‘Multi View Graphing’: Linked Multi Visualization utilising Brushing, Binning and Clustering

Citation for published version (APA):

Please note that where the full-text provided on Manchester Research Explorer is the Author Accepted Manuscript or Proof version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version.

General rights
Copyright and moral rights for the publications made accessible in the Research Explorer are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Takedown policy
If you believe that this document breaches copyright please refer to the University of Manchester's Takedown Procedures [http://man.ac.uk/04Y6Bo] or contact uml.scholarlycommunications@manchester.ac.uk providing relevant details, so we can investigate your claim.
'Multi View Graphing': Synchronous Linked Multi Visualization utilising Brushing, Binning and Clustering

A dissertation submitted to The University of Manchester for the degree of Master of Science in the Faculty of Engineering and Physical Sciences.

2007

Stephen Longshaw

School of Computer Science
Contents

Abstract ................................................................................................................................. 1
Declaration ............................................................................................................................ 2
Copyright .............................................................................................................................. 3
Acknowledgements ............................................................................................................... 4
1.0 Introduction ............................................................................................................... 5
  1.1 Problem Domain ...................................................................................................... 5
  1.2 Background ............................................................................................................. 9
  1.3 Goals ...................................................................................................................... 10
  1.4 Organisation .......................................................................................................... 11
2.0 Prior Research ......................................................................................................... 13
  2.1 Synchronous Linked Multi Visualization .............................................................. 13
     2.1.1 Visualization Styles ........................................................................................ 14
     2.1.2 Synchronous Linked Visualization .................................................................. 20
  2.2 Data Brushing ........................................................................................................ 22
     2.2.1 Brush Parameters ............................................................................................ 23
     2.2.2 Brush Shapes ................................................................................................... 25
  2.3 Data Binning .......................................................................................................... 30
     2.3.1 Binning Techniques ........................................................................................ 31
  2.4 Data Clustering and Classification ........................................................................ 34
     2.4.1 Clustering Algorithms .................................................................................... 35
     2.4.3 Presentational Methods ................................................................................... 36
3.0 ‘Multi View Graphing’ ........................................................................................... 39
   3.0.1 The Visualization Pipeline .............................................................................. 39
   3.1 Synchronous Linked Multiple Visualization ....................................................... 40
      3.1.1 Sharing the Data ............................................................................................ 41
      3.1.2 Visualization Styles ...................................................................................... 41
      3.1.3 Cross Visualization Interaction ..................................................................... 50
   3.2 Data Brushing ........................................................................................................ 51
      3.2.1 Brush Shapes ................................................................................................ 53
      3.2.2 Brushing Groups .......................................................................................... 60
      3.2.3 Data Visibility ................................................................................................. 61
A.3.1 General Options ........................................................................................... 128
A.3.2 Data Brushing .............................................................................................. 128
A.3. Data Binning .................................................................................................. 130
A.4 Parallel Coordinates ............................................................................................ 132
  A.4.1 General Options ........................................................................................... 132
  A.4.2 Data Brushing .............................................................................................. 135
  A.4.3 Data Binning ................................................................................................ 136
A.5 Star Glyphs ......................................................................................................... 138
  A.5.1 Data Brushing .............................................................................................. 138
A.6 Data Grid ............................................................................................................ 140
  A.6.1 Brushed Data .............................................................................................. 140
  A.6.2 Binned or Clustered Data ............................................................................. 141
A.7 Linked Interaction .............................................................................................. 142
8.0 Appendix B: OpenGL Geometry Lists within MVG ......................................... 144
9.0 Appendix C: Standardised Test Case Questionnaires .................................... 146
List of Figures

Figure 1.1 Radar Plot showing a small subset of an 11 variable dataset .................................. 5
Figure 1.2 'SpiralGlyph' representing a subset of a New York Stock Exchange dataset .......... 6
Figure 1.3 Two examples of data selection via brushing ......................................................... 8

Figure 2.1 Scatter Plot Matrix revealing a remote sensor dataset ........................................... 15
Figure 2.2 Chernoff Face Glyphs ............................................................................................ 16
Figure 2.3 Star Glyphs .............................................................................................................. 17
Figure 2.4 Visual representation of the Parallel Coordinates concept ................................... 18
Figure 2.5 Working example of a Parallel Coordinate visualization ..................................... 19
Figure 2.6 The three main presentational methods used for Synchronous Linked Multi
Visualization .......................................................................................................................... 21
Figure 2.7 Cross interaction between two visual styles .......................................................... 22
Figure 2.8 Brush 'expression' builder within 'XmdvTool' ......................................................... 24
Figure 2.9 Examples of deformable rectangular brushing within information visualization packages .......................................................... 25
Figure 2.10 'Paint Daub' brush shape within 'Mirage' ............................................................... 26
Figure 2.11 Three Dimensional scatter plot brushing ............................................................... 27
Figure 2.12 Angle based brushing within Parallel Coordinates ............................................. 28
Figure 2.13 Parallel Coordinates angular quadrilateral brush style ....................................... 29
Figure 2.14 Histogram example .............................................................................................. 31
Figure 2.15 Histogram overlaid on a Parallel Coordinates visualization ................................ 33
Figure 2.16 Cluster and Classifier outputs within 'Weka' ....................................................... 37
Figure 2.17 Visualization showing the results from a K-means clustering run ...................... 38

Figure 3.1 Haber-McNabb Reference Model with direct comparison to MVG ..................... 40
Figure 3.2 Visual representation of the concept of a computer generated two
dimensional scatter plot ........................................................................................................ 43
Figure 3.3 Example of two dimensional scatter plot as generated by MVG ......................... 44
Figure 3.4 Example of three dimensional scatter plot as generated by MVG ..................... 45
Figure 3.5 Visual representation of the concept of the Parallel Coordinates visualization .................................................................................................................. 46
Figure 3.6 Examples of the Parallel Coordinates visualization generated by MVG .......... 47
Figure 3.7 Visual representation of the Star Glyph concept .................................................... 49
Figure 3.8 Example of the Star Glyph visualization generated by MVG ................................ 50
Figure 3.9 Illustrated examples of the solutions to the problem of semi brushed data
points within MVG .................................................................................................................. 52
Figure 3.10 Deformable Rectangular brushing within MVG .................................................. 54
Figure 3.11 Annotated visual representation of the circular brushing concept within
MVG ...................................................................................................................................... 55
Figure 3.12 Resizeable circular brushing within MVG ............................................................. 56
Figure 3.13 Annotated visual representation of the Quadrilateral brushing concept within MVG ................................................................. 58
Figure 3.14 Quadrilateral brushing within MVG ........................................ 58
Figure 3.15 Three Dimensional 'Cube' brushing within MVG .................. 59
Figure 3.16 Three individual brushing groups within MVG ..................... 60
Figure 3.17 Data visibility alteration within MVG ..................................... 62
Figure 3.18 Binning within MVG using a two dimensional scatter plot and the arithmetic mean methodology ......................................................... 65
Figure 3.19 Binning within MVG using a three dimensional scatter plot and the arithmetic mean methodology ......................................................... 66
Figure 3.20 Two dimensional scatter plot showing the first letter compared against the second letter from a sample 'Scrabble' dataset ......................................................... 68
Figure 3.21 Binned two dimensional scatter plot showing the first letter compared against the second letter from a sample 'Scrabble' dataset ......................................................... 69
Figure 3.22 Density Map Binning within MVG using two dimensional scatter plots ......................................................... 71
Figure 3.23 Density Map Binning within MVG using a highly cluttered dataset ......................................................... 72
Figure 3.24 Density Map Binning within MVG using the Parallel Coordinates visual style ......................................................... 73
Figure 3.25 Two Dimensional scatter plot showing the first letter compared against the second letter from a sample 'Scrabble' dataset ......................................................... 74
Figure 3.26 Density binned Two Dimensional scatter plot showing the first letter compared against the second letter from a sample 'Scrabble' dataset ......................................................... 75
Figure 3.27 Clustering within MVG using a two dimensional scatter plot ......................................................... 77
Figure 3.28 Pseudo Code description of MVG's clustering algorithm .......... 79

Figure 4.1 Parallel Coordinates within MVG showing Knee and Stimuli values from nerve stimulation dataset ......................................................... 83
Figure 4.2 Parallel Coordinates visualization produced using MVG showing two distinct groups of values within the Anterior nerve stimulation dataset ......................................................... 84
Figure 4.3 Parallel Coordinates within MVG showing all 11 variables with brushing group definitions ......................................................... 85
Figure 4.4 Three Dimensional Scatter plot within MVG ......................................................... 86
Figure 4.5 Each stimuli value against every other from the Anterior ......................................................... 87
Figure 4.6 Quadrilateral brush in use within MVG ......................................................... 89
Figure 4.7 Two Dimensional scatter plot within MVG, showing the data subset used to test clustering functionality ......................................................... 90
Figure 4.8 The results of a clustering the test case within MVG .................. 91
Figure 4.9 Example high contrast MRI image ......................................................... 93
Figure 4.10 Highly cluttered Parallel Coordinates within MVG with binning enabled ......................................................... 96
Figure 4.11 Brushing groups used for MRI analysis ......................................................... 98
Figure 4.12 Star Glyphs within MVG being used to show MRI data ......................................................... 98
Figure 4.13 Brushed Parallel Coordinates within MVG being used to show MRI data ......................................................... 99
Figure 4.14 Initial Hadley data Parallel Coordinates within MVG .................. 103
Figure 4.15 Highly zoomed Parallel Coordinates of Hadley data within MVG ........ 104
Figure 4.16 Highly zoomed Parallel Coordinates of Hadley data within MVG, with brushing applied ................................................................. 104
Figure 4.17 Two dimensional scatter plot within MVG showing brushed Hadley data 105
Figure 4.18 Two dimensional scatter plot within MVG showing modified brushed Hadley data ........................................................................ 106
Figure 4.19 Two dimensional scatter plot within MVG of the Hadley data, with arithmetic mean binning applied .......................................................... 107
Figure 4.20 Two dimensional scatter plot within MVG of the Hadley data, with density map binning applied .............................................................. 108
Figure 4.21 Two dimensional scatter plot within MVG of the Hadley data, with density map binning applied and white set to high ........................................ 108
Abstract

Visualization can provide a distinctly advantageous overview of data, enabling the rapid identification of anomalies, patterns or correlations that would not otherwise be obvious. Different visualization techniques each offer their own unique insight into the same data; however the similarities that exist between them are not always clear. High data density can also be a very evident issue when exploring data using visualization. The densest datasets can ensure that even well suited visualization methodologies succumb to usability issues. The most powerful data analysis environments are arguably those that provide interactive exploration, however visual feedback in such environments is sometimes undesirably limited. The concept of linking different visualization styles using interactive techniques, such as brushing, is currently evident in multiple publically available software environments. To explore the concept of linked visualization a prototype application was produced, allowing up to four unique visual styles to be generated using the same data, at the same time. Current brushing methodologies were extended and included, in order to provide the ability to affect each visualization from within every other. The issue of data density was tackled through the use of a novel approach to binning based around a uniform grid. Visual cues were used extensively throughout the prototype, ranging from representing a brushing area through to defining the basic starting parameters of a clustering algorithm. Three distinctly different test cases are presented to demonstrate the techniques showcased within the prototype, each in conjunction with external collaborators. Results suggest that using linked multi visualization is a more effective method of data analysis, offering greater insight than using a lone visualization technique. In tackling data density, the grid based binning has the ability to offer an easily disseminated overview of even extremely cluttered visualizations. The extensive use of visual cues in the prototype vindicated the theory that offering clear feedback within interactive environments is of the utmost importance. Also the interactive definition of clustering parameters via visual cues shows promise as a concept but one requiring further research. This study highlights that the current trend towards linking multiple visualization techniques within advanced data analysis environments is correct; it also introduces novel brushing, binning and clustering concepts worthy of further investigation.
Declaration

No portion of the work referred to in the dissertation has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.
Copyright

(1) Copyright in text of this dissertation rests with the author. Copies (by any process) either in full, or of extracts, may be made only in accordance with instructions given by the author. Details may be obtained from the appropriate Graduate Office. This page must form part of any such copies made. Further copies (by any process) of copies made in accordance with such instructions may not be made without the permission (in writing) of the author.

(2) The ownership of any intellectual property rights which may be described in this dissertation is vested in the University of Manchester, subject to any prior agreement to the contrary, and may not be made available for use by third parties without the written permission of the University, which will prescribe the terms and conditions of any such agreement.

(3) Further information on the conditions under which disclosures and exploitation may take place is available from the head of the School of Computer Science (or the Vice-President and Dean of the Faculty of Engineering and Physical Sciences.)
Acknowledgements

The author would like to thank the following people for their support during the production of this dissertation, without those listed here; it would not have been possible in its current form.

- Dr. Martin Turner, for his invaluable insight and support in seeing everything through to completion.
- Professor. Terry Hewitt, for his constructive candour and the generous use of his personal library.

The author would also like to thank all those involved in the invaluable test cases:

- Mr. Peter Hoornaert of the School of Computer Science, Staffordshire University, for spending many hours of his time testing the prototype and also for imparting the author with his enthusiasm for visualization.
- Dr. Samantha Mills of the School of Medicine, the University of Manchester, for her time and notable effort.
- Louise. M. Lever of the Manchester Visualization Centre, for her extensive effort in manipulation of the Hadley dataset and invaluable input.
- The Hadley Centre for Climate Research for the kind use of their climate model.

Finally the author wishes to thank those who helped along the way:

- My Grandparents, for providing the necessities of life while I was otherwise occupied, for taking the time and effort to offer constructive criticism when asked and most importantly for giving me the gift of science while I was growing up.
- My parents for their unmitigated support regardless of the circumstances.
- All friends and other family members who have kindly offered their support and advice throughout this endeavour.
1.0 Introduction

1.1 Problem Domain

Large highly dimensional datasets provide numerous challenges when visualization is attempted. The main areas of concern are how to usefully represent more than two data dimensions at once and how to present very dense or cluttered data within the confines of a visual display.

Very rarely do datasets contain two variables or less, therefore it is unusual for a simple visualization style such as a two dimensional scatter plot to be sufficient. To overcome this problem many different visual styles have been conceptualised, all falling into one of two main categories, being based either around a coordinate system as seen in figure 1.1 or around abstract data encoding, also known as Glyph representation, as seen in figure 1.2. Generally speaking a Glyph is an abstracted graphical representation of specific values. Glyphs are sometimes intermixed with a coordinate system to either increase the number of variables that can be represented or perhaps to give context to the Glyphs through their geometric placement.

![Radar Plot showing a small subset of an 11 variable dataset, example generated using Microsoft Excel (Microsoft, 2007) using the Lumbar Anterior Root Dataset (Wood et al., 1999)](image_url)
While much effort is placed into finding new ways to visualize multivariate data using a single visual style, an aspect that is often overlooked is that no lone visualization is likely to ever be ideal for all aspects of data analysis. With this in mind there currently exist software tools such as ‘XmdvTool’ (Ward, 1994) which provide a generic data exploration environment offering the user the ability to view their data in multiple visual styles, exploiting the strength of one style to overcome the weakness of another. This provides an environment where a partially symbiotic relationship exists between the differing visualization styles, without one style the rest would be ultimately less useful for analytical purposes. Less generic specialised application environments such as ‘GeoDa’ (Anselin et al., 2006) also exist, employing the concept of multiple linked visualizations to facilitate the better analysis of a specific type of dataset.

An issue which is common to all visualization is that of how to present interactive elements to the user. Visual cues are a simple yet effective way to highlight exactly what a user can do or what they have done previously. Graphical cues are used in nearly every area of visual computing, from user interfaces through to highly interactive three dimensional scenes. As an example, it is expected within a graphical file browser that if you click and hold the mouse button, a rectangular selection area will appear and define
which files are to be selected. Visual cues are of the utmost importance within information visualization systems, offering the ability to show the user the effect of their actions or perhaps define an abstract concept such as a numerical boundary. It is important that visual cues are used whenever possible but they should never be intrusive or misrepresentative. The successful inclusion of visual feedback is an area of differing success within current visualization systems.

A problem which is common to all visualization techniques is that of data density; specifically the visual clutter that high density data almost invariably leads too. Highly dense datasets present many challenges within the context of computing, these range from the computational power and memory required to process large datasets, through to the amount of physical space available on a screen. High density cluttered data is by definition more difficult to visually analyse than low density data. A major goal of a system intending to allow exploration of a highly dense dataset must be to provide ways in which the data can either be thinned or at least viewed at a less detailed level.

The most basic method provided in visualization for specific data definition is that of interactive selection. The method by which this is offered varies widely from system to system but is generally referred to as ‘Brushing.’ There are many different styles of data brushing and arguably one is not necessarily better than another, rather the worth of a brushing method can be defined purely by the dataset being analysed or the visualization style being used. Selecting the best way of offering brushing to maximise the usefulness of the visualizations on offer is a primary concern when developing a visualization system. The two examples shown in figure 1.3 highlight that different brush styles can be suitable for different visualizations. The simple two dimensional scatter plot on the left uses a very simple rectangular selection area whereas the more complex Parallel coordinates visualization on the right uses a coloured quadrilateral to define which lines should be brushed.
If the system offers multiple visualization styles then the most important factor is whether a brushing technique that is suitable for one visual style is suitable for another. Beyond the act of simple data selection, techniques such as binning and clustering are also commonly used as methods to either reduce or reorganise dense datasets.

Binning refers to the act of collating data values that fall within a set boundary. The most commonly used form of binning within information visualization is that of the Histogram, which is usually presented as a sequence of bars with the boundaries that define each bar shown in place of variable names. Traditionally this form of binning is used in statistical analysis as it is a quick and intuitive method to find values of interest such as data frequency. To make best use of binning as a tool within a visual environment, the challenge is how best to integrate the concept of data boundaries or a ‘bin’ into the visualizations themselves. Ideally this should take the focus away from the numerical statistics and instead provide a less cluttered overview within the context of the original visualization.

At first glance, data binning and data clustering are easy to mix up, however the fundamental purpose of a clustering algorithm is different to that of a binning system. Binning intends to collate existing data, generating sub classes within the dataset that represent the more cluttered actual data below. The goal of clustering is generally to identify a data subset whose members all share a commonality. Often this is either
through geometrical properties such as proximity, or if this is inapplicable due to a lack of coordinate system through parameters which link the data together.

The method by which clustering algorithms are presented to a user is an area which perhaps lacks analysis. Much work has been done on the algorithms themselves but the method of application is often neglected, resulting in the user being presented with a non descriptive list of available algorithms and no insight as to what the algorithm is going to do until it has been executed. Defining the parameters of a clustering algorithm using visual cues and user interaction could provide a quicker and easier interface for some than simply allowing access to collections of pre-defined algorithms to be run covertly on a set of data.

1.2 Background

The interest in this project and the development of the software artefact to support it revolves around the concept of linking multiple visualizations that are generated from the same data, via user interaction. Prior analytical work of a dataset collected by Professor N. Donaldson at University College London (Wood et al., 1999), led to the conclusion that multidimensional visualization was extremely important for highly dimensional datasets, however the use of advanced visualization techniques intermixed and directly linked to more common visual styles should provide a far more powerful general exploratory environment than the use of one single visualization technique alone.

Previous analysis of the dataset had revealed very little, yet the correlation being sought was theoretically a relatively simple one. The dataset contains seven movement values and four stimulus values. A reliable correlative lookup between the two was desired so that values for the stimulus could be applied and the resulting movements occur as expected. Past efforts included neural net analysis and pseudo three dimensional viewing, applying user controlled nearest neighbour data collection (Hoornaert, 1999). The hypothesis was that if a line was drawn between a desirable start and end point within the movement values, the corresponding stimulus values would be the ones that produced that movement. Eventually this proved ultimately unsuccessful as an
analytical method. Recently through research that has resulted in established software environments such as ‘XmdvTool’ (Ward, 1994) interest has been rekindled as to whether the data could be better analysed using a collection of well defined linked techniques.

Software environments such as ‘XmdvTool’ and ‘Ggobi’ (Swayne et al., 2001), allow for data to be cross analysed using different visualization styles linked via user interaction. It was felt that through the use of a modern graphical library such as OpenGL, the power of current graphics hardware should allow for the production of a comparable software environment with enhanced interactive capabilities. An advanced API should also allow for more successful application of visual techniques such as transparency and pseudo three dimensional representations.

The application of linked interaction can take many different forms; well established techniques were therefore chosen for further exploration. Concepts such as data brushing, binning and clustering were all implemented. These techniques differ widely in their respective implementations; the concepts applied within this project are therefore studies into the possible advancements on current techniques that should improve the usability of linked, highly interactive data exploration environments.

It was decided that convincingly proving the worth of a tool produced to demonstrate a relatively large set of concepts may be difficult if it was engineered to only accept one specific dataset. This led to a less specialised effort, geared instead towards analysis of numerical multidimensional datasets in general.

1.3 Goals

The primary goal of this dissertation is to explore the concept of synchronous linked multi visualization, to prove the worth of linking interaction within different visual styles and show that the overall success of a visualization environment lies not in its individual components, but in the net result of their interactions.

For this task simply sharing the same data set is not enough, therefore actions taken on one visualization will be reflected in every other as far as is possible. To demonstrate
the concept, the application of accepted techniques such as data brushing, binning and clustering will be available individually on each visualization, but the results of each action will be reflected, with as close an approximation as the differing visual styles allow, on every other. An attempt will also be made to show that presenting traditionally numerical techniques such as binning and clustering in a more visual form can have merit, allowing quicker and easier access to these powerful visualization concepts.

Due to the relatively non deterministic and personal nature of visualization, proof of concept is deemed to be a very important factor. To this end multiple test cases are presented and evaluated in detail in chapter 4. External contributors are integral to each case, providing multiple external opinions on the validity of the work presented.

1.4 Organisation

This chapter presents a concise overview of the topics that this dissertation intends to address, including a brief overview of the challenges, an insight as to the intended purpose of the dissertation and an outline of the major goals intended for exploration.

Chapter 2 presents an overview of past research undertaken in appropriate areas to the dissertation.

Chapter 3 contains a descriptive overview of the new work presented. Due to the wide focus of topics this dissertation covers, chapter 3 is subdivided into four major areas; including synchronous linked multi visualization, data brushing, binning and clustering.

The three test cases that are presented as proof are documented in Chapter 4; each case is contained within its own sub section. While the content and participant for each case may be different, all three are outlined in the same manner so as to provide consistency for comparison of results. The three test cases are as follows:

- **Lumbar Anterior Root Stimulation Dataset**: A highly refined dataset with a long history of prior analysis. Study is in association with the school of computer science, Staffordshire University. Specifically Mr. P. Hoornaert.
• **WMIC MRI Data:** Analysis of MRI scans involving brain tumours is an ongoing and ever expanding field of study. This test case will look at numerical and statistical analysis of such MRI data. Study is in association with the school of medicine, The University of Manchester. Specifically Professor. A. Jackson and Dr. S. Mills.

• **Hadley Centre Weather Data (UK Met Office):** The Hadley weather centre has provided a predictive dataset showing the most likely path for global climate change. It is currently being visually represented using volume techniques; this test case looks at numerical analysis of the data. Study is in association with the Manchester Visualization Centre, The University of Manchester. Specifically Louise M. Lever.

Finally the conclusions of the dissertation are detailed in Chapter 5 along with any recommended future work.
Chapter 2

2.0 Prior Research

Information Visualization is often an essential part of any exploratory data analysis. Examples of large amounts of data being represented visually can be traced back through the annuls of time. The works of Minard such as the *Carte Figurative* depicting the numerical story of Napoleon's advance and subsequent retreat through Russia, between 1812 and 1813 (Tufte, 2001), shows immediately the strength of visually representing data compared to dealing directly with numbers. In this example the correlation between the number of men dying and the falling temperatures is quickly recognisable in a visual form.

Within modern scientific visualization, datasets are often so large and so highly dimensional that the ability to quickly, easily and automatically produce useful visual representation is essential although difficult to implement. Arguably even more important is the ability to then interact with the resultant images, to highlight specific data, cluster and simplify clutter and gain insight within overview. Techniques that have received much interest include brushing (Henze, 1998) (Wright and Roberts, 2005), binning (Sanders and Fabian, 2001), clustering (Jain et al., 1999) (DesJardins et al., 2006) and more recently synchronous linked multi visualization (Ward, 1994), (Anselin et al., 2006), (Swayne et al., 2001), (Ho, 2003).

2.1 Synchronous Linked Multi Visualization

‘Synchronous Linked Multi Visualization’ refers to the act of displaying the same data in different ways at the same time, within an environment that reflects user actions from one visualization, in all others. The purpose of this concept stems from the fact that it may not always be possible to discover everything within a dataset using a single visual style.
Traditionally data is examined using one style at a time; with the strengths and weaknesses of each being taken into account during the analysis. This does not allow for easy cross examination, where effects seen when one visualization is modified by the user may produce entirely unpredicted results on another. Updating multiple visualizations at the same time, regardless of which was actually modified allows the strengths of each visual style to be used whilst perhaps allowing workarounds for any weaknesses.

2.1.1 Visualization Styles

Visualizing data can be as diverse as the imagination allows, however certain visualization styles have come to be accepted as being more successful than others for representing specific data types.

The first visualization systems introduced the concept of an automatically generated scatter plot within the sciences. As a visualization technique it is both simple to create as well as easy to conceptualise. Generally speaking it was introduced alongside the Cartesian coordinate system although it is correct to say that the two are not intrinsically linked. It would be equally as correct to produce a scatter plot using a less common coordinate system such as the Curvilinear system (Weisstein, 2003) or Grassmann coordinates (Weisstein, 2007). Certainly the Cartesian coordinate system may not always be the most suitable, for example the Geographical coordinate system (IBM, 2007) is the most commonly used for defining a specific point on a planet’s surface.

What constitutes a scatter plot can be defined as a set of points representing two variables $X$ and $Y$, placed according to a set of appropriate axes, which are usually designed to follow the Cartesian system. More recently, the advent of virtual three dimensional environments available on computers has led to the more widespread usage of the three dimensional scatter plot. This is simply an extension of the two dimensional variant with an extra axis added along the $Z$ plane, and therefore the ability for each point to represent a third variable.

The main problem with coordinate based visual styles becomes apparent when high dimensional data sets are encountered, as to visually represent more than three
dimensions using scatter plots it is necessary to either utilise abstract visual representations, such as glyphs, or to conceptually subdivide the dataset. Once divided, each sub group can then be analysed individually or using comparative side by side techniques such as the scatter plot matrix (Hartigan, 1975b). Figure 2.1 shows a scatter plot matrix of a five dimensional dataset dealing with remote sensed data. Note the matrix layout allows for each variable to be plotted against every other variable.

![Figure 2.1 Scatter Plot Matrix revealing a remote sensor dataset containing 5 variables, example generated using 'XmdvTool' (Ward, 1994)](image)

While concepts such as the scatter plot matrix allow coordinate based methodologies to be expanded to highly dimensional data, research into whether more abstracted visualizations would provide even better insight and analysis potential has quickly surfaced. This has resulted in more geometrical representations and concepts such as Glyph encoding (Chernoff, 1973, Siegel et al., 1972), and even conceptual extensions of the basic concept of a coordinate system, such as Parallel Coordinates (Inselberg, 1985) to be developed.

Glyphs by definition are a visual abstraction of specific data values. Examples such as the Chernoff Face (Weisstein, 2007) and Star Glyphs (Siegel et al., 1972) are widely
accepted to have merit as standalone visualization techniques. An example of the Chernoff Face Glyph technique can be seen in figure 2.2 followed by an example of the Star Glyph technique in figure 2.3 Both examples were generated using the first three values from the first three rows of data from the Lumbar Anterior Root Stimulation Dataset.

The Chernoff Face is an unusual yet largely successful take on the Glyph concept. Chernoff’s argument was that the human brain is especially good at recognising facial patterns. Therefore by mapping the values from a dataset to control the characteristics of a face, such as the arch of the eyebrows or shape and size of the eyes, when placed side by side any ‘odd’ faces should become apparent. A frowning unhappy face amidst a collection of happy smiling faces could indicate an anomaly in the data for example.
The Star Glyph seen in figure 2.3 is described in more technical detail in section 3; however the basic premise is similar to that of the Chernoff face with the main difference being that each value is mapped to a specific line length. Each set of lines are placed so as to emanate from the same centroid and then pairs of lines are joined up to generate a triangular mesh, resulting in a complete geometric shape. This method also relies heavily on pattern recognition so placing them all side by side in a grid layout is considered to be the easiest way to allow for direct comparison. ‘XmdvTool’ does allow for placement of each Star according to a set of coordinates determined by two available variables, but its default and most commonly used layout is a uniform grid as shown in figure 2.3.

If a Glyph is to be used in place of a single point in a scatter plot it will generally be kept very simple, perhaps allowing up to a maximum of four variables to be encoded. Through this it is possible that a single two dimensional scatter plot could not only show $x$ and $y$ variables but also a reasonable number of extra variables at each point.
Parallel Coordinates were introduced by Alfred Inselberg (Inselberg, 1980) and provide a method which can represent $n$-dimensional data using a polygonal line based concept. If, for example, we were attempting to visualize a dataset with five dimensions, we can specify that the first two rows of the dataset are defined as $A$ and $B$ and each discrete value defined as $x_n$... the two resultant sets would be as follows:

\[ A = \{a_1, a_2, a_3, a_4, a_5\} \quad B = \{b_1, b_2, b_3, b_4, b_5\} \]

A set of five axes are placed vertically, numbered either according to the normalised maximum and minimum of each data dimension, or according to the dataset as a whole. Set $A$ would then be used to place a point on each of the five axes followed by each pair of adjacent points being joined by a single line. The same is repeated for set $B$. This concept can be seen in figure 2.4 and a real world example is shown in figure 2.5.
Parallel coordinates allow correlations and distinctive patterns throughout a dataset to be visually represented. For example if a dataset has two dimensions and the resulting parallel coordinate lines formed an ‘X’ shape. We would know that when the first columns data was high, the second was low and conversely when the first was low the second was high. Normally however datasets with more than two rows are used and this eventually leads to clutter. Even the most seemingly cluttered and dense dataset can reveal patterns and correlations when viewed using Parallel Coordinates as long as the viewer is aware of the geometric shapes that they must look for. The inclusion of other techniques such as brushing can also be used to reveal patterns hidden within the clutter.

An important point to note whilst developing a visualization and appropriate toolkit based around Parallel Coordinates is that of its intended purpose. Since Inselberg’s original conception there have been numerous publications (Inselberg, 1984), (Inselberg, 1985), (Inselberg, 2004), (Hung and Inselberg, 2007) outlining the geometric patterns and properties that can be found within Parallel Coordinates and how to interpret them. To be truly useful these techniques rely on each data axis to be normalised so that uneven data columns appear even. However if the goal is to provide a single representation of a whole dataset, in order to show the exact positions of a value.
in comparison to another, then it might in fact be prudent to generate axes which reflect the upper and lower limits of the whole dataset. This will ensure that a point on one axis is directly equivalent to the same point on another axis, allowing for direct comparison between them, this however is likely to affect the usefulness of the geometric properties inherent to the visualization style.

2.1.2 Synchronous Linked Visualization

Practically speaking there are implementation considerations that must be taken into account when considering presenting multiple individual visualizations of the same data at the same time. Some existing tools such as ‘GeoDa’ (Anselin et al., 2006) present the user with an MDI (Multiple Document Interface). This allows a single container window to provide the ability to load and perform global operations on the dataset, while separate child windows can be generated that exist individually within the scope of the main container. MDI applications are easily supported by the majority of high level programming environments and by most popular operating systems.

An alternative to the MDI solution is that of a pre segmented (or pre defined) presentation area. An example of this can be seen in ‘Mirage’ (Ho, 2003). This approach predefines where each visualization will appear, usually the ability to alter the size of each visualizations viewing area is present, but not their position.

The third solution, seen most clearly in ‘XmdvTool’ (Ward, 1994), provides the ability to load a single dataset and switch between different visualization styles but not view more than one at once. Interaction that affects the data, such as brushing, is applied throughout. This approach is simpler from a developmental point of view but removes an important element of synchronous linked visualization, that of being able to view the effects of interaction with one visual style in all others instantaneously.

A representation of all three described environmental layouts can be seen in figure 2.6, depicted as they would appear within a generic windowing based graphical user environment.
It is also important that the visualizations chosen will work together harmoniously. Certain visual styles complement each other while some may have the effect of complicating what appears obvious elsewhere.

Many solutions offer a wide variety of visualizations all in the same package; however it is important to note that often they do not allow for cross interaction between the visualizations. If the only aim is to provide standalone visualizations that are not linked, then the choice of visual styles becomes less important. If the aim is to produce a linked system, then it is imperative that the visual styles chosen all complement each other.

Very few visualization tools offer truly synchronous cross visualization interaction; there are a few notable examples. Specifically ‘Ggobi’ (Swayne et al., 2001) and ‘Mondrian’ (Theus, 2002) both automatically update brushed data on any other visualizations currently being viewed. ‘XmdvTool’ (Ward, 1994) and ‘Mirage’ both support the concept of reflecting changes from visualization in all others, but ‘XmdvTool’ is unable to do this synchronously as it only allows one visualization to be viewed at once and ‘Mirage’ requires the user to manually request an update be broadcast to every other visualization.

Figure 2.6 The three main presentational methods used for Synchronous Linked Multi Visualization
Figure 2.7 shows two of the data variables from a sample dataset being visualized as a two dimensional scatter plot on the right and all five of the datasets variables being shown on the left as a Parallel Coordinates visualization. In the figure a rectangular brush has been used on the Parallel Coordinates to define specific values, these values are then automatically highlighted on the scatter plot without any further interaction from the user. The same concept can be viewed in ‘Mondrian.’

2.2 Data Brushing

In its simplest form the term ‘Brushing’ refers to interactively selecting a specific subset of data, followed by visually differentiating the resulting subset from the remaining non-brushed data. The most commonly cited example of the first use of this technique is the visualization system ‘Prim-9’ (Fisherkeller et al., 1974). Prim-9 allowed up to nine data dimensions to be visualized. In the words of Fisherkeller et al. “PRIM-9 is an interactive computer graphics program for Picturing, Rotation, Isolation, and Masking—in up to 9 dimensions.” The ‘Isolation’ and ‘Masking’ referred to is what we now define as Brushing.

The basic concept of brushing is so simple yet effective that it has hardly needed to be changed in over 30 years; however, minor modifications to the concept have been added over the decades as computer graphics and computational power have improved. Thus modern visualization systems that employ brushing allow different brush ‘styles’,
referring to the geometric shape of the brushing area being controlled by the user. Some also allow multiple brushed groups to be defined, either by distinguishing the groups using different plot shapes or by using different colours.

Some systems have expanded the basic purpose of brushing so that it no longer simply aims to select data interactively, this is especially apparent in environments which utilise abstract visualization styles such as Parallel Coordinates. Often these systems allow the user to define a geometric shape or pattern that fits within the context of the visual style itself; any actual data that then meets those exacting specifications is brushed. This essentially extends the concept of the brush so that it mimics the attributes of the visual style itself.

### 2.2.1 Brush Parameters

As reliance on computerised information visualization has increased, so has the demand for control over advanced functionality such as brushing. Hovering the mouse cursor over a data point in a scatter plot and clicking to select it is no longer enough. Luckily modern computers are fast enough to allow more complex operations and brushing methods without impeding the overall usability of a software environment.

To this end, brushing systems now tend to incorporate multiple, user interactive, geometric shapes to facilitate data selection. These shapes often have definable properties, such as how much of the data that falls within their boundaries should be brushed or perhaps what colour or glyph should be used to represent the data that is selected. The shapes that are available are usually chosen as a direct result of the visualization styles offered by the system in question.

Perhaps the most obvious form of brush parameter is the ability to define the Boolean state of the brush, i.e. whether it is additive, subtractive or neither. A complex yet complete implementation of this concept is provided by XmdvTool and can be seen in figure 2.8.
Chapter 2. Prior Research

The application offers the ability to define up to four individual brushes, each with a separate set of interaction rules. This implementation is clearly geared towards the Computer Scientist or Mathematician, as the basic concept of Boolean logic is a prerequisite to being able to effectively use the functionality offered. An informed user should be able to produce a very complex brushing effect with one or more brushes by making some additive and some combinative.

Alternative implementations of this concept have used less complex yet arguably still as effective methods to provide this functionality. ‘Ggobi’ (Swayne, 2001) uses the simple method of detecting a specific key press by the user, resulting in the current brush being permuted to the opposite of its current action. Therefore if the designated button is not pressed the brush is in its default additive state, whereas if the designated button is activated the brush alters to a subtractive state. This methodology closely follows that of most popular drawing packages.

More recently the brush has been extended from simply selecting data to a method of user driven clutter reduction (Ellis and Dix, 2006). Hence the user controlled brushing area is defined using the metaphor of a lens; this has controllable parameters which allow the level of visible density below the lens to be defined. The article submitted by Ellis and Dix promotes random data selection in order to reduce clutter, stating that it is computationally less expensive and therefore easier to apply in a real time interactive environment, while at the same time successfully providing a reasonable degree of accuracy in data removal compared to non random algorithms.

Perhaps the most interesting element of the work performed by Ellis and Dix is that it easily transfers back to brushing in the more traditional sense. The lens was used to automatically select and entirely remove data wherever it was desired, but there is no
reason this concept could not be broken into its separate components and each be provided individually. Certainly the ability to take a random sample through a set of data being brushed could provide a method by which a randomly selected subset is defined, thus allowing the user to then apply any other tool available to them on that subset alone. This could be especially useful in a cluttered dataset where the desirable and undesirable values are unknown or indistinguishable from each other.

### 2.2.2 Brush Shapes

The term brushing is perhaps slightly emotive in that it conjures an image of a paint brush being used to daub specific data points with a colour. Due to this metaphor it is also generally expected that the brush will be available in different shapes and sizes, as would be the case with a physical brush.

The most common default brush style is that of the deformable rectangle. A rectangular selection area emanates from the users mouse click position with its opposite corner being placed at the current mouse position. Four examples of this type of brush shape can be seen in figure 2.9, taken from various software tools.

![Rectangular brushing](image1)

*Figure 2.9 Examples of deformable rectangular brushing within information visualization packages*
Often the rectangular brush is the only option offered, which is most likely due to the fact it is reasonably easy to implement programmatically as well as providing most of the functionality the average user expects or arguably even requires.

While the rectangle is well suited to accurate selection within a Cartesian based scatter plot, it is not always as applicable within alternative visualization styles, or even for every selection case within a scatter plot. Alternatives are therefore offered within some environments. Notably ‘Mirage’ provides the ability to place \( n \) circular ‘paint daubs’ by clicking at any point on the screen and moving the mouse whilst the button remains pressed. The size of each daub can be increased or decreased before it is placed; and all points that lie below each daub are brushed. This has the effect of offering the user the ability to paint over an area they are interested in.

![Figure 2.10 'Paint Daub' brush shape, example generated using 'Mirage' (Ho, 2003)](image)

Although slightly cumbersome, by setting the brush to be the desired size then placing a single daub in the right position, ‘Mirage’ provides the ability to brush an area of points that lie below a circular selection. This could prove useful in situations where a circular radius around a specific point or perhaps a circular area that encompasses a cluster of points needs to be defined.

When the two dimensional scatter plot is moved into three dimensions a definite problem with regards to the usability aspect of brushing occurs. Defining what should be brushed interactively is an easy task in two dimensions, wherever the mouse cursor...
currently lies is the current point of interest. Then whatever shape to be drawn can be given a definite point from which to emanate, begin or terminate. In three dimensions it is far more difficult to define the exact point of interest due to the depth of field evident in pseudo three dimensional environments.

The only feasible ways to tackle this are to either give the user the ability to ‘fly’ through the three dimensional space or to provide a visual cue that is placed within the scene which the user can move through the three dimensions. A working example of this concept can be seen in ‘GeoDa’ (Anselin, 2006).

As figure 2.11 shows, the brush is a deformable cuboid, placed initially at the exact centre of the visualization. The user is then given the ability to move the brush in any direction up to the outer boundaries defined by the white lines. The user is also able to increase the cuboid in size along any of the three dimensions. As a point passes within the brushes perimeter, the point becomes brushed and is highlighted by altering its colour from white to yellow.
Most of the software environments previously discussed offer differing visualization styles, however only a small number offer brush shapes actually tailored to each style. Specific brushing modes are normally found in environments offering the Parallel Coordinates visualization. Primarily this is because the current trend is to try to exploit the inherent geometrical properties that exist within Parallel Coordinates for discovery. Rather than use the Parallel Coordinates simply as a compact and succinct way to visualize large numbers of variables side by side.

Two very different examples of specialised Parallel Coordinate brushing exist, the first is within ‘Parvis’ (Ledermann, 2003). This relies on the angular nature of Parallel Coordinates by allowing the user to define an angle which they would like to be brushed. This method of brushing has been considered in detail (Hauser et al., 2002) and certainly holds merit if the visualization is being used to analyse geometric datasets or surfaces (Hung and Inselberg, 2007) as then the angles between variables are well defined. An example of this is provided in figure 2.12.

Figure 2.12 Angular brushing within Parallel Coordinates, example generated using 'Parvis' (Ledermann, 2003)
The angle of interest is specified in the top part of figure 2.12; this is shown visually between the third and fourth variables. Any lines that meet this angular requirement are then brushed throughout the whole visualization. This method certainly has merit but it must be noted that it is difficult to use effectively when implemented interactively as per ‘Parvis’. Another interesting issue that has not been considered in past papers is that whilst using this method, if the brushing is applied whilst the axes are all scaled between the maximum and minimum of each variable, rather than the maximum and minimum of the whole dataset, then an angle between each set of variables will differ in meaning due to the effect of vertical scale.

The alternative method can be seen in ‘XmdvTool’ (Ward, 1994). This employs an equally angular brush shape but simplifies the concept so that all the user is presented with is a continuous set of coloured quadrilaterals and all lines that fall within the parameters of the shape are brushed. As previously discussed ‘XmdvTool’ allows for complex logical structures to be built up for each brush. An example of this technique in use is shown in figure 2.13, the angular brush used is also equivalent to an ‘AND’ operator.

![Figure 2.13 Parallel Coordinates angular quadrilateral brush style, example generated using 'XmdvTool' (Ward, 1994)](image-url)
Chapter 2. Prior Research

The effect we see in figure 2.13 reflects a unique attribute of this type of specialised brushing shape. Through tailoring the brush to the visualization itself we are able to quickly define the properties that we are interested in throughout the whole dataset. Only the rows of data that meet all of the defined criteria are revealed. In the example shown we have explicitly stated that we want to see all of the cars that have the following attributes:

- High MPG (Miles Per Gallon)
- Low Cylinder Count (either 3 or 4)
- Low BHP (Brake Horse Power)
- Low Weight
- Any value for acceleration
- Developed circa ~1975
- Only of European origin

This form of brushing is ideal for users who know the parameters they are looking for, however if we were performing unstructured analysis it may prove to be cumbersome. In these cases the far simpler approach of providing a rectangular brush that can be dragged over any of the axes may be more successful.

2.3 Data Binning

Binning can be considered an umbrella term, describing in very little detail the ambiguous concept of gathering data together that falls within specific boundaries.

The most commonly used binning method is the histogram, but even this is a description of a concept rather than a definite method. Generally speaking when a Histogram is presented, a simple bar chart is used. Weisstein (Weisstein, 2002) describes a histogram as:

“The grouping of data into bins (spaced apart by the so-called class interval) plotting the number of members in each bin versus the bin number”
Through this definition it is clear that although a bar chart is the most logical presentational method, alternatives such as a scatter plot of the bin number against bin count or perhaps even a pie chart could be used and would technically be as equally correct. Clearly binning should be treated as a concept rather than tied to one method, therefore by defining an upper and lower boundary and generating a representative entity for all the values within those boundaries, we are implementing a binning algorithm.

### 2.3.1 Binning Techniques

Generally speaking the ability to apply binning within the majority of multi visualization based environments is very limited. A couple of the programs mentioned thus far offer basic abilities. ‘Mondrian’ allows a bar based histogram to be automatically generated for each variable; the user is then able to define the width of each bin up to a maximum of the number of atomic values within that variable.

![Histogram example, example generated using 'Mondrian' (Theus, 2002)](image)

Figure 2.14 is a histogram of the ‘Hip’ variable from the Lumbar Anterior Root Stimulation dataset. The bins are set to a width of 2.5 which has resulted in 14
individual bins. Clearly ‘Mondrian’s’ histograms are meant purely to offer an overview of the frequency within each variable, as the amount of labelling and added information is minimal. One interesting feature that ‘Mondrian’ offers is that the user can click on a bin and all the values that it contains are brushed in all the other visualizations.

Generally speaking visual implementations of binning algorithms are usually found in subject areas that make heavy use of in-depth image analysis, specifically medical and astrophysical applications. Data that is used to form a portion of an image or rendering is by its very definition spatially linked to those around it, if we introduce the element of time then a bin placed in exactly the same area of each image could be considered to also be temporally linked.

A common use of image based binning is within the field of spectroscopic data analysis. Generally x-ray or thermal spectrums are visually mapped two dimensionally using an $x$, $y$ grid extrapolated directly from the pixel locations of the CCD (Charge-Coupled Device) or other imaging source that they were obtained from. This is then used as the basis by which the binning structure is formed. One novel approach to this problem was presented by Cappallari and Copin (Cappellari and Copin, 2003) whereby a geometric concept known as Voronoi Tessellations are used to generate a web of bins. Diehl and Statler (Diehl and Statler, 2005) advanced this methodology to specifically suit X-ray data, amongst others.

As a method, binning algorithms are also used extensively in graphical applications. Due to the very linear grid like nature of pixel based systems, binning is ideally suited to grouping together conjoined pixels to form a larger single pixel that is representative of all that it contains.

This can be found in use in devices that use a CCD such as high resolution scanners. Binning methods are often used to increase the SNR (Signal to Noise ratio) of such devices (Roper, 2007). As CCD’s are not always entirely accurate, binning introduces a level of ‘smoothing’ whereby the effect of a noisy pixel will be partially cancelled out by the other pixels that fall within its bin. Conceptually the bins themselves are very much like a cell in a grid, if the grid is segmented in such a way that there are the same number of cells as there are pixels then clearly no binning effect will occur, the less cells there are, the greater the number of pixels that will be included in each bin.
A similar method is also used in more rudimentary image and video compression algorithms, whereby the average colour and hue of a subset of pixels is calculated, resulting in a new larger representative pixel with a hue and colour that is an average of all those that it encompasses.

An interesting concept that is slowly being explored is that of incorporating binning with less usual visualization techniques such as Parallel Coordinates. Until recently binning was generally performed on a dataset and the results shown using a bar chart as seen in ‘Mondrian’ (figure 2.14). However ‘Parvis’ introduces an interesting concept of overlaying a Histogram, thereby denoting the frequency of points within a specific area of each axis on a Parallel Coordinates visualization. The size of each bin can be varied, allowing the amount of detail the Histogram provides to be altered. An example of this is shown in figure 2.15.

As figure 2.15 shows, by overlaying the Histogram concept and combining it with the Parallel Coordinates visualization, we are offered a new level of detail that could not previously be seen.
2.4 Data Clustering and Classification

The way in which clustering is used can vary largely from one application to the next, perhaps the most common usage within a computing context is as a data mining tool. Although a relatively new area of research, data mining has spawned numerous new clustering techniques.

Generally speaking, the goal of a clustering algorithm is to find one or more data subsets in which all components have a quantifiable property in common. The specific commonality being tested is where the difference between each algorithm usually lies. Although some visualization systems offer clustering, the algorithm actually being run will nearly always be the same as those run within non visualizing environments.

Many different types of clustering algorithm exist. However all generally fall into one of two categories. The algorithms are usually referred to as being either partitional or hierarchical. As the names suggest, a partitional algorithm will work by trying to segment the data into a set number of partitions, each of which encompasses a subset of data. Hierarchical algorithms differ in that they generate new clusters based on the state of previously generated clusters (MacKay, 2003). Most of the algorithms in general use are of a partitional nature.

Clustering algorithms tend to fall into one of two categories, with regards to the amount of interaction they require to be successful. The most common are known as unsupervised algorithms, so called because once initially seeded they run through to completion with no further interaction from the user. Slightly less common are supervised algorithms, as the name suggests these differ from unsupervised algorithms in that although they may still require initial seeding to begin, they also require subsequent interaction with the user in order to successfully complete. Both forms of algorithm have advantages and disadvantages. Certainly a supervised algorithm is more likely to eventually lead to a set of clusters more closely related to that desired by the users, as the users themselves will have made the fundamental decisions the eventual conclusion is formed from (Griera et al., 2004).

It is also important to note that there exists a distinction between different types of algorithm. While the term clustering is the most commonly known there also exist
classification and associative algorithms, amongst others. As the name suggests, classification algorithms differ from clustering algorithms in that they attempt to assign a subset of data to a predefined class that best defines the quantifiable property that they are being measured on. Within data mining the two types of algorithms are used for different purposes, one tried to define clusters that actually exist in the data whilst the other attempts to fit the data to a premeditated model.

### 2.4.1 Clustering Algorithms

While it would be entirely impractical to note every single clustering algorithm in general use today, it is correct to highlight a select few that are almost definitive in nature and will be found in many software environments offering clustering as an option.

The first is ‘K-means’ which was proposed by McQueen (McQueen, 1967) and refined by Hartigan (Hartigan, 1975a). Although very simplistic in nature, it is arguably the base concept later algorithms have been formed around. Roughly speaking, if clustering is offered to a user, it is almost inevitable that the default algorithm will be K-means. We can describe K-means mathematically as an algorithm for clustering \( N \) data values into \( K \) subsets \( S_j \) which each contain \( N_j \) data values.

\[
C = \sum_{j=1}^{K} \sum_{n \in S_j} |X_n - \mu_j|^2
\]

where \( X_n \) is the current data point and \( \mu_j \) is the geometric centroid of all current data points in \( S_j \) (Weisstein, 2005).

Conceptually K-means is an unsupervised partitional algorithm which aims to define a specified number of clusters throughout a whole dataset. The algorithm begins by placing as many centroid’s as there are desired clusters, at random points within the confines of the dataset. The initial centroid’s are placed as far away from each other as is possible. A distance measure, usually Euclidean, is then used to collect the nearest points into the cluster. A new collative centroid for the entire cluster is then calculated.

\[\text{Multi View Graphing}\]
and this new centroid is used in the next iteration of the algorithm. Termination usually occurs either after a set number of iterations or when no change in the clustering of the data occurs (Matteucci, 2007).

The second is that of hierarchical clustering. Generally speaking hierarchical algorithms tend to take one of two approaches, heuristic or probabilistic. Most heuristic methods follow the methodology of the Self Organising Map (SOM) (Kohonen, 2000) which is a form of neural network. Heuristic methods based around a SOM are unsupervised in nature and the results generally well suited to visualization. Probabilistic methods are perhaps less widely used but are most suited to datasets that contain a high degree of unknown elements or uncertainty. Most probabilistic implementations are semi-supervised in their nature (Vicente and Vellido, 2004).

### 2.4.3 Presentational Methods

The presentation of clustering algorithms within visualization tools is generally very simplistic, entirely ignoring the visualizations themselves until the user requests the clustered data be shown. Certainly interaction with the visualization plays little to no part of the clustering process.

Generally most software environments that deal with clustering come from the subject area of data mining. Perhaps the most complete collection of clustering and classification algorithms in a single environment can be seen in ‘Weka’ (Witten and Frank, 2005). The goal of ‘Weka’ is to provide a homogenised environment in which many different algorithms can be run on the same dataset and the results viewed either numerically, statistically or visually. An example of the basic environment offered by ‘Weka’ can be seen in figure 2.16.
As figure 2.16 shows, the output from ‘Weka’ is predominantly text based. The user selects an algorithm from a list, defines the initial properties and then runs it until completion. They are then able to view the output metrics via a text pane, the results can then be visualized. The options available for this are limited to a two dimensional scatter plot or where appropriate, a hierarchical tree view. An example of the scatter plot available within ‘Weka’ can be seen in figure 2.17.
Although functional, figure 2.17 makes it is clear that the visualization within ‘Weka’ is limited, allowing for only simple visual styles, with almost no user interactivity.

There exist multiple examples of clustering or classification algorithms being offered as functions that can be applied to a dataset. The types of application can differ greatly, ranging from database server software through to visualization systems such as ‘Mirage’ (Ho, 2003), which offers the ability to run either a K-means or Hierarchical clustering algorithm. It is true to say though that very little in depth integration within visualization environments currently exists and application of the clustering via visual means is even less common.
Chapter 3

3.0 ‘Multi View Graphing’

To provide a demonstration of the techniques described throughout the rest of this chapter, an application, known as ‘Multi View Graphing’ (MVG), has been developed as a proof of concept.

To ensure a reasonably quick development period, MVG was developed using C#, and the OpenGL graphics API. Whilst perhaps not the most obvious choice for a graphics based project, C# provides a simpler development model when compared to alternatives such as C++, which leads to faster development. The use of OpenGL within a .NET application has been facilitated using the Tao Framework (Loach, 2007), which provides access to the standard Win32 OpenGL libraries within a .NET environment.

Reading of the CSV file format is facilitated using a publically available library, the ‘LumenWorks’ framework (Lorion, 2007). While rapid mapping of the internal data structure to a .NET Data Grid object was accomplished using the ‘MommoData’ public library (Stefanov, 2004) (for further details regarding the Data Grid, please refer to Appendix A).

3.0.1 The Visualization Pipeline

As with most scientific processes, visualization often follows a set pattern. To date one of the best flow based description of this process was published by Haber and McNabb (Haber and McNabb, 1990). Their reference model referred to as the ‘Haber-McNabb Model’ encapsulates the key stages the majority of visualization systems adhere to.

With this in mind, the structure of MVG has been designed so that it follows the model as closely as possible in the hope that the eventual prototype will be as accomplished as prior systems developed around the model. This structure is shown in figure 3.1.
The general structure of MVG maps onto the Haber-McNabb model reasonably well. Each stage on the left of the figure has an equal representative stage within MVG, detailed on the right. For a more detailed explanation of stage four and the usage of OpenGL geometry lists within MVG, please refer to Appendix B.

3.1 Synchronous Linked Multiple Visualization

As discussed in section 2.1.2 there are three layouts in general use for presenting multiple visualizations. MVG makes use of the MDI layout as this not only allows as many visualizations as the current screen resolution can handle, but also allows for each individual visualization to be resized or disabled as desired, thus maximising the usefulness of the screen space.

Certain limitations imposed by the object orientated manner in which the Tao Framework presents the traditional OpenGL context, means that having multiple instances of the same visualization is currently difficult, thus only a single copy of each visualization can be viewed at once within MVG. This is not considered a major
problem within an environment intended purely to present the concepts described in the rest of this section, it would however be an issue that should be urgently addressed were MVG developed beyond the experimental stage.

### 3.1.1 Sharing the Data

From a computational point of view, if the same data is to be utilised by multiple child windows within an MDI environment, then it must be contained within a thread safe data structure. The main parent window (or MDI container) provides the interface for loading the data file and also generating new instances of each child window. This means all data should be held in one public structure within the parent windows class. Although MVG is not currently optimised for multi threaded operation, sharing a single data repository with multiple child windows can present very similar problems seen in highly parallelised software, it is therefore also prudent to make use of thread safe system classes wherever possible.

In the case of MVG all data is held in a two dimensional array consisting of a combination of Array Lists and List Dictionaries. Array Lists are slower to access and utilise than a non class based array structure and in a graphics based application this would normally be unacceptable. However they are thread safe by design thus multiple child objects can access data held in an Array List simultaneously. List Dictionaries are used as they are a very efficient and fast method of non serial data storage and retrieval, albeit with a relatively large memory overhead.

Using these data structures, it is possible to recreate only one copy of the original data file in main memory, yet still serve a theoretically unlimited number of child objects at the same time.

### 3.1.2 Visualization Styles

The choice of visualization styles to present to the user can often be one of the most challenging design aspects to be taken into account in an environment that allows for
cross visualization interaction. Specifically this choice is important as some visualization styles are simply not suited to certain data exploration tasks. It would not be correct to provide pie chart functionality within a system designed to scientifically explore large multidimensional datasets for example. The pie chart can be a good way to provide a visually attractive and easily understood overview within an environment where the detail within the data is unimportant, but for good scientific analysis, it is inappropriate.

Within MVG the purpose was always to demonstrate concepts and not to specifically deal with a certain type of data, this very quickly narrows the choice of visualization styles down to some of the general techniques in common use, such as the two dimensional scatter plot. It is important to note however that just because only a small number of techniques are shown within MVG, this does not mean the core concept of synchronous linked multi visualization does not allow for expansion. In fact the success of the inter-visualization interaction within MVG demonstrates that if a non specific tool were being developed, it would most likely be preferable to provide as many visualization styles as possible to increase the overall usefulness of the environment.

Complex three dimensional visualizations are not evident within MVG; this is not to say more complex abstracted three dimensional styles or more conventional volume rendering methods such as Iso Surfacing, should not be included in a system which wishes to visually link one visual style with another via interaction; only that in the case of this study, the presented designs were more appropriate for demonstration of the concepts.

MVG employs the following visualization styles:

- Two Dimensional Scatter Plot
- Three Dimensional Scatter Plot
- Parallel Coordinates
- Star Glyph
- Data Grid

All of these styles have levels of user interactivity. It is possible to translate and scale (or zoom) each of them, while the three dimensional scatter plot also provides the ability to for rotation.
The two dimensional scatter plot is possibly the most common visualization style used during scientific analysis, for statistical work this accolade may fall to the bar chart, but if the goal is to show correlations and clustered areas for two data dimensions then simply placing points within a set of coordinates is generally very successful.

To generate the two dimensional scatter plot within MVG two data subsets $X$ and $Y$ from within the whole dataset $D$ need first be defined, resulting in the following:

$$X \subset D$$

$$Y \subset D$$

To produce a set of appropriate Cartesian coordinates it is then necessary to find the upper and lower boundaries of the two new data subsets, resulting in four new variables ($x_{\text{Maximum}}, x_{\text{Minimum}}, y_{\text{Maximum}}, y_{\text{Minimum}}$). This information is then used to generate the eventual visualization as shown in figure 3.2.

In addition to the basic elements of the scatter plot defined in figure 3.2, MVG also applies visual elements such as a background grid; an example of the actual visualization presented in MVG can be seen in figure 3.3.
The next visualization style offered is the three dimensional scatter plot. This is a natural progression of the two dimensional scatter plot seen in figure 3.3. The process of producing a three dimensional scatter plot within MVG is almost identical to that of the two dimensional scatter plot, with the addition of a third data subset \( Z \), of which the maximum and minimum values are used to produce a third axis. An example of the visualization presented by MVG can be seen in figure 3.4.
Multi View Graphing

Chapter 3. Multi View Graphing

One very important aspect to MVG was the ability to view highly dimensional datasets; both the two and three dimensional scatter plots were very limited in this respect. Parallel Coordinates were therefore added as the third visualization style, whose highly dimensional nature complements the accuracy but low dimensionality of the scatter plots.

Parallel coordinates allow for a theoretically unlimited number of data dimensions to be represented on the same visualization, whilst maintaining a two dimensional aspect thus reducing apparent complexity. The power of Parallel Coordinates lies in the ability to view how one variable correlates with another.

In order to generate the Parallel Coordinates visualization within MVG the upper and lower boundaries for the whole dataset are found (maxValue, minValue). These values are then used to draw equidistant vertical axes. The whole dataset is then drawn one row at a time, thus if the dataset contained 5 dimensions (or columns) and 200 rows of data, the resulting visualization would be made up of 5 vertical axes and 1000 individual line
segments spanning between the axes (200 between each pair). This concept can be seen diagrammatically in figure 3.5.

![Diagram of Parallel Coordinates visualization](image)

**Figure 3.5** Visual representation of the generation of a Parallel Coordinates visualization

As with both scatter plot styles, there are other elements added to the Parallel Coordinates within MVG, including labelling and a simple grid for numerical reference. There also exist a few options to alter the appearance, including the ability to apply a colour graduation spanning through the RGB scale, starting with the first data row as red and progressing to the final being blue. It is also possible to colour each pair of lines between a set of variables a single colour or for each complete set of lines to be given a unique colour. The ability to graduate how transparent each line is, where the last lines drawn are barely visible while the first is entirely opaque, is also evident.

The ability to zoom into any visualization is provided and has proven to be very useful in obtaining a better perspective on cluttered areas. An example of MVG’s Parallel Coordinate visualization can be seen in figure 3.6. The figure gives a labelled overview of some of the general options available within the visualization that are considered to be of importance.
Chapter 3. Multi View Graphing

Multi View Graphing

The default view of the Parallel Coordinates visualization within MVG

Table of Contents

Finally it was felt that to provide a well balanced spread of visualization techniques, a form of Glyph representation should be included. Star Glyphs have the advantage of being a simple concept whilst providing the possibility of interesting visual insight into certain datasets. They were therefore chosen for inclusion into MVG.

Figure 3.6 Examples of the Parallel Coordinates visualization generated by MVG including general display options; example generated using the Lumbar Anterior Root Dataset

Multi View Graphing
The implementation within MVG follows closely that of ‘XmdvTool’, but uses the increased power provided by OpenGL to add a colour fill to each star for identification purposes. Functionality such as the ability to zoom and translate is also important.

Some variants of Star Glyph visualizations use two data dimensions to place each glyph on a set of coordinates and the remaining dimensions to generate each glyph. MVG however places each glyph adjacent to the previous, starting in the top left corner of the screen and moving to the right. As pixel values can only be whole integers, this technique theoretically allows a maximum of 360 data dimensions to be shown on each glyph, this would however simply be a solid shape with no space between each line.

To produce the Star Glyph visualization within MVG the visual space is first segmented into a uniform ‘grid’. The upper and lower boundaries of the entire dataset are defined (maxValue, minValue). Each row of data is then used to produce a single Star Glyph.

It is necessary to produce a representative value for the length of each of the Glyphs lines. This is done by first defining a scaling value (ScalingFactor) applicable to the current dataset:

\[
\text{sourceRange} = (\text{maxValue} - \text{minValue})
\]
\[
\text{scalingFactor} = \frac{\text{[Maximum radius of circle within current grid cell]}}{\text{sourceRange}}
\]
\[
\text{negativeAdd} = \text{if (minValue < 0) } \{\text{minValue - SourceRange}\} \text{ else } \{0\}
\]

The scaling value is then used to generate lines emanating from the centroid of each glyph, of a length representative of each value. It is necessary to ensure none of the length values are negative as this would result in them not being rendered, whilst in reality it is entirely correct for the lowest negative value to actually be represented by the smallest, yet still visible line. Therefore an additive value (NegativeAdd) is applied to each line length before it is used to generate a Glyph.
The parameters $L_1$, $L_2$ and $L_{n..}$ that are defined within figure 3.7 can be described as follows.

\[
L_1 = ([\text{Row 1, Column 1}] + \text{negativeAdd}) \times \text{scalingFactor}
\]
\[
L_2 = ([\text{Row 1, Column 2}] + \text{negativeAdd}) \times \text{scalingFactor}
\]
\[
L_{n..} = ([\text{Row n, Column n}] + \text{negativeAdd} \times \text{scalingFactor}
\]

The resultant visualization within MVG is very closely linked to the concept shown in figure 3.7 with the only major difference being the addition of colour. An example of the actual visualization produced by MVG can be seen in figure 3.8 with a single glyph picked out and zoomed to show its detail.


3.1.3 Cross Visualization Interaction

Without cross interaction MVG could not be considered truly ‘simultaneous’ in its nature, thus actions that occur in one visualization are reflected in all others, whether they are currently being viewed or not.

Some aspects of MVG’s functionality cannot be reproduced in all visual styles. For example the data produced by binning two dimensions in a two dimensional scatter plot cannot then be shown on a three dimensional scatter plot as the new binned data was originally made up from only two subsets of the main dataset. Hence there would be no logical method to extrapolate a third z value for each bin that would form a correct representation. This lack of interoperability between different visual styles is one aspect highlighted during the development of the MVG application, core differences in how one visual style is produced compared to another means by definition some actions taken cannot be reflected in the others.

Figure 3.8 Example of the Star Glyph visualization generated by MVG, example created using the Lumbar Anterior Root Dataset
The only truly universal action of all those provided within MVG, is that of brushing or data selection. This is due to the intrinsic link of the main dataset that exists between all of the different visualizations.

### 3.2 Data Brushing

Data brushing, or the interactive selection of specific data, is available within all visualization styles in MVG. Brushing is also primarily the means by which inter-visualization interaction is achieved.

The brushing facilities aim to offer the ability to select very specific areas of data using only visual cues within the visualizations themselves. If data is being viewed using a visual interpretation, it logically follows that the best way to interact with that data is most likely going to be through the visualization itself.

MVG uses multiple techniques to provide the ability to select data with relative ease. Different brush styles are provided which are designed to be applicable for both more usual designs, such as the two dimensional scatter chart and also less usual designs such as Parallel Coordinates. The ability to modify the threshold of the brushed area is evident also. Perhaps most importantly, user inputted brushing groups can be defined. This allows different colours to be used to differentiate between different areas of brushed data. As an extension to the brush group concept, it is also possible to modify the currently visible data according to selection.

An interesting problem that presented itself was how to visually depict a single brushed variable. This issue was evident in all visualization styles as all can display more than one variable at a time. The two and three dimensional scatter plots offered their own neat solution, as both utilise a simple cross formation to represent the points, each require the same number of variables for their formation as their point representations have individual components. It was a logical extension therefore to represent semi-brushed points by only applying the brush colour to the applicable component. Thus if we were generating a two dimensional scatter plot and a point was constructed from two variables (X1, Y1) and only Y1 was found to be brushed, the X component of the point (the horizontal line) would be left the default non-brushed colour, while the Y
component (the vertical line) would be coloured accordingly. This is extended into the third dimension using the additional horizontal line running along the \( z \) axis that makes up each point within the three dimensional scatter plot.

When dealing with Parallel Coordinates or Star Glyphs it could be considered logically correct to say that if all of the values that make up a single line or Glyph are not brushed, then that whole representation is itself not brushed. However this means that brush actions taken on two or three variables in the scatter plots would not be reflected in the remaining two visualizations until every other was also brushed. To counter this, if a whole data row is not brushed then during the creation of either a Parallel Coordinates line or Star Glyph, that individual component is coloured accordingly. An example of each of these solutions can be seen in figure 3.9.

![Figure 3.9 Illustrated examples of the solutions to the problem of semi brushed data points](image-url)

**Figure 3.9 Illustrated examples of the solutions to the problem of semi brushed data points**
3.2.1 Brush Shapes

One major element of brushing data is that of the actual shape of the ‘brush’ itself. Current environments provide predefined brushing shapes which have been selected so they fit the geometric properties of the current visualization being viewed.

The implementation within MVG takes a different approach to this concept by initially providing a very simple rectangular selection area, which is one of the most familiar methods of highlighting areas within a graphical user environment. It is then possible to alter the shape of the brush to form less uniform selection areas if a simple deformable rectangle isn’t sufficient.

The brush shapes offered vary depending upon which visualization is being viewed; some shapes simply were not feasible or indeed useful on certain visualizations. Most notably the Star Glyph visualization only has the simple rectangular brush available. Due to its entirely uniform grid based layout, no other geometric shape offered better selection; similarly the circular brush can only be found in the two dimensional scatter plot, a circular selection area does not lend itself to the staunch angular nature of Parallel Coordinates. The three dimensional scatter plot utilises a different brush style, two dimensional techniques translate into three dimensions very poorly so a three dimensional variant of the rectangular brush was included which is controlled via user interaction with the scene.

All of the brushing techniques within MVG can be considered to be of order $n$; where $n$ is the number of rows that exist in the currently open dataset. All brushing is also additive; reselection of a previously brushed area will result in a new brushing action taking place, this is representative of a Boolean ‘OR’ operation.

The rectangular brush provided within MVG is the first that is selected whenever a visualization is activated. It operates by simply comparing the $x$ and $y$ coordinates of each point to the threshold values defined by the $x$ and $y$ coordinates of each corner of the rectangle. The points which fall within all four coordinates are brushed. To visually accentuate the area that is going to be brushed, the rectangle is provided as a solid yet semi transparent area rather than simply an outline, an example of this is shown in...
figure 3.10 where an arbitrary rectangular selection area has been used to brush a subset of data points.

The first alternative to the basic deformable rectangle was to add a circular brush. A circular selection area was a natural progression from a rectangle; it provides a well defined, flexible alternative to the rectangular brush shape but can be difficult to define definite group edges with and also be slightly more involved to use effectively.

Two alternative methods of presenting this to the user were considered. First the method that is most commonly used was tested. This involved generating a circular outline, emanating from the point initially clicked on the screen and extending to the diameter defined by the current position of the cursor. This method can be viewed in most drawing packages. Eventually it was decided that although this provided the ability to generate an elliptical selection area as well as circular, it also introduced a usability problem in predicting whether the circle would encompass all of the points originally desired according to the starting position.

To counteract this, a slightly less conventional method was defined. This involves clicking a specific point to form a small uniform circular selection area and by scrolling with the mouse wheel it is then possible to increase or decrease the diameter of the circle to the desired size. Changes are reflected visually to the user and the circular brush can also be translated around the screen. This technique does mean that the ability to produce an elliptical brushing area is compromised, but it is far easier to actually see what is going to be brushed before it happens due to the visual cue provided by the circular representation onscreen.
Technically, whether points fall within the circle or not is tested using a simple calculation based on a slightly modified variant of the equation of a circle:

\[(nx - \text{point } X)^2 + (ny - \text{point } Y)^2 = \text{radius}^2\]

Using this equation, we know that if both sides balance then the point being tested in fact lies on the circle, while if the left hand side of the equation is less than the right then the point is inside the circle and conversely if the right hand side is less than the left then the point lies outside the circle. This methodology avoids the need for more complex calculations such as that of Euclidean distance, which although just as correct would introduce the square root function and thus increase the computational overhead. This concept is illustrated in figure 3.11.

Within MVG, the circle is not simply an outline; rather it is a filled area employing a level of transparency to ensure the points below it are still visible. As is shown in figure 3.12, the persistent visual cue provided by the circular brushing offers a quick yet accurate insight into the current parameters of the brush.
Often when a selection is being made, it is desirable to have the ability to generate a non constrained area, both the rectangular and circular brush styles can be modified in terms of their size and position but not their actual shape. Multiple possibilities were considered on how to provide this functionality; initially an entirely non defined ‘lasso’ selection technique was explored.

The lasso concept is simple in operation but reasonably complex in development. It allows the selection of any area of a visualization by drawing a line directly below the current mouse position until the user has generated a wholly connected shape. This requires that each mouse coordinate for each movement be stored and when the mouse is released a check is needed to see if any point intersects another. Presuming intersection has occurred, the user drawn line is then culled beyond the first intersection and a resulting shape emerges. A highly recursive check is then required to see if any points fall within the bounds of this new, non linear shape.

This technique was eventually dismissed as although the ability to draw a totally unique shape was considered useful, it was also found to be a relatively difficult task to make good use of the lasso, requiring very accurate mouse control. A practical example of this technique can be found in ‘Mirage’ (Ho, 2003). Also due to the inability to utilise far more efficient geometric checks to determine whether a point lies within the selection area, rather having to rely on recursion, this technique was also intolerably slow on very large datasets. Finally, whilst the lasso brush style made sense within the two dimensional scatter plot it did not translate well to the very angular environment.
provided by the Parallel Coordinates. A compromise between total freedom and predefined brush shapes was therefore found.

OpenGL consists of a small number of basic primitives; which can be used to generate more complex objects, the circular brush is actually formed from a large number of two dimensional triangles for example. Two such primitives are the triangle and the quadrilateral, by providing three or four corner vertices, OpenGL renders a shape accordingly by connecting them with a straight line. Through providing the ability for the user to define these corner vertices, it is possible for them to draw either a triangle or quadrilateral of any shape or size, thus the final two brushing shapes are a triangle and a quadrilateral.

The concept used in both of these brushes is essentially the same, the three or four corner vertices are placed by clicking anywhere on the visualization, with each of these highlighted via a visual cue. Once the final corner has been placed the resultant shape is drawn and then a final confirmation click causes the points beneath to be brushed. This method is not only simple and quick in use, but also provides the ability to generate a large variation of brushing areas. As the general shape of the eventual brushed area will always be known it is then possible to use a refined algorithm based on the three or four vertex values to determine which points should be brushed rather than relying on high levels of recursion.

The geometric maths used in MVG to quickly and inexpensively determine whether a point falls within the defined triangular or quadrilateral brush area, is an adaptation of a simple algorithm (McRae, 2007) based around the concept of calculating where the current point of interest lies in comparison to the line between each pair of vertices. For the triangular brush, utilising Barycentric coordinates would also have been an option, however there was no real computational advantage to this and it would not have translated into use within quadrilaterals as easily therefore two very similar methods would have required two separate styles of implementation. An annotated diagrammatical description of the quadrilateral brushing technique can be seen in figure 3.13, the same concepts apply to the triangular brush minus one vertex.
Chapter 3. Multi View Graphing

To Generate the Brushing Area:

\[ v_n...; (\text{mouseX}, \text{mouseY}) \]

\[ \text{Quadrilateral} = \{v1 \rightarrow v2; v2 \rightarrow v3; v3 \rightarrow v4; v4 \rightarrow v1\} \]

Testing if a Point Lies Within Quadrilateral:

\[ (p = \text{Current Point}) \]

\[ v1v2 = (pY - v1[y]) \times (v2[x] - v1[x]) - (pX - v1[x]) \times (v2[y] - v1[y]) \]

\[ v2v3 = (pY - v2[y]) \times (v3[x] - v2[x]) - (pX - v2[x]) \times (v3[y] - v2[y]) \]

\[ v3v4 = (pY - v3[y]) \times (v4[x] - v3[x]) - (pX - v3[x]) \times (v4[y] - v3[y]) \]

\[ v4v1 = (pY - v4[y]) \times (v1[x] - v4[x]) - (pX - v4[x]) \times (v1[y] - v4[y]) \]

if \((v1v2 \times v2v3 \geq 0) \land (v2v3 \times v3v4 \geq 0) \land (v3v4 \times v4v1 \geq 0)\) \{ Point is inside or on the quadrilateral \}

else \{ Point is outside the quadrilateral \}

Figure 3.13 Annotated visual representation of the Quadrilateral brushing concept within MVG

As with the rectangular and circular brush styles, once the corner vertices of either the triangular or quadrilateral brushes have been defined, the resultant visual cue showing the area to be brushed is semi transparent and accurately visually defines the area.

Figure 3.14 Quadrilateral brushing within MVG
The triangular brush is formed and calculated as in figures 3.13 and 3.14 with the only difference being the removal of the fourth vertex and the calculation steps associated with it.

Brushing within the three dimensional scatter plot is fundamentally different than for the two dimensional visual styles in that a resizable and movable cube is used to define the area to brush. Technically the concept is very similar to that of the two dimensional rectangular brush however the shape is not deformable and the number of faces is increased from one to six. It is possible to alter the volume of the cube in much the same way as the circular brush can be resized. Technically the same mathematical calculations apply as with the rectangular brush, with the only notable difference being that each point is also checked against the maximum and minimum $z$ values of the brushing cube as well as the $x$ and $y$ values.

Visually the cube is semi transparent but is outlined in solid black. Initially the brush was simply a transparent volume in three dimensional space, but it was found that detecting the edges of the shape could become difficult, with the chance of losing all three dimensional perspective; hence a solid black outline was added. The cube is placed via mouse movement, it is a permanent entity once turned on and by default follows the $x$ and $y$ translations of the mouse. It is possible to transpose the $x$ movement of the mouse for $z$ movement on screen, thus providing the ability to move the brushed area through all three dimensions using only interaction and visual feedback. Figure 3.15 shows an example of a subset of the three main subsets that form the $x$, $y$ and $z$ coordinates being brushed.

![Figure 3.15 Three Dimensional ‘Cube’ brushing within MVG](image)
3.2.2 Brushing Groups

Whilst the physical act of data selection is very important, arguably being able to individually identify different selections within the same dataset is even more important. Following from the example set by ‘GGobi’ (Swayne et al., 2001) MVG offers the ability to alter the current brushing colour to highlight different subsets of data on the same visualization.

The Brush Group concept within MVG is as unrestricted as was feasible whilst maintaining usability. It is possible to define and individually label a theoretically infinite number of groups and provide each with its own custom defined colour. The brush groups are also integrated into the simultaneous multi visualization nature of MVG, if a group is added via one visualization it becomes available for use in all the others and the user defined colour of each brushed area is also reflected across all the visualizations.

This unrestricted form of the brushing group concept technically provides the ability to define a maximum of \( n \) subsets of the original dataset, where \( n \) is equal to the total number of rows of data.

*Figure 3.16 Three individual brushing groups within MVG, example generated using the Lumbar Anterior Root Dataset*
Figure 3.16 highlights how the concept of brushing groups within a multi visualization environment can allow data to be visually split into subsets. In this case we can see that by using three different brush groups, each with its own colour, then selecting specific subsets of the ‘Knee’ variable via the Parallel Coordinates visualization, the three groups become very clearly visible on the two dimensional scatter plot.

3.2.3 Data Visibility

A natural progression of the brushing group concept is the ability to alter the currently active data displayed in each visualization. Traditionally the input dataset is modified and refined before it is visualized; removing unwanted values and known outliers or erroneous results first. MVG provides the ability to perform these tasks visually. These values can be brushed and then set to invisible, effectively temporarily excluding them from the dataset.

This ability is an important manual clutter reduction technique and provides access to perhaps the most common and natural clutter reduction method. A person looking at data will have some knowledge of its context and what certain values mean. In turn this also implies that they will know which areas of data they are currently interested in and which they wish to remove entirely from study.

The data visibility functionality can be considered to be subtractive in nature. Data that already exists cannot be added, only removed or restored. To ensure total control over all data, a single persistent option exists to alter the visibility of unbrushed data, this ensures that regardless of the current brushing state, it is always possible to entirely remove all data from the visualization.

Conceptually if we define the main dataset as \( D \) and a single brushed group as \( \overline{D} \) then when \( \overline{D} \) is visible \( \overline{D} \subset D \) whereas when \( \overline{D} \) is not visible then \( \overline{D} \not\subset D \).
Figure 3.17 shows the effect of brushing an area within a two dimensional scatter plot and then turning the visibility of that brush group off. Quite clearly the brushed points are removed from the visualization, they are however not physically removed from the dataset itself and thus their visibility can be restored at any time.

The same conceptual problem of semi brushed data also appears whilst dealing with data visibility, specifically if the variables which make up a point or Parallel Coordinate line are only partially brushed, at which point that representation should be hidden. The decision was to treat all occurrences of a brushed variable with the same premise as a Boolean OR statement. Therefore if a Parallel Coordinate line is made up from four variables (v1, v2, v3, v4) and v2 and v4 are both brushed but within different brushing groups, the whole line although made up of 3 separate colours, would be hidden if either V2 or V4 (or both) were set to invisible. The same concept applies throughout every other visualization style.

### 3.3 Data Binning

Data binning can have two distinct uses, either to statistically analyse the frequency of specific number ranges or to generate a more general overview of an initially cluttered dataset. Within MVG data binning is primarily a clutter reduction tool and it also follows the general ethos that everything should be applied visually.
In order to use binning to reduce clutter while maintaining the meaning of the original dataset, it was necessary to ensure application of the binning was within the same confines as already existed for each visualization. The scatter plots naturally fall on a two or three dimensional grid whilst the Parallel Coordinates essentially form multiple one dimensional grids running along each vertical axis. The binning within MVG makes use of these facts, allowing an appropriate grid of theoretically infinite size to be defined for each visualization style. Within the confines of each cell of the resultant grid, an action is taken on all data, resulting in a less cluttered representation of the original data.

MVG currently only allows for a uniform binning grid to be defined. The ability to form a non uniform custom grid would be of great benefit and is something to be added in the future.

### 3.3.1 Grid Based Binning

In order to maintain the structure of the visualizations used within MVG, the concept of grid based binning has been implemented. This allows the user to define between 1 and an \( \infty \) number of individual cells, in which all data is collated in one of two ways to generate a new, representative version of the original visualization. The grid is presented to the user using a vivid red outline, showing the boundaries of each cell exactly.

Due to the inherent differences between the four visualization styles available within MVG, the grid itself differs slightly in each implementation. It does not exist as an option within the Star Glyph visual style. This was a deliberate omission as the highly uniform nature of the Star Glyph layout makes collating and re-representing the original data an almost pointless task. It is true that overlaying a grid on the Star Glyphs and then generating a new, representative Star would allow a large dataset to be represented using fewer glyphs. However due to the abstract nature of data encoding and also the fact that the placement of each Star has no significance other than where it lies in the dataset, the binned visualization would reveal little of the originals meaning.
The two dimensional binning grid is defined by setting the number of $x$ and $y$ cells desired. This automatically resizes the whole grid so the entire visualization is filled. The total number of cells in the grid can be calculated as $([\text{No. } x \text{ Cells}] \times [\text{No. } y \text{ Cells}])$. The physical dimensions of each bin are defined by the current dataset.

$$binWidth = \frac{\text{xMaximum} + \text{xMinimum}}{[\text{No. } x \text{ Cells}]} \quad binHeight = \frac{\text{yMaximum} + \text{yMinimum}}{[\text{No. } y \text{ Cells}]}$$

The three dimensional binning grid is very similar in concept to the two dimensional grid with the only major difference being the addition of a $z$ component. This results in each cell being a cuboid. The mathematics for calculating the correct size for each cuboid is identical to that of the two dimensional grid only a $binDepth$ value is also generated using a user specified number of $z$ cells value. The number of cells can be calculated using $([\text{No. } x \text{ Cells}] \times [\text{No. } y \text{ Cells}] \times [\text{No. } z \text{ Cells}])$.

Finally the Parallel Coordinates binning grid differs in concept to that of the coordinate based visualizations. The user is able to set a cell count that is then used to generate a single column of cells on each vertical axis with each subsequent column containing the same number of cells. This means that the width of all the cells is the same. The total number of cells can be calculated using $([\text{No. Cells}] \times [\text{No. Variables}])$.

### 3.3.2 Arithmetic Mean Binning Method

The first binning method available within MVG is the ability to generate a single point that represents all points that fall inside each cell. Whilst it would have been possible to provide a multitude of different numerical functions to calculate this new point, the general presumption is to use the arithmetic mean (often simply referred to as the mean). This is because it is the most obvious method of generating a new representative point from a collection of others, therefore it is also the method most expected.

Traditionally methods used to overcome the problem of outliers such as the truncated mean or truncated median are unsuitable as the scope of each cell is only a percentage of the overall visualization, thus it would be incorrect for detail to be lost in every single
cell. To remove outliers or erroneous data, they should be brushed and hidden before the binning takes place.

If we define each bin as $n$ and the data subset that it contains as $n_i$, then the resultant $(x, y)$ of our average point would be calculated using a recursive algorithm which iterates for the same numbers of times as there are cells in the current binning grid.

\[ x = \frac{1}{N} \sum_{i=1}^{N} x_i \quad y = \frac{1}{N} \sum_{i=1}^{N} y_i \]

This results in a new collection of points equal to the number of cells, each with an averaged $(x, y)$ value. Also stored is the count value of how many individual data points made up the new averaged point. This value is then used to visually identify how many original data points each new averaged point is formed from.

This method of binning produces a new representative visualization, which quickly and clearly gives an overview as to where the data is most clustered, as well as providing an accurate placement of the average of all points inside each bin. An example of this form of binning within MVG can be seen in figure 3.18.

![Figure 3.18 Binning within MVG using a two dimensional scatter plot and the arithmetic mean methodology, example generated using the Anterior Lumbar Root Dataset](image)

Figure 3.18 shows a 10 by 10 binning grid using the arithmetic mean binning method, clearly from what looked to be a reasonably even spread of points we can see that in
fact there is a definite clustered area denoted by the larger circles coloured towards the red end of the spectrum.

The arithmetic mean is also available in the three dimensional scatter plot where arguably it could be even more useful. It can be difficult to get a good overview of a three dimensional scatter plot as clutter very quickly becomes an issue and careful rotation and scaling of the visualization is required to extract any meaning from the data.

Often the abstracted yet representative binned visualization within MVG can make obvious that which is difficult to see, highlighting the densest areas and providing a less cluttered average overview of the original dataset. The inclusion of colour and size is vitally important within the three dimensional environment, providing a relatively quick method to pinpoint data hotspots. It must be noted however that even a binned three dimensional scatter plot requires careful interaction on the user’s part to gain the most from the visualization.

Figure 3.19 Binning within MVG using a three dimensional scatter plot and the arithmetic mean methodology, example generated using the Anterior Lumbar Root Dataset
Figure 3.18 shows the effect of using a 10 by 10 by 10 binning grid along with the arithmetic mean binning method on a three dimensional scatter plot. Whilst it must be remembered that placement of each sphere is exact according to the average $x$, $y$ and $z$ values contained within each cell, the size is representative of how many points made up each sphere. Therefore the maximum and minimum size of sphere seen in figure 3.19 will always be present. However the size and density of the dataset in use will determine how many points they represent. Once the data has been binned, we can more easily see the densest collections of points. In this case relatively evenly spread throughout the visualization.

The arithmetic mean binning method is not available in the Parallel Coordinates visual style; due to the fact that it logically does not transfer itself without modification of the method. If a uniform bin was applied to each axis this would result in straight lines being formed, running directly from the first axis to the last, losing any pattern found in the original dataset. Due to the fact a point from a set of variables may be at the bottom on one axis and at the top on the next, the relationship between these two points would be lost. The only conceivable manner in which an arithmetic mean binning method could be transferred to Parallel Coordinates would be to generate only one single binning grid and place it on a specific axis; the whole set that made up each point along that line would then be binned. This would result in a coarser set of lines without losing the basic layout of the original data.

One very interesting aspect of this form of binning comes to light when we examine datasets containing largely integer values. While the examples so far have shown datasets containing sparsely congregated floating point values, integer values quickly highlight the problem of data overlay. For example a large dataset containing only integer values between 0 and 255 is very likely to have repeated instances of exactly the same number; there is no alternative other than simply rendering the repeat data point over the prior instance. Grid based binning offers a possible solution to this problem.

To illustrate, a small sample dataset has been generated. Every single four lettered word from the official UK scrabble dictionary (3540 words at the time of compilation) has been placed in a dataset so that each letter maps to a corresponding integer value, ‘A’ becomes 0, ‘B’ becomes 1 etc. This then results in a 4 dimensional dataset with 3540 rows, whereby each letter from each word is placed in its own column. Clearly this will
result in a large amount of data overlap. When we first visualize the first and second letters using a two dimensional scatter plot we are greeted with figure 3.20.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{scatter_plot.png}
\caption{Two Dimensional scatter plot showing the first letter compared against the second letter from a sample 'Scrabble' dataset}
\end{figure}

It quickly becomes apparent that 3540 individual data points are not visible in the figure and while the overlaying of the points does result in some apparent darkening of some points compared to others, this is certainly not sufficient to calculate where the densest accumulation of points lie.

If we now apply a 20 by 20 binning grid using the arithmetic mean methodology we are presented with figure 3.21. Although the original exact placement of each integer value is not always maintained as some fall within the same bin, we can now quite clearly see which letters most commonly occur next to each other within 4 letter words used in the UK version of Scrabble. Specifically it would appear that it is most likely that ‘A’ will follow ‘F’ and that ‘I’ will follow ‘L’, amongst others.
3.3.2 Density Map Binning Method

As it was decided that the arithmetic mean binning was the only method that should be included in MVG that would generate a representative point in each bin, along with the need for a binning method which could be carried from the scatter plot to Parallel Coordinates, a subtly different concept was conceived. Contrast visualization, whereby an uncomplicated greyscale colour range is utilised to represent the magnitude of specific values, is presented within MVG as a method to visually extrapolate the density of data onscreen.

The density map binning technique involves filling each cell entirely with a greyscale value. The actual colour values are generated according to the number of data points that are contained within each bin, with the highest number being represented by either pure white or pure black depending upon which is set to represent the high value. All values in between receive an appropriate greyscale value. As the density map binning relies on a count of how many data points lie within each bin, it was possible to extend
the technique so that it applied to the Parallel Coordinates visualization, this is achieved by overlaying the coloured cells on each axis rather than replacing the original lines.

The binning grid concept is used as the basis for a new density map method. The similarities between the concept presented in MVG and that of the Histogram are pronounced; the areas in which the two concepts differ however lie in the manner in which each bin is defined. Histograms typically utilise bars that differ in size and sometimes also colour, to communicate the frequency of points that lie within each bin. Whilst the density map binning technique uses the greyscale spectrum and the uniform size of each cell to present the results. This relies on the human brains ability to quickly recognise contrasting regions and patterns within an image (Ronacher, 1984) to be successful. Also by not depending on specific colours, rather the hue of the greyscale spectrum; it also ensures that differing visual attributes such as colour blindness have little to no effect on the usefulness of the technique.

The density map method is not available within the three dimensional scatter plot, the reason for this is that filling each cuboid entirely would mean that the final result would be an impenetrable block with only the outermost cells visible. Methods to overcome this might include varying how transparent the outer layers of cells are, with only the very innermost being totally opaque. Alternatively the ability to view each layer of cells could be provided, very much in the same way as a threshold value is provided whilst visualizing a volume dataset using Iso Surfacing techniques. To successfully implement the density map within a three dimensional environment is certainly not impossible and could provide an interesting overview of a three dimensional scatter plot, the exact implementation requires further study to find the best methods for the purpose.

The strength of this technique is its ability to redefine cluttered visualizations in a manner which allows for the general pattern of the data density to be quickly disseminated, this is often one of the more important aspects highlighted by information visualization techniques.

The greyscale values themselves are generated based on the single count value for each cell. A representative value between 0 and 255 is generated, with 0 representing the lowest count value while 255 is representative of the highest. By default black is considered to be high therefore an RGB 3-tuple is generated for each cell by subtracting
the calculated value from 255, ensuring that the highest value becomes black (0) and the lowest becomes white (255).

Clearly the greater the number of cells, the more detailed a density image is going to be obtained. The granularity of the original dataset dictates at which point the resultant binned visualization is possibly less useful than the original scatter plot.

As figure 3.22 shows, increasing the number of bins also increases the accuracy of the resulting visualization. With a 5 by 5 binning grid we get an overview of the dataset and can see that there are two main areas denser than the rest. It is only as we increase the number of cells by using a 10 by 10 binning grid that we begin to see with more
accuracy where these areas truly lie. Increasing the granularity further would reveal more detail, however the low density of the Anterior Lumbar Root Dataset seen in figure 3.22 means that above a 20 by 20 grid, the density map provides almost the same information as the original scatter plot.

To gain a better insight into the usefulness of this technique in highly cluttered situations, figure 3.21 shows a dense dataset with a very small range of values but a large number of data rows. The density binning method is used to extrapolate a new visualization which provides us with an insight as to the true nature of the data. From a large mass of blue pixels we can now see the relative density of more specific areas, with more bins than the 225 seen in figure 3.23 we would be able to see even more precisely where the denser areas lie.

The density map binning method is also available via the Parallel Coordinates visual style. It is used to provide coloured overlays representing how dense the data in a specific area of each axis is. This can be very useful in disseminating information from a visualization style that is often cluttered. The basic concept for this form of binning is taken from ‘Parvis’ (Ledermann, 2003) but modified so that it does not rely on Histogram like bars to show the frequency within each bin. Rather the simple concept of a greyscale colour map is utilised as per the two dimensional scatter plot. Arguably this
offers a simpler method of providing an overview of data density without sacrificing any of the detail different sized bars would offer.

The 15 cell (165 cells in total) binning grid applied in figure 3.24 provides a general overview of the density of all eleven variables. As the resultant visualization only overlays the binned results, rather than actually replacing the original lines, none of the real data is lost while still making it clear which areas are denser than others. Clearly the more bins we apply, the greater the detail the process will reveal.

Arguably the density binning is even more successful than the arithmetic mean method for tackling the issue of data overlay caused by integer datasets. We can reuse the same example seen in section 3.3.2 to highlight the benefits of this methodology. Therefore we start with a two dimensional scatter plot as per figure 3.25.
Figure 3.25 Two Dimensional scatter plot showing the first letter compared against the second letter from a sample 'Scrabble' dataset

We then apply a 20 by 20 density binning grid with black defined as the high value, resulting in figure 3.26. The same conclusions that the arithmetic density mean binning offered are still present, however the results are perhaps more quickly disseminated due to the simpler presentation of the density method. Where this methodology does succumb is in the accuracy it offers. By providing an actual point representation within each bin, a more exacting location for each binned value is evident, whilst the density binning method loses this detail.
3.4 Data Clustering

Data clustering provided within MVG has a different goal to that of the majority of clustering techniques in general use. Traditionally clustering algorithms are provided to the user as a function to be applied, the results are then presented either numerically or visually. Some of these algorithms require basic input before they can begin; an example of this is perhaps the most simple of clustering algorithms ‘K-means.’ This algorithm requires the definition of how many clusters the user wishes to achieve and also how many times the algorithm should iterate before it can begin. It does not allow for specific values such as the starting location of each cluster to be defined as its intended goal is to find the most general clusters that are correct for the current dataset.

The clustering within MVG is a variation on the basic premise of most algorithms, with the major difference being that it aims to only find a single cluster through the data and that it allows the user to set its starting centroid interactively as well as define the Euclidean distance they wish to be taken into account. Its purpose is not to
automatically define multiple areas of data as clusters. It is proposed that this more manual method of clustering is most suited to users who know the current area of data that they are interested in and wish to find patterns within that specific area.

If defined within a Data Mining context the clustering within MVG would likely not be defined as a clustering algorithm, rather it would fall under the premise of a classification algorithm. It begins by the user pre defining the distance from each point that they wish to be taken into consideration, as is the case with Classifier algorithms when the user defines the classes they wish the data to fall into. The method within MVG also bears a resemblance to a nearest neighbour neural net, whereby each new point is used to draw its nearest applicable neighbour into the fold of the class or cluster.

The eventual decision was that, due to the interactive nature of the clustering within MVG, it did not truly fit any of the predetermined categories most often found in disciplines that make heavy use of such techniques; therefore it was most correct to simply refer to it as clustering, as this is instantly recognisable as a concept and does still accurately describe the eventual result of using the functionality.

Only the two dimensional scatter plot contains the clustering capability within MVG, however transference into the three dimensional scatter plot is certainly an interesting prospect that is deserved of future study. The main problem to overcome is that of the accurate placement of the initial cluster centroid within three dimensional space.

### 3.4.2 Visual Cues

As with every other aspect of MVG, the parameters for the clustering algorithm are defined almost entirely using visual cues. Specifically, the user is required to specify their initial starting centroid and also the maximum distance between any two nearest neighbours.

Placement of the starting centroid is done by simply clicking on the visualization, which results in a visible red dot. The user defines a maximum distance they wish to be taken into consideration for a new point to be included into the cluster. A simple slider is
provided within the interface, which allows a circle representing this distance to be drawn with a centre at the starting centroid.

The distance is not defined numerically; rather it is purely a visual measurement. Arguably this could be a hindrance if the user is exactly aware of the distance between each point they are interested in. However relying on a visual cue removes any element of confusion that can occur due to different scales of axes differing datasets can generate. It can also be argued that if a dataset was being analysed using a purely visual environment such as MVG then it is likely the user would wish all aspects of their interaction to be visual in nature; including the application of a clustering algorithm.

![Figure 3.27 Clustering within MVG using a two dimensional scatter plot, example generated using the Anterior Lumbar Root Dataset](image)

The example shown in figure 3.27 highlights the intended purpose of this form of clustering. In the top half we can see the properties as defined by the user (which are also shown via the actual control panel on the right hand side) and in the bottom half we can see the resultant cluster that is formed. It is then possible to export all of the values
which are contained within the cluster as a new CSV file, either to be analysed numerically or perhaps read back into MVG.

### 3.4.1 Algorithmic Concepts

The basic premise of the MVG clustering algorithm is that it is designed to identify which data points fall within a specific set of criteria while only ever being as recursive as is absolutely required. To categorise the algorithm it would be correct to say that it is partitional in nature, sharing similarities with a nearest neighbour neural net.

The algorithm is designed to select a single cluster from a set of points which are all linked via the property of a maximum Euclidean distance, to at least one other current member of the cluster. Other partitional algorithms such as K-means attempt to find a set number of clusters throughout a whole dataset based on a centroid derived from the average centres of all the points currently contained within each cluster.

To formally describe the algorithm within MVG we can say that when point \( Q \) is a member of the set \( S \):

\[
Q[x, y] \in S
\]

and the maximum Euclidean distance allowed from each tested centroid is defined as \( D_{\text{max}} \). Then for \( Q \) to be a member of the set \( S \) the following needs to be true.

\[
\exists (P[x, y] \in S) \text{ s.t. } (D_{\text{max}} \geq d(P[x, y], Q[x, y]))
\]

where the function \( d(P[x, y], Q[x, y]) \) is defined as.

\[
d(P[x, y], Q[x, y]) = (\sqrt{(P[x] - Q[x])^2 + (P[y] - Q[y])^2})
\]
Alternatively the algorithm can be described in pseudo code form as per figure 3.28.

```
| selPoints[0] = User Selected Starting Centroid (x,y);
| for (outer=0; outer < selPoints.Length; outer++)
|     for (inner=0; inner < mainData.Length; inner++)
|         currEuclid = SquareRt((mainData[inner].x - selPoints[outer].x)^2 + (mainData[inner].y - selPoints[outer].y)^2);
|         if(currEuclid <= userEuclid)
|             if(selPoints does not contain mainData[inner])
|                 Add mainData[inner] to selPoints[outer + 1];
|         end if
|     end if
| end for
```

(N.B. The actual implementation within MVG employs the same algorithm as the circular brush to calculate the Euclidean distance, not the traditional Euclidean equation seen in the above code. This reduces computational overhead)

*Figure 3.28 Pseudo Code description of MVG's clustering algorithm*
Chapter 4

4.0 Test Cases

4.1 Lumbar Anterior Root Stimulation Data

In 1999 an experiment was performed (Hoornaert, 1999) to test the response of a paraplegic patient to direct nerve stimulus. A bespoke set of measurement apparatus was developed known as the multi modal chair (Donaldson et al., 1999) (Wood et al., 1999). It rigidly held the patients lower limbs in place whilst measuring the isometric movement forces that differing levels of stimuli to four major nerves produced. Once the raw movement data had been recorded a set of neural net algorithms were run, resulting in a new refined dataset. Specific details of this process are described by (Hoornaert, 1999).

The dataset itself has eleven variables; the first seven are the isometric movement values:

- Hip (Hip Extension)
- Knee (Knee Extension)
- Ankle (Plantar Flexion)
- HipAb (Hip Abduction)
- AnkleAb (Ankle Abduction)
- HipExRot (Hip External Rotation)
- Inversion (Inversion)

and the remaining four variables are the stimulus values which are a percentage between a minimum and maximum μV value:

- L3 (Lumbar 3)
- L4 (Lumbar 4)
- L5 (Lumbar 5)
- S1 (Sacral 1)
As the dataset has been pre-processed it is relatively compact, consisting of 576 individual rows of data and therefore there are 6336 individual data values.

### 4.1.1 Goals of Data Exploration

The reason that this data was originally collected was to generate a concise inverse correlative mapping between the stimuli values and the specific movements they produce. The goal therefore is the stimulus values that produce a specific set of movements need to be extrapolated from the dataset and reliably catalogued.

Presuming that a reliably generated mapping was produced then in theory it should be possible to apply the exact same set of stimuli to the same patient and produce the desired physical movements, allowing for the possibility of artificially restoring movement to the immobile limbs.

### 4.1.2 Previous Analysis

Multiple attempts to extract definite stimuli values from the dataset have been attempted. The first and most in depth analysis of the dataset was performed by the data’s originator, Prof. N. Donaldson and his colleague Peter Hoornaert at University College London in 1999.

The goal was to analyse the data in a pseudo three dimensional fashion using multiple two dimensional scatter plot representations, this was facilitated using a graphical Delphi implementation (Hoornaert, 1999). They then considered the best way to use the system to pick out a very specific data subset which showed the stimuli values required to achieve a specific set of movements. This was attempted using a form of nearest neighbour selection (Liu and Hoornaert, 2000) in which the user was able to click two vertexes on any of the scatter plots; each vertex was then matched to the nearest actual data point and a resultant line drawn. A numerical threshold value was then set and the points that fell within the defined perpendicular distance of this line were highlighted.
Finally the selected points could be exported and then imported back into the application essentially allowing for a form of data reduction.

The eventual conclusion was that an accurate mapping was ultimately impossible using the tried techniques. The primary reason for failing to find an accurate method to identify the correct stimulus values was that the methods used were inadequate. It was also put forward that the dataset itself was too coarse in nature. Finally research emerging at the time suggested that human nerves actually regenerate over time, something that was previously thought to be untrue. This essentially meant that the same stimulus values applied to the same patient as little as 12 months later could result in different movements due to the altered state of the nerves being stimulated. The analysis was eventually abandoned.

### 4.1.3 Application of MVG

In order to best test the functionality and indeed suitability of MVG with the Anterior Lumbar Root Dataset. The person behind the majority of the datasets prior analysis, Mr. Peter Hoornaert was given a detailed demonstration of all aspects of the tool. He was then asked to present his thoughts and feelings with regards to the usefulness of MVG for this specific task. He was also asked to complete a standardised questionnaire, the results of which can be viewed in Appendix C. Any quotations of comments made by Mr. Hoornaert were either collated during the face to face meeting in which the functionality of MVG was demonstrated, or directly from the standardised questionnaire. Quotations taken from the face to face discussion may be paraphrased in part due to the informal nature of the meeting however the original meaning is maintained. Permission for further published study into the dataset was obtained from its originator, Prof. N. Donaldson.

Previous efforts to analyse the dataset had utilised only one form of visualization, the two dimension scatter plot. The logical starting point within MVG was therefore to test if any of the alternative visual styles available were likely to be more successful as a standalone technique.
Both the two dimensional and three dimensional scatter plots offered interesting insight into the data, but neither offered much more insight than the previous attempts. Similarly the Star Glyph visualization was too abstract a technique for this dataset, where direct correlations are desired rather than patterns. The Parallel Coordinates offered an entirely new method of looking at the data. The technique had not previously been used for this analysis but as a standalone visualization style its results were too cluttered.

In figure 4.1 it is difficult to pick out where the corresponding stimulus values for specific knee movements lie. To overcome this, two brush groups with opposing colours were introduced, as a test the top and bottom \( \sim 15\% \) of ‘Knee’ values were brushed. The advantages of being able to highlight desired ranges of motion in specific joints became quickly obvious. Once all non brushed data was set to invisible, we were left with figure 4.2.
We now begin to see a definite correlation between low and high movement values for the Knee joint. Although far from conclusive it would not be premature to state that higher L3, L4, L5 and lower S1 stimuli values produced higher, positive movements while lower L3, L4, L5 and higher S1 stimuli values produced lower, negative movements. Some ambiguity exists within L3.

“It will need more study, but what I am seeing [in this visualization] is very exciting. We certainly never managed to show so easily how different movement values mapped to the stimulus values.” (P. Hoornaert, Demonstration)

One important aspect and perhaps the most complicated part of this study is that each joint cannot be considered on its own, to produce a useful set of movements, each joint has to be taken into account compared to every other. The Parallel Coordinates was also ideal for providing a good overview of this effect.
As seen in figure 4.3, by enabling all eleven variables we are now able to see the relative effect of the defined stimuli values on all seven joints. Although the meaning requires further study by those with a deeper understanding of the movements themselves, the combination of brushing groups and the Parallel Coordinates visualization has proven to be an effective method to see general trends throughout all eleven variables for a specific subset of values.

“What I am seeing is entirely in line with what I would expect because of the limited range of movement that each joint has. To see it so clearly side by side though is very interesting. The possibility of being able to see where to look in the data to build a more exact model seems apparent. The fact that clear groupings can still be seen in the stimulus [data] columns is also very interesting.” (P. Hoornaert, Demonstration)

Although the Parallel Coordinates provided a very good visual overview, the ability to then see in closer detail which stimuli values had been selected was then tested. It was decided that the best way to do this would be to view the brushed points within the scatter plot visualizations. Initially the three dimensional scatter plot was used as this most closely mimicked the pseudo three dimensional concept originally implemented.
The $x$, $y$ and $z$ variables in figure 4.4 were mapped to the L3, L4 and L5 variables respectively, and the non brushed data was removed. The visualization seen in figure 4.4 was deemed quite interesting, we can clearly see that two distinct clusters of points exists within the three dimensional space, the brush colour assignments remain from the brushing effort undertaken in figure 4.2. Upon consideration it was decided that although it was interesting to see the clustering effect, it wasn’t really showing anything more than the Parallel Coordinates originally did, just from a different perspective.

Multiple two dimensional scatter plots were therefore examined, which proved to be a more accurate method of analysing specifically which stimulus values had been selected. Through generating scatter plots of each value against another and visually removing all non brushed data points, the values become far more apparent. This can be seen in figure 4.5.
Chapter 4. Test Cases

Figure 4.5 Each stimuli value against every other.
Figure 4.5 made it clear that the inclusion of a scatter plot matrix function within MVG would be desirable. The figure also shows that through linked interactive visualization we are able to define an interesting subset of data interactively via brushing any applicable visualization, in this case the Parallel Coordinates. We are then able to see in more detail exactly which points were brushed by using a different visualization, the two dimensional scatter plot is better suited for this specific purpose.

“I can see that by using both the Parallel Coordinates and the scatter plot together I should be able to explore the data according to the movement values themselves, this is something we could not do before, we simply had to take an educated guess as to the best set of points to look at.” (P. Hoornaert, Discussion)

Ultimately, the eventual desired result was a subset of the actual numerical values. The ability within MVG to extract a specific brushed set of data and save as a new dataset, provides this and should be considered an invaluable addition to any similar data visualization systems.

The next avenue of exploration was to try and mimic the previous analysis techniques to see if MVG offered a better method of implementation and could perhaps improve on the previous analysis efforts. Specifically, we wished to recreate the cylindrical based nearest neighbour mapping (Liu and Hoornaert, 2000). Prior attempts utilised a conceptual cylinder placed around a line drawn from two vertices interactively placed on the scatter plot by the user. Within MVG this functionality can be loosely recreated by using the Quadrilateral brush shape. An exemplar case of this being attempted can be seen in figure 4.6.
Conceptually the entirely freeform nature of the quadrilateral brushing shape allows an equivalent of the ‘cylinder’ to be defined visually on a two dimensional scatter plot. Where MVG currently falters is that only the two viewed points are brushed, to follow the exact methodology employed previously, the whole data row each point was associated with would also have to be brushed. This may not be conceptually correct for other cases examined using MVG and therefore must be used as a good example of how generic software solutions cannot always fulfil every requirement of a specialised user.

*Figure 4.6 Quadrilateral brush in use within MVG*
The ability to work with abstract shapes within the data was still considered an interesting prospect and worthy of further study.

“I would like to explore this brushing technique further as it is very similar to the removal of data I have previously explored but this treats it visually.” (Hoornaert, Questionnaire [Section: Brushing])

The final area that was examined was the clustering available within MVG. The method implemented actually bears resemblance to the nearest neighbour methods previously used to attempt to gain insight as to any distance based links within the dataset. It was decided that the best way to use this functionality would be to mask an area of interest using the Quadrilateral brush and then cluster based on these values.

As meaningful analysis using the clustering tool would require in depth knowledge as to the data that should be looked for. A subset of points surrounding the line of best fit for a scatter plot of the L3 stimulus variable against the Knee variable was used as an example. This initial brushed group can be seen in figure 4.7.

![Figure 4.7 Two Dimensional scatter plot within MVG, showing the data subset used to test clustering functionality](image_url)
A starting centroid was picked near the line of best fit and a distance radius defined. The cluster defined by the algorithm was as expected. While this demonstrated the technique to work, correct usage of the method for this dataset will require more in depth analysis by the dataset’s owner. The results of the test can be seen in figure 4.8.

It was agreed that theoretically the method could have merit as it is so similar to the basic concept of nearest neighbour analysis undertaken previously.

“I don’t want to say right away that it is going to be useful, but I like the general idea. I like the fact that I can see the distance that is going to be taken into account. One thought though is that it is not necessarily obvious which points are going to be included and which aren’t, I don’t know which points will be included so I can’t really visualize around which points that circles radius will be considered.” (Hoornaert, Discussion)
4.1.4 Conclusion

Reaching a definite conclusion as to whether MVG is a success for analysis of this dataset without further study would be incorrect. It is a fair assumption however that given time the functionality it offers should open different avenues of research.

Some fundamental issues such as the nerve regeneration factor would need to be taken into consideration. Further research has also shown that it is also likely a dataset would only be applicable to the test subject used to generate it, therefore a new set of results would need to be created for each patient.

A software tool such as MVG was only intended to offer an experimental environment, in which assumptions made about a dataset can be visually proven or disproved. In this capacity, at least for this test case, it has certainly proven the worth of linked multiple visualizations. Similarly the ability to define individual brushing groups, combined with the non constrained brushing shapes, such as the quadrilateral, appear to offer a powerful analytical combination.

Both the binning and clustering have perhaps not been entirely vindicated through this test case. The binning received little to no attention due to the low granularity of the dataset and therefore cannot be commented on. The clustering clearly has possible application, filling a different role to most clustering techniques, as does the visual application method used in MVG.

“I think the software you have produced would be useful for the dataset I was analysing in that it would allow me investigate further a subset of the data and be able to study that further. Your application handles visually what I tried to do with the cylinder idea, and the points that lie within it.

I think your application is particularly useful in that it visually represents the data in some different ways from those I was looking at, and I think this would encourage me to look at the data afresh. I can't say where this might lead me as I haven't had enough time to look again at your application and then time to reflect on it, but it certainly has made me think in some new ways, which is excellent.” (Hoornaert, Questionnaire [Section: Final Comments])
4.2 WMIC MRI Data

The Wolfson Molecular Imaging Centre at the University of Manchester is a technology hub, providing researchers from many disciplines access to some of the latest imaging hardware. One active area of study is that of human brain tumours, specifically using MRI (Magnetic Resonance Imaging).

The study of tumours is still a largely manual process, requiring in depth knowledge. MRI data is converted to a series of high contrast images, which are essentially slices through a three dimensional volume. This results in images most will recognise, an example of which can be seen in figure 4.9.

![Figure 4.9 Example high contrast MRI image](image)

Analysis of brain tumours for diagnostic purposes is often a manual process involving visual study of the MRI image data. In depth research however has often required a more statistical approach. The most common method is to mask a specific area of an MRI image and extract all of the appropriate pixel coordinates and their luminescence value. Once this is complete manual manipulation of the values eventually results in
compiled datasets representing specific types of tumour or perhaps tumours at different stages of development.

4.2.1 Goals of Data Exploration

The analysis involved in this test case aims to see how basic imaging characteristics and properties, relate to specific subtypes of tumour. With the goal of analysing these parameters in terms of the patient’s outcome, therefore aiding in predicting future patient prognosis.

4.2.2 Previous Analysis

Currently no software like MVG is used in the analysis process of the MRI data by the person involved in this study, Dr. Samantha Mills. Previous efforts have been made to incorporate a few of the less usual visualization styles such as Parallel Coordinates but generally speaking the main visualization technique used two dimensional scatter plot. All data to be analysed has been heavily manipulated before it is ever used to generate a visualization.

There exists publically available tools solely intended to deal with the standard imaging formats used in this type of research, namely ‘DICOM’ and ‘ANALYZE’. Specifically Dr. Mills currently uses ‘MRICron’ (Rorden, 2007) and ‘ImageJ’ (Abramoff, 2004). These tools are used to define and extract areas of the image that contain a tumour. The data is then collated and manually manipulated and finally compiled into a conglomerate dataset. This is then considered numerically. Occasionally the results are viewed with a simple visualization such as a scatter plot.
4.2.3 Application of MVG

The usage of MVG was overseen by Dr. Samantha Mills of the WMIC. Dr. Mills was given a detailed demonstration of all aspects of MVG and asked to present her thoughts as to the usefulness of the tool for this specific task. As with the first test case, Dr. Mills was also asked to complete a standardised questionnaire, the results of which can be viewed in Appendix C. Any quotations of comments made by Dr. Mills were collated during the demonstration, received in the time following or taken directly from the standardised questionnaire. Quotations taken from the demonstration may be paraphrased in part due to the informal nature of the meeting however the original meaning is maintained.

As the functionality offered by MVG was not currently in use by Dr. Mills, all of the concepts and their possible real world application were presented in a logical order. The first analysis attempt involved converting a DICOM image to an ASCII PGM image file. PGM is a form of encoding which lists each pixel in a greyscale image as a luminescence value between 0 and 65536. Therefore a PGM file of an image of size 128 x 128 pixels would contain 16384 individual ASCII values. The file is not logically laid out; rather it is presented as up to sixteen columns of data, in serial order, though the number of columns depends on the application used to generate the PGM. In our 128 x 128 example, presuming we were using the maximum sixteen columns, this would mean that the first eight rows of data would actually represent the first 128 pixels of our image. In actuality, the software used to generate the PGM file found in this test produced an 8 column layout.

The PGM file format is easily converted to a CSV input file for MVG and for this example a version of the image is seen in figure 4.9. It became instantly obvious why this form of analysis was rarely undertaken, the clutter and density of such a dataset makes visualizing it an almost entirely futile task. As was also pointed out by Dr. Mills the non uniform nature of the PGM file means correlating specific pixel values with the original image is a difficult task.

Although it is possible to produce scatter plots and parallel coordinates, they essentially have little to no usable meaning in the context of tumour analysis. Had this not been the case however the binning available within MVG certainly could have provided a usable
method of disseminating information from such a highly cluttered dataset. Data could be brushed in a scatter plot and the lines thinned in the Parallel Coordinates by turning off any non brushed data. Alternatively the density map binning method could be put to use in determining where the densest areas lie. An example of this can be seen in figure 4.10. The Parallel Coordinate visualization produced is nearly unusable but we are able to quickly see that the density of the ASCII values rapidly decreases the higher the number being considered. This is exactly as we would expect as the ASCII value for black is 0 in a PGM file and roughly 50% of the DICOM image used consisted of a black background.

![Figure 4.10 Highly cluttered Parallel Coordinates within MVG with binning enabled](image)

“*I don’t think looking at the data in this format is going to be of much use to me, I can’t equate what I’m seeing to the original image. The density binning is very interesting though and is actually something we have recently considered trying to implement as we feel it may be of use in our analysis efforts.*” (Dr. Mills, Discussion)

In theory, if the frequency of specific luminance values were of importance, a tumour could be masked off and exported as a new image and that converted to PGM. This would ensure that no extraneous ASCII luminance values existed.
Attention was then turned to extending the analysis style currently performed by Dr. Mills so that it was applicable to environments such as MVG. A dataset was prepared that contained specific metrics for four different types of tumour. Multiple entries existed for each tumour class with 28 rows of data in total. The metrics were as follows:

- KTrans \textit{(Measure of contrast passing from the intravascular to extra vascular space)}
- VE \textit{(Leakage Volume)}
- VP \textit{(Blood plasma volume)}
- ADC \textit{(Apparent diffusion coefficient)}
- FA \textit{(Fractional anisotropy)}

Traditionally these figures would be considered numerically, and any interesting correlations may have been presented visually using a simple two dimensional scatter plot. Within MVG however the ability to define a brushing group and the fact that one visualization can be used to affect another, offered an opportunity to quickly define and analyse the data.

“I used the brushing tool to identify subgroups of tumours (Metastases - black, grade IV Gliomas - red, grade III Gliomas - green and grade II Gliomas - blue). The variables included KTrans \textit{(a measure of contrast passing from the intravascular to extra vascular space)}, VE \textit{(leakage volume)}, VP \textit{(blood plasma volume)}, ADC \textit{(apparent diffusion coefficient)} and FA \textit{(fractional anisotropy)}. The Parallel Coordinates and Star Glyphs can then be used to show different patterns of the variables for the different tumours. I've found that quite useful. The dataset I used for this was generated by performing post processing analysis on a manually defined region of interest from a series of images for each given patient” (Dr. Mills, Successive communication)
The whole dataset was first examined using the Star Glyph visualization. A new brushing group with a distinct colour assignment was generated for each type of tumour contained in the dataset:

- Black – Metastases
- Blue - Grade II Gliomas
- Green - Grade III Gliomas
- Red - Grade IV Gliomas

The correct Star Glyphs were then brushed so they were included in the appropriate grouping. It was interesting to note that generally speaking every type of tumour generated very similar Glyphs. This can be seen in figure 4.12.

In order to view the results in more detail, a Parallel Coordinates visualization was then used. As each tumour type had been brushed previously, all lines within the Parallel Coordinates were already coloured. This view provided a new way to look at the data, allowing not only for side by side analysis of more than two variables, but also direct
analysis by tumour type. The resultant visualization can be seen in figure 4.13, where the first axis is a variable not defined in the previous text as its only use was as a definition for each tumour within the dataset.

Figure 4.13 Brushed Parallel Coordinates within MVG being used to show MRI data

Through the use of clearly defined colours as shown in figure 4.13, we can quickly begin to see that there exists a level of clustering of tumour types within ‘KTrans’ and ‘VE’ while a much more equal spread exists within ‘ADC’ and ‘FA’. This suggests that Grade IV Gliomas are more likely to have a higher value for the measure of contrast when moving from intravascular to extra vascular space. Through a collection of some of the techniques offered in MVG; it has been possible to visually highlight a possible pattern of interest within the dataset, a more definitive study would need to include more than 28 tumours.
4.2.4 Conclusions

As with the first test case, it would be incorrect to state that MVG definitely offers all the required tools and techniques needed. However this short study has opened up a few interesting avenues of exploration, the ability to visually define a single group of tumour types within a whole dataset, and then explore these patterns using different types of visualization, opens up the possibility of in depth data analysis. This may lead to patterns in the data becoming obvious that previously would have been unseen.

One conclusion that can be made from this test case and also from the first is that more often than not, post processed datasets are more easily explored than raw, unprocessed data, within tools offering advanced visualization techniques such as MVG.

“The dataset I used for this was generated by performing post processing analysis on a manually defined region of interest from a series of images for each given patient. I think once we’ve done the post processing analysis your software is particularly useful, this may not be as true with raw data.” (Dr. Mills, Successive communication)
4.3 Hadley Centre Weather Data (UK Met Office)

The Hadley centre for Climate Research is a specialised group within the UK Met Office involved primarily with research into global climate change. In association with the University of Manchester several climate models are currently being explored. The Manchester Visualization Centre (MVC) is involved in producing easily understood visualizations using the Hadley climate models, which shows the data with as much impact as possible. The chief instigator of this work within MVC is Louise M. Lever.

In depth statistical analysis of this data is not currently undertaken at Manchester. However all people currently working with the data have a good understanding of its general meaning. It was therefore chosen as an interesting dataset which should allow certain aspects of MVG to be explored. As statistical analysis on this data is not performed within MVC, there are no citable examples of previous techniques used for analysis. The section describing previous analysis efforts has therefore been omitted.

The original dataset was meant for two dimensional geometric rendering techniques; therefore a smaller downscaled version was generated for use in MVG. This consists of the following variables:

- Latitude
- Longitude
- Month
- Year
- Sealce (Level of sea-ice coverage from 0.0 (none) to 1.0 (total))
- AbsTemp (Absolute temperature in degrees Kelvin)
- DiffTemp (Change in temperature from the control dataset)
- AbsPrecip ($\sqrt[3]{\text{absolute level of precipitation}}$ (Kgs-1))
- DiffPrecip (Change in precipitation from the control dataset)
Two identically defined datasets exist that represent two separate scenarios as defined by the Intergovernmental Panel on Climate Change. The first, ‘A1B’, is the prediction of a model of rapid convergent growth, meaning:

- Rapid economic growth.
- A global population that peaks in mid-century and declines thereafter.
- Rapid introduction of new and more efficient technologies.

The second model, ‘A2,’ is the prediction of a model based around a fragmented world, meaning:

- Self reliance and preservation of local identities.
- Fertility patterns across regions converge very slowly.
- Continuously increasing global population.
- Economic development is regionally orientated.
- Per capita economic growth and technological change are more fragmented.

Each contains 497,665 data rows and therefore there are 4,478,985 individual data values, making a total of 8,957,970 data values between them.

### 4.3.1 Goals of Data Exploration

Although the main goal of those involved in the Hadley data analysis at Manchester University is not to perform in depth statistical investigation, some basic desirable outcomes of such analysis can still be defined.

When considering the Hadley datasets, ice levels decreasing at the poles, or increasing away from the poles is likely to be considered undesirable. Similarly a large increase in temperature or precipitation will show a notable shift in the current climate over the affected area.


4.3.2 Application of MVG

Due to the quantity and undetermined desired output of analysis of the data, it was decided that an equal but much smaller subset by year would be taken from the ‘A1B’ dataset. Normally analysis would be performed on both datasets to allow for comparison between the two scenarios, for the scope of this test case however it was sufficient to see if MVG offered any tangible benefits whilst trying to find patterns or correlations in the data. As all polar coordinates were included in the test dataset, results should be considered to be worldwide rather than localised to one specific region. Once the study was complete, the same standardised questionnaire was completed by Louise, the details of which can be seen in Appendix C.

To gain an overview of the dataset a Parallel Coordinates visualization was generated. It soon became apparent that the data was very closely clustered on each of its variables so the resultant visualization was not as informative as was hoped. An example from the ‘A1B’ dataset can be seen in figure 4.14. It is apparent that the 83,000 rows of data within the sample sub set all contained values within a relatively compact range for each variable.

![Figure 4.14 Initial Hadley data Parallel Coordinates within MVG](image-url)
An attempt to separate the data by month using brushing groups was therefore made. As figure 4.14 shows, in its initial state defining the twelve subgroups by month, via the Parallel Coordinates, would be difficult. The interactive zoom functionality was therefore used, resulting in the visualization seen in figure 4.15.

![Figure 4.15](image)

*Figure 4.15 Highly zoomed Parallel Coordinates of Hadley data within MVG*

This allowed each month’s data subset to be brushed using twelve separate brushing groups. The brushed variant is shown in figure 4.16 so as to provide a reference as to which colour maps to which month.

![Figure 4.16](image)

*Figure 4.16 Highly zoomed Parallel Coordinates of Hadley data within MVG, with brushing applied*

With brushing groups separating each month’s data, control was now gained over which months were currently being displayed and which were not. Two dimensional scatter plots were used at this point to place the absolute temperature against the absolute precipitation. The initial visualization can be seen in figure 4.17. Although it looks incomprehensible, it is interesting to note that a very similar pattern exists for each
month with outlying irregularities existing in seemingly random amounts for unspecific months.

![Figure 4.17 Two dimensional scatter plot within MVG showing brushed Hadley data](image)

In an attempt to view any interesting differences between the seasons, January and July were chosen to remain visible and all other months were set to hidden. This action greatly thinned the visible data points, resulting in figure 4.18.
The red points in figure 4.18 are the values for January while the light blue points are for July. At first glance the result we would expect is visible; the lower values for temperature are sparser in July than January, however it is worth remembering that the dataset is global rather than just from the Northern Hemisphere. This may blur the usual summer / winter distinction due to the seasons occurring at different times in different parts of the world. The Northern Hemisphere contains a larger amount of land than the Southern Hemisphere; this may be an explanation for the apparent imbalance in temperatures that exists roughly in the centre of the plotted points. Perhaps what is not expected is the trend of increasing precipitation compared to temperature. Upon first consideration it follows that the hotter the climate the less moisture is likely to be present. Clearly this is not the predicted case for 2010 as we can see in figure 4.18. The density of points makes it unclear as to whether the general trend is a reduction in precipitation past a certain temperature or if it is a steady increase.

“The effectiveness of the brushing is greatly improved by the ability to hide other non-selected groups.” (L. Lever, Questionnaire [Section: Brushing])
It was decided that the binning techniques within MVG may offer insight as to the general trend of the data. A high granularity binning grid of 45 by 40 was applied using the Arithmetic mean binning method, to all twelve months worth of data points. The results gave an insight as to the general trend of the data.

![Two dimensional scatter plot within MVG of the Hadley data, with arithmetic mean binning applied](image)

Over the course of twelve months there is evidently a peak point at which temperature and precipitation values converge, this appears to occur at a temperature of ~275 Kelvin. What is also interesting is the fact that there does appear to be a noticeable drop in precipitation after this point, with the most densely populated areas below the peak value. This perhaps suggests that within the model, there is a peak point at which precipitation begins to decrease as the temperature increases.

Alternatively, the same binning grid applied but using the density map method produces similar results that are perhaps less detailed but arguably more easily analysed. It should be noted that the arithmetic mean binning offers greater distinction between the most densely packed areas, with the obvious single hotspot from figure 4.19 at ~275 Kelvin appearing to be three areas within figure 4.20.
Through altering the contrast of the density binned visualization from black being high to white, the hotspot do become clearer, as can be seen in figure 4.21, however it is undeniable that the arithmetic mean is superior for this specific purpose in this case.
“This is a good way of reducing the clutter and conveying an overview of the data. User defined cells would be a useful addition, as would cells based on other scales e.g., logarithmic. Interactive refinement would help where one or more cells could be subdivided when required.” [L. Lever, Questionnaire [Section: Binning]]

4.3.3 Conclusions

This test case was unusual in that no predefined task was being attempted with the dataset; rather exploratory analysis by somebody with no prior experience of the data was attempted. It should therefore be treated as a documented test of the techniques within MVG rather than a test as to whether tools such as MVG are suitable for analysing climate model datasets.

The ability to define multiple brushing groups once again proved to be invaluable during the analysis of the Hadley data. Definition of the data by month offered the ability to refine which data was currently being viewed, meaning the scatter plots produced could be as detailed as required.

Aside from providing an easy platform in which to brush the data by month, the Parallel Coordinates visualization within MVG proved to be mostly redundant with this dataset. This test case highlighted why being able to scale each axis to its own maximum and minimum rather than the global values can be an invaluable feature. Had this been available within MVG then the Parallel Coordinates may have been far more insightful.

The usefulness of the binning within MVG was also highlighted by this test case, as a tool for providing insight into highly cluttered datasets. In the case of figure 4.17, we were able to see that a general pattern existed within the data when comparing temperature to precipitation; however the general trends throughout the dataset for 2010 were not easy to see although exacting values for each month could be extracted. The arithmetic mean and density binning methods both provided reasonable insight into the data, and the test also clearly highlighted that one of the two different binning methods can be more useful in certain situations.
Chapter 5

5.0 Conclusion

The primary goal of this investigation was always to try and verify the concept of linking multiple visualization techniques, in order to produce an analysis environment of more worth than the individual visualizations could provide on their own. The three test cases provided are the primary evidence presented as proof, however some of the content seen in the main body of the text should also be considered, as most examples seen throughout the dissertation were produced using real world datasets.

To briefly reiterate its initial goals, this study aimed to:

- Validate the concept of linked visualization environments employing multiple visual techniques.
- Extend and build upon brushing, binning and clustering techniques currently in general use.
- Show that the use of interactive visual cues could be considered a viable if not preferable alternative to numerical parameter definition.

The development of a working prototype was integral to all three goals, providing a platform through which ideas could be tested and more importantly demonstrated to others.

All three tests cases, which can be reviewed in chapter four, quite conclusively show that offering more than one visualization technique within the same environment and perhaps more importantly, providing an interactive link between them, allows for a more powerful analytical tool. The first test case, dealing with the Anterior Root Stimulation dataset perhaps best highlights this, whereby the Parallel Coordinates visualization offered a good overview of the whole dataset, but it was only by using different brushing groups on the Parallel Coordinates, then utilising these distinctions on a two dimensional scatter plot, that the actual detail within the data became apparent. Each visualization style offered a good overview in its own manner, but when the two
could be effectively combined, the strengths of both were used to disseminate more from the data.

Reasonably complex brushing, binning and clustering were all explored through the MVG prototype. The brushing tools not only extend the basic premise of data masking but also provide some novel approaches to the areas of brushing area definition and brushing group application. Certainly the concept of a scalable circular brush extends the basic paint ‘daub’ methodology initially seen in ‘Mirage’ (Ho, 2003). The persistent visual cue of a semi transparent circle, which can be altered in size during its placement, offers distinct usability advantages over traditional methods. The ability to define triangular and quadrilateral brushing areas allows for abstracted techniques such as the cylinder concept found in the initial analysis of the Anterior Root Stimulation dataset (Liu and Hoornaert, 2000) to be recreated quickly, easily and more importantly, visually.

The binning presented within MVG is a novel and arguably successful integration of the basic concept of a histogram directly into the structure of visualizations such as three dimensional scatter plots and Parallel Coordinates. As implemented, the grid concept has shown itself to have some flaws inherent to its uniform nature. Including the inability to only bin a specific area of the visualization or perhaps apply differing sized bins to different areas at the same time. The majority of the issues highlighted could be rectified in a more complete implementation.

With regard to the method used for the binning itself, both the bubble plot produced by the arithmetic mean algorithm and the density map appear to have validity in differing circumstances. The third test case, using the Hadley weather data, showed that the bubble plot can provide better accuracy when consecutive bins produce similar results; while examples in chapter three showed that the density map can offer a more easily understood overview of very densely packed data points.

Clustering implemented within MVG was arguably relatively uncomplicated in its nature, having more in common with basic neural net techniques than traditional clustering algorithms. The novel aspect lay with how the algorithm was presented to the user, rather than the eventual outcomes of applying the method itself. Originally it was hypothesised that by allowing the parameters of advanced functionality such as clustering to be defined using interaction and visual cues, a greater understanding of
how the algorithm would generate its clusters may be imparted to the user. The test cases were not entirely conclusive in this respect, with clustering itself only being applicable as a technique in the first case. The general consensus was that the basic premise was a good one. Being able to visually define the applicable Euclidean distance being tested was well received, as was the ability to manually define where the clustering should start from. However the actual worth of the algorithm itself was not entirely apparent; given more study its use in the context of data analysis may become better defined.

Whether for brushing, clustering or binning, one conclusion that can be reached is that the use of visual cues within tools such as MVG is an important and powerful method of offering easily understood feedback to the user. As an example, without the visual cue representing the binning grid it would be very difficult to conceptualise exactly where each bin will lie once the binning process has been applied.

Finally, all three test cases showed that analysis environments such as MVG are more likely to be useful with post processed datasets rather than raw data. Both the Anterior Lumbar Stimulation and Brain Tumour datasets had been manually or automatically manipulated before they were examined using MVG, similarly the Hadley dataset was handpicked to show a specific subset of data. This may not be a universal truth as not all datasets can be analysed in the same manner, however it certainly appears to be a general trend.

5.1 Future Work

Many avenues of further exploration could stem from the initial work presented by this study. Perhaps the most pressing is the inclusion of further visualization styles into the same application. MVG only offers a very small subset of a much larger library of accepted techniques. The advantages of providing a more varied selection of linked visualizations are self explanatory. Of course, not all will be of use for every dataset but as the test cases showed, what is not useful for one may well be useful for another.

MVG’s brushing elements were quite conclusively vindicated in the test cases. The concept of freeform quadrilateral brushing certainly appears to be a success. One
desirable addition may be that of a truly usable freeform selection tool. The ‘lasso’ concept was explored during MVG’s development but a solution that was both user friendly and computationally sound was elusive. The ability to define each brush action was also partially explored within MVG in the form of setting a brush threshold by percentage; this technique requires much more research before it can be considered useful.

The grid based binning seen within MVG certainly appears to have merit; however issues for consideration have presented themselves during real world use. The ability to define each bin individually or at least offer the ability to deform the grid in some way would certainly seem appealing, as would a larger, more varied collection of binning methods.

Clustering with visual parameter definition appears to have merit; however it requires further study as to what its intended purpose should be and also under which set of circumstances it is most applicable. The extension of the concept to well established algorithms and techniques may also reveal interesting results.

From a purely developmental point of view, the MVG prototype is currently missing functionality that would be very desirable. Its development platform of C# is also less than ideal, as its entire lack of cross operating system functionality. The ability to output specific portions of a visualization using the most common picture formats would be a sensible addition also.

Finally the integration of a more complex cross visualization pipeline similar to that presented by ‘VisTrails’ (Bavoil et al., 2005) may provide a more robust visualization environment, with the important addition of logged and recorded actions resulting in the ability to recreate a past sequence of events exactly.
6.0 Bibliography


Bibliography


Appendix A: MVG Overview

The goal of this appendix is to provide a logical overview of all functionality contained within MVG. Throughout this document snippets of what is contained within the application have been demonstrated. This appendix offers a self contained reference guide, with the intention of providing full context as to what was achieved.

The general layout of this appendix is by visualization style rather than by functionality, therefore repetition of specific functionality is apparent. All examples are produced using the same random sample dataset containing five columns and fifty randomly selected values between 0 and 20.

A.1 Input File Format

MVG accepts data input in the form of a CSV (Comma Separated Values) file. Due to its prototypal nature it is very specific in its input requirements. The contents of the CSV file must be numerical in nature, except for the very first line which must contain the data headers, which can be any form of character.

The input file should take the following standardised format:

<table>
<thead>
<tr>
<th>Header 1</th>
<th>Header 2</th>
<th>Header 3</th>
<th>Header 4</th>
<th>Header n...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value 1a</td>
<td>Value 2a</td>
<td>Value 3a</td>
<td>Value 4a</td>
<td>Value n... a</td>
</tr>
<tr>
<td>Value 1b</td>
<td>Value 2b</td>
<td>Value 3b</td>
<td>Value 4b</td>
<td>Value n... b</td>
</tr>
<tr>
<td>Value 1n</td>
<td>Value 2n</td>
<td>Value 3n</td>
<td>Value 4n</td>
<td>Value n... n</td>
</tr>
</tbody>
</table>

MVG applies no size restrictions to the input file, theoretically allowing an infinite number of data columns (and therefore headers) as well as an infinite number of data rows. In reality due to the fact the entire content is reproduced in main memory, the size of the input file is directly limited by the amount of physical memory available within the hardware currently being used.
A.2 Two Dimensional Scatter Plot

The two dimensional scatter plot visualization style provides the user with the following options:

- General Visualization Options
- Data Brushing
- Data Binning
- Data Clustering

Using the two dimensional scatter plot, any two data variables can be visually represented against another at any one time. Selection of two variables is required before any of the advanced functionality is activated. An example can be seen in figure A.2.

![Figure A.2 Example two dimensional scatter plot within MVG](image)

A.2.1 General Options
Two general options exist for the two dimensional scatter plot.

- The background gridlines can be disabled and re-enabled.
- A line of best fit can be overlaid using a Least Squares Regression algorithm.

A masked example of the original scatter plot from figure A.2 can be seen in figure A.3. The gridlines have been disabled and a line of best fit overlaid. This can be seen through the centre of the figure.

![Figure A.3 Example two dimensional scatter plot within MVG with grid disabled and Line of Best Fit overlaid](image)

**A.2.2 Data Brushing**

Brushing within the two dimensional scatter plot offers the following functionality:

- Interactive data brushing using any one of four brush shapes (seen in figure A.3)
  - Deformable Rectangle
  - Scalable Circle
  - Free form Triangle
  - Free for Quadrilateral
Appendix A

Figure A.3 Brushing styles available with the two dimensional scatter plot within MVG

- Infinite brushing group addition and subsequent removal, each group can be:
  - Given any string characters as a name
  - Assigned any system supported colour

Figure A.4 Addition of a brushing group within MVG
• Any brushed group of data points can be visually hidden or revealed, as can the currently unbrushed data group.

Figure A.5 Brushing group visibility alteration within MVG
A.2.3 Data Binning

The basis of all binning within the two dimensional scatter plot is the binning grid. Once enabled this provides a visual overlay of a red grid appropriate in size to the number of $x$ and $y$ cells requested.

![Figure A.6 10 x 10 binning grid enabled on a two dimensional scatter plot in MVG](image)

Once the binning grid is enabled, two possible binning methods are available, before either is applied; the ability to decide whether only the currently visible data or all should be included in the binning action is available. The methods are as follows:

- Arithmetic Mean Binning (seen in figure A.7)
  - The size gradient of the resultant bubbles can be turned on or off
  - The colour graduation of the resultant bubbles can be turned on or off
Appendix A

Multi View Graphing

Figure A.7 Arithmetic Mean binning applied via a 10 x 10 grid on a two dimensional scatter plot within MVG

- Density Map Binning (seen in figure A.8)
  - Black or White can be set to represent the highest value

Figure A.8 Density Map binning applied via a 10 x 10 grid on a two dimensional scatter plot within MVG
A.2.4 Data Clustering

Clustering within MVG allows the interactive designation of a centroid for the algorithm to start at. It also allows the visual definition of the Euclidean distance to be taken into account when testing the data points for entrance into the cluster. The clustered points are shown visually and can also be extracted as a new CSV file (see section A.6).

![Figure A.9 Clustering on a two dimensional scatter plot within MVG](image-url)
A.3 Three Dimensional Scatter Plot

The three dimensional scatter plot visualization style provides the user with the following tools:

- General Visualization Options
- Data Brushing
- Data Binning

Any three of the available variables are required to generate a three dimensional scatter plot; therefore the user must select three variables before all available options are activated. An example can be seen in figure A.10.

Figure A.10 Three dimensional scatter plot within MVG
A.3.1 General Options

Three general options exist for the three dimensional scatter plot.

- The gridlines can be disabled or re-enabled.
- The current effect of using the right mouse button can be altered from the default state of causing the visualization to rotate, to causing it to translate.
- ‘TraceLines’ can be enabled or disabling. When enabled, a line traces from each point directly to either the \( y \) plane or to the \( x/z \) plane, or to both.

Figure A.11 shows an example three dimensional scatter plot with the grid disabled and TraceLines enabled.

![Figure A.11 Three dimensional scatter plot within MVG with gridlines disabled and TraceLines enabled](image)

A.3.2 Data Brushing

The concept of brushing groups exists within the three dimensional scatter plot with exactly the same implementation as seen in section A.2.2.
Only one brush type exists for the three dimensional scatter plot, a resizable cube. The cube is a persistent object within the scene until it is directly turned off. It can be moved around the scene by dragging it via mouse movements.

Initially forwards and backwards mouse movements equate to the cube being moved positively and negatively along the \( y \) axis, and similarly left and right equal positive and negative movement along the \( x \) axis. Left and right movement can be transmuted to equate to positive and negative \( z \) axis movement, thus providing full three dimensional freedom. An example of the cube brush style can be seen in figure A.12.

![Figure A.12 Three dimensional scatter plot within MVG showing Cube brushing](image-url)
A.3. Data Binning

As per the two dimensional scatter plot, the basis of all binning within the three dimensional scatter plot is the binning grid. Once enabled this provides a visual overlay of a red grid appropriate in size to the number of $x$, $y$ and $z$ cells requested. An example of a $5 \times 5 \times 5$ binning grid can be seen in figure A.13.

![Figure A.13 Three dimensional scatter plot within MVG with 5 x 5 x 5 binning grid enabled](image)

The arithmetic mean binning method seen in the two dimensional scatter plot is the only available option within the three dimensional scatter plot. The size and colour of each bubble can be altered as before. The result of applying the binning grid seen in figure A.13 can be viewed in figure A.14.
Figure A.13 Three dimensional scatter plot within MVG with 5 x 5 x 5 binning grid applied
A.4 Parallel Coordinates

The Parallel Coordinates visualization style provides the user with the following tools:

- General Visualization Options
- Data Brushing
- Data Binning

Due to the nature of Parallel Coordinates, all available variables are used to generate the initial visualization, with the ability to re-order or disable them following as well as full access to all tools. The initial view of a Parallel Coordinates visualization within MVG can be seen in figure A.14.

![Figure A.14 Parallel Coordinates within MVG](image)

A.4.1 General Options

Three general options exist within the Parallel Coordinates visualization. They are as follows:

- The ability to alter the colour scheme applied to the unbrushed lines
  - ‘No Colour’ leaves the lines their default blue (as per figure A.14).
- ‘Each Header’ applies a single colour to all points between each set of axis (as per figure A.15).
- ‘Each Row’ applies a different colour to every set of lines that represent a single row of data, spanning through the RGB spectrum from red to blue (As per figure A.16).

**Figure A.15 Parallel Coordinates within MVG with ‘Per Header’ colour scheme**

**Figure A.16 Parallel Coordinates within MVG with ‘Per Row’ colour scheme**
• The ability to alter the line width, between the default value of ‘0.5’ through to a maximum value of ’10.0’.

[Figure A.17 Parallel Coordinates within MVG with line width set to 6.0]

• The ability to alter the transparency of each set of lines, with the first rendered being set at 100% and the final set at 30%.

[Figure A.18 Parallel Coordinates within MVG with line width set to 6.0 and transparency graduation enabled]
A.4.2 Data Brushing

The brush shapes and available within the Parallel Coordinates are identical to those within the two dimensional scatter plot with the omission of the circular brush. Brushing group functionality is also identical to the two and three dimensional scatter plot.

Within the Parallel Coordinates, brushing a point on one variable will result in the whole set of variables associated with that point being brushed also. This can be altered so that only the currently selected variable is brushed. An example of this difference can be seen in figure A.19.

Figure A.19 Parallel Coordinates within MVG showing whole and single variable brushing
A.4.3 Data Binning

As before, the binning grid concept is the basis for all binning within the Parallel Coordinates visual style. The concept is notably different to that of the scatter plots however as the grid is only ever a one dimensional stack of bins, repeated as many times as there are variables. An example of a 10 bin grid can be seen in figure A.20.

![Figure A.20 Parallel Coordinates within MVG with a 10 strong binning grid enabled](image)

The only available binning option within the Parallel Coordinates is the Density Map method. As before, it is possible to alter between white or black representing the highest bin count. An example of the results of applying the grid seen in figure A.20 can be seen in figure A.21 with both white and black set as the high value.
Figure A.21 Parallel Coordinates within MVG with a 10 strong binning grid applied with black then white set as the high value
A.5 Star Glyphs

The Star Glyph visualization style provides the user with the following tools:

- Data Brushing

Due to the nature of Star Glyphs, all available variables are used to generate the initial visualization, with the ability to re-order or disable them following as well as full access to all tools. The initial view of a Star Glyph visualization within MVG can be seen in figure A.22.

![Figure A.22 Star Glyphs within MVG](image)

A.5.1 Data Brushing

Only the rectangular brushing shape is available within the Star Glyph visualization. None of the others used in MVG are applicable due to the uniform grid like nature of each Glyph. Brushing group functionality is identical to all other visualization styles.
Brushing a single Glyph is representative of brushing one whole row of data. An example of this can be seen in figure A.23

Figure A.23 Star Glyphs within MVG with brushing applied
A.6 Data Grid

MVG provides the ability to view all currently loaded data as well as refined subsets of data in a raw numerical format. It also provides the ability to output new CSV files containing the currently visible data grid. Brushing is reflected in the data grid through the use of cell background colour. An example of the initial view offered by the data grid can be seen in figure A.24.

![Data Grid within MVG](image)

Figure A.24 Data Grid within MVG

A.6.1 Brushed Data

Brushed data can be viewed within the data grid as well as refined by brushing group for export purposes. An example of six non consecutive rows of brushed data, contained in two separate brushing groups, can be seen in figure A.25.
A.6.2 Binned or Clustered Data

The results of applying a binning algorithm or the members of a cluster can also be viewed and exported via the data grid view. Selection of these data sub groups can be made via the drop down list at the top of the window or alternatively reached directly by pressing the button marked ‘View Data’ in each respective visualization. An example of
A.7 Linked Interaction

Brushing actions employed in one visualization are instantly reflected in all that are currently open. Similarly, all brushing is persistent until a new dataset is opened, the application is closed or the ‘Clear Brushing’ button on the main windows top menu bar is pressed.

An example of this can be seen in figure A.26 where all four visualizations can be seen side by side, two sets of brushing using two different groups are then applied. The first via the two dimensional scatter plot and the second via the Parallel Coordinates visualization. The results can be seen at the bottom of the figure.
Figure A.26 All four visualizations within MVG, before and after two brushing groups have been applied via different visualizations
8.0 Appendix B: OpenGL Geometry Lists within MVG

Two distinct methods of rendering exist within OpenGL. Either frame by frame, as is most common with animated scenes. For more static scenes, it is possible to utilise OpenGL geometry lists to pre compile each element in main memory before it is actually rendered. This also means a continuously active rendering loop is not required, therefore once the display lists have been generated and rendered, the CPU usage of the application will be negligible.

GL Lists can take one of two forms, either a pre compilation whereby each ‘glBegin’ command that instigates the formation of a new OpenGL structure is compiled in memory and remains there until it is explicitly removed using a ‘glDeleteLists’ command or in an object orientated environment, the lists associated object is destroyed. Once compiled, each list can be called as many times as required via the ‘glCallList’ command. Through the use of GL Lists, once the compilation process for a visualization is complete, user actions such as interaction, alteration of currently visible components and even scaling or translation can all be performed without the high CPU usage caused by the calculation of the scenes geometry.

Creation of a GL List is a relatively expensive operation computationally, therefore in a scene where a list is going to have a very short useful lifespan, it may be prudent to directly render each structure in real time. Within MVG there exist many elements of each scene that do not change from creation to destruction of the child window. If we use the two dimensional scatter plot as an example, clearly elements such as the axes, grid and labelling are always going to be static bar scaling or translation commands. Similarly the points themselves will rarely change, only needing recompilation when the brushing functionality is used or the visibility of a data group is altered. This methodology is applied to all available visualizations within MVG.

Figure B.1 shows an example of a two dimensional scatter plot within MVG. The four distinct display lists have been highlighted and numbered in red for identification purposes. An explanation of each numbered list follows the figure.
Figure B.1 Breakdown of the two dimensional scatter plots GL Lists

1) The axis GL List, consisting of two ‘GL_LINES’ and two ‘GL_TRIANGLES’.
2) The label GL List, consisting of FREEGLUT ‘glutStrokeString’ renderings.
3) The background grid, consisting of \( n \) ‘GL_LINES’.
4) The points themselves, consisting of \( n \) ‘GL_LINES’.

Through the use of GL Lists it is possible to compile three out of the four rendered elements only once. List four is only re compiled as and when is necessary.

The main drawback of utilising the GL list method of rendering becomes apparent when a very large dataset is loaded. GL Lists can only be as large as the current systems main memory allows, therefore if the geometry stored in the list causes an overflow, the entire list is simply destroyed and an OpenGL error thrown. It is possible to catch this error and inform the user the list did not generate correctly but impossible to counteract the issue without breaking the data down into smaller pieces.
9.0 Appendix C: Standardised Test Case Questionnaires

Questionnaire 1: Mr. Peter Hoornaert

- This section is about the 'Multi View' aspect of 'MVG.' Thinking specifically about how you would use 'MVG' please rate the following items accordingly.

<table>
<thead>
<tr>
<th></th>
<th>Disagree Entirely</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Agree Entirely</th>
<th>No Comment / Unsure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Being able to view the same data, in different ways but at the same time is a useful concept.</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>X</td>
<td>□</td>
</tr>
<tr>
<td>Interacting with one visualization and this being visually reflected in the others is a useful concept.</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>X</td>
<td>□</td>
</tr>
<tr>
<td>Aspects other than 'Brushing' should also automatically be reflected in all other visualizations.</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td></td>
<td>□</td>
</tr>
<tr>
<td>Visualizations work better as individual entities, what one shows is rarely applicable to another. Consequently visually linking them is a futile exercise.</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td></td>
<td>□</td>
</tr>
</tbody>
</table>

- Please enter a short narrative of your own thoughts / musings about the 'Multi View' aspect of 'MVG'. Please be as detailed as you feel is reasonable, these comments are likely to be used in citations and quotes in the body of the dissertation to backup any claims made.

The visualisations presented gave a different view from those I have previously worked on and this was useful in encouraging me to think about my data from a different viewpoint.
- This section is about the 'Brushing' aspect of 'MVG' Where brushing refers to the data selection concepts seen in 'MVG'.

Thinking specifically about how you would use 'MVG' please rate the following items accordingly.

<table>
<thead>
<tr>
<th></th>
<th>Disagree Entirely</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Agree Entirely</th>
<th>No Comment / Unsure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Being able to specifically identify data visually is a useful concept.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The concept of 'Brushing Groups' (Multiple selection colours with identifiable names specified for each) is useful.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The default brush shape (simple rectangle) is the simplest brush that is most likely to be used in most situations.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The alternative brush styles, such as the quadrilateral shape, offer the ability to identify very specific areas of data relatively easily.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Numerical control over the data selection (defining numerical boundaries and the relevant data is automatically selected) would be a better solution than visual selection in more situations than not.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>'Brush group Hiding' (the ability to hide or show specific brushed groups) is a useful concept.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Being able to exclude or include specific brushed areas from other actions such as binning or clustering is a useful concept.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
• Please enter a short narrative of your own thoughts / musings about the 'Brushing' aspect of 'MVG'. Please be as detailed as you feel is reasonable, these comments are likely to be used in citations and quotes in the body of the dissertation to backup any claims made.

I would like to explore this brushing technique further as it is very similar to the removal of data I have previously explored but this treats it visually.

• This section is about the 'Binning' aspect of 'MVG' where binning refers to the different ways compacted visual representations of the original dataset can be generated within 'MVG.'

Thinking specifically about how you would use 'MVG' please rate the following items accordingly.

<table>
<thead>
<tr>
<th></th>
<th>Disagree Entirely</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Agree Entirely</th>
<th>No Comment / Unsure</th>
</tr>
</thead>
<tbody>
<tr>
<td>The grid based nature of MVG's binning is easy to understand and conceptualise.</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☒</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>The grid based nature of MVG's binning is a sensible approach to the problem of data clutter.</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☒</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Taking an arithmetic mean of the points that lie within each grid and then showing a new point based on that average produces a good, less cluttered, representation of the original data.</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☒</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Being able to manually input each cell of the 'Grid' would increase the usefulness of the grid based binning concept (i.e. different sized cells placed only where you want to bin data.)</td>
<td>☐</td>
<td>☐</td>
<td>☒</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
</tbody>
</table>
The visual cues used in the arithmetic binning method, of colour and representative size of the new points, makes the result easier to understand (please turn these features on and off and back on again for insight.)

The 'Density Map' binning method provides unseen insight into cluttered data, be it in the 2D scatter or Parallel Coordinates visual styles.

- Please enter a short narrative of your own thoughts / musings about the 'Binning' aspects of 'MVG'. Please be as detailed as you feel is reasonable, these comments are likely to be used in citations and quotes in the body of the dissertation to backup any claims made.

None

- This section is about the 'Clustering' aspect of 'MVG' Where clustering refers to the ability to highlight specific points based on a visually defined set of parameters.

Thinking specifically about how you would use 'MVG' please rate the following items accordingly.

<table>
<thead>
<tr>
<th></th>
<th>Disagree Entirely</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Agree Entirely</th>
<th>No Comment / Unsure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normally clustering would be done entirely automatically based on input parameters. MVG requires these parameters to be entered visually. This method of input gives greater insight into what the algorithm in MVG is going to do before it happens.</td>
<td>☐</td>
<td>☐</td>
<td>☑️</td>
<td>☒</td>
<td>☒</td>
<td>☒</td>
</tr>
</tbody>
</table>
Being able to define your own start point (Centroid) for the clustering is more useful than automated techniques such as K-means, which define the centroids for you in order to find the general clusters within the data.

The ability to 'Brush' an area of data, hide the rest and then perform clustering on that area alone is a powerful and useful concept.

Defining the algorithm parameters numerically rather than visually (centroid placement done by numbers not a click, the Euclidean distance defined purely numerically with no visual cues) would be more useful in more situations than not.

- Please enter a short narrative of your own thoughts / musings about the 'Clustering' aspects of 'MVG'. Please be as detailed as you feel is reasonable, these comments are likely to be used in citations and quotes in the body of the dissertation to backup any claims made.

None

- Finally, if there are any other general comments or thoughts you would like to add, please use the text box below to do so.

I think the software you have produced would be useful for the dataset I was analysing in that it would allow me to investigate further a subset of the data and be able to study that further. Your application handles visually what I tried to do with the cylinder idea, and the points that lie within it.

I think your application is particularly useful in that it visually represents the data in different ways from those I was looking at, I think this would encourage me to look at the data afresh. I can't say where this might lead me as I haven't had enough time to look again at your application and then to reflect on it, but it certainly has made me think in some new ways, which is excellent.
Questionnaire 2: Dr. Samantha Mills

- This section is about the 'Multi View' aspect of 'MVG.' Thinking specifically about how *you* would use 'MVG' please rate the following items accordingly.

<table>
<thead>
<tr>
<th></th>
<th>Disagree Entirely</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Agree Entirely</th>
<th>No Comment / Unsure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Being able to view the same data, in different ways but at the same time is a useful concept.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Interacting with one visualization and this being visually reflected in the others is a useful concept.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Aspects other than 'Brushing' should also automatically be reflected in all other visualizations.</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Visualizations work better as individual entities, what one shows is rarely applicable to another. Consequently visually linking them is a futile exercise.</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Please enter a short narrative of your own thoughts / musings about the 'Multi View' aspect of 'MVG'. Please be as detailed as you feel is reasonable, these comments are likely to be used in citations and quotes in the body of the dissertation to backup any claims made.

*When assessing a dataset containing multiple variables, it was particularly useful to be able to see both the Parallel Coordinates and Star Glyphs simultaneously.*

- This section is about the 'Brushing' aspect of 'MVG' Where brushing refers to the data selection concepts seen in 'MVG'.

Thinking specifically about how *you* would use 'MVG' please rate the following items accordingly.
<table>
<thead>
<tr>
<th></th>
<th>Disagree Entirely</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Agree Entirely</th>
<th>No Comment / Unsure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Being able to specifically identify data visually is a useful concept.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The concept of 'Brushing Groups' (Multiple selection colours with identifiable names specified for each) is useful.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The default brush shape (simple rectangle) is the simplest brush that is most likely to be used in most situations.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The alternative brush styles, such as the quadrilateral shape, offer the ability to identify very specific areas of data relatively easily.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Numerical control over the data selection (defining numerical boundaries and the relevant data is automatically selected) would be a better solution than visual selection in more situations than not.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>'Brush group Hiding' (the ability to hide or show specific brushed groups) is a useful concept.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Being able to exclude or include specific brushed areas from other actions such as binning or clustering is a useful concept.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix C

- Please enter a short narrative of your own thoughts / musings about the 'Brushing' aspect of 'MVG'. Please be as detailed as you feel is reasonable, these comments are likely to be used in citations and quotes in the body of the dissertation to backup any claims made.

Regarding my own imaging data set it is particularly useful for comparing individual tumours where multiple variables have been measured e.g. perfusion, permeability, enhancement, volume, T1 signal, T2 signal etc. By brushing the data to select different tumour types this provides a clear visualisation of the behaviour of these parameters with regard to different tumours.

- This section is about the 'Binning' aspect of 'MVG' where binning refers to the different ways compacted visual representations of the original dataset can be generated within 'MVG'.

Thinking specifically about how you would use 'MVG' please rate the following items accordingly.

<table>
<thead>
<tr>
<th></th>
<th>Disagree Entirely</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Agree Entirely</th>
<th>No Comment / Unsure</th>
</tr>
</thead>
<tbody>
<tr>
<td>The grid based nature of MVG's binning is easy to understand and conceptualise.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The grid based nature of MVG's binning is a sensible approach to the problem of data clutter.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Taking an arithmetic mean of the points that lie within each grid and then showing a new point based on that average produces a good, less cluttered, representation of the original data.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Being able to manually input each cell of the 'Grid' would increase the usefulness of the grid based binning concept (i.e. different sized cells placed only where you want to bin data.)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The visual cues used in the arithmetic binning method, of colour and representative size of the new points, makes the result easier to understand (please turn these features on and off and back on again for insight.)

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>

The 'Density Map' binning method provides unseen insight into cluttered data, be it in the 2D scatter or Parallel Coordinates visual styles.

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>

- Please enter a short narrative of your own thoughts / musings about the 'Binning' aspects of 'MVG'. Please be as detailed as you feel is reasonable, these comments are likely to be used in citations and quotes in the body of the dissertation to backup any claims made.

*The ability to bin data and obtain a density map is particularly useful in some of the voxel-by-voxel image analysis where large quantities of data are overlapping and may not be clearly seen on conventional 2D plots.*

- This section is about the 'Clustering' aspect of 'MVG' where clustering refers to the ability to highlight specific points based on a visually defined set of parameters.

Thinking specifically about how *you* would use 'MVG' please rate the following items accordingly.

<table>
<thead>
<tr>
<th></th>
<th>Disagree Entirely</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Agree Entirely</th>
<th>No Comment / Unsure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normally clustering would be done entirely automatically based on input parameters. MVG requires these parameters to be entered visually. This method of input gives greater insight into what the algorithm in MVG is going to do before it happens.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Being able to define your own start point (Centroid) for the clustering is more useful than automated techniques such as K-means, which define the centroids for you in order to find the general clusters within the data.

The ability to 'Brush' an area of data, hide the rest and then perform clustering on that area alone is a powerful and useful concept.

Defining the algorithm parameters numerically rather than visually (centroid placement done by numbers not a click, the Euclidean distance defined purely numerically with no visual cues) would be more useful in more situations than not.

- Please enter a short narrative of your own thoughts / musings about the 'Clustering' aspects of 'MVG'. Please be as detailed as you feel is reasonable, these comments are likely to be used in citations and quotes in the body of the dissertation to backup any claims made.

   At present I personally have no real use for the clustering tool. However, I have passed the software on to a colleague who is particularly interested in this aspect.

- Finally, if there are any other general comments or thoughts you would like to add, please use the text box below to do so.

   Being able to view a set of data in a variety of manners simultaneously is very useful in gaining further understanding into what the data means.
Questionnaire 3: Louise Lever

- This section is about the 'Multi View' aspect of 'MVG.' Thinking specifically about how you would use 'MVG' please rate the following items accordingly.

<table>
<thead>
<tr>
<th></th>
<th>Disagree Entirely</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Agree Entirely</th>
<th>No Comment / Unsure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Being able to view the same data, in different ways but at the same time is a useful concept.</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
</tr>
<tr>
<td>Interacting with one visualization and this being visually reflected in the others is a useful concept.</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
</tr>
<tr>
<td>Aspects other than 'Brushing' should also automatically be reflected in all other visualizations.</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
</tr>
<tr>
<td>Visualizations work better as individual entities, what one shows is rarely applicable to another. Consequently visually linking them is a futile exercise.</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
</tr>
</tbody>
</table>

- Please enter a short narrative of your own thoughts / musings about the 'Multi View' aspect of 'MVG'. Please be as detailed as you feel is reasonable, these comments are likely to be used in citations and quotes in the body of the dissertation to backup any claims made.

Visualization is a means of conveying information to obtain understanding of the data. While prior knowledge of the data may prompt certain visualization techniques to be employed, the process of visualization should also be more interactive and focused towards discovery of new information. Multiple techniques and linked views therefore lends itself to advancing the identification of new patterns and behaviour in the data.

- This section is about the 'Brushing' aspect of 'MVG' Where brushing refers to the data selection concepts seen in 'MVG'.

Thinking specifically about how you would use 'MVG' please rate the following items accordingly.
<table>
<thead>
<tr>
<th></th>
<th>Disagree Entirely</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Agree Entirely</th>
<th>No Comment / Unsure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Being able to specifically identify data visually is a useful concept.</td>
<td>[x]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The concept of 'Brushing Groups' (Multiple selection colours with identifiable names specified for each) is useful.</td>
<td>[x]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The default brush shape (simple rectangle) is the simplest brush that is most likely to be used in most situations.</td>
<td>[x]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The alternative brush styles, such as the quadrilateral shape, offer the ability to identify very specific areas of data relatively easily.</td>
<td>[x]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Numerical control over the data selection (defining numerical boundaries and the relevant data is automatically selected) would be a better solution than visual selection in more situations than not.</td>
<td>[x]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>'Brush group Hiding' (the ability to hide or show specific brushed groups) is a useful concept.</td>
<td>[x]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Being able to exclude or include specific brushed areas from other actions such as binning or clustering is a useful concept.</td>
<td>[x]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Please enter a short narrative of your own thoughts / musings about the 'Brushing' aspect of 'MVG'. Please be as detailed as you feel is reasonable, these comments are likely to be used in citations and quotes in the body of the dissertation to backup any claims made.

The effectiveness of the brushing is greatly improved by the ability to hide other non-selected groups. Particularly with the Parallel Coordinates view, many overlapping lines can obscure the selected group. A useful addition would be a quick multiple group tool to define several (binned) brush groups over a range of values on a Parallel ordinate and represent with a spread of colours.

This section is about the 'Binning' aspect of 'MVG' where binning refers to the different ways compacted visual representations of the original dataset can be generated within 'MVG'.

Thinking specifically about how you would use 'MVG' please rate the following items accordingly.

<table>
<thead>
<tr>
<th></th>
<th>Disagree Entirely</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Agree Entirely</th>
<th>No Comment / Unsure</th>
</tr>
</thead>
<tbody>
<tr>
<td>The grid based nature of MVG's binning is easy to understand and conceptualise.</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>☒</td>
<td>□</td>
</tr>
<tr>
<td>The grid based nature of MVG's binning is a sensible approach to the problem of data clutter.</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>☒</td>
<td>□</td>
</tr>
<tr>
<td>Taking an arithmetic mean of the points that lie within each grid and then showing a new point based on that average produces a good, less cluttered, representation of the original data.</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>☒</td>
<td>□</td>
<td>□</td>
</tr>
<tr>
<td>Being able to manually input each cell of the 'Grid' would increase the usefulness of the grid based binning concept (i.e. different sized cells placed only where you want to bin data.)</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>☒</td>
<td>□</td>
</tr>
</tbody>
</table>
The visual cues used in the arithmetic binning method, of colour and representative size of the new points, makes the result easier to understand (please turn these features on and off and back on again for insight.)

The 'Density Map' binning method provides unseen insight into cluttered data, be it in the 2D scatter or Parallel Coordinates visual styles.

- Please enter a short narrative of your own thoughts / musings about the 'Binning' aspects of 'MVG'. Please be as detailed as you feel is reasonable, these comments are likely to be used in citations and quotes in the body of the dissertation to backup any claims made.

  This is a good way of reducing the clutter and conveying an overview of the data. User defined cells would be a useful addition, as would cells based on other scales e.g., logarithmic. Interactive refinement would help where one or more cells could be subdivided when required.

- This section is about the 'Clustering' aspect of 'MVG' where clustering refers to the ability to highlight specific points based on a visually defined set of parameters.

  Thinking specifically about how you would use 'MVG' please rate the following items accordingly.
<table>
<thead>
<tr>
<th></th>
<th>Disagree Entirely</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Agree Entirely</th>
<th>No Comment / Unsure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normally clustering would be done entirely automatically based on input parameters. MVG requires these parameters to be entered visually. This method of input gives greater insight into what the algorithm in MVG is going to do before it happens.</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>☑</td>
<td>□</td>
<td>□</td>
</tr>
<tr>
<td>Being able to define your own start point (Centroid) for the clustering is more useful than automated techniques such as K-means, which define the centroid’s for you in order to find the general clusters within the data.</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>☑</td>
</tr>
<tr>
<td>The ability to 'Brush' an area of data, hide the rest and then perform clustering on that area alone is a powerful and useful concept.</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>☑</td>
<td>□</td>
<td>□</td>
</tr>
<tr>
<td>Defining the algorithm parameters numerically rather than visually (centroid placement done by numbers not a click, the Euclidean distance defined purely numerically with no visual cues) would be more useful in more situations than not.</td>
<td>□</td>
<td>□</td>
<td>☑</td>
<td>□</td>
<td>□</td>
<td>□</td>
</tr>
</tbody>
</table>
Appendix C

- Please enter a short narrative of your own thoughts / musings about the 'Clustering' aspects of 'MVG'. Please be as detailed as you feel is reasonable, these comments are likely to be used in citations and quotes in the body of the dissertation to backup any claims made.

  Did not use the clustering tools enough to comment.

- Finally, if there are any other general comments or thoughts you would like to add, please use the text box below to do so.

  The ability to link entries in the data based on matching variables would be useful for seeing patterns in the data and also help reduce clutter. Non-linked variables then become parallel ordinates in their own right with filters applied to select from the linked entries. E.g. variables A, B, C, D where A is 0 or 1, plot parallel coordinates as A, B, C, D (A=0), D (A=1).