Statistician*, *17*, 1–12.

Sokal, R. R. and Rohlf, F. J. (1980). An experiment in taxonomic judgement. *Systematic Botany*, *5*(4), 341–365.

Spence, R. (2001). *Information Visualization*. Addison-Wesley.

Spence, R. and Tweedie, L. (1998). The Attribute Explorer: information synthesis via exploration. *Interacting with Computers*, *11*(2), 137–146.

Swayne, D. F., Cook, D., and Buja, A. (1998). XGobi: Interactive dynamic data visualization in the X window system. *Journal of Computational and Graphical Statistics*, *7*(1).

Thorgard, P. (1996). *Mind: Introduction to Cognitive Science*. Cambridge MA: MIT Press.

Tobler, W. R. (1979). A transformational view of cartography. *The American Cartographer*, *6*(2), 101–106.

Tufte, E. R. (1983). *The Visual Display of Quantative Information*. Cheshire, Connecticut: Graphics Press.

Tufte, E. R. (1990). *Envisioning Information*. Cheshire, Connecticut: Graphics Press.

Turton, I., Openshaw, S., Brunsdon, C., Turner, A., and Macgill, J. (2000). Testing space-time and more complex hyperspace geographical analysis tools. In Atkinson, P. and Martin, D. (Eds.), *GIS and Geocomputation*. Taylor and Francis.

Tweedie, L. A. (1997). Characterizing interactive externalizations. In *Proceedings of CHI'97, ACM Conference on Human Factors in Computing Systems, Atlanta*, pages 375–382.

Unwin, A. (2000). Using your eyes - making statistics more visible with computers. *Computational Statistics & Data Analysis*, *32*(3-4), 303–312.

Unwin, A. (2001). Statistification or mystification? The need for statistical thought in visual data mining. In Raedt, L. D. and Siebes, A. (Eds.), *Lecture Notes in Computer Science: Principles of Data Mining and Knowledge Discovery: 5th European Conference, PKDD 2001, Freiburg, Germany, Proceedings*, volume 2168, page 494. Springer-Verlag Heidelberg. Invited talk.

Van Ness, J. W. (1973). Admissible clustering procedures. *Biometrika*, *60*, 422–4.

Veit, D., Muller, J., Schneider, M., and Fiehn, B. (2001). Matchmaking for autonomous agents in electronic marketplaces. In *Proceedings of Autonomous Agents2001*, Montreal, Canada. ACM press.

Velleman, P. F. (1992). *Data Desk 4.2*. Data Description Inc.

Wagner, G. (2003). The agent-object-relationship metamodel: Towards a unified view of state and behavior. *Information Systems*, *28*(5), 475–504.

Wang, D. and Zeevat, H. (1998). A syntax directed approach to picture semantics. In Marriott and Meyer (1998).

Ward, M. O., LeBlanc, J. T., and Tipnis, R. (1994). N-Land: a graphical tool for exploring n-dimensional data. In *Computer Graphics International Conference, Melbourne, Australia.*

Ware, C. (2000a). *Information Visualization: Perception for Design*. Morgan Kaufmann.

Ware, C. (2000b). Visualization as applied perception. Keynote speech at Joint Eurographics-IEEE TCVG Symposium on Visualization.

Webb, A. (1999). *Statistical Pattern Recognition*. Arnold.

Weiss, M. A. (2002). *Data Structures and Problem Solving using Java* (second ed.). Addison Wesley.

Williams, W. T. and Dale, M. B. (1965). Fundamental problems in numerical taxonomy. *Advanc. Bot. Res.*, *2*, 35–68.

Williamson, C. and Shneiderman, B. (1992). The dynamic homefinder: Evaluating dynamic queries in a real estate information exploration system. In *Proceedings SIGIR '92*, pages 339–346. ACM.

Wiss, U. and Carr, D. A. (1999). An empirical study of task support in 3D information visualizations. In *Proc. IEEE International Conference on Information Visualization IV'99*, pages 392–399.

Wong, P. and Bergeron, R. (1997). 30 years of multidimensional multivariate visualization. In Nielson, G. M., Hagan, H., and Muller, H. (Eds.), *Scientific Visualization - Overviews, Methodologies and Techniques*, pages 3–33. Los Alamitos, CA: IEEE Computer Society Press.

Wood, J. (1994). Visualizing contour interpolation accuracy in digital elevation models. In Hearnshaw, H. M. and Unwin, D. J. (Eds.), *Visualization in Geographical Information Systems*, pages 168–180. Chichester: Wiley.