INTUITIVE VISUALISATION OF MULTI-VARIATE DATA SETS USING THE EMPATHIC VISUALISATION ALGORITHM (EVA)

Andreas Loizides

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

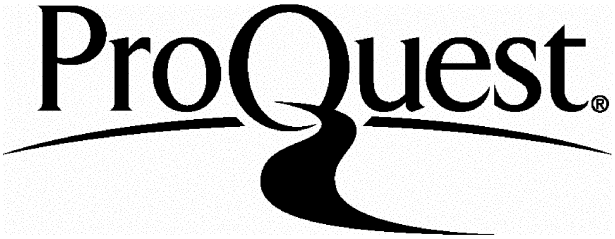
of the

University of London.

Department of Computer Science

University College London

October 5, 2003



To my parents...

Abstract

The central thesis of this research is that there exists an algorithm that can produce a naturalistic visual structure (such as a human face) that represents a multivariate data set that can be utilised to discover hidden features in the data.

Research in this thesis lies in the area of Information Visualisation and is concerned with techniques for visualising large scale multivariate data sets in order to help in understanding and exploration. Examples include financial data, business information and results from experiments. A visualisation method maps such data ideally into intuitive visual structures, however, in most cases there is no obvious mapping from the data to the visual structure.

This thesis explores the use of naturalistic visual structures (for example, human faces) as representations of such multivariate data sets. An automatic mapping from the data to the visual structure is constructed through the use of the empathic visualisation algorithm (EVA) implemented for this purpose. EVA, is a fundamental extension of the type of data visualisation first introduced by Chernoff, who exploited the idea that people are hardwired to understand faces and therefore can quickly interpret information encoded into facial features. We use faces as our paradigmatic example, but the method is not limited to this case only.

Given an $n \times k$ data matrix of n observations on k variables, the original Chernoff method assigns each variable to correspond to a particular facial feature like shape of the nose, or shape of the eyes. The mapping from data to visual structure is arbitrary, and the resulting faces have no correspondence to the underlying semantics of the data. Such faces are good for understanding pattern, but any individual face seen in isolation does not readily convey anything about the data without knowledge of the specific mapping used.

EVA provides an automatic mapping from semantically important features of the data to emotionally or perceptually significant features of the corresponding visual structure, such as a human face. In other words a single glance at the visual structure informs the observer on the global state of the data, since the visual structure has an emotional impact on the observer that is designed to correspond to the impact that would have been generated had the observer been able to analyse the underlying data itself. Finer details concerning interpretation of the

visual structure are then available through knowledge of the relationships between semantically important features of the data and emotionally significant aspects of the visual structure.

EVA uses a Genetic Program (GP) to map the quantitative measurements from the multidimensional data set to the qualitative measurements of the visual structure. The genetic program typically converges after about 75 generations. The experimental data supports the main thesis of this research.

Acknowledgements

The work presented in this thesis would not have been possible without the inspiration of my supervisor Prof Mel Slater. I owe special thanks to Mel Slater for his creative guidance in what turned out to be a fruitful area of Information Visualisation. His assistance, support and patience have been invaluable throughout my years as a Phd student. Moreover, he was willingly subjected to several early, very rough, versions of this thesis and his productive comments and suggestions have undoubtedly improved things.

I am extremely indebted to many of my colleagues and friends who have read and provided feedback on this thesis. Many special thanks go to Evelina Georgiades for spending long hours reading and suggesting changes to my thesis. She has been a true angel. There are no words to thank her enough. I am also very grateful to Dr Celine Loscos and Dr Yiorgos Chrysanthou for giving me the permission to use and abuse their office for several months and especially Celine for lending me her laptop to work with. I would also like to thank Dr Peter Bentley, Dr Anthony Steed, Dr Bill Langdon and Prof Bernard Buxton for their positive comments and feedback.

Special thanks goes to NCR Financial Services, and more specifically, to the Knowledge Lab of NCR for awarding me a fellowship that funded a significant portion of my graduate education.

I am very grateful to Prof Bob Spence of Imperial College London, for willingly agreeing to read my thesis despite his busy time. Also I am grateful to Prof Ben Shneiderman, Prof Alf Inselberger, Prof Rao Ramana and Prof Steve Feiner for granting me permission to use images from earlier research work they performed in the field of Information Visualisation. The images have been used as examples in the literature review.

I am also grateful to my friends who have been there for me whenever I needed them. They accepted my sometimes moody behaviour, that were not used to, and my long spells of going missing. I will avoid mentioning names in case I forget someone. I am sure they know who they are. Special gratitude also goes to someone special. I wish things would have been different.

Lastly, but most important, I thank my family and especially my parents for making this

long journey possible. I am deeply indebted to them. Their advice, support, vision and encouragement have been invaluable throughout the course of my education and more generally my life. My mother Maro and father Michalakis never forgot to remind me of my priorities. I would like to thank my sister Stephania for being a role model in my education life. Finally, special thanks go to my aunt Stephi Stephanidou and uncle Spyros Stamatiou.

"Graphical Excellence is that which gives the viewer the greatest number of ideas in the **shortest time** with the least ink in the smallest space" Edward R. Tufte.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The central thesis of this research is that there exists an algorithm that can produce a naturalistic visual structure that represents a multivariate data set that can be utilised to discover hidden features in the data. The precise meaning of the terms used here will be set out in the remainder of this chapter. Suffice it to say that a naturalistic visual structure is one that needs no special knowledge or skill to interpret – for example, the emotions expressed in a human face. Following this example, the user can understand features of the data by interpreting the emotional expressions in the face, and relating these emotions to the user's own value system or set of meanings attributed to aspects of the underlying data.

Advances in science and commerce have often been characterised by inventions that allow people to see old things in new ways. Computers allow these inventions to be presented in a visual medium, resulting in the emerging field of *information visualisation.* By taking advantage of the processing speed and graphical capabilities of computers, information visualisation enables users to interpret large amounts of information to reveal structure, extract meaning, and navigate large and complex information worlds.

Visualisation can be split into a number of stages starting from the collection and storage of data itself. This can be followed by the preprocessing step, designed to transform the data into something meaningful. Continuing is the display hardware and the graphic algorithms that produce an image on the screen, to finally, the observer and more specifically, the human perceptual and cognitive system. Moreover, visualisation can be approached in many ways [War00]. It can be studied from the artistic perspective of graphic design, within computer graphics for the creation of novel algorithms to display data, as part of semiotics and the constructivist approach to symbol systems or as a scientific approach based on perception.

Furthermore, there are a number of issues that information visualisation techniques try to address [Spe01], depending on the visualisation stage to which they belong. For example, the issue of the selection of data available or the representation of abstract quantities (e.g. colour,

shape). Also the issue of presentation, since usually the number of elements to be displayed from the data naturally takes a larger space than the area of the screen. Moreover, there is the issue of the scale and dimensionality, and more importantly, how scale influences the design of the visualisation tool and what techniques are available for handling high dimensionality. In addition to the above, there is the issue of externalisation, the visual presentation the user sees which is crucial to the success of visualisation, the issue of trying to understand and assist the creation of the internal model in the mind of the observer, and finally the invention of new techniques using experience and skill.

This project describes research that investigates the use of naturalistic visual structures for representation of multivariate data sets with the aim of allowing better and faster understanding of data sets. It is a new algorithm that is part of the preprocessing stage that transforms the data into something users can understand. Additionally, the Empathic Visualisation Algorithm (EVA), the visualisation method that is the result of this thesis, addresses the problems of scale and dimensionality, as well as the problem of externalisation.

1.1 Thesis Terminology

There are a number of fundamental terms, used throughout this thesis. These terms are defined in a number of ways in the data visualisation literature, hence, this section is concerned with explaining them. Most of these terms are inherited from the literature, adjusted to serve the purposes of this research work.

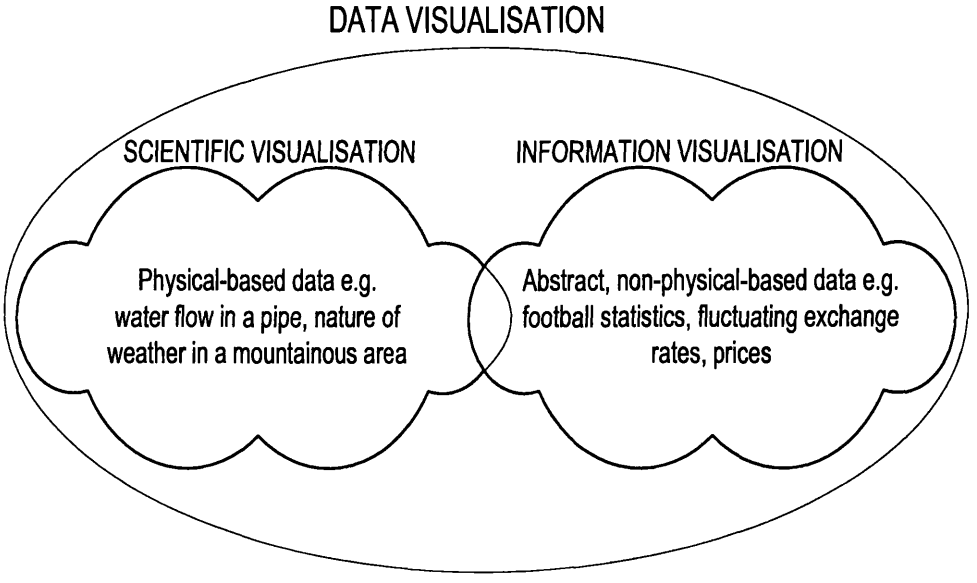


Figure 1.1: Information Visualisation vs Scientific Visualisation.

Figure 1.1, shows the placement of both Information Visualisation and Scientific Visualisation with regards to the broader area of **Data Visualisation** (the term Visualisation is used in this thesis as a synonym to Data Visualisation). **Scientific Visualisation** tends to represent visually aspects of the natural world that have a "physical" representation e.g. natural phenomena and the flow of water in a pipe, whereas, **Information Visualisation** tends to deal with abstract quantities such as financial data and baseball scores. These abstract quantities, although they are also associated with real physical things, are far more important than their view. For example, water flow in a pipe is usually best displayed in the immediate context of the pipe itself whereas the view of a football provides little benefit when visualising football statistics and therefore does not have a natural **spatial mapping**. From the same figure, it is obvious that the two sub-areas of Data Visualisation overlap. However, this thesis concentrates solely on the area of Information Visualisation. Formal definitions of the above terms are given in the background Chapter 2.

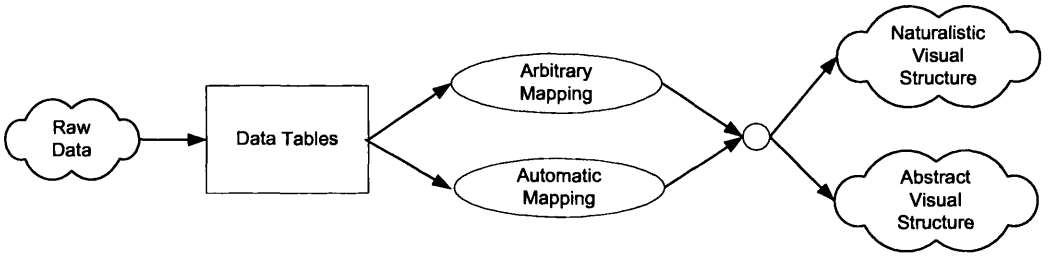


Figure 1.2: Placement of different terms used in this thesis.

Additional terms used throughout this thesis, are shown in Figure 1.2. The Figure shows their placement in regard to the visualisation stage each of the term belongs.

To begin with, **Raw Data** is a collection of abstract quantities. It is usually the product of research, but, the representation in this form is not in an adequate form for communication. To have informational value it must be organised, transformed and presented in a way that gives it informational meaning. It is important here to be able to distinguish between data and information. **Data** are numbers or other methods of recording quantifiable properties in a form that can be assessed by a human or (especially) become input to a computer. **Information** on the other hand, is the derivation of understanding or insight from the data that is not apparent, and we attempt to facilitate this by means of visualisation tools.

The collection of Raw Data, is then transformed into Data Tables. **Data Tables** are data that can be expressed as an $n \times k$ data matrix of n observations over k quantifiable variables. For example, n can be different companies and k can be their "profit and loss account" variables

like *Long Term Liabilities* and *Profit after Tax*.

In Visualisation, Data Tables are then transformed into visual form, a process that is called **mapping**. The mapping can be either abritrary or automatic. **Arbitrary mappings**, are hand-crafted transformations constructed manually by a designer or a user using appropriate tools. **Automatic mappings** on the other hand, are transformations composed on the fly by the underlying system according to a set of pre-supplied layout rules and components.

These transformations whether arbitrary or automatic, lead to a pictorial medium of representation called the **visual structure**. These visual structures can be seen as graphical marks, which are visible things that occur in a space and usually have these four elementary types: Points (0D), Lines (1D), Areas (2D) and Volumes (3D). As with the mapping above, the visual structures can also be subdivided further into abstract and naturalistic visual structures. **Abstract** visual structures include colour, position, size, shape and glyphs whereas **Naturalistic** representations are based on entities encountered in everyday life that need no special knowledge for understanding by a normal human observer. An example of such visual structure is the human face. Everyone is an expert in interpreting this kind of entity.

The visual structure is the tool for communication in Information Visualisation. It is used for understanding compact representations of information. This understanding can unfold through **View Transformations**, which is the ability to interactively modify and augment visual structures, turning them from static presentations into interactive visualisations. These transformations can be achieved using **viewing controls** such as *zoom*, *pan* or *distortion* which is commonly known as focus plus context technique in the Information Visualisation literature.

Finally, there is the end **user**, whether expert or non expert. This is the **observer** interested at interpreting the data and consequently interested in what information the visualisation can convey for scientific or business purposes, or whatever the domain of the raw data.

Having defined the thesis terminology, the following section explains what led to research in Information Visualisation and more specifically the visualisation of Data Tables in the form of multivariate data sets.

1.2 Motivation

The most interesting data are multi-dimensional. How can this be represented on a graph? Only a relatively small class of problems are directly amenable to statistical analysis – those where the variables are understood, potential relationships are already suspected, and there are hypotheses to be tested. Numerous techniques have been described in the literature that attempt to visualise such data with their advantages and disadvantages.

Here we consider cases of data where even the questions to be posed on the data are not fully formulated. A classic example, is that of financial stock exchange data over a number of years - the movement of share prices, for example. Here the question might be *"What determines the changes in share prices?"* Such a vague question is not yet ready for statistical analysis - there are a huge number of potential variables (economic indicators, social conditions, the interrelationships of world stock exchange movement) and the amount of data itself is large (over how many years?).

In such situations, information visualisation helps in "understanding" a set of data, which may be dynamically unfolding in time, by allowing users to visualise representations of the data, thus using vision to build "understanding", and allowing the formation of hypothesis for later statistical analysis.

There are two main problems involved in this. Firstly, it is the choice of a suitable mapping from the data to the chosen visual representation, specifically the difference between arbitrary and automatic mapping. Investigating the possibilities of automatic mapping that take into account the impact of the visualisation on users' emotions, is at the heart of this thesis. Secondly, it is the choice of a suitable paradigm for the representation of the data (in the stock exchange example: portray shares as wheat, the height of the wheat as the price, the wind as the "economic climate", and so on). In other words, the choice of a suitable paradigm, whether abstract, or more realistic. The possibilities of naturalistic (realistic) representations as a standalone application and also used together with different representations is also the focus of this thesis.

1.3 The Goals of this Thesis

The investigation of an automatic mapping paradigm leading to a naturalistic visual representation is the problem tackled in this thesis through the *"Empathic Visualisation Algorithm (EVA)"*. Figure 1.3 shows the placement of EVA in visualisation terms. The x-axis specifies the interest at multivariate data sets, the y-axis the interest at naturalistic visual structures and finally, the z-axis specifies interest at automating the mapping from data to the visual presentation.

Instead of processing individual details in the data set and having numbers and text as output, we examine the use of the visual system to process visual structures holistically and, thus, obtain an overall global view of a data set. Information is gained from an overall view of qualitative measurements. It allows the examination of the important features of the visual structures and the noticing of abnormalities very quickly. It can be viewed as the process of "using vision to think", or the "mind's eye" [CMS99] to gain insight into what are often complex abstract data systems.

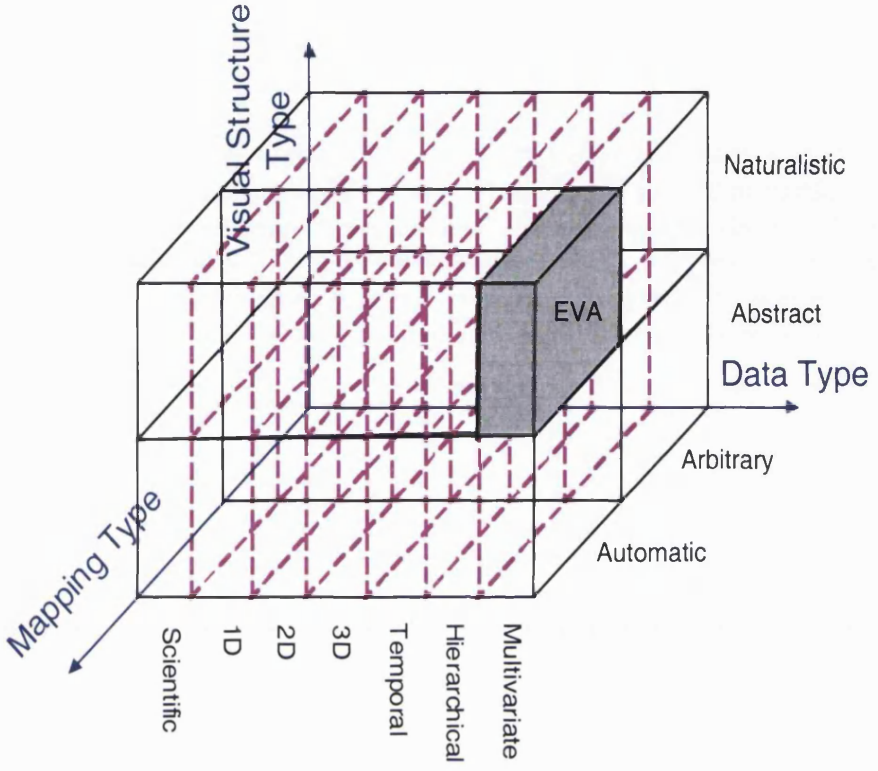


Figure 1.3: Placement of EVA in research interest terms.

The method automatically maps data to visual structures using genetic programming techniques [Koz92], [Sla99], [LS01]. It is called, **Empathic Visualisation Algorithm (EVA)** since the impact of the visual structure on the user's emotions, is taken into consideration. The objective of EVA is, given a data set and an observer, to construct a visualisation such that salient features of the data can be intuitively recognised by the observer. In other words, humans can detect patterns that reveal the underlying structure in the data more readily than a direct analysis of the numbers would. In order to achieve this we use visual structures that are *naturalistic* in the sense that no special knowledge for interpretation is required, and also the mapping from data to the visual structure is *automated*. The mapping should be such that the "important" features of the data set are mapped to features of the visual structure that are "important" and significant to human perception or human emotions.

EVA, the method introduced here to visualise complex data sets, can be thought of as the initial step in establishing a clearer pictorial representation of a problem. It is by no means an alternative to statistical analysis, but a complementary one. It can be seen as the first step in unveiling understanding of the data.

The purpose of the thesis is to construct a system for such a representation, and then test

this in an experimental setting. The system should be such that it can be used with as many different data sets as possible – i.e. that is a generic rather than tied to a particular type of data.

1.4 Scope of this Thesis

For this thesis we are interested in abstract data that has the following attributes: the amount of data is large, multidimensional, of non-physical nature and has hidden information, often in the form of complex relationships among the data variables. Visualisations from abstractions of physical nature (earth, molecules) can also be derived, but since such data is inherently geometrical, appropriate visualisations are ready to hand.

Furthermore, the data must be able to be expressed as Data Tables, and the user is able to itemise interesting functions over the data variables. Examples include, financial data, business information, results from experiments and other abstract conceptions. Such information has no obvious spatial mapping.

On the other hand, the generated visual structure should be naturalistic. A human face was chosen as an example of such naturalistic visual structure and is used throughout this thesis. It is something encountered in everyday life and something that needs no special knowledge for interpretation by a normal human observer. However, it is the only one example and the method is not limited to this.

Moreover, meaningful mappings from data to visual structure that are automated, are of central importance to this thesis. Arbitrary mappings however, are believed to be the barrier to a generalisable system and are not included in the aims of this thesis. Firstly, arbitrary mapping widens the gap between experts and non-expert users of the visualisation, due to the learning time needed. Secondly and more importantly, we believe that, presentations from such types form a barrier to "holistic understanding" of the data set.

EVA, the method used throughout this thesis, requires a search technique to find an appropriate mapping from data to visual structure. We have chosen to use genetic programming (GP) but we do not carry out research in the GP domain. However, since the validity of the method depends upon the validity of this search technique for this problem, an investigation of the GP literature will be considered in detail in Chapter 5.

1.5 Contributions

The overall contribution of the thesis can be summarised as follows: *that a meaningful, automated mapping, from multi-dimensional abstract data to naturalistic visual structures is achievable and provides a useful way to visualise abstract data.* More specifically, the contributions

are:

1. A comprehensive survey of information visualisation research to date.
2. The formalisation and implementation of a technique to automate mapping from data to visual structures, using genetic programming.
3. The use of naturalistic visual cues as the representation of this data in this automated mapping. The emotional expressions presented as part of this visual structure could correspond to the emotions of the users if they analysed the data themselves and knew the results.
4. A method to automatically quantify emotional expressions of a specific facial model based on movements of a number of "landmarks" on the face and users' subjective measurements.
5. Proof of concept through experimental data. A series of experiments were carried out using EVA upon known simulated data, real data and already statistically analysed data providing evidence of the utility of the method.

1.6 Thesis Outline

In Chapter 2 a critical review of previous related work in the field of Information Visualisation is presented, focusing on multi-dimensional visualisation. Advantages and disadvantages of previous systems are discussed. Discussion leads us to argue that a certain sub-area in multi-dimensional visualisation is unexplored.

Chapter 3 presents the methodology. It is a formal description of EVA, presenting the fundamentals of the algorithm.

Chapter 4 is concerned with human facial expressions. A face is considered as the epitome of naturalistic visual structures for the purposes of this thesis. A brief description of the facial model used is followed by a user experiment that results in a novel technique to automate the quantification of facial expressions.

In Chapter 5 a brief literature survey on genetic programs (GP) is presented, a technique that is fundamental to the method used in this research work.

Chapter 6 presents a series of experiments performed on the method. A detail explanation of the results achieved is given in this chapter.

Finally, Chapter 7 discusses conclusions arrived from this thesis, together with future work.

Chapter 2

Background

Information Visualisation is all about gaining insight into data by observing a visual presentation of it. Existing Information Visualisation tools are used for drug discovery by pharmaceutical researchers and credit card fraud detection by financial analysts [CEWB97]. This visual data exploration compliments the algorithmic approaches for exploring worlds of data. Surprising patterns that appear in data sets can sometimes be found by algorithms, but visual presentations can lead to deeper understanding and novel hypotheses. The more common applications of information visualisation are for decision making. These might be for personal tasks such as finding a house [WS92] or choosing a film [AS94b], or for business decisions such as finding a stock in which to invest [MB93]. Information visualisation also serves to explain processes in ways that may lead to better predictions or to more informative insights, which can become a basis for action. For example, visualisation of data access patterns on the World Wide Web may explain why congestion occurs in the early afternoon at a given server [HDWB95].

It is important to distinguish between information visualisation that represents the core of this thesis and data mining which complements this research work. Data mining, is an analysis technique based on statistics and machine learning. Researchers in this field, believe that statistical algorithms and machine learning can be relied on to find interesting patterns, while information visualisation researchers believe in the importance of giving users a visual overview and insight into data distributions. Visual presentations can give users a richer "sense" of what is happening in the data and suggest possible directions for further (statistical) analysis and exploration.

The focus of this thesis is on information visualisation techniques enabling visualisation of abstract multivariate data sets. The visualisation tool used throughout the thesis, is the Empathetic Visualisation Algorithm (EVA) which is a technique that automatically maps multivariate data sets into naturalistic visual structures taking into consideration the impact of the visual structure on the emotions of the observer. The reason for this choice will become apparent as

the existing, different methods are described. It is always important to overview previous work, hence work related to Information Visualisation and especially visualisation of multivariate data sets will be discussed in detail.

2.1 Information Visualisation

Information Visualisation is a relatively new research area that focuses on the use of visualisation techniques to help people understand and analyse data. Spence [Spe01], regards Information Visualisation as about the following: "You are the owner of some numerical data you feel is hiding some fundamental relation which you can exploit to your advantage. You then glance at some visual presentation and exclaim: A ha! Now I understand". Information Visualisation can be defined as the process of transforming data, information and knowledge, into visual form making use of the human natural visual capabilities [GEC98]. Research in decision analysis, cognitive psychology and computer graphics [HE99], [RPR96] conclude that the human mind assimilates information more efficiently in a pictorial form than in raw data form, i.e., in numeric and alphanumeric form. Apparently, the human brain possesses a "narrow bandwidth" for processing raw numbers as opposed to a surprisingly "wide bandwidth" for processing visual data.

Jern [Jer99] explains the role of visualisation as being the interface between what machines are good at (data, information) and what humans are good at (knowledge, experience). Figure 2.1, provides a graphical description of human versus computer abilities. Visualisation can be seen as a powerful link between these two influential information processing systems. Hamming [Ham73] correctly identified that "the purpose of computation is insight, not numbers." Likewise, for visualisation, "the purpose is insight, not pictures" [CMS99].

2.1.1 Overview

In our everyday life external aids are often used to enhance cognitive abilities. These are physical representations of abstract information in what can be called "external cognition" [SR96]. For example, if we want to perform a complicated division, we use notation to store the results of each stage of the calculation. This simple example helps to extend working memory. External representations can also be used to allow patterns, clusters, relationships, cluster gaps and outliers of the data to become apparent. Alternatively, they can be used for quick searches in vast amounts of data for something specific, by giving an overview and powerful navigation. As Norman [Nor93] says "The power of the unaided mind is highly rated. The real power comes from devising external aids that enhance cognitive abilities. How have we increased memory,

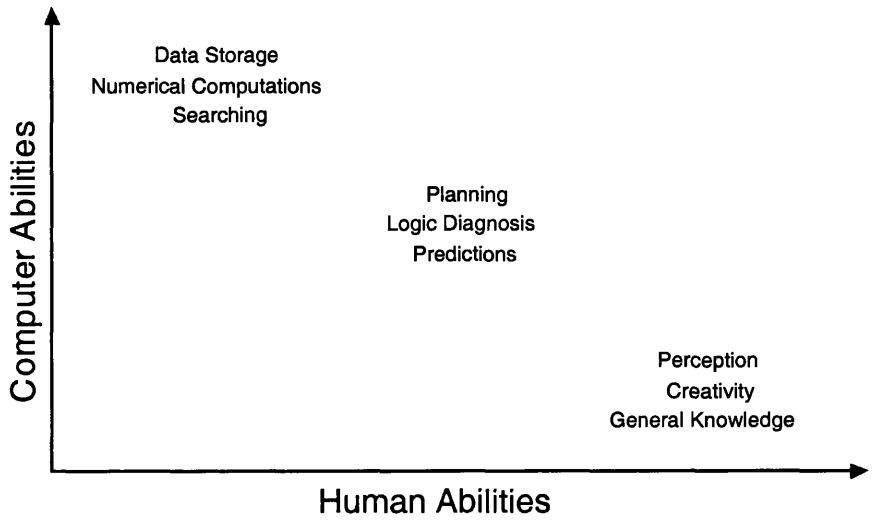


Figure 2.1: Human versus Computer Abilities [Kei97].

thought and reasoning? By the invention of external aids: It is things that make us smart." Information Visualisation is just about that - exploiting the dynamic, interactive, inexpensive medium of graphical computers to devise new external aids enhancing cognitive abilities. These visual artifacts have profound effects on peoples' abilities to gather information, compute it, understand it and create new knowledge.

2.1.2 How Visualisation Amplifies Cognition?

Card, Mackinlay and Schneiderman [CMS99] stress that Information Visualisation has, from its definition, three goals: to aid discovery, assist exploration and help with decision making. In other words, Information Visualisation aids in *Explorative analysis*, *Confirmative analysis* and *Presentation*. *Exploratory analysis* for which EVA is most suited, begins with no preconceived hypotheses about the data, and progresses to the process of searching and analysing the data to find implicit, but potentially useful information. *Confirmative analysis* begins with some hypotheses about the underlying data. The process involves examination of the hypotheses, and hence, their confirmation or rejection. Finally in *Presentation*, facts to be presented are fixed in advance. The outcome is a high quality visualisation of the data presenting the facts.

So, what are the tools that help humans discover, explore and make decisions? How does information visualisation amplify cognition in the users/observers?

Larkin and Simon [LS87] compared solving physics problems using diagrams versus using non-diagrammatic representations. They concluded that diagrams helped in three basic ways. Firstly, by grouping all the information used, large searches were avoided. Secondly, by using a location to group information relating to a single element, the need to match symbolic labels

is avoided, leading to reductions in search and working memory. Finally, the visual representation automatically supports a large number of perceptual inferences that are extremely easy for humans to comprehend. For example, with a diagram, geometric elements like interior angles can be immediately and obviously recognised.

Importantly, in order to understand the effectiveness of information visualisation, we need to understand what it does to the cost structure of a task. In addition to Larkin and Simon mentioned above, Card, Mackinlay and Schneiderman [CMS99] proposed a number of ways in which visualisation can amplify cognition.

To begin with, visualisation can be thought of as increasing the memory and processing resources available to the users. This is achieved by offloading cognitive work to the human visual perception system. Furthermore, visualisation can help to reduce the search for information. The visual representation enhances the detection of patterns. Additionally, visualisation can further assist cognition by enabling perceptual inference operations and by using perceptual attention mechanisms for monitoring. Finally, they acknowledged that, the encoding of information in a manipulable medium helps understanding, usually unfolding through interaction.

It can be concluded that Information Visualisation has a lot of potential to alter the way in which we interact with the multitude of information around us. It has already delivered important benefits in certain application domains. It is important, however, not to overestimate it; instead users should recognise it as an aid. Like numerous software and technological tools, it is not a solution to all information overload and complex problems. Rather, it is a new medium, which when properly and effectively used can augment our cognitive abilities.

A brief historical perspective in visualisation is presented next, before defining the field in a more formal way.

2.1.3 History

Visualising data, and especially scientific data, is not a new concept. The idea of representing data visually has been around for much longer than computer based visualisation. Legend has it that Archimedes was slain while drawing geometrical figures in the sand. Astronomical charts were produced in the Middle Ages in which there were arrow plots of prevailing winds over the oceans and magnetic charts that included isolines.

In ancient Hellas, distinguished scholars like Euclides and Pythagoras visually presented advances in the field of geometry. This area of mathematics not only benefits greatly by visually communicating its findings, but also its very basis and significance lies in objects naturally existing in two and three-dimensional space. In the middle ages (1570) mathematicians used

paper to construct three dimensional models of these geometries in an attempt to make it easier to communicate information [Tuf90]. In the seventeenth century, Galileo used visual reasoning to support his conclusions about the solar system. Quoting Tufte (1990, p.19) "His argument [Galileo's] unfolds the raw data (what the eye of the forehead registers) into a luminous explanation of mechanism (what the eye of the mind envisions)". What is common in the above cases is that the information visualised has the property of naturally existing on spatial axes, so its visual mapping is straightforward.

The challenge of Information Visualisation however, is the understanding of non-physical information through visualisation. Such data, usually, does not have a natural mapping to spatial axes. Ground breaking advances related to this area occurred in the nineteenth century with the linking of the spread of cholera to water supply in central London. During the 1853-54 cholera outbreak in London, Dr John Snow, a physician, identified a large grouping in the Soho area. He went on to plot the homes of the 500 victims who died in the first 10 days of September 1854 on a map of the area, from abstract data such as name, age and addresses of the victims. This simple representation of the data he had collected showed that the grouping of cholera sufferers in this area was centered around a particular water pump. Investigation of this water pump established that it had been contaminated by a leaking cesspool.

In addition to the above example, there are a number of cases where visualisation was used to enhance understanding, well before Information Visualisation was established as a scientific field. Namely, Minard's map of Napoleon's march to and the subsequent retreat from Moscow [Tuf83], Florence Nightingale's diagram showing dramatic reduction of the death rates in a British Army hospital in Grimea, Sir Edward Playfair's circles showing gross national product and tax gathered for certain empires, and Harry Beck's London Underground map [Spe01]. Through these examples, it is evident that the role of visual perception in data understanding, has been long understood.

2.1.4 Definitions and Classification

Computer visualisation has been with us almost since the first digital computers. The 1980s however saw fundamental changes, due to the need for more complex visualisation algorithms and tools to cope with the large amounts of data that sensors and supercomputers supplied. Scientific Visualisation's birth as a discipline is generally placed with the publication of the 1987 Report of the National Science Foundation's (NSF) Advisory Panel on Graphics, Image Processing, and Workstations. The report used the term "Visualisation in Scientific Computing" (ViSC), now generally shortened to "scientific visualisation". The term scientific visualisation

in this context is preferred to the more general term "data visualisation" (see Figure 1.1), due to the fact that the latter has connotations of statistical methods that were outside the scope at that time. Since then, scientific visualisation has experienced vast growth and emerged as a recognised discipline. Visualisations from this discipline show abstractions, but the abstractions are based on physical space. What is seen primarily relates to and represents something that is visually "physical". The data is scientific and the use of visual images for such information has great benefits in enhancing cognition, given that the data maps naturally to the spatial axes.

Information visualisation on the other hand, uses graphic images to represent abstract, non-physical data. Examples of such data include financial data, business information, collections of documents, traffic flows through the internet, statements in a computer program, purchasers at a grocery store, and other abstract conceptions. In certain cases of the above data, graph theory [BM76] has a significant role. The applicability of graph visualisation can be understood better when considering the following question: "is there an inherent relation among the data elements to be visualised?". If the answer to the above question is "yes", then data is "structured" and can be represented by the nodes of a graph, with the edges representing relations. Visualisation of structured data sets is relatively straight forward. However, this thesis is concerned with "unstructured" data elements where the goal of information visualisation is to aid in the discovery of the relations among data variables through visual means. For such data there is no built-in relation between the data elements, and hence, the answer to the above question is negative.

Visually presenting this kind of information poses great challenges, since there is no natural mapping to the spatial axes and, thus, no right or wrong metaphor for representing it. There is a great deal of such abstract information in the contemporary world and its mass and complexity are a problem, motivating attempts to extend visualisation into the realm of the abstract [CRM91]. A more detailed description of the difference between scientific and information visualisation can be found in a discussion by Gershon and Eick [GE97].

Card, Mackinlay and Schneiderman [CMS99], define information visualisation as: *"The use of computer-supported, interactive visual representations of abstract data to amplify cognition."* Figure 2.2 shows the *Information Reference Model* presented in their book. The different stages and transformations presented in their model, have already been discussed in the context of this thesis, in Section 1.1.

Table 2.1 shows a number of definitions, most of which are also adopted from Card's book, that clarify the relationships among concepts related to information visualisation. Some of these terms have already been explained, although here are formally defined.

External Cognition is concerned with the interaction of cognitive representations and pro-

Definitions	
External Cognition	Use of external world to amplify cognition
Information Design	Design of external representations to amplify cognition
Data Graphics	Use of abstract, nonrepresentational visual representations of data to amplify cognition
Visualisation	Use of computer-supported, interactive visual representations of data to amplify cognition
Scientific Visualisation	Use of interactive visual representations of scientific data, typically physically based, to amplify cognition
Information Visualisation	Use of interactive visual representations of abstract, nonphysically based data to amplify cognition
Data Mining	Use of statistical methods and machine learning to identify previously unsuspected relationships

Table 2.1: Definitions/Concepts related to information visualisation.

cesses across the internal/external bounder in order to support thinking. **Information design** is the explicit attempt to design external representations to better acquire or use the knowledge. **Data graphics** is the design of visual but abstract representations of data for this purpose. **Visualisation** uses the computer for data graphics. **Scientific Visualisation** is visualisation applied to scientific data and **Information Visualisation** is visualisation applied to abstract data. Finally, **Data Mining** is concerned with the use of statistical methods and machine learning to help in exploration, analysis and decision making.

It is important to note that while emphasising visualisation, the general term is *perceptualisation.* It is possible to design systems for information sonification or tactilisation of data and there are advantages in doing so [CMS99]. However, the visual system is a pattern seeker of enormous power and subtlery. The eye and the visual cortex of the brain form a powerful processor with by far the largest bandwidth [War00] (in fact half of the neurons in the human brain are dedicated to vision). Therefore visualisation is an obvious place to start.

To put Information Visualisation into context, the different types of research that are performed in this area will be classified. Information Visualisation research can be categorised by data type and by techniques used. Most classification attempts so far have been data centric. This is very useful because implementers can quickly identify various techniques that can be applied to their domain of interest. A very common taxonomy is that first proposed by Schneiderman [Shn96], called OLIVE, with seven data types: 1-, 2-, 3- dimensional data, temporal

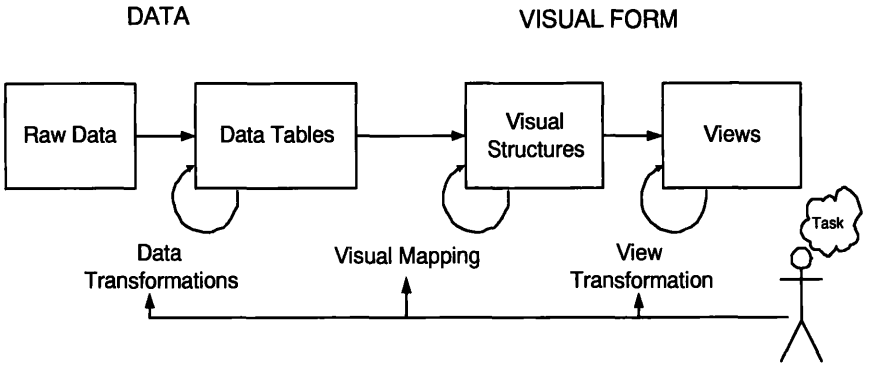


Figure 2.2: Information Reference Model [used by permission of Ben Shneiderman, HCIL].

and multi-dimensional data, tree and network data and seven tasks: overview, zoom, filter, details-on-demand, relate, history and extract. Young [You96] proposed a different taxonomy based on visualisation techniques: surface plots, cityscapes, fish-eye views, benediktine space, perspective walls, cone trees and cam trees, sphere visualisation, rooms, emotional icons, self-organising graphs, spatial arrangement of data and the information cube. Card and Mackinlay proposed an expanded data-oriented taxonomy [CMS99], which divides the field of visualisation into the following subcategories: Scientific Visualisation, GIS, Multi-dimensional Plots, Multi-dimensional Tables, Information Landscapes and Spaces, Node and Link, Trees and Text Transforms. The most comprehensive taxonomy to date, is the one proposed by Ed Chi [Chi00] using the Data State Model [CR98]. Information Visualisation techniques are broken down, not only based on their data type, but also by their processing operating steps.

Figure 2.3 shows a taxonomy adjusted for the purposes of this thesis, influenced by the above related work. Starting from the actual data, its type, whether physically based or not, classifies the visualisation as either Scientific (the left hand branch of the Figure) or Information Visualisation (the right hand branch). In the latter case, Information Visualisation can be subdivided further according to whether the data is structured or unstructured. This thesis is primarily concerned with the visualisation of unstructured, non-physical data, and more specifically with the area of multivariate visualisation. However, a brief description of each of the sub-areas of Information Visualisation divided by data type, will be given below. This way, it is likely to give a better overview of the whole field.

Linear data types (1D) include sequential lists which are often text based. Interface design issues include what fonts, colour, size to use and what overview, scrolling or selection methods can be used. Tasks include, traversing long lists with changeable sort orders, filtering out unwanted data, viewing summary data about many ordered items, and finding important specific

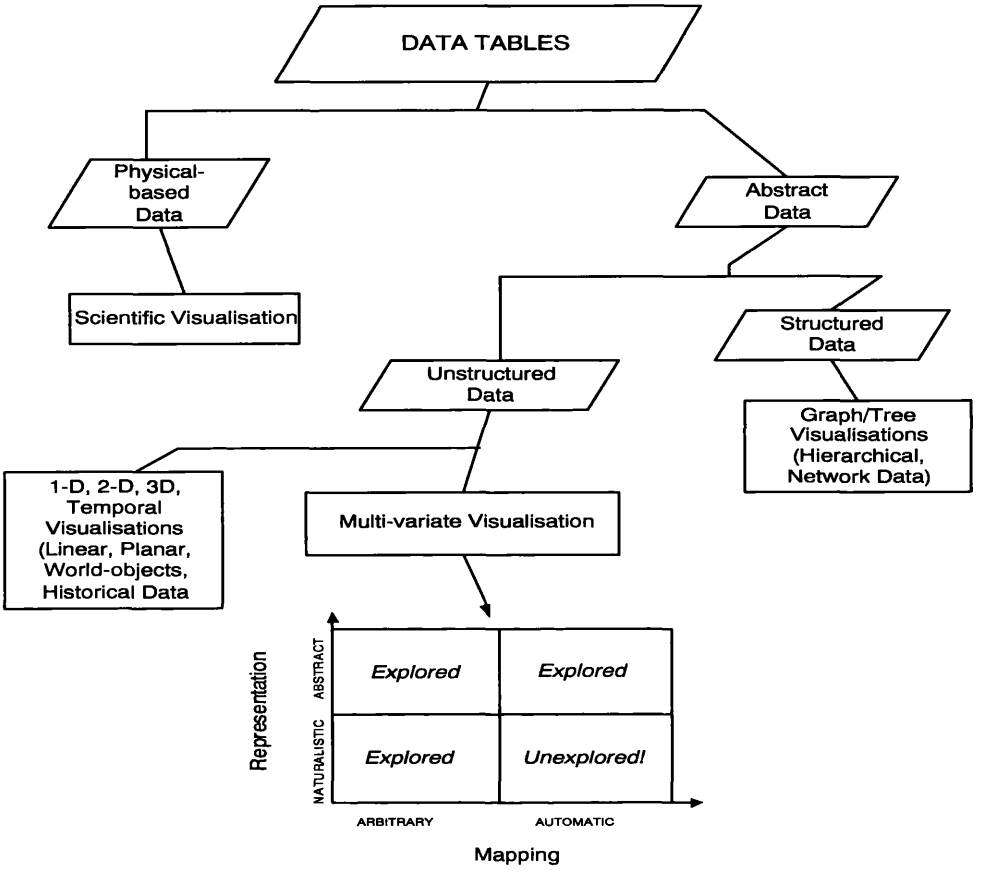


Figure 2.3: Classification of Data Visualisation.

elements.

Planar or Map data (2D) include geographic maps, floorplans, or newspaper layouts. The users' tasks include finding adjacent items, containment of one item by another, paths between items, and the basic tasks of counting, filtering and obtaining details on-demand.

Real world visualisation (3D) is used to view real world objects such as the human body, buildings, or molecules for the purpose of extracting information. Volume visualisation is the form most widely used in world Visualisation and has significant impact in medicine.

The use of temporal information visualisation has a fundamental quality that separates it from 1-dimensional data. The distinction in temporal data is that items have a start and finish time and that items may overlap. Frequent tasks include viewing and creating historical overviews of events or data, and viewing events or data in sequence.

Hierarchical and Network visualisations are seen as a promising medium for information searching. Using tree structures (hierarchies or custom types of networks), the information in digital libraries, documents and the internet can be catalogued and searched quicker and easier, compared to the use of conventional techniques.

Multi-dimensional information visualisation represents data that is not primarily spatial. The number of attributes of a given item in the collection is more than three. Tasks include understanding, or getting an overview of the whole or a part of the n-dimensional data. For example, finding patterns, relationships, clusters, gaps and outliers in the data. Other tasks include finding a specific item in the data. For example, zooming, filtering and selecting a group or a single item from the data. As Multi-dimensional information visualisation is the focus of this thesis a more detailed background review on this area follows.

2.2 Multivariate Information Visualisation

The challenge of visualising multivariate data is ongoing, motivated by many situations in which the interrelationships between many variables are of vital interest. For some examples, an experiment is run and feedback for two-three variables is obtained and analysed, using spreadsheets and graphs. For such problems, these kind of visualisations are very effective indeed. But, what about cases where there are 5, 10, 20, 70 variables?

More than three dimensions requires the more difficult problem of multidimensional visualisations, where data tables have so many variables that an orthogonal Visual Structure is not sufficient. This is the case for most visualisations and they are the most interesting. They start with multivariate data sets that have too many variables to be encoded directly using 1-, 2- or 3-dimensional visual structures. For data like this, graphs and charts lose their effectiveness.

Variables/attributes of such multivariate data sets can be divided into three basic types as shown in Figure 2.4.

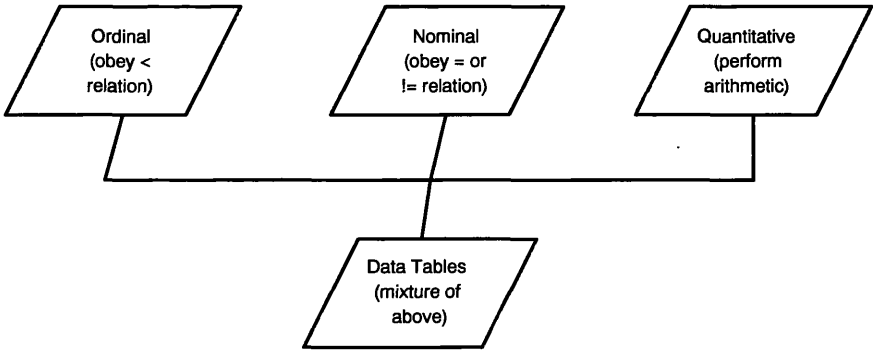


Figure 2.4: Collection of Raw Data.

A *nominal variable* N is an unordered set, such as car models (BMW, FIAT, HONDA). An *ordinal variable* O is a tuple (ordered set), such as film ratings $< G, PG, PG-13, R >$. A *quantitative variable* Q is a numeric range such as weight [35, 210]. These distinctions are very important for some of the systems that will be described below, since they can determine

the type of axis that should be used to represent them, in a visual structure.

The thesis mainly concentrates on quantitative variables of multivariate data sets. Of course, simple rules could be provided that would take Nominal and Ordinal variables and quantify them but this is not relevant to this research. For the purposes of this thesis their existence is assumed and concentration is drawn on data tables of quantifiable variables, irrespectively of the variables' initial types.

Furthermore, there are a number of techniques in statistics, such as *Principal Component Analysis* (PCA), *Factor Analysis*, *Subsetting*, *Segmentation*, *Aggregation* etc, that are used prior to analysis, as a form of data reduction. Despite the fact that these techniques are sometimes very useful, they are beyond the scope of the research area in this thesis and therefore will be discussed no further here.

For the purposes of this thesis, research carried out in the visualisation of multivariate data sets is divided into classes, according to the mapping that was used, whether arbitrary or automatic and, according to the visual structure used to represent the data, whether abstract or naturalistic. As already mentioned in Section 1.1, arbitrary, handcrafted visualisations are constructed manually by a designer or end-user, whereas automatic visualisations are composed on the fly by a system according to a set of pre-supplied layout rules and components. Abstract visual structures include colour, position, size, shape and glyphs whereas naturalistic representations are based on entities encountered in everyday life that need no special knowledge for understanding by an average human observer.

Both of these decisions, mapping type and visual structure type, lie at the heart of EVA, the method proposed for the purposes of this thesis. These distinctions resulted in four subfields of multivariate information visualisation, as shown at the bottom of Figure 2.3, and will be looked at in more detail.

2.2.1 Abstract Visual Structures - Arbitrary Mapping

There are a number of issues that this area attempts to address. How does scale influence the design of the visualisation tool? How many features can be incorporated in the abstract visual structure? What techniques are available for handling high dimensionality?

Presented below, are numerous techniques and tools that are widely used today, that solve, or attempt to solve, some of the problems of visualisating multidimensional data sets.

The very first technique to be described, is that of **multiple views**. The idea is to give each variable its own display. So, if we have n dimensions, n variables, we could have n bar charts, one for each of the variables. In a way the dimensions are broken down to individual

components, that can be visualised in 1-dimension. For some data, multiple view analysis may be utilised, but for others it has a major drawback. Unfortunately, is easy to get lost in the details and hard, if not impossible, to find multivariate trends in the data as well as interrelationships such as correlations.

Consider financial information systems which are multivariate and hence multidimensional. The data components are correlated and their values (or range) affect each other with respect to the decision analysis process. Smith and Taffler [ST96] define that, for such correlated data: "their assessment depends on the simultaneous effect of several variables in different spheres of activity". As an example, the turnover of a firm is, usually, highly correlated with the sales level. When viewing the whole information system and analysing it with respect to another variable, capital expenditure, the former figures introduce a new dimension. From the above, it can be argued that financial information systems are complex. In fact Lux [Lux97] presents the argument that financial information can be, figuratively speaking, described as an iceberg. One can see the tip of the iceberg but its sheer mass is hidden under underwater. Financial information systems are not the only ones that can be described as an iceberg; there are data warehouses, business information, library databases, documents and others that could be described in the same way. Therefore, their highly correlated variables makes multiple views unsuitable for these kind of data.

Bertin believed in two choices for the 'problem' of multiple variables: construction of several images and sacrifice of the overall relationship, or construction of a matrix and discovery of relationships through permutations. He developed a direct technique for creating multidimensional visual structures from multivariate data tables which he called **permutation matrices** [Ber81]. The technique that was developed before computers were used to support visual thinking, involves representing rows of data as bar charts and sorting them. Graphical icons of data values were placed on cards and permuted with metal rods. The goal of the permutations is to form patterns, typically to place the large values on the diagonal of the matrix, thereby clustering similar cases with their representative variables.

When computers became available, permutation matrices became an example of information visualisation. An example is the **TableLens** [RC95], which is a data analysis application intended to give non-experts the ability to visually spot trends and correlations in the data set. An example of TableLens can be find in Figure 2.5.

TableLens, a focus and context visualisation tool, starts with a regular data table and displays the whole table graphically. In essence it is the graphical equivalent of a relational table in which the rows represent cases and the columns represent variables. TableLens is best used for

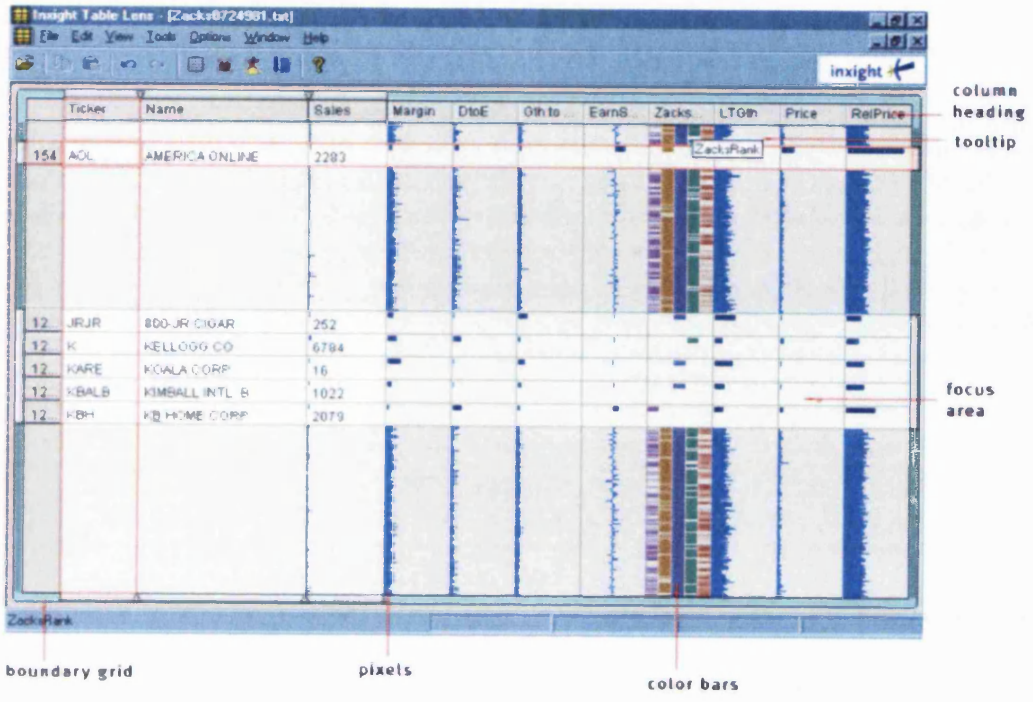


Figure 2.5: TableLens [used by permission of R. Rao, Xerox Parc]

numerical and categorical data. For quantitative variables, a graphical bar is used to represent the values. The bars are aligned to the left edge, that may represent a minimum value, zero or a lower boundary. The length of the bar indicates the relative size of the represented value. This visualisation provides a scale advantage, since bars can be scaled to one pixel wide without perturbing relative comparisons, and also an exploration advantage, since large numbers of tiny bars can be scanned much more quickly than a bunch of textually represented numbers.

There are a wide range of manipulators one can use to discover trends in the data set without affecting the underlying data. One can sort any column, break down the display by categories, focus on any of the rows, or columns, or even spotlight rows i.e. change their colour. Moreover, one can filter categories and create a new column computed using a formula based on other columns. Using these manipulators it is possible to search for patterns and outliers in a multivariate data set. For example, sorting can be seen as the first step of looking for correlations among variables. After a variable has been sorted, if another variable is correlated with it, then its values will also appear sorted.

A similar approach is that taken by Becker and Chambers [BC84] with the **Splus** system. Splus is an interactive statistical analysis environment with similarities to TableLens, in that it integrates several data manipulation and viewing techniques as a library of primitive functions

that can be performed on the data. In particular, after the data is loaded, a user can invoke a "brush tool", commonly known as a scatterplot matrix which displays a matrix of all pairwise scatter plots. Optionally, a histogram of each variable could be placed at the base of each column of scatterplots associated with that variable. A series of other manipulators is used in the tool to help understand more about the data.

There are a number of key problems associated with this method, that are similar to those of TableLens. First of all, as the dimensionality increases, the complexity of these systems increases as well, showing a heavy load of information on the screen. In addition to the above, considerable user learning time is required to understand and enable manipulation of these interactive mediums effectively. Also, it is very hard to visualise relationships of variables beyond second order that involves more than two of them. What if the relationship is a multidimensional one; one that encompasses many or even all of the variables in the data set? Finally, as a conclusion to the previous problem, there are no means to get a good holistic overview of such a data set.

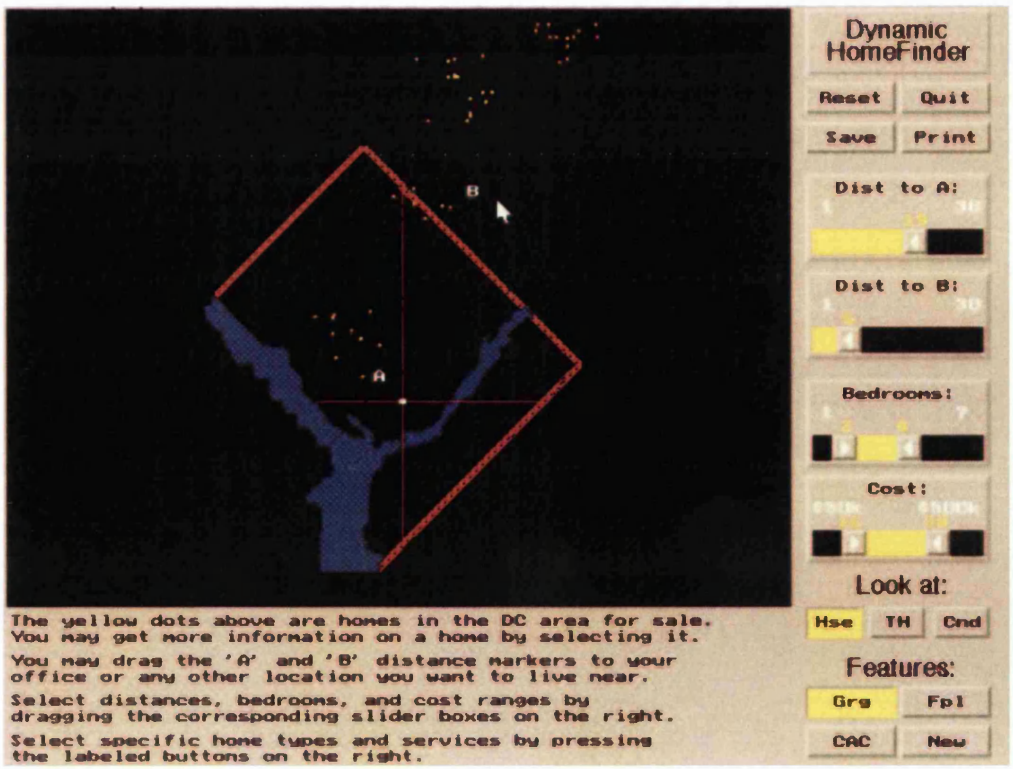


Figure 2.6: Homefinder [used by permission of Ben Shneiderman, HCIL]

Late visualisations like **HomeFinder** [WS92] and **FilmFinder** [AS94b] deal with some of these problems. An example of the Homefinder can be found in Figure 2.6. The interface

representation is a 2-dimensional scatterdiagram that offers *dynamic queries* [Shn93] for interactive user-controlled visualisations of additional dimensions. Dynamic queries allow users to formulate queries by adjusting graphical widgets, such as *alphaslider* first proposed by Osada [OLS93] and described in detail by Ahlberg [AS94a]. By using alphasliders one can see the results immediately. For example in the Filmfinder, colour represents the type of the film, horizontal position the year, vertical position the duration and sliders can be used for other attributes such as type, director, actor for the purposes of filtering.

By providing a graphical visualisation of the database and search results, users can easily find trends and exceptions. User testing was carried out with eighteen undergraduate students who performed significantly faster using a dynamic queries interface compared to both a natural language system and paper printouts. The interfaces were used to explore a real-estate database in search of finding homes meeting specific search criteria. However, in these systems we believe that only a small number of independent variables must be of significant importance to the user in order for the visualisation to work effectively. Also the nature of the data must be of great importance. Data is disclosed only when it satisfies a query.

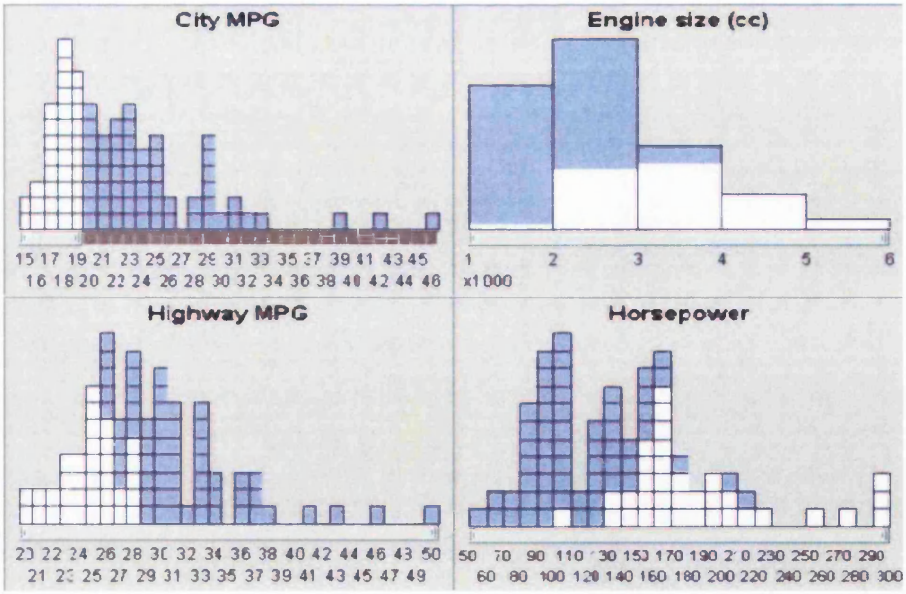


Figure 2.7: Attribute Explorer [used by permission of Bob Spence, Imperial College]

On the other hand, the **Attribute Explorer** [TSWB94], shown in Figure 2.7, uses direct manipulation of a series of histograms to tackle situations in which the display of all data, provides contextual information. Hence, there is valuable guidance for exploration. In Attribute Explorer, each attribute is assigned to a scale with the population spread running up on one side.

Initially each item in the total population is displayed. Users can interact with the scales using sliders for continuous attributes and buttons for discrete (i.e. type of house). The effect an attribute has on other attributes can be explored by selecting values of interest on one scale and viewing where these items appear on the other attribute scales. This is a technique called "brushing" and is achieved by colouring matching objects.

A different technique for representing multivariate data sets is through iconic displays, more commonly known as glyphs. A glyph is a single graphical object that represents a multivariate data object. To create a glyph, multiple data attributes are mapped in a systematic way to show the different aspects of the appearance of the graphical object. For example, a marketing specialist may have data for every person in a particular geographical area, including their income, educational level, employment category and location of residence. In this example a glyph might be used to represent income to the size of the glyph, educational level to its colour, employment category to its shape and geographical location to the (x, y) location where the glyph is plotted.

One such display of multivariate data sets, is using **stick figure** icons [PG88]. This technique is intended to make use of the user's low level perceptual processes, such as perception of texture, colour, and motion. In this visualisation, two attributes of the data are mapped to the display axes and the remaining attributes are mapped to the angle and length of the limbs. An example of stick figures can be seen in Figure 2.8. The goal and hope of this technique, is that users will try to make sense of the data presented through texture patterns created.

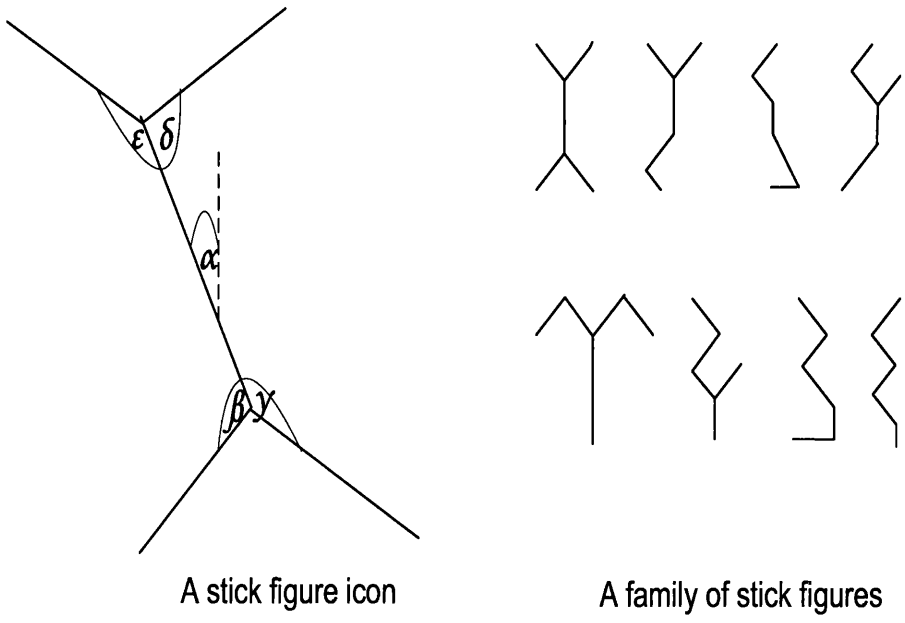


Figure 2.8: Stick Figures.

Another technique for solving multidimensional data problems, through iconic displays, is that of **starplots** [Fie79]. Figure 2.9 shows an example of such a technique with five different variables. The variables are spaced out at equal angles around a circle where each spoke encodes a variables' value (line size). It can be seen even from this simple diagram that as the dimensionality increases (e.g. bigger than 10) the available space will be narrowed making the visualisation hard to read. For small sets it is a good method for comparison of the values of the variables. Alternatively, a number of objects, each represented by a star plot can be compared, at least qualitatively on the basis of their shape. A **whisker** plot, is the same as a starplot without the ends of the lines connected.

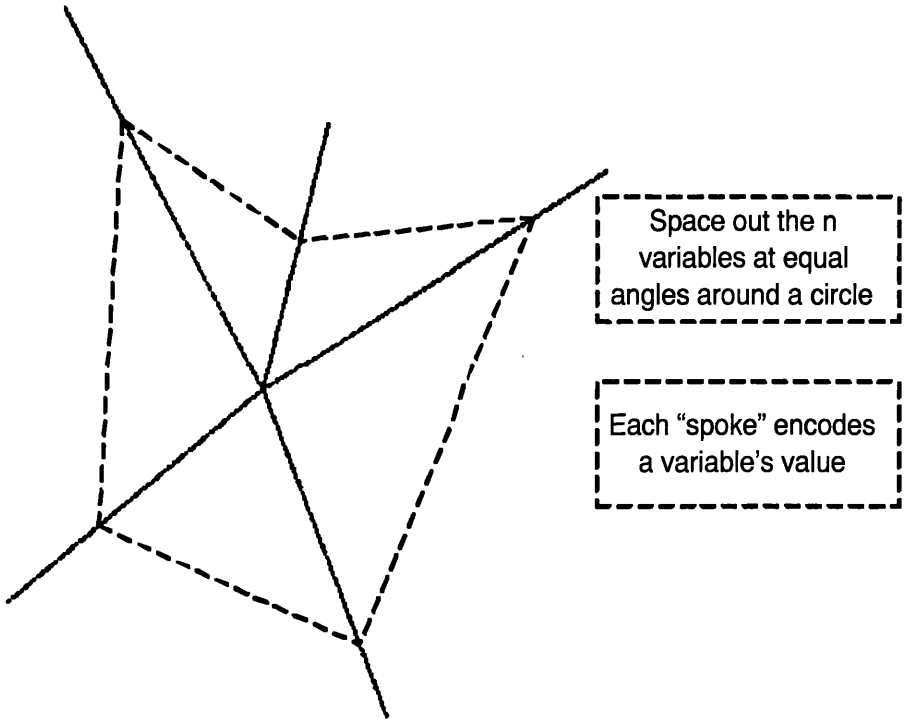


Figure 2.9: Starplot technique.

A similar technique to starplots is that of **Parallel Coordinates** [Ins84]. It is a very popular technique that involves the parallel placement of axes in 2D. The fact that orthogonality "uses-up" the plane very fast and also that parallelism does not require a notion of angle led to the inspiration of parallel coordinates. In drawing each axis separately the technique is reminiscent of Bertin's *permutation matrices*. The axes are scaled to the range (minimum, maximum) of the corresponding attribute. Each case in the data set is encoded as a polygonal line, each line intersects each of the axes at the point which corresponds to the value for that attribute. A line may be colour encoded. Correlated cases often create recognisable patterns between

adjacent axes. The challenge of parallel coordinates is to recognise these patterns. Interactivity is provided in the system to help people find these relationships. Interaction allows the user to reduce the complexity by limiting the range of an axis or brushing specific lines. Therefore users can focus on specific data items. Figure 2.10 shows an example of parallel coordinates. The data set used, is based on the production of VLSI chips for 473 batches as described in the "Multidimensional Detective" [Ins97]. Each data point is for a specific batch.

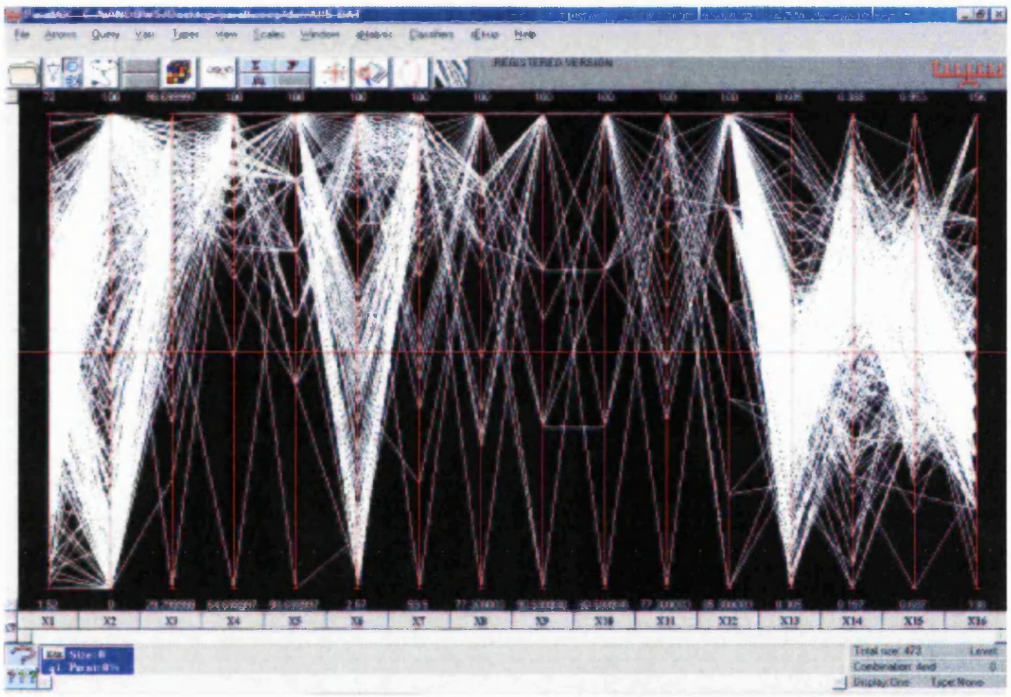


Figure 2.10: Parallel Coordinates technique [used by permission of Alfred Inselberg]

Parallel coordinates can be seen as the method to model relations, or a 2d pattern recognition problem. However, the method requires long learning time and skill from the user in order to gain geometrical understanding and properly query the picture. Also, due to the narrowness of information the colour encoding being performed is unclear, and one does not really see colour. Moreover, the overlap of polygonal lines, can make it difficult to identify structures in the data. Additionally, there is the problem of ordering associated with this technique, since different assignments of the variables to the parallel coordinate axes, might lead to different patterns in the data to become visible/invisible.

Having said that, case studies show that this visualisation in the hands of a skilled user supports complex visual thinking. This conclusion arises from the following observations [Ins97]. The complexity of the visualisation increases linearly with the number of variables. If the num-

ber of dimensions is N then the complexity is O(N). In theory it can represent an unlimited number of dimensions. Every variable is treated uniformly. A line axis corresponds to each variable. This is unlike the many representations mentioned above (if not all), and others that will be mentioned below. The data is encoded in the visualisation without any loss of information and the display, easily conveys information on the properties of the N-dimensional object it represents. Finally, according to the author, the methodology is based on rigorous mathematical and algorithmic results.

A different technique is that proposed by Mihalisin, Timlin and Schwegler [MTS91]. They developed a technique to visualise a scalar dependent variable that is a function of many "independent" variables.

Each variable is plotted within the space delimited by the previous variable, for each discrete value that represents the variables range. More specifically, the working space is divided into multiple windows. Following this, a reasonable set of values is selected for the variable and the window for these values is subdivided further. The process is done recursively until the whole function has been plotted i.e. all the variables have been realised.

It is a hierarchical technique for visualising and visually analysing multivariate functions, data and distributions in various ways. The order in which the variables are placed in the hierarchy, affects the overall visualisation. It is like sampling the axes of the input variables at slow, medium and fast quantities to create a new variable that is a function of the input ones. Some visualisations are comparatively easier to understand than others. Therefore, it gets quite complicated as the dimensionality of the variables increases and as the range of values of your variables increases. A problem that becomes apparent for this method is that of treating the variables non-uniformly.

The technique mentioned above, seems to be influenced by an earlier method. Beshers and Feiner [FB90] described a technique called **"worlds within worlds"**, a multidimensional visual structure based on overloading. They visualise high-dimensional functions by placing 3d coordinate systems inside other 3D coordinate systems recursively until all dimensions are included. Changing the position of the inner coordinate system results in changes in the surface displayed, since three variables are changed. However, at any one time, the surface displayed is constructed out of only 3 variables (the outer coordinate system) and the constant values of the rest of the variables. The main idea of the method (worlds within worlds) is a way to gain information lost in the process of reducing the complexity of the data in order to be displayed in 3-dimensions.

All of the techniques described in this section use abstract visual structures to visualise

multivariate data sets, and the mapping from data to visual structure is an arbitrary one. In fact this category (abstract visual structures, arbitrary mapping) includes the vast majority of the tools created nowadays to visualise data of multiple dimensions. Visualisation systems that attempt to automate the process of mapping are the focus of the next section.

2.2.2 Abstract Visual Structures - Automatic mapping

The problem addressed here, as already mentioned, is that of automatically mapping data to visual structures in a meaningful way. In other words, the problem of automatic design based on some pre-supplied rules. Authors of visualisation tools must represent abstract quantities in some way. Which methods of encoding are useful? How can they be combined?

A wide range of symbolic representations for encoding are available [Tuf83], [Tuf90]. Which is the best? From one point of view the answer is, "it depends on the task", since the task for which the user seeks to form a mental model, the internal model in the mind of the user, can take many different forms and the domain for which the data is encoded can also vary widely. From a different perspective the answer is "we do not know". Various attempts have been made to identify comparative benefits of separately encoding numerical, ordinal and categorical data [Mac86], [CM84] with the view to automate the design of visualisation tools. Jacques Bertin [Ber81] attempted to classify all graphic symbols in terms of how they could express data. It was mostly based on his own judgement, albeit a highly trained and sensitive one.

One of the first attempts for automatic design is Jock Mackinlay's **APT (A presentation tool)** [Mac86] based on a formalisation of Bertin's scheme, composition algebra and artificial intelligence techniques. Mackinlays' goal was to create automatic visualisation from data based on generating and testing possible solutions that satisfy rules of *expressiveness* and *effectiveness*. Expressiveness refers to language's ability to represent the data correctly, whereas, effectiveness is about the psychological performance, how good is the visualisation. The effectiveness criterion is used to rank the primitive languages according to accuracy in perceiving quantitative, qualitative and nominal characteristics of the data. The effectiveness and expressiveness criteria are used in the selection step of APT's matching procedure.

The significance of this research is that it showed the theoretical analysis of graphical presentations were an adequate basis for partial mechanisation. Data are composed to data tables that are mapped to visual structures that are, in turn, composed into complex presentations. For example, two variables from a data table might be mapped to two 1-dimensional visual structures and then composed to create a 2-dimensional visual structure.

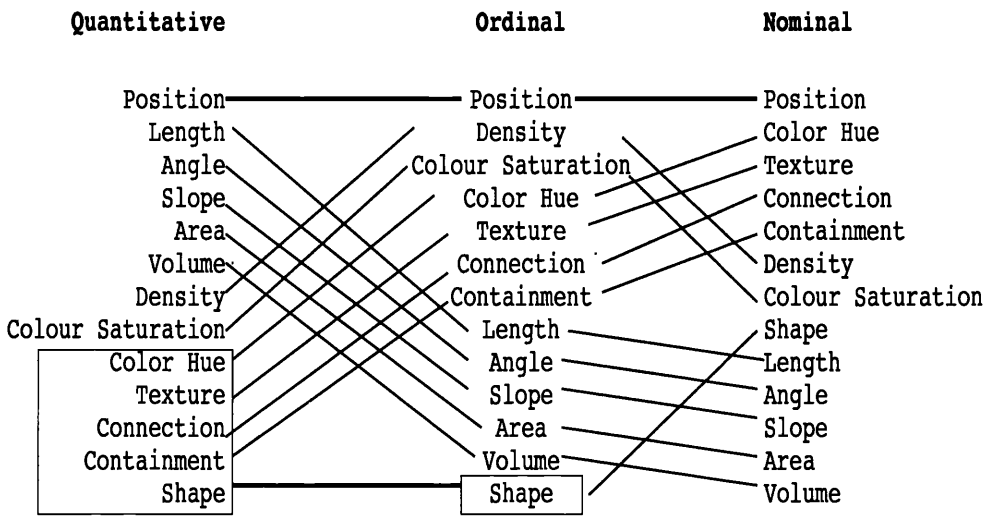


Figure 2.11: Ranking of perceptual tasks. The tasks shown in the boxes are not relevant to these type of data.

Figure 2.11 [Mac86] shows a ranking of the perceptual tasks. It is an extension of Cleveland and McGill's ranking [CM84]. A wide variety of designs can be systematically generated by using composition algebra that composes the small set of primitive graphical languages. The matching procedure uses backtracking when some of the choices made at the various stages do not allow for a feasible design.

It is a very good early attempt for automatic design but it appears not to be a knowledge-based *interactive* data exploration. There must be a way for users to express their needs even if it is only from the point of view of what is important in the data set. It is important that users have a say in how they want this data to be visualised. But where is the interaction coming from? Moreover, the rankings produced have not been empirically validated, in contrast to Cleveland and McGill [CM85], who have empirically validated their ranking of graphical devices for quantitative data. A further study [LNH02], empirically measures the relative effectiveness of colour, shape and size in conveying both nominal and quantitative data. It is interesting to note that the different rankings provided in the literature are conflicting which suggests the need for further research.

A much more interactive system is **SAGE** [RKM+94], a knowledge-based presentation system that automatically designs graphics and also interprets a user's specifications of how to present the data. It is similar to APT, but also allows the user to supply none, some or all of the specifications of the visualisation. It can visually integrate many forms of information, including combination of quantitative data, relational, temporal, hierarchical, geographical, categorical and other. The system addresses the automatic of "visualisation" and "manipulation".

SAGE is the engine that handles the automated creation of images. *IDES* (Interactive Data Exploration System) is used for the data manipulation task. IDES tools include dynamic queries with the use of sliders and aggregate manipulator to aggregate and decompose groups. The latter makes disjunctions possible. *SageBrush* assembles graphics from primitive objects like bars, lines and axes. It is used to search a portfolio for relevant examples.

The combined environment of SageTools (Sage, SageBrush, SageBook), enhance user-directed design by providing automatic presentation capabilities with styles of interaction that support data-graphic design. A user uses "directives" to communicate goals to Sage engine with SageTools (task, style, aesthetic, data).

There are a few shortcomings in the system. Despite the help of dynamic queries, the complexity of the visualisation results in a large learning time. Also the aggregate manipulator is too complex to be used effectively and the total flexibility in interaction allows the user to break the rules.

The visualisation systems described here allow for automatic mapping from data to abstract visual structures based on some criteria. Earlier attempts ignore the support for user interactions whereas most recent attempts tackle the problem of interactivity as well as improving the quality of the visualisation.

However, are we ready to eliminate graphic designers and provide a general purpose, automatic, visualisation tool? We believe that we are not even close to this and that there are other possibilities for exploration in the process for automatic design. One such possibility, is the use of naturalistic representations described below.

2.2.3 Naturalistic Visual Structures - Arbitrary mapping

Some disadvantages of information visualisation systems encountered so far, arise from the fact that visualisations are not natural nor easy to understand and therefore require learning time from the user. Moreover, in some cases the visualisation does not offer a complete, and holistic view of the data set. Effectively, naturalistic visualisations are believed to handle the problems mentioned above.

"Naturalistic" visual structures refer to visual representations of things encountered in everyday life, things that do not require special knowledge for interpretation by a "normal" human observer, preferably irrespective of nationality.

The first to use such a technique was Herman Chernoff [Che71] when he recognised the potential of using a human face as a representation for data. The vast majority of the techniques in this category expand this idea. Chernoff proposed using a mapping of variates in data into

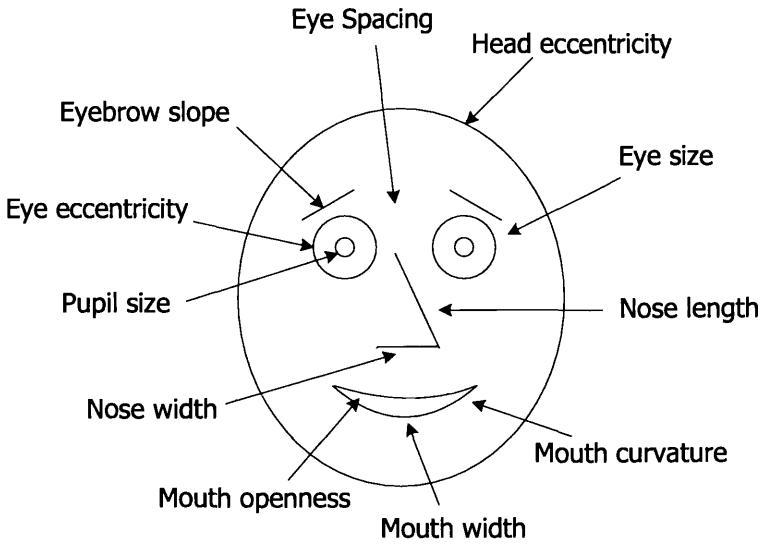


Figure 2.12: An example of a Chernoff face.

features of faces as a method of visualising multi-dimensional data (Figure 2.12). The method is most commonly known as **Chernoff faces**. More examples of Chernoff faces are shown in Figure 2.13. The method involves assigning to each column of the data, a facial feature such as width of the eye, position of the mouth etc., and for each row of the data constructing a face associated with the assignment. It is believed to be able to represent a total of 20 different dimensions as shown in Appendix C.

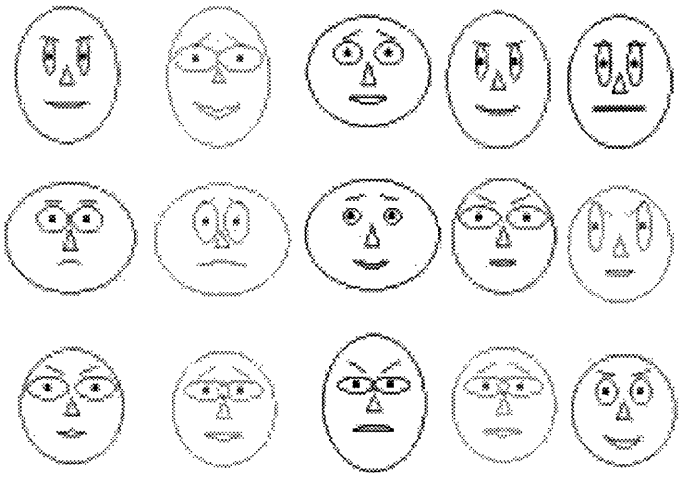


Figure 2.13: Further examples of Chernoff faces.

There are psychological reasons for choosing such a display object. Faces are probably the most important class of objects in the human environment. First of all, is the familiarity with human faces and ability to rapidly process even the smallest nuances and changes in a

human face due to everyday interaction. Secondly, the fact that the human face often evokes an emotional response in us and can therefore affect the way in which we behave.

The hope is that one can use this information to group the data into interesting ways, and to determine an interesting structure of the data. Since humans are optimised in some sense, for face recognition, it was hoped that using faces (as opposed to other object types) would aid in the grouping of the data and also to illustrate "trends" in multi-dimensional data.

Wilkinson has shown that, when considering similarity of data sets, faces prove to be a better representation than other forms [Wil82]. The use of faces as a means for communication has also been shown to lead to fewer mistakes being made and to users becoming more involved in what they are doing [JLR94].

Walker has shown that the use of a face does not introduce an extra element that could distract a user from the task they are trying to achieve, thereby potentially reducing their effectiveness. Instead, it actually increases the impact of the information, while providing a familiar medium for communication.

Another potential advantage that Chernoff does not discuss in great depth is the way in which humans perceive faces. Instead of considering each individual variable as we would have to if we were to look at a table of numbers, work by Homa has shown that humans process facial expression as a whole [HS76]. This means that in addition to the multivariate data being encapsulated in one face, it is also processed as one object when we look at it. The face therefore provides an excellent abstraction of the data.

Many people have built upon Chernoff's initial idea. Shane and Moriarity [Mor79] used Chernoff's idea to investigate whether schematic faces were a useful abstraction of communicating financial information. They concluded that persons with a limited knowledge of financial analysis, as well as practising accountants, were able to distinguish bankrupt and non-bankrupt firms more efficiently than using financial statements or ratios. Smith and Taffler [ST96] showed renewed interest in the area and pointed out that previous work such as Moriarity's failed to compare the performance of users of varying levels of sophistication of subject. They also argued that since no indication is given in previous studies of the statistical nature of the information, the results might be partly due to superior information and not superior means of visualising it. Their study included considerations from psychology literature which previous work had ignored. For example this includes mapping data variables onto the features of the face which psychologists have shown to be of more importance when humans consider facial expressions. Emphasis is placed on the mouth and eye areas which are considered the more important features of the face when conveying information [Lau71]. This is due to the high

amounts of movement seen in these features relative to other parts of the face such as the ears and nose.

Through their own experiments they suggest that users of all levels gave speedier, more accurate results than when using the raw data form, despite the fact that the most specialised group were reluctant to accept that their own results using this technique were superior to standardised decision processes.

Having said that, there are a number of limitations when using Chernoff faces. To begin with, the variables in the representation are treated non-uniformly. In fact it seems obvious that an identification and placement of the important features in the area near the mouth and eyes would make the visualisation more effective. In addition to that, it is obviously necessary to spend some time training test subjects as to which features apply to which variables. Furthermore, the nature of the visualisation implies that one does not see the actual quantitative values of the variables being plotted and, finally, visualisation loses its effectiveness when we have extreme values, since these might produce unrealistic faces.

The latest attempt to use naturalistic visualisations is by Alexa and Muller [AM99] in a technique called **Visualisation by Examples**. Based on the specification of only a few correspondences between data values and visual representations, complex visualisations are produced. The foundation of this approach is the introduction of a multidimensional space of visual representations. The basis of their method is the "morphing" technique (for more about morphing see Alexa and Muller [AM99]). Morphing between two graphical objects results in a one dimensional space. Morphing between an element of such space and a third graphical base object results in a two dimensional space. By repeating this process they construct a space of any given dimension.

This technique is better illustrated by an example. Suppose we want to visualise an overall (scalar) ranking of cities in the USA. We might be interested in finding a nice place to live. The data for this particular example, contains values from nine different categories. That means there is a need to project nine values to one scalar value. To visualise the rankings a Chernoff-like approach is used. They take a smile and a frown and produce a 1-dimensional visual scale. Thus, the degree of smiling represents the living quality determined by a combination of the nine attributes. The way they find the mapping is by allowing the user to supply a ranking based on personal experience. A ranking of a subset of all the cities is sufficient. In this example they mapped Chicago to a smiling face, Miami to a frown and Washington to a neutral face. The result is an image for every city by combining the given example ones.

This method is nice and simple. However, a question to be asked, is that since Visual-

isation by examples only visualises one dimension what is the point of having a naturalistic representation? Techniques like barcharts are very effective at representing such data. If it to be used, then a more sophisticated technique is needed to map the data to the visual structure. The first principal component might be a good candidate for such case.

2.3 Discussion

The relevant work needed to put EVA in context has been critically reviewed. The techniques described in the literature review, capture all kinds of attempts to visualise multiple-dimension data sets of variables. From those we can conclude that there are numerous techniques that use arbitrary mapping and abstract representations, few techniques that use arbitrary mapping and naturalistic representations, few systems that use automatic mapping and abstract representations and *no* techniques that use automatic mapping combined with naturalistic visual structures. Figure 2.3 shows that information visualisation systems that produce "automatic" mappings from multi-dimensional data to "naturalistic" visual structures have not been exploited. It is believed that such systems will realise advantages from both automatic mapping and naturalistic representations mentioned above to produce visualisations of certain qualities.

Firstly, the visual structure will give a holistic overview of the whole data set, allowing the formation of hypotheses about the underlying data. In addition to the above, the nature of the visualisation and mapping used, implies that the visual structure will be very simple, easy to understand and that the learning time of the users will be minimised. Furthermore, the background of users will be irrelevant. Both experts and non-experts will be able to get a quick understanding of the underlying data. Moreover, decisions on the fly may be reached for real time data and finally, it is hoped that all significant relationships between the data variables will be incorporated into the visual structure.

In the next chapter we begin the design of this method in detail.

Chapter 3

EVA Methodology

There are a numbers of problems identified in the previous Chapter that are quite common with existing visualisation techniques. Re-iterating a few, the fact that most visualisation tools need significant user learning time is important. It is known that users prove slow in adapting to new visualisation tools simply because it usually means relearning how to do things. Also, the tools presented in the literature, tend not to have a meaningful way of combining the information and relationships among the data, into one visual structure.

These problems and others already mentioned in Section 2.3, inspired the development of the Empathic Visualisation Algorithm (EVA) [LS02]. The methodology behind this new approach is the focus of this Chapter. An introduction to the problem is presented first.

3.1 Statement of the Problem

The problem presented here is that of visualising multi-dimensional data sets. The numerous techniques which have been described in the literature that attempt to visualise such data have both advantages and disadvantages. However, no method claims to achieve the overall objective of this approach. The overall objective is to construct a visualisation such that the salient features of the data can intuitively be recognised by an observer and where the representation gives a holistic view of the data set, embodying the interest of the user.

Complex data in the sense that it is presented here, is data that is relatively large both in terms of the amount of data present, and the number of variables (dimensionality) that the data encompass. Data tables, have so many variables that an Orthogonal Visual Structure (such as a 3D graph) is not sufficient. Also, the variables themselves are usually correlated and hence cannot be treated separately.

An example of such multi-dimensional data set is that of accounting (financial) data. The data shown in Figure 3.1 represent a Balance sheet and profit and loss accounts for a single company over a period of 5 years. Apart from the fact that there is a great amount of data,

the data components are correlated and their values (or range) affect each other with respect to the decision analysis process. The assessment depends on the simultaneous effect of several of these variables in different spheres of activity. One, must also take into consideration the fact that this is data from a single company. Imagine having hundreds of companies. How is such data to be visualised and understood? The prospect of visualising and understanding data, therefore from multiple companies is, in the least, a daunting task.

Balance Sheet					
Capital & Reserves					
ORDINARY SHARE CAPITAL	131000	131000	132000	141000	141000
SHARE PREMIUM A/C	836000	840000	856000	1441000	1460000
OTHER RESERVES	39000	104000	163000	230000	308000
PROFIT & LOSS A/C	2380000	2629000	2690000	2381000	2644000
EQUITY CAP. AND RESERVES	3386000	3704000	3841000	4193000	4553000
PREFERENCE CAPITAL	0	0	100000	200000	325000
TOT. SHARE CAPITAL & RESERVES	3386000	3704000	3941000	4393000	4878000
Fixed Assets					
INTAGIBLE					
TANGIBLE	509000	534000	585000	715000	698000
INVESTMENTS	25312000	32374000	35298000	39782000	46558000
OTHER	1415000	1433000	2074000	2038000	2908000
	27236000	34341000	37957000	42535000	50164000
Current Assets					
STOCKS	0	0	0	26000	14000
DEBTORS	50857992	53668000	57512992	71361992	86548992
INVESTMENTS	0	0	0	0	0
OTHER	1080000	1620000	1753000	1880000	2023000
CASH	757000	598000	391000	339000	1957000
	52694992	55886000	59656992	73606992	90542992
Current Liabilities					
PROVISION FOR TAX	442000	498000	496000	299000	266000
PROVISION FOR DIVIDENDS	129000	158000	191000	245000	290000
CREDITORS <1 YEAR	55757992	60526000	64129992	72659000	91462992
OTHER	11373000	11372000	15334000	22796000	25594000
	67701992	72554000	80150992	95999000	1.18E+08
Net Current Assets	-15007000	-16668000	-20494000	-22392008	-27070000
Total Asset Less Current Liabilities	12229000	17673000	17463000	20142992	23094000
Long Term Liabilities					
PROVISIONS	296000	399000	630000	970000	1144000
LOAN CAPITAL	8547000	13570000	12892000	14780000	17072000
OTHER					
	8843000	13969000	13522000	15750000	18216000
	3386000	3704000	3941000	4392992	4878000
Profit & Loss					
OPERATING PROFIT-ADJ	-446000	-504000	-1169000	-611000	-394000
TOTAL NON-OPERATING INCOME	1247000	1514000	2337000	2019000	2011000
TOTAL INTEREST CHARGES	66000	80000	134000	167000	193000
PROFIT BEFORE TAX	735000	930000	1034000	1241000	1424000
TAX	310000	319000	337000	403000	480000
PROFIT AFTER TAX	425000	611000	697000	838000	944000
ORDINARY DIVIDENDS	184000	233000	288000	360000	434000
TO SHAREHOLDERS FUNDS	241000	378000	409000	478000	510000

Figure 3.1: Sample of data for a single company.

In situations such as the above, information visualisation helps in "understanding" a set of data (which may be dynamically unfolding in time) by allowing users to visualise representations of the data, thus using vision to build "understanding", and allowing the formation of hypotheses for later statistical analysis.

As already mentioned, there are two main problems involved in this, the first being, the choice of a suitable mapping from the data to the chosen visual representation (structure) and the second being, disinguishing between arbitrary and automatic mapping. The possibility of taking

into account the user's emotions during visualisation in an automatic mapping, is the focal point of this method. The second problem highlighted is the choice of a suitable paradigm for the representation of the data, whether abstract (e.g. using colour) or more realistic. Different representations may lead to quite different understandings. The possibilities of naturalistic representations, is also addressed in the present approach.

Figure 3.2 shows the placement of *"Empathic Visualisation Algorithm (EVA)"* in the information visualisation classification presented in the literature.

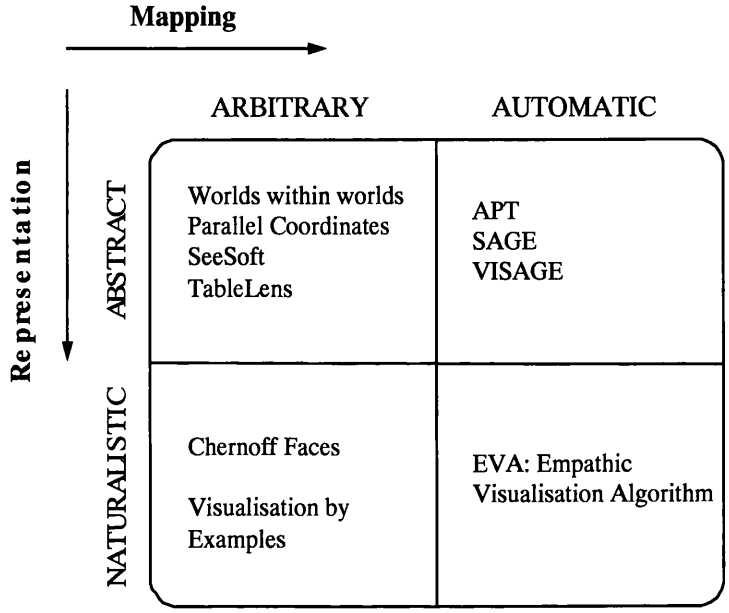


Figure 3.2: Classification of EVA.

The specific aim of this study is the construction of a system for such a representation, and then to test this in an experimental setting. The system should be such that it can be used with as many different data sets and visual structures - i.e. that is, a generic system rather than one tied to a particular form of data or visual structure.

3.2 Fundamentals

Throughout this thesis the representation of multivariate data tables in an $n \times k$ data matrix X, consisting of n cases on k "quantifiable" variables $x_1, x_2, ..., x_k$ is assumed. Each row in the data matrix typically represents an individual and there are k observations made for each individual. As identified above, the objective is to construct a visualisation of the data table such that the salient features of the data can intuitively be recognised by an observer and the representation gives an overall view of the data set. Within this overall objective there are two further fundamental objectives:

1. Naturalistic visual representation. It should be something encountered in everyday life, something that does not require special knowledge for interpretation by a normal human observer.

2. Automatic mapping. The mapping should be that semantically "important" features in the data are mapped to "important" features of the visual structure that are significant to human perception or emotion.

Examples of (1) include faces, buildings, body posture and scenery. No human needs to be an expert to recognise the emotional content of another human face - it is recognisably "happy", "sad", "angry", "relaxed", "scared", "neutral" and in addition various combinations of these basic emotions. Frequently used throughout this thesis, is an example of a "face" because it is the epitome of a naturalistic visual structure in the sense that is presented here.

Given a set of data and a visual structure, it is trivially easy to construct a mapping from the data to a visual structure - for example map variable x_i to the ith facial feature as in Chernoff faces described in the literature. However, such a mapping is arbitrary - it does not take into account the impact of the face on the emotions of the observer. Now the data is of interest to the observer for some reason; associated with the data is some **value system** reflecting the interest, importance or consequences of aspects of the data for the situation of the observer. A fundamental goal, reflected in (2), is that the perceptually or emotionally significant features of the visual structure directly reflect the value system over the data. The term used in this thesis to describe this, is **visual homomorphism**. Hence the mapping from data to visual structure cannot be arbitrary, but must be constructed in such a way that this visual homomorphism is realised.

What follows is an explanation of how such a mapping can be constructed. Two different types of visualisation problem are considered: the first is the representation of the data matrix as a whole - one visual structure representing the entire data matrix, i.e. a face. The second, is where each row (i.e., individual) in the data is to be represented by a different instance of the same type of visual structure - so using faces again, each row is mapped to a different face. In the first case, the problem is to capture overall features of the same data set. In the second case the problem is to examine the differences in the individuals, or to examine one individual changing through time. In fact the method is very similar in both cases and does not affect the computation. The difference is on the focus and interpretation. The overall representation is first considered.

3.3 Assumptions and Notation

Let $\nu_s(X)$, $s = 1, 2, ..., p$ be p functions over the data, potentially all, representing "values" over the data matrix and the interest of the user.

Consider the example where the data table represents a set of customers of a telephone company, and the variables are quantities such as age, gender, marital status, income, number of years with the company, number of telephone calls made per week, monthly phone bill, and so on. One value might be a function of the overall age distribution of the population - such as the mean age, the percentage over 65, or the percentage under 20. Another value might be the "flatness"of the data - for example the ratio of the variance of the first principal component to the total variation in the data. Another value might be the quality of service given by a telephone company and so on, and many other quantities that characterise the interests or "value system" of the observer.

Consider a visual structure (Ω). Similarly there are p important aspects, of Ω, called **characteristics**, that are measurable and significant to human perception or emotions, $e_s(\Omega)$, $s = 1, 2, ..., p$. In the example of the face these might be the degree of anger, happiness, boredom, fear - or characteristics such as age, beauty, gender and so on.

The fundamental goal, in terms of any X and any Ω, is to produce a mapping $\mu(X) \longrightarrow \Omega$ such that the "value system" of the user over the data matrix, is reflected in characteristics of the visual structure. In particular that $e_s(\Omega)$ is a monotonically increasing function of $\nu_s(X)$ for each $s, s = 1, 2, ..., p$. For example, an increase in profitability of a company should result in an increase of happiness of the corresponding visual structure.

A **characteristic** is a measurement of some aspect of Ω as a whole (such as the emotions on a face) rather than some individual **feature** (such as the shape of the mouth). It is some measure representing the totality of the face i.e. the degree of "happiness". The appearance of "happiness" depends on many different individual features of the face - specific configurations of muscle tensions, for example. Similarly, the appearance of beauty, age or gender is derived from many different features - such as size of the eyes, inter-ocular distance, shape of the mouth, symmetry, and so on. In other words, features are the individual components that make up a face - such as the specific configuration of muscle tensions for a specific face, or the geometric and material properties of the actual features like eyes, colour of the eye, mouth, nose and lips. Knowing its features enable us to *render* or produce a displayable image of a face. Once rendered the face will have a set of measurable "characteristics" (qualities).

Suppose that there are r features of the visual structure: $\phi_t(\Omega)$, $t = 1, 2, ..., r$. Knowing these features Ω can be rendered. Once rendered, we can measure it to determine its character-

istics $e_s(\Omega)$, $s = 1, 2, ..., p$.

Finally, we introduce **feature functions** over the data matrix: $f_t(X)$, $t = 1, 2, ..., r$. These functions completely determine the features of the visual structure, in fact

$$\phi_t(\Omega) = f_t(X),\ t = 1, 2, ..., r \tag{3.1}$$

The values of these functions are *interpreted* as the values of the features of the visual structure. The aim is to choose these functions f in order to attain the required correspondence between the value system over the data and the characteristics of the visual structure and therefore, to attain the visual homomorphism.

Let $\nu = (\nu_1, \nu_2, ..., \nu_p)$ and $e = (e_1, e_2, ..., e_p)$. Suppose $\|\nu - e\|$ is a measure of the 'distance' between these two *vectors*. Then the specific goal is to choose $f_t(X)$, $t = 1, 2, ..., r$ such that $\|\nu - e\|$ is minimised. If this is achieved, then the characteristics of the visual structure produced by the *feature functions*, best represent the "value system" of the data set. Figure 3.3, shows the notation diagrammatically. There are a series of transformations taking place bearing in mind the minimisation between the quantitative measurements of the data of interest to the use ν, and the quantifiable qualitative measurements of the visual structure e. More specifically $e = C(B(A(X)))$ and $\nu = D(X)$.

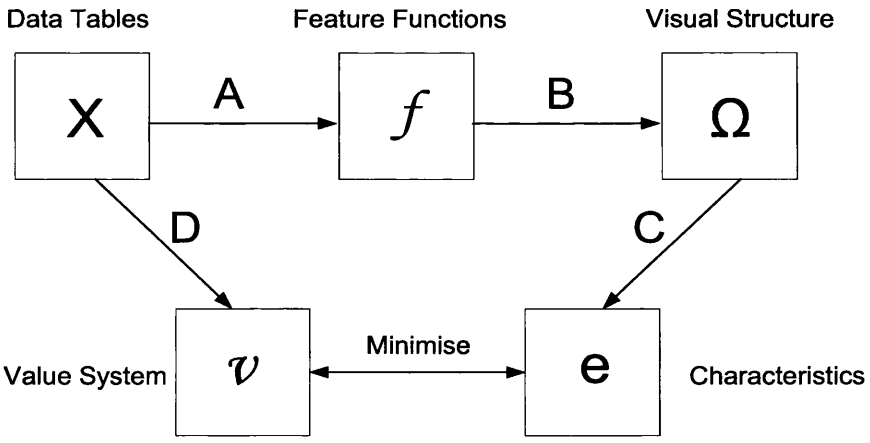


Figure 3.3: Different stages and transformations of EVA

3.3.1 Assumptions and Notation by Example

This Section explains further the notation presented above, through the illustration by a simple example. The data and visual structure chosen for this example, although they do not fall in the appropriate categories as described in Section 3.1, serve the purpose of giving a better understanding of the concepts involved.

Suppose there is a collection of data X, in a data table of size $n \times k$, where $n = 50$ and $k = 7$. Each row (50 of them) in the data set is represented by 7 quantities, 6 of which are known to be input variables $x_1, x_2, x_3, x_4, x_5, x_6$, with the final column x_7 representing the output variable of a function.

Let us assume that what is of interest to the user, is whether the data set originates from the dot product function (in this case $x_7 = x_1 * x_2 + x_3 * x_4 + x_5 * x_6$), either regarded as whole, or in a row by row basis. Therefore, $\nu_1(X) = |x_1 * x_2 + x_3 * x_4 + x_5 * x_6 - x_7|$ is the "value system", the only function in this example, of interest to the user.

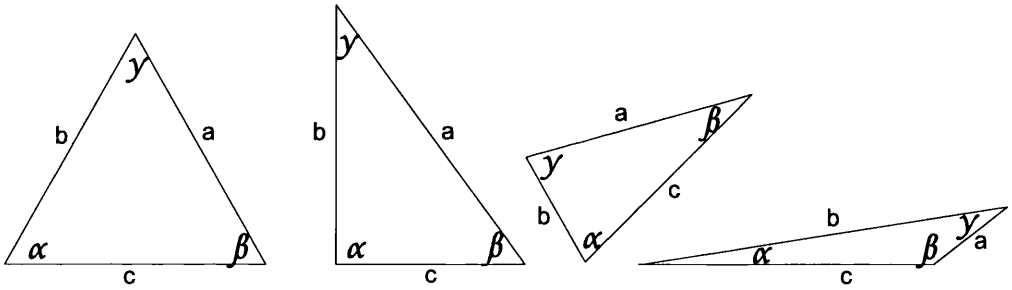


Figure 3.4: Examples of triangles

Also consider a triangle to be the visual structure Ω with sides a, b, c and angles α, β, γ. Although a triangle does not have naturalistic characteristics, as they are defined here, there are a number of qualities that can easily be observed by a user. How close to an equilateral triangle the visualisation is, is the quality aspect chosen (that is significant to human perception), to represent how close to the dot product the data is, whether one deals with the whole data set or is treating the data as individual rows. It is the only characteristic chosen of the selected visual structure, for the purposes of the example. The closer to the dot product the data is the closer to an equilateral triangle the visualisation produced by EVA should be. Hence, $e_1(\Omega) = (60 - \alpha)^2 + (60 - \beta)^2 + (60 - \gamma)^2$ is an obvious candidate for measuring the "closeness to an equilateral triangle".

On the other hand, there are the features of Ω, that allow the visual structure to be rendered. For this example, degrees for at least two of the angles are needed[1]. These two quantities are given from functions over the data. Hence, $\phi_1(\Omega) = \alpha = f_1(X)$ and $\phi_(\Omega) = \beta = f_2(X)$. Without any loss of generality it can be further assumed that these functions return results from $(0to90)$ degrees, with 0 and 90 excluded. Therefore $\gamma = (180 - \alpha - \beta)$ and the visual structure can be constructed.

From the above, in order to attain the visual homomorphism, the right set of functions

[1]For simplicity reasons position of the vertices of the triangle is not important and will be ignored

f_1, f_2 need to be selected, so that $\|\nu_1 - e_1\|$ is minimised. In other words, the set of functions f should be chosen so that the difference between how far the data is from the dot product function, fits closely with how far the triangle is to an equilateral. In such a case, data closely related to the dot product function, either rows or whole data tables, will be mapped to equilateral or close to equilateral triangles whereas for any other data, the visualisation will degrade away from equilateral triangles appropriately.

In Figure 3.4, the perfect equilateral triangle on the left, represents the visualisation of data closely related to the dot product function. The other shapes represent, from left to right, a gradual degradation away from this close fit with the dot product function.

3.3.2 Using Genetic Programming (GP)

The minimisation problem introduced above, that attempts to find the best set of functions to represent the visual structure Ω, can be tackled using Genetic Programming (GP). Let $F_i = (f_{1i}, f_{2i}, ..., f_{ri})$ be a specific set of feature functions which, when applied to X, produce the visual structure features. A large collection of such sets of functions F_i, $i = 1, 2..., N$ is chosen at random. This collection defines a *population* of sets of feature functions. The ith member of the population produces a specific visual structure Ω_i. This visual structure has characteristics $e_s(\Omega_i)$, $s = 1, 2, ..., p$. These characteristics can be used to produce the distance measurement $\|\nu_i - e_i\|$. The distance measurement is directly associated with the "fitness" for the ith member of the population. Hence, each member of the population has an associated fitness, which can be expressed as a probability. An example might be to set minimum fitness of population to 1, maximum fitness to 100 and interpolate for values in between. These probabilities determine survival into the next generation and selection for mating - thus producing a second generation. The process continues until (possible) convergence. The most fit member of all populations is chosen for the required mapping.

It is expected that at each successive generation, the average fitness would increase [Koz92], until a generation is reached such that subsequent iterations produce only negligible increments in fitness. This is interpreted as convergence to a local minimum (solution). Different runs of the GP result in different solutions due to randomness of the technique and absence of a known optimal solution. The most fit member of each population is chosen for the required mapping.

The basic theory of GPs will be described in Chapter 5, to give a better understanding of how this optimisation function is realised in this method. Also that Chapter explains the reasons for choosing such a search technique as opposed to other existing ones.

3.3.2.1 Visualising Individuals in a Data Matrix

The method above produces a visualisation for an entire data matrix. Instead the observer may be interested in visualising each individual (row) of the data matrix - in order to look for "special" individuals (eg. companies to add to an investment portfolio), or where the rows represent the changing of one individual through time, representing its evolution.

The method is fundamentally unchanged, and only involves a reconsideration of the domain of the feature functions f. Previously, the domain of these functions was the whole of the data matrix X. Instead, the domain is now over the variables represented by the columns of the data matrix. So each individual row of the data produces a set of feature function values, which therefore determines (renders) a visual structure for each row.

Similarly, the domain of the value functions ν is restricted also to the variables represented by the columns. Hence each row now produces a distance measure - that between the characteristics of the visual structure for that row, and the values over the data for the row. The overall distance for the ith member of the population of sets of feature functions can therefore be taken as a combination (e.g., a sum) of the distances over all rows of the data. The method then proceeds as before.

3.4 Overview

A summary of the steps required in order to implement the method is described below, using the notation above. Figure 3.5 gives a graphical visualisation of the overview of the method.

1. Decide on Ω, the type of visual structure to be used. Determine the number of features Ω has. Assuming there are r, $f_t(X)$, $t = 1, 2..., r$ feature functions are required in order to render an individual.

2. The user, identifies the p important values, functions over the data set, $\nu_s(X)$, $s = 1, 2, ..., p$.

3. Identify $e_s(\Omega)$, $s = 1, 2, ..., p$ the p characteristics of the visual structure that measure its totality and are significant to human emotions and perception. This can be selected by the user or automatically selected by the system.

4. Identify the fitness function i.e. the function we are trying to minimise, which is, for example:

$$\sum_{d=1}^{n}\sum_{s=1}^{p}(\nu_{ds} - e_{ds})^2 \tag{3.2}$$

5. Identify the GP parameters and run the GP.

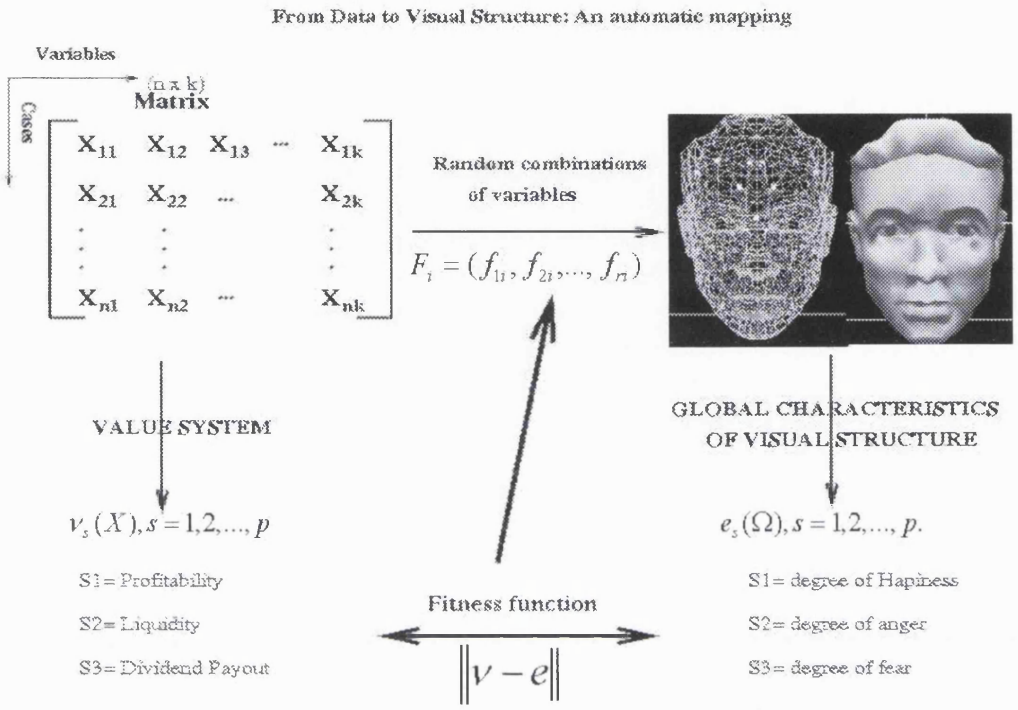


Figure 3.5: Overview of the method.

The steps shown above achieve the visual homomorphism described earlier. In other words, it can be defined as the 'extraction and visualisation of qualitative data from quantitative one'.

3.5 Discussion

This section has presented a method for automatic determination of a mapping from data to visual structure. It requires a user (someone interested in the data) to construct a set of value functions of interest over the data. The designer of the visualisation must decide on a type of visual structure, and a set of perceptually or emotionally significant characteristics of this visual structure that matches the number of value functions inserted by the user. The visual structure must be determined by a set of quantifiable features. Genetic Programming is then used to construct the mapping that minimises some given measure of distance between what the user inserted as value functions over the data and what the designer provided as characteristics of the visual structure. The minimisation criterion is the only factor that specifies the exact nature of the mapping.

For the method to work it is assumed that the GP will converge. In such a case we can hypothesise that the observer will pick up out important features of the data from this mapping.

In order for the method to be of any use it will have to be tested. Before presenting results of testing the method, a description of a particular visual structure, the human face, how we render it as well as how we measure its characteristics will be presented. This will be followed by a review of the genetic programming literature.

Chapter 4

Using Human Faces and Measuring Human Emotions

This chapter describes the use of faces in EVA. As mentioned before, this study regards human faces as the epitome of naturalistic visual structures. In this thesis, based on EVA, multi-variate data sets are mapped into, and therefore represented by, facial models.

What follows is a brief description of the available technologies on facial models, succeeded by a description of the 3D facial model used in this study. It is also, a pre-requisite of EVA to be able to generate, but most importantly, measure synthetic facial expressions automatically. Therefore, details of a new technique that quantifies emotional expressions based on what people say, will also be presented.

4.1 Parameterised vs Muscle Face Models

The human face is the most complex part of the human body when it comes to the number of possible deformations of its surface. For this reason, selecting a suitable surface description was vital. Polygon topology is a rapid method commonly used, as the classical graphics pipeline uses this representation most efficiently.

It is this representation that Parke [Par72] used for developing his first parameterised model, and has been used extensively since then. Although Parke's work may seem dated by todays' standards, his parameterised models still provide an important template for much of the modelling that is carried out today. His technique involves building a set of parameters which describe a human face and it's expression [Par82]. Parke also included conformational parameters such as the size of the head and neck, skin colour and jaw width that all vary from one individual to another and therefore make each of us different. Using simple topology, face images are generated by moving the vertices in line with the parameter sets provided by the user. A series of simple geometric transformations produce the features of the face and interpolation

is used to move between each expression.

Muscle models may also be used. Waters [Wat87] introduced a simple yet effective technique for simulating the muscles of the human face. He considered deforming a polygon mesh using a muscle vector function. His technique regards muscles as single directional vectors, with a "head" that is connected to the bone of the skull, and a "tail" that is connected to the flesh of the face. As a muscle contracts it pulls the skin towards the point at which is connected to the skull. Each muscle has a zone of influence and the vertices of the polygon mesh describing the surface of the face are moved according to the contraction of the muscles. Waters' work has been extended by many currently working in facial animation and more realistic models are being built all the time based on his ideas.

4.2 The Face Model - GEOFACE

The work of Keith Waters from the Digital Research Labs in Cambridge resulted in the original Geoface program, a face-muscle model utility, that is based on the theories of Keith Waters and Fred Parke [PW96]. Their model demonstrates a method of producing expressions on a computer generated face, by using mathematically modelled muscle deformations.

This thesis uses an improved model of the one produced by Frederik Parke and Keith Waters. The muscles used for this improved model can be seen as red lines in Figure 4.3. Appendix B, gives a more detail description of Geoface and the extensions to the current version being used. Figure 4.1 shows a particular face model in different display models, from left to right: Gouroud shading, polygonal, line and point display model.

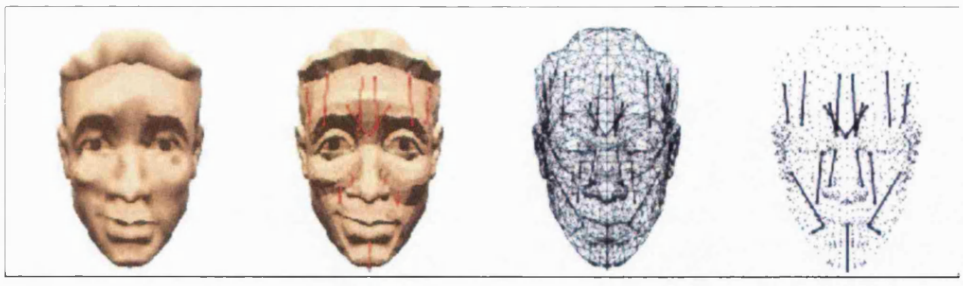


Figure 4.1: Facial geometry of a specific human model.

4.2.1 Generating Facial Expressions

Ekman [EF78] developed the Facial Action Coding System (FACS). This system provides a notation for recording and describing the expressions of the face by considering the combinations of muscles that are used to create them. Ekman [Ekm79] identified six key expressions of

emotion which can be recognised across most cultures. Some can be seen, in the model being used here (Figure 4.2 i.e. "neutral", "anger", "sadness", "happiness", "fear", "surprise" and "disgust".

Figure 4.2: From top left, "happy", "sad", "fear", "angry", "disgust" and "surprise" faces, generated using the improved Geoface model used in this study.

The FACS guide, uses an intentionally literal nomenclature to describe its measurement units, called "action units" (AUs), in order to avoid built-in biases about the possible meanings of facial activity. So, instead of "angry", "sad", "frown" or "smile" the units have names such as "nose wrinkle", "cheek puffer", and "dimpler"(there are 46 such measurement units) most denoting the activity of a single muscle.

FACS uses numerical formulas that break down every facial expression into its basic units: which muscles move, for how long, with what intensity, in what overall pattern. For example the simplest expression is happiness. It can either be broke down to the number "12", representing the "lip corner puller", or the value "6+12" with the value "6" representing the "cheek raiser".

By using FACS, generating facial expressions becomes a simple task. It simply requires assigning contraction values to the muscles of the facial model, taking into consideration the maximum contraction values for producing realistic faces. For example, Ekman describes happiness as a contraction of the left and right zygomatic major muscles (this is one variation of happiness). From inspection, we can set a contraction value for these two muscles which describes what would be considered a perfect smiling face. We can then set those values as limits

of the contraction values for the muscles in question.

Therefore in order to generate a realistic number of random facial expressions, the production of random sets of contraction values (one set for each face) is all that is required. In each set there will be a contraction value for each of the muscles, taking into consideration the range of values each muscle can have.

4.2.2 The Need to Measure Facial Expressions Independently

As mentioned in the previous chapter, EVA requires automatic measurement (quantification) of the emotional expressions of interest. An easy way to measure the emotional expressions would be using Ekman's FACS system. As described above, there would be a muscle set associated with a "perfect" emotion i.e. a muscle set describing what would be considered a "perfect" happy face. This muscle set is then compared with those on a specific generated face and produce a mean square error from the ideal.

However, is it the right thing to do? It will be more appropriate to have user subjective responses on the matter, since it is real people who will draw conclusions from observing the facial models.

Of course having a group of users, during training of the Genetic Program, to evaluate every single face produced, for a number of emotions is a very time consuming task. In the simplest run, there could be 500 faces per generation, with at least 50 generations in total and 3 different emotions. Ideally, a number of user responses is required for each emotion, say 3, with safely speaking 15 seconds needed on average per response. This accumulates to 3,375,000 seconds or 937.5 hours of non-stop work, an impossible task especially when one takes into account that more than one run of the Genetic Program is usually needed.

Having said that, there is a way round this problem, which still relies on users' subjective responses, indirectly. What follows is an explanation, analysis and results from a technique that measures emotional expressions on a specific facial model automatically, based on user's subjective evaluations [LSL01].

4.2.3 Measuring Emotional Expressions

An experimental set-up was created to quantify emotional expressions on the face, for the following emotions: scale of happiness-sadness, scale of anger-calmness and scale of fear. This is achieved based on movement of a number of points ("landmarks") on the face that are significantly influenced by muscle contractions in relation to a stationary landmark. 25 such points are used. These are mainly points near the eyes and the mouth, in addition to a stable point that is used as a reference. This reference point was selected to be a point on the surface of the face

which is not affected by muscle contractions. The number 25, of landmarks on the face, was selected using a trial-error method. More than 25 points were found not to improve the results further. The main criterion for the selection was that the points chosen are within areas affected by muscle contractions. In fact for any single muscle a number of landmarks was assigned in its area of influence. Figure 4.3 shows the position of these points together with the reference point highlighted as small squares. In the same figure the vertices in the area of influence of the muscles are drawn in white. The 25 points lie in this area hence their squares are white, where as the reference point lie outside, hence highlighted in skin colour (bottom right corner of the image).

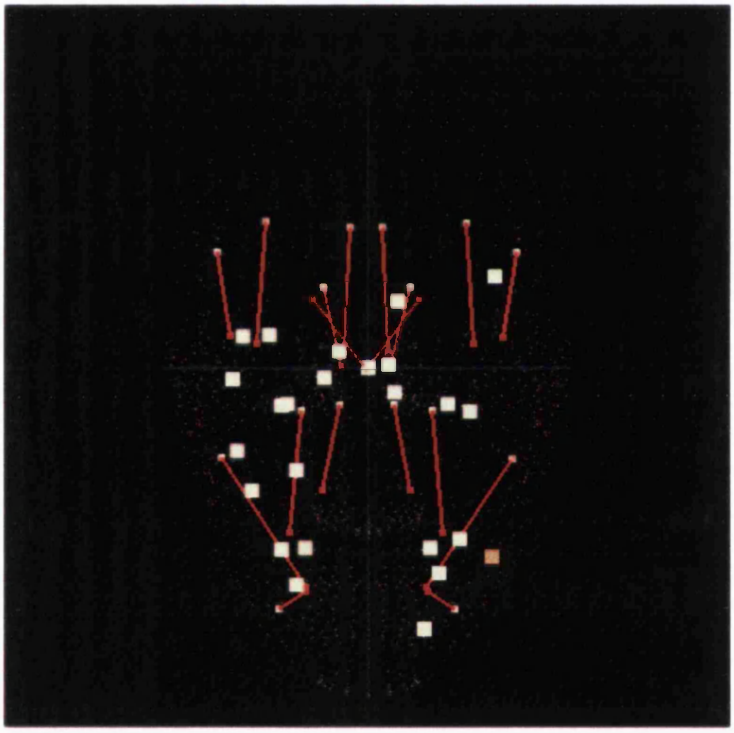


Figure 4.3: The 25 landmarks and reference point chosen. Muscles are shown as red lines.

4.2.4 Experiment and Results

Two sets of data (randomly created facial expressions) were created, the first consisting of 200 faces and the second of 150 faces, and positions of each of the "landmarks", for each face, were recorded with respect to the reference point. Specifically, the distances of the 25 points from the reference point were recorded for each face. Users, postgraduate students at the Computer Science Department of UCL, were asked to assess all of the emotions on 50 different faces. In total, each of the faces was subjectively assessed for emotional state by 3 different people

for each emotion (from a pool of 30 people) for both data sets. The answers of the subjects (the mean for each facial expression) for the first data set only, were used to create symbolic regression equations with a response variable y_i representing the individual emotional expression and explanatory variables x_i, $i = 1, ..., 25$ representing each of the landmarks on the face. This is a process of fitting, for each expression, the best (or one of the best) possible function that best represents the 200 faces[1] lying in a 25 dimensional space. This search space problem was also implemented using a Genetic Program, resulting in a different equation per emotion. These are equations over the distances of the landmark variables influenced by different muscle contractions in relation to the stationary point.

The second data set was then used to verify the equations formed. Highly significant positive correlations (as shown below) were found, between the results from the equations produced by the Genetic Program, and the user's subjective evaluations. This result, allows us to conclude that these estimated regression equations are a good approximation to the user's subjective evaluations of the emotional expressions for this particular synthetic face model.

4.2.4.1 Training data - GP configuration

Objective:	Find a function of one variable y_i and 25 dependent variables, in symbolic form, that fits a given sample of 200 data points (faces).
Terminal set:	$x_i, i = 1, ..., 25$ where x_i are the landmark points on the face.
Function set:	$+, -, \%, *$.
Fitness cases:	200 faces.
Raw fitness:	The sum, taken over the 200 fitness cases, of the RMS error between value of the dependent variables produced by the S-expression and the target value y_i.
Standardised fitness:	Equals raw fitness.
Hits:	Number of fitness cases for which the value of the dependent variables produced by the S-expression comes within 0.01 of the target value y_i.
Wrapper:	None.
Parameters:	Population Size = 750, Generations = 70.
Success predicate:	An S-expression scores 20 hits.

Table 4.1: Genetic Programming symbolic regression details .

[1]The 200 randomly created facial expressions from the first data set which was used for training

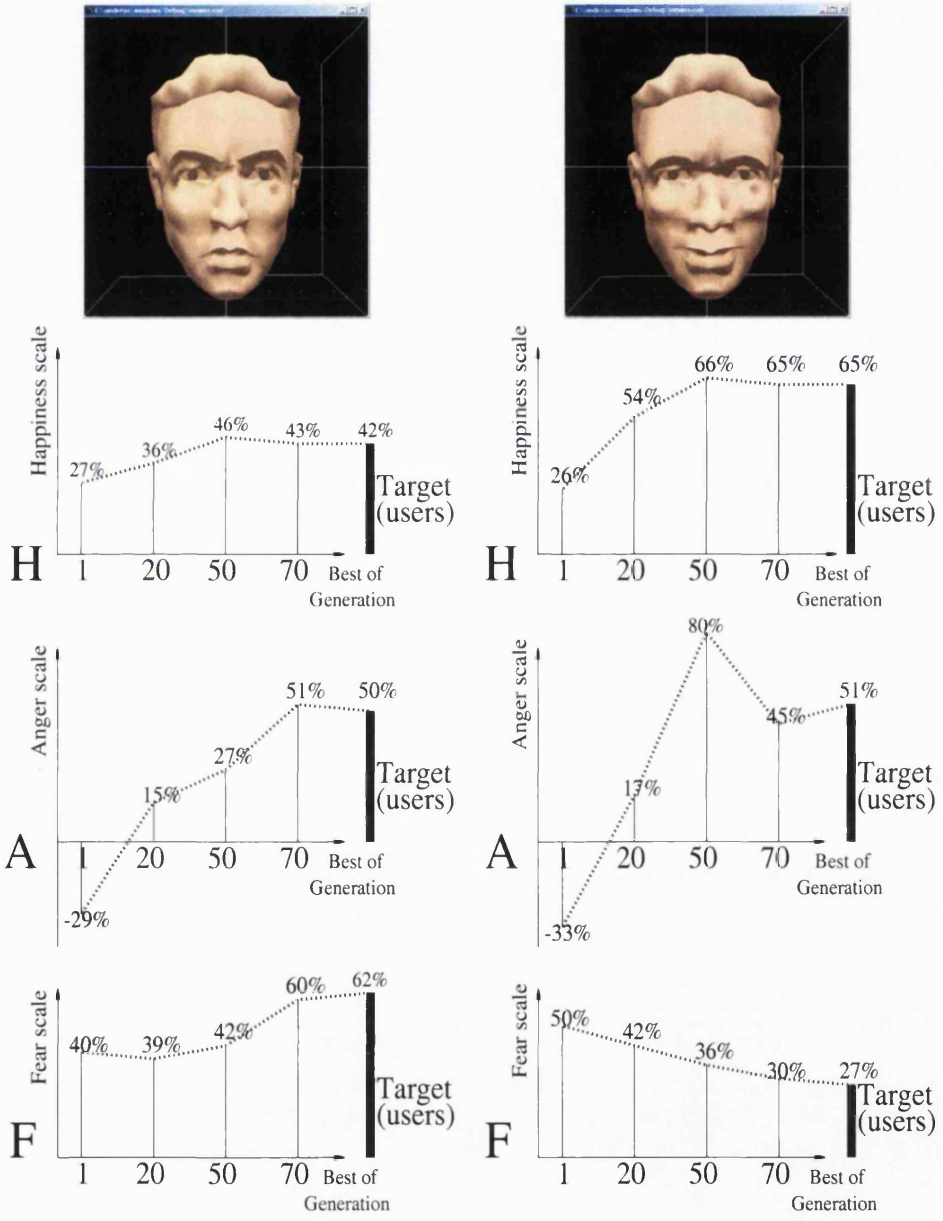


Figure 4.4: The corresponding measurements for degree of happiness (H), degree of anger (A) and degree of fear (F) for the 2 faces, for the best individual in generation 1, 20, 50 and 70 (last generation) are presented.

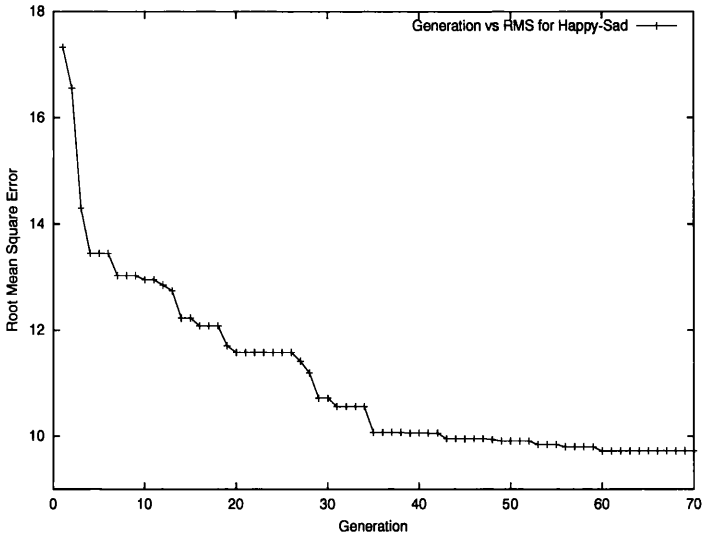


Figure 4.5: The evolution of RMS error of best-of-generation individual against the number of generations for the Happiness-Sadness scale.

In Figure 4.4, the evolution of the Genetic Program[2] for the different expressions is shown: degree of happiness, degree of anger and degree of fear. We arbitrarily select the two faces shown on top of figure 4.4 for presentation in more detail here. All the other cases show similar types of results. The measurements for the three different expressions are shown with degree of happiness identified by the letter H, degree of anger identified by the letter A and degree of fear by the letter F. Next to each graph there is a wider bar, rendered black, showing the average response for the emotional expression encountered, from a pool of at least 3 measurements. This is in fact the target result of the symbolic regression of the GP. The rest of the bars in each graph show measurements for the different expressions of the best-of-generation individual GP for generations 1, 20, 50 and 70. As can be seen, the difference between target measurements which are user evaluations and GP measurements decreases, as the number of generations increases. This decrease is also evident in Figures 4.5, 4.6 and 4.7 where for each expression there is a plot for the root mean square error (RMS) of best-of-generation individual against the actual number of generations.

From these figures it can concluded that the GP is learning over time, and from the two examples given in Figure 4.4, the measurements for emotions of best-of-generations GP individual appear to be almost identical to the user's subjective evaluations.

Therefore, the evidence suggests that the estimated regression equations produced by the GP are a good approximation to user subjective evaluations for the emotional expressions that

[2]A description of Genetic Programs can be found in Chapter 5

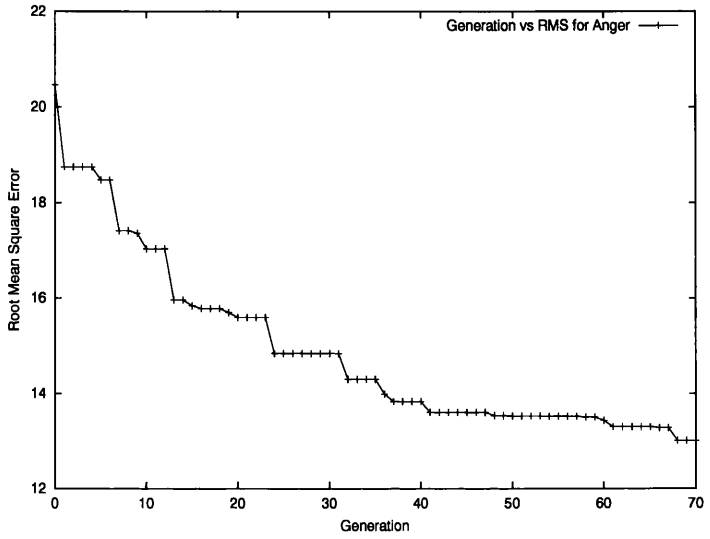


Figure 4.6: The evolution of RMS error of best-of-generation individual against the number of generations for the Angry-Calm scale.

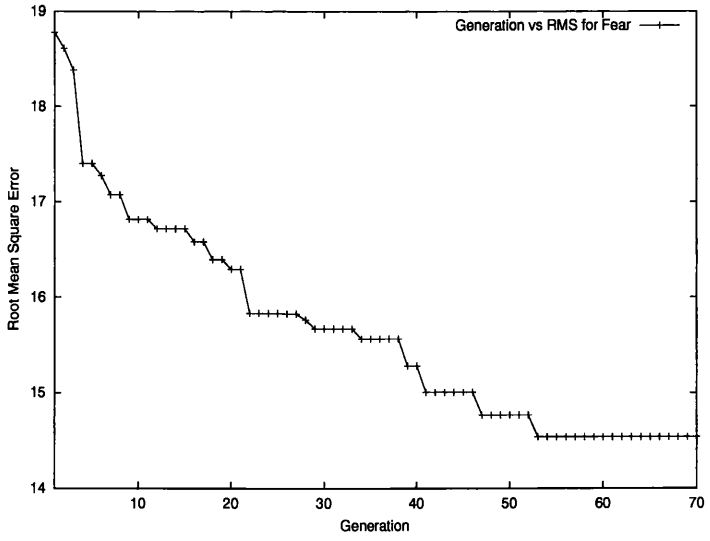


Figure 4.7: The evolution of RMS error of best-of-generation individual against the number of generations for the Fear-Relax scale.

are encountered here.

4.2.4.2 Results

Is there evidence of positive correlation between X and Y, with X being users evaluations of the emotional expressions and Y being the scores for the emotional expressions produced from the estimated regression equations?

Emotion	${r_i}^2$	observed t_i	critical t at $p = 0.01$ and 148 d.o.f.
Happy/Sad	0.85	19.7	
Angry/Calm	0.75	13.8	2.6
Fear/Relax	0.67	10.5	

Table 4.2: Tabulated t.

For the first emotional expression, as shown in Table 4.2, the happiness-sadness scale ${r_1}^2 = 0.85$ and the test statistic is $t_1 = 19.7$. For anger-calm scale ${r_2}^2 = 0.75$ and $t_1 = 13.8$ whereas for fear-relax we have ${r_3}^2 = 0.67$ and $t_3 = 10.5$.

The t-test is performed on the evaluation (and not the training) data set of 150 faces. On 148 degrees of freedom the critical $t = 2.6$ at 1% significance level.

Hence the correlations are significantly different from 0, for all three cases, and there is evidence of high positive correlation between variables X and Y. Therefore, there is evidence that the estimated regression equations will produce similar results to user evaluations of emotional expressions for the facial model we use, *and hence we can replace user's evaluations with these equations.* The similarity is evident in Figures 4.8, 4.9 and 4.10 where scatterplots of the two measurements (user evaluations and estimated regression equations) are shown. As can be seen from the graph very clearly there is a positive correlation between the two variables.

4.2.5 Using Principal Component Analysis to Improve this Method

Although the results from the method above appear satisfactory there might be scope for improvement. It is obvious that the data set used to perform the symbolic regression is high dimensional and very "noisy" making the search task difficult. Not only is not possible to know which of the 25 "landmarks" on the face mostly influence the results, the search method is dealing with subjective measurements. Therefore noise comes as a natural consequence. It was believed that using Principal Component Analysis (PCA) could improve results or more correctly, may prevent results from being more inaccurate. PCA is a statistical analysis technique that enables us to represent most of the variation (95% - 99%) observed in a set of data, which is characterised by a huge number of parameters, using a relatively small set of different

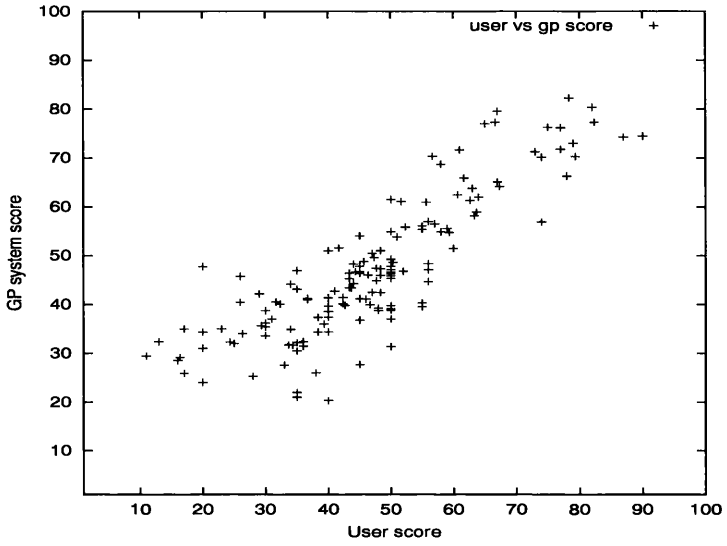


Figure 4.8: The mean user measurement of happiness-sadness scale plotted against the value produced by the symbolic regression for the evaluation data set of 150 faces. The two sets of values are highly correlated, $r^2 = 0.85$.

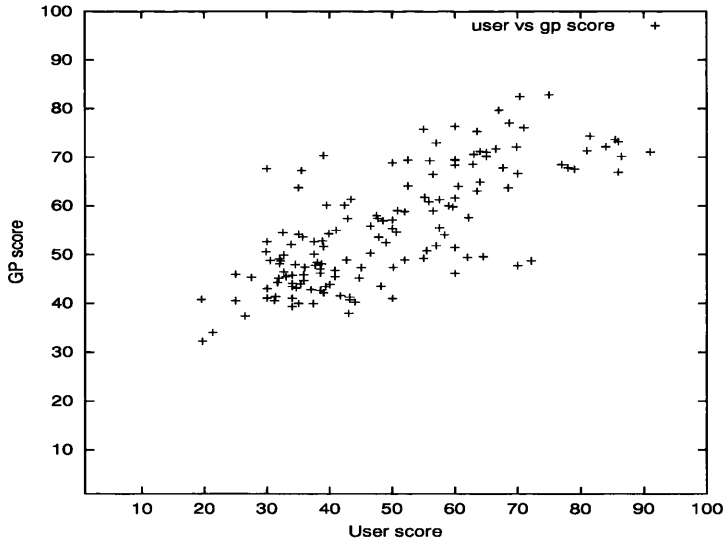


Figure 4.9: Mean user measurement of angry-calm scale plotted against the value produced by the symbolic regression for the evaluation data set of 150 faces. The two sets of values are highly correlated, $r^2 = 0.75$.

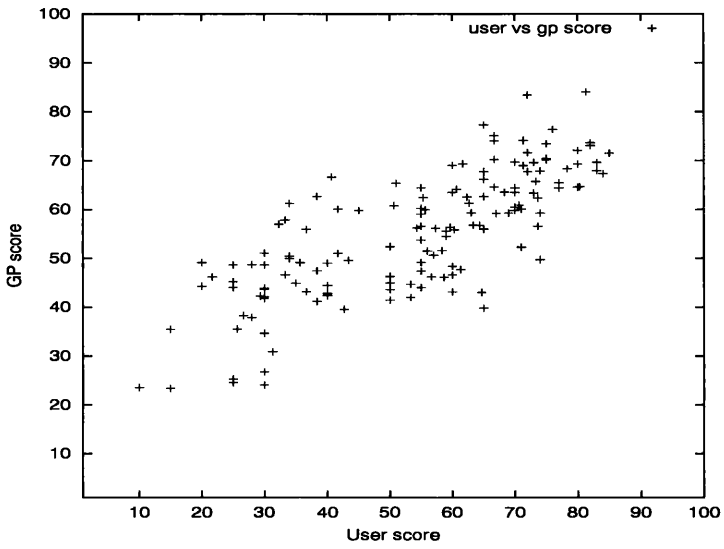


Figure 4.10: Mean user measurement of fear-relax scale plotted against the value produced by the symbolic regression for the evaluation data set of 150 faces. The two sets of values are highly correlated, $r^2 = 0.67$.

parameters. What follows is a step by step description of how PCA was performed in our data set. The theory is based on Jackson [Jac96].

To begin with a data vector, x, is constructed, which consists of all the parameters that characterise the initial data known as the training data. In this case the data vector x would consist of the 25 control points, which are the distances of each "landmark" on the face from the reference point. Therefore:

$$x = (x_1, x_2, ..., x_k)^T \tag{4.1}$$

where $k = 25$.

Given this vector representation and m examples in the training data set, in our case $m = 200$ (the number of faces in our training set), we compute the mean vector $\overline{x}$ as follows:

$$\overline{x} = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{4.2}$$

where x_i, is the data vector of the i^{th} example in the training set.

Next we have to compute the deviations $(x_i - \overline{x})$, from the mean and construct the scatter matrix (also known as the covariance matrix) S, which is given by:

$$S = \frac{1}{m}\sum_{i=1}^{m}(x_i - \overline{x})(x_i - \overline{x})^T \tag{4.3}$$

and compute its eigen solutions.

It is important to note that the scatter matrix is positive semi-definite and hence all the eigenvalues, $\lambda_i \geq 0$. Now, we order the eigenvalues and number them as $\lambda_1, \lambda_2, ..., \lambda_k$ such that $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_k \geq 0$, where k is the number of elements in a data-vector (in our case it is equal to 25). The corresponding unit eigenvectors are then given the same subscripts and will be represented here by the symbols $p_1, p_2, ..., p_k$. Thus we have:

$$Sp_k = \lambda_k p_k, \lambda_k \geq \lambda_{k+1} \wedge p_k^T p_k = 1 \; for \; k = 1, 2, ..., 25 \tag{4.4}$$

The eigenvectors form an orthogonal set of vectors, which span the n-dimensional space of the components of the data vectors. Therefore, by taking the mean and adding linear combination of the eigenvectors to it, any point in this n-dimensional space may be reached.

The trace of the covariance matrix is equal to the total sum of the squared differences, which represent the mean total variation of all elements of the data-vector, x, over all the examples in the training data set, and is equal to the sum of all the eigenvalues. Hence, the eigenvalues describe the variance of each component of the data vector. It can be shown that the eigenvectors describe the modes of variation in the data vectors across the training set, and that the variance explained by each eigenvector is equal to the corresponding eigenvalue. This means that the eigenvectors corresponding to the largest eigenvalues describe the most significant modes of variation in the variables used to derive the covariance matrix.

This results in a set of rapidly decreasing eigenvalues whose total is bounded above by the trace of S. It is also evident that the use of all the eigenvalues would correspond to over-fitting, and that by discarding the small eigenvalues the noise and unrepresentative details in the data would be minimised. This is illustrated in Figure 4.11 where the first 5 principal components are adequate to represent 99% of variation in the data. This means that most of the relevant variation can be explained by a relatively small number of dimesions, t. One method for calculating t is to choose the smallest number of principal components, such that the sum of their variances explains a sufficiently large proportion of λ_T, the total variance of all the variables, where:

$$\lambda_T = \sum_{k=1}^{n} \lambda_k \tag{4.5}$$

Therefore, any data-vector in the training data set can be approximated using the mean and a weighted sum of the deviations obtained from the first t modes using the following equation:

$$x = \overline{x} + Pb \tag{4.6}$$

where,

$$P = (p_1 p_2 ... p_t) \tag{4.7}$$

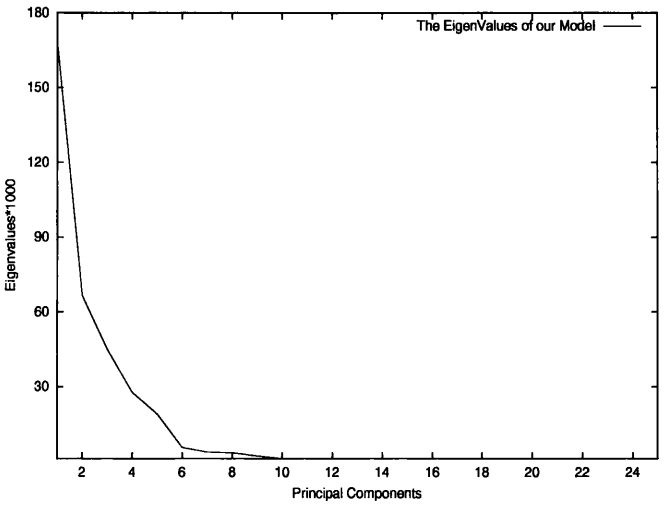


Figure 4.11: A plot showing the λ, the eigenvalues of our data set.

is the matrix of the first t eigenvectors, and

$$b = (b_1 b_2 ... b_t)^T \tag{4.8}$$

is a vector of weights. It's also necessary to note that, by definition (equation 4.7), P is an $n * t$ matrix and hence is, in general, not square. Thus, in general, P^{-1} is not defined. However, since $p_k^T p_k = 1$ (equation 4.4), it is possible to use P^T in place of P^{-1}, to project out the weights required to parameterise a new data-vector via equation 4.9, which is given below:

$$b = P^T(x - \overline{x}) \tag{4.9}$$

Equation 4.6, allows us to generate new examples of the data by varying the parameters, b_k, within suitable limits, so that the new examples will be similar to those in the training set. The parameters, b_k, are linearly independent, though there maybe some statistical non-linear dependencies still present. The limits for the b_k are derived by examining the distributions of the parameter values required to generate the training set. Since the variance of b_k over the training set can be shown to be λ_k, suitable limits are typically of the order of:

$$-3\sqrt{\lambda_k} \leq b_k \leq 3\sqrt{\lambda_k} \tag{4.10}$$

since most of the population($\tilde{9}9.73\%$) lies within three standard deviations of the mean.

Equation 4.9 can then be used, to map our original data set in to this new environment which is less "noisy". Having mapped both our training and evaluation sets into this new environment we can proceed as before. The training set was used to perform symbolic regression on it and the evaluation set was used to measure whether the equations produced by the symbolic regression are significantly correlated with users' subjective evaluations.

4.2.5.1 Results after PCA

As expected there was a slight improvement in the results if symbolic regression was performed in this less "noisy" environment. Table 4.3 underlines this.

Emotion	$r_i{}^2$	observed t_i	critical t at p=0.01 and 148 d.o.f.
Happy/Sad	0.87	20.6	
Angry/Calm	0.77	14.7	2.6
Fear/Relax	0.75	13.8	

Table 4.3: Tabulated t after PCA.

For the first emotional expression, the happiness-sadness scale ${r_1}^2 = 0.87$ and the test statistic is $t_1 = 20.6$. For anger-calm scale ${r_2}^2 = 0.77$ and $t_1 = 14.7$ whereas for fear-relax we have ${r_3}^2 = 0.75$ and $t_3 = 13.8$.

The t-test is performed on the evaluation (and not the training) data set of 150 faces. On 148 degrees of freedom the critical $t = 2.6$ at 1% significance level.

Hence the tests are significant for all three cases and thus there is evidence of a high positive correlation between the two variables. Therefore, there is evidence that the estimated regression equations will produce similar results to user evaluations of emotional expressions for the facial model we use and hence we can replace users with these equations.

4.3 Conclusions

In this chapter a synthetic human facial model was presented, Geoface, improved and used in EVA to represent multi-dimensional data sets. A technique to randomly generate facial expressions was described in addition to a technique that enables us to quantify the individual emotional expressions based on user assessments. It is a pre-requisite of EVA that such a technique exists, hence its development. The next Chapter concentrates on Genetic Programming, the search technique used to find a good mapping from the value system of the data table, to the significant characteristics of the visual structure, as described in Chapter 3.

Chapter 5

Genetic Programming Review

Genetic Programming (GP) is a technique pioneered by John Koza [Koz92] which enables computers to solve problems without step by step, instructions. It is one of a wide range of Evolutionary Computation (EC) techniques and a descendant of John Holland's genetic algorithms (GA). GP (and GA) is one of several problem solving methods based on computational analogy to natural evolution.

GPs automatically generate computer programs. The theory states that there is no need to know anything about the problem trying to solve, as long as there is a "black box" which evaluates the solutions proposed. However, in practice the use of these methods should not be a substitute to thought, since performance of the methods can be considerably enhanced with built in problem specific knowledge.

5.1 Why are we Using GP

Koza in a foreword address in the book[BNKF98], identifies future areas for practical applications of GPs based on a number of characteristics. Some of these characteristics are true for the problem tackled in this thesis and are mentioned below.

More specifically, he identifies areas where the interrelationships among the relevant variables are poorly understood, or where it is suspected that there may be more to the current understanding. The complexity of the interrelationships of the variables of the data used in this thesis (e.g. in the financial problem described in Section 3.1), and the lack of an existing underlying theory, makes our understanding poor by default.

Additionally, Koza mentions the area where an approximate solution is acceptable, or it is the only result that is ever likely to be obtained. The subjectiveness of both interpretation of our data set and interpretation of the naturalistic visual structure, makes an optimal solution non-existent.

Also areas are mentioned, where conventional mathematical analysis does not, or cannot,

provide analytic solutions. The fact that the solutions cannot be treated independently (e.g. the financial ratios in the financial data example), and that solutions are subjectively interpreted makes mathematical analysis hard to interpret.

Finally, Koza identifies areas where there is a large amount of data, in computer readable form (e.g. data tables), that requires examination, classification, and integration, such as financial data. Such data are of main interest to this study.

The above reasons, qualify GP techniques as a good candidate to solve the minimisation problem of interest to this study, explained in Chapter 3.

5.2 Introduction

Over the years, biologists have identified many principles which govern the evolution of living things, at several levels of detail. At the highest level, the *Darwinian* theory of natural selection or survival of the fittest governs the evolutionary adaptation of the biological world from the smallest virus to the most complicated mammal. Entities that are better able to perform tasks in their environment (i.e. fitter individuals) survive and reproduce at a higher rate than less fit entities. The power of Darwin's theory lies on its general applicability which is even now reshaping many of the sciences [Den95]

Offspring usually result from the growth of a single cell which contains a single specific example of the genetic material for that species. Through that genetic material the parent(s) influence the inheritable structure and function of the offspring. Different species transfer the genetic material for offspring differently, but in all species the parent(s) provide the material in some way or other with some random modification to the genetic material to distinguish the genetic endowment of the offspring from that of the parent.

In sexual recombination (crossover) we have two parents that merge and some genetic material is taken from each one of them. Mutation is the random modification briefly explained above.

The evaluation of the individual through its reproductive performance (called fitness) and the creation of a genetic plan for the offspring are central themes in the evolutionary adaptation of biological organisms.

These themes can be applied to the evolutionary adaptation of computer structures, resulting in the field of Evolutionary Computation whose central focus is to apply the concepts of selection based on fitness to a population of structures in the memory of a computer.

5.3 Evolutionary Computation

The natural world abounds with structures of incredible complexity and apparent cleverness. Since the time of Darwin, it has been increasingly clear that these structures didn't just happen, but rather evolved. The concept of natural selection or survival of the fittest has been seen to be a very powerful paradigm, given the existence proof of the biological world [Koz92].

Any solution of a problem can be conceptualised as a search through the space of all potential solutions. However, the search may be radically different. For example it can be deterministic or stochastic, complete or incomplete, blind, partially sighted and heuristic.

Evolution can also be seen as a search process. This search process is usually trying a few possibilities, deciding how "fit" they are and using the information to decide which others to try next. The procedure is repeated until an individual is found that solves the problem, i.e. has a high enough "fitness".

A computer program can be thought of as a particular point within a search space of all such programs. The search space is likely to be infinitely large. Evolutionary Computation can sometimes be considerably more powerful than alternative search techniques, such as exhaustive or random search. Its directed search makes it possible to find solutions to problems quickly, on tasks that would take random search considerably longer than the life of the universe. The search is directed by rules and principles used to guide the algorithm to portions of the search space that are likely to contain better solutions. Different search techniques are described below, and reasons are given for choosing Genetic Programming techniques to solve the optimisation problem, presented in EVA.

5.4 Search Techniques

Generally search techniques can be divided into calculus-based, enumerative and stochastic, as shown in Figure 5.1. Typically there are a number of aspects taken into account for each of these search techniques. Namely, the representation of the solutions, i.e. the data structure used, the search operators that help move from one configuration to the next and finally the search mechanism that defines how the search space is explored.

Enumerative Techniques in principle, perform a blind search of every point in the search space to find the solution. No knowledge of the search space is used to perform the search. The procedure is indeed simple to implement but in the learning domains, the possible points are far too large for direct search, with the most interesting problems. In contrast, GP works in a combinatorial space suffering from the so-called curse of dimensionality. The volume of the solution space is so big that performing an exhaustive search on such space is potentially

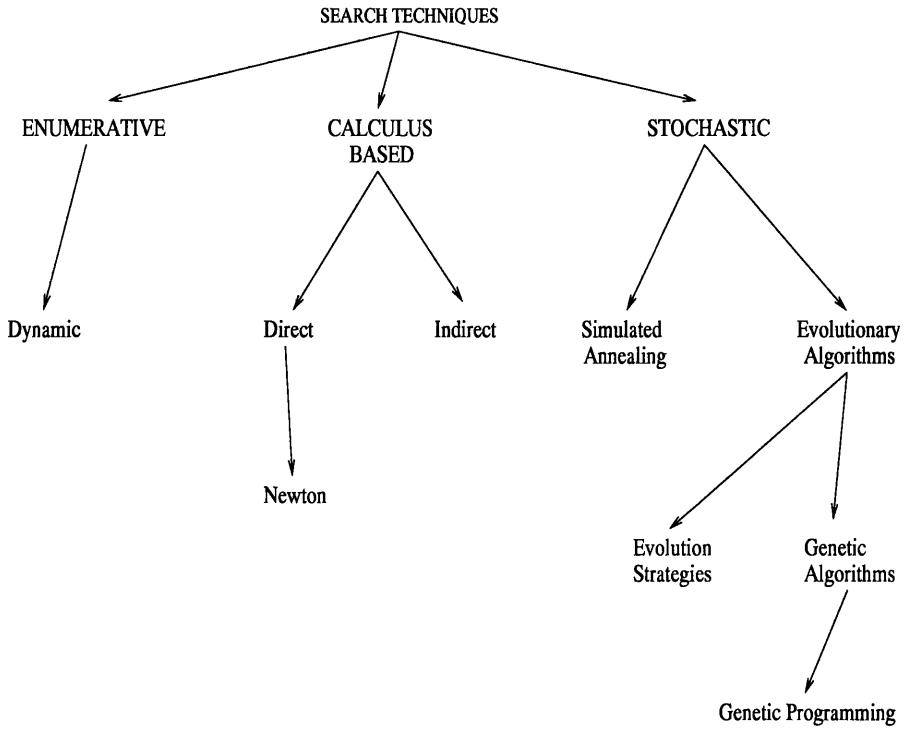


Figure 5.1: Search Techniques [LQ95].

impossible.

Calculus based techniques require the search space to be "smooth", well structured with not many local optima. The search space is treated as a continuous multi-dimensional function and search is looking for maxima (or minima). Search is initiated at some point of the search space with each step moving to a neighbouring position with better objective function values. Hill Climbing techniques belonging to this category, tend to get stuck into local optima, especially when dealing with complex search spaces. Calculus-based techniques are better used upon well behaved problems or problems which can be transformed to become well behaved.

Stochastic search (using pseudo random numbers) uses information from the search to guide the probabilistic choice of the next point(s) to try. They are general in their scope, being able to solve some very complex problems that are beyond the abilities of either enumerative or calculus based techniques.

Simulated Annealing searches for minimum energy states using an analogy based upon the physical annealing process where large soft low energy crystals can be formed in some metals (e.g. copper) by heating them and then slow cooling. It is a single point to point search and therefore can converge to local optima.

Evolutionary algorithms simulate a collective learning process within a population of individuals. Each individual represents a point structure in the search space of potential solutions to a specific problem. After arbitrary initialisation of the population, the set of individuals evolves towards better regions of the search space by means of partly stochastic processes. Partly stochastic since the environment provides quality or fitness information as feedback about the search points.

In **Evolutionary Strategies** [Rec73], [Bac96] unlike GAs [Hol75] and GPs, encoding of individuals in the search space, is through the use of vectors of real values. Each new point is created by adding some random noise to the current one. If the new point is better, search proceeds from it, otherwise the older point is retained. In contrast a GA represents points in the search space by a vector of discrete (typically fix sized) bit values.

GA and **GP** are a form of beam search due to the fact that they retain a population (rather than a point-to-point search) of candidate solutions that is smaller than the set of all possible solutions [Ang93] [Tac94] [Alt94]. Beam search is a compromise between exhaustive search and hill climbing. In a beam search, some "evaluation metric" (the fitness function in GP) is used to select out a certain number of the most promising solutions for further transformation. All others are discarded. The solutions that are retained are the "beam" (the "population" in GP). A beam search limits the points it can find in search space to all possible transformations that result from applying the search operators to the individuals in the beam. GP also regulates the contents and ordering of the beam. The contents are regulated by the genetic operators and the ordering is, for the most part, regulated by fitness-based selection. Simply put, the more fit an individual, the more likely it will be used as a jumping-off point for future exploration of the search space.

For the problem Empathic Visualisation Algorithm is trying to address, there are a number of important aspects that were taken into account, before selecting GP as the appropriate search technique. First of all, the search space of all combinations of sets of feature functions is huge, hence enumerative approaches can be safely excluded. Moreover, nothing can be said about this spaces' nature and more importantly whether it is relatively smooth, well behaved, with few local optima. In fact, it can be safely assumed from its complexity, that the search space is quite rough and high dimensional, making stochastic techniques the only possibility. Additionally, the nature of population-based stochastic techniques make them more suitable for the unknown, complex search space the method is trying to explore. Furthermore, bearing in mind that a combination of muscular functions are collectively evolved, the need for modular programming can be assumed. The fact that GP supports modular programming through automatically defined

functions (ADFs) and hence a different ADF can be used per muscular function, strengthes the choice made.

Below there is a brief description of Genetic Algorithms followed by a more detailed one for Genetic Programming.

5.5 Genetic Algorithms (GA)

The original work that led to Genetic Programming was Genetic Algorithms. The GA, pioneered by Holland [Hol75] his students and colleagues at the University of Michigan, Goldberg [Gol89] and others, is a highly parallel mathematical algorithm that transforms a population of individual objects each with an associated fitness value, into a new generation of the population. It uses the Darwinian principle of reproduction and survival of the fittest and naturally occurring genetic operations such as *crossover* (sexual recombination which is in fact the principal genetic operation) and mutation. Each *individual* in the population represents a possible solution to a given problem. The GA attempts to find a very good or best solution to the problem by genetically breeding the population of individuals.

Genetic Algorithms frequently operate on fixed length character strings, often binary, as the structure undergoing adaptation. Fitness is determined by executing task specific routines and algorithms using an interpretation of the character string as the set of parameters.

Crossover is the principal genetic operator employed, with mutation included as an operator of secondary importance. The three dominant different approaches of crossover are *one-point crossover, two-point crossover and uniform crossover.* With one-point crossover one value representing the position in the string is chosen at random, and the values of the bit strings are interchanged at that particular point. With two-point crossover two values representing the positions in the string are chosen at random, and the values of the bit strings in-between the two values are interchanged. In uniform crossover individual vector positions between parents are swapped with a 50% probability. It must be pointed out that individuals are selected for recombination almost always based on their fitness. Figure 5.2 shows an example of the results that each of the operations produce.

Besides crossover, other genetic operations are used to create offspring from parents. Allele mutation, usually called bit-flipping mutation when working on binary vectors modifies a single position in the parent vector by assigning it a new random value from the appropriate range for the feature.

In preparing to use the conventional GA operating on fixed length character strings to solve a problem, the user must determine; the representation scheme, the fitness measure, the

Individuals: A: A_0 A_1 A_2 A_3 A_4 A_5 B: B_0 B_1 B_2 B_3 B_4 B_5
Results:

One point crossover (at position 2):

C: A_0 A_1 B_2 B_3 B_4 B_5 D: B_0 B_1 A_2 A_3 A_4 A_5

Two point crossover (at position 2 and 4)

C: A_0 A_1 B_2 B_3 B_4 A_5 D: B_0 B_1 A_2 A_3 A_4 B_5

Uniform crossover (not unique)

C: B_0 B_1 A_2 A_3 A_4 B_5 D: A_0 A_1 B_2 B_3 B_4 A_5

Figure 5.2: Example of different crossover mechanisms.

parameters and variables for controlling the algorithm, and finally a way of designating the result and a criterion for terminating a run, [Koz94].

Specification of the representation scheme starts with a selection of the string length L and the alphabet size K. The fitness measure assigns a fitness value to each possible fixed-length character string in the population. The primary parameters for controlling the GA are the population size, M, and the maximum number of generations to be run. Each run of the GA requires specification of a termination criterion for deciding when to terminate a run and a method of result designation.

The GA involves probabilistic steps for at least three points in the algorithm. First of all, the creation of the initial population. Secondly, the selections of individuals from the population on which to perform each genetic operation (reproduction, crossover), and finally, the choice of a point in which to perform the genetic operation.

As a result of this probabilistic nature of GA, it may be necessary to make multiple independent runs of the algorithm in order to obtain a satisfactory result for a given problem.

In practice, the GA is surprisingly rapid in effectively searching complex, highly non- linear, multi-dimensional search spaces [Koz92]. This is all more surprising because the GA does not know anything about the problem domain or the internal workings of the fitness measure being used.

The power of GA is being demonstrated for an increasing range of applications; financial, imaging, VLSI circuit layout, gas pipeline control and production scheduling [Dav91a]. But one of the most intriguing uses of GA - driven by Koza [Koz92]- is automating program generation.

5.6 Genetic Programming (GP)

GP is an offshoot of GA, but in some ways it is more of a generalisation rather than a specialisation of its parent discipline, since it is more expressive. Expressiveness is achieved, due to the nature of the representation of a GP, a computer program as opposed to "fixed" length bit strings in conventional GAs.

Existing methods do not seek solutions in the form of computer programs. They include specialised structures which are nothing like computer programs (e.g. weight vectors for neural networks, chromosome strings in conventional GA). Each of these specialised structures can facilitate the solutions to certain problems and many of these facilitate mathematical analysis that might not otherwise be possible. However they are an unnatural and constraining way of getting computers to solve problems without being explicitly programmed.

GPs on the other hand use the principles that govern nature into computer programs. These programs have the flexibility needed to express the solutions to a wide variety of problems, it is an executable genetic material. Also, computer programs can take the size, shape and structural complexity necessary to solve problems. This is because its non-linear tree structure genetic material can vary in length. Finally, computer problems have a way to solve the problem of program induction, the actual GP algorithm.

5.6.1 Emergent Intelligence

In contrast to the traditional knowledge based approaches to artificial intelligence (AI), this method of problem solving is termed emergent intelligence [Ang94]. At a very high level, AI problem solving methods divides into two distinct categories: weak and strong. A strong method is one that contains a significant amount of task specific knowledge in order to solve it. In contrast, a weak method (best-first search, heuristic search etc.) requires little or no knowledge in order to solve it. As the name implies, strong methods are more powerful than weak methods. However, they are only narrowly applicable due to their task specific knowledge. In AI emphasis is given on expert knowledge.

Emergent intelligence (EI) reduces or removes the reliance on explicit knowledge in a problem solver by favouring instead the emergence of problem specific constraints as a direct consequence of the action of problem solving. Emergent phenomena are defined as local interactions creating global properties [Ban93]. Emphasis is given on learning.

5.6.2 Structures Undergoing Adaptation - Algorithm

Intuitively, if two computer programs are somewhat effective in solving a particular problem, then some of their parts probably have some merit. By recombining randomly chosen parts of

somewhat effective programs, we may produce new computer programs that are even fitter in solving the problem. At each stage of this highly parallel, locally controlled, decentralised process, the state of the process will consist only of the current population of individual programs.

The structures that undergo adaptation in GP (the actual computer programs) are active. They are not passive encoding of the solution to the problem as with simple GA. Instead, given a computer on which to run, the structures in GP are active structures that are capable of being executed in the current form. The following steps, from Koza's book, define the algorithm:

1. Generate an initial population of random compositions of the functions and terminals of the problem (computer programs).

2. Iteratively perform the following sub-steps until the termination criterion has been satisfied:

   (a) Execute each program in the population and assign it a fitness value according to how well it solves the problem.

   (b) Create a new population of computer programs by applying the following two primary operations. The operations are applied to computer program(s) in the population chosen with a probability based on fitness

      i. Copy existing computer programs to the new population.

      ii. Create new computer programs by genetically recombining randomly chosen parts of two existing programs.

3. The best computer program that appeared in any generation (i.e. best so far individual) is designated as the result of GP. This result maybe a solution (or an approximate solution) to the problem.

5.6.3 Functions and Terminals

The size, shape and contents of these computer programs can dynamically change during the process. The set of all possible structures in GP is the set of all possible compositions of functions that can be composed recursively from the set of N functions from $F = (f_1, f_2..., f_N)$ and the set of N terminals from $T = (a_1, a_2..., a_N)$.

Function and Terminal set should be selected so as to satisfy the requirements of *closure* and *sufficiency*.

Closure [Koz92] states that any function should be well defined and closed for any combination of arguments that it may encounter. The boundary conditions for the functions are very

important. What do the functions do when handed data that are illegal or in some way out of range? *Typing* is one way out of this dilemma, where each node carries its type as well as the types it can call, thus forcing function calling it to cast the argument into the appropriate type (see section 5.6.10).

Sufficiency [Koz92] property requires that the set of terminals and the set of primitive functions are capable of expressing a solution to the problem. There is a balance between having a rich enough set of functions and terminals and having too many - a balance that must be determined empirically. A large function set enlarges the search space.

In general, numerous extraneous functions in a function set degrade performance to some degree; however, a particular additional function in a function set may dramatically improve performance for a particular problem. For some problems, where it is not clear in advance what set of functions is minimally sufficient to solve the problem, it is generally better to include potentially extraneous functions than to miss a solution altogether. Extraneous terminals on the other hand, reduce performance.

5.6.4 Initial Structures - First Population

Each program individual can be expressed as a tree. Figure 5.3 shows two examples of such programs expressed as trees. There are different methods used to create the randomly chosen trees that will constitute the initial population of the GP.

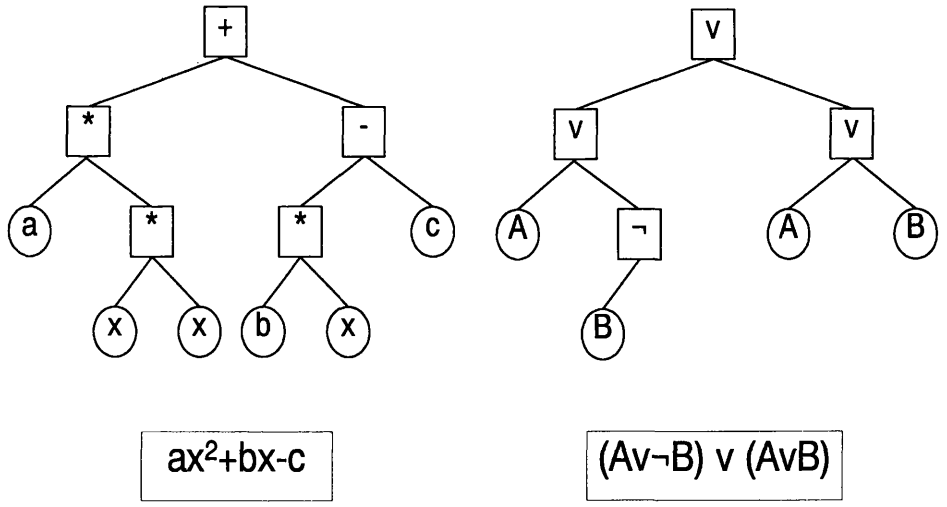


Figure 5.3: Example of GP programs expressed as trees.

First is the *full method*. The *full method* of generating the initial random population involves creating trees (programs), for which the length of every non-backtracking path between an endpoint and the root is equal to the specified maximum depth. This is accomplished by restricting the selection of the label of points at a depth less than maximum to the function set

F, and the selection for points at maximum depth to set T.

In addition to the above we have the *grow method* which, involves growing trees that are variably shaped. The length of a path between an endpoint and a root is no greater than the maximum depth. This is accomplished by making the random selection of the label for points at depth less than the maximum from the combined set $C = F \bigcup T$, while restricting the random selection of the label at maximum depth to terminal set.

Finally, we have the *ramped half and half* hybrid method [Koz92]. This method produces a wide variety of trees of various sizes and shapes. It involves creating an equal number of trees using a depth parameter that ranges equally between two and the maximum specified depth. Then for each value of depth, 50% of the trees are created using the *full* method and 50% via the *grow* method. This is actually the dominant method. It is preferred because it creates trees having a wide variety of size and shapes.

Duplicate individuals in the initial population are considered to be unproductive dead-wood. They are a waste of computational resources plus they reduce the genetic diversity of the population. Hence there is usually a check for uniqueness in the initial population.

5.6.5 Fitness

Fitness is the driving force of Darwinian natural selection and likewise both GA and GP. GP population is the beam; to perform GP operators, GP implements fitness-based selection. Any and all boundary conditions in the fitness function will be ruthlessly exploited by the individuals in the population.

The fitness function is the only chance of communicating ones intentions to the powerful process that genetic programming represents. It is a necessity that it communicates precisely what one desires.

A fitness function needs to rate an individual on how well it performs on the problem to be solved. It also needs to rate individuals in such a way that they can be compared and more successful individuals are distinguished from less successful ones. There are different forms of fitness (raw, standardised, adjusted and normalised) all described below.

Raw fitness [Koz92] is a measurement that is stated in the natural terminology of the problem itself. In most cases but not always it is evaluated over a set of fitness cases representative of the domain space, since they form the basis for generalising the results. The most common definition of raw fitness is **error**. If used taking into account the square of distances then it increases the influence of the distant points.

Standardised fitness [Koz92], restates the raw fitness so that lower numerical value is al-

ways a better value. If maximum value for fitness is not known and a bigger value for fitness is better we either use adjusted or normalised fitness.

Adjusted fitness [Koz92] is computed from standardised fitness. $a(i,t) = \frac{1}{1+s(i,t)}$, where $s(i,t)$ is the standardised fitness of the i^{th} individual at any generational time step t and lies between 0 and 1. Adjusted fitness is bigger for better individuals in population. This form of fitness shows importance of small differences if the value of standardised fitness approaches 0. Therefore, it distinguishes a good individual from a very good one. It is used in fitness proportionate selection.

Normalised Fitness [Koz92], also for fitness proportionate selection methods. It is computed from the adjusted fitness, $n(i,t) = \frac{a(i,t)}{\sum_{k=1}^{M} a(k,t)}$ where M is the population. It has three desirable characteristics; it ranges between 0 and 1, it is larger for better individuals and the sum of normalised fitness is 1.

5.6.6 State of the System

In GP, the state of the adaptive system at any point during the process consists only of the current population of individuals. No additional memory or centralised bookkeeping is necessary.

In a computer implementation of the GP paradigm, it is also necessary to cache the control parameters for the run, the terminal set, the function set (if mutation is being used) and the best individual so far (if used as part of a process of result designation for the run).

In conventional generational GP the entire new generation is created and replaces the old one, in one go. With steady state, developed based on work done by Gilbert Syswerda on steady state GA [Sys91], individuals are created one at a time, evaluated immediately for fitness, and then merged into the population in place of an existing low fitness individual. A comparison of steady state GP to generational GP can be found in [Kin93]. A *generational equivalent* in steady state GP exists when the number of new individuals that have been generated is equal to the population size.

5.6.7 Termination Criterion

A run terminates when either a pre-specified maximum number G of generations have been run (the generational predicate) or some additional problem specific success predicate has been satisfied.

The method of "best so far individual" is the one that used most often for result designation, but the "best-of-generation individual" can be used as well. Note that in most cases the "best so far individual" is indeed in the generation that terminated the run therefore in these cases both methods will produce the same result.

5.6.8 Different Representations

The most common form of representation for GP, as stated already, is that of a tree. This representation is in fact directly taken from Reverse Polish Notation (RPN). A population of computer programs can be represented by a collection of RPN trees.

Another structure is the linear one. A linear structure is simply a chain of instructions that are executed from left to right. Unlike trees the linear structure has no obvious way for a function, to receive input. What is missing is memory, a place to hold the inputs to the functions. A typical way to give memory to the instructions is by using a register machine [Nor94] [Hue96]. A register machine uses a linear string of instructions operating on a small number of memory registers. The instructions read and write values from and to the registers.

There is another representation for GP, that of minimal Directed Acyclic Graph (DAG). In fact it is possible to change from trees to minimal DAG without any change in the functionality of a GP system.

The basic difference between these two approaches is the choice of elements. RPN uses the pre-defined symbols that make up the basic language, while DAG uses the evolving subtrees as primitives. It is shown [Kei96] that the DAG representation is equivalent to RPN with respect to the programs it can represent, the execution of programs, and the mechanism of subtree crossover. In fact the amount of memory required can be reduced considerably in the expense of more complexity.

5.6.9 Different Selection Mechanisms

Apart from the *fitness proportionate* selection [Koz92] mechanism which uses adjusted or normalised fitness values of the individuals to decide which one to choose, there are two further mechanisms.

Firstly, *rank selection* [Koz92], where each individual in the population is assigned a rank, hence sorted (from 1 to the size of the population). Selection is then performed so that the best individual receives a predetermined multiple of the number of copies that the worst individual receives (thus equalising selective pressure). Rank selection can prevent premature convergence.

Secondly, *tournament selection* [Koz92], which parallels the competition in nature among individuals for the right to mate. Tournament selection is performed each time it is needed, on a different random sample drawn from the population which is quite small compared with the population. Best individual in the population has probability 1.0 for prevailing in the competition, the worst individual has probability of 0 for prevailing, and a middle-ranking individual

has approximately 0.5 probability of prevailing. Tournament selection is in effect a probabilistic version of rank selection.

5.6.10 Different GP Systems Based on Syntax

While the assumption of representational closure, discussed above, reduces the size of search space by excluding non-viable programs and makes the definition of subtree crossover trivial for trees, for some problems it is necessary to include distinct return types for some programmatic components.

Strongly typed genetic programs [Mon95] evolve programs that use typed languages. In this case crossover must be defined to enforce all syntactic restrictions introduced by the type constraints. Note that this has the effect of reducing the size of the search space further for some problems by excluding tree organisations that in a single type language using analogous primitives might be created. On the other hand, assuming too many restrictions may actually impede the progress of the GP especially if the restricted search space becomes fragmented with respect to crossover.

Pedestrian GP is a system devised by Banzhaf [Ban93] that uses a traditional GA binary string representation. He used this approach to successfully evolve programs that predict simple integer sequences.

Minimum Description Length (MDL) is a method devised so that fitness measures, not only how well a program performs, but also its size. MDL is calculated by considering how many bits are needed to code the program and how many to code a description of its error, i.e. those cases where it returns the wrong answer.

5.6.11 Primary Genetic Operations

The are only two primary genetic operations used. The first one, *reproduction*, is an asexual operation. The individual is selected based on fitness and simply copied across to the new population.

The second one, *crossover*, creates new offsprings that consist of parts taken from each parent. It is a sexual operation in that there are two parents combining together. Both individuals are selected using fitness. Then using a uniform probability distribution, one random point from each parent is selected to be the crossover point [LP01]. Two offsprings are then produced. Because entire subtrees are swapped and because of the closure property GP assures syntactically correct crossover.

In fact in GP the general effectiveness of crossover is an issue of concern [SE91], [Dav91b], [Jon95]. This work does not imply that crossover is an inappropriate operator for

evolving solutions, just that it is not necessarily the most efficient choice for all problems and in particular that its range of effectiveness may be significantly smaller than expected [ES93]. For the time being it is still the main operator and there are a number of variations of the method. The exploration of different crossover techniques rose from the fact that GP crossover is very different from biological crossover. Crossover in standard GP is unconstrained and uncontrolled. Crossover points are selected randomly in both parents. There are no predefined building blocks (genes). Crossover is expected to find the good building blocks and not to disrupt them even while the building blocks grow.

Most of the variations of the crossover techniques described below try to force the crossover to a particular point.

Self-Crossover uses a single individual to represent both parents. The single individual can be selected using fitness proportionate or tournament selection methods. Kim Harries [HS97] concludes that self-crossover is generally a bad hill climber.

Single Crossover: where two individuals are selected based on fitness, and a point is chosen inside a copy of each. A single new individual is created by replacing the subtree selected in one individual with the subtree from the other individual.

Non-fitness Single Crossover: just like single crossover, except that the two individuals are selected using uniform random selection over the entire population instead of based on fitness.

Modular/Cassette-Crossover gets over the limitation of standard crossover operator of not being possible to swap blocks in the middle of a tree path. Developed by Kinnear and Altenberg Modular or Cassette crossover can be seen as a module swap between individuals [Alt94].

Tackett [TC94] devised a method based on the above, for reducing the destructive effect of the crossover operator called *Brood Recombination.* With this the parents are randomly crossed-over N times, each time creating a pair of children and only the two most fitted ones are selected. However, this method, obviously, increases delays in our system due to performing more evaluations than normal.

D'Haeseleer [D'h94] suggested alternative genetic operators called *Strong Context Preserving Crossover* (SCPC) and *Weak Context Preserving Crossover* (WCPC). SCPC allows two subtrees to be exchanged during crossover between two parents only if the points of crossover have the same co-ordinates (also called one-point crossover). WCPC relaxes this rule slightly by allowing crossover of any subtree of the equivalent node in the other parent.

Langdon [Lan96] devised a variation he calls Directed Crossover in which he succeeds in dynamically redistributing crossover locations to code in need of improvement as the population evolves. The location is chosen to avoid disrupting code that is working.

"Intelligent" Crossover has been attempted by Teller [Tel96] by letting the crossover operator select the crossover points in a way that is less destructive to the offspring. Using this method, the percentage of recombination events that resulted in offspring better than the parents, approximately doubled compared to traditional GP crossover. Zannoni used a cultural algorithm (a more traditional machine learning algorithm) to select crossover points with similar results [ZR97].

Iba and De Garis offer an adaptive crossover, termed *recombinative guidance* that uses statistical methods to estimate the relative worth of each subtree in a GP [IG96].

As a conclusion, crossover can be improved substantially in both the quality and efficiency of the search it conducts. But there is a cost associated with each improvement. Each of the techniques described above, may carry additional digital overhead such as less efficient use of memory and CPU time. Therefore, it can be argued that crossover acting as something other than a macromutation operator does not come free, in biology or in GP.

5.6.12 Secondary Operations

There are a large number of secondary genetic operations that researchers develop and use in order to better solve specific problems in the area of Genetic Programming. A small number of them, are briefly described.

Mutation [Koz92]: where an individual is selected based on fitness and then an internal point in a copy of the individual is replaced by a randomly created tree. Point mutation is a variant of these in that only the internal point selected for mutation is replaced.

Hoist [Koz92]: (a special case of single crossover) where a new individual is created by selecting a point inside a copy of an existing individual based on fitness, and elevating it up to be the entire new individual.

Create (a special case of mutation) [Koz92]: where an entirely new individual is created in the same way as the initial random generation. This locates a random new point of departure in the search space.

Encapsulation [Koz92]: where the encapsulation operator is a means for automatically identifying a potential useful subtree and giving it a name so that it can be referenced and used later. Encapsulated subtrees become single points and hence not affected by crossover.

5.6.13 Improving the Speed of GP With Parallelisation

A simple way to accelerate the GP and at the same time to keep diversity high, is to use parallelisation. Parallel populations, also known as *demes*, might possess different parameter settings that can be explored simultaneously, or they might cooperate with the same set of parameters,

but each work on different individuals.

Koza developed one such parallel population system [AK96] by using a different processor for each population. There is good speedup in processing a large population. Koza and Andre, have reported intriguing results of a more than linear speed-up due to the arrangements. It is possible that a large population allows the system to find solution with fewer evaluations [KM94].

A different technique to improve speed is by evaluating the programs this time in parallel. In [JP96], a system is reported to implement parallel evaluations of programs using a very powerful parallel computer.

5.6.14 Induction of Hierarchically Organised Structures

Given that even the most basic programming courses emphasise modular programming, it is a small conceptual jump from the evolution of programs to the question of evolving modular programs. Modular programs bring with them a number of desirable features and some interesting research issues. Three distinct methods have been investigated for inducing modular GPs: *automatically defined functions* (ADF), [Koz92], *adaptive representation* [RB94] and the *Genetic Library Builder* [AP92] typically referred to as modules. Each of these methods attempts to adaptively elevate the expressivity of the defined language by dividing the solution into smaller blocks of code (modules) but with distinct mechanisms.

ADFs are inspired by how functions are defined in LISP during normal manual programming. The program containing ADFs is a tree just like any program in regular tree-based GP. However, when using ADFs, a tree is divided into two parts or branches as shown in Figure 5.4.

The first one, the result-producing branch, which is evaluated during fitness calculation, and the second one, the function-defining branch, which contains the definition of one or more ADFs.

These two branches are similar to any program written in C, C++ for example. The resulting branch is the main function and the function-defining branch (ADFs) corresponds to the function definition part of the program. Both of these program components participate in evolution, and the final outcome of GP with ADFs is a modular program with functions. In Figure 5.5, a very simple example of a GP using a single ADF is being demonstrated.

5.6.15 Effect Of Primary Control Parameters

GPs typically operate with large population sizes. Much work remains to be done to determine what makes a problem easy or hard for GP, and what population sizes are best suited to solving each type of problem. The size is different for each problem, but it is probably true that each

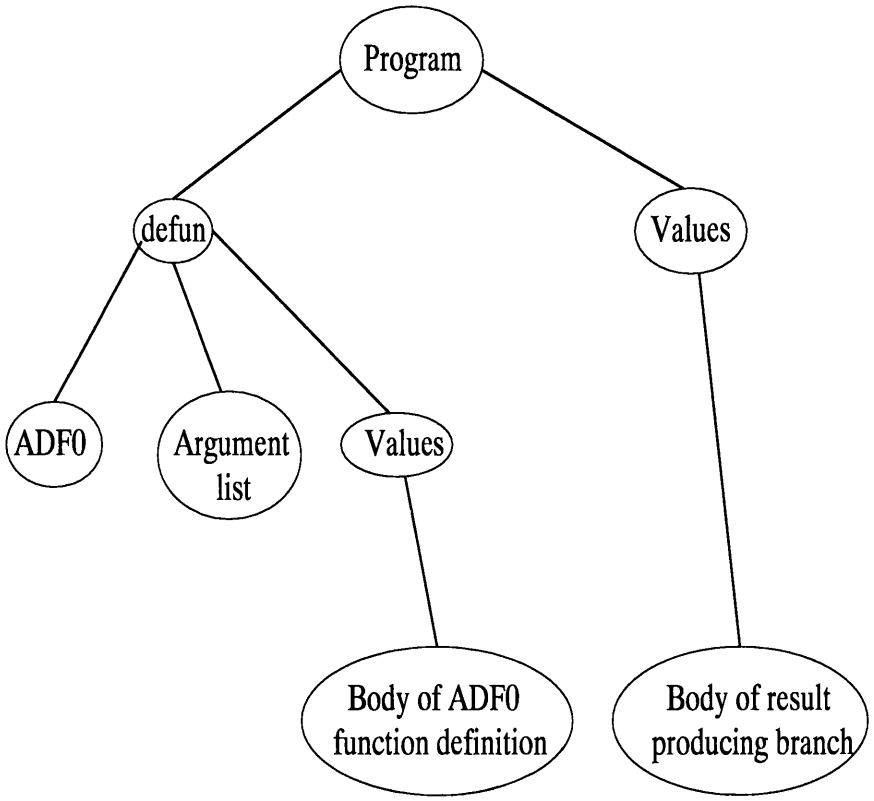


Figure 5.4: A typically automatically defined function definition.

problem has a minimum size. It appears to be the case that below this critical minimum size, running for more generations will usually not produce a correct solution.

Moreover, using multiple runs is also an effective way to utilise a much larger virtual population. It isn't the same as running one much larger population, but can be effective nonetheless.

As a rule of thumb, because the normal variation on one run is very large, one should never generalise from one run.

5.6.16 Conclusion

GP is a robust and efficient paradigm for discovering computer programs using the expressiveness of symbolic representation. It is a very powerful method and as Koza identified in his book, On the Programming of Computers by Means of Natural Selection [Koz92], it has a number of characteristics.

Firstly, GPs do not have any pre-processing requirements, like conventional GAs do for representation in a bit string format. Moreover, the internal representation of the problem matches the original natural representation of the problem. Furthermore, usually no post-processing requirements are needed and finally, the results are portable since they are always

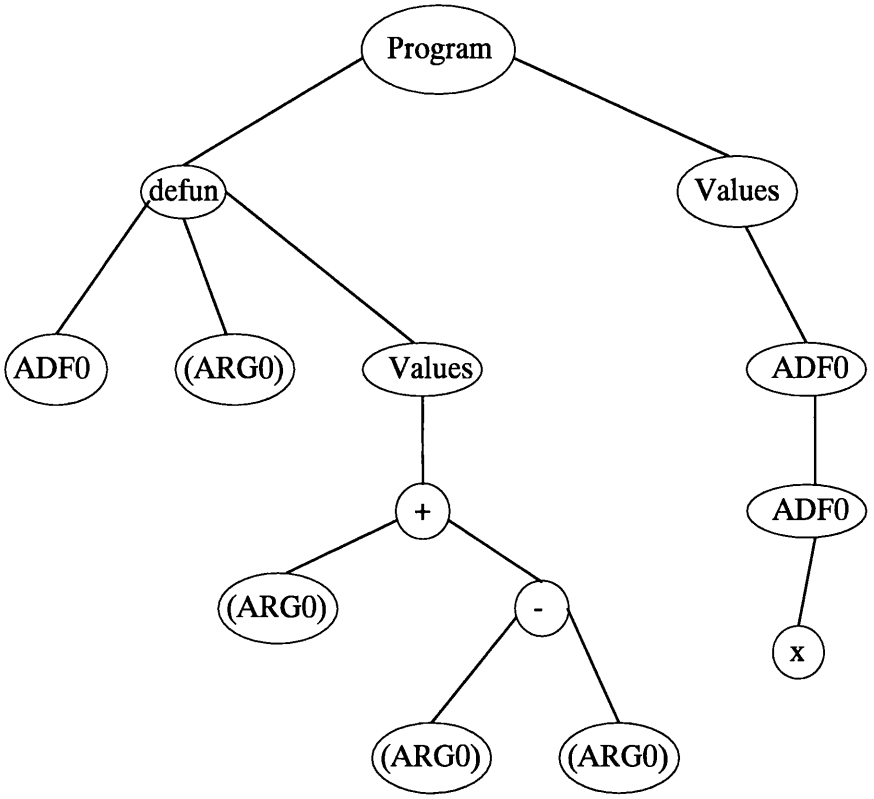


Figure 5.5: An example of an ADF program tree.

computer programs.

Also, GPs have a hierarchical character, which permits problems to be solved even in the presence of inaccurate, inconsistent or incomplete information (data). Genetic methods have the ability to be modified incrementally, and are robust. They are highly amenable to parallelisation, and they are fault tolerant. Genetic methods in general, allocate future trials in the search process in a near-optimal way.

As to precision, GP is able to adaptively change the size and shape of the solution dynamically during the run and thus hope on a solution with increasing precision.

5.7 Discussion

We have presented the basis of a GP. It seems logical to expect that a genetic program can be implemented so that it searches the problem space to achieve the desired visual homomorphism described in Section 3. In other words it will search the space of possible variations of sets of feature functions that produce our visual structure, and pick up the one (set of functions) which best represents our data set according to the fitness function given. In the case where rows of the data set are considered individually (thus an individual fitness case), the GP will search and

find the best possible set of functions that when applied to every row produces values for every feature of the visual structure to create it, and therefore create its characteristics so that it best represents the value system of that row. Therefore at the end, if N rows of data exist, there will be N visual structures to represent them, so that the visual homomorphism is attained. It is important to note that each individual genetic program consists of a number of GP trees (the set of different functions) that will be evolved in parallel. In fact there should be one GP (one function) for each feature of our visual structure. A complete set of them will result in rendering the visual structure.

In the case of using ADFs this parallelisation comes free. Each ADF can also be treated as a separate entity, without any interaction with the result-producing branch and hence, a separate GP program. For example in the case of the visual structure being the human face, the main component might correspond to the Left and Right (if symmetry of face is assumed) Zygomatic Major contraction value, the first ADF to the Left and Right Frontalis Inner and so on.

In fact, such a system has been implemented for the purpose of this thesis, the experimentation of which will be the subject of the next chapter.

Chapter 6

Experimentation and Results

The development of any new method requires examination of its effectiveness and validity. Several tests of the method were performed to assess its effectiveness for certain tasks, in addition to establishing whether the algorithms used were appropriate. This Chapter will focus on describing these experiments and their results, with relevant background information where required.

Initially, the procedure described in Chapter 3 will be illustrated by means of an example which utilises a set of three circles as the visual structure. It is a pilot study designed to test the method under controlled conditions. The data used, is simulated financial data, the properties of which are known. Also the visualisation outcome is intuitively obvious. For example, one can see that the set of circles in Figure 6.1 is more ordered than the set illustrated in Figure 6.2. The results from the user experiment of the method are then reported [LS99].

6.1 First Experiment - Set of Circles

6.1.1 Setting Up of the Experiment

Firstly, the dataset to be visualised must be obtained. Consider a simulated multivariate financial data set. Table 6.1 shows an example of such data, with its variable names shown at the top. It has 8 variables and 100 cases. Cases are individual companies and variables are different measurements for each company in a certain period of time (e.g. year 1999). The data is simulated so that it falls into clusters, with each cluster having the same number of companies. Details of the clusters are described later on. We chose simulated data because in this way, expected results are known and hence it is easier to evaluate user's responses.

What follows is a description of the steps required to run the GP. Such a run produces the automatic mapping from the data to this particular visual structure of three circles.

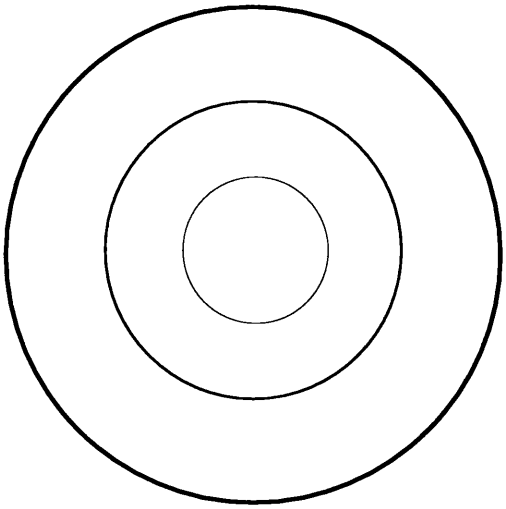


Figure 6.1: Picture of a "healthy" company.

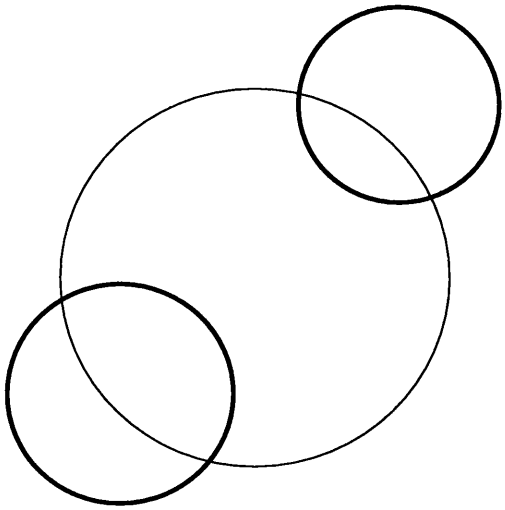


Figure 6.2: Picture of an "ill" company.

6.1.1.1 Step 1

For this example, there are 12 feature functions, $f_t(X)$, $t = 1, 2, .., 12$. These 12 functions will correspond to the 12 features $\phi_t(\Omega)$, $t = 1, 2, ..., 12$ that determine Ω, an individual set of three circles, allowing this set to be rendered. The twelve features that uniquely determine an individual set of three circles are:

- Six values for the three pairs of x, y coordinates representing the position of the centre of each of the three circles.
- Three values representing the sizes of the three circles, their radii.
- Three values representing the grey scale value, from 0 to 1, for the colour of the circles. Zero is black one is white.

From the GP point of view, each of these features is a different function (ADF[1]) over the variables of the data set, with all functions evolving in parallel. So when the variables of an individual row (company) in the data set are applied to any of these functions, the function returns a value that gets assigned to the corresponding feature in the visual structure Ω.

6.1.1.2 Step 2

The three measurements over the data set (value system) that are assumed to be of interest to the user, for this example, are the following $\nu_s(X)$, $s = 1, 2, 3$:

$$Profit = \frac{earnings}{costs} = X_1 \tag{6.1}$$

[1]Refer to Chapter 5 for details of ADFs and modular programming

C	No. of shares	Earnings	Dividend payout	Liquid Assets	Liabilities	Costs	Sales	Staff
1	0.12	0.25	0.70	0.60	0.61	0.21	0.20	0.11
2	0.34	0.15	0.24	0.82	0.44	0.74	0.31	0.05
.	.	.	.	.	.	.	.	.
N	.	.	.	.	.	.	.	.

Table 6.1: Example of the simulated data set.

$$Dividend\% = \frac{dividendpayout}{earnings - costs} * 100 = X_2 \tag{6.2}$$

$$Liquidity = \frac{liquidassets}{liabilities} = X_3 \tag{6.3}$$

Furthermore, we need a scoring system for these measurements, in other words to be able to quantify these. Each of the companies is scored on a scale of 0 to 100 under each category. The score is based on what financial experts (financial ratios) consider a reasonable value in each case [GU94]. The scoring system was kept simple since it does not affect the operation of the method:

Profit(ν_1) Ranges from $[0, 100]$, so that when earnings are at least five times the costs then a score of 100 is given, if it is equal or less than the cost, then a score of 0 is given. Intermediate values are linearly interpolated.

Dividend(ν_2) %: Ranges from $[0, 100]$, so that when the ratio is 30% or higher then a score of 100 is given, if it is 0% then a 0 is given. Intermediate values are linearly interpolated.

Liquidity(ν_3) It also ranges from $[0, 100]$, so that when the ratio of liquid assets to liabilities is 1.5 or more then a score of 100 is given, if 1 or below then a score of 0 is given. Again the intermediate values are linearly interpolated.

6.1.1.3 Step 3

Aspects of this visual structure that are measurable and significant to human perception $e_s(\Omega)$, $s = 1, 2, 3$ need to be identified and scored. For this particular example, the following aspects in order of importance are considered.

- The distance between the centres of the circles - mapped to profit. The closer to concentric the circles are, the more profit is assumed to be indicated.
- The proportionality of the radii - mapped to dividend payout %. The closer to an arithmetic progression the radii of the circles are the better the dividend payout % is.

- The proportionality of the colour - mapped to liquidity. The closer to an arithmetic progression the grey scale colours of the circles are, irrespective to which circle they belong, the better the company is performing in terms of profitability. For example a white circle, a grey circle and a black circle indicate that the company is performing better in terms of liquidity, rather than two white circles and a black one.

As with the important aspects of the data set, a scoring system needs to be devised in order to quantify these important aspects of the visual structure. All three of them are also given a value in the range 0 to 100.

1. e_1: The maximum distance between every possible pair of centres of circles. We use a (100 $\times$ 100) grid.

2. e_2: How far the second biggest circle is from the mean of the other two, in terms of their radii (i.e. if their radii follow an arithmetic progression). If they coincide then a score of 100 is given. Intermediate results are linearly interpolated.

3. e_3: How far the grey scale colour of the second biggest circle, in terms of grey scale colour, is from the other two. In other words whether their grey scale colours follow an arithmetic progression. If they coincide then a score of 100 is given. If at maximum distance then a 0 score is given. Intermediate results are linearly interpolated.

The characteristics defined above show that a "targeted" set of three circles with symmetry for both radii and colour will be more 'ordered' or less 'disordered', compared to a set of circles with their centres at maximum distances between them, proportionally meaningless ratio of radii and values of colour not changing in a smooth fashion. The former should be mapped to a company that is performing really well (relatively) on all aspects that are important to the hypothetical user (Profit, Liquidity and Dividend Payout) and the latter to a company performing relatively badly on all three aspects. In fact given "perfect" examples of the above EVA should produce aesthetically extreme cases out of the range of possible visualisations. Figure 6.1 shows an example of a "healthy" looking company, whereas Figure 6.2 shows an example of an "ill" one.

6.1.1.4 Step 4

The minimisation function in this case is:

$$\sum_{d=1}^{100}\sum_{s=1}^{3}(\nu_{ds} - e_{ds})^2 \tag{6.4}$$

where s are the three measurements and d the number of cases in the data set.

6.1.1.5 Step 5

This final step involves finding the suitable GP parameters for this problem. These parameters, as with almost all GP problems, are chosen after empirical testing. There is no set of rules defining what are the best parameters for individual cases, just rules of thumb according to the complexity of the problem you are dealing with.

A population size of 750 running for 150 generations is the selected choice. The creation type is a ramped half-half, with a maximum depth of creation 7. The terminal set consists of the 8 variables shown in Table 6.1, and the function set consists of the four arithmetic operations $(+, -, \%, *)$ where % represents protected division. Protected division takes care of division by zero case. Tournament selection is used for mating with a tournament size of 6. In generating subsequent populations 90% of the times crossover was used, 5% of the times swap mutation and 5% of the times shrink mutation. The best of the previous population was always added to the next population thus making sure the GP ends up with the best of all populations individual. Finally generational state GP was used for this experiment.

During the process of implementing the method, it was noted that the initial population was not equally distributed in the entire search space. Specifically, the results of the first population indicated that circles had a tendency to lie in the middle of the graphical area, with similar radii and similar colour, with few exceptions. It appeared, therefore, that the standard crossover technique was not performing optimally, as only small steps were being taken, in addition to taking a long time to move to different areas of the search space, with some of the space possibly left unutilised.

Two possible solutions were considered for the problem highlighted above. The first involved forcing a greater spread of the initial population in the search space and the second applied a more aggressive crossover for the first generations, in order to generate bigger steps in the space, thus exploring more of it. The second solution was implemented, to address a rapid crossover from generation to generation. In the first 20 generations a crossover to four out of the twelve trees constituting an individual (one tree for each feature) was performed, in the next 30 generations a crossover of two trees, and in the remainder a crossover of one tree was performed.

Illustrated here are results from the GP. Figure 6.3 shows, by generation, the standardised fitness of the best-of-generation individual, the average-of-generation and the worst-of-generation individual in the population for all generations 0 to 150 of one run of this problem. This run of the GP was chosen as the best one, out of a pool of 15 different runs. The value for fitness is calculated using equation (6.4). As can be seen, the error measurement of the

best-of-generation individual generally improves (i.e. decreases). The results shown in Figure 6.3 are from the same run of the program that produced the visual structures for the experiments described below. Appendix D shows an example of a tree evolved from the best run of the GP.

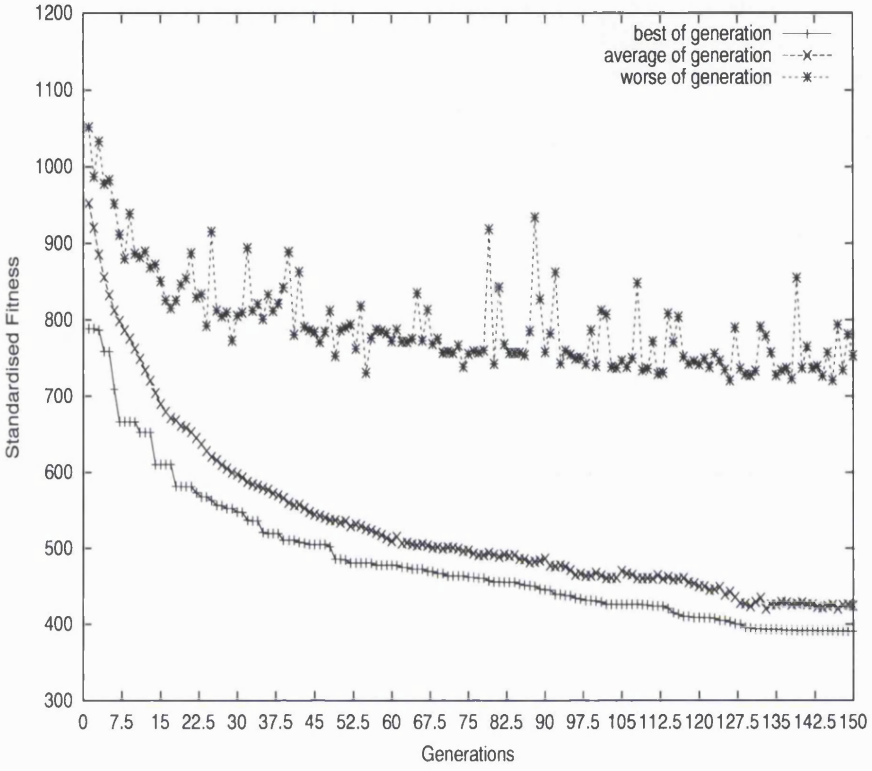


Figure 6.3: Fitness curves for one run of the GP.

6.1.2 Details of the Experiment

Three classification tests were run with the 10 subjects for each one of them. The subjects were postgraduate students of the University College London with no knowledge of the method prior to running the experiments. Moreover, they had limited knowledge of finance and financial data.

In the **first test**, the subjects were asked to classify 60 visual structures into 3 clusters without any knowledge of how the visual structures were produced. The three clusters are:

1. Companies that are performing relatively well in terms of profitability, dividend payout and liquidity.

2. Companies with average performance in terms of profitability, dividend payout and liquidity.

3. Companies that are performing relatively badly in terms of all three above importance functions over the variables.

As mentioned above, the visual structure is a set of three circles on a (100×100) grid, each circle having arbitrary radius and a grey scale colour.

The motivation behind this test was whether the subjects would identify the qualitative measurements of the set of three circles without being told about what they were. Success in such an experiment strengthens our case that the characteristics we have chosen resonate with human perception, for this particular case (a set of three circles). Furthemore, success in this experiment, will give support to the hypothesis that the method can be used to generate visual structures that can convey meaning intuitively. Inherited from the above, the test was used to test the accuracy with which the test subjects could extract information from a set of three circles.

The **second test** involved performing the same classification all over again for the same data set and the same people. However, this time the subjects were actually told what the important characteristics of the visual structure were and a brief summary of how the mapping was produced, but they were not told of the outcome of the previous test. The 60 visual structures used for the purpose of this test were the same as in the first test, but in a different order to the one given for the first test. This detail was not given to the users, and in fact no one realised that it was the same visual structures in a different order.

We wished to investigate, whether knowledge of the user of the characteristics of the visual structure that are significant, would have significant advantage on the results. Moreover, we wished to assess the effectiveness of the method in conveying information for this particular example.

In the third and **final test** 100 visual structures were used from a larger data set. Here, two more clusters were added, making the total five.

4. Companies that are performing relatively well in two out of three 'value system' variables (profitability, liquidity and dividend payout) and moderately on the third.

5. Companies that are performing relatively well in one of the three important features of the data set and moderately on the other two.

Clearly, there is a only a slight difference and big overlap between some of the clusters ($1-4$ and $2-5$ for example). The main reason of this experiment was to investigate if the subjects would recognise these subtle differences.

The simulated data was produced bearing in mind the five clusters above, with an equal number of companies in each cluster. Since there is an equal number of companies in each cluster there is an equal probability for a visual structure to belong to any of the three clusters. However, this information was not given to the subjects for obvious reasons. Times were also recorded for all three experiments.

Before moving on to the results of the experiment, by just looking at the selected 10 sets of circles of Figure 6.4, can the reader cluster each of the sets into the 5 different categories mentioned above? Moreover, is it possible to rank the sets from 1 to 10?

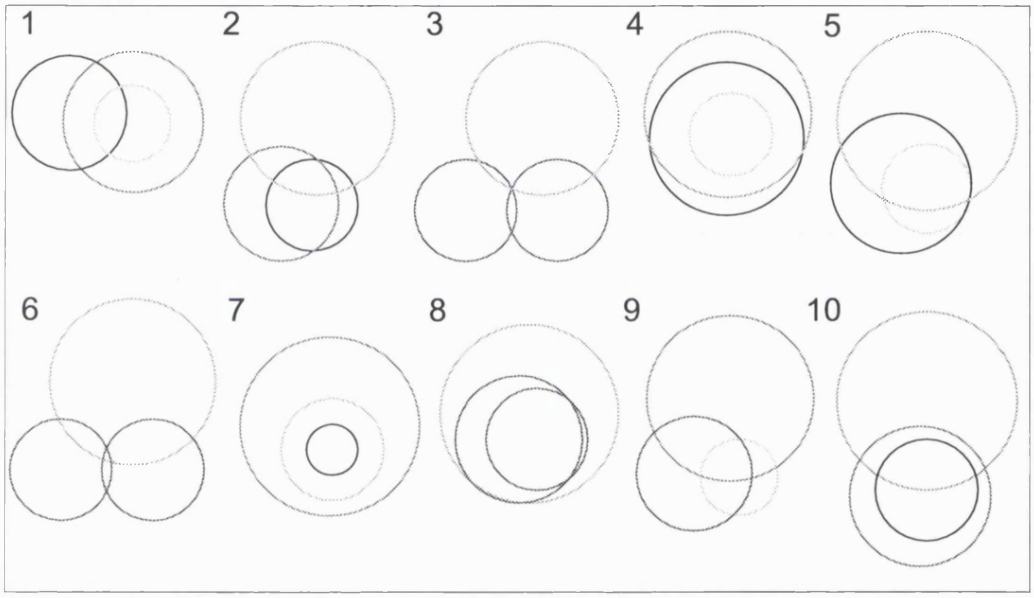


Figure 6.4: Ten Examples of sets of circles produced by the method.

6.1.3 Results

The performance of this method was compared to what a random classification would produce. In the first two tests a $\frac{1}{3}$ probability of guessing the right cluster from a random technique would be expected and, therefore, 20 'hits' (since there are 60 visual structures). A 'hit' represents a correct response from the subject. For the final test a probability of $\frac{1}{5}$ 'hits' could be expected, as there are 5 clusters with an equal probability of occurrence. So, there should be 20 hits once again since the number of visual structures for this test is 100.

For the first test a hypothesis test was performed on the number of successes the subjects had against the expected number of successes that would have been randomly produced. It can be confidently concluded (at $p = 0.005$) that the important aspects of the visual structures used, can perform significantly better than a random technique, even when the users are not being

informed of those aspects. On average, the 10 subjects responded correctly 34 times out of 60 with a standard deviation of 3.39. Therefore the success sample mean is 0.57 against 0.33 with the random technique.

The same hypothesis testing was performed for the second experiment. The results were improved further and from the new hypothesis testing it can confidently concluded (at $p = 0.005$) that the method performs significantly better than a random technique. This time subjects had, on average, 41 right responses out of 60 with a standard deviation of 3.46, and therefore the success sample mean increased to 0.69. A random example of the differences in the values for a particular user is shown in Figure 6.5. Figure 6.6, illustrates results from the same user. The values on the $x - axis$ are the actual versus expected clusters and the $y - axis$ represents the number of occurrences for this particular subject. For example "$3 - 3$" is the set where the user got it right, the company belonged to the third cluster and it was classified accordingly, whereas "$1 - 2$" is the set where the expected cluster for the company was 1 and the user classified it as type 2, and vice versa.

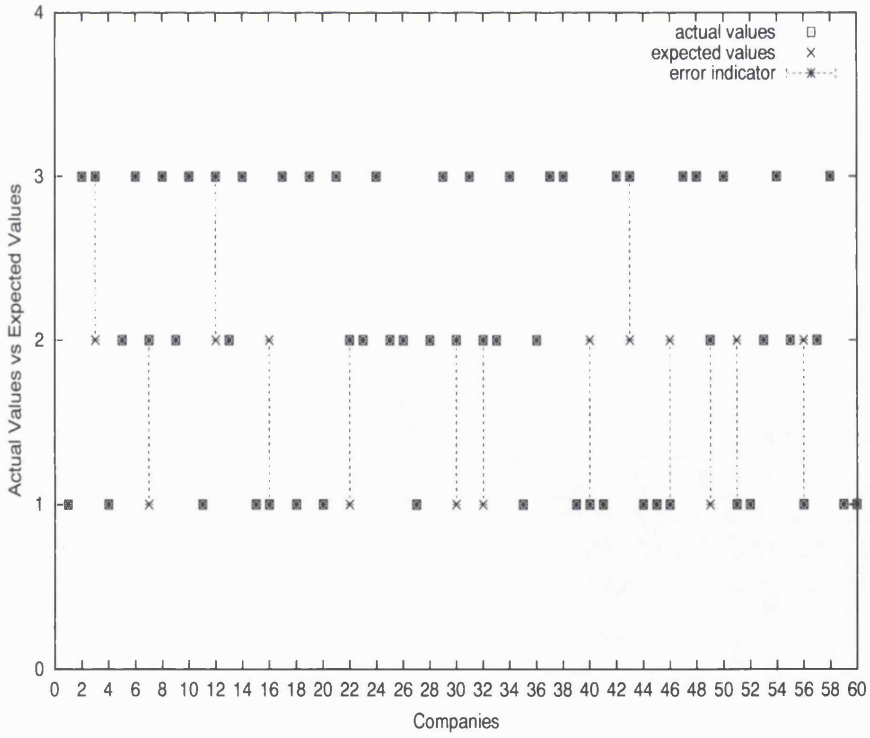


Figure 6.5: An example of actual data vs user responses from second test.

For the final test the same hypothesis testing was performed. In this experiment the success rate was lower than before with a mean of 0.46, or an average of 46 right guesses out of a 100 with 4.3 the standard deviation. The diminished mean value is a direct consequence of the fact

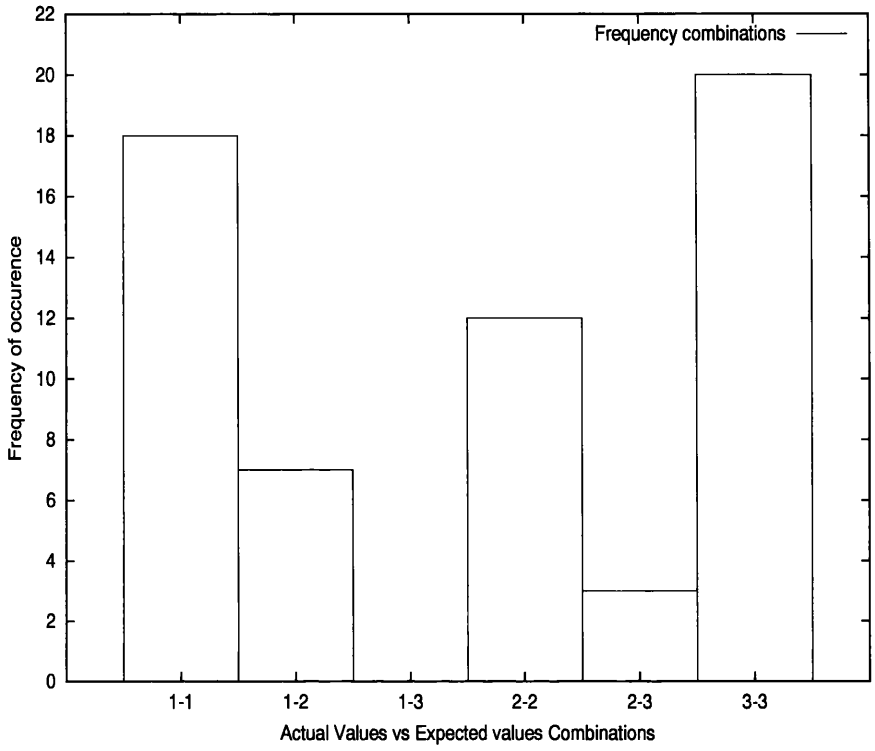


Figure 6.6: An example of actual data vs user responses as a histogram.

that it is hard to distinguish between clusters 1 and 4, or clusters 2 and 5. In fact Figures 6.7 and 6.8 show that for almost all cases the misjudgement from the user occurred for the pairs from these sets. Figure 6.8 clearly shows that the number of times this particular user confused clusters 1 and 4 or clusters 2 and 5, is very high indeed. However, one can still confidently conclude (at $p = 0.005$) that the method is performing much better than a random technique, which would be expected to have a success sample mean of 0.2.

Going back to the question posed at the last paragraph of the previous Section 6.1.2, a detailed analysis of the answer can now be given. Referring back to Figure 6.4, the extreme cases can easily be spotted. These are companies that are performing generally well or generally badly for the measurements of interest in the data described at Step 2, Section 6.1.1. By looking at the same sets of circles from left to right, top to bottom, the fourth set first row (4) and the second set of circles second row (7), can be spotted as the "best performing individuals". Or by looking at the third set first row (3) and the first set of circles in the second row (6), we can label them as the "worst performing individuals". Both sets under this category have relatively big maximum distance between every pair of circles, their radii proportionality do not follow an arithmetic progression and neither do their grey scale colour. In fact the former set of circles were produced by EVA as a mapping from data of relatively well performing companies,

whereas the latter set of circles, were produced by EVA as a mapping from data of relatively badly performing companies.

A less obvious point to be made out of the selected images of Figure 6.4 is that of the small difference between clusters 1 and 4, or clusters 2 and 5 already mentioned above. The third set of circles in the second row (8) at a first glance show little differences with the fourth set of circles first row (4) that was already classified as generally performing well individual in all aspects of the measurements of interest. Both sets have nearly concentric circles and their proportionality of their radii seems to be almost perfect. However, by looking more carefully at the third set of circles second row one can see that the grey scale colour of the three circles clearly does not lie on an arithmetic progression as described at Step 3 of Section 6.1.1 whereas for the fourth set, first row performs better under this aspect.

Moreover, there also seems to be little difference between the set of circles second first row (2) and fourth second row (9). Both sets seem to be on the average side in terms of maximum distance between every pair of circles and proportionality of their radii. However, the fourth set of circles second row seems to be performing better in terms of proportionality of grey scale colour. In fact this set was created by EVA as part of mapping from data of a company "belonging" to cluster 5 whereas the other set was created as part of mapping data of a company "belonging" to cluster 2 and hence a company performing relatively moderate in all three measures of interest.

Finally, by looking at the same Figure 6.4 labelling each set from 1 to 10 from right to left, top to bottom, the following ranking can be given to them based on how well they performed: $7, 4, 8, 1, 5, 9, 10, 2, 3, 6$. Small changes are indeed debatable. As a matter of fact 7 and 4 were produced from companies in cluster 1, 8 and 1 from companies in cluster 4, 5 and 9 out of cluster 5, 10 and 2 out of cluster 2 and finally 3 and 6 from companies "belonging" to cluster 3.

6.1.4 Discussion

Presented above are results from performing an initial user experiment of the method. Questions such as, "can subjects extract information from the visual representations of the data set?", or "can the visualisation act as an aid to the decision making process?", can be answered positively, at least for such a small number of variables. The results also indicate that the method can be used, prior to statistical analysis, perhaps as the first step of a classification process. If one considers, for example, the cases of an undergraduate admissions office or a bank offering loans to customers, this method could be adopted to produce a visual structure for each of the applicants and then trivial acceptances and trivial rejections could be identified rapidly and

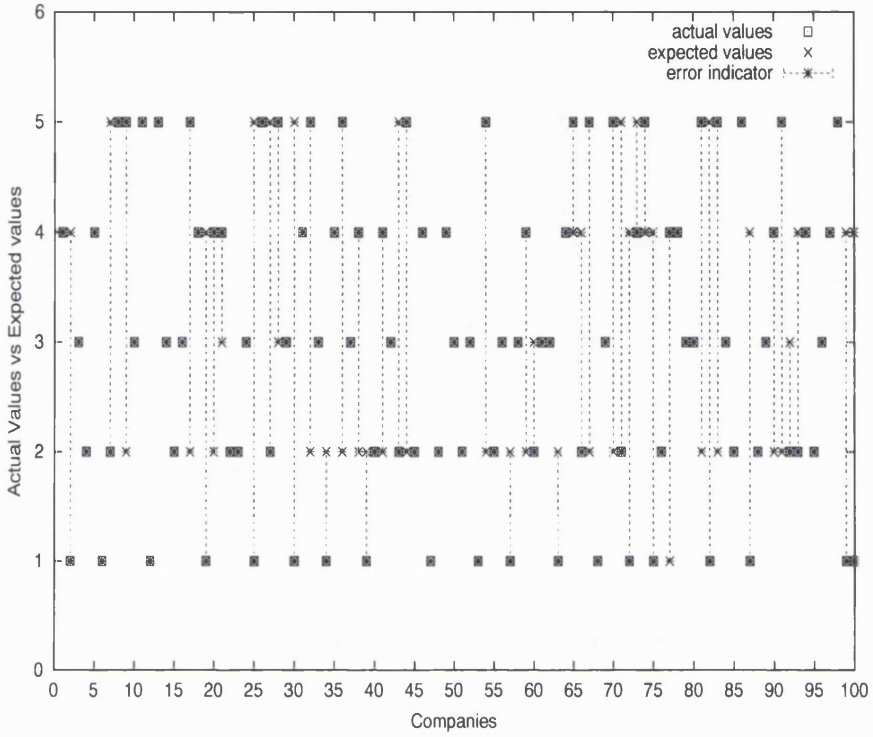


Figure 6.7: An example of actual data vs user responses from the third test.

with confidence. There will be applicants, however, requiring further investigation. As can be seen, for example, in Figure 6.6 or Figure 6.8, some particular users always recognised the "ill" performing companies (the values for the set "3 – 3" is 20 in both cases as it should be) which, in the case of financial data and investors for example, is of critical importance to avoid bad investments.

In addition, the results of the first two tests in this experiment showed a rational decision based on the characteristics of the set of three circles that are significant to human perception. In fact, apart from one test subject, who had a totally different understanding of what is good or bad (thinking the more screen space the circles were occupying, the better), the subjects' understanding corresponded well to our choices. So *empathetic visualisation* appeared to be successful on the whole.

A final note to be made, concerns the time taken for the users to make decisions during the first experiment. The test subjects used, although being of limited financial knowledge, took very little time to decide on the clustering of each of the companies. Although time was not a prime measurement in this experiment, it was easily recognised as a clear advantage of this particular method.

Although circles have been used in this first pilot study, the main goal was to use and inves-

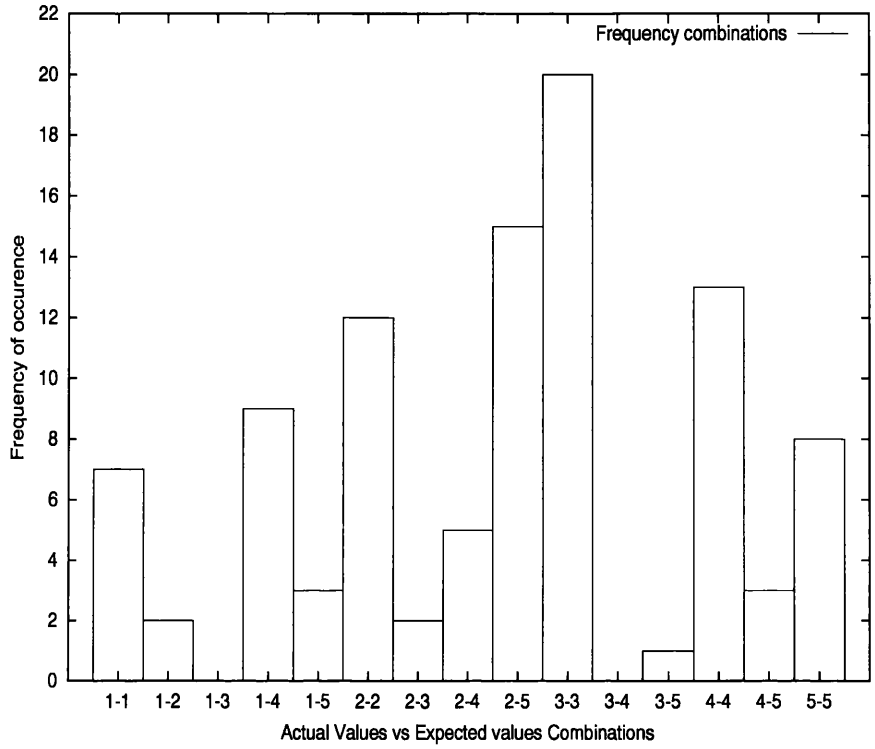


Figure 6.8: An example of actual data vs user responses as a histogram.

tigate the potential naturalistic visual structures. Despite the fact that qualitative measurements were utilised over the set of circles in a meaningful way, this particular visual structure is still very much abstract.

Therefore, the next step is to test the method with a more natural representation, the face, which is considered to be the epitome of such structures, as well as with real financial data. The familiarity of people with faces, the fact that it is fast and relatively easy to recognise salient features and abnormalities and the fact that a face evokes an emotional response in us, creates potential for further improvement of the method. Returning to the example of the bank offering loans, each applicant may be considered as being represented by a human face. In addition, each face may represent the emotional state of the banker after having analysed the applicants credentials. A "happy" face of course could mean that the loan will be granted, whereas "sad" or "angry" faces should lead to rejection of the applicant. Mixed emotions encapsulated in a single face (applicant) should lead to further investigation.

6.2 Second Experiment - Faces

This second experiment has two main targets. Firstly, to see if EVA can be used with real data and secondly whether using a more naturalistic visual structure such as a face improves the

results even further.

Figure 6.9: Sample faces produced by the method.

6.2.1 Setting Up of the Experiment

Description of the real data gathered for the purposes of this experiment is given below. An application of real data to EVA should give good feedback to effectiveness of the method. However, the basics of accounting and finance are needed to understand the details of the data gathered better [GU94].

The data for 150 companies were collected from "DATASTREAM ADVANCE". They include variables from a *Profit and Loss Account* and the *Balance Sheet* items. The data collected concerned organisations that were public companies and excluded partnerships, sole-traders as well as non-profit and governmental organisations whose motives, policies and charters might have shown significantly different results.

The 10 variables upon which the analysis is focused are the following:

1. *Ordinary Share Capital:* Share in the issued capital of a company which are held on terms that make the holder a "member" of the company entitled to vote at annual meetings and elect directors, as well as participate through dividends in the profits of the company.

2. *Equity Capital and Reserves:* The sum of equity capital and other pools of capital gener-

ated through the sale of shares at a premium (Share Premium), revaluation (Revaluation reserve), etc.

3. *Fixed Assets:* Any non-financial capital asset of the company which is relatively long-lived and specific to particular productive processes, and the cost of which is normally recoverable only over an operating period of some duration, e.g. plant and buildings.

4. *Current Assets:* These are main components of current assets: stock, accounts receivable or short term debtors, cash and other liquid resources. The size of current assets is, particularly in relation to other financial indicators, a main indicator of the liquidity of the company.

5. *Current Liabilities:* Sum of company's debts that have to be settled within the subsequent year. Measure of the company's short term (immediate) obligations.

6. *Long Term Liabilities:* Obligations of the company that are not immediate in the subsequent year.

7. *Preference Share Capital:* Shares in a company which rank before the Equities for a payment of a dividend, which is usually a fixed percentage on the nominal value of the share. Most have no voting rights attached.

8. *Earnings Before Interest and Tax*(EBIT)

9. *Profit after Tax*

10. *Ordinary Dividends:* Payments to shareholders in a company in the form of cash.

Potential users of financial statements are: shareholders, banks and other capital providers, potential investors (with or without much knowledge of financial information), employees, creditors, and the government. Each user has different needs and has different yardsticks to perform analysis and assessment of the financial data.

6.2.2 Details of the GP

6.2.2.1 Step 1

A face based on an underlying muscle model, was developed by Waters [Wat87]. This allows a variety of facial expressions to be produced by controlling the underlying musculature of the face as described in Section 4.1.

Waters work has been the basis used to construct the face for this experiment. In the existing implementation there are 18 muscles. However, by assuming symmetry of the face

for the left and right side of it, the number of facial features becomes 9. So, for this particular example, 9 feature functions $f_t(X)$, $t = 1, 2, .., 9$ are needed. These correspond to the 9 facial features, the muscle contractions, $\phi_t(\Omega)$, $t = 1, 2, ..., 9$ that determine Ω, an individual face. The 9 muscles that determine an individual face are (for each one of them there is a left and right muscle):

- Zygomatic Major
- Angular Depressor
- Frontalis Inner
- Frontalis Major
- Frontalis Outer
- Labi Nasi
- Inner Labi Nasi
- Lateral Corigator
- Secondary Frontalis

6.2.2.2 Step 2

The three measurements over this data, are the following:

$$ProfitoverTotalAssets = \frac{EBIT}{(FixedAssets + CurrentAssets)} \tag{6.5}$$

$$CurrentRatio = \frac{CurrentAssets}{CurrentLiabilities} \tag{6.6}$$

$$GearingRatio = \frac{(LoanCapital + PreferenceCapital)}{EquityCapitalandReserves} \tag{6.7}$$

The three variables that are of interest to us are based on *Financial Ratios*. The ratios chosen cover a big enough area to represent important aspects of the financial state of a company. Financial ratios have their limitations and these limitations should be considered by users who base their decisions upon these figures. Ratios are not definitive; they are only a guide. Interpretation needs careful analysis and ratios should not be considered in isolation of the financial data or the relationships with other ratios. Therefore, they make a good candidate for EVA.

The financial ratios (calculated from the above data) upon which the financial situation of a company will be assessed are:

Profit over Total Assets(ν_1) This ratio measures profitability and of how effectively assets are used in the aim of maximising profits. A high figure is preferable to a lower figure and as a rule of thumb, positive results of the ratio are preferred to negative results.

Current Ratio(ν_2) The current ratio assesses short-term liquidity. It gives an indication of the levels of liquid resources available in an organisation to cover its immediate liabilities. As a rule of thumb, figures above unity (1) are preferred with 1.5 being the normal, and any figures above 2 usually indicate that a more than necessary amount of resources is tied up in liquid assets.

Gearing Ratio(ν_3) The gearing ratio measures risk and long term solvency. It is an indication of the amount of long term finances that the company has taken up relative to its capital base. For companies less is preferred to more.

Each company in the data set is given a score $0-100$ for each ratio. The convergence from the ratio to such a number was kept simple and the decision was affected by the data themselves. Maximum and minimum values were recorded for each ratio as well as the spread of the data itself before the scoring system was devised.

6.2.2.3 Step 3

The aspects of the visual structure $e_s(\Omega)$, $s = 1,2,3$ that are measurable and significant to human perception are based on the argument that facial expressions of emotion (such as fear, happiness, surprise, anger, sadness and disgust) are universal although the rules for display can vary from culture to culture.

Therefore, the following scales of expressions are taken into account:

- e_1 Degree of Happiness.
- e_2 Degree of Fear
- e_3 Degree of Anger.

As mentioned in Section 4.2.3, a method has been developed that enables automatic measurement of emotional expressions using user's subjective evaluations. The functions produced by these methods are used here as a substitute a user, to quantify the emotional expressions.

6.2.2.4 Step 4

The minimisation function is similar to before:

$$\sum_{d=1}^{150}\sum_{s=1}^{3}(\nu_{ds} - e_{ds})^2 \tag{6.8}$$

6.2.2.5 Step 5

The GP was run with similar parameters to the first experiment. A population size of 750 running for 150 generations was chosen. Again the creation type is ramped half-half with a maximum depth of creation 7. The terminal set were the 10 variables described in section 6.2.1 and the function set consists of the four arithmetic operations $(+, -, \%, *)$ where $\%$ represents protected division. Tournament selection is used for mating with a tournament size of 6. In generating subsequent populations, 90% of the times crossover was used, 5% of the times swap mutation and 5% of the times shrink mutation. The best of the previous population was always added to the next population (called elitism) thus ensuring the best of all populations individual remains.

Figure 6.10 shows by generation, the standardised fitness of the best-of-generation, the average-of generation and the worse-of-generation individuals in the population for all $0 - 150$ generations for a single run of the GP. Once again this run was chosen as the best one out of 15 different runs of the GP, and the value for fitness is calculated by equation (6.8). It is clear from the diagram that the fitness of the best-of-generation individual generally improves (i.e., decreases) by generation.

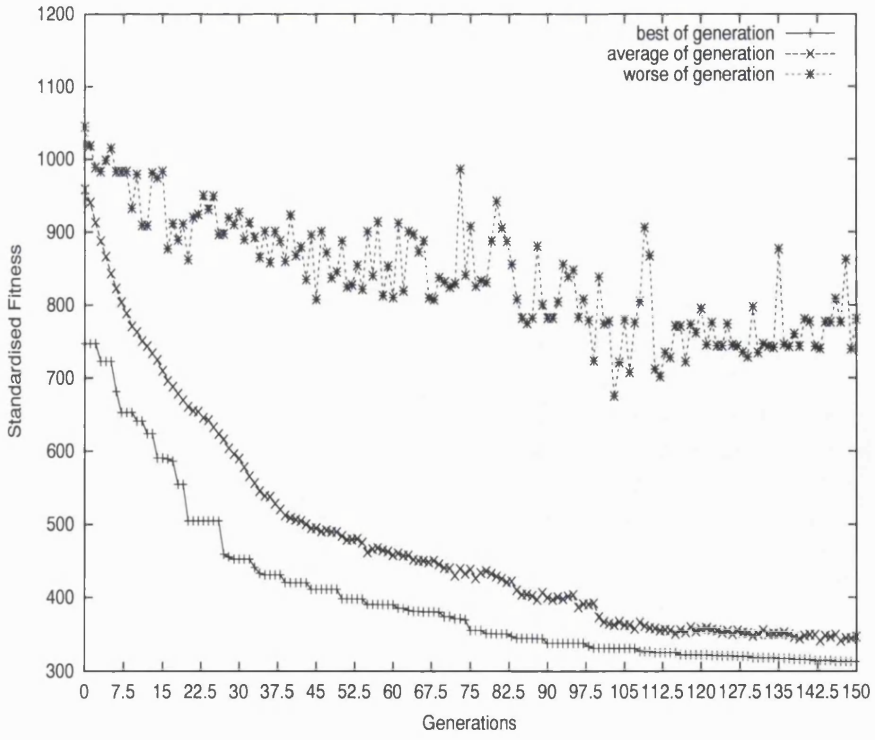


Figure 6.10: Fitness curves for one run of the GP.

Figures 6.9 and 6.12 show faces produced by the method in this experiment. There are

a number of emotions encapsulated in each face/company some of which are very obvious indeed. The bottom right face in Figure 6.9 shows a "healthy" looking company, whereas the fourth face in rows 2 and 3 of Figure 6.12 show "ill" looking companies, probably for different reasons. Mixed emotions are represented in other faces, namely the third face in row 2 of the same Figure.

6.2.3 Details of the Second Experiment

The same 3 classification experiments were run for this data but with different subjects. The subjects were postgraduate students at University College London with no knowledge of the method prior to running the experiment. The only difference to this experiment was having to classify each company (solely based on figures) to the 3 and then 5 categories described in Section 6.1.2. Since we are dealing with data rather than simulated data, in order to be able to measure the effectiveness of the method, we need to classify each company ourselves based on the data which involves subjective judgement. Details of the analysis performed on the data can be found in Appendix A.

However, for both the targets set this was a minor issue. It was still possible to get an indication of effectiveness of the method on real data as well as improvement on "natural" visual structures.

In the **first test** users were asked to classify 30 faces into the following three clusters:

1. Companies that are performing relatively well in terms of the financial ratios described in Step 2 of Section 6.2.2.

2. Companies that are performing moderately on the above financial ratios.

3. Companies that are performing relatively badly on all three of them.

This time a simple description on the effect each of these ratios has on every company was given. Subjects were told they are practically assessing profitability, short-term liquidity and riskiness of each company. They were then given a face for each company and they were asked to classify them accordingly.

Again the motivation was to test whether the right facial emotion was chosen to map our results, as well as accuracy of the subjects to extract information with very little knowledge.

The **second test** involved performing the same classification all over again for the same data set. However, this time the subjects were told of the facial emotions used and were given a brief summary of how the mapping was produced. The same 30 faces were used for this face in although in a different order.

As with the first set of experiments, we wished to assess whether knowledge of the characteristics in the visual structure improved our results.

In the **third** and final test users were asked to classify 50 visual structures in 5 clusters. We added two extra clusters in order to test the effectiveness of the method in finer classification. The added clusters were:

4. Companies that are performing relatively well in any two out of the three financial ratios and moderate on the third.

5. Companies that are performing relatively well in any one of the three important features of the data and moderate on the other two.

There is still the problem of little differences between some of the clusters, namely (1-4 and 2-5). The aim of this experiment was to test whether subjects would recognize these subtle differences. Figure 6.11 shows a number of such mixed emotions namely anger and fear for top left face, small happiness and anger for bottom face and so on.

Figure 6.11: Faces with mixed emotions produced by the method.

6.2.4 Results

For each experiment an equal number of companies for each cluster (10) was selected, as given by the analysis of the data, as described in Appendix A.

Performance of EVA was compared to what a random classification technique would produce. In the first two tests a $\frac{1}{3}$ probability of guessing the right cluster was expected from a random technique and therefore 10 'hits' (since there are 30 visual structures). A 'hit' represents the times the subject got it right. For the final test the probability of hits was expected to

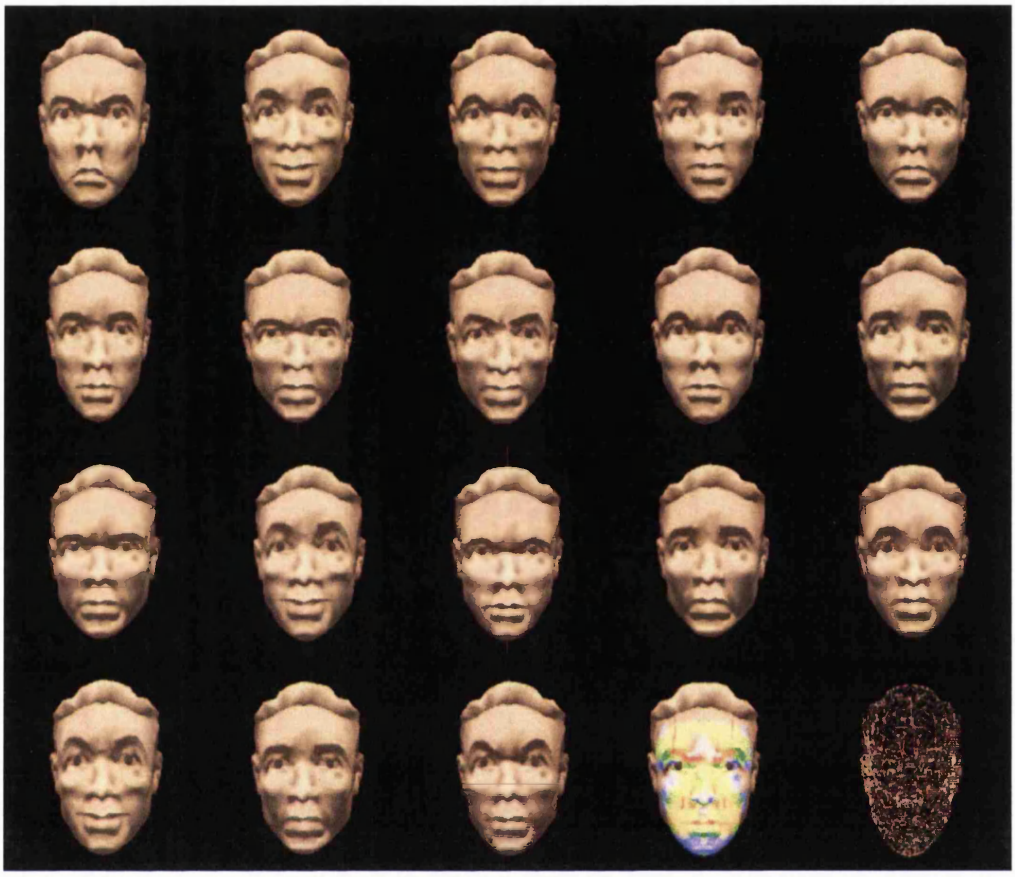


Figure 6.12: Faces produced by the method during the 2nd experiment. The last two faces in the last row are shown, the first in a toggle-muscle (see Appendix B) mode and the second in a line display model.

be $\frac{1}{5}$, since we have 5 clusters with an equal probability of occurring. So, there are 10 hits again since the number of visual structures for this test is 50.

For the first test a hypothesis test was performed on the number of successes the subjects had against the expected number of successes that would have been randomly produced. We can confidently conclude (at $p = 0.005$) that the important aspects of the visual structures used can perform significantly better than a random technique, even when the users are not being told of those aspects. On average the 10 subjects got it right 20 times out of 30 with a standard deviation of 2.58, resulting in a success sample mean of 0.67 against the 0.33 with the random technique.

The same hypothesis testing was performed for the second experiment. The results improved and from the new hypothesis testing we can confidently conclude (at $p = 0.005$) that the method performs significantly better than a random technique. This time subjects had on av-

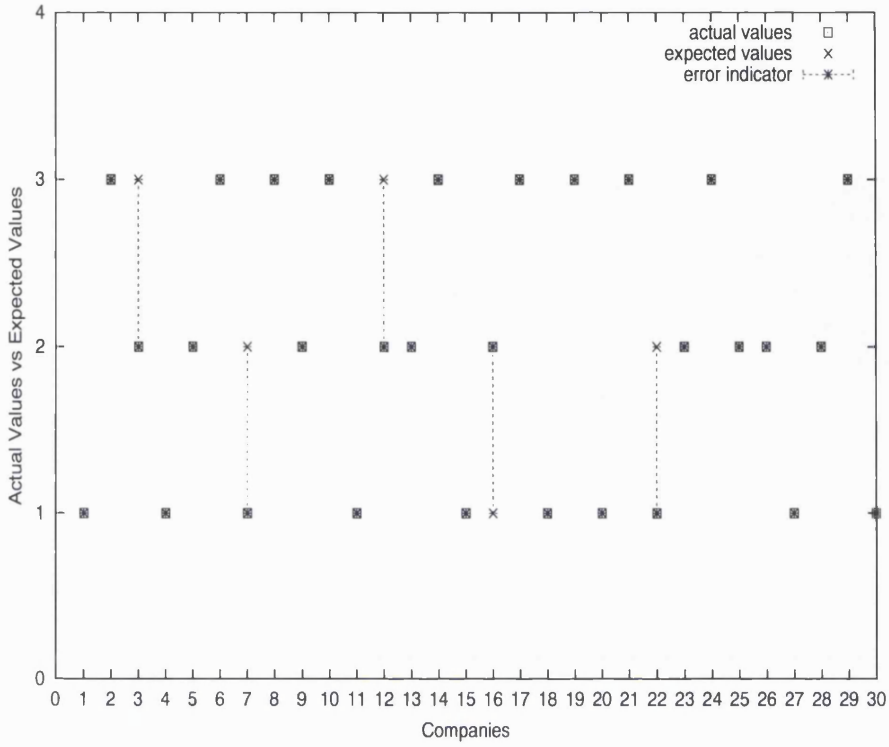


Figure 6.13: An example of actual data vs user responses from second test.

erage 23 correct responses out of 30 with a standard deviation of 1.89 and therefore the sample mean has increased to 0.77. A random example of the differences in the values for a particular user is given in Figure 6.13. It is obvious from the graph that the difference in responses occurred between clusters 1, 2 and clusters 3, 4. It is pleasing that no users classified a "healthy" looking company as "ill" and vice versa despite the few "errors" in the responses. Figure 6.14 shows results from the same user. The values in the horizontal axis are the actual versus expected clusters and for the vertical axis the number of occurrences for this particular subject. For example "$3-3$" is the set where the user got it right, the company belonged to the third cluster and it was classified accordingly whereas "$1-2$" is the set where the expected cluster for the company was 1 and the user classified it as of type 2 and vice versa. There were 4 such occurrences.

For the final test the same hypothesis testing was performed. In this experiment the success rate is a bit lower than before at a mean of 0.62, or an average of 31 correct guesses out of a possible 50 and a standard deviation of 4.32. Again it is obvious that the small decrease to the success rate is a consequence of the fact that it is hard to distinguish between clusters 1 and 4, or clusters 2 and 5. In fact Figures 6.15 and 6.16 show that almost all of the times the misjudgement from the user occurred between the pairs from these sets. Figure 6.16, clearly

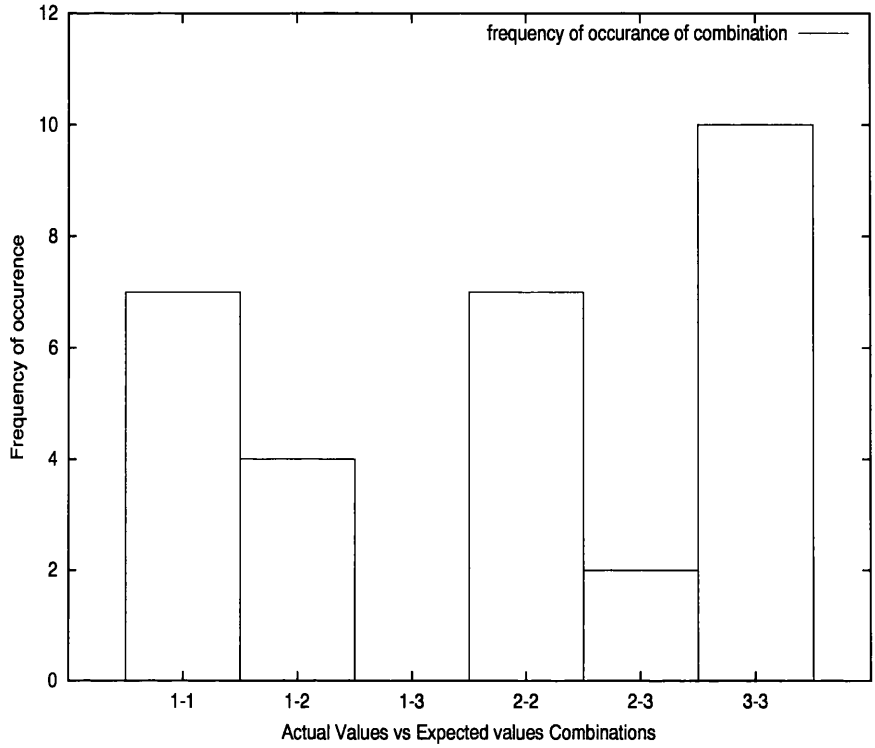


Figure 6.14: An example of actual data vs user responses as a histogram.

shows that the number of times this particular user confused the clusters 1 and 4, or clusters 2 and 5 is very high indeed. However, it can still be confidently concluded (at $p = 0.005$) that the method is performing much better than a random technique expected to have a success sample mean of 0.2.

6.2.5 Discussion

The results from this experiment indicated that the method can be applied to larger realistic data sets. Moreover, a quick look at the means from the first test of the first and second experiment (0.57 vs 0.67), gives evidence that naturalistic visual structures such as facial emotions as opposed to the more abstract set of circles, further improve the method. This evidence is strengthened when taking into consideration the complexity of this data set as opposed to the one used in the first experiment.

6.3 Third Experiment - Fear of Public Speaking

What follows is the analysis and results of a third experiment performed on the method. The data was gathered from a Virtual Reality (VR) experiment on Fear of Public Speaking (FOPS). Users were asked to give a talk while in a VR environment, in two different settings. During

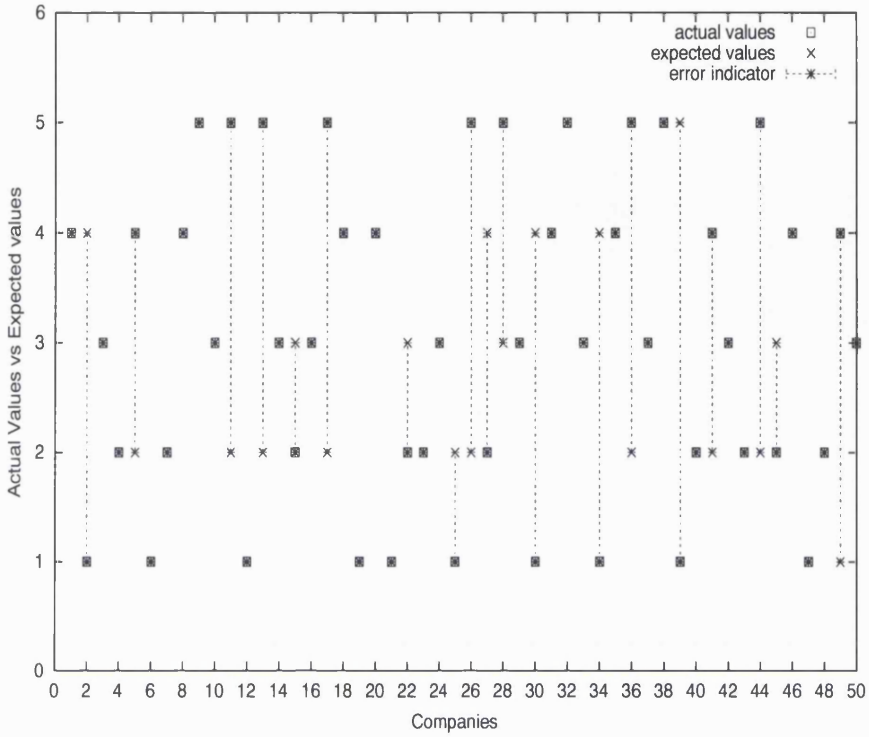


Figure 6.15: An example of actual data vs user responses from the third test.

the experiment, data was gathered based on physiological state (for example heart rate) and questionnaire responses. EVA was used on this particular data, with the aim of expressing the overall (known) emotions of each of the users during their talk.

This third experiment also uses faces as the medium of visualisation of our data set [LS03]. Therefore, each human face represents the facial emotions a specific user had when giving a talk to either a virtual audience or to an empty room.

However, this particular experiment involves no subjects and is used to test validity of EVA when compared to statistical analysis. Another objective is to test the method in a different context to that of the financial world.

6.3.1 Setting Up of the Experiment

The data set as mentioned already was gathered from a VR experiment on Fear of Public Speaking (FOPS) [DPP01a], [DPP01b]. The subjects were initially distinguished as male/female, phobic/confident and whether they gave their VR talk to an empty audience or to an audience of virtual people as shown in Table 6.2. Figure 6.17 shows one environment with virtual audience. There were a total of 40 subjects.

Here what is of interest to us is the difference in responses, of phobic and confident speak-

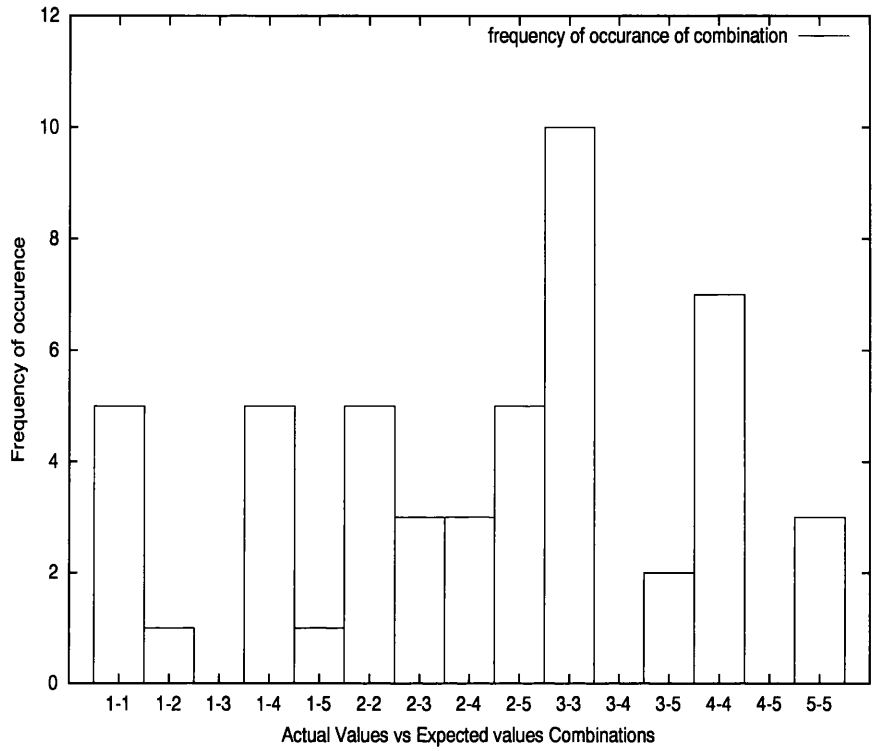


Figure 6.16: An example of actual data vs user responses as a histogram.

Condition:	Empty		People	
	M	F	M	F
Phobic	5	5	6	5
Confident	5	5	5	5

Table 6.2: Division of subjects into categories

ers, to both audience conditions. In particular a relatively large difference is expected in anxiety for phobic users giving talks to a virtual audience rather than an empty room, whereas, the difference in anxiety for confident users should be relatively small with regards to the same two conditions. In general, it is also expected that the phobic users are more anxious than the confident ones, regardless of the condition the speech was given.

The 13 variables used in this analysis are the following:

1. *Gender:* Male or Female subjects.

2. *Anxiety Level:* Phobic or Confident subjects.

3. *Fear of Negative Evaluation (FNE):* A questionnaire driven measurement of social evaluative anxiety [DW69]. It is measured out of 30, where 30 is a perfectly phobic response.

Figure 6.17: An environment with an audience of virtual people (avatars).

4. *State Trait Anxiety Inventory (STAI):* A questionnaire driven measure of trait anxiety [Spi83]. Anxiety scores are in the range of 20, minimum anxiety, to 80 representing maximum anxiety.

5. *Personal Report of Confidence as a Speaker (PRCS):* A questionnaire driven measurement used to screen subjects for the experiment [Pau66]. The PRCS is scored out of 30, where 30 represents the response of a "pure" phobic user. Subjects were accepted if they scored 8 or less (confident group) and if they scored 20 or more (phobic group).

6. *Audience Type:* Either empty or with a virtual audience.

7. *Presentations:* Zero or one indicating whether users do not give presentations at all or they sometimes give presentations.

8. *Subject:* A $1-7$ scale indicating subjects' comfort with the topic of their talk. A score of 1 indicates not at all comfortable.

9. *Prepared:* A $1-7$ scale indicating how prepared subjects felt for this talk, with 1 denoting not at all.

10. *Emotions before:* A $0 - 100$ score of confidence drawn out of 6 scales for adjectives (anxious, happy, confident, relaxed, sad, angry). Zero represents anxious, 100 confident.

11. *Emotions after:* Same as above but this time after the talk was given.

12. *Self rate:* A self assigned score out of 100 of their own performance.

13. *Somatic:* A questionnaire driven measurement for subjectively assessed the feeling of somatic responses. The list of somatic responses includes sweating, discomfort in stomach, heart palpitations, tremors, nerves/feelings of being scared, tightness in chest, tenseness and loss of balance.

6.3.2 Details of the GP

6.3.2.1 Step 1

The same nine feature functions (nine muscles) were used, as described in Section 6.2.2. Again symmetry of the visual structure is assumed hence there is a left and right muscle for each one of the muscles.

6.3.2.2 Step 2

The three measurements over the data that are of interest to us, are the following:

Somatic(ν_1) This measures subjectively the somatic responses of the user when giving the talk.

Modified Report of Confidence as a Speaker (MPRCS)(ν_2) This measures the degree of confidence of users after giving the talk.

Somatic×MPRCS(ν_3) This is a combination of the two measurements above, to allow for an interaction affect.

Each subject (row in the data set) was given a score $0 - 100$ for each of these three measurements.

6.3.2.3 Step 3

The same facial emotions and scoring system were used to represent our data set. Namely:

- e_1 Degree of Happiness.
- e_2 Degree of Fear
- e_3 Degree of Anger.

6.3.2.4 Step 4

The minimisation function is similar to before:

$$\sum_{d=1}^{40}\sum_{s=1}^{3}(\nu_{ds} - e_{ds})^2 \tag{6.9}$$

6.3.2.5 Step 5

The GP was run with similar parameters to the previous two experiments. A population size of 750 running for 150 generations was chosen. Again the creation type is ramped half-half with a maximum depth of creation 7. The terminal set were the 13 variables described in Section 6.3.1 and the function set consists of the four arithmetic operations $(+, -, \%, *)$ where $\%$ represents protected division. We used tournament selection for mating with a tournament size of 6. In generating subsequent populations, 90% of the times crossover was used, 5% of the times swap mutation and 5% of the times shrink mutation. The best one of the previous population was added to the next population thus making sure the best of all populations individual resulted. A generational state GP was used.

Figure 6.18 shows by generation, the standardised fitness of the best-of-generation, the average-of generation and the worse-of-generation individuals in the population for all 150 generations for a single run of the GP. This run of the GP was chosen as the best one, out of a pool of 15 different runs. The value for fitness is calculated by equation (6.9). It is clear from the diagram that the fitness of the best-of-generation individual generally improves (i.e., decreases) by generation.

6.3.3 Statistical Analysis

What follows is statistical analysis over the following response variables: *Modified Personal Report of Confidence as a Speaker* (MPRCS), *Self Rating and Somatic*, where MPRCS is a modified PRCS (Section 6.3.1) with total maximum count of 13 questions modified to refer to the talk just given by the subjects. Self rating is a score out of 100 with 100 being complete satisfaction of users with their talk. Somatic is measured out of 300 where a score of 300 indicates that the user was extremely aware/bothered by all the items in the list of somatic responses.

Table 6.3 shows the mean and standard deviation for the Modified Personal Response of Confidence of a Public Speaker variable. From a quick look at the data there is an indication of more anxiety for phobic subjects giving their talk to a virtual audience rather than an empty room, whereas this change of "audience" seems less significant for confident students. The data are analysed using logistic regression which is equivalent to Analysis of Variance (ANOVA)

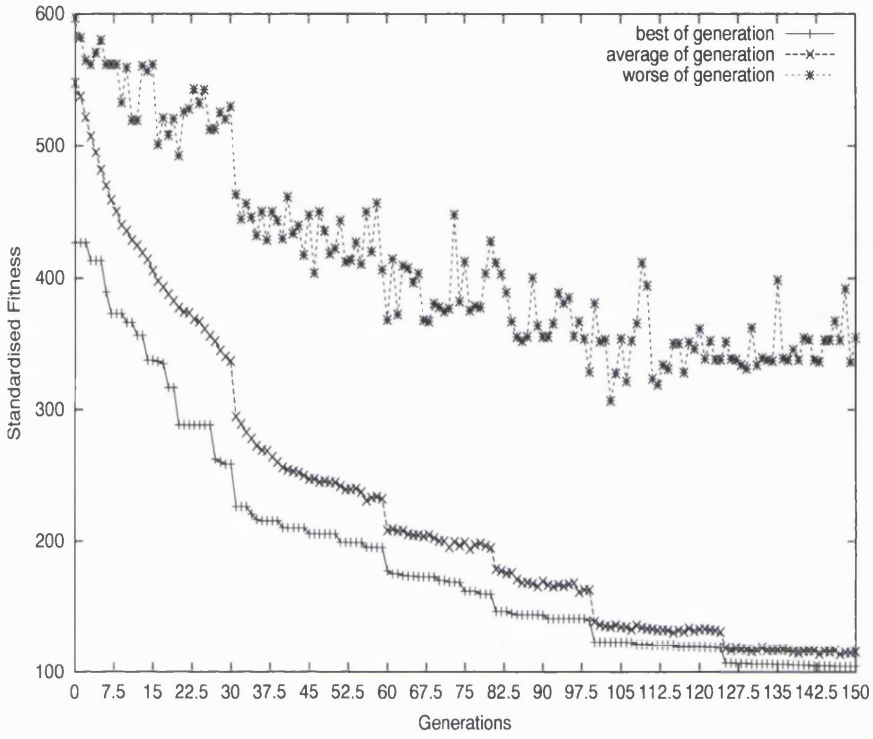


Figure 6.18: Fitness curves for one run of the GP.

except with the binomial distribution 'number of negative responses out of 13 questions of MPRCS' and logistic link function. Number 13 comes from the Modified Personal Report of Confidence as a Speaker (MPRCS) questionnaire. Table 6.4, shows the parameter estimates and the standard errors for this logistic ANOVA model.

The same conclusions would be reached treating the MPRCS as a normal variate and using standard ANOVA, in fact the overall fit has R-Squared $= 0.67$. We only present the logistic regression results here, in Table 6.7, where one can see the effect of the variable (MPRCS) on all combinations.

No variable can be deleted from the model without significantly reducing the fit (measured by the deviance which is a log-likelihood ratio) at least at the 5% level (mostly the p-value is much lower than this). This is clearly shown in Table 6.7. Examining coefficients of the model (Table 6.4), we conclude:

- The presence of an audience significantly increases the MPRCS score for the phobic subjects.
- Being confident rather than phobic significantly decreases the MPRCS score. There is an interaction effect, the phobic group's MPRCS is significantly higher in the presence of an

Condition:	Empty	People
Phobic	6.7	10.5
	(3.0)	(1.4)
Confident	2.4	2.7
	(2.8)	(2.5)

Table 6.3: Mean (SD) for **MPRCS**

Condition:	Empty	People
Phobic	0.1	1.4
	(0.18)	(0.2)
Confident	-1.5	-1.4
	(0.2)	(0.2)

Table 6.4: Parameter Estimates and Standard Errors for the logistic ANOVA model with MPRCS the dependent variable

Condition:	Empty	People
Phobic	39	31
	(24)	(15)
Confident	57	57
	(16)	(19)

Table 6.5: Mean (SD) for **Self Rating**

Condition:	Empty	People
Phobic	97	165
	(43)	(38)
Confident	42	43
	(39)	(31)

Table 6.6: Mean (SD) for **Somatic**

Variate	Deviance for deletion	Df	Chi-Squared tabulated
Audience	16.96	1	3.841
Condition	125.5	1	3.841
Audience.Condition	7.851	1	3.841

Table 6.7: Logistic Regression Results

audience.

An ANOVA, in Table 6.5, shows that there is a significant difference between the self rating of phobic and confident groups (self rating is clearly higher for the confident group). However, there is no significant difference related to the presence of an audience.

The somatic response for phobic subjects, as presented in Table 6.6, is significantly higher with the audience. There is a strong interaction effect.

6.3.3.1 Conclusion

The hypothesis for the experiment was that there would be a difference in response to a virtual audience as between the confidence speakers and the phobic speakers. This is born out on both subjective response variables (MPRCS, Somatic). The only variable for which this is not found is 'self rating' which is not based on a standard questionnaire unlike all the others.

The results of this experiment provide further evidence for the conclusion that virtual audiences do have an appropriate impact on speakers: the observed anxiety responses are similar to what would be expected for speaking in front of real audiences.

6.3.4 Results from Using EVA on this Data Set

The same data, used for the statistical analysis, was used as input to EVA as described in Section 6.3.1.

Figures 6.19, 6.20, 6.21 and 6.22 show a sample of the faces produced by our method for this particular data set. The faces are presented in groups of 4 according to *Anxiety Condition* and *Audience Type*.

It is clearly evident from the resulting visualisation, that there is no, or little, visual difference between confident users with no audience (Figure 6.19) and with the presence of a virtual audience (Figure 6.20). The faces are generally in the range from neutral to happy. On the other hand with phobic users there is a considerable difference. Faces in Figure 6.22 (with audience) are significantly more anxious, sad and angry than faces in Figure 6.21 (no audience). In general, the generated facial visualisation of confident subjects look much happier, less anxious, and less angry than the generating faces corresponding to phobic subjects. This provides an illustration of our expected results, set prior to the analysis of the data, and thus validating the mapping produced by EVA.

6.4 Conclusions from Experimental Data

The experimental data supports the main thesis of this research. Namely, in this Chapter we have shown that naturalistic visualisations (such as a human face) of multivariate data sets,

Figure 6.19: Sample of confident subjects without audience.

together with a meaningful automatic mapping can be achieved and it works.

Experiments using simulated and real financial data sets have shown that even non-expert users can use the constructed visualisation to quickly interpret the significance of the data. Subjects were able to cluster faces according to company performance, each being represented by a visual structure, in little time with a high success rate and without being told of the significance in the visual structure. This can easily be compared to a time consuming procedure of going through raw spreadsheet data. An example of such analysis is presented in Appendix A. Moreover, an experiment on Fear of Public Speaking data, has shown that EVA performs an accurate mapping from this data, to the emotions the subjects had while giving their talk. The faces produced by the method clearly corresponds to the subject's emotional state as shown in the statistical analysis of the data set.

Figure 6.20: Sample of confident subjects with audience.

Figure 6.21: Sample of phobic subjects without audience.

Figure 6.22: Sample of phobic subjects with audience.

Chapter 7

Conclusions and Future Work

EVA, uses a fundamentally different approach compared to the vast majority of Information Visualisation techniques to date. Existing techniques and visualisation tools are mainly concentrated in the area of semiotics. Research in this area is concerned with the invention of a new "axis" to display data, for example in the form of new shapes, colour, and/or the dynamic manipulation of data with the aim of achieving the *Information Visualisation Seeking Mantra* "Overview, zoom, filter details on demand", as proposed by Prof. Ben Shneiderman. Although the target is always the same, better understanding of the underlying data, the approach taken in this thesis is different.

The aim of this research has been to explore the possibilities of visualisation of highly correlated multivariate data sets holistically, in the form of naturalistic visual structures, and for automatic performance of the mapping, in such a way that the impact on the emotions of the users of the visualisation are taken into consideration. EVA enforces a homomorphism between important characteristics of the data and the emotional or perceptual impact of the visual structure. Salient global aspects of the data (the utility or value functions) are mapped to emotional or perceptually significant aspects of a visual structure. The features that allow rendering of the visual structure are determined by a genetic program that breeds generations of visual structures, such that in each successive generation there is a greater match between the utility functions and the visual structure characteristics. The type of visual structures produced by this method are meant to be informative "at a glance", and can also reveal important detailed information or unusual characteristics present in the data (e.g., a happy face with a hint of anxiety). The method is not put forward as an alternative to other types of visualisation, but rather it provides a first-pass visualisation that may, in particular applications, raise interesting features that subsequently may be explored in detail through traditional visualisation techniques, or indeed statistical analysis.

7.1 Review of Contributions

The contributions of this thesis have included a critical literature review of Information Visualisation to date, the formalisation and implementation of EVA as an algorithm that automatically maps multivariate data sets to visual representations, the use of naturalistic visual cues so that they represent the emotions the observers of the data would have if they analysed the data, the invention and implementation of a technique that quantifies emotional expressions of a human face, and finally results from the experiments of this novel approach.

In Chapter 2, we critically reviewed the background work on Information Visualisation, focusing mainly on the use of multivariate data sets, hence achieving the first target. The literature review presented in that Chapter, revealed that the area of mapping multivariate data sets to naturalistic visual structures in an automatic, but meaningful, mapping has been unexplored. Not only has it not been investigated until now, but, it shows great promise when considering the separate advantages outlined in the research of naturalistic visual structures and in automating the mapping from data to visual representations. Therefore, a further investigation of the above areas of research looked reasonable.

Chapter 3, achieved the second target. Namely, the methodology behind EVA, was formally presented. Formalisation of the algorithm introduced a number of problems. Firstly, an optimisation problem arose between the quantitative measurements of the data which are significant to the user and the qualitative measurements of the visual structure that are significant to human perception. Secondly, there was a need for a technique to automatically quantify emotional expressions in order to achieve the above. The Genetic Program that was implemented, was derived as part of this Chapter with the purpose of solving the optimisation problem mentioned above.

Completion of EVA through the method to automatically quantify emotional expressions and hence realisation of contributions 3 and 4 were presented in Chapter 4. In this Chapter, the visualisation tool, "Geoface 2" was introduced. This was followed by a detailed description, implementation and results of the technique, that automatically quantifies the emotional expressions of a specific facial structure, based on the movement of certain landmarks on its skin (points affected by a number of muscle contractions). There are 25 such points and these were "randomly" selected through trial and error as long as they satisfied the prerequisite that they were directly affected by at least one muscle contraction. This technique is very important to EVA, since it replaces the user input in the evolvement/training of the GP from having to assess every face produced. Having said that, the results produced by this technique are still based on user's subjective evaluations for each emotion.

The final target was met through results from experiments on this approach, and was presented in Chapter 6. A pilot user experiment that utilised a set of three circles as the visual structure, together with simulated financial data was initially undertaken. This simple example was implemented in order to test the feasibility of EVA as an information visualisation technique. Results were very encouraging, leading to the repetition of the experiment replacing, however, the visual structure with a human face and the simulated data with real financial data. The data was gathered from balance sheets and profit and loss accounts of certain companies belonging to similar sectors and hence similar interpretation of the variables. Despite the use of a more complicated data set, results improved further giving us confidence in the efficiency of faces as a good visual representation.

A final experiment was performed to further test the validity of EVA in presenting the correct qualitative visual cues from quantitative information. Real data was gathered based on a Virtual Reality experiment on Fear of Public Speaking. The multivariate data set collected, one row per speaker, distinguished the speakers as confident and phobic and, also, those who gave their talk in an empty room and those who gave their speech in front of a virtual audience. Results from EVA, a human face per speaker, were illustrative. One can clearly distinguish, in general, a phobic user from a confident one, especially in the presence of virtual audiences. There is a clear match between the faces produced by EVA and the statistical analysis of data collected from the experiment.

The above contributions give support to the main thesis of this research work. In the remainder of this Chapter, we perform a critical review of EVA followed by guidance for future work.

7.2 Critical Review of EVA

It can be safely concluded from the above that EVA shows promise and can become an important visualisation tool when used correctly. We propose the use of EVA, especially for the purposes of Exploratory Analysis as a first pictorial representation of a highly correlated multivariate data set. As already mentioned, before this Exploratory Analysis is performed there are cases where we know very little about the data and such visualisation of the data should provide the means for the user to formulate hypotheses. These hypotheses, when needed, can later be tested using conventional statistical analysis or more traditional information visualisation techniques.

It must be said that analysis of data heavily depends on levels of expertise. The use of naturalistic visual structures and the consideration of the emotions of the user in the mapping performed by EVA minimises the gap between expert and non expert users. Correct hypotheses

can be formulated by both with the same minimal effort.

Further applications of EVA include data changing in real time. For example stock brokers can have one face representing each stock of their portfolio. Generally positive feedback should indicate "hold" of that stock, and negative feedback should give an indication to "sell", with mixed feedback indicating the need for further investigation. Similarly, for a project manager with several projects, EVA could be used to visually represent each of the projects. In this case, generally speaking, positive feedback should give confidence that the project is running smoothly, whereas the need for action can be taken in all other cases.

However, visual data exploration usually follows 3 steps as already mentioned in this thesis: Overview, zoom filter details on demand - Ben Shneidermans' "mantra" of Information Visualisation. With EVA we achieved the first step in a very natural way, examining human faces, taking into consideration the whole data set. No data reduction techniques were included and variables were not treated independently but rather holistically. Exploring ways of achieving the other two steps of the mantra, zoom and filter details on demand should further improve the method. This remains to be done.

Moreover, validity of EVA also depends upon the proof of convergence of the GP. Such proof of Genetic Programs is not possible with today's theoretical knowledge of them and is highly unlikely it will ever be feasible. Each tree in each population of Genetic Programs is different, being constructed from parent trees through the use of crossover. Although, experimentally it is obvious that the steps made in the search space become smaller and smaller with results becoming at least as good as previous generations, the search space produced by the huge amount of crossover possibilities becomes vast and impossible to be handled theoretically. If a tree consists of n nodes, then the number of possibilities for crossover becomes of order $O(n^2)$. Considering that one usually has 500 such trees and approximately 50 generations, the numbers become impractical.

7.3 Guidance for Future Work

Computational tools for discovery, such as data mining and information visualisation have advanced dramatically in recent years. Unfortunately these tools have been developed by largely separate communities with different philosophies. Future work should involve the tight integration of EVA with traditional techniques from disciplines such as statistics, operational research or even with other information visualisation techniques. Integration of EVA and these established methods would quickly combine advantages from all, improving the quality and speed of the visual data mining process.

For information visualisation to be effective, it is important to include the human in the data exploration process and combine the flexibility, creativity and general knowledge of the human with the enormous storage and computational capabilities of today's computers. This can be achieved in EVA through "backtracking". Users will be able to query the visualisation tool, for example verbally, and the system should respond appropriately by backtracking through the evolvement history of the GP, back to the original data set. Being able to backtrack through the data is effectively achieving zoom and filter details on demand, therefore completing the Information Visualisation mantra.

Having shown that the basic idea works, it will be interesting to identify possible boundaries for particular parameters that define the operation of the method. For example, through further study we can define a reasonable range, in terms of the number of variables in the data set (dimensionality) and also in terms of the number of aspects in the data set of importance to the user (value system). The latter is strongly related to the visual structure, and more specifically to the maximum number of characteristics a user can easily distinguish.

Moreover, looking at EVA from a different perspective we can examine the power of EVA through the discipline of psychology. More specifically, the confidence a human face provides in the decision making process at different levels of seriousness, in terms of consequences, is very interesting. Results from the above can be used to distinguish further areas where EVA should not be applied or is safely applicable. Another area in psychology of great interest, is that of asymmetrical studies with human faces.

It is indisputable that Information Visualisation has opened many horizons for data discovery. The last $10 - 15$ years have seen the development of the first computer-based prototypes, which have been welcomed with enthusiasm and their academic and business potential has been recognised. Quoting the Economist, June 19th 2003, "Information Visualisation is about to go mainstream. While it may not be the killer application some expect, *Information Visualisation* is going to help users to manipulate data in wholly new ways".

EVA, the approach presented here, is still in its early days. Further research in this area will, unquestionably, improve EVA further as a visualisation tool. We are hoping that EVA will expand the possibilities of Information Visualisation in allowing people to see and manipulate data in new ways, efficiently and effectively.

Appendix A

Spreadsheet Analysis for Financial Data Set

An analysis of the data used for the second experiment described in Section 6.2 is presented here. This analysis was performed in order to categorise the data into the 5 clusters, described in Section 6.2.3, in order to be able to evaluate user responses for this experiment.

Table A.1 shows a sample of the data used and tables A.2, A.3, A.4, A.5, A.6 show the 10 companies in each cluster used for the experiment.

Figures A.1, A.2, A.3, A.4, A.5, A.6 and A.7 show all possible combinations of the ratios per company plotted as separate graphs.

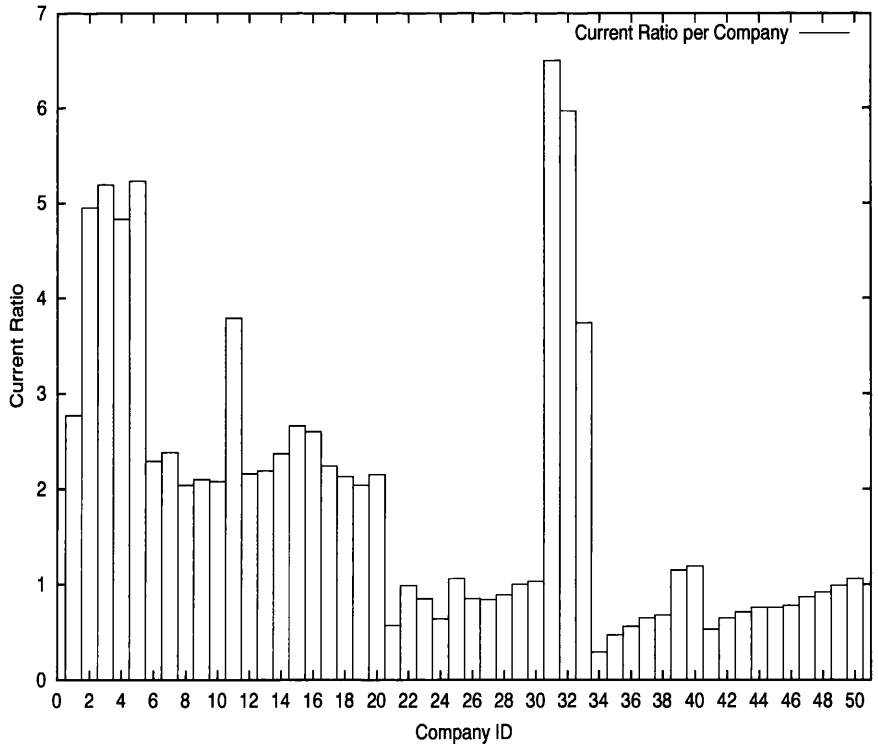


Figure A.1: Current Ratio for Each Company.

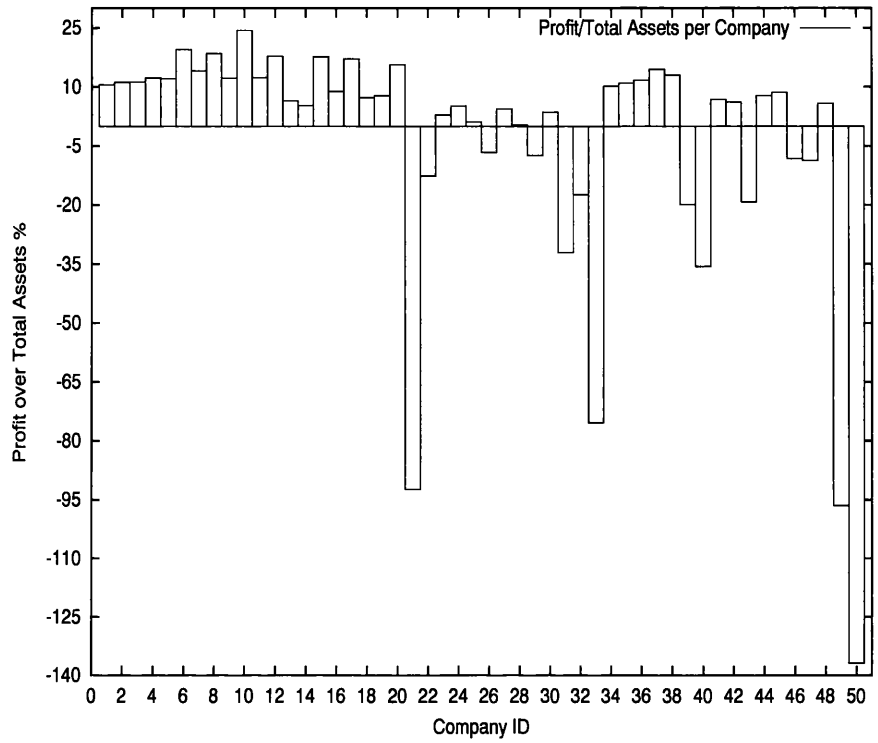


Figure A.2: Profit over Total Assets Ratio for Each Company.

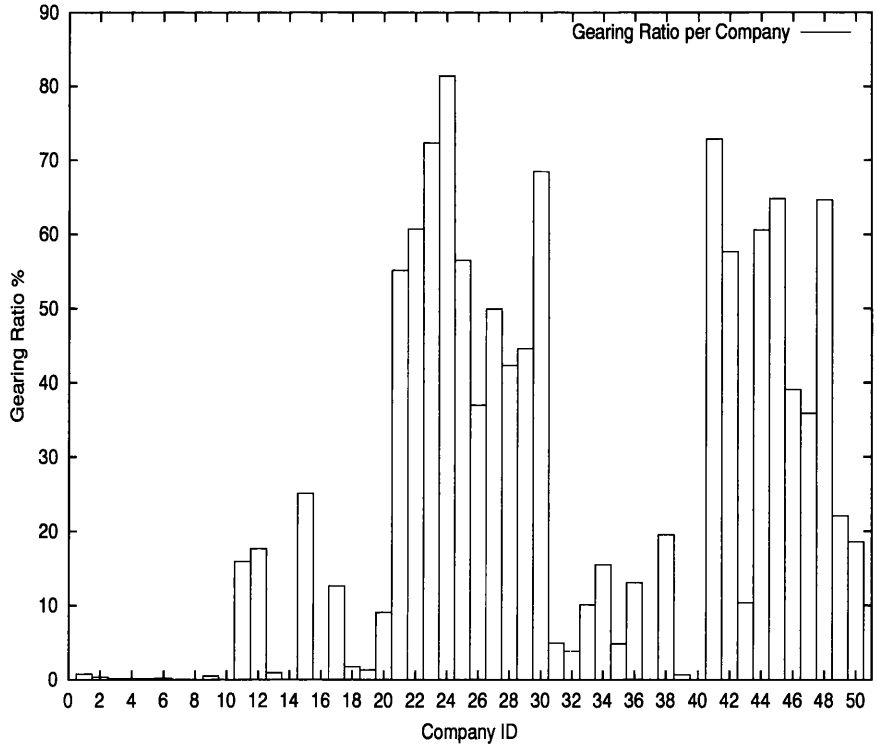


Figure A.3: Gearing Ratio for Each Company.

Ordinary Share Capital	Equity Capital & Reserves	Preference Capital	Fixed Assets	Current Assets	Current Liabilities	Loan Capital	Long Term Liabilities	EBIT	Profit Before Tax	Ordinary Dividends
97000	811100	0	1057800	1078100	528200	617900	796600	185100	185100	38800
106900	950100	0	1188500	1386000	761100	543700	863300	212800	212800	47100
107100	966200	0	1210800	1351800	842300	446300	754100	266000	266000	51900
107200	1062300	0	1423700	1478900	880300	631400	960000	322900	322900	57200
118400	1016600	0	1532400	1578700	1009600	698000	1084900	411900	411900	73400
9230	61882	10120	50040	312109	275270	233	14877	36671	36535	6696
11374	135312	10120	125261	448056	385110	6605	42775	34657	45529	8304
11536	117007	10104	202898	481473	443294	50375	113966	68195	75762	13649
13574	197243	9980	262933	678034	602337	50904	131407	49696	59066	16106
14200	240000	5800	319900	755600	622600	91500	207100	76805	86826	21719
220000	2373000	9000	3990000	2165000	1670000	1617000	2103000	731000	507000	175000
222000	2322000	9000	4109000	2070000	1685000	1705000	2163000	648000	450000	185000
258000	2281000	9000	3849000	2459000	2063000	1345000	1955000	754000	551000	197000
259000	2134000	9000	3182000	2203000	1871000	1090000	1371000	1081000	801000	406000
260000	2127000	9000	3137000	1978000	1687000	1065000	1292000	729000	548000	245000
724600	1567700	0	2036200	833000	724100	408800	577400	187700	86100	46800
724700	1376400	0	1922600	722300	717000	392700	551500	213300	139600	43000
726200	1493200	0	2020100	821500	868700	328400	479700	219200	181700	51000
731800	1657700	0	2536600	741200	974100	515100	646000	278200	243500	63900
753500	2061500	0	2944200	806700	1125700	474600	563700	350900	308400	75500
508900	2047100	0	3047000	461600	405300	791200	824700	358400	285200	81400
510900	2543000	0	3616200	390000	486500	755800	781700	378000	322400	92000
1028200	2845200	0	4113700	281200	664600	715300	744500	405800	366100	103900

Table A.1: Data sample

Cluster 1:Good Overall Performance			
Company	**Current Ratio**	**Profit/Total Assets**	**Gearing Ratio**
90	2.77	10.54%	0.76%
106	4.95	11.13%	0.37%
107	5.19	11.29%	0.17%
108	4.83	12.29%	0.16%
109	5.23	12.09%	0.16%
168	2.29	19.45%	0.21%
164	2.38	14.07%	0.00%
165	2.04	18.45%	0.00%
167	2.10	12.26%	0.50%
170	2.08	24.35%	0.00%

Table A.2: Good Overall Performance

Cluster 2:Medium Overall Performance			
Company	**Current Ratio**	**Profit/Total Assets**	**Gearing Ratio**
110	3.79	12.35%	15.91%
43	2.16	17.78%	17.65%
138	2.19	6.43%	0.94%
162	2.37	5.20%	0.00%
42	2.66	17.65%	25.06%
163	2.60	8.84%	0.00%
44	2.24	17.11%	12.62%
137	2.13	7.23%	1.77%
139	2.04	7.74%	1.36%
45	2.15	15.67%	9.08%

Table A.3: Moderate Overall Performance

Cluster 3:Bad Overall Performance			
Company	**Current Ratio**	**Profit/Total Assets**	**Gearing Ratio**
181	0.57	-92.44%	55.13%
129	0.99	-12.61%	60.70%
56	0.85	2.91%	72.34%
104	0.64	5.13%	81.41%
130	1.06	1.09%	56.46%
113	0.85	-6.65%	36.97%
156	0.84	4.42%	49.92%
51	0.89	0.36%	42.32%
128	1.00	-7.42%	44.58%
58	1.03	3.62%	68.47%

Table A.4: Bad Overall Performance

Cluster 4: 2 Good - 1 Moderate Performance			
Company	**Current Ratio**	**Profit/Total Assets**	**Gearing Ratio**
182	6.50	-32.07%	4.96%
88	5.97	-17.39%	3.84%
184	3.74	-75.36%	10.10%
93	0.29	10.21%	15.50%
116	0.47	10.96%	4.85%
119	0.56	11.67%	13.06%
147	0.65	14.45%	0.00%
65	0.68	13.02%	19.50%
140	1.15	-19.90%	0.71%
87	1.19	-35.65%	0.00%

Table A.5: Slightly worse than good overall performance

Cluster 5: 1 Good - 2 Moderate Performance			
Company	**Current Ratio**	**Profit/Total Assets**	**Gearing Ratio**
105	0.53	6.80%	72.84%
60	0.65	6.15%	57.66%
89	0.71	-19.22%	10.36%
101	0.76	7.78%	60.55%
120	0.76	8.58%	64.77%
157	0.78	-8.22%	39.08%
127	0.87	-8.71%	35.87%
59	0.92	5.84%	64.60%
185	0.99	-96.51%	22.08%
183	1.06	-136.88%	18.58%

Table A.6: Slightly better than moderate overall performace

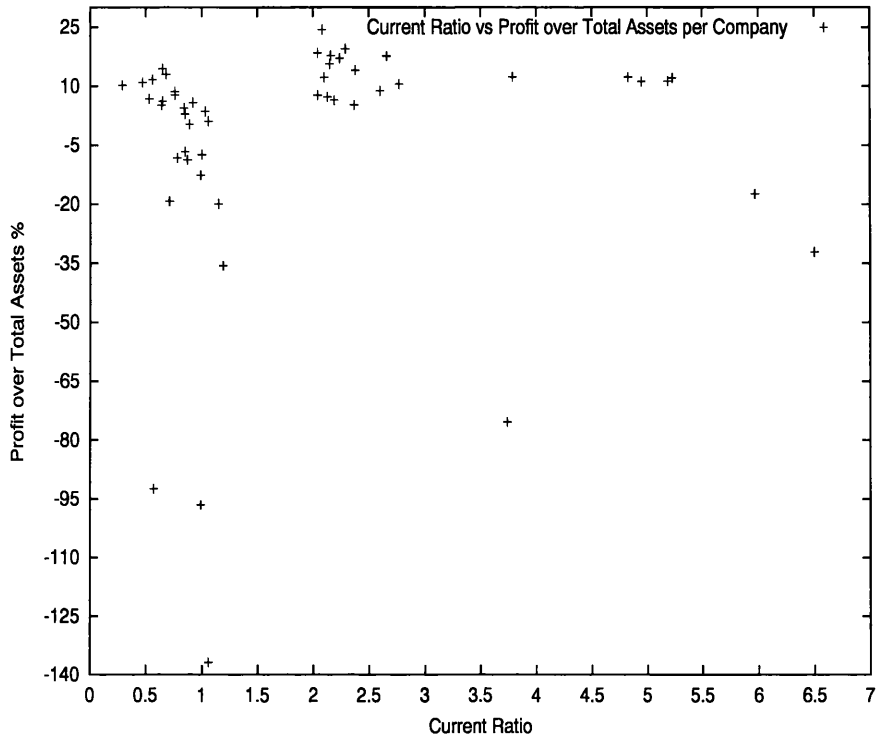


Figure A.4: Current Ratio against Profit over Total Assets for Each Company.

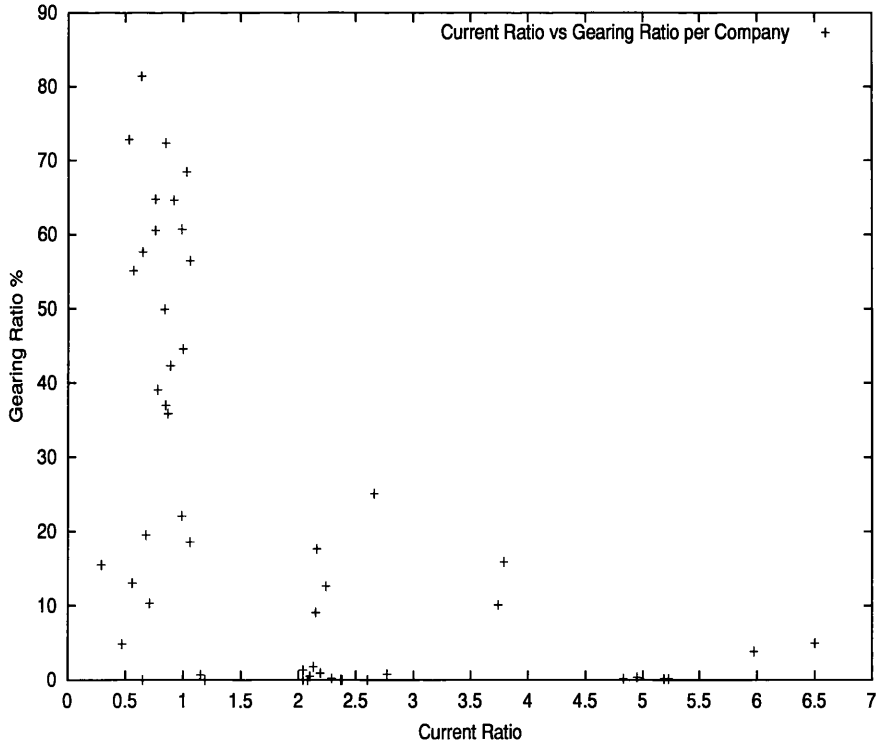


Figure A.5: Current Ratio against Gearing Ratio for Each Company.

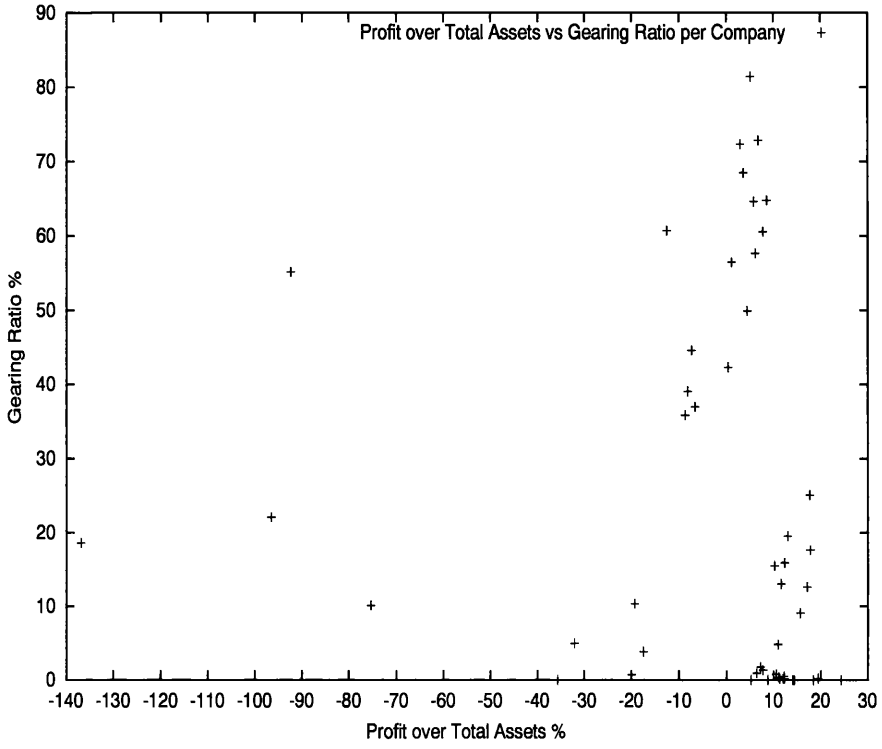


Figure A.6: Profit over Total Assets against Gearing Ratio for Each Company.

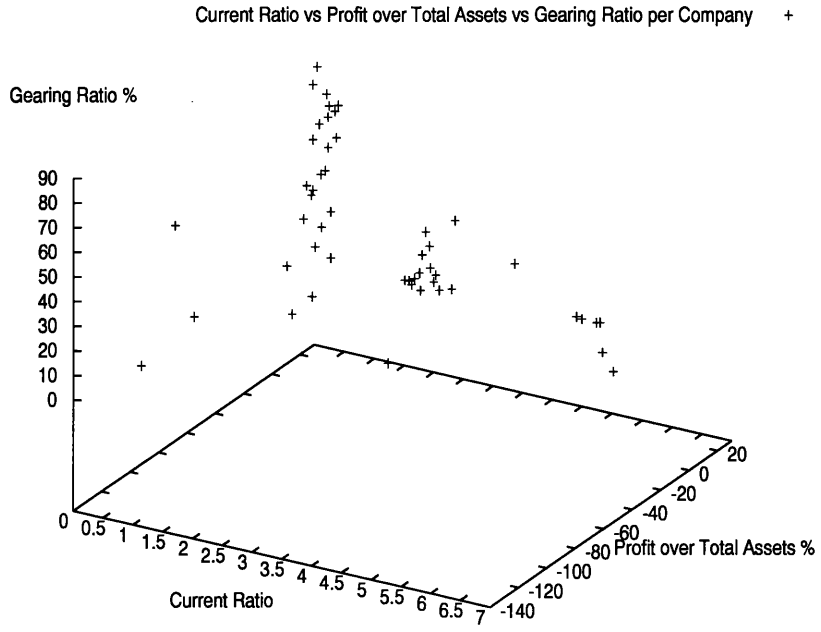


Figure A.7: All Three Ratios for Each Company.

Appendix B

GEOFACE and GEOFACE 2

The original Geoface model [PW96], together with extensions programmed to the model, is presented here. Some of the extensions were first implemented in [Dri00]. As described in Section 4.2, Geoface forms the backbone of our visualisation tool.

B.1 The original Geoface

As already mentioned in Section 4.1, the original program demonstrates a muscle model approach for animating facial expressions. The program can be subdivided into three conceptual levels of abstraction; the geometric model, the emotional model and the muscle model.

The geometric model is a human face that is represented by a set of three dimensional vertices and polygons that represent the topography of the face. To display the surface of the face, it is necessary to transform these points into a polygonal mesh. In Waters' model, just the front view of the head is defined as a facemask as shown in Figure B.1.

For the original emotional model, there are six sets of muscle configurations stored in an expression vector file. These are, happiness, anger, surprise, sadness, fear and disgust, and are lists of floating point values corresponding to a contraction value for each muscle. Upon the appropriate input, the model selects the appropriate expression vector and applies them to the muscle set, resulting in deformations that simulate the facial configuration of an exception. The expressions above are based on a system called Facial Action Coding System (FACS), described in Section 4.2.1.

The muscle model, which is the most interesting one, animates the face by deforming the polygonal mesh using a muscle vector function. There are eighteen muscles on the original model, nine on each side of the face. The muscles represent real muscles found in the face, such as the Right Zygomatic Major and Left Angular Depressor. The former is used to pull the outer parts of the mouth area hence it is the primary muscle used to smile whereas the latter performs the opposite operation (pulling down the middle and outer region of the mouth and often forms part of a frowning face. Figure B.2 shows a frontal view of facial muscles.

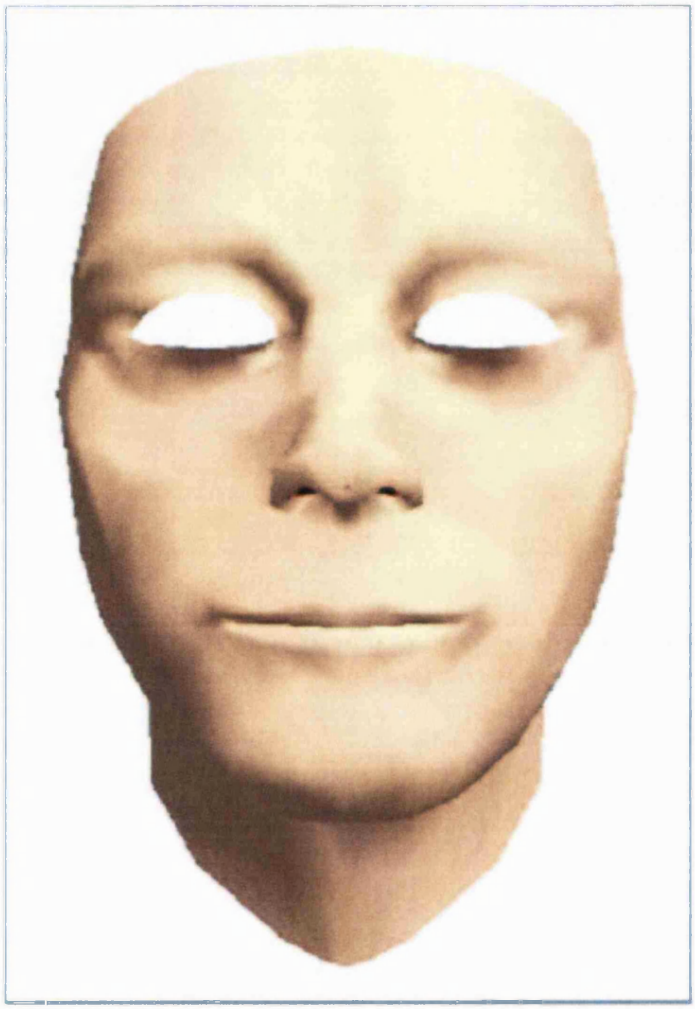


Figure B.1: The original Geoface model.

Waters' technique [Wat87], considers muscles as single direction vectors, with a "head" that is connected to the bone of the skull, and a "tail" that is connected to the flesh of the face. As a muscle contracts it pulls the skin towards the point at which it is connected to the skull. Each muscle has a zone of influence and the vertices of the polygon mesh describing the surface of the face are moved according to the contraction of the muscles.

Typically facial muscles are grouped depending on their orientation and shape. There are three such groups; Linear, Circular or Sphincter and Sheet muscles. The muscles in the original Geoface model, model the deformations of the skin that result from a linear contracting. Theories were developed by Parke and Waters [PW96], to model the other two muscle types, namely sheet and sphnicter muscles.

Linear muscles are the most common, an example of which is the zygomatic major shown

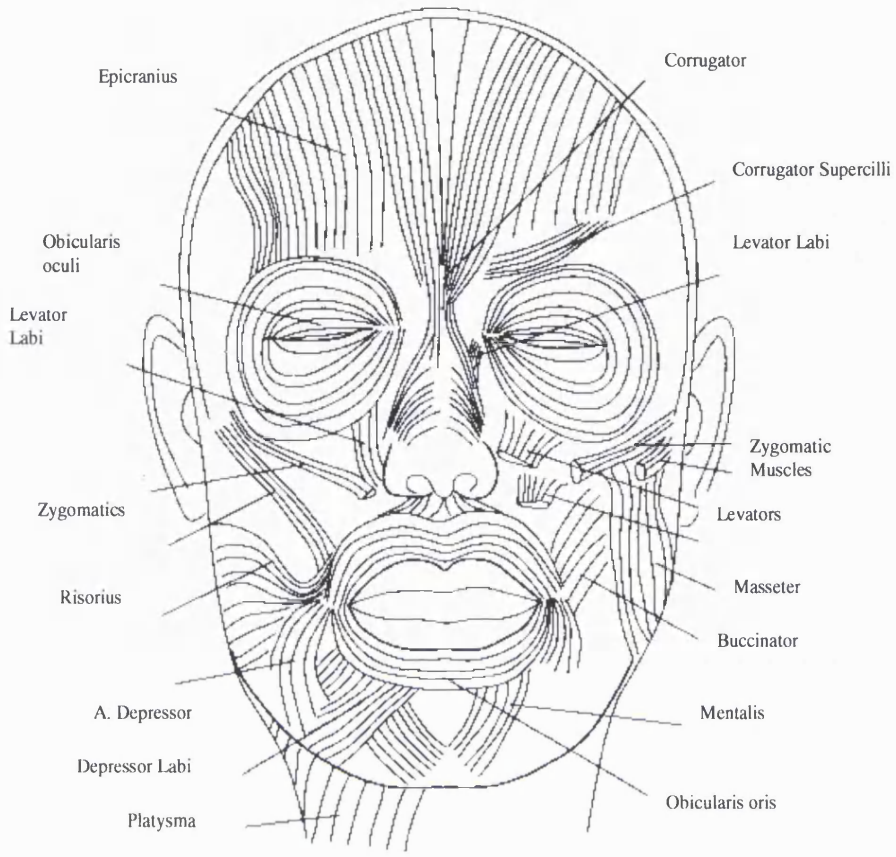


Figure B.2: Frontal view of facial muscles [PW96].

in Figure B.2. As the muscles contract they pull the point at which they are attached to the skin towards the point of attachment on the skull. The circular or sphincter muscles such as the obicularis oris are found around spherical regions such as the mouth and the eyes. These pull inwards towards their center producing expressions such as pursed lips. A sheet muscle such as the frontalis is found in the forehead and can be considered as a large collection of linear muscles all placed next to one another.

B.1.1 Linear Muscles

As already mentioned, the model for linear muscle vectors developed by Waters assumes that the muscle is connected to skin at one end and to the bone of the skull at the other. The muscle contracts in a direction along a line from the skin connection point towards the bone connection point. As the muscle contracts each vertex of the mesh is warped in accordance with a number of relationships that represent the movement of skin under muscular forces. Figure B.3 and Figure B.4 show samples of deformations of a polygon mesh for contraction and expansion.

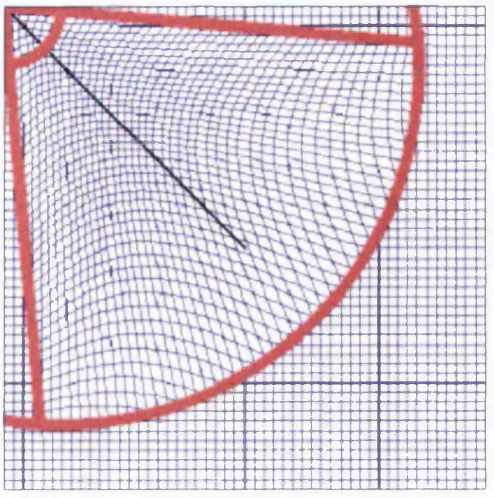


Figure B.3: A linear muscle with a contraction value of 1.0 [PW96].

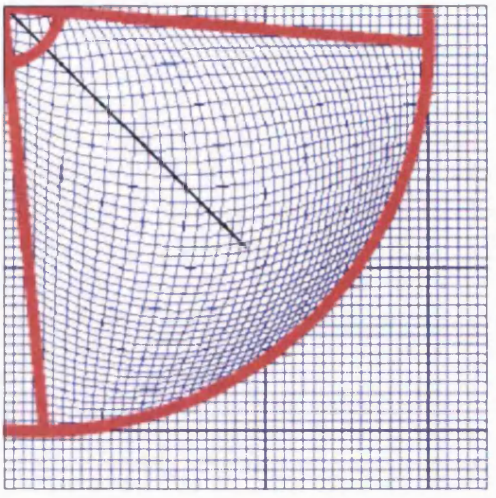


Figure B.4: A linear muscle with an expansion value of 1.0 [PW96].

B.2 Extensions to Geoface

A number of extensions have been implemented on top of the original GEOFACE model, resulting in the refined version used as a visualisation toolkit for the purposes of this thesis.

Work at MIT has shown the importance of many characteristics when considering the relationship that a user can form with a synthetic character model [Cas94]. As a result of this, eyeballs and hair have been added to the model to improve the overall effect of the facial representation. Adding eyes is easily achieved by placing spheres in the empty eye sockets of the facial mask. Texture maps of pupils and hair are then added to improve realism.

Moreover, the muscle model which is composed of geometric distortion functions that influence particular sets of vertices in a 3D space, is independent of the geometric model. This means that the muscle model can be applied to any 3D object and hence to any face. In fact the program can display any 3D object provided that is defined in a certain way: using two source data files; one containing a vertex list and one containing an index list, and these lists must describe a polygonal object composed of triangles.

In addition to the above, a facility has been added to the program that allows interactive viewing of muscle zones of influence. This functionality was very important during the selection of the 25 landmark positions on the face, movements of which determine the degree of individual emotional expressions as described in Section 4.2.3. The colour scheme used to produce this is analogous to a thermograph, with blue areas being influenced by only one muscle, through the colours of the rainbow to white.

Furthemore, a variety of manipulation methods to the muscle model are provided by the

system (scaling, rotating and translating) enabling the user to create a muscle model suitable for the current facial model and their needs. These functions can be applied to individual, groups or all of the muscles.

New muscles can also be created at run time and added to the muscle set. In order for this muscles to be fully functional, the user must enter a few configuration parameters namely zone-depth, zone-angle and group id (if they belong to a particular group of muscles). The new muscle is then created at origin of the viewing space and can be placed to the target position using the functions described above.

Finally a menu system is provided with the click of a mouse button for user input, complemented with keyboard shortcuts for the execution of standard tasks. Figure B.5, shows the menu system and more specifically how you can select an individual muscle.

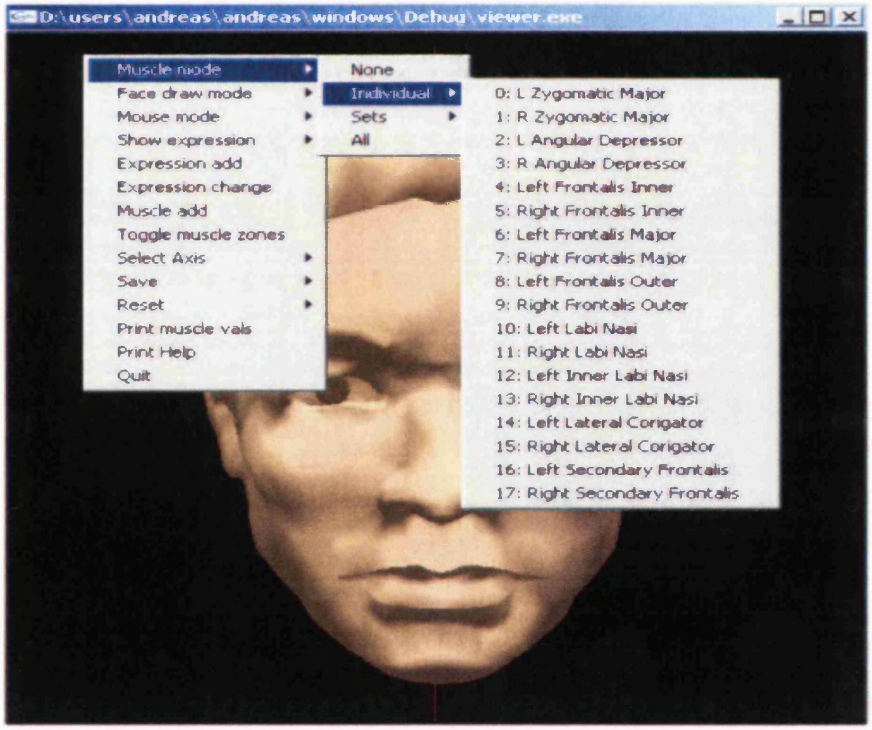


Figure B.5: Selecting an individual muscle from the menu.

Appendix C

Chernoff Facial Features

Dim	Facial Feature	Dim	Facial Feature
1	Face width	2	Ear level
3	Half face height	4	Eccentricity of upper ellipse of face
5	Eccentricity of lower ellipse of face	6	Length of the nose
7	Position of centre of mouth	8	Curvature of mouth
9	Length of mouth	10	Height of centre of eyes
11	Separation of eyes	12	Slant of eyes
13	Eccentricity of eyes	14	Half length of the eye
15	Position of pupil	16	Height of eyebrow
17	Angle of brow	18	Length of brow
19	Radius of ear	20	Nose width

Table C.1: Description of facial features of Chernoff faces.

Appendix D

Example of a GP Tree Evolved by EVA

What follows is a set of functions (12), produced by EVA, belonging to the best-of-generation 50 individual in the best run of the GP as part of the first experiment. These functions determine the features of the visual structure, such as radius and colour, and hence allow for its rendering. One can clearly see that although certain variables are not involved in the "value system", as defined in Section 6.1.1, they are still part of the trees produced.

In these formulae a represents *Number of shares*, b represents *Earnings*, c represents *Dividend*, d represents *Liquidity*, e represents *Liabilities*, b represents *Earnings*, f represents *Costs*, g represents *Sales* and h represents *Staff*.

$$
\begin{aligned}
f_1 &= \left(\frac{c^2 e\,(h-f) + \frac{c^3}{g^2}\left(\frac{f-c}{cf} + e - g\right) + cf - c^2}{2df^2 g - g^3 - f - c}\right)\left(\frac{c-a}{cf} + eh\right) \\
f_2 &= \frac{c+e}{d-h} + \left(\frac{3c}{a}\,(h-b) - e^2 ac + \frac{edbc^2}{a}\right)\frac{db}{fa} \\
f_3 &= \frac{\frac{fd}{h}}{\frac{f}{d} + c - h} - (b+g)\,dg - a - d + f + \frac{fd}{be}\,\frac{\frac{c}{d} + ghd^2}{\frac{(b+g)d^2 - ad - d^2 + c^2}{e}\,(d - 2g + f) - \frac{(b+g)(c-a)}{c}} \\
f_4 &= (2e+b)\,\frac{baf}{cd} \\
f_5 &= \frac{g-f}{eg}\,(g - a - hd) + 2c + \frac{b}{e} - b - d + d^2 f^2 \\
f_6 &= 2 + gb - gb\,(2gb - ed) - 2bg\,(cd - ef) \\
f_7 &= 2\,(h+g) + \left(\frac{f\,(g-c)}{h\,(b-g)} - ce - \frac{d}{c}\right)\left(\frac{h}{f}\,(b-e) - b^2 + e^2\right) + ce - be + (g-c)\,(be - a) \\
f_8 &= \left(\frac{a-d}{c-b} - \frac{2g}{d} + 1 + f + d\right)(a+f) + \frac{c - b - \frac{d-h}{g-e}}{\frac{c+d}{c-2b+e}} \\
f_9 &= \frac{(2g - a + c)\,f^2}{\frac{bf+c-g}{c-g}} - 2h + 3a + hc^2 f - dhfc - \frac{b}{2c} \\
f_{10} &= b - d \\
f_{11} &= \frac{b - d - \frac{g}{d} - \frac{h+b}{\frac{c}{d}}}{\left(\frac{\frac{c}{g}}{\frac{e}{d}} + f - d\right)\left(f + f - \frac{d-g}{f-d}\right)\left(b - d - cd - \frac{\frac{d-g}{f-d}}{d-g}\right)} \\
f_{12} &= ge\,(c+b) + (g-b)\,(b-f) - b^2\,(g - 2b + c) + 2be\,(h+a) + \frac{4a^2}{3e}
\end{aligned}
$$

Bibliography

[AK96] David Andre and John R. Koza. A parallel implementation of genetic programming that achieves super-linear performance. In Hamid R. Arabnia, editor, *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications*, volume III, pages 1163–1174, Sunnyvale, 9-11 August 1996. CSREA.

[Alt94] Lee Altenberg. The evolution of evolvability in genetic programming. In Kenneth E. Kinnear, Jr., editor, *Advances in Genetic Programming*, chapter 3, pages 47–74. MIT Press, 1994.

[AM99] Marc Alexa and Wolfgang Muller. Visualisation by examples: Mapping data to visual representations using few correspondences. In *Proceedings of the Joint EUROGRAPHICS and IEEE TCVG, Symposium on Visualisation in Vienna, Austria, May 26–28, 1999*, pages 23–32. Springer-Verlag/Wien, 1999.

[Ang93] Peter John Angeline. *Evolutionary Algorithms and Emergent Intelligence.* PhD thesis, Ohio State University, 1993.

[Ang94] Peter J. Angeline. Genetic programming and emergent intelligence. In Kenneth E. Kinnear, editor, *Advances in Genetic Programming*, Complex Adaptive Systems, pages 75–98, Cambridge, 1994. MIT Press.

[AP92] P. J. Angeline and J. B. Pollack. The evolutionary induction of subroutines. In *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*, Bloomington, Indiana, USA, 1992. Lawrence Erlbaum.

[AS94a] Christopher Ahlberg and Ben Shneiderman. The alphaslider: A compact and rapid selector. In *Proc. ACM Conf. Computer-Human Interaction, CHI*, pages 365–371, 24–28 April 1994.

[AS94b] Christopher Ahlberg and Ben Shneiderman. Visual information seeking using the filmfinder. In *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems*, volume 2 of *VIDEOS: Part I: Browsing Navigation*, page 433, 1994. Color plates on page 484.

[Bac96] T. Back. *Evolutionary Algorithms in Theory and Practice.* Oxford University Press, New York, 1996.

[Ban93] Wolfgang Banzhaf. Genetic programming for pedestrians. In Stephanie Forrest, editor, *Proceedings of the 5th International Conference on Genetic Algorithms, ICGA-93*, page 628, University of Illinois at Urbana-Champaign, 17-21 July 1993. Morgan Kaufmann.

[BC84] R. A. Becker and J. M. Chambers. *S: An Interactive Environment for Data Analysis and Gra phics.* Wadsworth Advanced Books and Software, 1984.

[Ber81] J. Bertin. *Graphics and Graphic Information Processing*. deGruyter, 1981.

[BM76] J. A. Bondy and U. S. R. Murty. *Graph Theory with Applications*. Macmillan, London, 1976.

[BNKF98] Wolfgang Banzhaf, Peter Nordin, Robert E. Keller, and Frank D. Francone. *Genetic Programming – An Introduction; On the Automatic Evolution of Computer Programs and its Applications*. Morgan Kaufmann, dpunkt.verlag, January 1998.

[Cas94] Cassell. et al. animated conversation: Rule-based generation of facial expression, gesture and spoken intonation for multiple conversational agents. In *Proceedings of SIGGRAPH 1994 (ACM Special Interest Group on Graphics)*. ACM Press, 1994.

[CEWB97] Kenneth C. Cox, Stephen G. Eick, Graham J. Wills, and Ronald J. Brachman. Visual data mining: Recognizing telephone calling fraud. *J. Data Mining and Knowledge Discovery*, 1(2):225–231, 1997.

[Che71] H. Chernoff. The use of faces to represent points in n-dimensional space graphically. Research Note NR-042-993, Department of Statistics, Stanford University, december 1971.

[Chi00] E. H. Chi. A taxonomy of visualization techniques using the data state reference model. In *2000 IEEE Symposium on Information Visualization (InfoVis '00)*, pages 69–76, Washington - Brussels - Tokyo, October 2000. IEEE.

[CM84] William S. Cleveland and Robert McGill. The many faces of a scatterplot. *Journal of the American Statistical Association*, 79(388):807–822, 1984.

[CM85] W. S. Cleveland and R. McGill. Graphical perception and graphical methods for analyzing scientific data. *Science*, 229(4716):828–833, 30 1985.

[CMS99] Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman. *Readings In Information Visualisation: Using Vision To Think*. Morgan Kaufman, 1999.

[CR98] Ed Huai-Hsin Chi and J. T. Riedl. An operator interaction framework for visualization systems. In *IEEE Symposium on Information Visualization (InfoVis '98)*, pages 63–78, Washington - Brussels - Tokyo, October 1998. IEEE.

[CRM91] Stuart K. Card, George G. Robertson, and Jock D. Mackinlay. The information visualizer, an information workspace. In *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, Information Visualization, pages 181–188, 1991.

[Dav91a] L. Davis. *Handbook of Gentic Algorithms*. Van Nostrand Reinhold, 1991.

[Dav91b] Lawrence Davis. Bit-climbing, representational bias, and test suite design. In Lashon B. Belew, Richard K.; Booker, editor, *Proceedings of the 4th International Conference on Genetic Algorithms*, pages 18–23, San Diego, CA, July 1991. Morgan Kaufmann.

[Den95] Daniel Dennett. *Darwin's Dangerous Idea*. 1995.

[D'h94] Patrik D'haeseleer. Context preserving crossover in genetic programming. In *Proceedings of the 1994 IEEE World Congress on Computational Intelligence*, volume 1, pages 256–261, Orlando, Florida, USA, 27-29 June 1994. IEEE Press.

[DPP01a] C. Barker D-P. Pertaub, M. Slater. An experiment on fear of public speaking in virtual reality, medicine meets virtual reality. volume 81, pages 372–378. IOS Press, 2001.

[DPP01b] C. Barker D-P. Pertaub, M. Slater. An experiment on public speaking anxiety in response to three different types of virtual audience. *Presence: Teleoperators and Virtual Environments*, 11(1):68–78, 2001.

[Dri00] Wil Driver. Geoface 2: A parameterised face model. Msc dissertation, UCL - University College London, Gower Street WC1E 6BT, London UK, 1999-2000.

[DW69] R. Friend D. Watson. Measurement of social-evaluative anxiety. *Journal of Consulting and Clinical Psychology*, (33):448–457, 1969.

[EF78] P. Ekman and W. V. Friesen. *Facial Action Coding System (Investigator's Guide)*. Consulting Psychologists Press, Inc., Palo Alto, California, USA, 1978.

[Ekm79] P. Ekman. The argument and evidence about universals in facial expressions of emotion. *Handbook of Social Psychopysiology*, pages 143–146, 1979.

[ES93] Larry J. Eshelman and J. David Schaffer. Crossover's niche. In Stephanie Forrest, editor, *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 9–14, San Mateo, CA, USA, July 1993. Morgan Kaufmann.

[FB90] S. Feiner and C. Beshers. Worlds within worlds: metaphors for exploring n-dimensional virtual worlds. In ACM, editor, *UIST. Third Annual Symposium on User Interface Software and Technology. Proceedings of the ACM SIGGRAPH Symposium, Snowbird, Utah, USA, October 3–5, 1990*, pages 76–83, New York, NY 10036, USA, October 1990. ACM Press.

[Fie79] S. E. Fienberg. Graphical methods in statistics. *American Statisticians*, 33:165–178, 1979.

[GE97] Nahum D. Gershon and Stephen G. Eick. Guest editors' introduction: Information visualization. *IEEE Computer Graphics and Applications*, 17(4):29–31, July/August 1997.

[GEC98] Nahum Gershon, Stephen G. Eick, and Stuart Card. Design: Information visualization. *interactions*, 5(2):9–15, 1998.

[Gol89] Goldberg. Genetic algorithms in search, optimization, and machine learning. In *Addison-Wesley Publishing Company (Addison Wesley Longman, Inc.)*. 1989.

[GU94] M.W.E. Glautier and B. Underdown. *Accounting Theory and Practice*. Pitman Publishing, 1994.

[Ham73] R. W. Hamming. *Numerical Analysis for Scientists and Engineers*, 1973.

[HDWB95] R. J. Hendley, N. S. Drew, A. M. Wood, and R. Beale. Narcissus: Visualising information. In Nahum D. Gershon and Steve Eick, editors, *Proc. IEEE Symp. Information Visualization, InfoVis*, pages 90–96. IEEE Computer Soc. Press, 30–31 October 1995.

[HE99] C. Healy and J. Enns. Large datasets at a glance: Combining textures and colours in scientific visualisation. *IEEE transactions on visualisation and computer graphics*, 5(2):145–167, 1999.

[Hol75] John H. Holland. *Adpatation in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor, MI, 1975.

[HS76] Haver Homa and Schwartz. Perceptbility of schematic face stimuli: evidence of a perceptual gestalt. *Journal of Memory and Cognition*, 4, 1976.

[HS97] Kim Harries and Peter Smith. Exploring alternative operators and search strategies in genetic programming. In John R. Koza, Kalyanmoy Deb, Marco Dorigo, David B. Fogel, Max Garzon, Hitoshi Iba, and Rick L. Riolo, editors, *Genetic Programming 1997: Proceedings of the Second Annual Conference*, pages 147–155, Stanford University, CA, USA, 13-16 July 1997. Morgan Kaufmann.

[Hue96] Lorenz Huelsbergen. Toward simulated evolution of machine language iteration. In John R. Koza, David E. Goldberg, David B. Fogel, and Rick L. Riolo, editors, *Genetic Programming 1996: Proceedings of the First Annual Conference*, pages 315–320, Stanford University, CA, USA, 28–31 July 1996. MIT Press.

[IG96] Hitoshi Iba and Hugo Garis. Extending genetic programming with recombinative guidance. In Peter J. Angeline and K. E. Kinnear, Jr., editors, *Advances in Genetic Programming 2*, chapter 4, pages 69–88. MIT Press, Cambridge, MA, USA, 1996.

[Ins84] A. Inselberg. Parallel coordinates for multidimensional displays. In *Spatial Information Technologies for Remote Sensing Today and Tomorrow, The Ninth William T. Pecora Memorial Remote Sensing Symposium*, pages 312–324. IEEE Computer Society Press, 1984.

[Ins97] A. Inselberg. Multidimensional detective. In *IEEE Symposium on Information Visualization (InfoVis '97)*, pages 100–107, Washington - Brussels - Tokyo, October 1997. IEEE.

[Jac96] J. E. Jackson. *A User's Guide to Principal Components.* John Wiley and Sons, 1996.

[Jer99] Mikael Jern. Visual intelligence turning data to knowledge. In *International Conference on Information Visualisation.* IEEE, 1999.

[JLR94] Walker J., Sproull L., and Subramani R. Using a human face in an interface. In *Human Factors in Computer Systems*, pages 85–91. ACM, 1994.

[Jon95] Terry Jones. Crossover, macromutation, and population-based search. In Larry J. Eshelman, editor, *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 73–80, San Francisco, July15–19 1995. Morgan kaufmann Publishers.

[JP96] Hugues Juille and Jordan B. Pollack. Dynamics of co-evolutionary learning. In Pattie Maes, Maja J. Mataric, Jean-Arcady Meyer, Jordan Pollack, and Stewart W. Wilson, editors, *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior: From animals to animats 4*, pages 526–534, Cape Code, USA, 9-13 September 1996. MIT Press.

[Kei96] Maarten Keijzer. Efficiently representing populations in genetic programming. In Peter J. Angeline and K. E. Kinnear, Jr., editors, *Advances in Genetic Programming 2*, chapter 13, pages 259–278. MIT Press, Cambridge, MA, USA, 1996.

[Kei97] Daniel Keim. Visual techniques for exploring databases. In *3rd Int. Conf. Knowledge Discovery and Data Mining, KDD*, 14–17 August 1997.

[Kin93] Kenneth E. Kinnear, Jr. Evolving a sort: Lessons in genetic programming. In *Proceedings of the 1993 International Conference on Neural Networks*, volume 2, San Francisco, USA, 1993. IEEE Press.

[KM94] Mike J. Keith and Martin C. Martin. Genetic programming in C++: Implementation issues. In Kenneth E. Kinnear, Jr., editor, *Advances in Genetic Programming*, chapter 13, pages 285–310. MIT Press, 1994.

[Koz92] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* MIT Press, 1992.

[Koz94] John R. Koza. Introduction to genetic programming. In Kenneth E. Kinnear, editor, *Advances in Genetic Programming*, Complex Adaptive Systems, pages 21–42, Cambridge, 1994. MIT Press.

[Lan96] W. B. Langdon. Directed crossover within genetic programming. Internal note, University College London, 1996.

[Lau71] Laughery. et al. recognition of human faces. *Journal of Applied Psychology*, 51:447–483, 1971.

[LNH02] R. Schulman L. Nowell and D. Hix. Graphical encoding for information visualisation: An empirical study. In *IEEE Symposium on Information Visualization (InfoVis '02)*, pages 43–50, Washington - Brussels - Tokyo, October 2002. IEEE.

[LP01] W. B. Langdon and Riccardo Poli. *Foundations of Genetic Programming.* Springer, 2001.

[LQ95] William B. Langdon and Adil Qureshi. Genetic programming – computers using "natural selection" to generate programs. Research Note RN/95/76, University College London, Gower Street, London WC1E 6BT, UK, October 1995.

[LS87] Jill H. Larkin and Herbert A. Simon. Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, 11:65–99, 1987.

[LS99] A. Loizides and M. Slater. An application of the empathetic visualisation algorithm (eva). Research Note RN/99/50, Department of Computer Science, University College London, 1999.

[LS01] Andreas Loizides and Mel Slater. The empathic visualisation algorithm, chernoff faces revisited. In *Technical Sketch, ACM Siggraph 2001 Conference Abstracts and Applications*, page 175. ACM Siggraph, 2001.

[LS02] Andreas Loizides and Mel Slater. The empathic visualisation algorithm (eva) - an automatic mapping from abstract data to naturalistic visual structure. In *Proceedings of Information Visualisation, Sixth International Conference on Information Visualisation*, pages 705–712. IEEE, JulySeptember 2002. Published 2002.

[LS03] Andreas Loizides and Mel Slater. The empathic visualisation algorithm (eva) and its application on real data. *Ylem Journal: Artists Using Science and Technology*, 23(12):11–13, November-December 2003.

[LSL01] Andreas Loizides, Mel Slater, and William B. Langdon. Measuring facial emotional expressions using genetic programming. In Rajkumar Roy, Mario Köppen, Seppo Ovaska, Takeshi Furuhashi, and Frank Hoffmann, editors, *Soft Computing and Industry Recent Applications*, pages 545–554. Springer-Verlag, 10–24 September 2001. Published 2002.

[Lux97] M. Lux. Visualisation of financial information. In *Workshop no New Paradigms in information visualisation and manipulation*, pages 58–61. NPIV' 97, 1997.

[Mac86] Jock Mackinlay. Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics*, 5(2):110–141, April 1986.

[MB93] S. Margarita and A. Beltratti. Stock prices and volume in an artificial adaptive stock market. *New trends in Neural Computation: International Workshop on Artificial Networks*, pages 714–719, 1993.

[Mon95] David J. Montana. Strongly typed genetic programming. *Evolutionary Computation*, 3(2):199–230, 1995.

[Mor79] Moriarity. Communicating financial information through multi-dimensional graphics. In *Journal of Accounting Research*, volume 17, pages 205–224, 1979.

[MTS91] Ted Mihalisin, John Timlin, and John Schwegler. Visualizing multivariate functions, data, and distributions. *IEEE Computer Graphics and Applications*, 11(3):28–35, May 1991.

[Nor93] Donald A. Norman. *Things That Make Us Smart: Defending Human Attributes in the Age of the Machine.* Addison-Wesley Publishing Perseus, Reading, Massachusetts, 1993. ISBN 0-201-62695-0.

[Nor94] Peter Nordin. A compiling genetic programming system that directly manipulates the machine code. In Kenneth E. Kinnear, Jr., editor, *Advances in Genetic Programming*, chapter 14, pages 311–331. MIT Press, 1994.

[OLS93] Masakazu Osada, Holmes Liao, and Ben Shneiderman. Alphaslider: Searching textual lists with sliders. Technical Report CS-TR-3078, University of Maryland, College Park, April 1993.

[Par72] Frederic I. Parke. Computer generated animation of faces. *Proc. ACM annual conf.*, August 1972.

[Par82] Frederic I. Parke. Parameterized models for facial animation. *IEEE Computer Graphics and Applications*, 2(9):61–64, 66–68, November 1982.

[Pau66] G. Paul. *Insight vs. Desensitisation in Psychotherapy: An Experiment in Anxiety Reduction*, volume 8, page 148. Stanford University Press, Stanford, California, 1966.

[PG88] R. M. Pickett and G. G. Grinstein. Iconographic displays for visualising multidimensional data. In *Proceedings of IEEE Conference on Systems, Man and Cybernetics*, pages 514–519, Piscataway, NJ, USA, 1988. IEEE Press.

[PW96] Frederik I. Parke and Keith Waters. *Computer Facial Animation.* A K Peters, Ltd, 1996.

[RB94] Justinian P. Rosca and Dana H. Ballard. Hierarchical self-organization in genetic programming. In *Proc. 11th International Conference on Machine Learning*, pages 251–258. Morgan Kaufmann, 1994.

[RC95] Ramana Rao and Stuart K. Card. Exploring large tables with the table lens. In *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, volume 2 of *Videos*, pages 403–404, 1995.

[Rec73] I. Rechenberg. *Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien der Biologischen Evolution.* Frommann-Holzboog, Verlag, Stutgart, 1973.

[RKM+94] Steven F. Roth, John Kolojejchick, Joe Mattis, Mei C. Chuah, Jade Goldstein, and Octavio Juarez. SAGE tools: A knowledge-based environment for designing and perusing data visualizations. In *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems*, volume 2 of *DEMONSTRATIONS: Demonstrational Interfaces*, pages 27–28, 1994.

[RPR96] Josephine M. Randel, H. Lauren Pugh, and Stephen K. Reed. Differences in expert and novice situation awareness in naturalistic decision making. *International Journal of Human-Computer Studies*, 45(5):579–597, 1996.

[SE91] J. David Schaffer and Larry J. Eshelman. On crossover as an evolutionary viable strategy. In Richard K. Belew and Lashon B. Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms (ICGA'91)*, pages 61–68, San Mateo, California, 1991. Morgan Kaufmann Publishers.

[Shn93] Ben Shneiderman. Dynamic queries: for visual information seeking. Technical Report CS-TR-3022, University of Maryland, Department of Computer Science, January 1993.

[Shn96] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. Technical Report CS-TR-3665, University of Maryland, College Park, July 1996.

[Sla99] M. Slater. From data to visual structure: An automatic mapping. Research note, Department of Computer Science, University College London, 1999.

[Spe01] Robert Spence. *Information Visualization.* ACM Press, Addison-Wesley, 2001.

[Spi83] C. D Spielberger. *Manual for the State-Trait Anxiety Inventory (STAI).* Consulting Psychologists Press, PaloAlto, CA, 1983.

[SR96] Mike Scaife and Yvonne Rogers. External cognition: How do graphical representations work? *International Journal of Human-Computer Studies*, 45(2):185–213, 1996.

[ST96] G. M. Smith and R. Taffler. Improving the communication of accounting information through cartoon graphics. In *Accounting, Auditing and Accountability Journal*, volume 9, pages 68–85, 1996.

[Sys91] Gilbert Syswerda. A study of reproduction in generational and steady state genetic algorithms. In Gregory J. E. Rawlins, editor, *Proceedings of the First Workshop on Foundations of Genetic Algorithms*, pages 94–101, San Mateo, July 15– 18 1991. Morgan Kaufmann.

[Tac94] Walter Alden Tackett. *Recombination, Selection, and the Genetic Construction of Computer Programs.* PhD thesis, University of Southern California, Department of Electrical Engineering Systems, USA, 1994.

[TC94] W. A. Tackett and A. Carmi. The unique implications of brood selection for genetic programming. In *Proceedings of the 1994 IEEE World Congress on Computational Intelligence*, Orlando, Florida, USA, 27-29 June 1994. IEEE Press.

[Tel96] Astro Teller. Evolving programmers: The co-evolution of intelligent recombination operators. In Peter J. Angeline and K. E. Kinnear, Jr., editors, *Advances in Genetic Programming 2*, chapter 3, pages 45–68. MIT Press, Cambridge, MA, USA, 1996.

[TSWB94] Lisa Tweedie, Bob Spence, David Williams, and Ravinder Bhogal. The attribute explorer. In *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems*, volume 2 of *VIDEOS: Part I: Browsing Navigation*, pages 435–436, 1994.

[Tuf83] Edward R. Tufte. *The Visual Display of Quantitative Information.* Graphics Press, Cheshire, Connecticut, U.S.A., 1983.

[Tuf90] Edward R. Tufte. *Envisioning Information.* Graphics Press, Chechire, Connecticut, 1990.

[War00] Colin Ware. *Information Visualization: Perception for Design.* Morgan Kaufmann Publishers, 2000.

[Wat87] K. Waters. A muscle model for animating three-dimensional facial expression. In M. C. Stone, editor, *SIGGRAPH '87 Conference Proceedings (Anaheim, CA, July 27–31, 1987)*, pages 17–24. Computer Graphics, Volume 21, Number 4, July 1987.

[Wil82] Wilkinson. An experimental evaluation of multivariate graphical point representations. In *Human Factors in Computer Systems*, pages 202–209. ACM, 1982.

[WS92] Christopher Williamson and Ben Shneiderman. The dynamic homefinder: Evaluating dynamic queries in a real-estate information exploration system. Technical Report CS-TR-2819, University of Maryland, College Park, January 1992.

[You96] P. Young. Three dimensional information visualisation. Research note, Department of Computer Science, University of Durham, 1996.

[ZR97] Elena Zannoni and Robert G. Reynolds. Learning to control the program evolution process with cultural algorithms. *Evolutionary Computation*, 5(2):181–211, summer 1997.