Visualization Techniques for the Analysis of Neurophysiological Data

by

Martin Alan Walter

A thesis submitted to the University of Plymouth
in partial fulfillment for the degree of

Doctor of Philosophy

School of Computing, Communications and Electronics
Faculty of Technology

December 2004

Abstract

In order to understand the diverse and complex functions of the Human brain, the temporal relationships of vast quantities of multi-dimensional spike train data must be analysed. A number of statistical methods already exist to analyse these relationships. However, as a result of expansions in recording capability hundreds of spike trains must now be analysed simultaneously.

In addition to the requirements for new statistical analysis methods, the need for more efficient data representation is paramount. The computer science field of Information Visualization is specifically aimed at producing effective representations of large and complex datasets. This thesis is based on the assumption that data analysis can be significantly improved by the application of Information Visualization principles and techniques.

This thesis discusses the discipline of Information Visualization, within the wider context of visualization. It also presents some introductory neurophysiology focusing on the analysis of multi-dimensional spike train data and software currently available to support this problem. Following this, the Toolbox developed to support the analysis of these datasets is presented. Subsequently, three case studies using the Toolbox are described. The first case study was conducted on a known dataset in order to gain experience of using these methods. The second and third case studies were conducted on blind datasets and both of these yielded compelling results.

Contents

List of Figures

Acknowledgments

I would like to take this opportunity to express my heart felt thanks and gratitude to my supervisor, Doctor Liz Stuart, and my studies advisor, Professor Roman Borisyuk for their support during my studies and the considerable time they have invested in this work.

I would like to thank my Mother, Father and Brother and all my family for their continuing encouragement and support throughout my research.

Thanks is also due to the staff and research students of the School of Computing (and later the School of Computing, Communications and Electronics) at the University of Plymouth who create a stimulating and enjoyable work environment. The school consists of too many amazing people to mention, but special thanks goes to Linda Lanyon, Theocharis Kyriacou, Richard Holden, Rohana Rajapakse, and Andrew Hannell.

I also thank my friends Suchindra-Kumar Reddy, Robert Everrit, Chris Hull, Marion Miramo, Catherine Wolstencroft for putting up with me during this period. Also I wish to thank Marian Haynes for her help, support and assistance.

This research was funded by the Engineering and Physical Sciences Research Council.

Finally, I am indebted to Dennis Hockings for his extraordinary support and encouragement, without whom I would not have made it. Thank you all.

Author's Declaration

At no time during the registration of the degree of Doctor of Philosophy has the author been registered for any other University award without the prior agreement of the Graduate Committee.

This study was financed with the aid of a studentship from the Engineering and Physical Sciences Research Council (EPSRC).

Relevant scientific seminars and conferences were regularly attended at which work was presented and several papers were published.

Publications:

Walter, M., Stuart, L. and Borisyuk, R., **The Representation of Neural Data using Visualization**, *Accepted for publication in Journal of Information Visualization, (2004).*

Stuart, L., Walter, M. and Borisyuk, R., **The Correlation Grid: Analysis of Synchronous Spiking in Multi-dimensional Spike Train Data and Identification of Feasible Connection Architectures**, *Accepted for publication in BioSystems Journal (2004).*

Walter, M., Stuart, L. and Borisyuk, R., **Evaluation of Spike Train Analysis using Visualization**, Presented at the 9th IEEE International Conference on Information Visualization, InfoVis2003 (interactive conference poster), Seattle, (2003).

Stuart, L., Walter, M. and Borisyuk, R., **Analysis of Visualisation of multi-dimensional Spike Trains using Visualisation**, In Proceedings of the 5th International Workshop on Neural Coding'2003, Aulla, (2003).

Walter, M., Stuart, L. and Borisyuk, R., **Spike Train Correlation Visualization**, In Proceedings of the 7th IEEE International Conference on Information Visualization, IV03 (2003).

Stuart, L., Walter, M. and Borisyuk, R., **Visualisation of Neurophsyiological Data**, Presented at the 8th IEEE International Conference on Information Visualization, InfoVis2002 (2002).

Stuart, L., Walter, M. and Borisyuk, R., **Visualisation of Synchronous Firing in Multi-dimensional Spike Trains**, In BioSystems, 67, 265-279, (2001).

Stuart, L., Walter, M. and Borisyuk, R., **Visualisation of Interspike Association**, Proceedings of 4th International workshop Neural Coding'2001, 47-48, (2001).

Conferences Attended:

4th International workshop Neural Coding in 2001.
5th IEEE International Conference on Information Visualization, IV01 in 2001.
6th IEEE International Conference on Information Visualization, IV02 in 2002.
7th IEEE International Conference on Information Visualization, IV03 in 2003.
9th IEEE International Conference on Information Visualization, InfoVis2003 in 2003.

Word count of the main body of this thesis: 33500

Signed: ....

Date: 11\5\2005

Chapter 1

Introduction

The human brain consists of in excess of 10^{12} neurons. These neurons communicate with one another via short (1ms) electrical signals (spikes) forming neuronal structures; and by doing so, provides humans with incredible information processing capability. A capability for learning, memorising, reasoning, and the experience of emotion, consciousness and self awareness. However, the scientific explanations of these key facets of the human brain are still partial; how does the brain work?

The desire of scientists to understand how the brain functions is as strong as their desire to understand the function of other major organs. However, the very complexity that makes the brain so amazing also means that the feat of understanding how it functions is monumental.

For decades scientists have had the ability to record the electrical activity of the brain; more recently the ability to record the activity of specific neurons. These recordings, known as spike trains, show the activity of a neuron over time, they are a record of the spikes generated by that neuron. A number of theories have been proposed on how information is processed and transmitted in the brain. The principle of temporal coding (ascribed to in this research) proposes that information is transmitted in the temporal patterns of spikes from a given neuron and the correlation between the spikes of connected neurons.

In recent years the precision with which experimental recording could be achieved, has dramatically increased. Currently, hundreds of individual neurons can be recorded simultaneously[WM93, BKM04]. Furthermore, it is these multi-dimensional spike train recordings that are believed to hold the key to many questions regarding brain functions. The identification of synchronicity between the spike trains in these recordings permits the functional relationship between the neurons to be explored. These relationships in turn aid in the understanding of how information is stored and processed within the brain.

Despite the dramatic developments in recording technology, the development of methods to analyse these vast datasets has been relatively slow. In particular, the analysis of synchronous spiking between neurons is believed to be very important to understanding many of the brain's functions.

Traditionally, methods of analysis are based on pairs of spike trains, such as cross-correlation [Bri79, AG85]. These pair-wise methods do not scale well. For example, the analysis of a recording of 20 simultaneous spike trains would result in 190 unique cross-correlations. Each of these cross-correlations requires analysis for significant results and then comparison with all others to identify trends.

Some methods exist for the analysis of complete assemblies of neurons, such as the Gravity Transform [GA85]. However, these techniques also create large quantities of resultant data that must be studied to extract information regarding the temporal relationships between the spike trains.

In 1987 McCormick et al. [MDB87] identified the need for visualization support in scientific computing. This corner stone report led to a new research discipline called Information Visualization (IV). The key principle of the Information Visualization field is to provide users with tools to effectively represent, interact and explore their data, in order to aid the extraction of information and understanding.

With the large quantities of data involved in the analysis of the temporal relationships within multidimensional spike train datasets, it is clear that techniques from the field of Information Visualization could provide support. In particular this research identifies the needs of Neuroscientists (with regard to this analysis) and reports on a number of analysis and display methods.

1.1 Research Achievements

This thesis contributes to the Visualization of Inter-Spike Association project, within the Visualization Lab, at the University of Plymouth, under the direction of Dr L Stuart.

This research consists of the development, testing and refinement of a number of analysis and visualization techniques. Furthermore, it includes the empirical testing of these methods, with both known and unknown (blind) datasets.

The results of this empirical testing have been evaluated by a small user group; consisting of the author, Supervisor and Studies advisor (an eminent neuroscientist). In addition, the results have been published at relevant conferences, such as the Neural Coding Symposium, where they have received positive feedback.

The datasets used for testing consisted of simulated spike trains; generated using a well respected Enhanced Integrate and Fire model neuronal generator[Bor02]. The neuronal assemblies for these datasets were designed by the neuroscientist in the user group and included common structures found in the brain. Moreover, these assemblies contain structures that other analysis methods have considerable difficult in identifying.

This research has led to the publication of a number of referred papers, enclosed as appendices to this thesis.

1.2 Thesis Structure

This thesis demonstrates the usefulness of analysis tools based on the principles of Information Visualization to aid scientists and investigators in the analysis of synchronous firing within multi-dimensional spike train datasets.

Chapter two presents a broad overview of the field of Visualization. In particular, chapter two concentrates on the exploitation of the human visual system to aid in understanding complex datasets. In addition, the field of Visualization is explored; including Human Computer Interaction, Computer Graphics and Virtual Reality.

Chapter three presents an overview of the field of Information Visualization. This chapter presents a discussion of the principles of IV and the design issues involved. These include Shneiderman's Mantra[Shn96], user interaction[PRS$^+$94, PRS02] and multiple views[Rob03]. Subsequently, it reinforces Information Visualization as the main focus of this thesis.

Chapter four introduces the problem domain of Neurophysiological data analysis. Moreover, it describes the investigation of synchronous spiking within multi-dimensional spike train datasets to extract information regarding the functional relationships of the neurons. The use of Information Visualization to aid this analysis is recommended and the requirements for software support are presented.

Chapter five describes a number of IV techniques which have been developed to start the process of meeting the requirements identified in chapter four.

Chapter six presents the results of three case studies. Each of these case studies demonstrates the usefulness of the techniques proposed in chapter five and the power of combining the results of the techniques together.

Chapter seven draws conclusions regarding the methods proposed. In addition, extensions and further developments to the methods are presented. Finally, new research directions are proposed.

Chapter 2

Visualization

"visual adj. Of or used in seeing."

"visualize v. (also -ise)(-zing or -sing) imagine visually."

Summary

In this chapter the term Visualization is defined and the general scope of the field is discussed. In addition, several key historic examples are discussed, such as Minard's map and Beck's London Underground map.

2.1 Overview

'A picture paints a thousand words', a quote that most people are familiar with and one that represents a key principle of Visualization. Since around 3200b.c. with the Sumerians and Egyptians, people have been representing information in a visual form. Visualization is the process by which the human brain forms a mental image of the information presented to it. This mental image is a perceived visual representation of the relationship between items of information. For example, when a person is presented with a set of directions, they will form a mental model when a set of directions is given to them, remembering key intersections between streets, as opposed to the full details of the directions. Thus, Visualization is a mental process; it is not a computer based method.

In the last two decades following the ViSC report [MDB87], with developments in computer graphics, there have been great strides made in the computer science field of Visualization. These advances have exploited the human ability to understand and glean information from data represented in a graphical form.

2.2 The Human Visual System

The human visual system has vast capability to scan, recognise and recall images quickly. Moreover, the system is capable of rapidly and 'automatically' recognising patterns and changes in the size, colour, shape, movement and texture of objects.

Text based data representation poses a large cognitive load on the user. When presented with textual information the user must first construct an internal mental representation (mental or cognitive map) of this data prior to attempting to interpret it.. By presenting data in a visual form the capabilities of the human visual system can be exploited. This visual representation aids the user by presenting the data in a form similar to their internal mental model.

The key aim of the field of visualization research is to move the load from the cognitive system to the human perceptual system[And00]. Thus, by exploiting the perceptual system, trends and relationships in data can be observed enabling the cognitive system to analyse the details.

2.3 The Usefulness of Visualization

The power of Visualization can be demonstrated by a number of historic examples:

1. Minard's Map

2. Nightingale's Roses

3. John Snow's graphical representation of an outbreak of Cholera

4. Harry Beck's map of the London Underground System

2.3.1 Minard's Map

It is possible to represent a number of data values using the geometric primitives of an object. These objects are known as Glyphs. Minard utilised glyphs to represent the progress of Napoleon's armies on their march to and retreat from Moscow, as shown in figure 2.1.

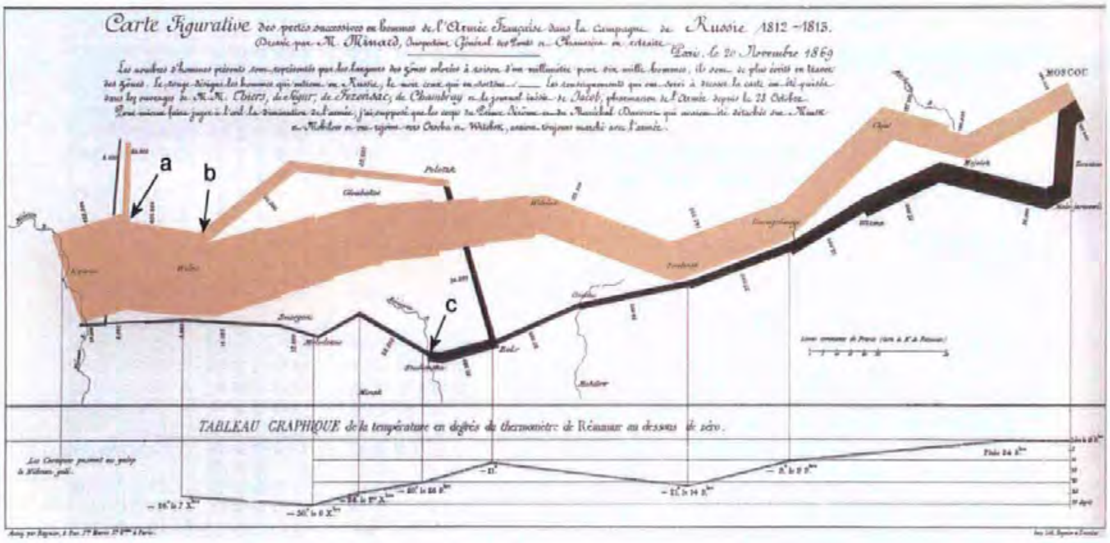


Figure 2.1: Minard's map of Napoleon's armies on their march to, and retreat from Mascow. The thickness of the line represents the number of soldiers present, the colour of the line represents the stage of the campaign (brown for attack, black for retreat)

Minard's map portrays the progress of the armies over the terrain. In addition, the size of the armies is reflected in the width of the line. The map further shows the effect of the falling temperature (plotted on the bottom of the map) on the retreating soldiers. Minard's simple representation enables a user to gain an understanding of the overall progress of the campaign.

First note the two different coloured paths on the map, the lighter path represents the armies march to Moscow (progressing from the left of the map) and the darker path represents their retreat (progressing from the right of the map). The width of the path is proportional to the number of soldiers present at that point. From this map it is possible to appreciate that only approximately 25% of the soldiers arrived at Moscow and that less than 5% returned home. In addition, it is possible to observe that a portion of the troops left the main body of the army to fight other battle, indicated by the separation of the lighter line at points 'a' and 'b' on the map. Finally, the catastrophic effect of crossing a freezing river is shown at point 'c'.

2.3.2 Nightingale's Roses

Florence Nightingale's roses show the presentation principle of object alignment, as shown in figure 2.2. The roses are histograms, tracking the change in death rate at field hospitals in the Crimea between 1^{st} October 1854 and 30^{th} September 1855.

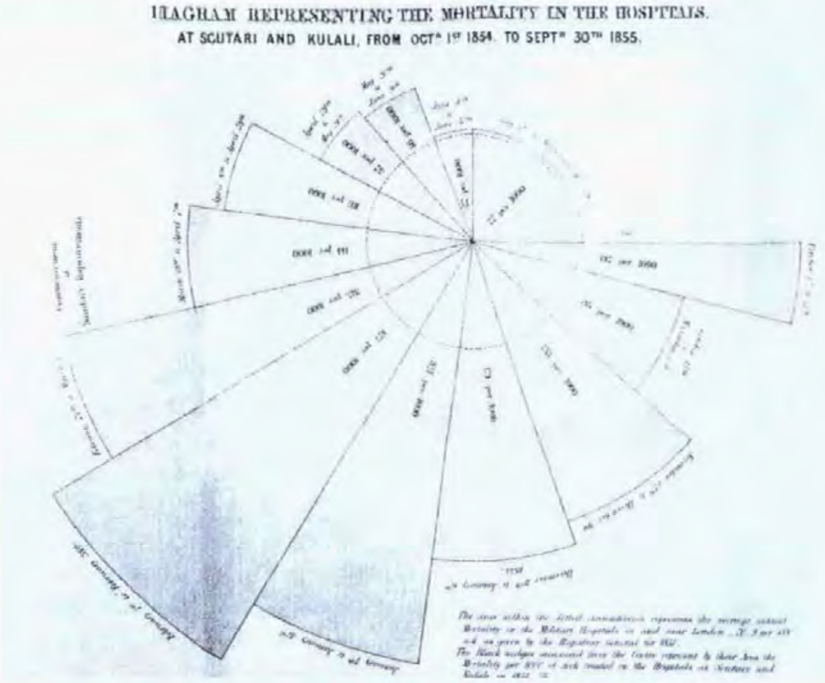


Figure 2.2: Nightingale's Roses: Florence Nightingale's diagram showing the death rate in field hospitals Crimea between 1^{st} October 1854 and 30^{th} September 1855

As opposed to the standard histogram layout, the roses align the segments around a centre point, similar to the petals of a flower. This alignment facilitates easier comparison of the segments.

From these Roses, Nightingale was able to argue that the introduction of the sanitary improvements resulted in a significant reduction of deaths in the hospital. This is illustrated by the reduction of the area of the 'wedges' representing the deaths for the months following the introduction; these are easily comparable due to their alignment. Moreover, it is possible to observe that the death rate falls to a level in line with that of hospitals within the British Isles, indicated by the circle.

2.3.3 John Snow's investigation of Cholera

In 1854 Dr. John Snow (medical officer for London) was investigating the cause of epidemic levels of Cholera in the Soho area of London. Snow obtained a list of the 83 related deaths from the General Register Office. From this data, a map of the deaths was constructed, as shown in figure 2.3. This map noted the address of each of the deceased (marked by single dots). In addition, the locations of the local water pumps were marked (with crosses).

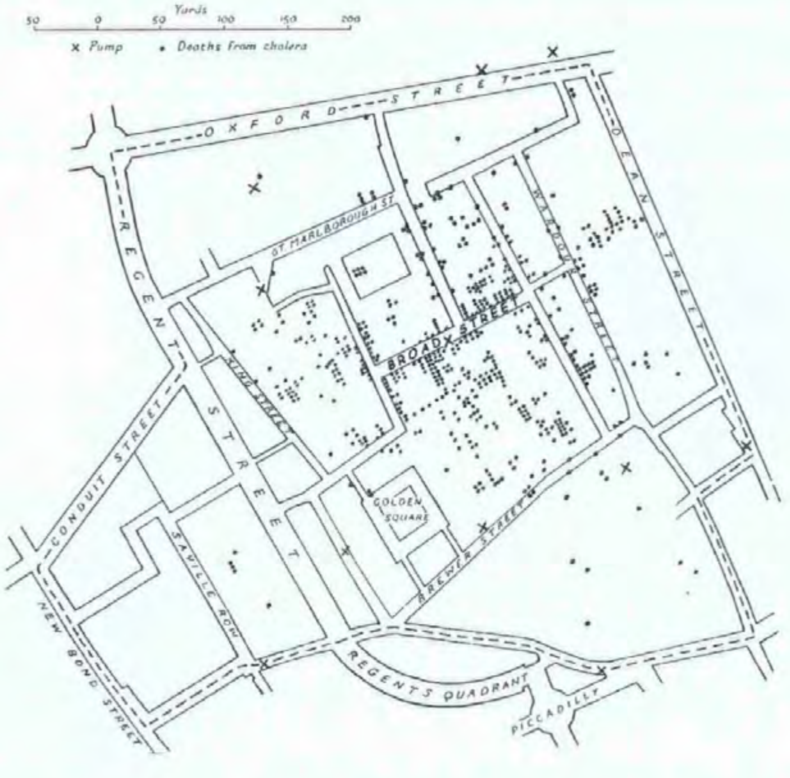


Figure 2.3: The map constructed by Snow, mapping the deaths due to an outbreak of Cholera in the Soho area of London in 1854. The address of each deceased person is marked by a dot; the local water pumps are marked by crosses.

From this map, Snow observed a significant clustering of deaths around one specific pump, the Broad Street pump. In addition, fewer deaths were evident among the workers at the Brewery and those in the Work House; note the people at these locations predominantly drank beer. Aided by this visualization, Snow concluded the source of the contamination was the Broad Street pump and subsequently had the handle removed to prevent its use. It is unclear how critical the removal of the handle was, as the epidemic was already in recession at the time. However, the power of this visualization is clear. Identifying this pattern of deaths from the Register's list would be possible, but it would also be very time consuming, and it would place a large cognitive load on the investigator [Tuf97].

2.3.4 Harry Beck's London Underground Map

The principle of building a mental map of information is demonstrated by the London Underground system. When travelling on the London Underground, it is common for a passenger to remember their journey as "three stops East on the Red line, then up the Black line for two stops". The passenger remembers the intersections of the lines rather than the names of the stations.

The process of building a mental map, of the London Underground system, is aided by the modern Tube Map. This map is based on the design developed by Harry Beck in 1933. Until Beck's map was adopted, all Tube maps had been based on the geography of London, showing the exact geographical position of all lines and stations, see figure 2.4.

Beck understood that the geographic position of each line was unimportant, as travel is restricted to predefined paths. Thus intersections between, and the general direction of these paths was sufficient information for the traveller. Beck proposed a simpler map based on the layout principles of electrical circuit diagrams, see figure 2.5.

Beck's map showed the general direction of each train line and the line's stations. However, the map does not represent the true geographical relationships between the stations. By producing this visual representation of the London Underground System, Beck was able to show the full extent of each line, something that was not achieved on the original maps. In addition, Beck was able to portray the information in a clear and concise form. This 1933 Tube map was so effective that the design of all

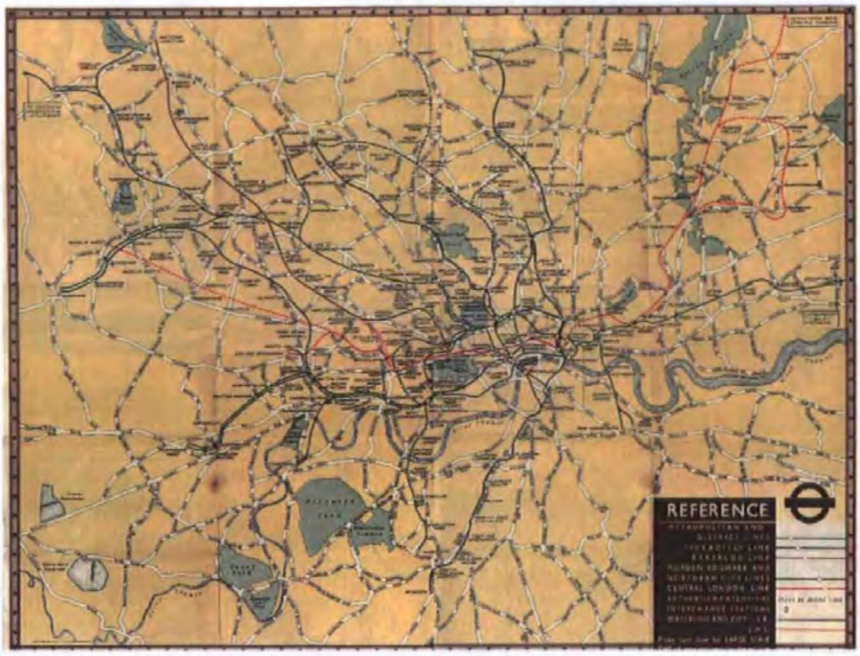


Figure 2.4: London Underground Map, circa. 1927

modern Tube maps have been based on it. Moreover, Beck's method has been utilised for a number of other transport systems such as bus routes and the metro system in the US.

2.4 The Field of Visualization

The requirement for efficient data representations is a long standing problem.

> *"...the single biggest problem we face is that of visualisation..." Richard P. Feynman (Los Alamos, 1945)*

In 1987, McCormick et al.[MDB87] highlighted the desirability of Visualization support for scientific computing. The field of Visualization consists of many disciplines including: image processing, ray tracing, volume visualization, scientific visualization, and information visualization. Moreover, the field draws on research and understanding from a number of other related disciplines. In particular, work in the fields of human computer interaction, computer graphics and virtual reality are all utilised by Visualization practitioners.

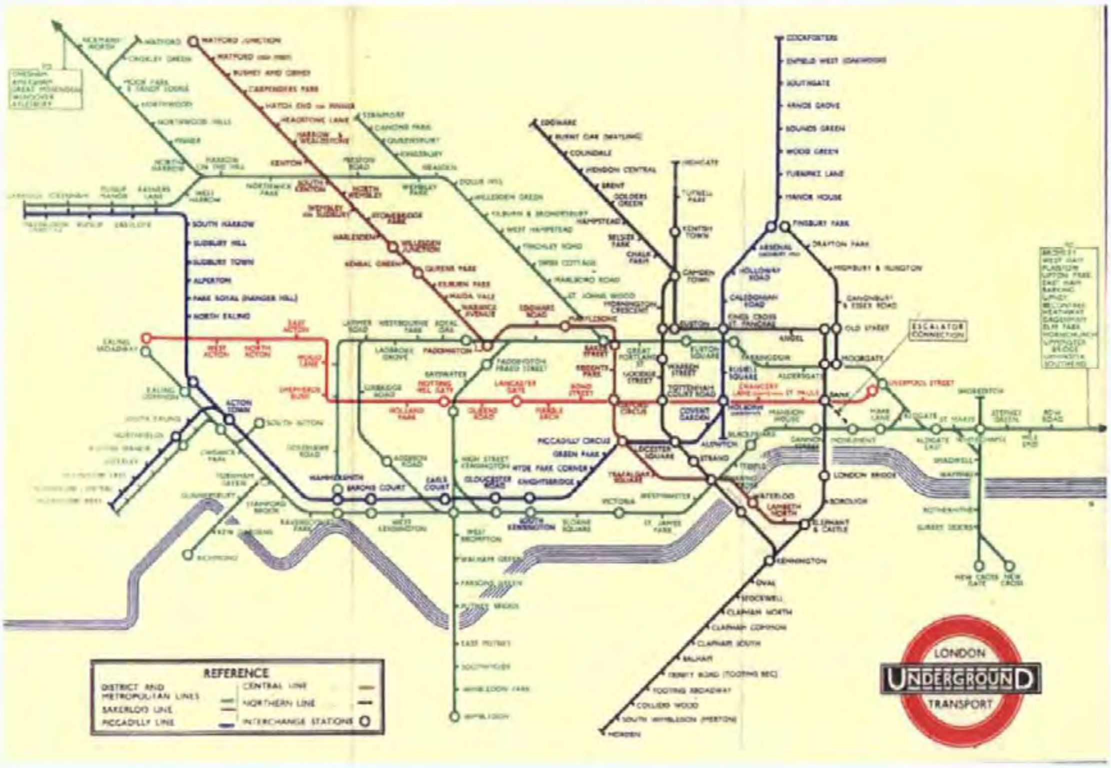


Figure 2.5: Harry Back's London Underground Map, 1933

2.4.1 Human Computer Interaction

The power of modern computer visualizations lies in the ability of the investigator to interact with the dataset. This interaction facilitates the tailoring of the visualization to the specific needs of the individual task. The design of an interface that supports efficient interaction with the visualization system is a complex process.

The Object-Action Interface (OAI) model [Shn98] attempts to model a user's interaction with a computer to perform a task as a set of actions on interface objects. These interface objects, also know as widgets, are metaphorical representations of real world objects involved in a specific task. The actions on the interface objects represent the real world process. These objects should provide feedback to the user to indicate their state.

The OAI model is based on producing an interaction process, on the computer, that is semantically similar to the real world process. The system designer must analyse the real world task and determine the decomposition of sub-tasks that the users have undertaken. This decomposition of tasks and objects should then be represented in the system.

A very common real world example of this is the process of disposing of a file. In this scenario the user may pick the file up and drop it into the rubbish bin. On the computer system, to achieve the same task, the user could drag a graphic representing the file into a graphical representation of a rubbish bin. Thus the user's semantic process is mirrored in the computer system using a different syntax.

Thus, HCI is an important aspect of Visualization. In order to develop an efficient Visualization, an understanding of the fundamentals of cognitive-psychology and Human Computer Interaction is required.

2.4.2 Graphics

The majority of current Visualizations exploit the power and capabilities of modern computer systems to represent data. Many of these visualizations rely upon the ability to efficiently generate graphical images on a computer display. The quality and level of detail of these images is continually growing, as graphics researchers develop more sophisticated systems[BBC$^+$03]. In addition, the end users demand an increasingly high quality of display.

In addition to high quality graphics, users require these systems to exhibit smooth animation and transition between graphics. Researchers in the field of Computer Graphics are concerned with the development of efficient hardware and software algorithms for generating and manipulating graphical images.

2.4.3 Virtual Reality

The origins of Virtual Reality Simulation date back to the 1930's with the development of basic flight simulators during the Second World War. These simulators consisted of an aircraft cockpit mounted on a movable platform, responding to the pilot's actions on the controls[Enc99].

The term Virtual Reality (VR) was coined in the 1980's by Jaron Lanier. Currently, the term has a number of different meanings, one useful definition being:

> *"Virtual Reality is a way for humans to visualize, manipulate and interact with computers and extremely complex data"*[AB92]

In addition to VR, there are now a number of other terms in common use, including: *Synthetic Environment*; *CyberSpace*; *Artificial Reality*; and *Simulator Technology*[Isd93].

Virtual Reality systems exist in four general forms:

1. Full immersion environments

   A full immersion environment separates the user from the real world and places them into a virtual environment that is fully encompassing. The environment responds to the users actions and provides feedback. The interaction methods used in the environment should be a close to real-world actions as possible.

2. Cab simulators

   Cab simulators are used to reproduce an environment where the user would interact with the real world via a set of controls and displays. For example, in a flight simulator the pilot will use real (or realistic) controls to fly the aircraft. The virtual environment will provide the feedback to the pilot's actions on the controls via a simulation of the outside world.

3. Projected virtual reality environments

   A projected environment allows one or more users to explore a virtual environment. The user is able to interact with the environment via a number to tools, designed to perform specific tasks. A projected environment does not provide a fully immersive feeling, as the user cannot directly interact with the virtual world.

4. Desktop virtual reality

   A desktop environment provides a user with a number of views into a virtual world through devices such as monitors. The user is able to interact with the virtual world using a number of controls, either using software or real devices.

VR can be used to create an environment in which to explore data [NCCN98]. Users can then explore and interact with their datasets on a real-time basis within this environment. VR hardware, such as data gloves and 3D mice, can be used to aid this interaction and to provide the user with feedback.

It has been argued that for a system to be classified as true Virtual Reality it must have:

> "*... response to user action, real-time 3-D graphics, and a sense of immersion ... An immersive experience is one so absorbing that you cease to notice your surroundings or "how you got there."...* " [PT95]

Augmented Reality

In addition to Virtual Reality, there is also the concept of Augmented Reality. Augmented Reality is where a view of the real world is overlaid with additional information. Augmented Reality has been used for a number of years in fighter aircraft, in the form of Head-Up-Displays (HUD's). A HUD is a transparent screen in the front of the cockpit, which displays tactical, navigation and status information about the aircraft. With training the HUD enables a pilot to monitor all the necessary information regarding the aircraft's status and any approaching dangers, without losing context of the real world.

2.4.4 Scientific Visualization

The field of Scientific Visualization is concerned with the representation of physical objects or datasets that have direct spatial representation. For example geographical data[JM95], ancient archaeological artefacts or cities and buildings[MMKM01].

Researchers in these fields are engaged in developing methods to facilitate greater understanding of the physical environment. These methods often involve modelling a part of the real-word and augmenting the display with relevant data and information. This segmented display can then be utilised by investigators to study the object in the system.

2.4.5 Information Visualization

In contrast to Scientific Visualization, the field of Information Visualization is concerned primarily with the representation of abstract data. For example, Stock Market share prices[CHLS03] or the comparison of large tree structure, such as version of a website [MGT$^+$03].

Researchers in the field of Information Visualization attempt to develop methods that permit abstract datasets to be viewed in a meaningful form. The key requirement of the discipline is to convert inherently non-visual data into a visual form. These representations can be visual, sonic (generated through sound), or haptic (created by touch).

Chapter Summary

In conclusion, cognition can be amplified by presenting information in a form that reinforces the observer's internal mental model. In addition, the speed with which anomalies and trends within datasets are identified can be dramatically improved by exploiting perceptual cues. These principles have been demonstrated using a number of historic example.

In the next chapter the field of Information Visualization is explored and the main design issues are discussed.

Chapter 3

Information Visualization

Summary

In this chapter the field of Information Visualization is discussed. The main design issues within field are discussed and Shneiderman's mantra is defined. In addition, some key Visualizations are presented, such as Cone Trees and Treemaps.

3.1 Overview

Researchers in the field of information visualization are engaged in developing support for the analysis of many different types of abstract data: for example, the flow of data in a communication network [net04], the content of a document [Palb] or micro array time series data [HBMS03].

In addition to these specialist tools, researchers are also engaged in the development of generic Information Visualization tools[MDH^{+}03].

3.2 Design issues of IV Systems

3.2.1 Shneiderman Mantra

The power of modern visualization lies in the ability of the user to interact with the dataset under investigation. A common framework for the design of these interactions is Shneiderman Mantra[Shn96]:

```
Overview, Zoom, Filter and Detail-on-Demand
Overview, Zoom, Filter and Detail-on-Demand
Overview, Zoom, Filter and Detail-on-Demand
Overview, Zoom, Filter and Detail-on-Demand
Overview, Zoom, Filter and Detail-on-Demand
Overview, Zoom, Filter and Detail-on-Demand
Overview, Zoom, Filter and Detail-on-Demand
Overview, Zoom, Filter and Detail-on-Demand
Overview, Zoom, Filter and Detail-on-Demand
Overview, Zoom, Filter and Detail-on-Demand
```

The mantra was quoted ten times in this paper[Shn96], once for each time Shneiderman had rediscovered the process when developing tools. Furthermore, Shneiderman defined the following *seven tasks for data visualisation*.

"**Overview**: Gain an overview of the entire collection". The overview permits the investigator to view the entire dataset. In addition, the distribution of data items can be ascertained and any cluster, or other 'interesting' patterns and/or anomalies noted for further investigation.

"**Zoom**: Zoom in on items of interest". Having identified an area of interest, the investigator must have the facility to zoom in on that area, to examine the data in greater detail. In addition, the investigator should be able to control the zoom factor and the rate at which zooming occurs. Smooth zooming is required for the investigator to maintain context with the dataset[vWN03].

"**Filter**: Filter out 'uninteresting' items". A key principle of Information Visualization is the ability to apply dynamic queries to the dataset, permitting the user to control the content of the display. These dynamic queries permit the investigator to quickly focus on items of interest by eliminating 'clutter'.

"**Details-on-demand**: Select an item or group of items and get details when needed". With the dataset zoomed and filtered the investigator should have the facility to easily obtain details about individual data items or groups of items.

"**Relate**: View relationships among items". Having selected a particular data item, the user should have the facility to modify the filters to show all items that match a specific attribute of the selected data item. This relationship filtering aids in the identification of similarity between items in a dataset.

"**History**: Maintain a history of actions to support undo, replay and progressive refinement". Commonly a number of filtering actions will be required to identify the data items of interest. By maintaining a history of actions, the system permits the user to retrace the process that led to the discovery. Moreover, the user can recover from 'poor' filtering decisions.

"**Extract**: Enable extraction of sub-collections and the query parameters". Having identified a subset of the dataset, the system should have the facility to export the underlying data, of that subset, to a separate data file. This file could then be used within other systems or sent to colleagues for discussion. In addition, an extraction system should exist to export the history log that produced the dataset.

These tasks are used as the basis for designing much of the current research in Information Visualization. However, these tasks should be viewed as a framework for the system and not the limit of the system's functionality. Different datasets will require additional, specialist tools for meaningful investigation to take place. Moreover, the specific techniques used for each task should be tailored to the individual system. Generally a number of options for each task should be made available to the investigator. The investigator can thus choose the most appropriate options for the task at hand.

3.2.2 Interaction

The main advantage of information visualization is the ability of the user to interact with the data being analysed, in order to gain greater understanding. However, interaction has become an increasingly difficult problem as information visualization datasets expand.

In 1986 Buxton[Bux86] highlighted the shortcomings of the human computer interface of the day. He stated that if a scientist were to 'backward engineer' the interface to see what a human looked like then:

> "... *[humans] would be pictured as having a well-developed eye, a long right arm, uniform-length fingers and a "low-fi" ear. But the dominating characteristic would be the prevalence of our visual system over our poorly developed manual dexterity...* " [Bux86]

This statement is equally applicable to modern computers, as the interface has developed little over the past 18 years.

The majority of computers even now only have the basic mouse and keyboard interface devices available to the user for input. In addition, the user is commonly restricted to using one input device at a time. Work by Buxton and Myers[BM86] explored the use of parallel device input to aid interaction. The experiment presented subjects with two independent input devices, a slider and a graphics tablet. The subjects were instructed on the operation of each device. However, no encouragement was given to subjects to use the devices in parallel. The results of the experiment showed that (averaged over all experiments) subjects were engaged in two-handed input for 40% of the time. In addition, 6 out of 14 subjects used two-handed input from the start, without instruction to do so. Moreover, two-handed input was found to be more efficient than the standard one-handed input.

With a 3D environment, a key problem faced by investigators is how to interact with that environment using inherently 2D tools such as a mouse. Work by Chen et al.[CMS88] and Oshiba and Tanaka[OT99] explored methods for manipulating 3D objects with 2D tools, experimenting with the layout and location of sliders used to manipulate the objects. The conventional arrangement of three sliders under the object, one to control rotation about each of the axes x, y and z, was compared to overlaid sliders, continuous XY+Z controls and a virtual containing sphere. Overlaid sliders place the X and Y sliders over the object to be manipulated; the Z slider is 'wrapped' around the object. Continuous XY+Z sliders define a perimeter around the object, inside which any mouse action will cause proportional rotation around the X and Y axes. Mouse action outside of the perimeter will cause rotation around the Z axis. The virtual sphere places the object within a virtual tracker ball; any action on or around this sphere translates into rotation about the relevant axes. The results of these experiments showed that for complex manipulations the continuous methods were more efficient. However, for single axis manipulation the discrete sliders were most effective.

With the falling price of Virtual Reality equipment, 3D input devices have become more available (such as wands, DataGloves and 3D mice). These devices provide users with a more intuitive interface to the data space. These interfaces have the ability to manipulate objects directly, with 6-degrees of freedom, as shown in figure 3.1. This 6-degrees of freedom interaction, enables an object to be rotated around all three axes simultaneously and translated in any direction.

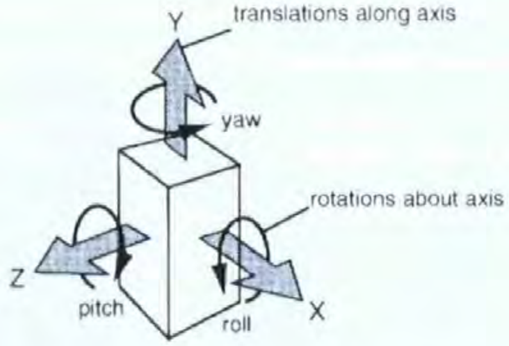


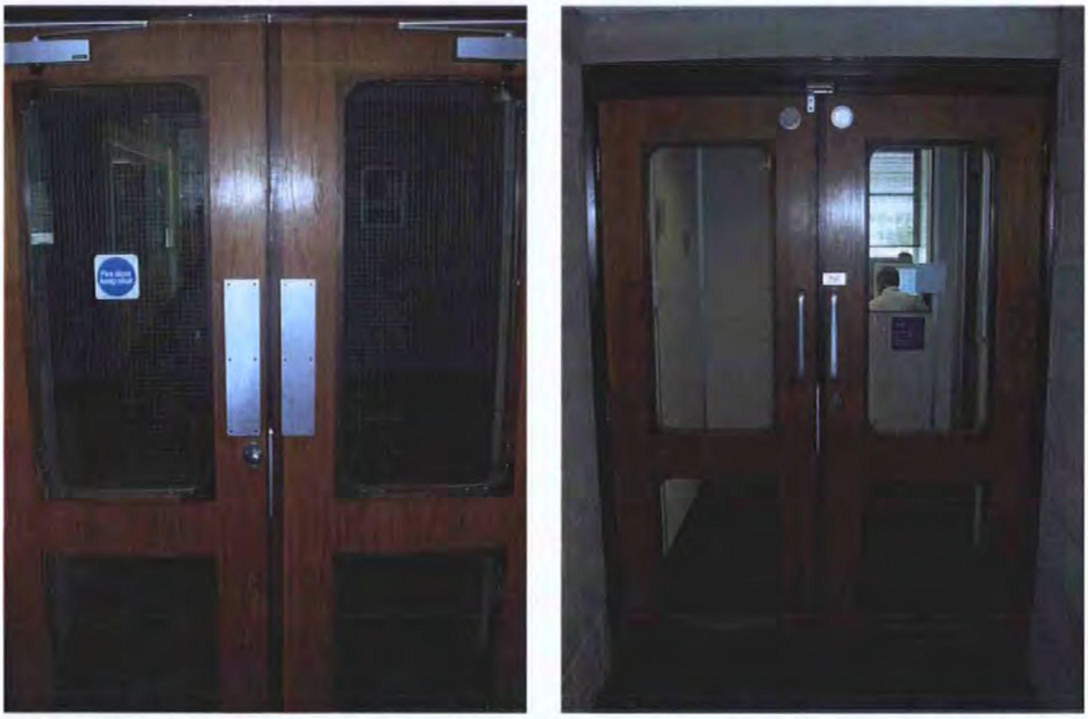
Figure 3.1: 6-Degree of Freedom Model

Therefore, 3D input devices can be used to manipulate an object, or a set of objects, within a visualization. In addition, these devices can be used to navigate with the dataset.

3.2.2.1 Affordance

Affordance is the resource which an item, or interface object, offers. Affordance is both perceived and actual; the perceived affordance of an object is what the user believes the object offers. The actual affordance is what the object actually offers[Gib77]. For example, a door with a push panel, to most people, will have the perceived affordance of pushing to gain entry. Likewise, a door with a pull handle will likely have the perceived affordance of pulling to gain entry.

In figures 3.2(a) and 3.2(b) a set of doors are shown from both sides, the push plate and pull handle, respectively, are clearly visible. In this case the perceived affordance and actual affordance of the doors is the same.

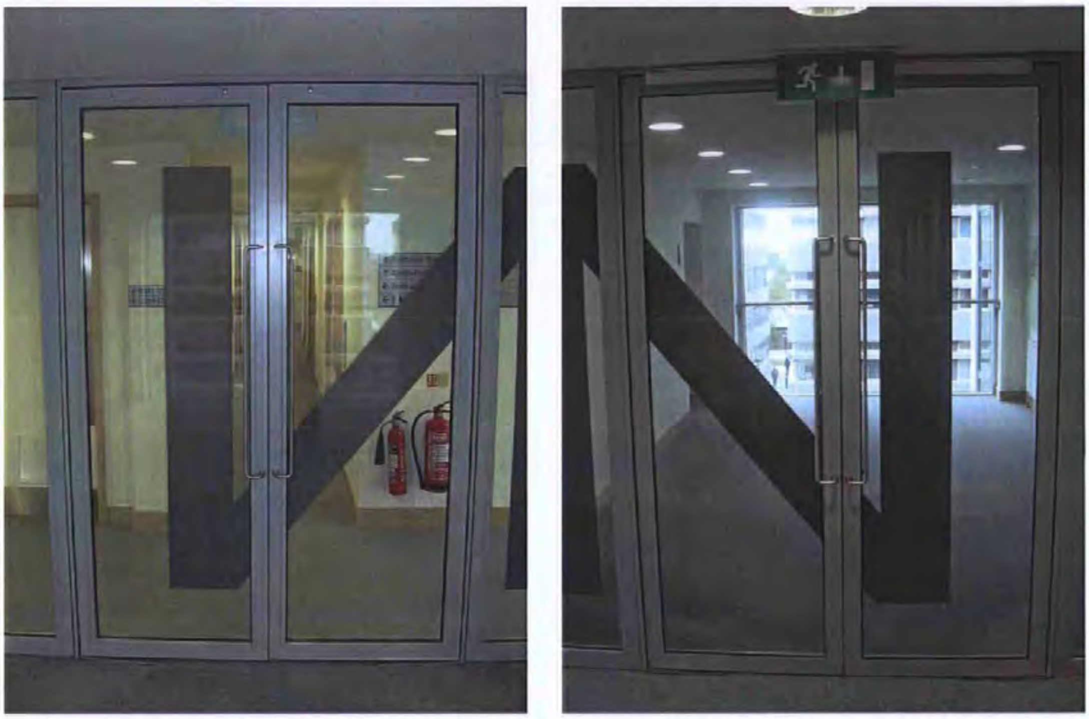


(a) Example of doors with perceived and actual affordance of push

(b) Example of doors with perceived and actual affordance of pull

Figure 3.2: Example of doors with matched perceived and actual affordance

In contrast figures 3.3(a) and 3.3(b) show a second set of doors; these doors have pull handles on both sides. Thus for these doors the perceived affordance from both sides is the same, to pull to gain entry. This perceived affordance is correct for figure 3.3(a), the doors actual affordance is to pull to gain entry. However, in figure 3.3(b) the doors actual affordance is to push to gain entry, thus the perceived and actual affordances do not match.



(a) Example of doors with perceived and actual affordance of pull

(b) Example of doors with perceived affordance of pull and actual affordance of push

Figure 3.3: Example of doors with mismatched perceived and actual affordance

For an interactive visualization to be efficient [PRS$^+$94] the designer must ensure the actual affordance of interface objects match the users perceived affordance of the objects. Moreover, the affordance of a specific object should be consistent[PRS02].

3.2.2.2 Brushing and Selection

Having identified a data item, or collection of data items, of interest within an interactive visualization, the user may want to mark them. This marking or selection can be carried out in a number of ways. For example, "brushing"[BC87], selecting a number of individual points of interest, or "lassoing", selecting a region of the dataset, containing the items of interest, inside which all points will be selected[Wil96].

Brushing within data views can be combined with filtering, via queries or sliders, to refine the selection of data items and viewing relationships[LN03]. In addition, the combination of a number of sequential selections can aid in refining the filter[Che03].

3.2.2.3 Navigation

In addition to interacting with a visualization, the user will need to navigate around the visualization. Users often need to navigate around the dataset and move between views of the dataset to gain insight and understanding. However, a common problem with navigating within large datasets is users 'getting lost', or losing orientation. This problem can be alleviated by restricting the user to navigation along predefined paths. For example, animation[RCM93] can be used to 'fly' the user around the outside of a dataset in a preset search pattern designed to present an overview of the data space. In addition, frames of reference (see §3.2.3.2) can be used to provide the user with a 'map' for orientation.

If a user's view point is switched from one view to another, they can become disoriented. The user must regain context with the data before they can gain any additional understanding from the new position. Animation provides a fast and effective method to move a user from one place to another, without the user losing context[vWN03]. For example, the animated navigation of the Cone Tree[RMC91] representation, which is discussed in more detail in §3.3.3.1.

3.2.3 Multiple Views

A current and important theme that has emerged in recent years is the consensus among researchers in information visualization that 'one view is not enough'[Rob03, Nor01, BWK00, Rob98a]. For most datasets, a single representation does not adequately support the user's needs for extracting information. Moreover, different methods of visualization are more suitable for different levels of investigation. For example, the user may wish to use a different method to gain an overview of the data as opposed to obtaining specific details of the data.

3.2.3.1 Linkage Between Views

With multiple views, there is an inherent requirement to link action in one view to action in other views. For example, if a number of data items, or points, are selected in one view, this brushing should be shown in the other(s). In 2001 North [Nor01] proposed a 'Language, Taxonomy, and System' to define the linkage between interaction in different views. In this paper, North defines a language to specify the result of performing `action A` on `items A` in `view A`, would be, `action B` on `items B` in `view B`, i.e. `((viewA,actionA,itemsA), (viewB,actionB,itemsB))`. Moreover, North states the coupling is commutative, it works both ways (`A` ⇔ `B`), and transitive, if `A` ⇒ `B` and `B` ⇒ `C` then `A` ⇒ `C`. North defines three distinct combinations:

1. **Select ⇔ Select** - `((viewA,`*`select`*`,itemsA),(viewB,`*`select`*`,itemsB))`

   Linking selection in one view to selection actions in another.

2. **Navigate ⇔ Navigate** -`((viewA,`*`navigate`*`,itemsA),(viewB,`*`navigate`*`,itemsB))`

   Linking navigation actions in one view to simultaneous navigation actions in another.

3. **Select ⇔ Navigate** - `((viewA,`*`select`*`,itemsA),(viewB,`*`navigate`*`,itemsB))`

   Linking selection actions in one view to navigation actions in another. This is often called Overview and Detail.

Multiple views, for different tasks, have been used by a number of projects, including the Navigation View Builder project[MFH95]. In this project, Cone Trees and Treemaps (see §3.3.3.1 and 3.3.3.2) are used to overview different levels of a hierarchical data structure. However, a web browser is used to display details of specific nodes. Roberts et al. [RBR02] use multiple, linked, views to aid users in refining web searches.

Linked navigation is used in a number of difference viewers. For example, SeeDiff[BE96], a program that displays two files, or versions of a file, in windows side-by-side to enable visual comparison of differences. In SeeDiff the action of the scroll bars is coupled, thus the user does not need to synchronize the file views after scrolling.

3.2.3.2 Frames of Reference

Salzam et al.[MCLA98] used frames of reference to assist users in the comprehension of abstract information, specifically electric fields. They compare the effectiveness of egocentric (from within the dataset), exocentric (as an external observer of the dataset) and bicentric (both ego and exocentric) frames of reference on a users ability to understand unfamiliar data. The experiment examined the effect of frames of reference on a student's ability to grasp the subject of electric fields.

If the student viewed an exocentric frame of reference, the student explored electric fields as an observer from the field boundary. With an egocentric frame of reference, the student explored electric fields as a test particle immersed in the field. Students with bicentric frames of reference explored electric fields from both the egocentric and exocentric frames of reference on successive learning activities. The experiment showed that alternating between frames of reference best supported the learning experience.

3.2.4 Using 3D

Inherently, the standard computer display is 2D, but for an immersive or semi-immersive Virtual Reality system, 3D is required. In the real world, the third dimension of depth is created by the brain interpreting the horizontal discrepancies between the image seen by the viewers left and right eyes[Vin95]. This discrepancy between left and right eye images is due to the distance between the pupils, the interpupillary distance (IPD)[McA93].

Thus, to create a 3D view of a virtual environment, two view transforms must be calculated, for the current point of view to render the left and right eye images[Sal99]. These individual images must then be presented to the correct eyes, see §3.2.4.2.

3.2.4.1 Stereo Vision

Stereo viewing was pioneered over 100 years ago. In the early Victorian era, the stereo camera was invented, enabling people to take stereoscopic photographic images.

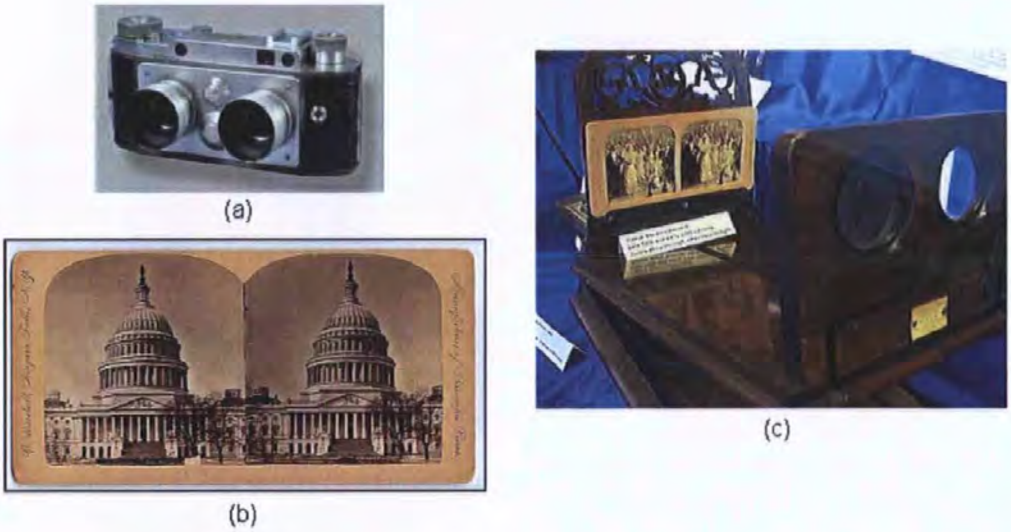


Figure 3.4: Example of 3D photography equipment, (a) a 3D Camera, (b) example photographic image taken with a 3D camera, showing the left and right eye image, (c) a 3D viewer used to present the photographic images to the correct eye

The stereo camera took two simultaneous images, which were slightly offset from one-another. The developed images were then viewed using a special stereo viewer, see figure 3.4(c). The viewer is designed so that the left and right eyes can only see the images taken by the left and right lenses respectively.

A number of methods exist to present stereo images[McA93, FvDFH96, SML97] to the viewer, but the most common are Direct Presentation, Active Filtering and Passive Filtering.

3.2.4.2 Direct Presentation

Direct presentation is a technique in which the left and right eye images are presented directly to the users left and right eyes see figure 3.5.

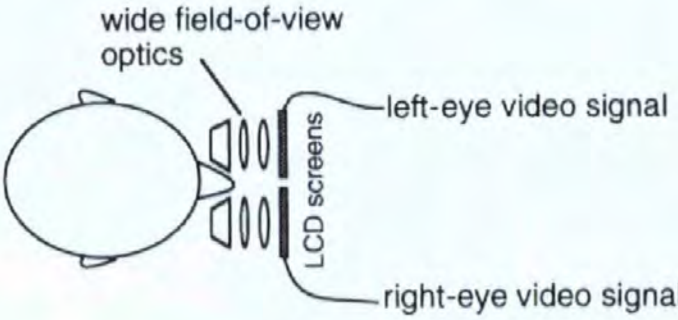


Figure 3.5: The principle of parallel image direct presentation to create a 3D images.

3.2.4.3 Active Filtering

Active filtering, also known as Time-Multiplexing[McA93], uses a single display screen, such as a computer monitor, which rapidly switches between the left and right eye images. The observer wears a pair of special 'glasses' that alternate the view between the left and right eye. The display and glasses must be synchronised so that the left and right eye images are seen only by the left and right eyes, respectively, see figure 3.6.

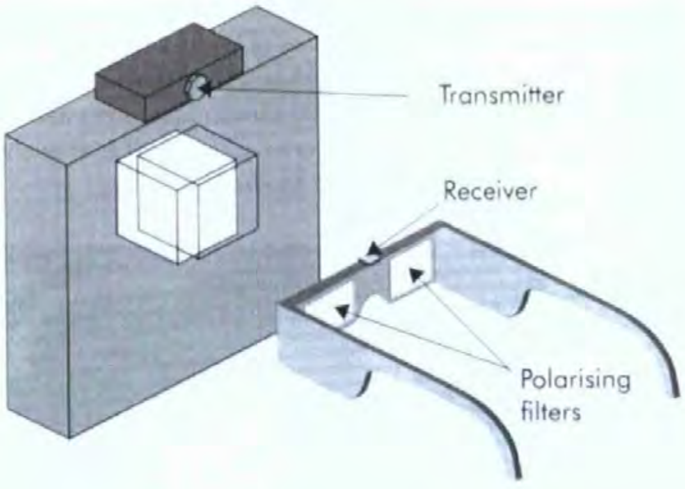


Figure 3.6: Principle of active filtering (time multiplexing) presentation

3.2.4.4 Passive Filtering

Passive filtering presents both the left and right eye images simultaneously. The observer wears a pair of filtering glasses that isolate the left and right eye images, thus presenting the correct image to each eye, see figure 3.7. This filtering can be achieved using a simple anaglyph system, where the two images are presented as shades of different primary colours (for example, red and green, or, red and blue). However, this limits the display to 'greyscale'. For a colour image, either linear or circular polarizing filters must be used[Ans01]. Circular polarization is preferable as it is unaffected by movement in the observers head position. This polarizing method is similar to the IMAX[Cor] cinema system[3Dm]. When viewing IMAX films the observers wear a pair of polarizing glasses. Two images are projected onto the screen polarized to match the left and right eye filters of the glasses. Thus, the viewer sees the movie in 3D.

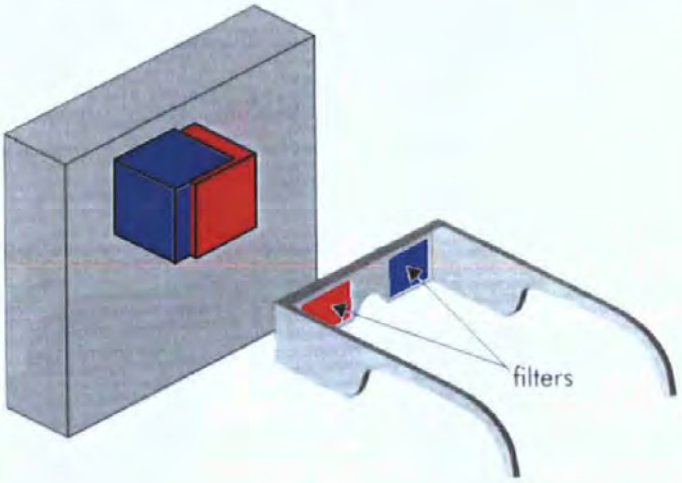


Figure 3.7: Principle of passive filtering

3.3 Datasets

3.3.1 Data Structures

Datasets exist in many different formats, commonly with multiple variables. However, there are seven underlying data structures [Shn96]:

- linear
- 2- dimensional
- 3-dimensional

- multi-dimensional
- temporal
- tree (hierarchical)
- networks

Each structure requires different considerations when being visualised. However, these groups are not mutually exclusive. For example, it is possible to have multi-dimensional temporal datasets which are made up of a number of attributes whose change is measured over time.

When viewing visualizations there is a limitation of 1, 2 or 3 physical dimensions. It is possible to represent temporal information using animation. Further variables of a data item can be represented by assigning their values to different attributes of an object representing the data item. For example colour and size can be used, thus representing each data item by a glyph. Attribute coding, in the form of colour, is common when dealing with heat. For example, when modelling heat dissipation in mechanical design, objects are often represented in 3D with colour representing the heat at a given point. These models can be animated to show the alteration of temperature over time. However, the selection of the mapping of each data variable to a glyph attributes is a complex task[Now97, Tay02]. A poor glyph encoding will produce a visualization that is either hard (or impossible) to use or one that induces false data anomalies. Moreover, certain glyph attributes influence each other. For example, applying a texture to a coloured glyph will alter the colour, due to the effects of shadow. In addition, altering the size of a coloured glyph will affect the user's perception of the shade of the colour.

3.3.2 High Dimension Data

A number of methods are available for dealing with high dimensional data. These range from visualizing multiple dimensions simultaneously to dimension reduction techniques.

The simplest method of displaying high dimensional data is to view all possible 2- and 3-dimension sub planes in sequence. With linked views, the investigator could then 'brush' or 'lasso'[Wil96] points in one sub plane and see where they lie in another[SCK$^+$97]. A collection of sub-plane combinations can be animated allowing users to rapidly overview the data, Rapid Serial Visual Presentation (RSVP).

However, this method can become time consuming and difficult with significantly large numbers of dimensions. In 1996, Symanzik et al.[SCKCN96] explored a number of methods for displaying high dimensional data, within the CAVE environment, an immersive projected virtual reality system. In 1997 within the same environment, Cook et al.[CCNL$^+$97] used the *grand tour* method to deal with high dimensional data.

3.3.2.1 Parallel Coordinates

One particularly notable multiple dimension display technique is Parallel Coordinates. Parallel coordinates represent multi-dimensional data as points on a number of vertical axis (one axis for each dimension) distributed along a horizontal axes, see figure 3.8.

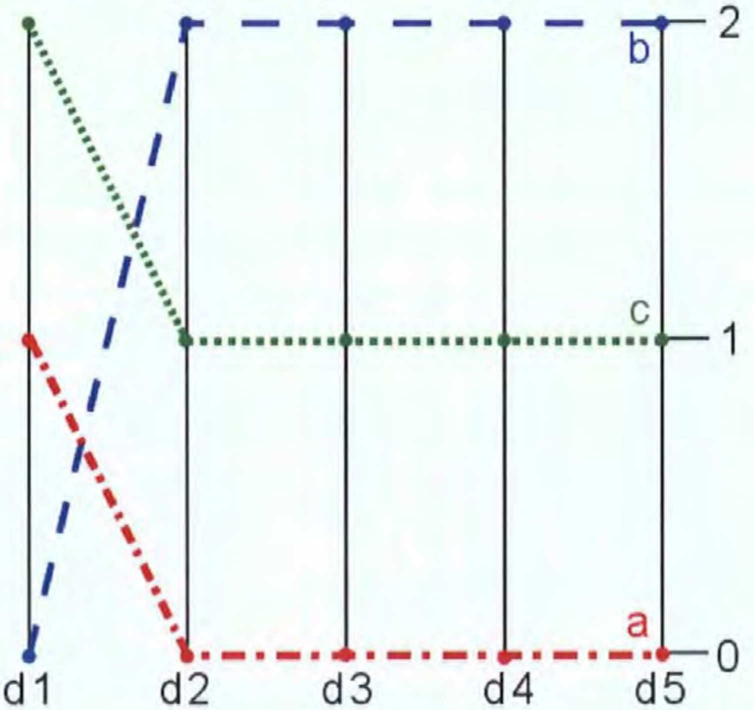


Figure 3.8: Example of 3 coordinates in 5 dimensional space; a(1,0,0,0,0), b(0,2,2,2,2) and c(2,1,1,1,1); using parallel coordinates

Parallel coordinates were originally pioneered in the 1980's. In 1990 Inselberg[ID90] and Wegman[Weg90] regenerated the use of parallel coordinates for analysing large multi-dimensional datasets. Further work with parallel coordinates has been undertaken by Fua et al.[FWR99].

3.3.2.2 Dimension Reduction

Another method of dealing with high dimension data is to use a mathematical dimension reduction technique such as Principle Component Analysis (PCA). PCA forms a projection of high dimensional data onto a lower dimensional hyper-plane. Thus, reducing the number of dimensions to display, whilst maximising the amount of information[BJD81, Ter73]. PCA is used to find the projection with the greatest variance in data. Similar to PCA is the technique of Independent Component Analysis (ICA)[HO00] which attempts to find a projection with the greatest separation between the data. PCA and ICA are both variants of projection pursuit[JS87, Fri87, Hub85, FT74]. In addition, other projection algorithims exist, such as Sammon[Sam69, BJD81] non-linear reduction.

3.3.2.3 Cluster Analysis

Methods also exist to analyse groups and clusters in high dimensioned data. For example, using minimal spanning trees or gestalt analysis[Zah71]. These methods deal with datasets that have inherent spatial relationships.

In addition, cluster analysis can be undertaken on any dataset for which a mathematical metric can be defined to measure the 'distance' between data points. In addition to defining the distance between individual pairs of data points, the distance between clusters must be defined; this can be achieved by a number of algorithms. This includes the following methods of cluster linkage: *Single Linkage* which measures the distance between clusters by the distance between the two closest points within the clusters; *Complete Linkage* which measures the distance between clusters as the distance between the furthest pair of points within the two clusters; *Average Linkage* which measures the distance between two clusters as the average distance between all possible pairs of points within the two clusters. At each iteration of the cluster analysis algorithms, the two clusters with the smallest distance between them are merged. This process continues until the whole dataset forms a single cluster or the separation between clusters reaches a pre-determined threshold. The cluster linkage can then be examined and clusters in the dataset identified.

3.3.3 Hierarchical Data

A considerable amount of data exists in hierarchical form, such as a company management structure, a computer filling system or the worldwide web. Traditional tree representations do not represent large hierarchical structures well due to the amount of space the representation requires. To alleviate this problem two different methods are examined: the Cone Tree and Treemaps.

3.3.3.1 The Cone Tree

One method of displaying hierarchy of objects is the Cone Tree[RMC91, EA95]. The Cone Tree moves standard 2D tree diagrams into 3D, allowing more items to be displayed, see figure 3.9. However, representing the tree in 3D created a new problem, *occlusion*; where data items in the foreground 'hide' others in the background. This problem can be eased by making the data items semi-transparent, thus allowing users to look 'through' them.

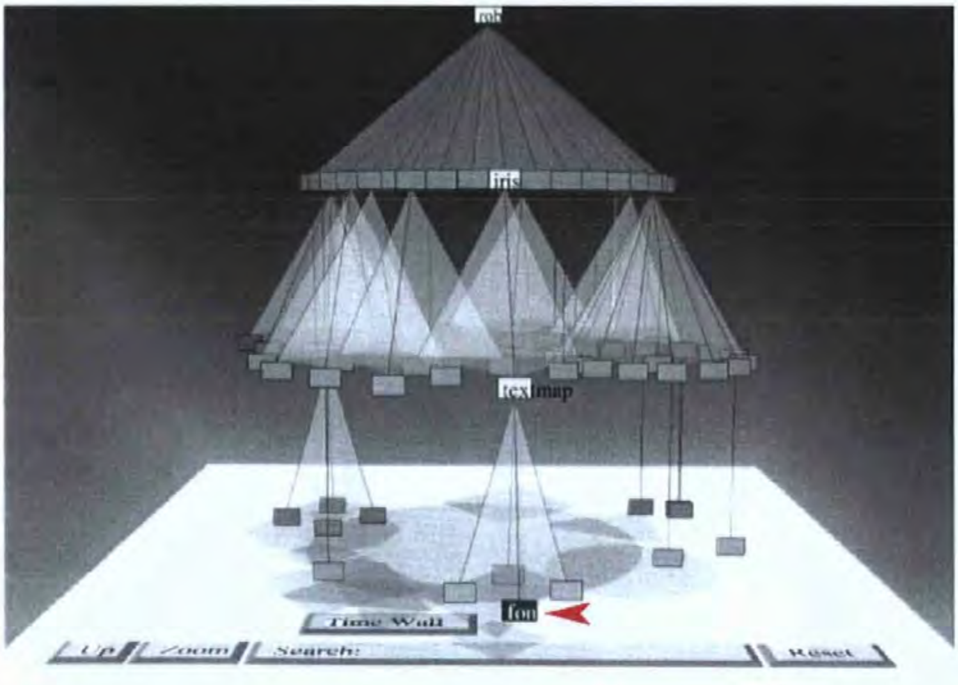


Figure 3.9: A Cone Tree representation of a partial file system, demonstrating the problem of labelling and the use of semi-transparency (red arrow indicates file for comparison with figure 3.10)

Tree structures usually exist in one of two 'shapes', either 'short and fat' or 'tall and thin'. Shifting a Cone Tree from a vertical to a horizontal orientation (or visa versa) can increase the amount of space available to display labels improving data legibility, see figure 3.9 and figure 3.10. The red arrows in each of these figures indicate the same file, to aid comparison; the buttons on the floor of the diagram are user controls.

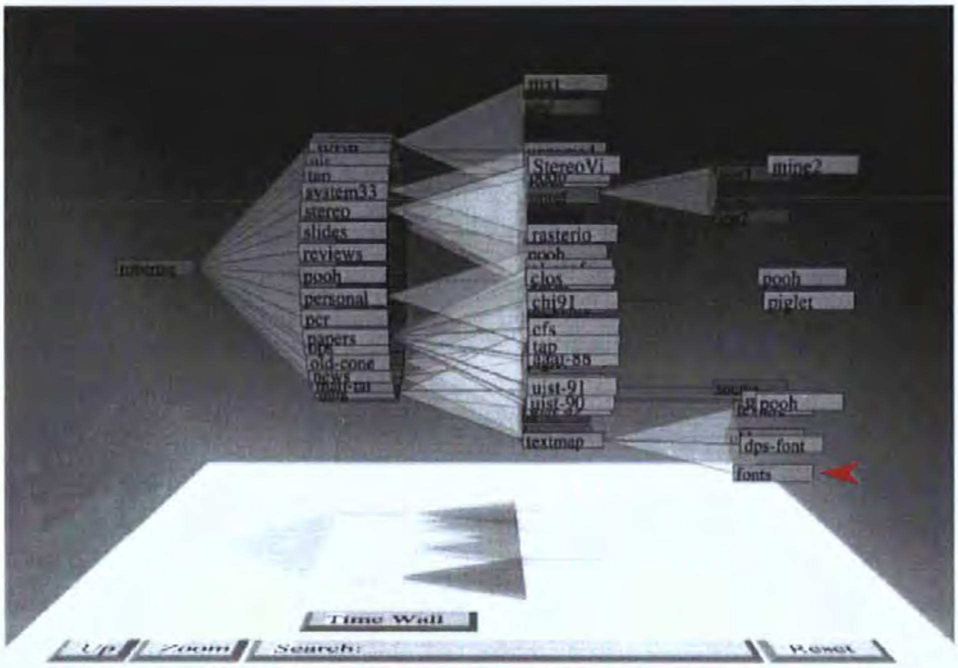


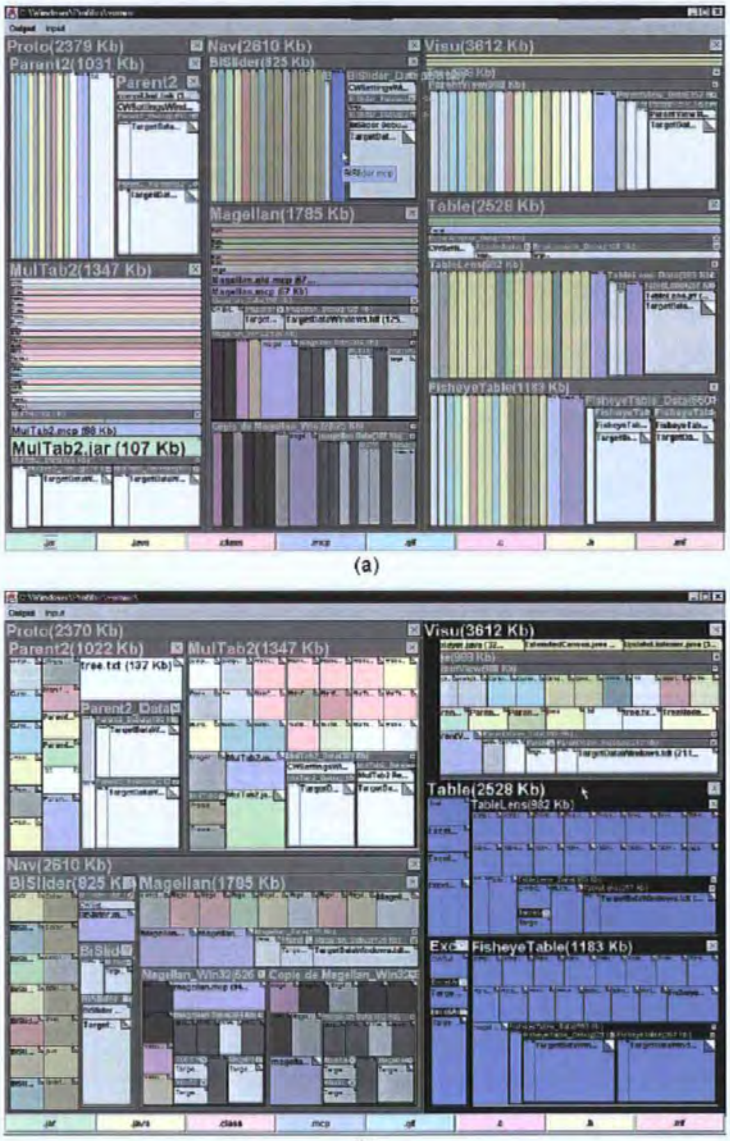
Figure 3.10: A Cone tree representing the same information as figure 3.9, however the tree is shown in the Horizontal axis, thus improving labelling and clarity (red arrow indicates file for comparison with figure 3.9)

When the user selects a node of the tree to focus upon, the cone tree smoothly rotates through the shortest path to bring the selected node to the front. This rotation is achieved by smoothly animating the tree. This smooth animation, combined with the shadow of the tree, helps the user perceptually track the changes in the tree layout through object constancy, thus no time is needed to re-assess the tree after navigation.

3.3.3.2 Treemaps

Another method to represent hierarchical data is Treemaps, which subdivide the data space to show the hierarchical structures. Treemaps can encode data variables in the size and colour of the units of the Treemap. One problem with the space sub-division method used by the traditional Johnson and Shneiderman Treemap is that is produces 'thin rectangles'. These thin rectangles are difficult to compare, either with each other or with square units, which is a fundamental principle of treemaps. In addition, thin rectangles are difficult to label. Vernier and Nigay[VN00] demonstrated a *Modifiable*

Treemap and compared it to the original version, see figure 3.11. This method is aimed at overcoming the problems caused by 'thin rectangles'.



(a)

(b)

Figure 3.11: Treemap visualization of file size within a partial file system, using (a) original Johnson and Shneiderman Treemap representation, (b) Modifiable Treemap representation

Bruls et al.[BHvW00] demonstrate a *Squarified Cushion Treemap* and a *Framed Squarified Treemap*, see figure 3.12 parts (c) and (d) respectively. These methods also deal with the problem of thin rectangles. Moreover, the cushioning and frames attempt to show the nesting of the treemap more clearly.

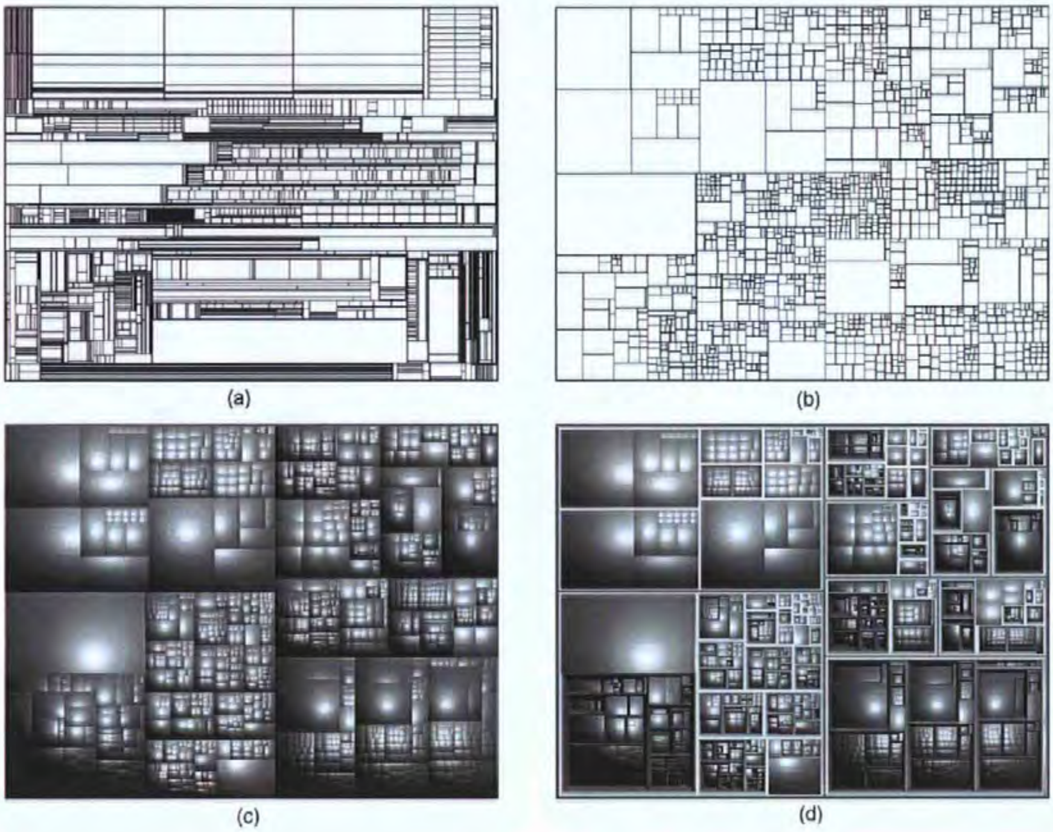


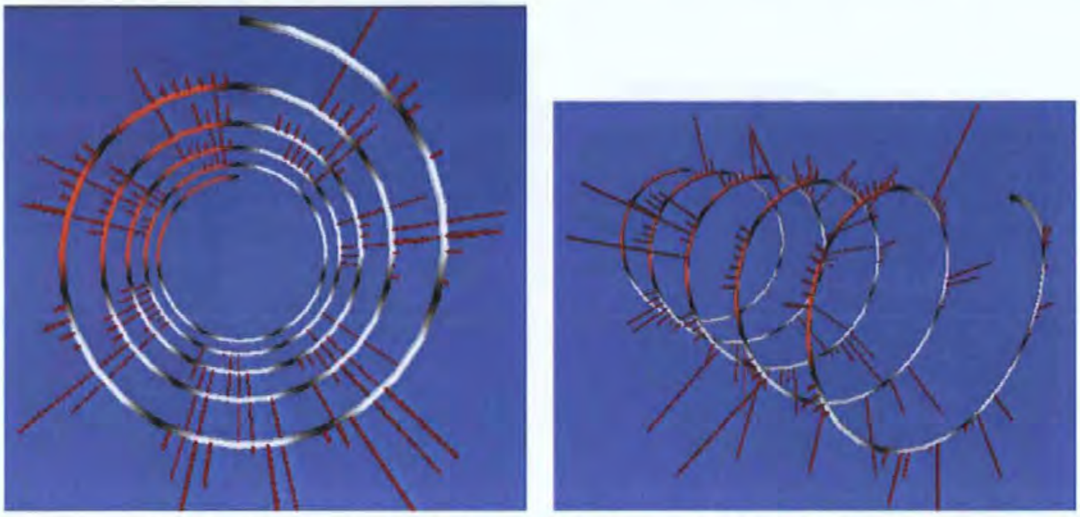
Figure 3.12: Squarified Cushion Treemap representing a partial file system, (a) original Johnson and Shneiderman Treemap representation, (b) Squarified Treemap representation, (c) Squarified Cushion Treemap representation, (d) Framed Squarified Treemap representation

3.3.4 Temporal Data

Many datasets deal with multiple variables and record the change in those variables over time. In addition to displaying these multiple attributes, representing the relationship and change with time is key to analysing this data[MHSG02].

3.3.4.1 Helix

In 2001 a Helix metaphor was proposed by Hicks [Hic01] to represent customer behaviour data. This data tracked the number of emails received and phone calls made by customers over a given time frame. The data was then plotted as columns perpendicular to the back-bone of the helix, the height and colour of the segments of the columns encoding the data, see figure 3.13.



(a) An example Helix showing the distribution of emails received by 20 UK households on ADSL, frontal view

(b) An example Helix showing the distribution of emails received by 20 UK households on ADSL, rotated view

Figure 3.13: Example of the Helix metaphor

One rotation of the helix represented 7 days of data, thus by looking along the major axis of the helix the data can be compared on a week-by-week basis, see figure 3.13(a). The colour of the helix differentiates week days (white) and weekends (red), also the shade of the helix shows daytime (light) and night time (dark). This shading uses a clearly recognisable metaphor, sun light, to aid the user in quickly identifying the part of the day the helix represents.

3.3.5 Document Visualization

The field of document visualization is concerned with providing users with an accurate overview of the content of a document, or set of documents, without the need to read them[SB03, YPWR03, WTP+95].

3.3.5.1 Text Arc

TextArc, developed by Paley [Palb, Pal02], is designed to provide an overview of the frequency and location of words within a document. To generate a TextArc, first the text of the document is displayed in an ellipse around the outside of the screen, line by line. This display of the text maintains the typesetting of the original document, thus chapter breaks can be distinguished. The text of the document is then repeated, word for word, around an inner ellipse. If a word appears in the text more than once then it is printed in the central region of the ellipse, its position being determined by the distribution of the locations where the word 'should' appear on the ellipse. Thus, words that appear evenly distributed though-out the document will appear in the centre of the display. Figure 3.14 shows an example TextArc for the book Alice's Adventures in Wonderland.

The words are rendered onto a black background, the luminosity of each word is dependent on its prevalence in the text. When a word on the display is selected, occurrences of that word are highlighted in the text ellipse (outer ellipse). This is shown in figure 3.15, where the word 'Alice' has been selected.

By selected multiple words the relationship of those words, within the document structure, can be examined. In addition to selecting individual words, the flow of the text can be animated to provide an overview of the flow of the terms in a document.

This visualization of document content can be used to examine where different terms are defined in the text. In addition, the selection tools can aid in identifying how different terms are related, or if the text was a novel how two character are related, such as the King and Queen of Hearts in Alices Adventure in Wonderland.

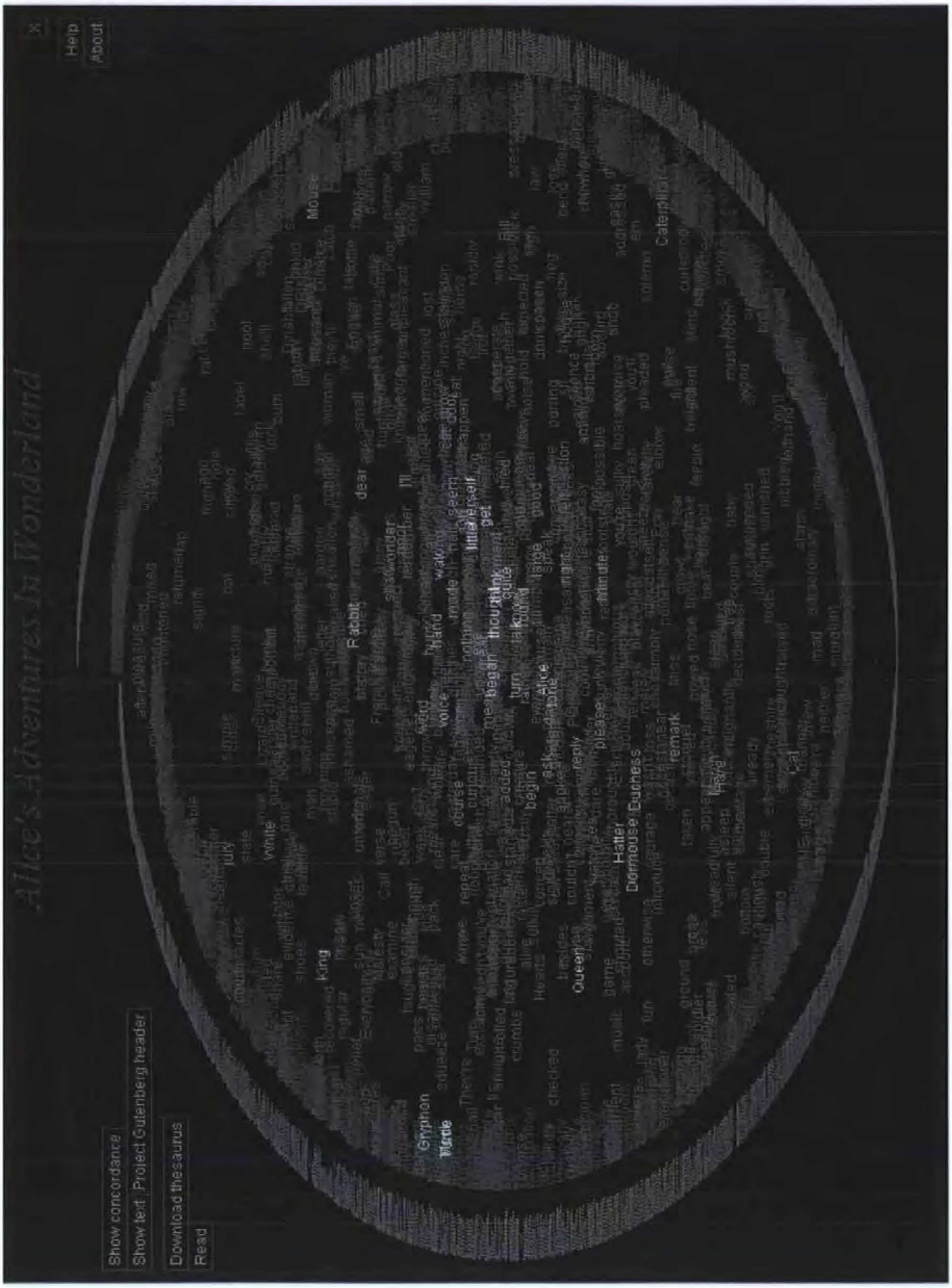


Figure 3.14: Example of a TextArc representing the book Alice's Adventures in Wonderland[Pala]. The text of the novel is shown in the outer arc, the individual words of the text and shown on the inner arc, the most common words occupy the central region of the graphic.

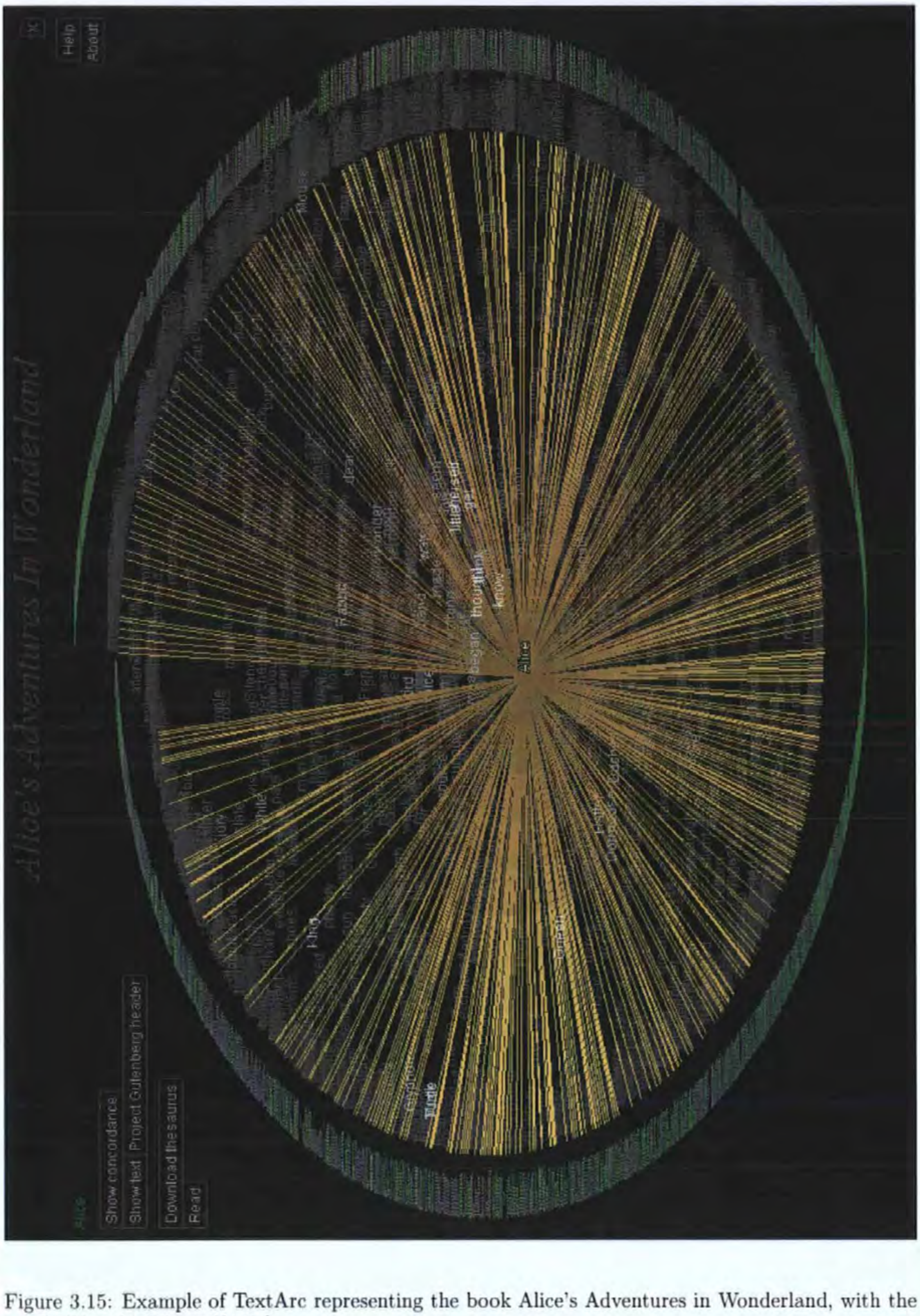


Figure 3.15: Example of TextArc representing the book Alice's Adventures in Wonderland, with the word 'Alice' highlighted[Pala].

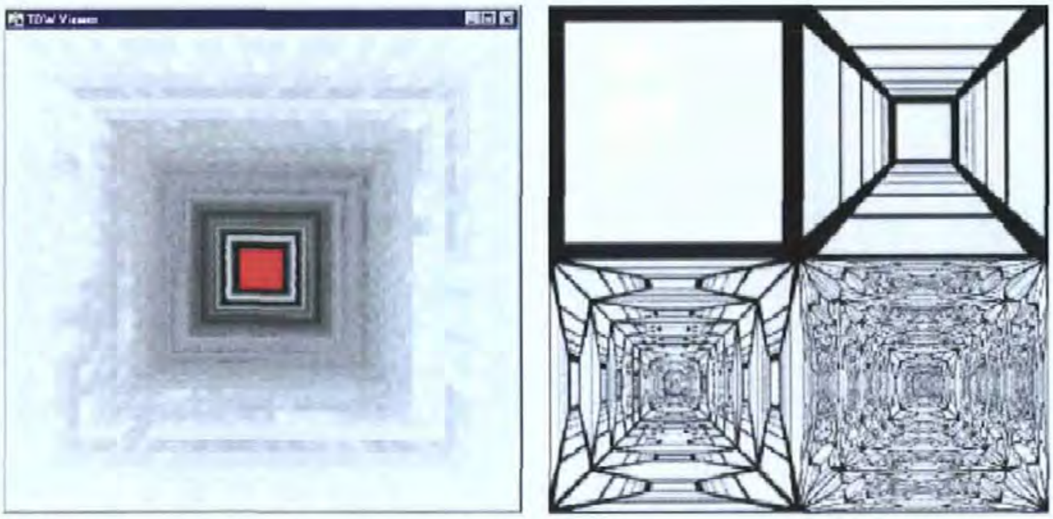
3.3.6 Increasing Dataset Size

The increasing quantity of data that users wish to visualize is becoming as much a problem as the data dimensionality. There is a continuing trade-off between the amount of data displayed and the clarity and legibility of the data. New improved methods for displaying and representing data are needed in order to make the most effective use of the display space available. To date, a number of different approaches for expanding virtual display size have been suggested:

1. Perspective
2. Spatial morphing
3. Window layout
4. Level-of-detail

3.3.6.1 Perspective

Perspective Tunnels[MK97] segment the display into a number of virtual displays, which themselves can be divided thus creating an 'unending' workspace. Figure 3.16(a) shows an example of the Time Tunnel, a helical perspective tunnel layout.



(a) Helical Perspective Tunnel visualization of daily aggregate share prices on the Wall Street stock exchange from 1907 (far) to 1992 (near)

(b) Recursive perspective projections technique for visualizing graph structures shown at successive levels of recursion

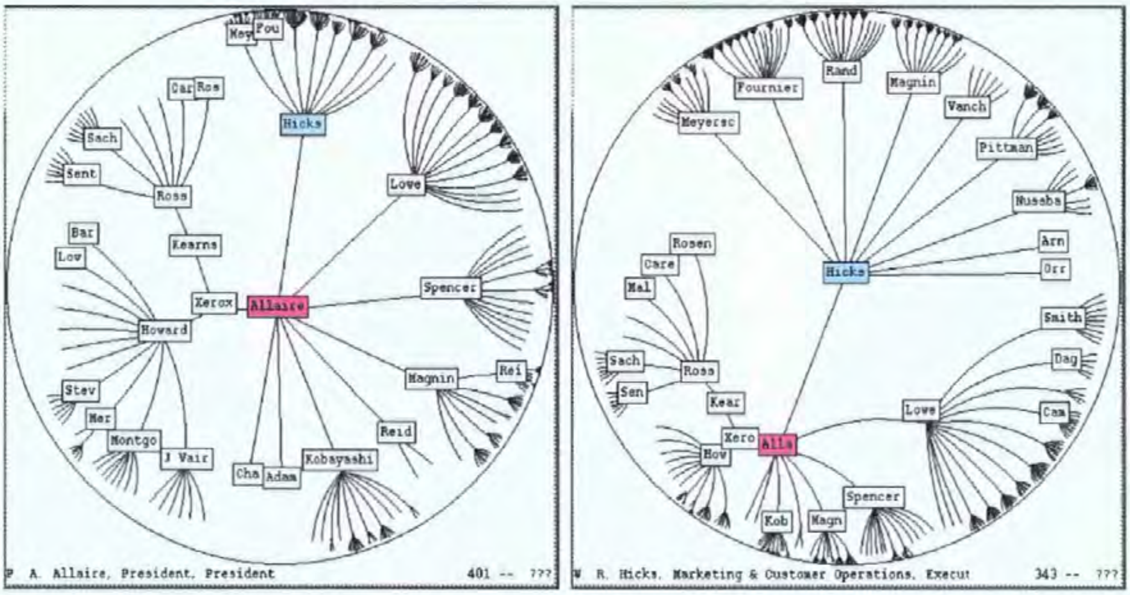
Figure 3.16: Perspective Tunnel Examples

This tunnel represents stock market date from 1907 (far) to 1992 (near), showing the daily aggregated total over the whole Wall Street exchange, from black (low) to white (high). The values are wrapped around the tunnel walls in a helix producing the visualization.

Figure 3.16(b) shows a perspective tunnel showing hierarchal data. The walls of the tunnel are successively sub-divided to show the levels of the hierarchy.

3.3.6.2 Spatial Morphing

'Spatial morphing' techniques have also been demonstrated. These techniques are similar to viewing the data through a fish eye lens; which allows a photograph to be taken of an object in high magnification, in the centre of the image, while showing the objects surroundings in lower magnification. Data in the centre of the display area is magnified while peripheral data is 'diminished'. This supports a clear view of the data area of interest, while maintaining context with the overall data structure. A prestigious implementation of this lensing is Xerox Parc's Hyperbolic Browser[LRP95], see figure 3.17.



(a) Browser example showing company management structure, President (Allaire) in centre of browser

(b) Browser example showing company management structure, having navigated to Marketing and Customer Operations Executive (Hicks)

Figure 3.17: Xerox Parc's Hyperbolic Browser

This figure shows a company management structure, first with the President (Allaire) as the focus (figure 3.17(a)) and secondly the tree having been navigated to place the Marketing and Customer Operations Executive (Hicks) in focus (figure 3.17(b)). In 2003 Jankun-Kelly and Ma[JKM03] demonstrated MoireGraphs, which are based on a similar principle to the Hyperbolic browser. In addition, heat models have been proposed as a method to control the expansion and contraction of tree structures[Osa01].

3.3.6.3 Window Layout

In addition to expanding the virtual workspace, studies have been undertaken to determine the most effective layout of windows and other components. Work by Kandogan and Shneiderman[KS96] explored the use of *Elastic Windows* to maximise the use of the available workspace. The Elastic Windows system is based on a hierarchical, space filling system and manages the size and layout of a set of open windows.

This system permits a user to structure the window layout of a display, forming groups and hierarchies of windows that map to the user's needs. When the user resizes a specific window, the other windows in the group will alter in size and location to accommodate it, thus making the most effective use of the space. Elastic Windows eliminate overlapping of windows and thus can aid in effective multi-task operations.

The user can position similar information in adjoining windows, thus reinforcing the relationship between the contained data. Advantages and disadvantages of inferred relationship due to placement of data have been explored by Ward and Keim[WK97].

3.3.6.4 Level-of-Detail

Information can be selectively displayed at different levels of 'magnification'. The data is not lost; the level-of-detail of the data is adapted to help users explore the dataset. When users wish to overview the data, clusters of points can be rendered as single objects. As the user 'zooms in' these objects can be split to show their content.

This approach is also known as Semantic Zooming and has been exploited by Summers et al. [SGK03] for visualizing the structure of computer programs. The program is initially displayed as a set of high level, connected, objects. As the user zooms in on an object of interest, the object turns transparent and its content can be seen and explored in detail.

Chapter Summary

In conclusion, Information Visualization techniques can be utilised to produce powerful and effective representations of complex data. Shneiderman's mantra provides a key framework for structuring these techniques. In addition, it is clear that logical affordance of interface objects is critical for effective user interaction and navigation.

In addition to individual representation, multi-view systems provide comprehensive environments for exploring data. These systems provide the user with several inter-linked representations of the same dataset, exploiting the most effective features of each representation.

In the following chapter, neurophysiology is introduced, including methods of data capture and analysis.

Chapter 4

Neurophysiological Data Analysis

Summary

This chapter introduces the field of Neurophysiology, and more specifically multi-dimensional spike train data. The theory of temporal coding which supports information processing in the brain is also described. Subsequently, the current methods for analysing temporal synchrony are presented. Finally, the requirements for software support in this area are detailed.

4.1 The Nervous System

There are many different types of neuron in the mammalian nervous system, each of which performs a different task. For example, sensor neurons, interneurons and motor neurons. *Sensor neurons* receive input from stimulus receptors, such as touch, taste, sound and vision. *Interneurons* or *association neurons*, found exclusively in the spinal cord and brain, form connections between other neurons. *Motor neurons* transmit impulses from the central nervous system to muscles and glands [Rob98b].

The basic structure of all neurons is the same. A neuron consists of a cell body from which a long thin tube-shaped fibre, called the axon, extends. Also a collection of shorter, and often highly branched, extensions called dendrites protrude from the cell body, see figure 4.1. This is known as the dendritic tree.

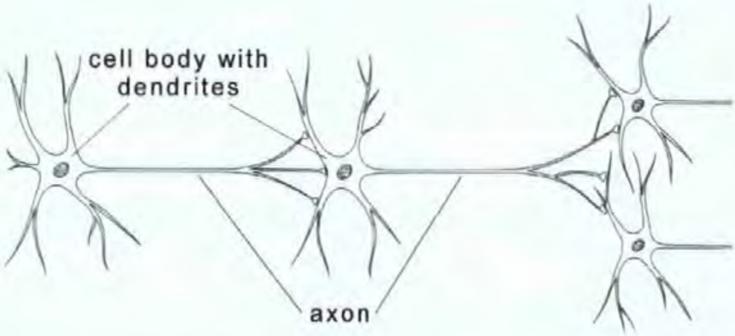


Figure 4.1: The main features of a typical vertebrate neuron, indicating the dendritic tree, cell body, axon and synapse [Rob98b].

The shape of the cell body, axon and the layout of the dendritic tree differ depending on the purpose of the cell. This is illustrated in figure 4.2.

Axons are the structure of the neuron used for connectivity, or coupling. They connect one neuron to another and often branch to support multiple connections. Connection of the axon can take place at either the cell body or along the dendritic tree. The position of the connection determines its strength and type. A connection made close to the cell body, of the recipient neuron, has a greater effect than one made further down the dendritic tree. At the point of connection the axon swells, this is known as the axon terminal. The region between the axon terminal of neuron A and the dendrite of neuron B is called a synapse, as shown in figure 4.3. A synapse consists of a pre- and post-synaptic membrane. The small gap between these two membranes is approximately 20nm, it is called the synaptic cleft.

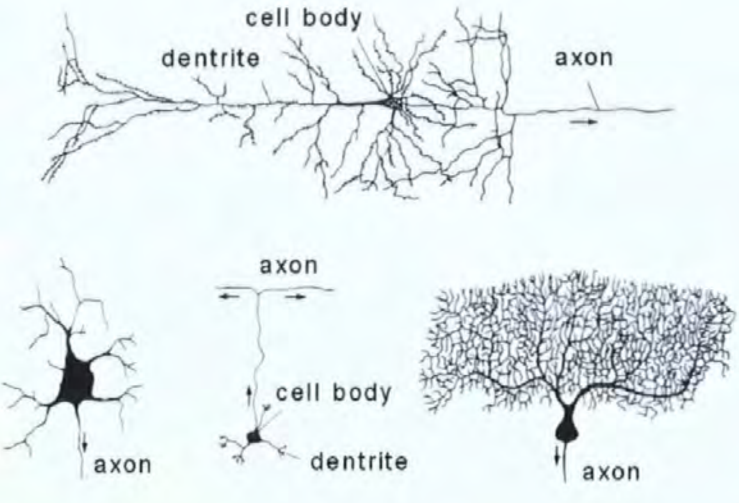


Figure 4.2: Neurons from different regions of the mammalian nervous system showing variations in cell body shape and the structure of dendritic tree [Rob98b].

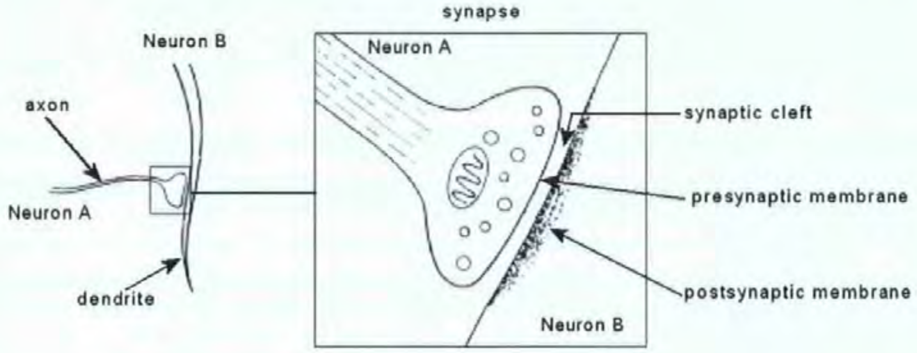


Figure 4.3: Example of axon of neuron A connects to the a dendrite of neuron B [Rob98b]

4.1.1 Action Potentials

Neurons communicate via short electrical impulses called action potentials or spikes. Each neuron accumulates charge as a result of the action potentials it receives from other neurons that are connected to it. If this accumulation of charge exceeds the neuron's internal threshold then the neuron receiving action potentials will generate an action potential itself. Action potentials are commonly referred to as spikes and a series of spikes is known as a spike train.

When a neuron generates a spike at the cell body, the electrical impulse is transmitted from the cell body along the axon with no degradation in the amplitude of the spike. This propagation characteristic is commonly referred to as an 'all-or-none' transmission. However, when the spike is received by the dendritic tree of the target neuron the amplitude of the spike reduces as the spike is transmitted along the dendrite to the cell body. The dendritic tree acts as a temporal integrator, the further away from the cell body a spike is received, the weaker the effect of that spike. In addition, the effect of a spike is transitory, the charge of the spike decays over time.

4.1.2 Neural Coding

The ability to record neuronal activity within the brains of mammals has resulted in large quantities of experimental data. This data is in the form of multi-dimensional spike train recordings. Within Neuroscience, it is essential for scientists to understand this data and to gain insight from it.

There are currently two main views on how information is encoded in the brain[Wan], with a number of intermediate possibilities. One proposal is that the *average rate* of action potentials over a given time interval holds the information, this theory is know as *Rate Encoding*. The other is the theory of *temporal coding*, that the information is encoded in the patterns of spikes, even in their exact temporal sequence.

The principle of *temporal coding* is supported by evidence from a number of sensory systems. For example work by Davison and Brown[DFB00] on the olfactory bulb; Simon et al.[SPPC01] *in the auditory brainstem of the barn owl*; Victor[Vic00] examines *How the brain uses time to represent and process visual information*; Quenet et al.[QHDD01] examine *Temporal coding in an olfactory oscillatory model*; and work by Cariani[Car] on *Temporal coding of sensory information in the brain.*

4.1.3 Coupling

In addition to the information encoded in the temporal patterns of individual neurons, the temporal synchrony of assemblies is believed to encode more complex information, such as cognitive information, such as memory[BB97, FNE$^+$99]. There are two fundamental cases of coupling where synchrony will occur between the spike train output of two neurons. These cases are direct synaptic coupling and common input coupling.

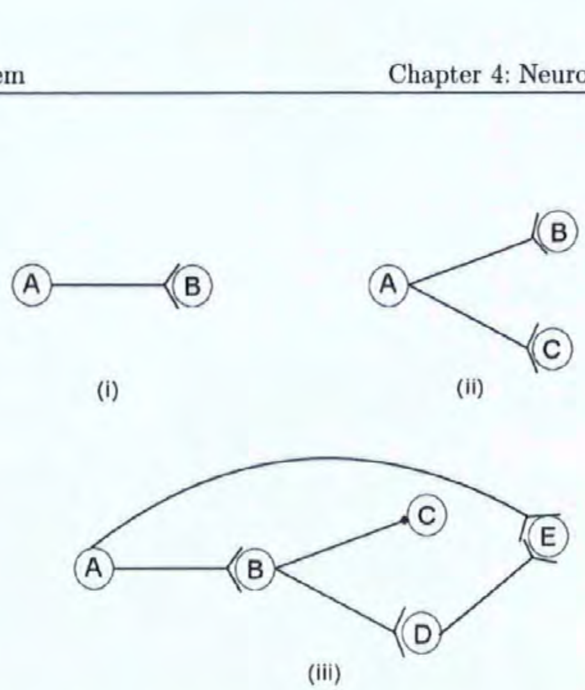


Figure 4.4: Example of (i) direct synaptic coupling, neuron A connects to neuron B. (ii) Common input coupling, neuron A connects to both neurons B and C. (iii) Excitatory (represented by forks) and inhibitory (represented by dots) connections, neuron A has an excitatory connection to both neurons B and E, neuron B has an inhibitory connection to neuron C and an excitatory connection to neuron D and neuron D has an excitatory connection to neuron E.

Direct synaptic coupling is where the axon from neuron A connects to Neuron B, as shown in figure 4.4(i). In figure 4.4, note that the excitatory connections are shown as a 'fork' at the end of the connecting line and the inhibitory connections are shown by a 'dot'. In this type of coupling neuron A excites neuron B thereby increasing the probability of a spike from neuron B.

Common Input is where neurons B and C both have a connection from a third neuron A; see figure 4.4(ii), resulting in correlated input. In this type of coupling, if neuron A fires then both neurons B and C have increased probabilities of firing.

These coupling capabilities may be combined to connect larger assemblies of neurons as shown in figure 4.4(iii), where both excitatory and inhibitory connection are represented. In this assembly, if neuron A fires then neuron B has a greater probability of firing. This in turn causes neuron D's probability of firing to increase and inversely neuron C's probability of firing to decrease. As neuron E has a connection from both A and D, its firing probability is even higher.

4.2 Data Acquisition

The spike is the base unit of communication. Spike train recordings are one of the most common forms of data collected during neurophysiological experimentation on a neuron. When a number of neurons are under investigation, recordings from each neuron are made simultaneously, this type of recording is referred to as multi-dimensional spike train recording. Thus, multi-dimensional spike train recordings are a record of the activity of a collection of neurons simultaneously under investigation.

As each spike from a single neuron is identical, the form of the individual spike is believed to carry no information[Rob98b]. In contrast, it is the spiking frequency and the inter-spike-intervals that are believed to encode the information.

4.2.0.1 Recording Data

These methods are invasive, because electrodes damage the nervous tissue. In this type of recording, a number of very fine electrodes are implanted into the subject's brain to obtain accurate readings of individual neurons at various locations. A number of researchers[VWTDS99, Ltd01] are developing even more precise methods for placing and maintaining these electrodes during investigation. The raw data gained from this intercellular recording is classed as 'dirty' as it contains noise and cross signal contamination. To obtain a 'clean' dataset of these recordings, a number of statistical pre-processing methods can be employed. The result of this pre-processing is the production of a number of spike trains, one for each neuron recorded.

4.2.1 Abeles File Format

One of the most common data file formats, for storing multi-dimensional spike train recordings, is the Abeles[NHa, NHb] format. This format permits a number of spike trains and experimental trials to be recorded in one file with comments.

The Abeles file format encodes spike train data as a record of events. Each event is stored as a set of three numbers, known as triples, and comments may also be entered. The first two numbers of the triple describe the event. The final number defines the quantity of time that has elapsed since the previous event. This time is measured as the number of specified time units, such as milliseconds. The first descriptor is known as the event Type and the second is the Qualifier. For example the Qualifier could show the spike train number is the Type is spike. Both of these descriptor numbers are specified in a hexadecimal format.

`0 1 0`	Start of recording
`1 1 15`	
`1 3 5`	
`1 1 10`	
`0 0 99`	Empty event, used for padding
⋮	
`0 2 15`	End of trial
`'x = 12, y = 78'`	Comment
`0 1 0`	Start of trial
⋮	
`0 2 6`	End of trial
`0 FFFF 0`	End of file

Figure 4.5: A fragment of a data file recorded in Abeles file storing spike train data (left) and brief explanation (right).

Figure 4.5 shows a fragment of a data file recorded in Abeles format with key lines explained. The event Type `0` is reserved for control events. These control events can have one of four qualifiers: `1`, the start of a recording; `2`, the end of a recording; `FFFF`, the end of the file or `0`, an empty event. The empty control event is used to pad the file if the time between events is greater than 99 time units.

Comments can be placed in the file by delimiting them with single quotes. In addition, there are a number of keywords defined [NHa]. Also note that the excerpt shown in figure 4.5 uses a new line for each event, it is possible to use other delimiters, such as tabs or commas. The Abeles format will be discussed further in chapter 5. Other formats exist, but this research focuses on Abeles format.

4.3 Current Methods of Spike Train Analysis

One of the main theories within temporal coding is the principle of synchronisation of neural activity [BB97, FNE$^+$99]. This theory proposes that information is encoded in the specific temporal arrangement of the spike in the spike train generated by an individual neuron. Moreover, information is represented in the synchronous spiking behaviour of inter-connected neurons.

There are currently a number of methods available to investigate the temporal synchrony within spike train datasets. These methods can be split into two groups, 'raw' spike train data display methods and statistical analysis methods.

4.3.1 Raw Spike Train Data Display

It is useful to display data obtained from spike train recordings with minimal post-processing. This type of display can permit the investigator to gain an overview of the recorded data.

4.3.1.1 Raster Plots

The raster plot is a direct representation of the temporal arrangement of spikes recorded from a neuron. In a raster plot, each spike train is displayed as a line of dots along a temporal axis. Each dot denotes the presence of a spike at that time.

Raster plots can also be used to compare a number of recordings from the same neuron. This aids in the identification of similarities between trials. In addition, raster plots can be used to view a number of spike trains from different neurons. Figure 4.6 shows two different raster plots. In the first plot (i) the two trains are related, note the spikes of train a are followed by a delayed spike on train *b*, this indicates that train *b* has a tendency to spike after train a. In contrast, in the second plot (ii) the trains are unrelated; it is not possible to observe any similarity in spiking pattern between the trains.

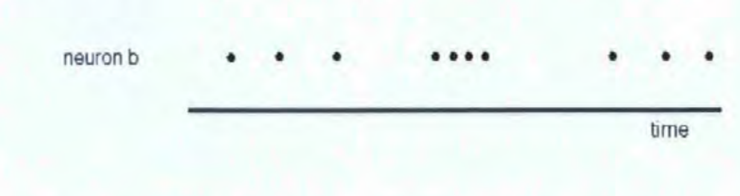


Figure 4.6: Example raster plots, (i) two related trains, (ii) two unrelated trains

Spike train comparison can be problematic where there are a large number of trains, or when 'similar' trains are not close to each other in the plot. This problem is demonstrated in figure 4.7, where a raster plot is shown with (a) the trains in a random order and (b) the trains ordered so that similar trains are adjacent. To alleviate these problems an algorithm should be used to reorder the trains in the display so that 'similar' trains are next to each other.

The data for the raster plots shown in figure 4.7 was produced using an enhanced Integrate and Fire neuronal generator[Bor02]. The connections are defined using set notion, thus $s_x = \{c_{ij} : i = 1, j = 2, \ldots, 5\}$ indicates connections from neuron one to neurons two to five. The connection of this 100 neuron assembly is as follows: $s_1 = \{c_{ij} : i = 1, j = 2, \ldots, 15\}$, $s_2 = \{c_{ij} : i = 16, j = 17, \ldots, 30\}$, $s_3 = \{c_{ij} : i = 31, \ldots, 40, j = 31, \ldots, 40, i \neq j\}$, $s_4 = \{c_{ij} : i = 41, \ldots, 45, j = 41, \ldots, 45, i \neq j\}$, $s_5 = \{c_{ij} : i = 46, \ldots, 50, j = 46, \ldots, 50, i \neq j\}$. The trial lasted 20000 ms, however, only a portion of the plot from 200ms to 1200ms is shown.

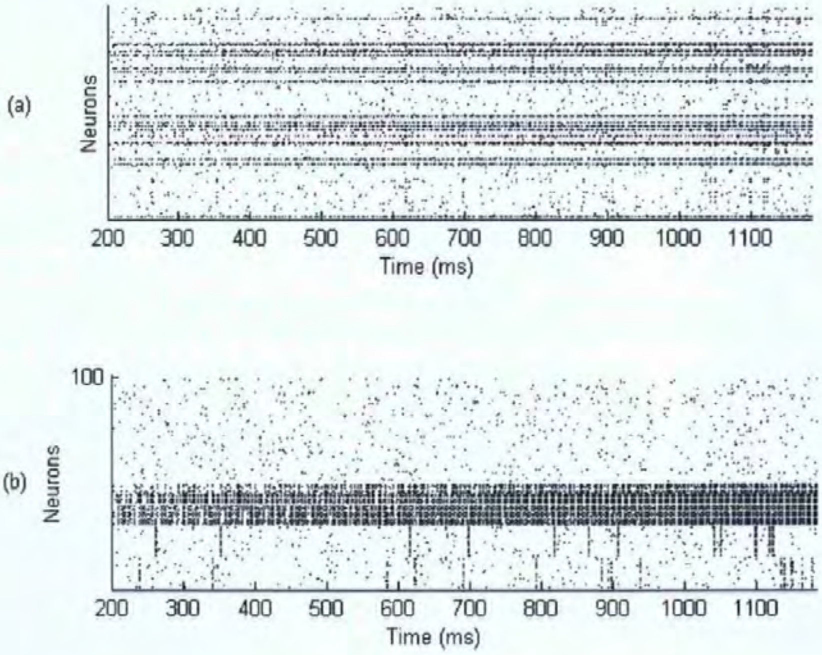


Figure 4.7: Raster plot comparing 100 Spike Trains of a trial lasting 20000ms, only the 200ms-1200ms portion is shown, with the trains (a) Random, (b) Ordered

Thus, the raster plot provides an overview of the data in the time domain. The overall firing pattern and 'quiet periods' are visible in the plot as areas of low spiking activity (primarily white areas). In addition, it is sometime possible to infer functional grouping from the raster plot. However, this is a very time consuming process and very difficult to do accurately.

4.3.1.2 The Inter-spike-Interval Superposition Plot (IISP)

The IISP [Awi88] plots the spiking activity of a neuron, due to a stimulus, over a number of recordings. Each spike that occurs during the recording is plotted on the IISP. The spikes are plotted as dots, with their position on the x axis representing the time delay of the spike time from the initial stimulus; and its position on the y axis denoting the time delay from the previous spike, see figure 4.8.

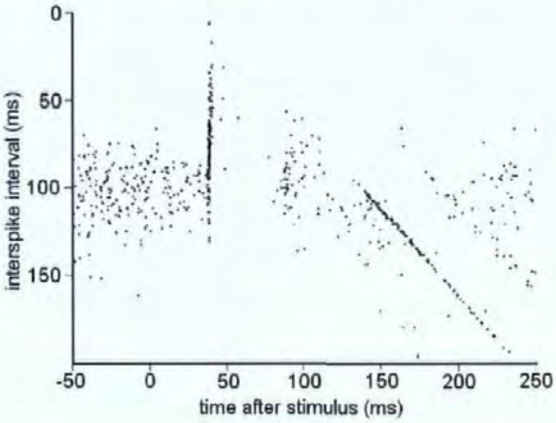


Figure 4.8: Example Inter-spike Interval Superposition Plot (IISP)[Awi97]

The IISP can then be visually examined to extract information regarding the spiking properties of the neuron, due to the applied stimulus. If the neurons have a tendency to generate a spike due to the applied stimulus, then clusters or lines of dots will be observed in the IISP. These lines are evident in the plot shown in figure 4.8, around 40ms after the stimulus is applied. It is also possible to observe that these spikes tend to occur rapidly, with short delays from the previous spike, noted from their position on the y axis.

Shorter inter-spike intervals are assumed to coincide with excitation, as a neuron that is receiving excitatory input is more likely to fire. Thus, plotting the reciprocal of the inter-spike interval can aid analysis[BLP68b, BLP68a].

4.3.1.3 The Joint Impulse Configuration Scatter Diagram

This diagram displays the inter-spike intervals between the spikes of three simultaneously recorded neurons[PGST75]. Each dot on the scatter diagram corresponds to an ordered trio of spikes, one from each of the recorded neurons. A dot at the centre of the plot denotes the fact that the three neurons fired simultaneously. Dots further from the centre have longer inter-spike intervals.

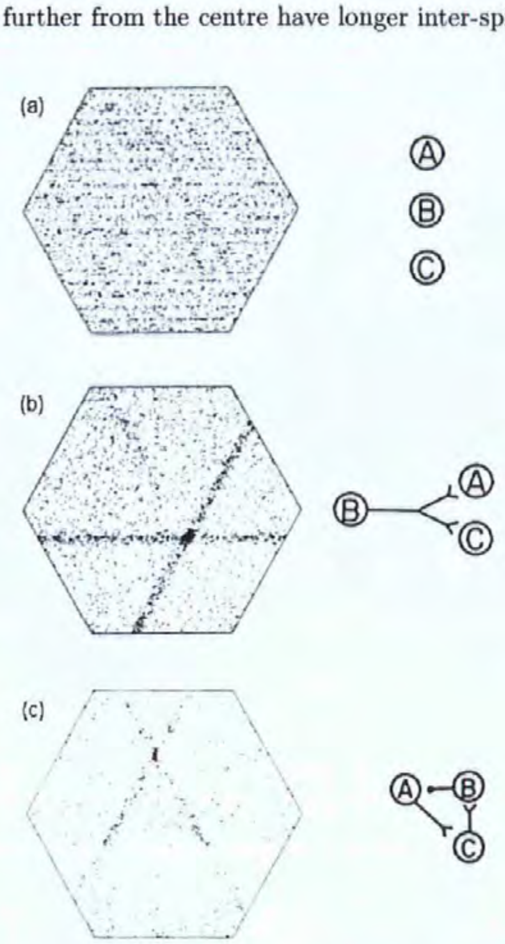


Figure 4.9: Example Joint Impulse Configuration Scatter Diagrams, with assembly diagrams. (a) The null case, three independent neurons (Figure 3(a) in [PGST75, page 277]). (b) Common input excitation case (Figure 5(b) [PGST75, page 283]). (c) Excitation and inhibition case (Figure 11(b) in [PGST75, page 293])

The scatter diagram enables the investigator to infer information about the functional relationships that exist between the three neurons represented. Neurons that have correlated spike trains will exhibit lines, or "ribs", of coincident dots in the scatter plot. The position of these ribs denotes the type of connections present. Figure 4.9 shows the effect of excitatory and inhibitory connections between the neurons under investigation (note the assemblies on the right hand side of the figure), ribs can be seen on the scatter diagrams. These ribs are darker, showing more incidents of trios in that area, for excitatory connections and are lighter, showing a reduced number of trios in that area, for inhibitory connections.

For example, two darker ribs can be observed in the second scatter plot (figure 4.9(b)), one running left to right across the lower portion of the plot, and the other running from the top right to bottom left of the plot. These ribs represent the excitatory connections between neuron B and neurons A and C. Likewise, in plot (c) or the diagram it is possible to observe to dark ribs, one running top left to bottom right and the other top right to bottom left. In addition, a light rib can be observed running from left to right across the lower portion of the plot. These darker ribs represent the excitatory connection for neuron A to C and neuron C to B; the lighter rib represents the inhibitory connection from neuron B to C.

In contrast the first scatter plot 4.9(a) has no discernible darker of lighter ribs indicating that the spike trains of the neurons represented exhibit no significant correlation.

4.3.2 Statistical Methods

There is a limit to the information that can be obtained by displaying the raw spike train data. However, the raw display methods provide a convenient means of assessing the data quality, by comparing the results of several recordings. The primary focus of the research in this thesis is the analysis of multi-dimensional spike train recordings. A number of statistical methods exist for the analysis of these datasets. However, these methods are mainly intended for the analysis of single spike trains or pairs of spike trains. These methods include:

- Single spike train analysis methods
    1. Peri-Stimulus Time Histogram
    2. Estimate of the Rate of Inhomogeneous Poisson Process by J^{th} waiting times
    3. Cumulative Sum of the PSTH
    4. Auto-correlation
- Pair wise spike train analysis methods
    1. Cross-Correlation
    2. Joint Peri-Stimulus Time Histogram
- Multiple spike trains analysis methods

1. Gravity Transform

4.3.2.1 Peri-Stimulus Time Histogram (PSTH)

The PSTH[GK60] is a measure of the spiking density of a spike train with a specified time resolution. The spike train is divided into a number of equal length time sections starting for the moment of stimulus presentation. The time section consists of several bins. The number of spikes that fall in each bin is summed. This process is repeated for a number of recordings from the same neuron. Figure 4.10 shows an example of PSTH computation, the time section has been divided into 14 bins.

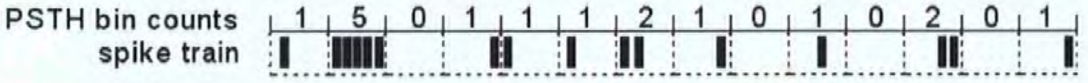


Figure 4.10: Example of a 14 bin PSTH computation for a single spike train

To facilitate comparison between PSTH's the value of each bin is normalised for bin-width and the number of trials used. The result of the PSTH can then be plotted as a histogram, see figure 4.11.

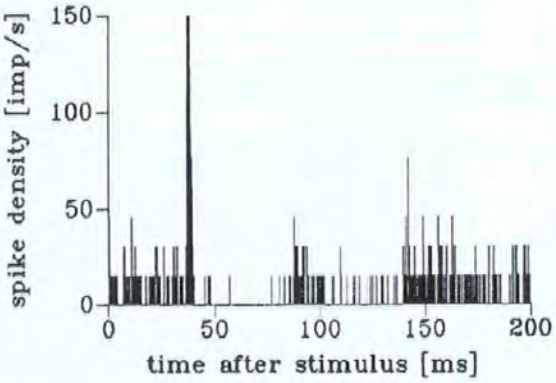


Figure 4.11: An example PSTH for the spike train recorded from a motor neuron, calculated with a bin size of $250\mu s$

From the PSTH shown in figure 4.11 it is possible to observe a peak in the plot at approximately 40ms. This peak indicates that the neuron has a tendency to produce a spike 40ms after the stimulus is applied.

However, there is an inherent problem with converting from continuous data to discrete data by binning. The binning process causes data regarding the relationship between spikes in separate bins to be lost, such as the relationship between the 7^{th} and 8^{th} spikes in figure 4.10 which are split over the 4^{th} and 5^{th} bins. In addition, the PSTH only represents a single spike train. If comparison between a number of trains is required, then numerous PSTH's must be calculated and compared. The PSTH is best suited to the analysis of the effect of a stimulus on a single neuron, as opposed to the analysis of the interconnection between many neurons.

4.3.2.2 Estimation of the Rate of an Inhomogeneous Poisson Process by J^{th} Waiting Times (RIPP)

The RIPP overcomes the main problem of the PSTH, which is the loss of data due to binning. The investigator must select a 'smoothing' parameter, the window size J, which is similar to the bin-width in a PSTH.

To calculate a RIPP the times of all spikes in all trials are considered. The time interval of size J covered by consecutive spikes is examined. This interval starts from the stimulus presentation. The spike density within that time interval is then calculated and normalised by the window size, the number of trials and the duration of the time interval covered. The spike density is then plotted against the midpoint of the interval to produce the RIPP. An example RIPP is shown in figure 4.12.

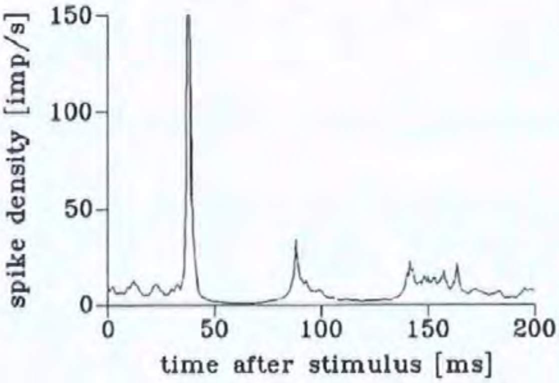


Figure 4.12: An example RIPP for a motor neuron recording, calculated with a window size of 10ms

Note that compared to the PSTH the RIPP presentation provides no additional information. Like the PSTH, the RIPP analyses the firing pattern of a single neuron due to a stimulus. Thus the RIPP is unsuitable for analysing the relationship between assemblies of neurons.

4.3.2.3 Cumulative Sum of the PSTH (CUSUM)

The CUSUM[Ell78] is a post-processing algorithm to enhance the PSTH. This algorithm is designed to enhance the detection of small peaks and troughs in the result[Awi97]. To calculate a CUSUM, the mean value of the pre-stimulus bins of the PSTH is calculated. Then the pre-stimulus mean is subtracted from all post-stimulus bins. Finally the resultant function is then integrated with respect to time and normalised.

4.3.2.4 Cross-correlation

A cross-correlogram [AG85, COS80] shows the number of spikes on one train (the target train) that fall within a defined period of spikes on a second (reference) train. The defined time, known as the correlation window, consists of a number of time bins of equal length either side of the reference spike, see figure 4.13.

If a significant peak (or trough) exists in the correlation window, the two trains are temporally correlated. However, it is possible for this function to show 'false' peaks meaning that the peak is not due to coupling of the neurons but the increased activity of one neuron. To overcome this problem, a number of correction methods can be employed.

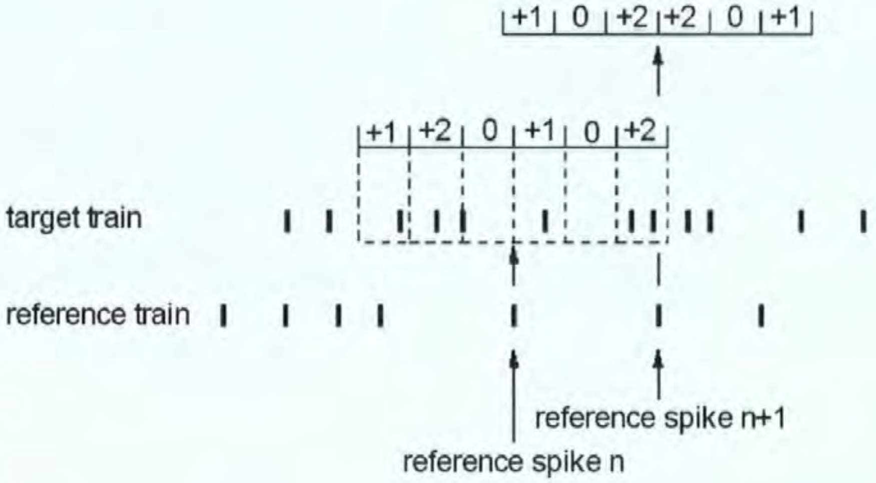


Figure 4.13: An example of a basic Cross-Correlation calculation, indicating how the spikes on the target train that occur within the correlation time fame of the current target spike are added to the bin values of the function.

Having computed the cross-correlation counting function the results must be corrected. This correction ensures that any peak exhibited in the output of the function is due to spike correlation and not due to random correlation or co-stimulation. This correction can be achieved in using several methods such as *trial shift correction (shift predictor)*, using data from second trial recorded for that experiment; *random train correction*, where a randomised version of the target train is used to correct for increased firing rates; *window normalisation*, where the window values are corrected for average spike rates; *Brillinger normalization and confidence interval*, where the correlation function is normalised and a statistical confidence interval is calculated.

Brillinger Normalisation The Brillinger normalisation [Bri79] takes account of spike train length and the number of spikes on both trains. The value of the counting function in each bin is then averaged over time and bin size.

The final correlation is then normalized to have a mean equal to one. Subsequently, a confidence interval can be defined and peaks within this interval are classified as random. Peaks exceeding the confidence interval are said to be significant, figure 4.14 shows an example.

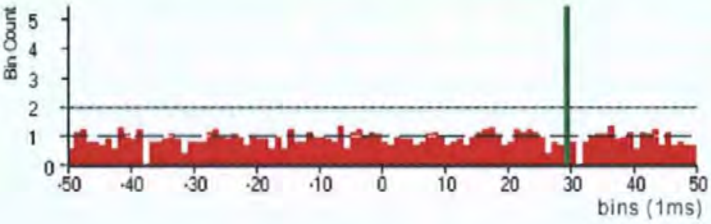


Figure 4.14: Example Brillinger Cross-correlogram, bin size 1ms, window size 100. The cross-correlogram exhibits a significant positively delayed peak at 29ms.

The cross-correlogram provides an effective means to analyse the temporal relationship between the firing patterns of a pair of neurons.

The analysis of any significantly sized neural assembly requires the production of a large number of analyses, for n spike trains the are $\frac{n!}{2!(n-2)!}$ pair-wise results.

For example, let us suppose there is a dataset made up of 25 spike trains. For this dataset there would be 300 unique cross-correlograms, each of which would need to be studied in order to understand the overall connectivity of the 25 neurons. The problem of the large quantity of resultant data is compounded by the standard methods of representation since, each cross-correlogram is plotted as a separate histogram, the scientist would be required to compare and contrast these 300 cross-correlograms in order to derive the structure of the interconnections in the neural assembly.

4.3.2.5 Auto-Correlation

Auto-Correlation is the cross-correlation of a spike train with itself [COS80]. This produces a symmetric distribution. If a neuron has been stimulated, it is likely that the spike train recorded from the neurons will exhibit a repeating pattern of spikes. This technique is used to detect periodicities within the spike train. The plot of the auto-correlation function can be visually inspected for repeating patterns or periodicity. An example auto-correlation is shown in figure 4.15, which represents the auto-correlation of a 20000ms spike train recorded from a simulated neuron.

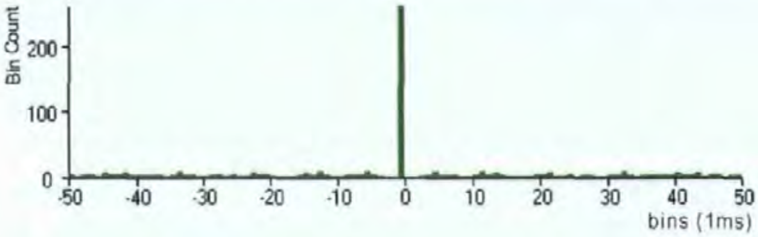


Figure 4.15: Example Auto-Correlation Function calculated for a 20000ms spike train recorded from a simulated neuron, note the symmetry of the negative bins with the positive bins.

4.3.2.6 Joint Peri-Stimulus Time Histogram (JPSTH)

The cross-correlogram is capable of indicating whether the target neuron has a tendency to fire within a certain time period before or after the reference neuron. However, it is impossible to know if this is true for the whole trial. Moreover, any abrupt change in spiking behaviour of the two neurons cannot be detected. Therefore the cross-correlogram is a one-dimensional data representation. It collapses the trial and only displays the correlation based on averages [MUL].

The JPSTH[IT00] shows the dynamics of correlation. Also the JPSTH is a more sensitive in detecting correlation. The final output shows a cross-correlogram of the two neurons, individual PSTH's and an overview of the spike distribution, see figure 4.16.

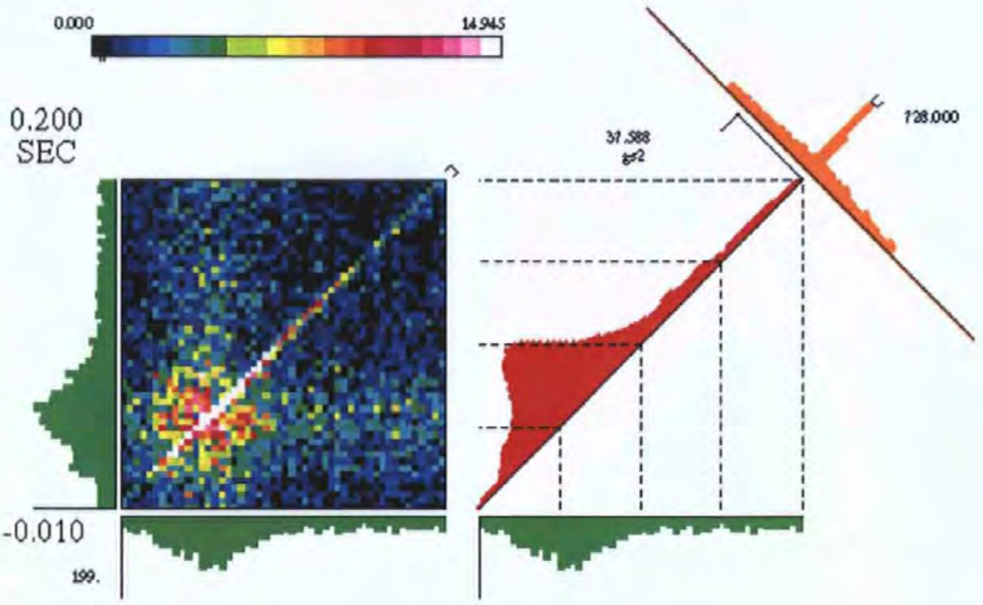


Figure 4.16: Example Joint Peri-Stimulus Time Histogram, for a pair of simultaneously recoded, stimulated neurons [MUL].

The JPSTH display shows the individual PSTH's for both spike trains on the horizontal and vertical axes of the main plot. The main plot displays a count of the number of coincident spikes. An individual element in the matrix is increment if a spike occurs on both trains at the time that element represents. The correlation along the main diagonal (bottom-left to top-right), and close by it, is of interest as this relates to the correlation within the neurophysiological time frame. For this reason the correlation on the diagonal is shown as a separate histogram to the right of the main display. The cross-correlation for the two spike trains is shown to the top right of the diagonal histogram.

As with cross-correlograms, correlation due to co-stimulation must be removed. This correction is commonly achieved by using a shift predictor [MUL].

4.3.2.7 Gravity Transform

The gravity transform[GA85, GPE85] is a method of analysis based on the principle of gravitational interaction of particles. Each neuron is represented by a particle. The movement of particles is described in n-dimensional space, where n is the number of neurons under investigation. All particles start equidistant. The gravitational force (or charge) exerted by a particle is defined by the spike train of the corresponding neuron. Each spike of the train contributes to the charge, which decays exponentially over a given time period. If two or more neurons spike coincidently, their corresponding particles will have an attractive force causing the particles to move closer together.

If two neurons have an above average synchrony, over time this would result in a strong attractive force between their corresponding particles. In turn, this force would cause the particles to aggregate in n-dimensional space. Gerstein [GA85] specifies that over time all particles will eventually collapse together into a single cluster, due to these forces. Thus, this artefact of the algorithm needs to be considered when analysing data using the Gravity Transform algorithm.

It is already established that significant synchrony can indicate synaptic coupling [BG00]. Therefore, aggregation of the particles can aid the definition of the assembly represented by the spike train dataset.

The Gravity Transform has three parameters. These are the charge increment, i, charge decay, τ, and an overall aggregation constant, a, that are arbitrarily chosen by the investigator. The effect of these parameters on the system is as follows. The charge increment i defines how much charge is added to a particle per spike in its corresponding neuron. This increment quantifies the effect of one spike upon the particle's charge. The value of τ specifies the decay time of the charge, defining 'how long' each spike can effect the system. The overall aggregation of the particles, a, controls the speed at which the entire collection of particles aggregate.

The output of the Gravity Transform is plotted in two ways: (i) as a distance graph, showing the Euclidean distance between each pair of particles in the system; (ii) as a two-dimensional plane selection. These outputs are shown in figures 4.17 and 4.18, for an assembly of 10 neurons, with the following connections $s_1 = \{c_{ij} : i = 1, j = 3, 5, 7, 8\}$, $s_2 = \{c_{ij} : i = 2, j = 4, 6 \ldots 8\}$ and neurons 9 and 10 being unconnected.

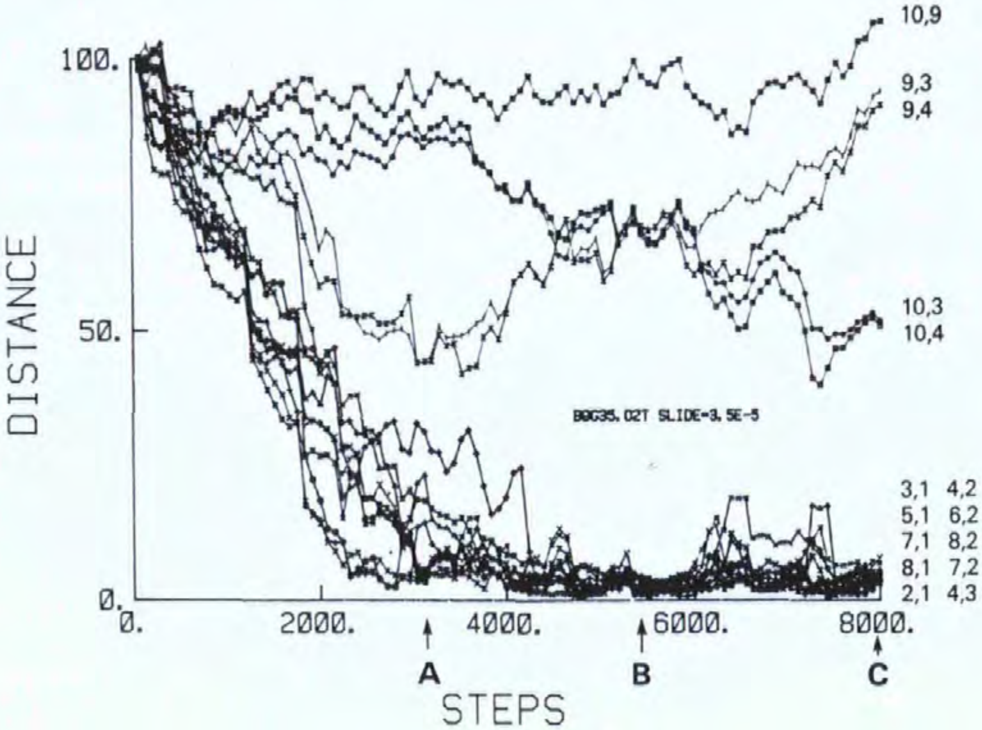


Figure 4.17: An example distance graph output of the Gravity Transform computed for the example dataset (figure 7 in [GPE85]

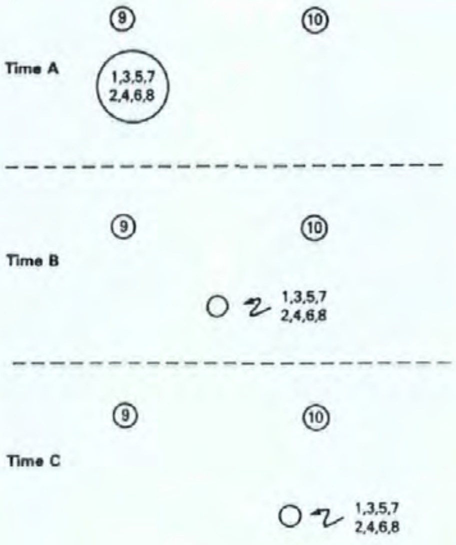


Figure 4.18: Sub plane selection plots of the Gravity Transform computed for the example dataset, taken a times A, B and C as indicated on the distance graph (figure 8 in [GPE85]

The gravity transform defined by Gerstein has made a significant contribution to the field, however there are some difficulties with (a) the display of output data for a large number of neurons and (b) the undefined selection of values for attributes increment (i), charge decay (τ) and overall aggregation constant a.

4.4 Software Support for Neural Analysis

Commonly, many investigators produce their own algorithms to implement the common analysis methods, with specific adaptations to suit their datasets, using programs such as MatLab. However, a number of libraries of functions are available for use with MatLab, providing methods to analyse spike train dataset recordings. In addition, specialist packages exist for the analysis of temporal relationships within spike train datasets.

4.4.1 MatLab

MatLab is a collection of Mathematical and Statistical routines. A large number of researchers use *MatLab* as a basis for their computation, as it has many useful built-in functions. Also the creation of more complex algorithms is relatively simple. In addition to its computational strength, *MatLab* is capable of producing graphical displays of data to aid in analysis.

The main drawbacks of algorithms implemented in MatLab are that computation times can be considerable, as algorithms run on top of the MatLab engine. In addition, transfer of data between algorithms implemented by different investigators can be difficult as common standards are not always adopted. Moreover, as MatLab is commercial software anyone wishing to use packages developed for MatLab must purchase a license for the product.

MatLab does not provide any methods specifically targeted at the analysis of temporal synchrony within spike train datasets. However, it provides a good, if expensive, base for the development of tools for spike train analysis. Two suites of tools for use with MatLab are reviewed in the following sections.

4.4.1.1 DataMunch

DataMunch[Dat] is a collection of free open-source-code routines for use with MatLab. DataMunch can read data files in a number of formats, including Spike2 and DataWave UFF1. Once a data file is loaded the investigator can select what section of the data to analyse. Moreover, data from a number of files can be analysed automatically. DataMunch will produce results from a selected portion of a file or group of files.

DataMunch is aimed at single cell analysis and can produce spike rasters, spike histograms and perform comparisons between trial segments. The output of the results of an analysis session can be summarised and exported, to packages such as Microsoft Excel. In addition, the investigator can filter the results and recall individual plots to examine them in more detail.

However, DataMunch has limited capacity to analyse the relationship between spike trains.

4.4.1.2 MEATools

MEATools[oF] is a collection of Matlab based algorithms for analysing spike trains, from extracellular recording. These algorithms are designed to work with recordings from 8x8 Multi-Electrode Arrays.

MEATools provide methods to sort and filter the recorded spikes based on shape and timing parameters. In addition, there are methods to display a raster plot of a spike train, calculate the PSTH of a spike train and to perform spike rate analysis of the data. The investigator can also produce a graphic representation of the raw data, by viewing a colour rendered representation of the data matrix. This matrix can be animated to show the change in voltage (action potential) over time.

MEATools are primarily aimed at the analysis of stimulus driven results and currently provide no methods to analyse continuously recorded data. Moreover, the tools are designed to support the analysis of single cell data, the support for the analysis of temporal relationships between multiple simultaneous spike trains is limited.

4.4.2 Cortex Windows Suite

The *Cortex Window Suite*[Bau] is a collection of Windows-based programs. The suite is designed to analyse the results of the Cortex simulation package. The output of the analysis can be displayed on screen, saved to a text file or output in Microsoft Excel file format. The investigator can filer the dataset to select what trial and spike trains to analyse.

The investigator can generate a raster plot of the selected data and PSTH's can also be calculated for the data. Both the raster and PSTH can be displayed on the screen simultaneously, permitting the contribution of spikes, on different trials, to the PSTH to be examined. In addition, the relationship between two spike trains can be analysed using a JPSTH. The program also provides a 'wizard' so that results for a number of spike trains can be produced automatically.

The Cortex Windows Suite is designed to analyse the data generated from the Cortex program and is aimed at stimulus driven analysis. Moreover, the analysis of data from other sources is not supported and the analysis of multiple neurons is limited.

4.4.3 DataWave Technologies

DataWave Technologies[Tec] produce a number of software packages to support the analysis of neuronal datasets including the *Experimenter* package which contains tools for data acquisition, real-time analysis, experimental control and graphical display. Experimenter provides support for the recording and separation of up to 128 channels of waveforms.

The package can produce raster plots of the sorted data, PSTH's and Cross-correlograms. The package provides methods to analyse both stimulus driven and continuous recordings. However, there is limited support for the analysis of simultaneous recordings.

4.4.4 NeuroExplorer

NeuroExplorer is a self-contained analysis package for Neurophysiological data [neu]. The package is capable of reading a number of standard file formats and performing common analytical processes on the data.

The package can calculate the 'standard' histograms. However, instead of calculating the results for each spike train one at a time, *NeuroExplorer* calculates and simultaneously displays all results for the file. This feature is useful if a number of trains contained in the same file are to be analysed. It is also possible to perform other analyse for example, Rasters and JPSTH. However, there is limited support for the analysis of multiple simultaneous recordings.

4.4.5 Neuronal Time Series Analysis (NTSA)

The *NTSA Workbench*; from the Neuronal Pattern Analysis Group, Beckman Institute, University of Illinois[NPAG] implements a number of tools for the analysis and visualization of time-series data. The workbench provides methods to produce the standard raster plot and histograms. In addition, the gravity transform is implemented providing support for the analysis of neuronal assembilies.

A 'Digital Brain Atlas' is provided to display the neuroanatomical information permitting the investigator to view the change in electrical activity at recording sites in the brain.

4.5 Requirements for Software Support

The analysis and identification of temporal relationships between simultaneously recorded spike trains is key to understanding many aspects of the brain's function. The ability of scientists to record neural activity of several hundred simultaneous spike trains is now possible[WM93]. However, the methods to analyse these growing datasets are still based on the analysis of individual pairs of neurons[BKM04].

Thus, it is clear that additional support is required to aid in the analysis of temporal synchrony within multi-dimensional spike train datasets.

4.5.1 Possible system approaches

In general, there are two main styles of system that can be developed to support these needs. The first method is to attempt to specify all the user's requirements, completely and fully, and develop a tool that provides complete support for these requirements. This method would result in the ideal support tool. However, as user requirements are continually expanding, it is highly likely that this type of static system would swiftly become obsolete.

The second, and preferred, method is to provide a Toolbox of analysis, manipulation and display techniques, and an investigation environment in which these techniques can be combined. In addition, it is critical that the Toolbox supports the addition of new modules for use in the investigation environment, to support expanding needs. This approach, will produce a dynamic and expandable support environment, thus creating an effective tool that is capable of expanding to support the analyst's increasing needs.

The second approach has been adopted for this project, developing a Toolbox of methods and an investigation environment. The overall principle of this module structure is shown in figure 4.19.

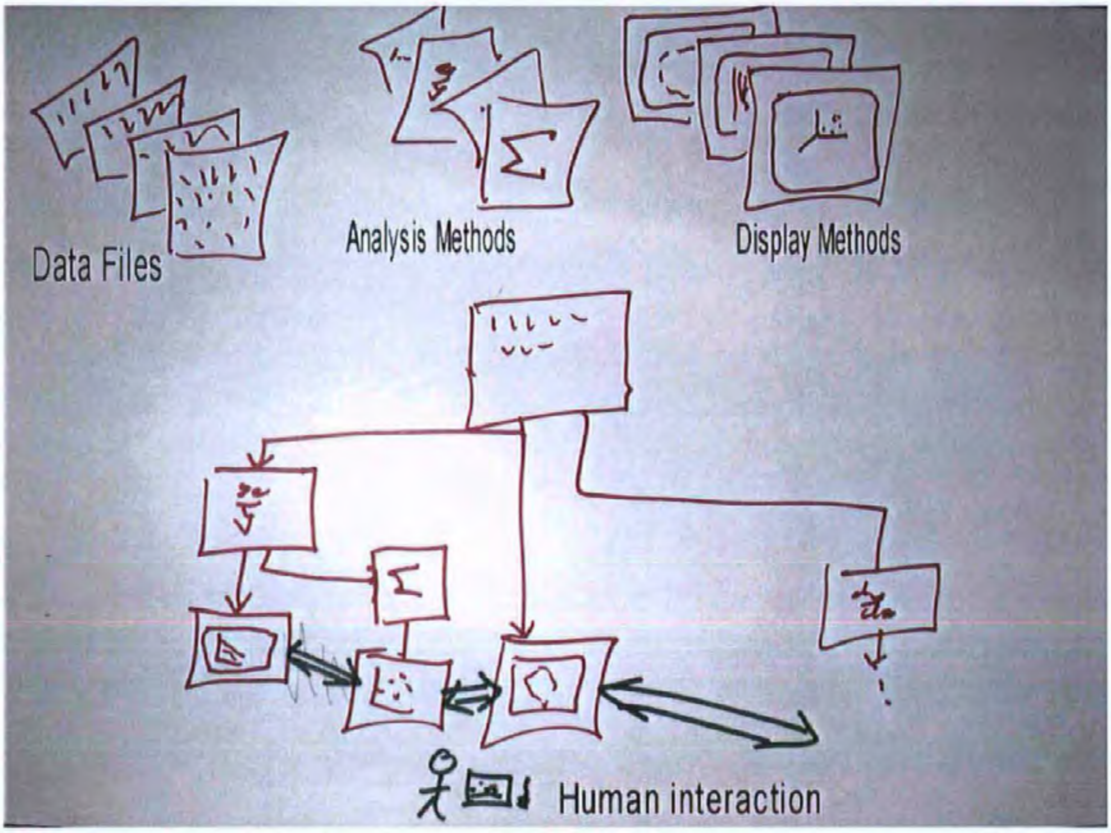


Figure 4.19: Diagrammatic representation of the Toolbox and Investigation Environment principle

In this environment the analyst can load a number of datasets to analyse. These datasets can then be used as the source for a number of analysis methods, which in-turn can form the input to other methods. The result of these methods can then be sent to various display methods. The different display methods are linked to form a multiple-view environment. This methodology permits the investigator to construct tailored analyses.

4.5.2 Hardware/Software Platforms

Currently, investigators working with multi-dimensional spike train datasets are using a number of computer platforms. These platforms include Windows, various Unix and Linux systems, Apple Mac and Silicon Graphics workstations. For this reason it is critical that the Toolbox is platform independent.

4.5.3 Construction of the Toolbox and Investigation Environment

This section describes the key requirements defined during the initial design of this Toolbox. Note that the methods defined in this thesis form part of the Visualization of Inter-Spike Association project, at the University of Plymouth [SWB]; thus only a subset of these requirements and methods have been implemented and tested.

1. **Toolbox should be user centred**

   A key principle of Visualization is efficient user interaction and control. Thus, the Toolbox must be user centred, providing an environment in which the analyst can explore their datasets. Within this environment, the user could construct a tailored Visualization system from the Toolbox components. By interacting with and refining this system, the user should be able to resolve the dataset to a plausible function connection architecture.

2. **Toolbox should adopt a modular approach**

   To provide a flexible and expandable system, each method in the Toolbox must be implemented as an individual module. These modules should all use standardised input and output formats and conform to a standard set of interaction primitives.

3. **Toolbox should use a visual programming environment**

   To facilitate the integration of different Toolbox components in order to form useful analysis pipelines, a visual programming interface will be developed. This interface will permit users to link graphical objects, representing the different methods and datasets, together to form an analysis pipeline. This style of interface removes the necessity for the user to be familiar with the implementation details of each method.

4. **Toolbox should support multi-format I/O**

   As identified previously, different investigators use different file formats to record multi-dimensional spike train data. Thus, a standard internal format will be adopted and the Toolbox will provide methods for importing and exporting data in various formats.

5. **Data Manipulation**

   The Toolbox must provide methods to analyse the temporal synchrony within multi-dimensional spike train recordings. The Toolbox should provide implementation for the current analysis methods. In addition, new methods specifically aimed at the simultaneous analysis of assemblies need to be developed.

6. **Data Presentation**

   Dynamic and user controllable display methods are required; these methods should support the key interaction principle of Shneiderman's Mantra[Shn96]. Simple methods to display the results of these analysis methods and the raw spike train data are required. Additionally, powerful dynamic representations are required, to provide an overview of large quantities of experimental data simultaneously.

7. **Software Platform**

   The Toolbox will be implemented in Sun Microsystem's Java. The development and testing will be conducted on a Windows PC platform (Microsoft Windows 2000/XP).

8. **Hardware Platform**

   The implementation and evaluation of the methods detailed in this thesis will be undertaken on two different systems.

   (a) Intel Pentium III 999 MHz, 512Mb Ram, triple monitor output (1260x1024)

   (b) AMD twin Athlon 2000+ MP, 2Gb Ram, triple graphics cards running two XGA data projects (1024x768) and Primary monitor (1600x1200). The data projectors form a passive projected 3D system on an 8'x6' screen.

Chapter Summary

In this chapter the general principles of temporal coding for information processing within the brain were discussed. Further, the current methods for analysing multi-dimensional spike train datasets were presented. The shortcomings of these methods and the associated software is the limited support that is available for the analysis of synchronous spiking activity.

Finally, the requirements for a fully interactive exploratory environment based on the principles of Information Visualization were proposed. The following chapter explores these requirements and presents software solutions.

Chapter 5

Prototype Tool to Support the Analysis of Neurophysiological Data

Summary

In this chapter, the Toolbox developed for this research is presented. This presentation includes file formats supported, analysis and manipulation methods, and presentation methods.

This chapter describes the methods contained in the Toolbox. This Toolbox has been implemented using Java and Java3D. Java was chosen for the following reasons:

- Platform Independence: Java virtual machines are available for all operating systems, including Windows, Mac, Unix and Linux
- Object Orientation: Java is an object-oriented programming language
- Availability: the Java Development Kit (JDK) and run-time environments are available for free
- Graphics Capability: Java contains comprehensive 2D and 3D (Java3D) graphics facilities

Each of the different computational and display methods, used within the Toolbox, has been implemented as a separate package of objects. Overall the methods fall into three categories:

1. Data filters
2. Data manipulations methods
3. Data presentation methods

5.1 Data Filters

5.1.1 Neurophysiological Data Input

It is anticipated that the Toolbox will be widely used for analysing experimental evidence from multiple simultaneously recorded electrodes. This data consists of several spike trains which can be considered to be multi-dimensional point processes. Different neurophysiological laboratories use different formats to represent these recordings. The Abeles format (see §4.2.1) was developed by Professor Moshe Abeles (Jerusalem) and is one of the most common formats for representing this type of data.

The approach taken in the Toolbox is to transform input data from any format to the binary representation, which is a universal and convenient representation for all Toolbox calculations. The current version of the Toolbox includes a filter that converts a set of spike trains from an Abeles format into a Boolean array representing the spike trains. Each spike train is stored as a separate array of Boolean values, one value representing each time unit of the recording. Note, when spike train data is stored, it is stored at discrete time steps δ (in all the examples shown in this thesis, δ = 1ms), thus spikes can only occur at prescribed times. If a spike occurs at a specific time unit, then the value representing that time unit is set to true (=1).

Figure 5.1 shows a fragment of an Abeles file, containing three spike trains. Figure 5.2 shows the content of the file in figure 5.1 represented as a Boolean array. The file fragment contains the recordings of three neurons over 20ms, the descriptor (first digit of each triple) value for a spike event is 1.

```
1,3,2
1,1,3
1,2,0
1,1,5
1,3,3
1,1,2
1,3,0
1,2,1
1,3,1
1,3,2
1,1,1
```

Figure 5.1: Fragment of a trial of an Abeles format spike train file containing the recordings from three neurons, with spike event descriptor value = 1

Note that each spike train is represented by a separate array of Boolean values. In figure 5.2 each array is displayed on a separate line.

```
0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
0 1 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0
```

Figure 5.2: The result of converting the Abeles format spike train file fragment in figure 5.1 into a Boolean array of spike trains

All methods in the toolbox that use spike train data utilise the Boolean array format.

5.1.1.1 Multi-Dimensional Matrix Data

Some modules in the Toolbox produce output data files. These files can be used as input to other modules. To facilitate communication between the Toolbox modules a standard file format has been adopted, this format is detailed below.

The multi-dimensional matrix input filter reads the data output from a number of Toolbox functions. Specific filters have been implemented to read the entire file and to read the file at a specific resolution. In addition, it is possible to specify a matrix for use with calculations as well as reading the file's content.

Multi-dimensional matrix file format

All matrices within a specific data file produced by a toolbox method are the same size. Each matrix has N rows (variables) and M columns (dimensions). A variable is an M dimensional vector of real numbers and the coordinates of this vector are elements of the row in the matrix.

The header of the output file contains details about the number of matrices contained in the file and their composition. These values are written at the top of the file. The format of this file header is:

- Number of variables (N)
- Number of dimensions (M)
- Number of matrices (S)

Each row of the matrix is written as a line of space separated values (double precision floating point numbers). Each matrix in the file is separated by a blank line. Figure 5.3 shows the header and a data matrix of an example output file. The file contains 200 data matrices, each consisting of six, four dimensional variables. The figure shows the file header and an example data matrix.

6				Number of *variables* (N)
4				Number of *dimensions* (M)
200				Number of *matrices* (S)
2.5	15.32	12	3	Coordinates of variable one
5.9	37	11	30.6	Coordinates of variable two
1.4	2.5	13	12.3	Coordinates of variable three
12.8	4.5	12.5	6.9	Coordinates of variable four
20.5	36.8	12.3	15.9	Coordinates of variable five
7.81	36.4	12.1	6.5	Coordinates of variable six

Figure 5.3: Excerpt of a multi-dimensional matrix file, showing the file head and an example data matrix, with comments

5.2 Data Manipulation Methods

The following sections detail the current data manipulation methods contained in the Toolbox.

5.2.1 Cross-Correlation

Several modifications of the cross-correlation function have been implemented. Every implementation takes two spike trains, the bin size and the number of bins as input. The spike trains are supplied as two arrays of Booleans and the constants are Integers. The output of the correlation function is an array of double values, one value for each bin.

In the Toolbox, which is based on the principle of object-orientation, the basic counting function forms the super class for all cross-correlation methods. Inheriting from this super class are classes implementing cross-correlation utilising different correction methods. In total there are five variations of the cross-correlation function implemented.

Basic counting algorithm

The basic counting algorithm counts the number of spikes on the target spike train that fall within a defined time frame of a spike on the reference spike train (see §4.3.2.4).

Spike frequency normalized cross-correlation

The spike frequency normalization takes the result of the basic counting function and normalizes each bin value for the average spiking frequency of the reference spike train. The average spiking frequency of the reference spike train is the number of spikes in this spike train divided by the total time epoch.

Brillinger normalised cross-correlation

The Brillinger normalization[Bri79] function takes the result of the basic counting function and normalises each bin for the number of spikes in each train and the time of the trials involved, see §4.3.2.4.

Random train corrected cross-correlation

This correction method uses the basic counting algorithm to produce two cross-correlation functions. Initially, the cross-correlation of the specified spike trains is calculated. Secondly, the cross-correlation of the reference spike train and a randomised (shuffled inter-spike intervals) target spike train is calculated. Finally, to produce the random spike trains, the inter-spike intervals of the target spike train are randomised. The resultant spike train contains the same number of spikes as the original and has the same inter-spike intervals. However, the distribution of the inter-spike intervals in the resultant spike train is random. The algorithm used to randomize the inter-spike intervals is as follows:

```
private boolean[] randomize(boolean[] spikes){
    /** Set up storage for randomised spike train */
    boolean[] randomSpikes = new boolean[spikes.length];
    /** Set up test array to see if position is occupied */
    boolean[] test = new boolean[spikes.length];
    for (int i = 0; i < test.length; i++){
        test[i] = false;
    }
    /** Test to track if current value has been
     * successfully placed */
    boolean done = false;
    /** Variable to hold new random location on spike */
    int pos;
    /** Progress through target spike train */
    for (int i = 0; i < spikes.length; i++){
        done = false;
        /** Find new position */
        while (!done){
            pos = (int) (Math.random()*spikes.length);
            /** Is the current position occupied? */
            if (!test[pos]){
                randomSpikes[pos] = spikes[i];
                test[pos] = true;
                done = true;
            }
        }
    }
    return randomSpikes;
}
```

Thus, the randomised target spike train has the same number of spikes as the original target spike train. However, the distribution of the spikes is random. The cross-correlation using the randomised train is then subtracted from the cross-correlation of the original spike trains. This is to correct the original cross-correlation for increased spiking frequency.

Trial-Shift corrected cross-correlation

Similar to the random train correction methods, the trial-shift method computes two cross-correlations using the basic counting algorithm. Initially, the cross-correlation of the specified reference and target spike trains is calculated. Secondly, the cross-correlation of the reference spike train and a specified, randomly chosen, time shifted target spike train is calculated. Finally, the bin values of the second cross-correlation are subtracted from the first, to account for influence of increased spiking.

5.2.1.1 Examples of cross-correlation functions for different neuronal circuits

Example One: Linear Coupling

This example shows the output of the cross-correlation function, with the Brillinger normalization correction and confidence interval applied, for neurons that are directly coupled. In addition, the effect of intermediate neurons is shown. The dataset was generated using an enhanced integrate and fire neuronal generator [Bor02] and the trial lasted for 20000ms. The assembly of neurons is shown in figure 5.4, note all cross-correlation functions in this example were calculated using a bin size of 1 and a window size of 200 bins. The output of the cross-correlation functions have been plotted using the cross-correlogram display methods (for more information see §5.3.1).

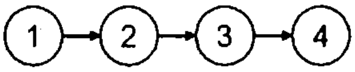
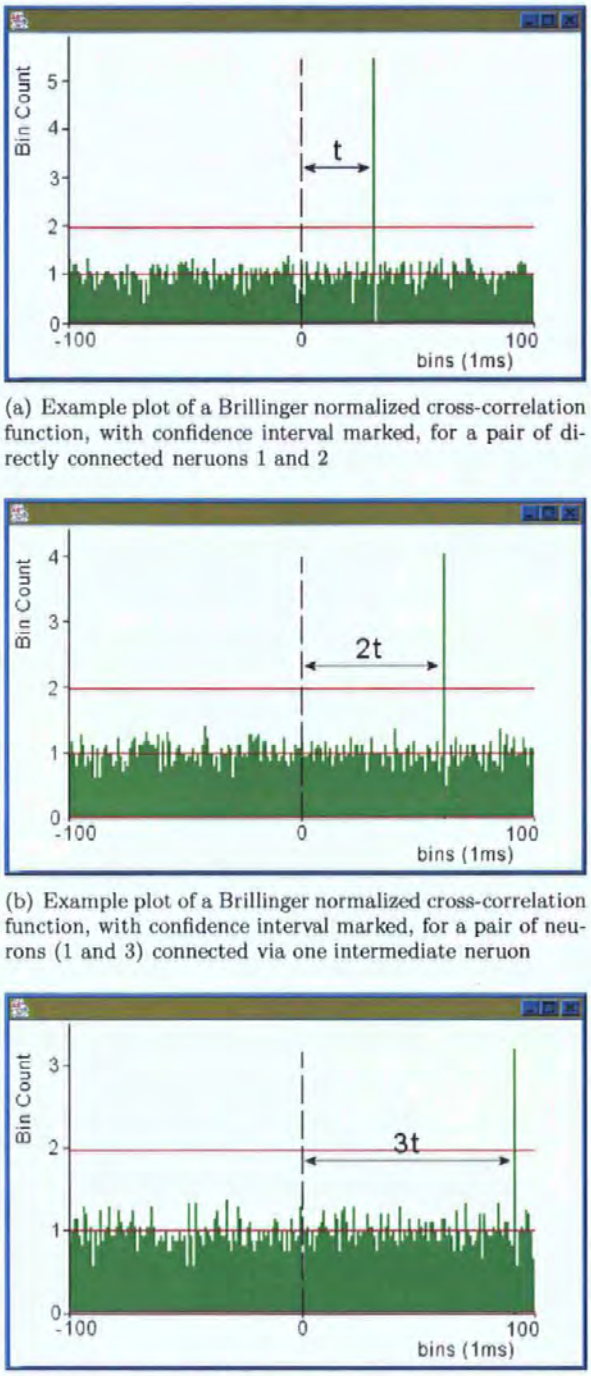


Figure 5.4: Assembly of four neurons used to illustrate the result of cross-correlation function on directly connected neurons

Figure 5.5(a) shows the output of the cross-correlation function between the spike trains of neurons one and two.



(a) Example plot of a Brillinger normalized cross-correlation function, with confidence interval marked, for a pair of directly connected neruons 1 and 2

(b) Example plot of a Brillinger normalized cross-correlation function, with confidence interval marked, for a pair of neurons (1 and 3) connected via one intermediate neruon

(c) Example plot of a Brillinger normalized cross-correlation function, with confidence interval marked, for a pair of neurons (1 and 4)connected via two intermediate neurons

Figure 5.5: Example cross-correlations for directly connected neurons

A peak in the output can be observed, which has a positive delay; the peak is present in a positive bin, to the right of zero. This indicates that the target neuron has a tendency to spike after the reference neuron. If the peak had a negative delay, the peak would occupy a bin before zero, thus, the connection would be reversed. Note that the connection delays are artificially high in this example in order to illustrate the effect of intermediate neurons.

In addition to specifying the direction of the connection, the delay of the peak can also aid in identifying if the connection is directly between the two neurons in question or via an intermediate neuron. The effects of intermediate neuron are shown in figure 5.5(b) and figure 5.5(c).

Figure 5.5(b) displays the cross-correlation function for the spike trains of neurons one and three. Note two things: firstly, the peak is lower (scales vary), the correlation between the spike trains involved is less; secondly the delay of the peak is approximately twice that of the delay of the peak in figure 5.5(a).

Figure 5.5(c) displays the cross-correlation function for the spike trains of neurons one and four. Again note the peak is lower (again scales vary) and the delay of the peak is approximately three times that of the delay of the peak in figure 5.5(a).

Thus, the time delay of the peak can assist in identifying if a correlation is due to a direct connection between two neurons, or, if the connection is via an intermediate neuron.

Example two: Common Input coupling

This example illustrates the output of the cross-correlation functions with the Brillinger normalization and confidence interval applied, for a pair of neurons that have correlated input. The dataset used in this example was generated using an Enhanced Integrate and Fire neuronal generator [Bor02] and the trial lasted for 20000ms. The assembly of neurons is shown in figure 5.6, note the cross-correlation function in this example was calculated using a bin size of 1 and a window size of 100 bins. The output of the correlation function has been plotted using the cross-correlogram display methods, detailed in §5.3.1.

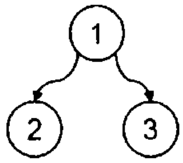


Figure 5.6: Assembly of three neurons used to illustrate the result of cross-correlation function on neurons with correlated input

Figure 5.7 shows the output for the Brillinger normalized cross-correlation function for the spike trains of neurons two and three, in figure 5.6.

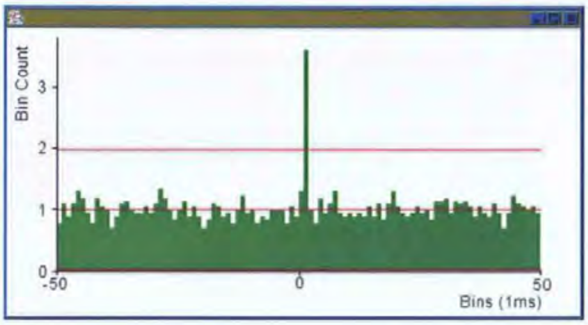


Figure 5.7: Example output of basic cross-correlation function for spike trains from a pair of common input neurons

A significant peak in the output can be observed, this peak is in the centre of the correlation window, at zero. This indicates that the reference and target spike trains have a tendency to spike synchronously. Thus, this indicates that the two neurons involved are likely to be receiving the same or correlated input.

5.2.2 Gravity Transform

The Gravity Transform[GPE85] was proposed to analyse the synchronous spiking within assemblies of neurons.

5.2.2.1 The Algorithm

The Gravity Transform algorithm as defined by Gerstein at al.[GPE85] has been implemented. There are n simultaneously recorded spike trains, of time $[0, T]$. If the i^{th} spike train is represented by spikes at times $T_1, T_2, \ldots, T_k$ the 'charge' on the i^{th} particle, corresponding to the spike train is described by the following rule: each spike contributes a quantity of charge a, which decays to zero exponentially over a given time τ. The charge on a particle at any given time is the sum of the effect of all preceding spikes. Thus, for a given time t, the charge of particle i is given by:

$$q_i(t) \quad = \quad \sum_{m=1}^{k} Q(t - T_m) - \lambda_i \tag{5.1}$$

$$Q(x) = \begin{cases} a & if\ x = 0 \\ ae^{\frac{-x}{\tau}} & if\ x > 0 \\ 0 & otherwise \end{cases} \tag{5.2}$$

where the average firing rate of neuron i is $\lambda_i = \frac{a\tau k}{T}$.

The charge function for each particle is pre-calculated and stored with time step Δ, thus $q_i(t_j), t_j = j\Delta, j = 0, 1, \dots$, and intermediate values of charge are linearly interpolated from the stored values.

The interaction dynamics of the particles in n-dimensional space are governed by the following equations:

$$\frac{dx_i^k(t)}{dt} = bq_i(t)\sum_{j=1}^{n} q_j(t)\frac{x_j^k(t) - x_i^k(t)}{R_{ij}(t)} \quad k = 1, 2, \dots, n;\ i = 1, 2, \dots, n \tag{5.3}$$

where $(x_i^1(t), \dots, x_i^n(t))$ represents the position of particle i at time t; a and τ are constants as described above; constant b controls the overall aggregation of the system; R_{ij} is the Euclidean distance between particles i and j:

$$R_{ij} = \sqrt{\sum_{k=1}^{n}(x_i^k - x_j^k)^2} \tag{5.4}$$

All particles in the system are initialised by the following rule and thus start equidistant:

$$x_i^m(0) = \begin{cases} 100 & if\ i = m \\ 0 & otherwise \end{cases} \tag{5.5}$$

5.2.2.2 Improvements to accuracy

Consider an ordinary differential equation, or a system of first order differential equations:

$$\frac{dy}{dx} = f(x, y) \tag{5.6}$$

In the original algorithm [GPE85] the numerical solution was achieved by using a small fixed time step, h, the Euler method:

$$y_{n+1} = y_n + hf(x_n, y_n) \tag{5.7}$$

This integration method has the advantage of being simple. However, the method can have accuracy and speed problems. Therefore, a more appropriate method of integration could be used. For example, the *Runge-Kutta* fourth order method given by the formula:

$$\begin{aligned} k_1 &= hf(x_n, y_n) \\ k_2 &= hf(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}) \\ k_3 &= hf(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}) \\ k_4 &= hf(x_n + h, y_n + k_3) \\ y_{n+1} &= y_n + \frac{k1}{6} + \frac{k2}{3} + \frac{k3}{3} + \frac{k4}{6} + O(h^5) \qquad (5.8) \end{aligned}$$

The Runge-Kutta method requires four evaluations of the right hand side equations per time step (h). However, the use of weighted intermediate points allows for a larger h, in comparison to the Euler method. Thus, usually, making the algorithm more efficient and produces a lower cumulative error.

In addition to the basic Runge-Kutta algorithm, an adaptive step size has been used to increase the speed and accuracy with which the integration is executed. The basic premise of adaptive step size is:

> *"Many small steps should tiptoe through treacherous terrain, while a few great strides should speed through smooth uninteresting countryside."*[Wil92]

The integration will progress quickly, with a larger step, when there is little change in the data values. However, the step size is reduced to maintain accuracy when there is significant change in the data values.[Wil92]

The Runge-Kutta algorithm was adapted from sections 16.1 and 16.2 of Numerical Recipes in C[Wil92].

5.2.2.3 Example output of the Gravity Transform

A Gravity Transform analysis was performed on a set of spike trains generated for the neuronal assembly shown in figure 5.8.

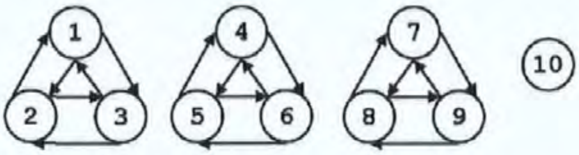


Figure 5.8: Specification of the connections between the 10 neurons used as input to the spike train generator.

The spike train dataset was generated for 5s and had an average firing rate of 0.2670 spikes/s. Figure 5.9 shows a raster plot of 200ms of the spike trains generated.

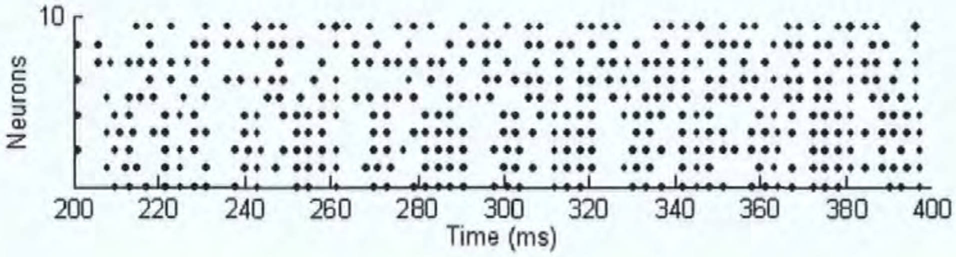


Figure 5.9: A raster plot depicting a portion of the 10 spike trains generated for the neuron assembly shown in figure 5.8.

The Gravity Transform of the spike trains was computed with charge increment $a = 0.3$, charge decay $\tau = 0.4$ and overall aggregation $b = 1$. The output of the computation is plotted as a graph of the Euclidian distance between all pairs of particles. This graph is shown is figure 5.10.

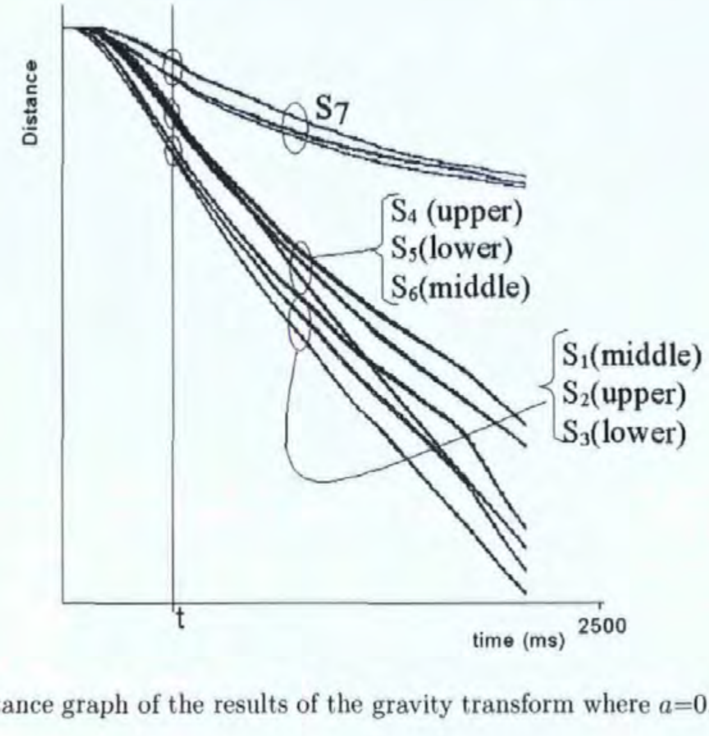


Figure 5.10: Distance graph of the results of the gravity transform where a=0.3, τ=0.4 and b=1

It is possible to observe that the distance pair forms three distinct groups. These groups are distinguishable and indicated at time t on the graph in figure 5.10. The top group represents the distances between neuron 10 and all the other neurons (1 to 9). The lower cluster of distances, indicated at time t, represents all the intra-group distances and the middle group, also indicated at time t, represents all the inter-group distances.

The final intra-group distances, $S_1 = \{d_{12},\ d_{13},\ d_{23}\}$, $S_2 = \{d_{45}, d_{46}, d_{56}\}$ and $S_3 = \{d_{78},\ d_{79},\ d_{89}\}$ show the three groups of neurons aggregating. The final inter-group distances, $S_4 = \{d_{ij} : i = 1, \ldots, 3,\ j = 4, \ldots, 6\}$, $S_5 = \{d_{ij} : i = 1, \ldots, 3, j = 7, \ldots, 9\}$ and $S_6 = \{d_{ij} : i = 4, \ldots, 6,\ j = 7, \ldots, 9\}$, show the distances between the three aggregating groups. The final distances between neurons 1 to 9 and neuron 10, $S_7 = \{d_{ij} : i = 10,\ j = 1, \ldots, 9\}$, show that the solitary neuron has no tendency to group with any of the other neurons.

From this examination of the distance graph it is possible to identify the functional relationships, based on synchronous spiking, between the neurons, shown in figure 5.8. The synchronous spiking of the neurons in the three small groups (1, 2, 3), (4, 5, 6) and (7, 8, 9) is apparent as is the absence of synchronous firing of neuron 10 with the other neurons. It is therefore possible to show that the results of the Gravity Transform have a high correspondence with the underlying neuronal assembly.

5.2.2.4 Calculation Constants

The gravity transform is sensitive to the "*appropriate*" specification of the constants that represent the decay, increment and aggregation. Inappropriate choices of these values can result in all particles becoming coincidental, within n-dimensional space, before any useful information about their relationships can be extracted. Alternatively, it could result in the particles aggregating at such a low rate that they appear unrelated. Hence, it is sometimes necessary for the investigator to 'fine-tune' the specification of these constants in order to gain useful results.

To demonstrate this problem, first of all, consider the neuron assembly in figure 5.11.

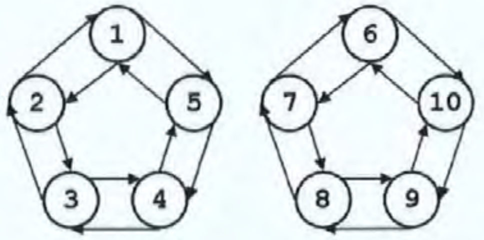


Figure 5.11: Specification of the connections between the 10 neurons used as input to the spike train generator.

This assembly was used to produce a spike train dataset lasting 20s. The gravity transform was then computed for this dataset with parameters a=0.5, τ=0.5 and b=5.

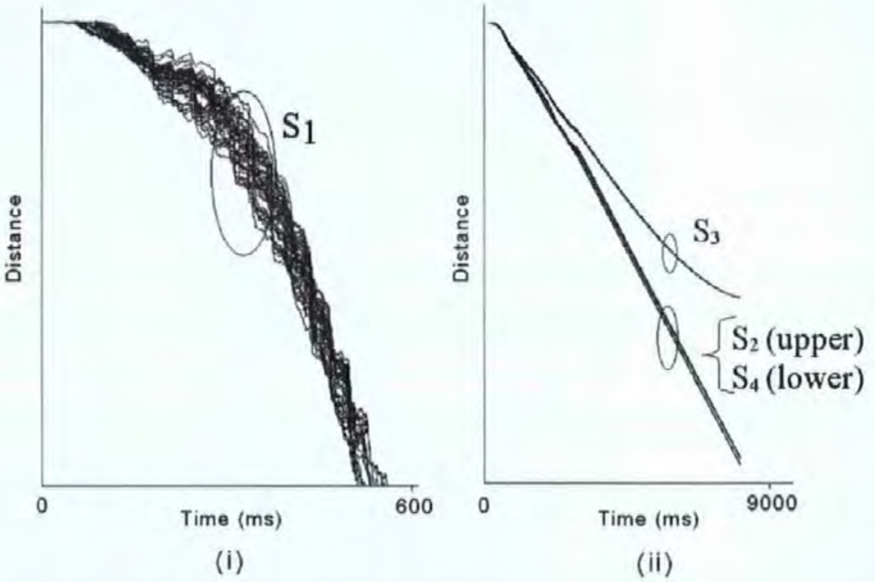


Figure 5.12: Distance graphs of the results of the gravity transform where n=10 and (i) a=0.5, τ=0.5, b=5 and (ii) a=0.5, τ=0.5, b=0.1. Note the different scales.

The resultant distance graph is shown in figure 5.12(i) where $S_1 = \{d_{ij} : i = 1, \ldots, 10, j = 2, \ldots, 10, i < j\}$. Note, it is not possible to distinguish the two groups in the assembly, even though the scale is relatively small.

Using the same input data set, the gravity transform was completed with another set of parameters: a=0.5, τ=0.5 and b=0.1. The resultant distance graph is shown in figure 5.12(ii) where $S_2 = \{d_{ij} : i = 6, \ldots, 10, j = 7, \ldots, 10, i < j\}$, $S_3 = \{d_{ij} : i = 1, \ldots, 5, j = 6, \ldots, 10, i < j\}$ and $S_4 = \{d_{ij} : i = 1, \ldots, 5, j = 2, \ldots, 5, i < j\}$. From this graph, it is possible to conclude that there is a correspondence between the distance pair groupings from the Gravity Transform and the coupling scheme. Hence, the groups of neurons 1 to 5 (S_4) and 6 to 10 (S_2) are notable and S_4 corresponds to the distance between the particles representing each group, which have no inter-connections.

5.2.3 Principle Component Analysis

The result of the Gravity Transform method is the solution of the system of equations 5.3. This solution is an n-by-n dimensional vector-variable, dependent on time. To visualize this highly dimensional solution and make decisions about grouping of the particles (neurons) it is useful to project the solution to a 2-dimensional plane. The choice of the plane is a difficult problem and several possibilities have been investigated. One possibility is to use Principle Component Analysis [BJD81, GW69].

The algorithm used to implement the Principle Component Analysis method is adapted from Numerical Recipes in C [Wil92]. The PCA algorithm aims to find a plane to project the content of the dataset onto, which results in the maximum variance in data values. PCA is used to find the best subspace for the projection of multi-dimensional data. This method achieves a higher degree of representational accuracy by maintaining as much of the overall data structure as possible.

The PCA method used in these trials was achieved by analysing the covariance matrix of a selected time slice of the output from the gravity transform. A co-variance matrix depicts the position of each particle, in each dimension, at a chosen time t. Note that t is chosen to be a point after which useful aggregation has occurred. The eigenvalues and eigenvectors are derived using Householders reduction and an Implicit QL algorithm[Wil92]. Subsequently, the same eigenvectors are used to project each time slice of the output from the Gravity Transform.

The projection is performed at a resolution specified by the investigator. The investigator is required to select how often a projection is performed. For example, the investigator may select to project every 10^{th}, 100^{th} or 1000^{th} data matrix.

The PCA algorithm can be used to generate an overview of a multi-dimensional dataset, such as that produced by the Gravity Transform analysis. An example of this overview is shown in figure 5.13. This plot was produced from the PCA of the Gravity Transform computed for figure 5.12(a). This plot was produced using Matlab.

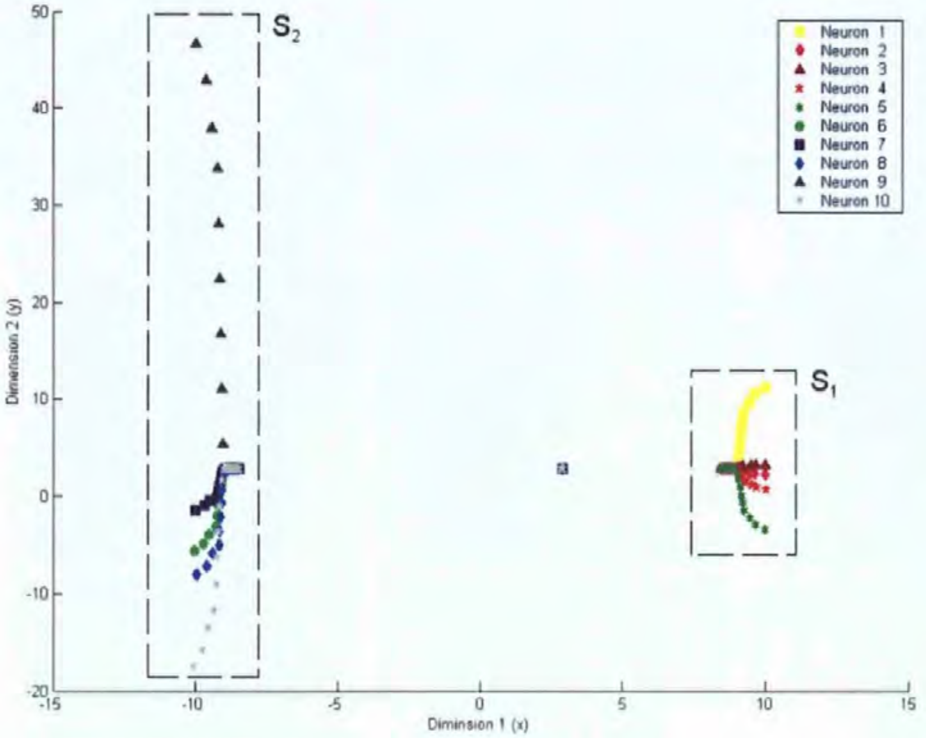


Figure 5.13: Example PCA overview plot

First note, where it was not possible to discern any information from the Euclidean distance plot of this data the PCA overview clearly shows two groups. These groups are noted in figure 5.13, and relate to the following neurons: $S_1 = \{1, 2, 3, 4, 5\}$, $S_2 = \{6, 7, 8, 9, 10\}$. Note the PCA plot directly shows the particles and not the distances between them. Thus, it is possible to discern the neural groups represented in the original dataset.

5.2.4 Independent Component Analysis

In the previous section, PCA was used to find an appropriate sub-plane to project the results of the Gravity Transform analysis onto. Another approach for selecting this sub-plane is Independent Component Analysis (ICA)[HO00]. Similar to PCA, ICA forms a projection of a multi-dimensional dataset on a lower dimension sub-plane. Where PCA attempts to find the sub-plane where the data values have the greatest variance, ICA attempts to find a sub-plane where the data values have the maximum dissimilarity.

Similar to the PCA method, the investigator is required to select the data matrix to perform the ICA upon. The number of dimensions desired and the resolution with which to generate results must also be supplied.

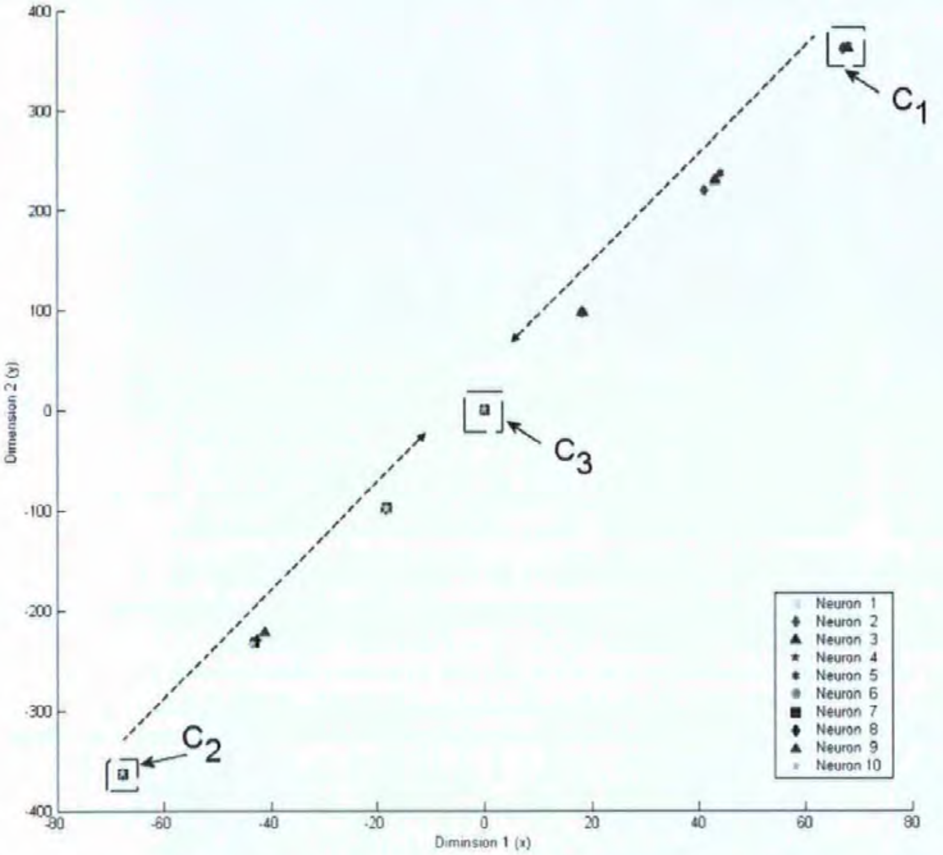


Figure 5.14: Example ICA overview plot

An example overview, generated from the ICA method, is shown in figure 5.14. This overview has been generated for the same dataset as the PCA overview, shown in figure 5.13. From this plot, it is possible to observe two distinct groups, initially indicated as C_1 and C_2 and representing neuron groups (1, 2, 3, 4, 5) and (6, 7, 8, 9, 10) respectively. Further, these two groups can be observed moving towards each other. C_3 in figure 5.14 shows the final aggregation of all neurons.

5.2.5 Cluster Analysis

It is often desirable to identify objects within a dataset that are similar. This identification of clusters of objects can aid in specifying the order in which to display a dataset.

Let us suppose that a measure of similarity between objects can be defined. This measure can be used to calculate a matrix of similarities, or distances, between the objects. A cluster analysis algorithm could then use this matrix to analyse any groups in the dataset.

A standard cluster analysis algorithm has been implemented as part of the Toolbox. In addition, a number of cluster linkage calculations have been implemented, these include: single linkage, complete linkage and average linkage[AQ98].

- Single Linkage, defines the distance between two clusters as the distance between the two closest points within the clusters.
- Complete Linkage, defines the distance between two clusters as the distance between the farthest pair of points within the clusters.
- Average Linkage, defines the distance between two clusters as the average distance between all possible pairs of points within the clusters.

The most effective algorithm for use in spike train analysis was found to be the complete linkage method[SWB04]. Intuitively this algorithm creates tight clusters and all objects inside the cluster have limited dissimilarity.

At each iteration of the cluster analysis algorithm, the two clusters with the smallest distance between them are merged.

Figure 5.15 shows a cluster analysis dendrogram for a dataset of 15 spike trains. This figure shows the order in which the objects (spike trains) in the dataset formed clusters. The distance metric between spike trains is defined as the value of the largest peak in the corresponding Brillinger normalised cross-correlation function. Thus, the distance between two clusters i and j is the value of the largest peak of the cross-correlation function.

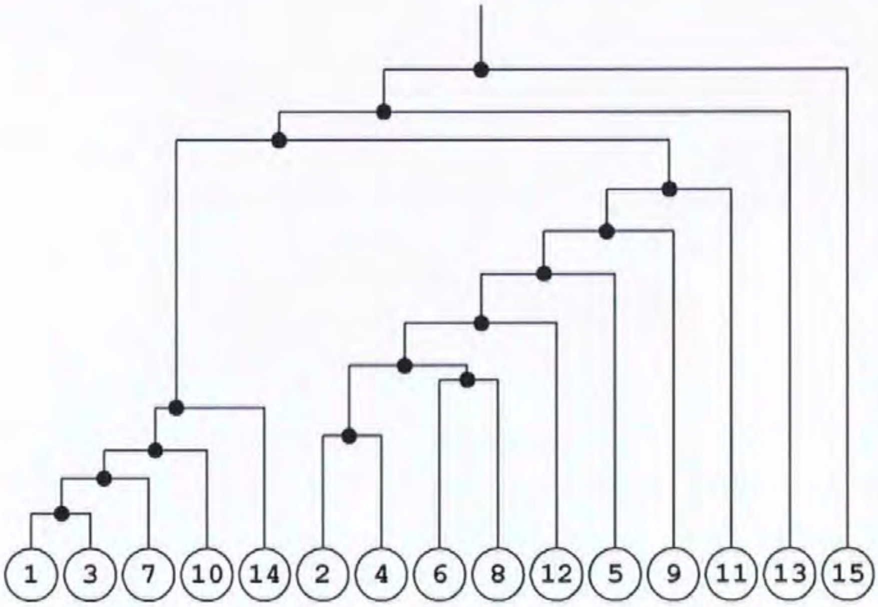


Figure 5.15: Example dendrogram computed to demonstrate the cluster analysis algorithm

The Cluster Analysis algorithm then descends this binary tree (see the dendrogram, figure 5.15) and at each node, it initially follows the leftmost branch. Thus, the algorithm recursively follows the leftmost branch until it reaches a leaf node. Upon finding a left node, the algorithm then traverses the rightmost branch of the current sub-tree before ascending back up the tree to find the next rightmost branch to be followed.

From the dendrogram it is possible to extract the following groups of spike trains: (1, 3, 7, 10 and 14) and (2, 4, 6, 8, 12, 5, 9, and 11) and also that spike trains 13 and 15 are independent. These clusters relate to the connections between the neurons in the assembly that the spike trains were generated from. This assembly is shown in figure 5.16.

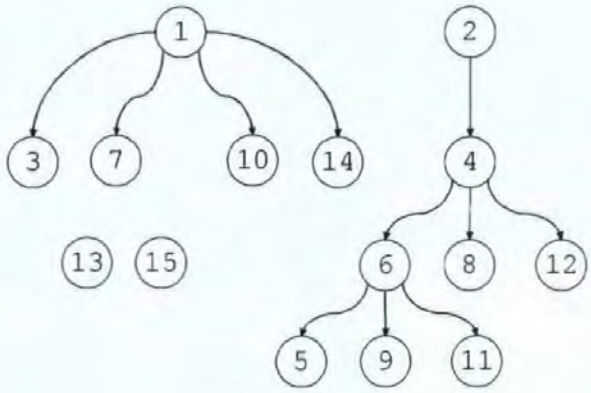


Figure 5.16: Assembly of 15 neurons used to generate the spike train dataset used to demonstrate the cluster analysis algorithm

5.3 Data Presentation Methods

5.3.1 Cross-Correlogram

The output of the cross-correlation function (see §5.2.1) is commonly plotted as a histogram. Thus, a dynamic histogram plot has been implemented, which supports two different formats of histogram. The first version of the histogram plots a counting function histogram. The second version plots the histogram with the Brillinger confidence interval marked. Note that the histogram plot dynamically regenerates and resizes when the display window is resized.

5.3.2 Distance Graph

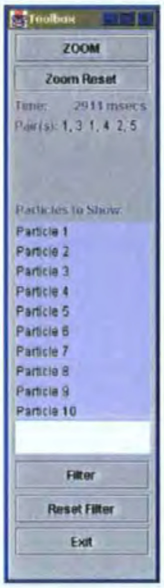
The Distance Graph package plots the change in Euclidean distance between a set of objects, in multi-dimensional space, over time. This package can be used to plot the output of the Gravity Transform algorithm. The data is a file containing a set of position vectors, one vector for each particle and one set of vectors for each recording over time. The i^{th} set of vectors shows the position of each object in multi-dimension space, as recorded at time i. The Euclidean distance between each pair of objects is then calculated and plotted.

The distance graph has the following functionality:

- Overview of the whole time period and all pairs

- Zoom in on a specific area of the plot
- Filtering of which pair-wise distances are plotted
- Details about a specific point on the plot, time and object pair(s)

Control of the different features is achieved via a Toolbox window, shown in figure 5.17. In addition, the details about a selected point in the plot are displayed on the toolbox window.



Time: 2911 msecs
Pair(s): 1, 3 1, 4 2, 5

Particles to Show:
Particle 1
Particle 2
Particle 3
Particle 4
Particle 5
Particle 6
Particle 7
Particle 8
Particle 9
Particle 10

Figure 5.17: Gravity Transform Visualization Toolbox, for user manipulation of the distance graph display

5.3.2.1 Overview

Initially, when a data file is loaded, the plot displays all the pair-wise distances and the entire time frame of the file. This initial display provides the user with an overview of the whole file. The user can identify any general trends and clustering of any areas of interest.

To illustrate the distance graph, an example dataset was generated. This is the same dataset as use in §5.2.2.4. The assembly is shown again in figure 5.18 for reference. The Gravity Transform was computed with the following constants a=0.5, τ=0.5 and b=0.1.

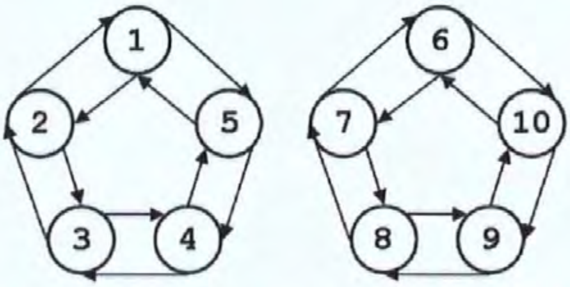


Figure 5.18: The assembly of 10 neurons used to demonstrate the distance graph

A distance graph for this Gravity Transform dataset was then generated and is shown in figure 5.19.

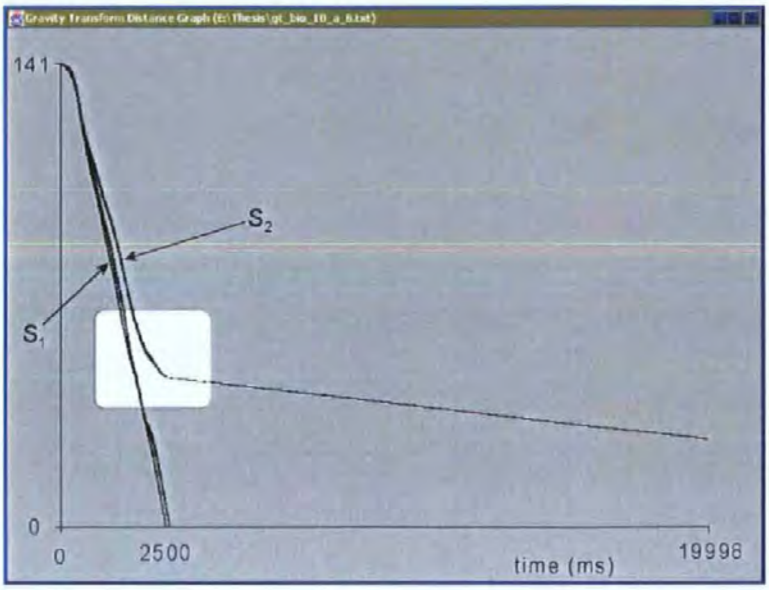

Gravity Transform Distance Graph (E:\Thesis\gt_bio_10_a_6.txt)
141
S2
S1
0
0
2500
time (ms)
19998

Figure 5.19: Gravity Transform Distance Graph Display

From the plot in figure 5.19, it is possible to observe that all particles are initially attracted to each other, as all distance pairs get smaller. However, at approximately 2500ms into the analysis the distance pairs form two groups, which are indicated on the display. The two groups of distances relate to distance between the particles, representing the intra-assembly distances (S_1), and the inter-assembly distances (S_2). This separation indicates that the distances between the particles representing the neurons in each ring continue to diminish. However, the separation between the two groups is maintained.

5.3.2.2 Zooming

The investigator has the ability to zoom in on a selected part of the display. This permits the enlarged section of the plot to be examined in greater detail. Selecting the ZOOM button on the toolbox window activates the function. The user can then select an area of the plot to enlarge by clicking and dragging an area of the plot with the mouse. This is demonstrated in figure 5.20, which magnifies the area highlighted in figure 5.19.

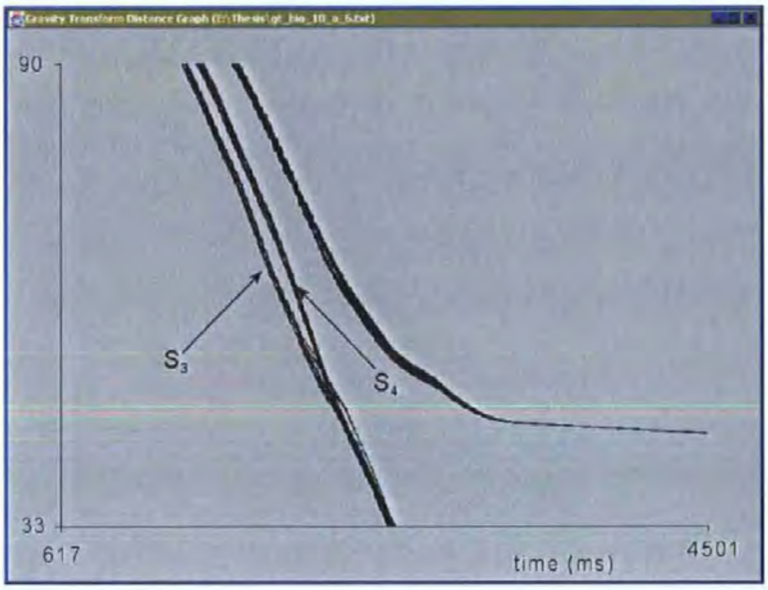


Figure 5.20: Gravity Transform Distance Graph example, showing enlargement of area highlighted in figure 5.19

From this enlarged plot, it is possible to observe that the group of intra-assembly distance pairs identified in the previous section consists of two separate groups of distance pairs. These groups are indicated in figure 5.20 and represent the following distance pairs $S_3 = \{d_{ij} : i = 1, \ldots, 5, j = 6, \ldots, 10, i < j\}$ and $S_4 = \{d_{ij} : i = 1, \ldots, 5, j = 2, \ldots, 5, i < j\}$.

With the zooming function selected any mouse action on the plot area will further enlarge the selected area. If the zooming function is deactivated, the plot remains at the current level of detail; the plot does not revert to the original display. The user can revert to viewing the whole dataset by clicking the Zoom Reset button within the toolbox window.

5.3.2.3 Filtering

The user can select which object distances are displayed. This filtering is achieved via the toolbox window, which lists all objects in the data file. The user can select or de-select objects from the list. When the `Filter` button is clicked the plot is re-drawn with only distances relating to the highlighted objects shown. The `Reset Filter` button cancels the current filter and shows the lines for all pairs. The filter options can be used in conjunction with the zooming function. This supports further refinement of the investigation.

5.3.2.4 Details

In addition to filtering and zooming the display, the investigator can select a line (or lines) on the graph to identify which object pair(s) it represents. Also the time of the selected point is displayed. If multiple lines are practically coincident then the selected point of all pairs are listed. The pair(s) and time are displayed in the toolbox window.

5.3.3 Correlation Grid

The Correlation Gird presents users with an overview of the cross-correlation function results for a number of spike trains.

For a given dataset of n spike trains, all unique cross-correlograms are generated, where the user specifies the bin and window size. Subsequently, the cross-correlograms are normalised using the Brillinger method. Finally, the results of these cross-correlograms are displayed as an n-by-n grid of grey scale cells, representing the individual correlations between all pairs of spike trains.

The grid encodes the 'height' of the largest peak in each cross-correlogram. The peaks are encoded from white, representing no peak, to black, representing the largest peak in the grid.

The user can select whether to view 'all peaks' or just significant peaks. Significant peaks are those that lie outside of the Brillinger confidence interval specified for the grid. In addition, the individual cross-correlograms can be viewed by selecting the corresponding cell in the grid.

The Correlation Grid has been implemented as a package of objects and utilises the cross-correlation (§5.2.1), cluster analysis (§5.2.5) and cross-correlogram (§5.3.1) packages.

The functionality of the Correlation Grid is detailed in the following sections.

5.3.3.1 Overview

To demonstrate the Correlation Grid, a dataset of ten spike trains was generated over 2000ms for the assembly of neurons shown in figure 5.21.

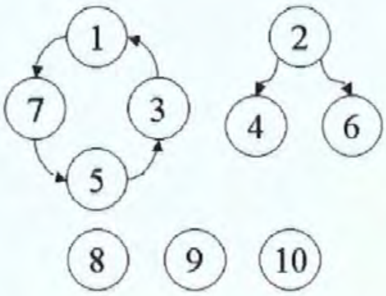


Figure 5.21: Neuron assembly for test data set

In this assembly, the spike trains of neurons one, three, five and seven will be correlated, as their corresponding neurons are connected. Likewise the spike trains of neurons two, four and six will correlate. In contrast, the spike trains of neurons eight, nine and ten will not correlate with any others as they are unconnected, and thus have a 'random' firing pattern.

A Correlation Grid for this dataset was generated, with a bin size of 2ms and window size of 100 bins. The resultant grid, showing all peaks, is shown in figure 5.22.

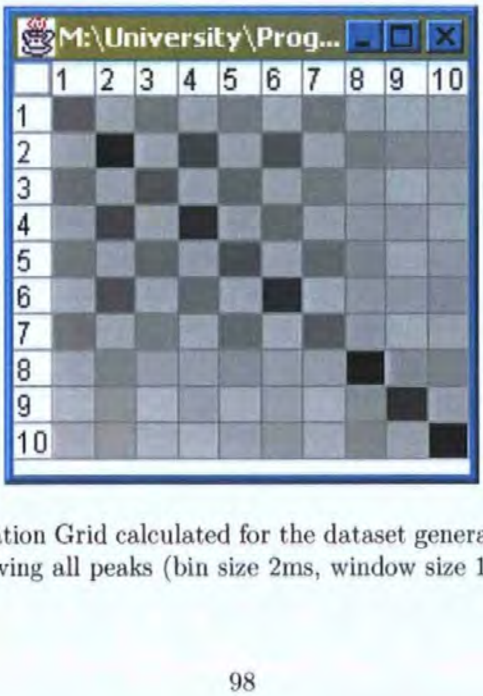
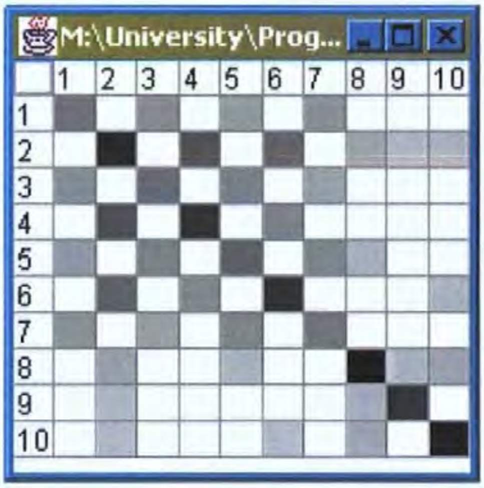


Figure 5.22: Example Correlation Grid calculated for the dataset generated from the neuronal assembly shown in figure 5.21, showing all peaks (bin size 2ms, window size 100)

On closer examination of this Grid, it is possible to hypothesise that spike trains one, three, five and seven are correlated. Note the higher peaks (represented by darker greys) between these columns/rows. Likewise, it is possible to conclude that a relationship exists between the spike trains of neurons two, four and six.

5.3.3.2 Filtering

The clarity of these relationships is greatly improved by filtering the grid so it only contains significant peaks. This filtered grid is shown in figure 5.23. Recall the correlation between spike trains one, three, five and seven which is more prevalent in this filtered Grid. Likewise, the correlations between trains two, four and six are easier to see.



	1	2	3	4	5	6	7	8	9	10
1	▓		▒		▒		▒			
2		■		▓		▓		░	░	░
3	▒		▓		▒		▒			
4		▓		■		▒				
5	▒		▒		▓		▒	░		
6		▓		▒		■				░
7	▒		▒		▒		▓			
8		░			░			■	░	▒
9		░						░	■	
10		░				░		░		■

Figure 5.23: Example Correlation Grid showing only significant peaks (bin size 2ms, window size 100)

With the removal of the non-significant peaks the grid is less cluttered. It is now easier to identify the correlation between the connected neurons, shown in figure 5.21.

Moreover, the lack of correlation between spike trains is more apparent. From an examination of figure 5.23, it is possible to observe that spike trains eight, nine and ten have little or no correlation with any of the other spike trains.

5.3.3.3 Sorting/Clustering

The identification of groups, or clusters, of correlations is key to understanding the relationships between the underlying neurons. It is possible to identify these clusters visually; however this is not easily expanded for problems with larger datasets.

To aid with the identification of correlation clusters, the cluster analysis method is used. This method uses the height of the highest peak of each cross-correlogram to build a dendrogram for the Correlation Gird. This, in turn, is used to generate the initial display order of the spike trains.

The effect of clustering a correlation Grid is shown in figure 5.24. This grid was generated from the Grid shown in figure 5.23, by re-ordering the elements of the Correlation Grid according to the results of the cluster analysis. Cluster analysis was performed on the Grid, to determine the optimal sequence of spike trains in the Grid for comparison of synchronous spiking.

With the aid of filtering and clustering it is possible to clearly identify the groups in the neuronal assembly shown in figure 5.21.

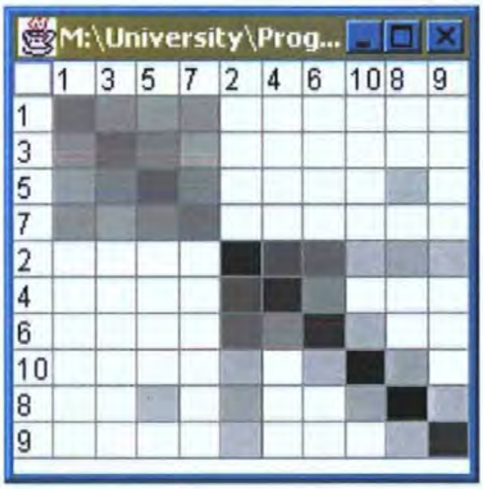
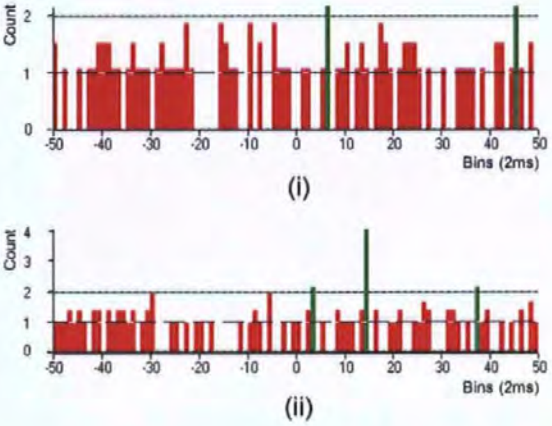


Figure 5.24: Example Correlation Grid, showing only significant peaks and clustered (bin size 2ms, window size 100)

5.3.3.4 'Zooming' and Details

From figure 5.24, it is possible to identify the correlations in the neuron 'ring'; which includes neurons one, three, five and seven. They are shown in the top left portion of the grid. Additionally, the common input group neurons two, four and six are also grouped together in the centre of the diagram. Finally, the independent neurons, eight, nine and ten, are all in the lower right portion of the grid.

From closer examination of the cross-correlograms of the final cluster, of unconnected neurons, it is apparent that the peaks are relatively small. See figure 5.25 (i), which shows the cross-correlation for neurons two and ten.



(i)

(ii)

Figure 5.25: Cross-correlograms for (i) neurons two and ten (ii) neurons two and four from the previous Grid

From the cross-correlogram it is possible to see that the peaks are relatively small. They are only slightly greater than the upper bound of the confidence interval. In contrast, the correlation of connected neurons shown in figure 5.25 (ii) for neurons two and four, shows relatively large peaks (scales vary). Thus, for neurons eight, nine and ten, it would be possible to infer that these peaks do not represent true correlation between the spike trains. Hence, it can be deduced that these neurons are not coupled to any others in the assembly.

5.3.3.5 Summary

This overview of cross-correlograms facilitates improved fine-tuning of the correlation parameters. This fine-tuning is commonly needed to obtain the maximum clarity of the result.

Increasing the bin size, for example, means more spikes are taken into consideration. Thus, a peak must be higher to be considered significant. This principle is illustrated in figure 5.26 where the bin size for the grid has been changed to 3ms. The same dataset and previous window size were used to create this grid.

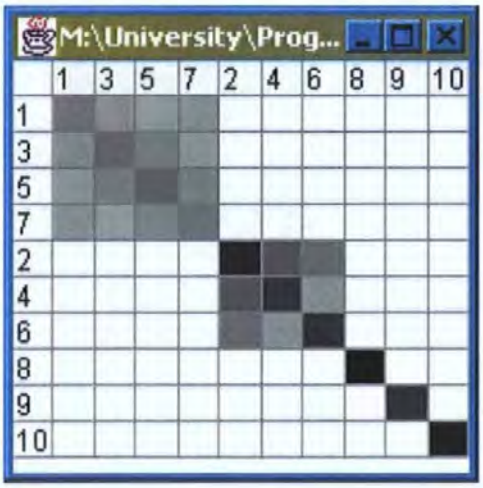


Figure 5.26: Example Correlation Grid showing only significant peaks (bin size 3ms, window size 100)

The grid in figure 5.26 has been filtered so as to only show significant peaks and clusters. Note that correlation only exists between the spike trains of the coupled neurons. For example, spike trains eight, nine and ten show no correlation with any other spike trains.

Thus, from the overview of the dataset provided, by the correlation grid, it is possible to extract the functional groupings of the underlying neurons. Moreover, by examining specific cross-correlograms, it is possible to specify the functional relationships of the neurons that generated the spike trains.

5.3.4 Spike Train Tunnel

The Spike Train Tunnel is a method of analysing the firing patterns of multiple neurons to identify synchronous spiking, groups of coherently spiking neurons covering some period of time and other functional dependencies. The method is based on visualization of spike trains in a specially developed environment called the Tunnel. This environment presents different views of the dataset and an additional overlay that encodes spike coincidence. It enables the user to focus on a specific subset of the dataset using a set of interaction tools. Different frames of reference are provided to enable investigators to track their location within the data space.

The Tunnel is a 'cylindrical' environment that supports user interaction. Figure 5.27 shows the Tunnel visualisation of a randomly generated dataset of spike trains over 200ms.

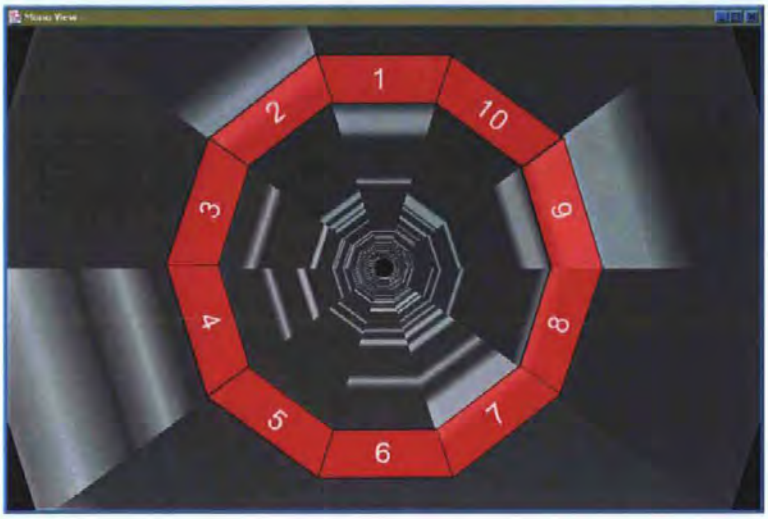


Figure 5.27: A snapshot of the Tunnel representation of the randomly generated dataset over 200ms.

Each of the numbered horizontal bands, that comprise this Tunnel visualisation, encodes the spike train of the corresponding neuron. The two 'end' bands, band 1 and band 10 in figure 5.27, are adjacent to each other, thus forming the 'cylindrical' environment. Note, time is represented through the Tunnel, away from the current viewpoint.

Overall, illumination inside the environment represents the firing of neurons in the currently displayed portion of the dataset. Synchrony is detected by the perception of the position, intensity and frequency of light sources at different parts of the Tunnel.

The investigator is able to 'fly' through the Tunnel to identify sections of the Tunnel (subsets of the data) that are of specific interest. The user has control over both, the speed and direction of the flight. However, to minimise the possible side-affects of dis-orientation during navigation, motion is restricted to being along the Tunnels length. Thus, the user is restricted to forward and reverse motion.

5.3.4.1 Filtering Data

The Tunnel has filtering functionality known as 'dimming'. Whilst in the filtering mode, dimming may be switched on or off for each of the spike trains individually.

Figure 5.28 illustrates filtering of another 200ms dataset. In this dataset, the spike trains of neurons 4, 6 and 8 are identical. The remaining spike trains were all randomly generated. In this figure, all spike trains are dimmed with the exception of spike trains 4, 6 and 8. This filtering is designed to enable investigators to highlight spike trains of interest while maintaining context within the dataset.

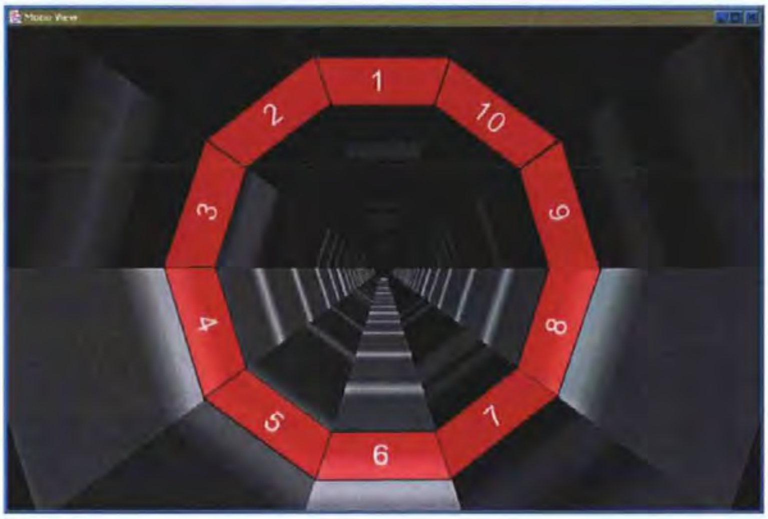


Figure 5.28: A snapshot of dimming (filtering functionality) within the Tunnel environment of a 200ms dataset.

5.3.4.2 Coincidence Sorting

Within the Tunnel environment, the investigator is able to progressively sort the order of the spike trains allocated to the walls of the Tunnel to view spike coincidence. To do this the user selects a spike on a reference train. Subsequently, the spike trains in the Tunnel are reordered, so that trains with spikes coincident to the selected spike are adjacent. Trains that do not have any coincident spikes are inherently moved away from the reference train.

To illustrate coincidence sorting, another 200ms dataset, based on an assembly of ten neurons, was generated. In this assembly, neurons three and ten fire every 12ms and neuron eight fires every 7ms. To illustrate the progressive sorting feature of the Tunnel, neurons four and six fire every 7ms and 12ms.

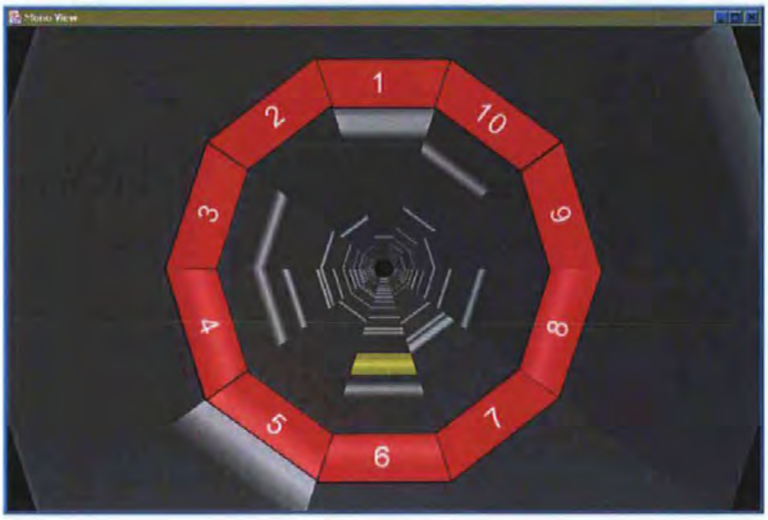


Figure 5.29: A snapshot of the Tunnel visualization for the unsorted dataset

The Tunnel visualization of this unsorted dataset is illustrated in figure 5.29. Note the first coincident spikes (at 12ms) on spike trains three and four, and the following non-coincident spike (at 14ms), solely on train four. Furthermore, in this figure, the selected spike on the reference train is highlighted in yellow. Subsequent to sorting, spike trains four and eight are moved adjacent to the reference train, six, due to coincidence with the selected spike. The result of this reordering is shown in figure 5.30.

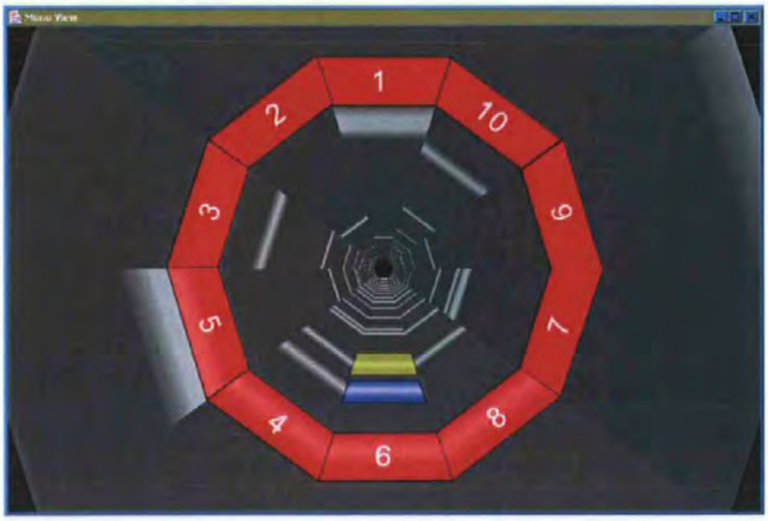


Figure 5.30: A snapshot of the Tunnel visualisation depicting sorting

When the Tunnel's progressive sorting facility is applied, the order of sorted spike trains is initially preserved. However, as this ordering is applied, any train with a spike correlated to the currently selected spike overrides this order. This supports 'fine-tuning' of the spike train order within the Tunnel.

As a result of this type of successive sort, highly correlated spike trains are nearer to each other. This is demonstrated in figure 5.31 where spike train three, four and ten are near to train six due to the existence of a spike of those trains coincident with the currently selected spike (highlighted in blue). Note that, the previous ordering of train four adjacent to train six persists, as it has a spike coincident with both the first and second selected spikes. Since train three solely has a coincident spike with the currently selected spike, it cannot displace train four. In contrast, note that train ten displaces spike train eight.

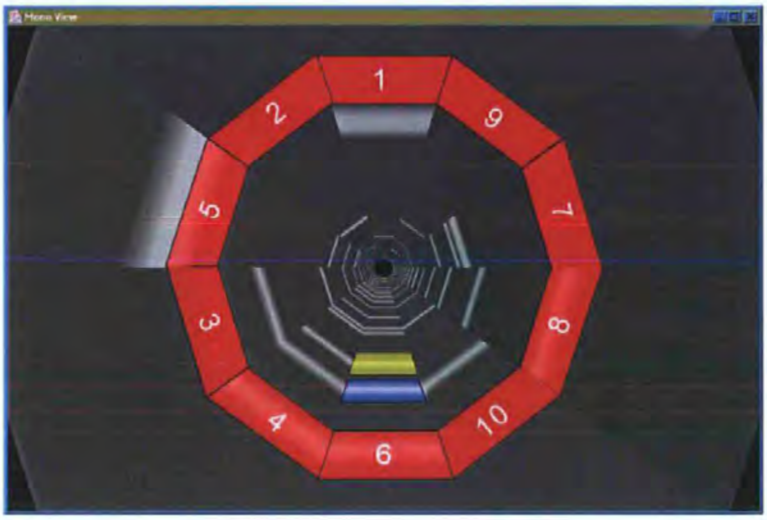


Figure 5.31: A snapshot of the Tunnel visualisation depicting progressive sorting

5.3.4.3 Coincidence Summary

In addition to individual spike coincidences, the overall spike coincidence of the dataset is also of interest. Thus, the Coincidence Summary was developed. This representation derives a summary of neuron firing and colour codes this data.

Each spike train in the dataset is divided into a number of equal time slots, n, referred to as bins. The size of bin is specified by the user, but is usually relative to the neural transmission delay time.

Each spike train has an associated array made up of n elements, where each element of the array is associated with a time segment or bin. Each bin is inspected and if one, or more, spikes occur within the bin, the corresponding element in the associated array is set equal to one, otherwise zero. Note, the total number of spikes in each bin is not important. It is only the presence, or absence, of a spike within that bin that is significant.

When all of the associated arrays have been calculated, an intermediate summary array, of n elements, is created. It is computed from the associated arrays, such that, the ith element of this intermediate summary array is equal to the sum of the ith element of each associated array.

Subsequent to computation, if an element of the intermediate summary array is less than two, there is no spike coincidence. Thus, these elements are set equal to zero in the final summary array. Note that the maximum value of an element in this final summary array is equal to the total number of spike trains in the dataset. This summary array is used to create the Coincidence Summary Visualisation.

The dataset used to create the visualisation in figure 5.33 comprised ten spike trains each lasting 200ms. Each of the spike trains was created by appending four 50ms trains together, such that the first and third segments were low in spike frequency in comparison to the second and fourth segments. Thus, for this 200ms dataset, the final summary array is made up of 67 elements where there are sixty-six 3ms bins and one 2ms bin. Due to the way in which the dataset was generated, elements 1-17 and 35 -51 of the final summary array will be relatively low in value with respect to elements 18-34 and 52-67.

This data is encoded in the Coincidence Summary Visualisation using colour based on the Hue map shown in figure 5.32.

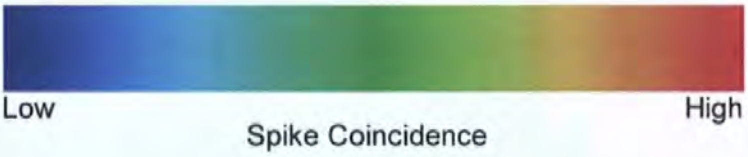


Figure 5.32: Colour coding for Coincidence Summary visualisation

The Coincidence Summary Visualisation (CSV) for the Tunnel is illustrated in 5.33. Recall, that the red labels of the trains bear no relation to coincidence, they have been added for clarity.

Note that the majority of the visualisation viewed at this position in the Tunnel primarily displays the first segment of the final summary array. Due to the relatively low values of the first segment, this is represented by different hues of blue.

It is possible to distinguish the second segment of the final summary array, at the centre of the visualisation. Due to the relatively high values in this segment, it is represented by mainly red and yellow hues.

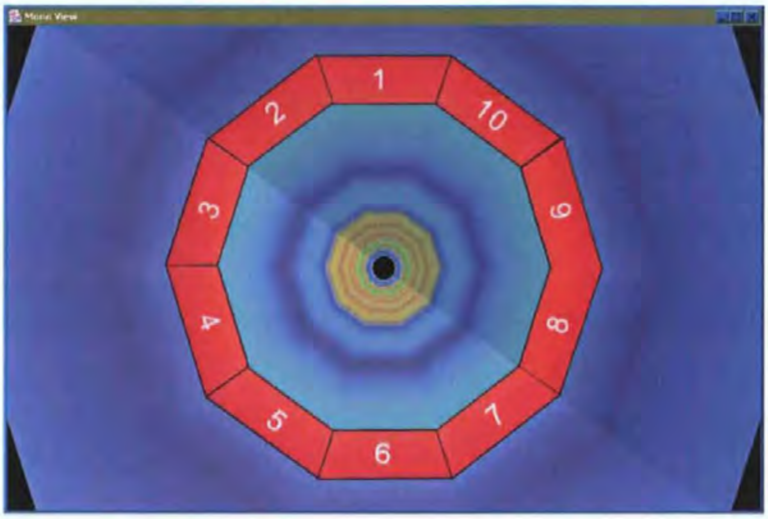


Figure 5.33: A snapshot of the Coincidence Summary visualisation for a 200ms dataset

5.3.4.4 Combining the CSV and Spikes Together

In addition to viewing the CSV, it is also possible to superimpose the Tunnel visualisation onto the CSV. An example of this is shown in figure 5.34 where the data used to generate the CSV in figure 5.33 has been superimposed onto the corresponding Tunnel visualisation of this dataset.

This combination of the CSV and the Tunnel visualisation increases the complexity of the display but helps identify coincidence between spike trains.

Note that in this combined visualisation, the colour used to represent a spike is defined by the corresponding colour in the CSV. This relates to the overall firing activity at that time in the Tunnel.

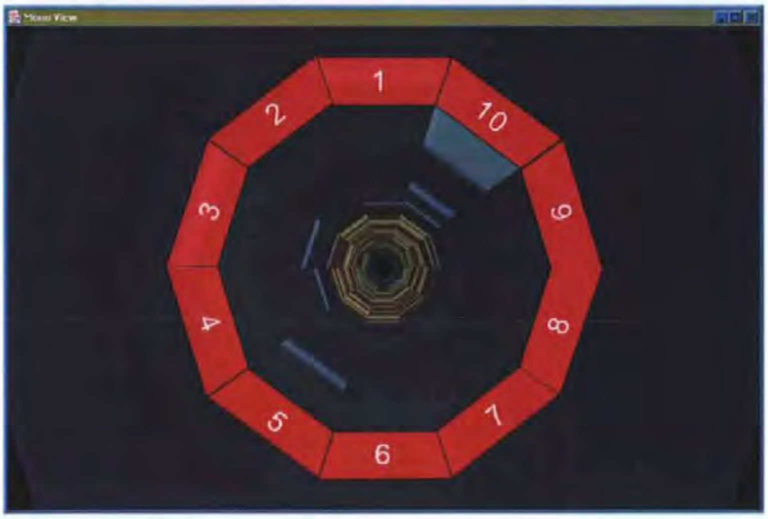


Figure 5.34: A snapshot of the Tunnel visualisation superimposed onto the CSV

5.3.4.5 The 'Flat Map' Representation

Investigators can also overview the data using the 'flat map' representation as shown in figure 5.35. This is similar to the raster plot (see §4.3.1.1) but it has additional functionality. It enables the investigator to select a subset for analysis, thus it can be used to zoom in on "interesting" data. In order to zoom, the investigator highlights an area of the 'flat map'. The red boundary line indicates the section that will be accentuated. The boundary line enables the selection to be fine-tuned. Moreover, it enables the user to track their location within the data set.

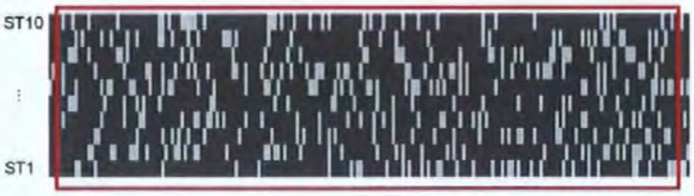


Figure 5.35: An enlarged section of the flat map denoting the ten spike trains, ST1 to ST10.

From figure 5.35, all ten spike trains, ST1 to ST10, are distinguishable but it is difficult to draw any conclusions about their pair-wise correlation to each other.

5.3.4.6 Undo/Redo Facility

In addition to user interaction and navigation, the Tunnel supports an undo/redo facility. Shneiderman asserted that the ability to back track adaptation to the visualization was key to the refinement of understanding [Shn96]. Thus, the user should be able to easily return to previous states of the visualisation.

To this end, the environment tracks all changes to the spike train order enabling the user to selectively undo/redo refinements as required.

Chapter Summary

This chapter presented the analysis, manipulation and presentation methods implemented. Initially, the data formats supported were detailed. Following this the statistical analysis methods were described along with some post-processing manipulation methods. Finally, the various presentation methods were described.

The next chapter describes three case studies undertaken to demonstrate the effectiveness of these methods.

Chapter 6

Empirical Testing

Summary

This chapter presents the empirical testing undertaken to demonstrate the usefulness of the Toolbox. This empirical testing is presented as three case studies. The first dataset was used for training and thus, the assembly was known prior to analysis. The assemblies of the second and third datasets were unknown to the analyst at the time of investigation.

In this chapter, three case studies are presented to demonstrate the effectiveness of combining the modules of the Toolbox together.

Each of the case studies presented in this chapter involved the following three stages (see figure 6.1):

Stage 1 Gravity Transform analysis

Stage 2 Correlation Grid analysis

Stage 3 Spike Train Tunnel analysis

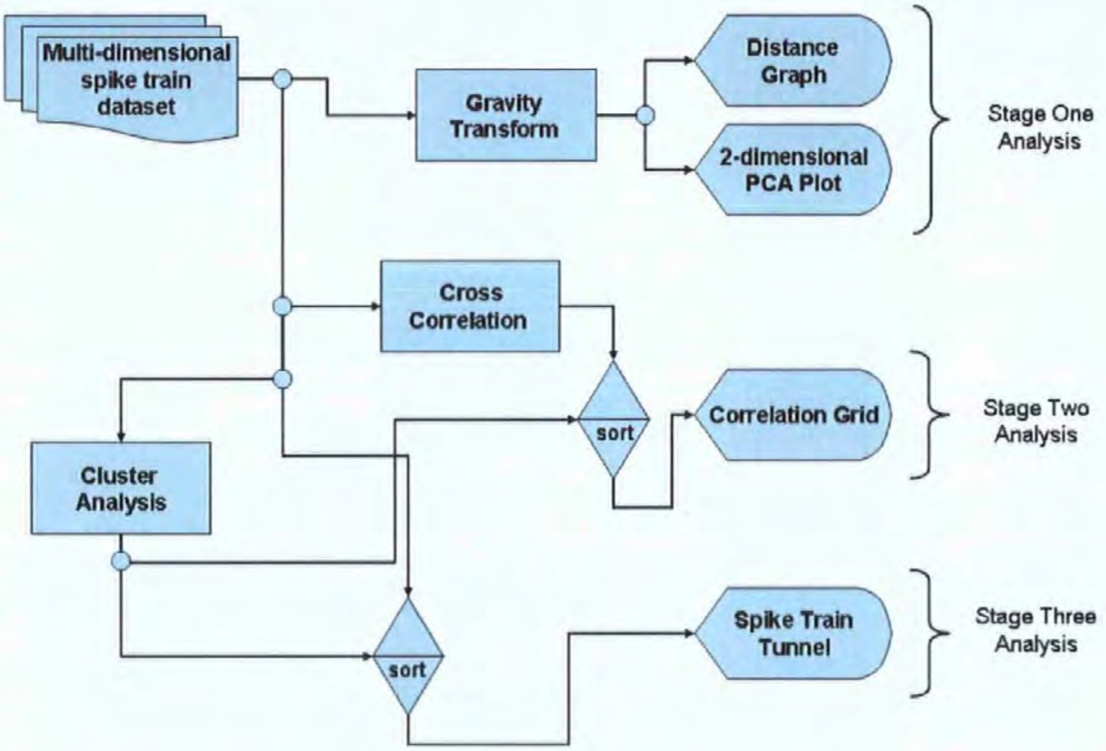


Figure 6.1: Analysis flow of the case study datasets

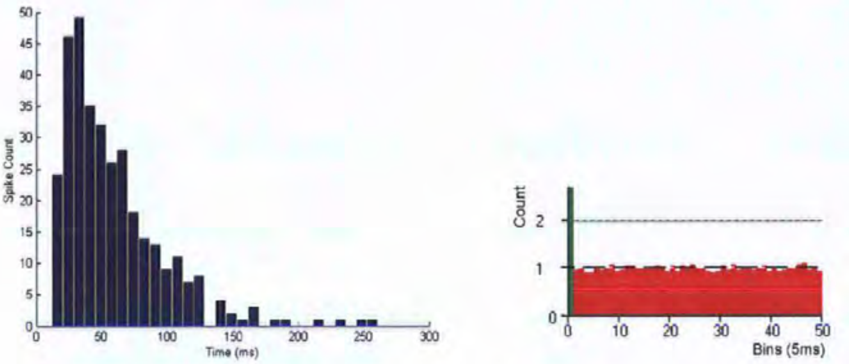
In stage 1, the Gravity Transform analysis was used to identify any major neural groups that existed and any functional separation within the neural assembly. The results of this analysis were used to aid in separating related groups of neurons during further analysis.

In stage 2, the Correlation Grid was initially used to confirm the results of stage 1. Secondly, the Grid was used to aid the specification of the functional relationships between the neurons in those groups.

In stage 3, the Spike Train Tunnel was used to confirm the results of the stage 1 and 2 analysis.

6.1 Trial One

In trial one, an assembly of 15 neurons was simulated for a period of 20000ms. The mean inter-spike interval (ISI) was 75ms, the standard deviation of the ISI of the dataset was 53 ms and its coefficient of variation was 0.7. The ISI Histogram for spike train number 6 is shown in figure 6.2(a) and its autocorrelation is shown in figure 6.2(b).



(a) The ISI histogram of spike train 6

(b) The autocorrelation of spike train 6

Figure 6.2: This figure depicts details regarding spike train 6 from the trial one dataset.

The raster plot from 200ms to 3000ms of this dataset is shown in figure 6.3.

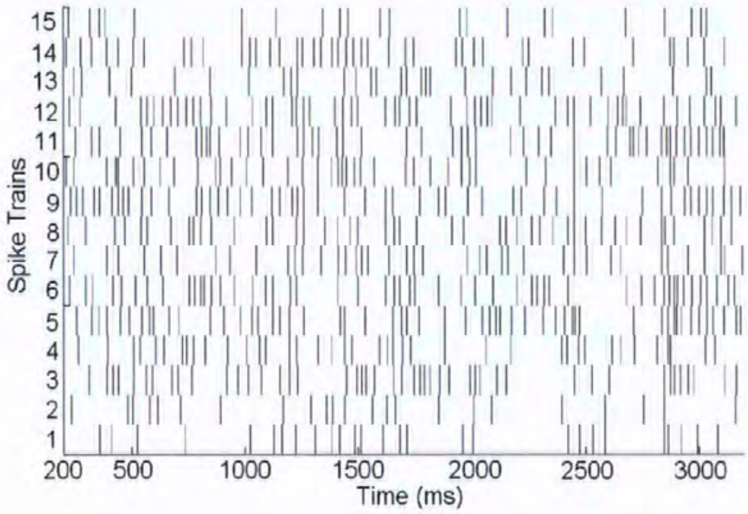


Figure 6.3: Raster plot of trial one spike trains.

6.1.1 Stage 1: Gravity Transform analysis

In this first stage, the trial one dataset was analysed using the Gravity Transform. For this analysis the Gravity Transform was computed with a charge increment, $a = 0.1$, charge decay, $\tau = 0.05$ and overall aggregation, $b = 15$.

Recall, in the Gravity Transform each neuron is represented by a 'particle' and the behaviour of each particle is governed by the spike train of the corresponding neuron. In the following analysis, of the Gravity Transform results, the particles representing the neurons are analysed.

6.1.1.1 Distance Graph

The Euclidian distance graph for this computation is shown in figure 6.4. Note, all distance pairs are displayed.

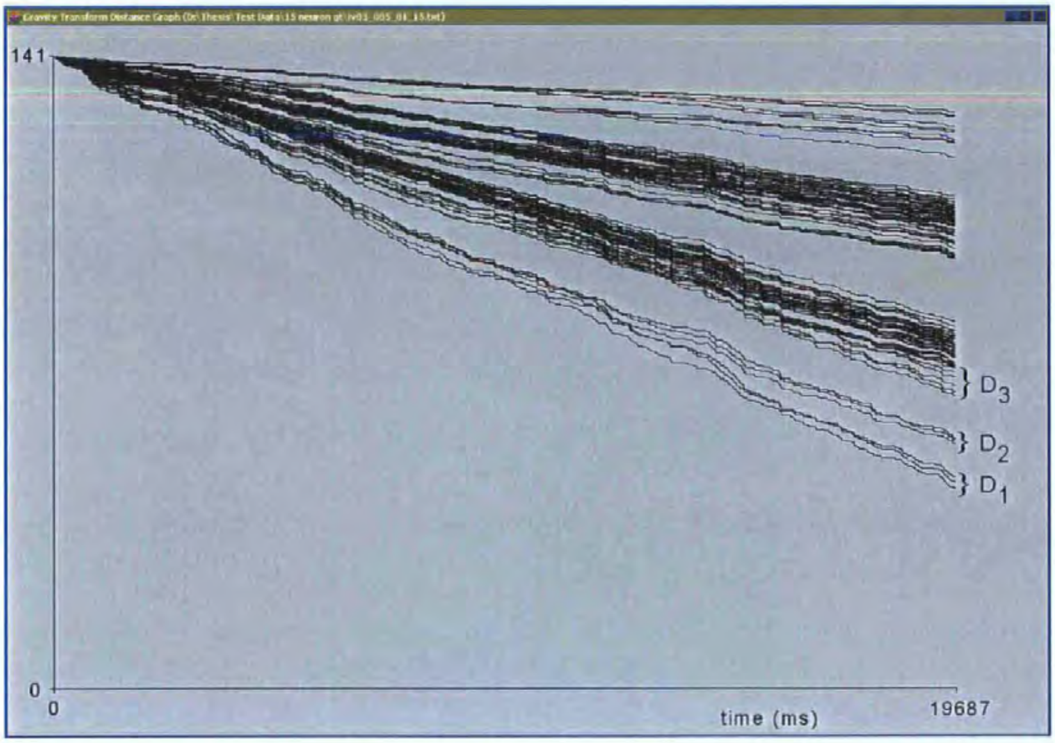


Figure 6.4: Euclidian distance graph of the Gravity Transform for the trial one dataset, with constants $a = 0.1$, $\tau = 0.05$ and $b = 15$, in which all distance pairs (overlapping) are plotted.

From the distance graph plot, it is possible to observe some separation and grouping of the distance pairs. Firstly note the lower two groups, labelled as D_1 and D_2 in figure 6.4. D_1 represents the distances between particles 5, 9 and 11. D_2 represents the distances between particles 6, 8 and 12.

In addition, it is possible to distinguish a sub-group of distance pairs at the bottom of the larger group, labelled as D_3. This group represents all of the distance pairs for particles 3, 7, 10 and 14.

6.1.1.2 PCA

In addition to viewing the results of the Gravity Transform as a distance graph, PCA was applied to this data. PCA was used to form a projection of this multi-dimensional dataset onto a 2-dimensional sub-plane. The result of this analysis was plotted in MatLab and is shown in figure 6.5.

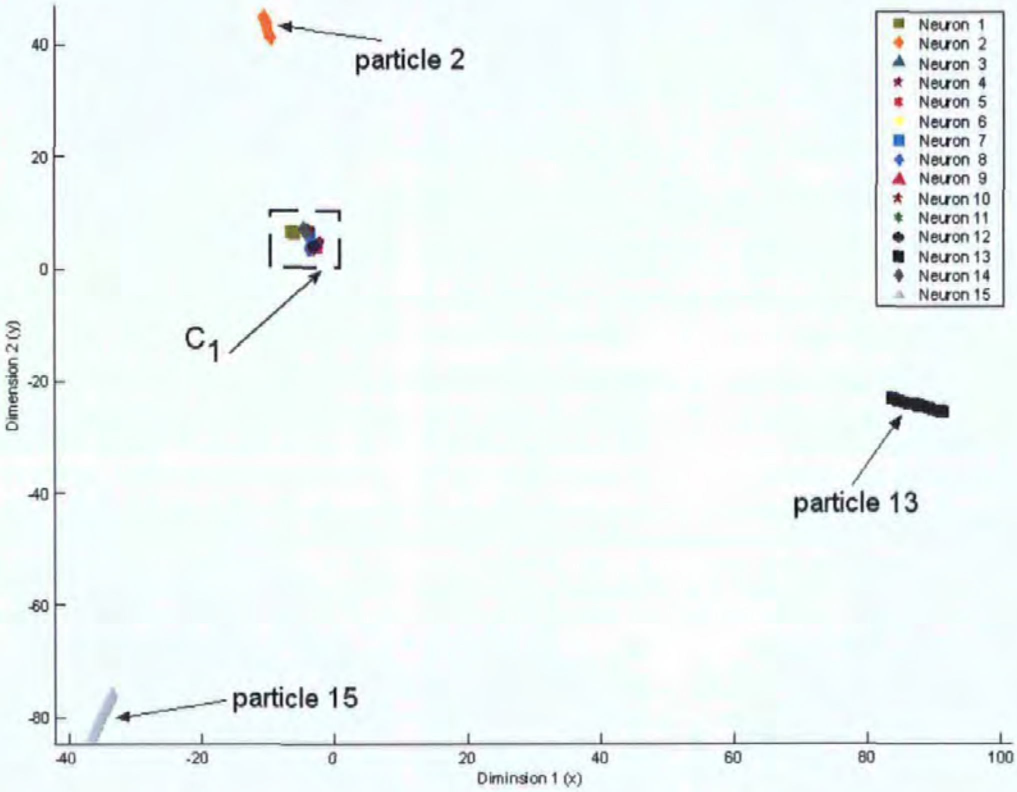


Figure 6.5: This plot shows the results of the gravity transform for the trial one dataset, with constants $a = 0.1$, $\tau = 0.05$ and $b = 15$, projected onto a 2-dimensional sub-plane using PCA.

Firstly, note the cluster of twelve particles, labelled as C_1 in figure 6.5. Although there is extensive overlapping, by a process of simple deduction using the legend, it is clear that C_1 is made up of particles 1, 3 to 12 and 14. Also note that particle 2 is relatively close to this cluster, in contrast to particles 13 and 15, which are relatively distant. From this, it is possible to hypothesise that the neurons represented by particles 13 and 15 have a significantly weaker functional relationship with the other neurons in the assembly. At this stage, it is difficult to make any clear hypothesis about neuron

2, represented by particle 2. This neuron may belong to another group of inter-connected neurons via a weaker connection or neuron 2 may also be significantly separated from all of the inter-connected group(s) of neurons. Further analysis is required to dispel this ambiguity.

In order to analyse the cluster of particles in figure 6.5, C_1 has been enlarged and is shown in figure 6.6.

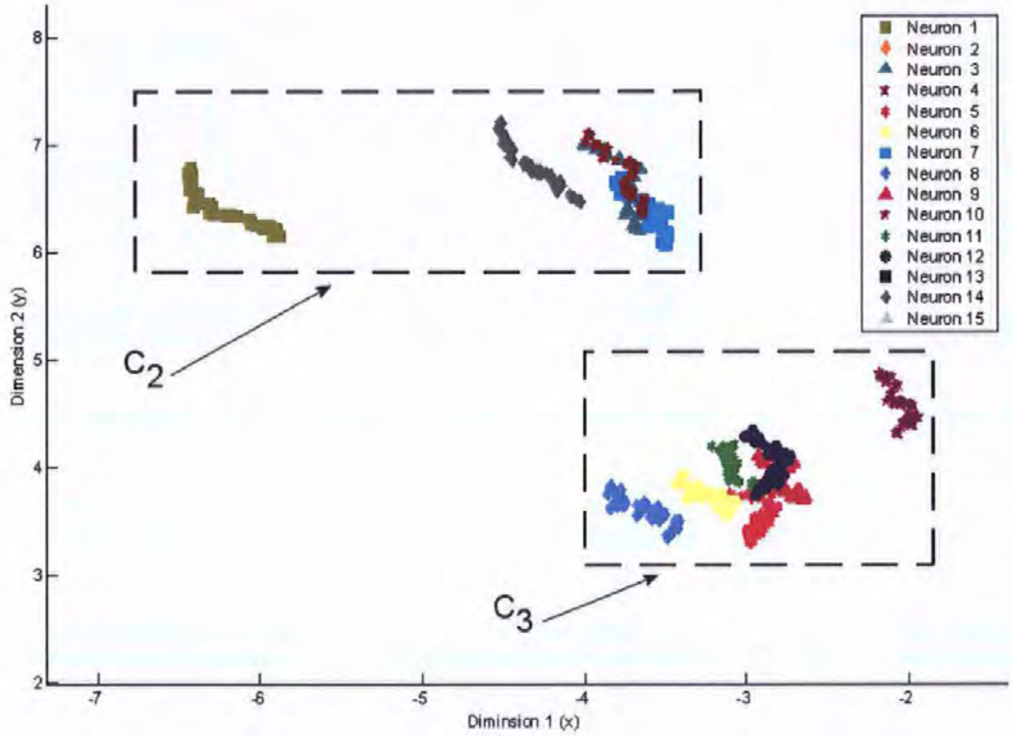


Figure 6.6: Enlargement of the tight cluster shown in figure 6.5, showing the two separate clusters of particles (1, 3, 7, 10, 14) and (5, 6, 8, 9, 11, 12).

As shown in figure 6.6, it is possible to divide the enlarged version of the cluster C_1 into two smaller clusters. Despite overlapping of data, it is possible to observe that C_2 is made up of particles 1, 3, 7, 10 and 14. Note the separation of particle 1 from the rest of this cluster. From this separation it is possible to deduce that there is a stronger relationship between the neurons represented by particles 3, 7, 10 and 14 due to closer proximity.

From figure 6.6, it is possible to deduce that C_3 is made up of particles 5, 6, 8, 9, 11 and 12. Note that there is a stronger relationship between the neurons represented by particles 5, 6, 8, 9, 11 and 12.

6.1.1.3 Summary of Gravity Transform analysis

From the Gravity Transform analysis, it is possible to make the following hypotheses.

1. Neurons 1, 3, 7, 10 and 14 form a functional group. Within that group, there is a stronger relationship between the spike trains of neurons 3, 7, 10 and 14.

2. Neurons 4, 5, 6, 8, 9, 11 and 12 form a separate functional group. Within that group, there is a stronger relationship between the spike trains of neurons 5, 6, 8, 9, 11 and 12.

3. Neurons 13 and 15 seem unrelated to any other neurons in the dataset.

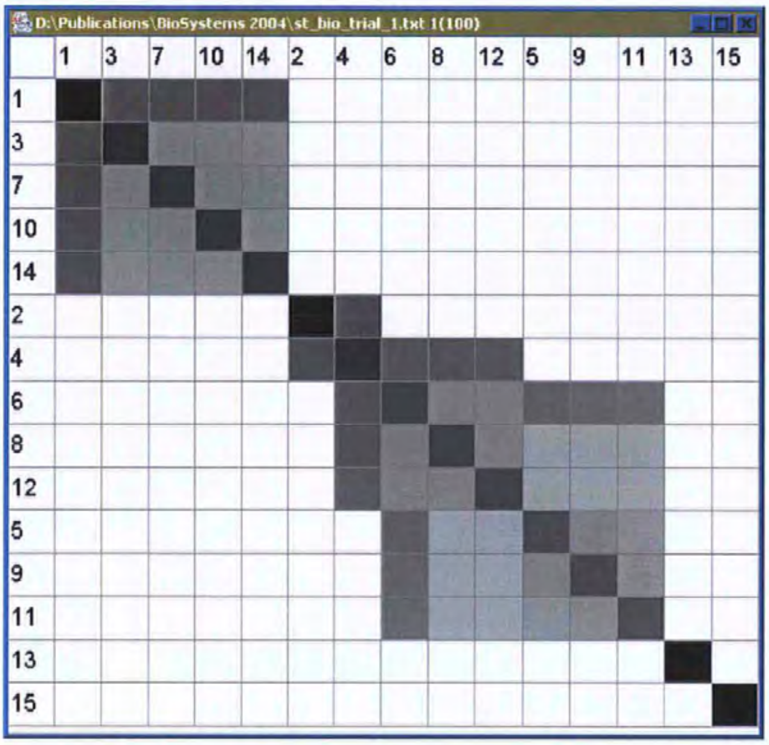
4. The functional relationship of neuron 2 is still ambiguous. It may be completely unrelated to any of the neurons or it may have a weak connection to a group of inter-connected neurons.

Further analysis is required to test these hypotheses and to define the functional relationships of the groups of inter-connected neurons. The second stage of the analysis of trial one uses the Correlation Grid to achieve this.

6.1.2 Stage 2: Correlation Grid analysis

6.1.2.1 Creating the Correlation Grid

The Correlation Grid for the original trial one multi-dimensional spike train dataset was generated with a correlation bin size of 1 ms and a correlation window of 100 bins (100ms). This Grid was subsequently filtered, to show significant peaks only and reordered, based on the results of the Toolbox cluster analysis algorithm. Thus, figure 6.7 shows the filtered, and subsequently clustered, Correlation Grid.



D:\Publications\BioSystems 2004\st_bio_trial_1.txt 1(100)

	1	3	7	10	14	2	4	6	8	12	5	9	11	13	15
1	■	▓	▓	▓	▓										
3	▓	■	▒	▒	▒										
7	▓	▒	■	▒	▒										
10	▓	▒	▒	■	▒										
14	▓	▒	▒	▒	■										
2						■	▓								
4						▓	■	▓	▓	▓					
6							▓	■	▒	▒	▓	▓	▓		
8							▓	▒	■	▒	░	░	░		
12							▓	▒	▒	■	░	░	░		
5								▓	░	░	■	▒	▒		
9								▓	░	░	▒	■	▒		
11								▓	░	░	▒	▒	■		
13														■	
15															■

Figure 6.7: The filtered and reordered Correlation Grid for trial one data.

From figure 6.7, it is clear that significant groups of neurons are visible.

6.1.2.2 Interpretation of the Correlation Grid

Initial inspection of the Grid reveals that three main groups exist. In this analysis, these groups will be referred to as the upper, middle and lower groups. They are denoted by dashed boxes in figure 6.8 and from an initial brief inspection it appears that the Grid reinforces the hypotheses from the stage 1 analysis.

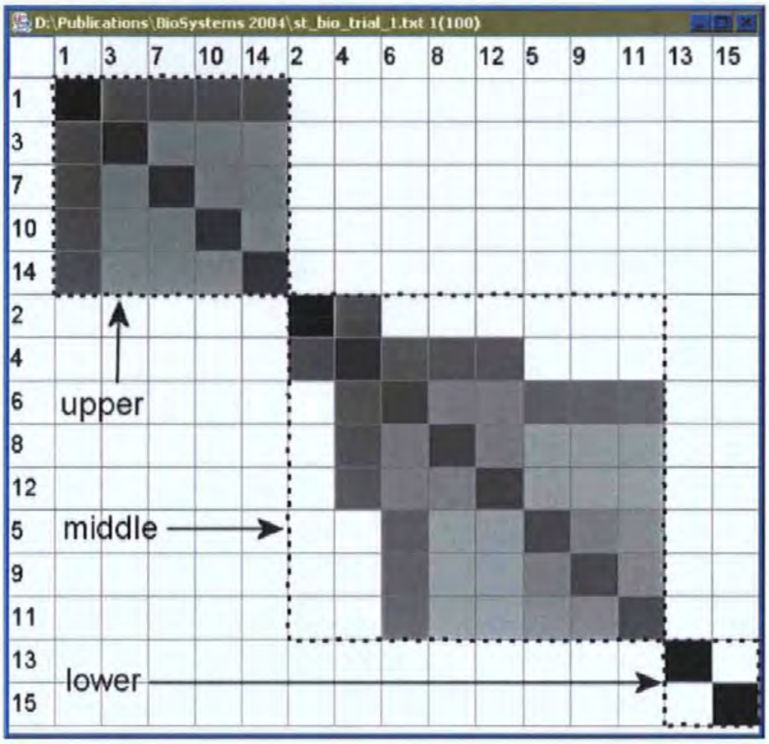


Figure 6.8: The final Correlation Grid depicting the three main groups in the trial one dataset.

The top and middle groups are examined in further detail in §6.1.2.3 and §6.1.2.4 respectively. However, the lower group is very clear. As there is no indication of any significant correlation of either spike train with any of the other spike trains in the dataset, it is feasible to conclude that both neurons 13 and 15 are completely unconnected. These findings support the earlier hypothesis from the Gravity Transform analysis.

6.1.2.3 The upper group of trial one

In order to aid the interpretation of the upper group, the top portion of the Correlation Grid has been enlarged and is shown in figure 6.9.

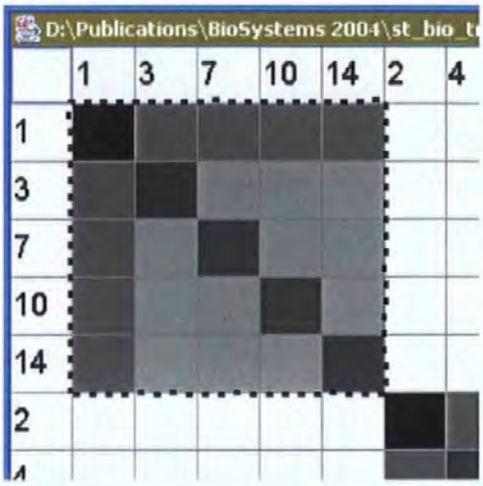


Figure 6.9: Enlargement of the upper group of the Correlation Grid of trial one data.

Note, the upper group is made up of the correlation of the spike trains of each neuron pair in that group (1, 3, 7, 10 and 14). Recall that the density of colour in a cell of the Correlation Grid increases as the strength of the correlation between the spike trains of the two neurons increases. Thus, there is a stronger correlation between spike trains 1 and all of the other spike trains in the upper group. In addition, the spike trains of neurons 3, 7, 10 and 14 all exhibit a similar level of correlation to one another. Thus, it is likely that neuron 1 is connected to all of the other neurons in this group: 3, 7, 10 and 14. Moreover, it is likely that the correlation between neurons 3, 7, 10 and 14 is due to the fact that they have a common input.

This hypothesis is confirmed by inspecting the cross-correlograms of the spike trains pairs (1, 3) and (3, 7), shown in figures 6.10 and 6.11, respectively.

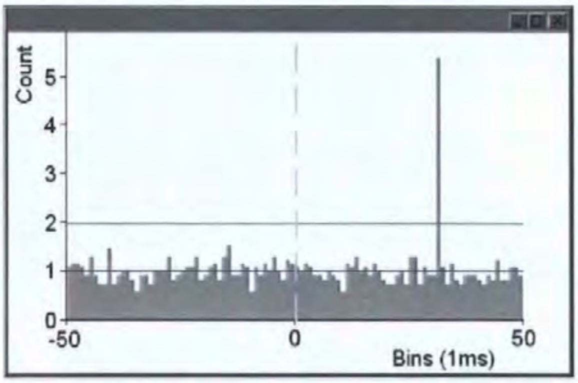


Figure 6.10: The cross-correlogram of spike trains 1 and 3 of trial one.

Figure 6.10 depicts the cross-correlation between the spike trains for neurons 1 and 3. Note that in the histogram a clear peak can be observed. This peak has a positive delay, which denotes that neuron 3 tends to generate a spike after a spike on neuron 1. Thus, there is likely an excitatory connection from neuron 1 to neuron 3.

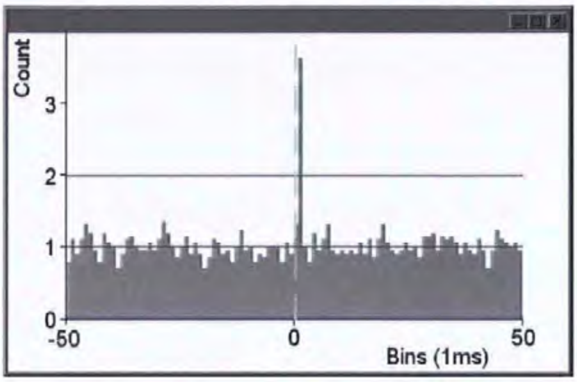


Figure 6.11: The cross-correlogram of spike trains 3 and 7 of trial one.

Figure 6.11 depicts the cross-correlation of the spike trains for neurons 3 and 7. As anticipated, there is synchronous spiking activity between these two neurons. This synchronicity of spiking is denoted by a peak around zero in the output of the cross-correlation function. This behaviour is due to the fact that both are stimulated by neuron 1.

All of these observations support the hypothesis that the upper group sub-assembly is a common source circuit. This proposed circuit is shown in figure 6.12.

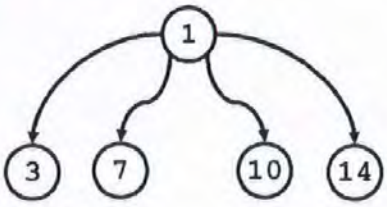


Figure 6.12: The neuronal assembly of the upper group of trial one.

6.1.2.4 The middle group of trial one

As with the upper group, to aid analysis of the middle group, the centre portion of the Correlation Grid is enlarged and shown in figure 6.13.

From this figure, three main sub-groups are apparent. These groups are noted in the figure as: the

2-group consisting of the spike trains of neurons 2 and 4; the 4-group consisting of the spike trains of neurons 4, 6, 8 and 12; and the 6-group consisting of the spike trains of neurons 6, 8, 12, 5, 9 and 11. Note, the 2-group and 4-group overlap, as do the 4-group and 6-group. This overlapping indicates that there are inter-connections between these groups and the neuron represented by the overlapping cell with the highest correlation, in principle, could provide these couplings.

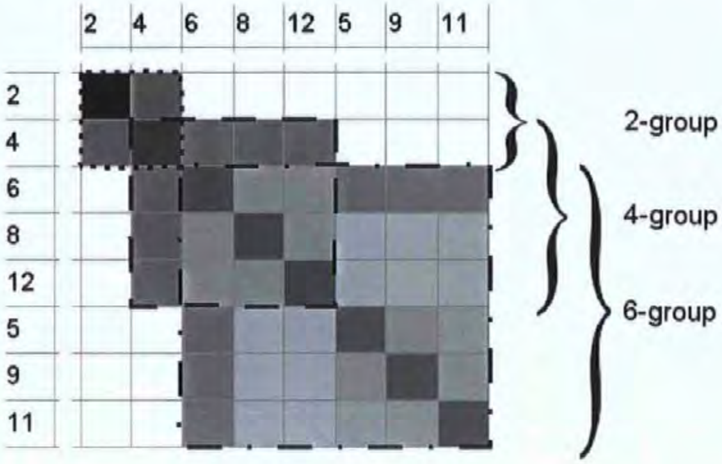


Figure 6.13: Enlargement of the middle group of the Correlation Grid of trial one data.

The 2-group is relatively straightforward. It clearly depicts the relationship of neurons 2 and 4 as their spike trains are highly correlated and only to one another. By inspecting the cross-correlogram for this pair, it is possible to observe a peak with a positive time delay. Thus, neuron 2 connects to neuron 4. The 4-group has a similar structure to the upper group depicted in figure 6.9 and indeed many other trials. Thus, it is possible to directly infer that neuron 4 is a common input to neurons 6, 8 and 12.

The 6-group requires further analysis. Thus, it is enlarged and shown again in figure 6.14. In this figure, the density of the remaining key cells of the Correlation Grid, representing the level of correlation between the spike trains of the underlying neurons, have been evaluated and grouped. Thus, a "very high correlation is represented between the spike trains of neuron 6 and neurons 5, 9 and 11. High correlation exists between these spike trains: 5, 9 and 11. Finally, there is a lower correlation between the spike trains of neurons 8 and 12 with neurons 5, 9 and 11.

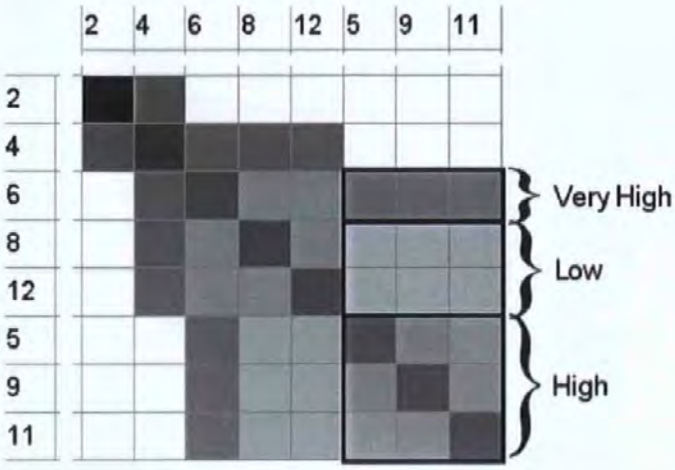


Figure 6.14: Enlargement of the middle group of the Correlation Grid of trial one data with further classification.

If the spike train correlations of neuron 6 solely with neurons 5, 9 and 11 are considered, a familiar hierarchical pattern is observed. Thus, neuron 6 connects to 5, 9 and 11. However, there is still some ambiguity regarding the connections of neurons 5, 9 and 11 with neurons 8 and 12.

Experience of this type of analysis has shown that it is probable that these lower correlations are a result of the shared common root neuron, neuron 4. Thus, it is likely that this neuron provides a common input to neurons 8 and 12 and also to neurons 5, 9 and 11 via neuron 6. To verify this assertion, the cross-correlogram of neurons 8 and 11, shown in figure 6.15, is examined. In this figure it is possible to observe a significant peak, with a positive time delay. Also note that this peak is lower than the direct connection correlations shown in this Grid. This peak indicates that neuron 11 tends to spike after neuron 8. The correlation is due to both neurons 8 and 11 being stimulated by neuron 4. Moreover, the delay is due to the fact that neuron 11 receives its stimulus via neuron 6.

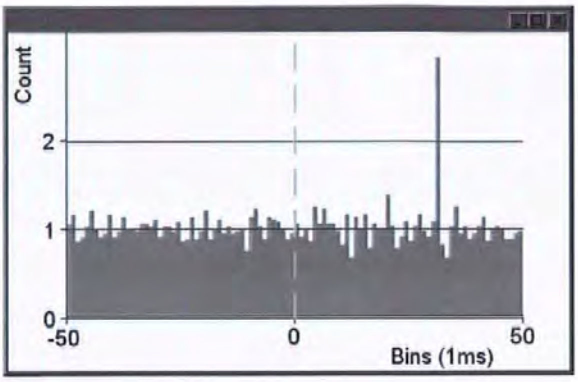


Figure 6.15: The cross-correlogram of spike trains 8 and 11 of trial one.

6.1.2.5 Summary of the Correlation Grid observations

From the second stage of analysis, it is now possible to propose a coupling structure for the functional relationships within this assembly of neurons. This is depicted in figure 6.16

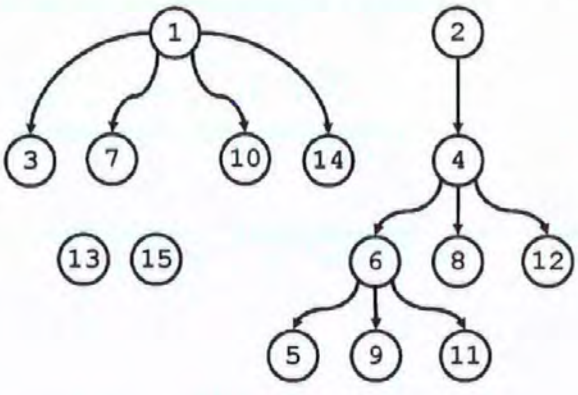


Figure 6.16: The proposed neuronal assembly for trial one.

Note that this coupling structure reinforces the functional relationships extracted during the Gravity Transform analysis. In addition, the solitary nature of neurons 13 and 15 are confirmed. Moreover, the ambiguity regarding the association of neuron 2 has been resolved.

After completion of stage 2 analysis, the Spike Train Tunnel is used to examine the finer details of the temporal relationships between the spike trains in this dataset.

6.1.3 Stage 3: Spike Train Tunnel analysis

To provide confirmation of the neuronal assembly proposed in figure 6.16, the dataset was further analysed using the Spike Train Tunnel. As with the Correlation Grid, reordering the spike trains in the Tunnel display generally results in a more useful display. Thus, the Spike Train Tunnel for the original trial one multi-dimensional spike train dataset is generated and subsequently sorted using the Toolbox cluster algorithm. This is shown in figure 6.17.

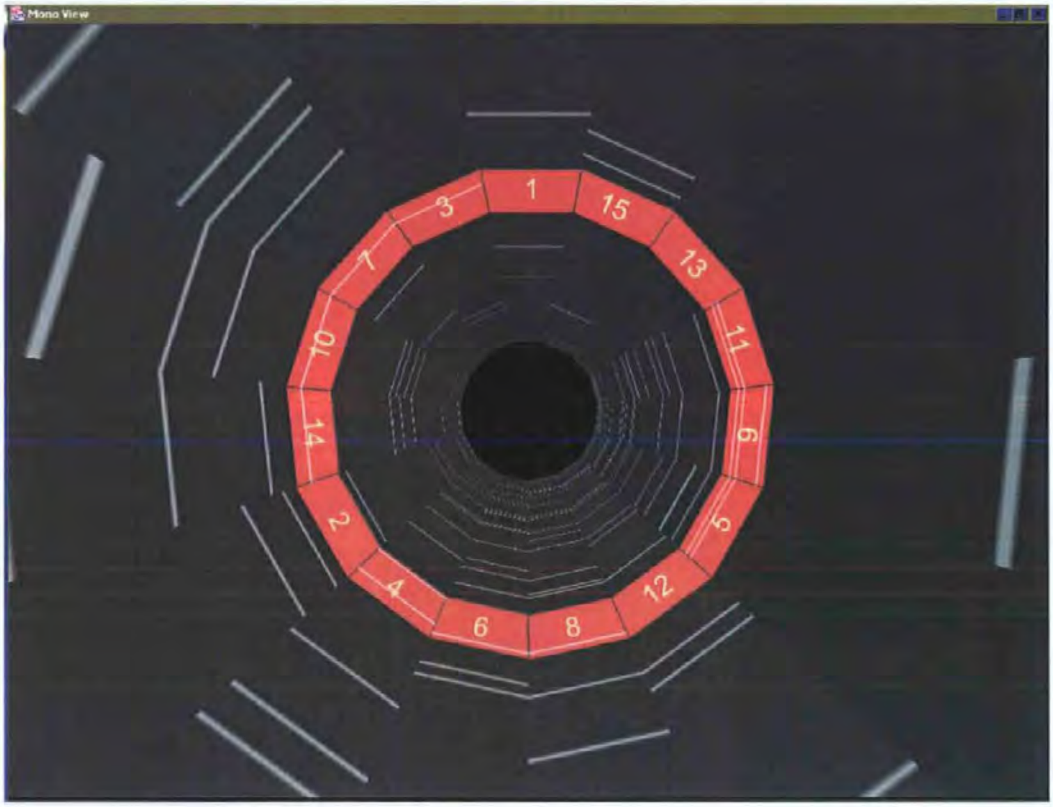


Figure 6.17: A snapshot of the Tunnel representation depicting the 15 spike trains of trial one, in which the order of the trains is defined by the Toolbox clustering algorithm

6.1.3.1 Unconnected neurons

With the clustered order of the spike trains in the Tunnel representation, see figure 6.17, it is clear that spike trains 13 and 15 have no correlation with any of the other trains in this dataset. This is notable from the lack of temporal relationships shown for these spike trains in figure 6.17, as opposed to the other groups.

The analysis of the Tunnel is separated into two parts, the analysis of the smaller and larger inter-connected groups of neurons in the neural assembly shown in figure 6.16.

6.1.3.2 The smaller inter-connected group of trial one

For clarity the architecture of the smaller group from the stage 2 analysis is shown in 6.18.

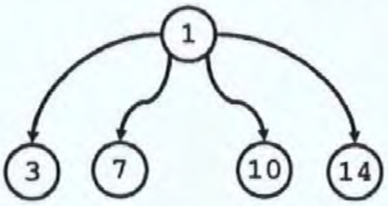


Figure 6.18: The proposed neuronal assembly of the smaller group of trial one.

Figure 6.19 shows the original snapshot of the Tunnel representation in which spike trains 3, 7, 10 and 14 are highlighted. Note the first spike on train 3, which is coincident with spikes on the other three trains. Note that, the occurrence of synchronous spikes on trains 3, 7, 10, 14, was observed in other parts of the tunnel. As these neurons tend to spike coincidently, it is reasonable to suggest that these neurons are all receiving a similar input. Note that it is also possible to see that this synchronous firing is commonly preceded by the occurrence of a spike on train 1 as shown in figure 6.19. This reaffirms the proposed architecture for the smaller group of inter-connected neurons in trial one.

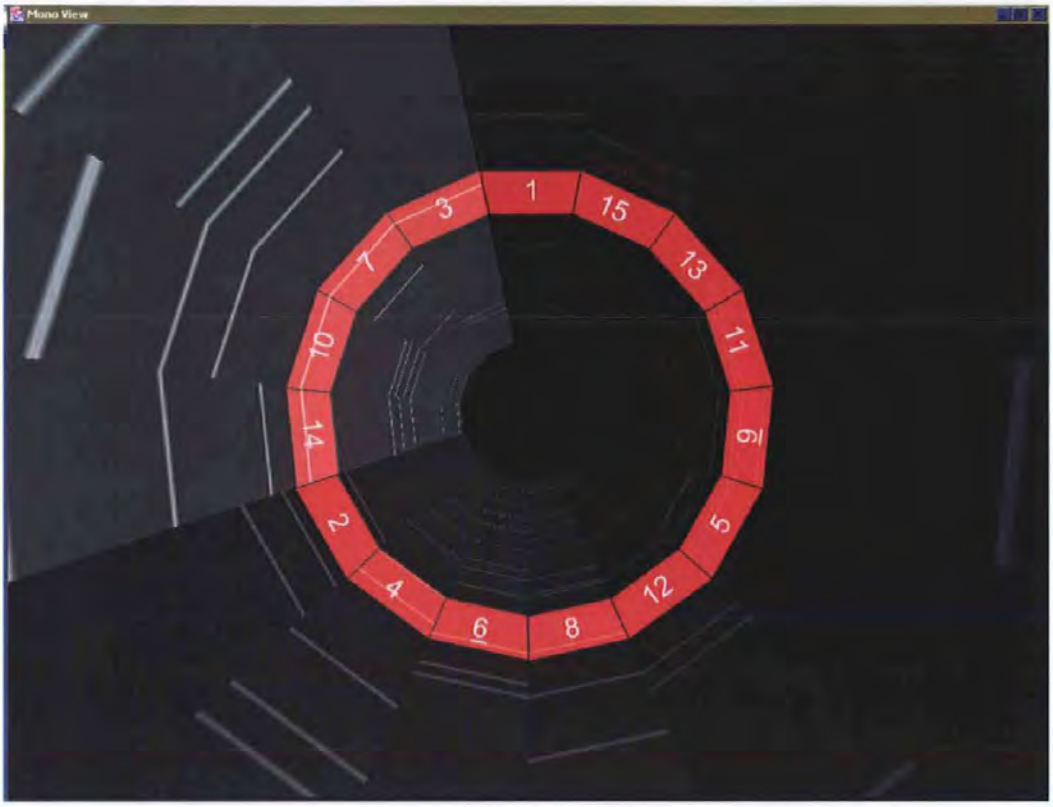


Figure 6.19: A snapshot of the Tunnel representation depicting the 15 spike trains of trial one, in which the order of the trains is defined by the Toolbox clustering algorithm and spike trains 3, 7, 10 and 14 are highlighted

6.1.3.3 The larger group of trial one

For clarity, the proposed architecture of the larger group of inter-connected neurons, identified in the stage 2 analysis is shown in 6.20.

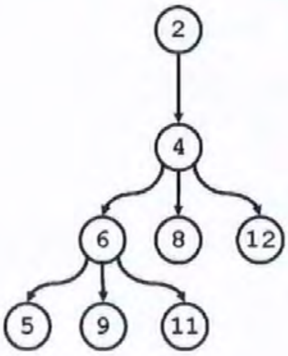


Figure 6.20: The proposed neuronal assembly of the larger group of trial one.

Figure 6.21 shows the original snapshot of the Tunnel representation in which spike trains 6, 8 and 12 are highlighted, it is possible to observe synchronous firing on these spike trains. Firstly, observe the initial spike of train 6, which is synchronous with spikes on trains 8 and 12. Likewise, this synchronicity can be observed again for the 4^{th}, 7^{th}, 9^{th} and 11^{th} spike on train 6. Closer examination also shows that this synchronous firing is commonly preceded by the occurrence of a spike on train 4. Thus, it is likely that neuron 4 is common input to neurons 6, 8 and 12.

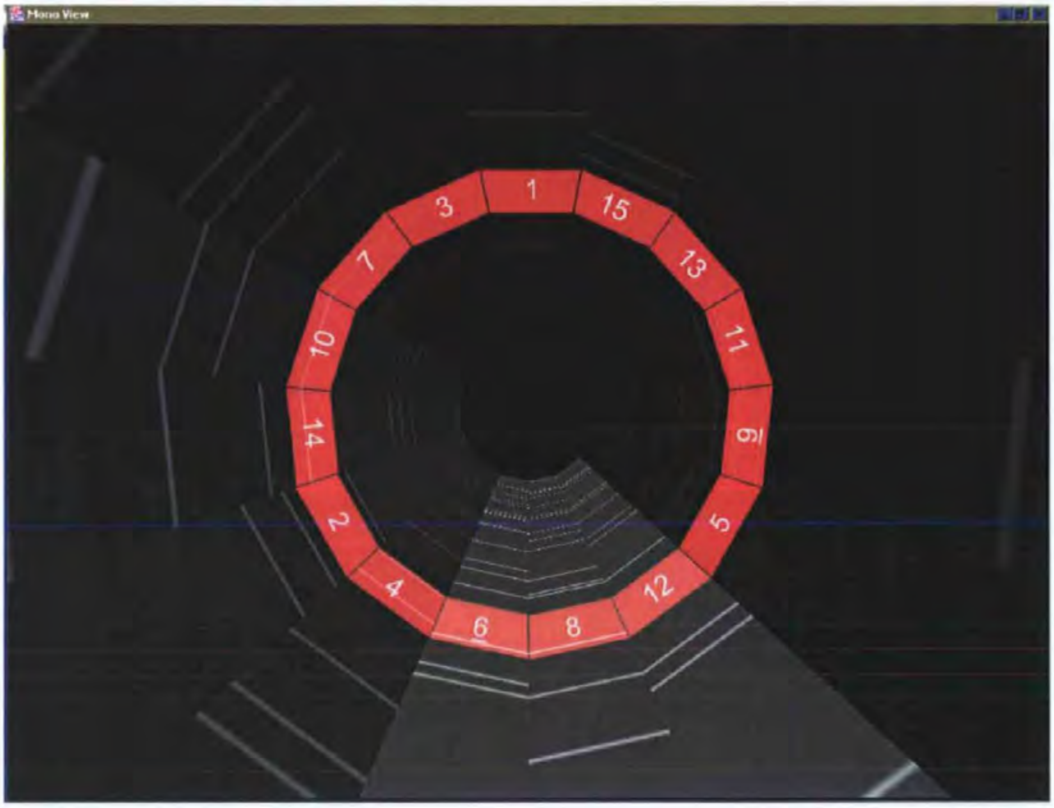


Figure 6.21: A snapshot of the Tunnel representation depicting the 15 spike trains of trial one, in which the order of the trains is defined by the Toolbox clustering algorithm and spike trains 6, 8 and 12 are highlighted

Figure 6.22 shows the original snapshot of the Tunnel in which spike trains 5, 9 and 11 are highlighted and it is very clear that a relationship exists between these trains.

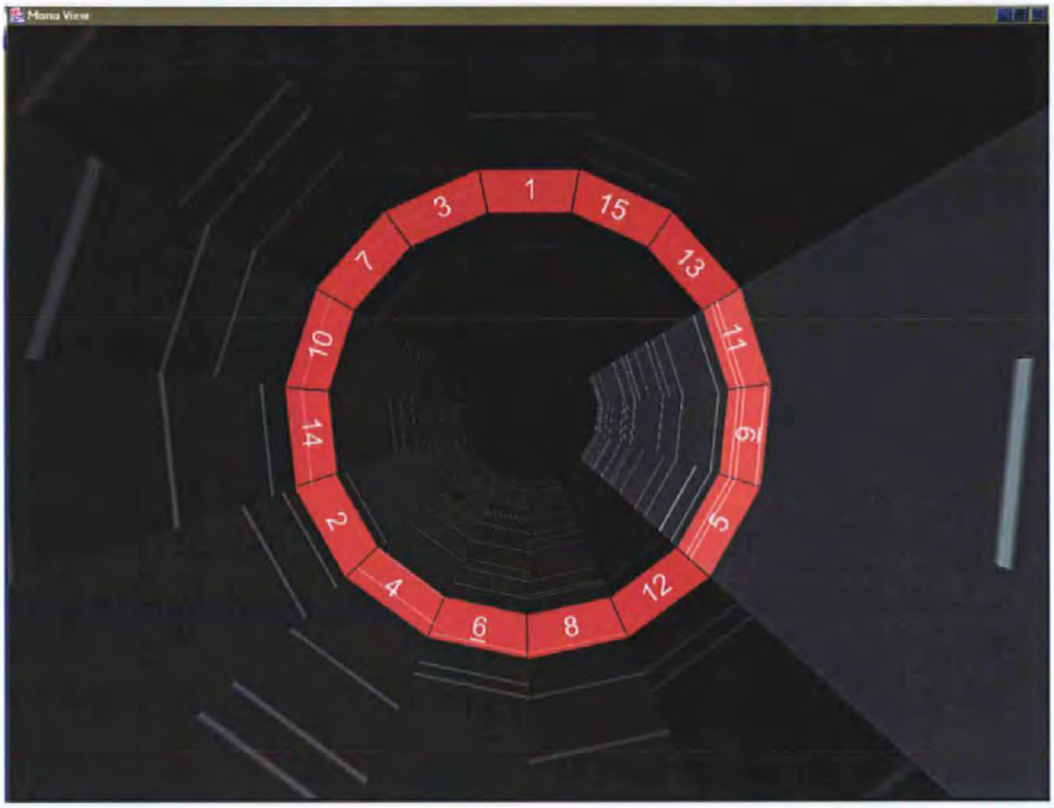


Figure 6.22: A snapshot of the Tunnel representation depicting the 15 spike trains of trial one, in which the order of the trains is defined by the Toolbox clustering algorithm and spike trains 5, 9 and 11 are highlighted

From figure 6.23, it is possible to note that the synchronous spiking in trains 5, 9 and 11 is preceded by a spike in train 6. As it is already known that neuron 4 is a common input to neurons 6, 8 and 12, it is deduced that this is a three level hierarchy of neurons.

Inspection of train 2 using figure 6.23 and additional snapshots of the Tunnel also revealed that a relationship exists between trains 2 and 4. Therefore, it is possible that neuron 2 connects to neuron 4, which subsequently connects to neurons 6, 8 and 12.

The final snapshot of this analysis is shown in figure 6.23 where the original snapshot is shown with spike trains 2, 4, 5, 6, 8, 9, 11 and 12 highlighted.

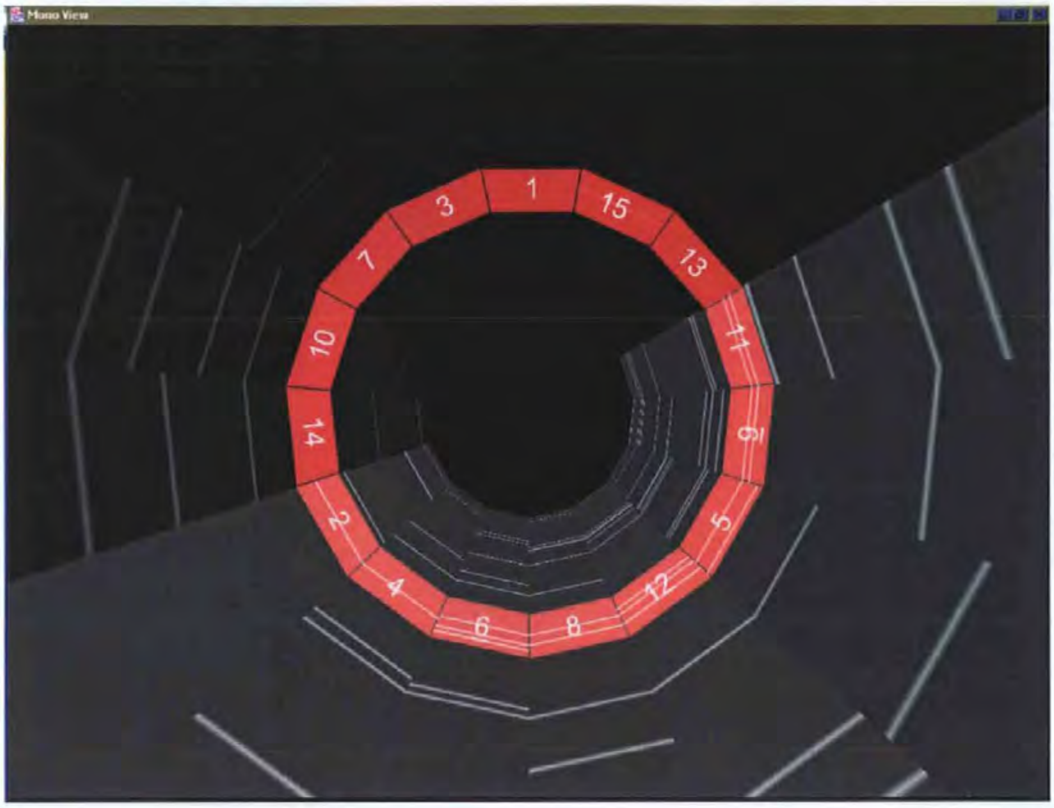


Figure 6.23: A snapshot of the Tunnel representation depicting the 15 spike trains of trial two, in which the order of the trains is defined by the Toolbox clustering algorithm and spike trains 2, 4, 6, 8, 12, 5, 9 and 11 are highlighted

These observations help to reinforce the assembly proposed from the Correlation Grid analysis for this larger group of neurons.

6.1.3.4 Summary of the Spike Train Tunnel analysis

The results of the Spike Train Tunnel analysis permit the proposed coupling shown in figure 6.16, to be confirmed with greater reliability. The Tunnel facilitates the finer details of the relationship of synchronous and related spikes to be extracted. This finer detail permits the 'direction' of the functional relationships between the neurons to be confirmed. Moreover, the absence of a functional relationship between the two groups and neurons 13 and 15 is apparent.

6.1.4 Summary of the analysis of Trial One

The initial Gravity Transform analysis identified two functional groups, (1, 3, 7, 10 and 14) and (4, 5, 6, 8, 9, 11, and 12), two likely unrelated neurons, 13 and 15, and an ambiguous neuron, 2.

The existence of two functional groups was confirmed using the Correlation Grid. In addition, the ambiguity of neuron 2 was resolved; it was weakly related to the larger group. Moreover, the solitary nature of neurons 13 and 15 was reinforced. Thus the two groups, (1, 3, 7, 10 and 14) and (2, 4, 5, 6, 8, 9, 11, and 12) and two solitary neurons, 13 and 15 were confirmed.

In addition, a structure for the connections of the underlying neurons was proposed. This structure was further confirmed by analysing the dataset in the Spike Train Tunnel.

The results of this three stage analysis permitted the original assembly of neurons to be reconstructed. It should be noted that this was one of the earlier trials in this project and as such the original dataset was created and generated by the investigator. Thus the relationship between the neurons was 'known'. However, the assembly was not referenced during the analysis; it was solely used at the end to confirm if the correct structure had been regenerated.

6.2 Trial Two

For this trial, an assembly of 10 neurons was simulated for a period of 300,000ms. Note, the structure and connections of the neurons in this assembly were unknown to the analysts prior to the investigation.

The raster plot for the first 3000ms of this dataset is shown in figure 6.24.

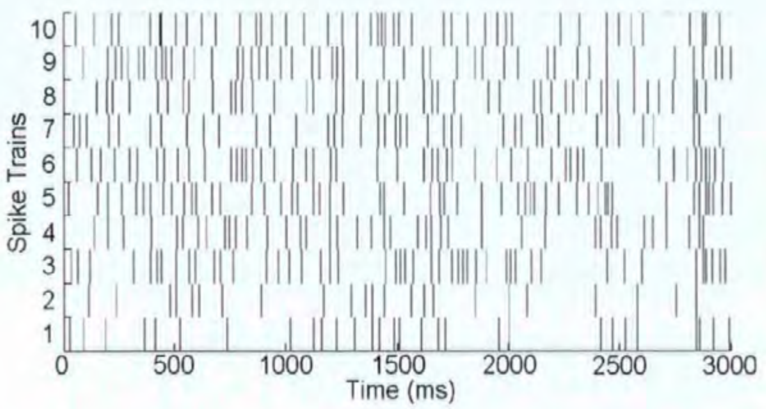


Figure 6.24: Raster plot of trial two spike trains.

6.2.1 Stage 1: Gravity Transform analysis

The Gravity Transform algorithm was used to analyse the undertaken on the dataset of trial two. It was computed with a charge increment, $a = 0.2$, charge decay, $\tau = 0.05$, and system aggregation, $b = 10$.

Recall, in the Gravity Transform each neuron is represented by a 'particle and the behaviour of each particle is governed by the corresponding neurons spike train. In the following analysis, of the Gravity Transform results, the particles representing the neurons are analysed.

6.2.1.1 Distance Graph

The Euclidian distance graph for this computation is shown in figure 6.25, note that all distance pairs are shown.

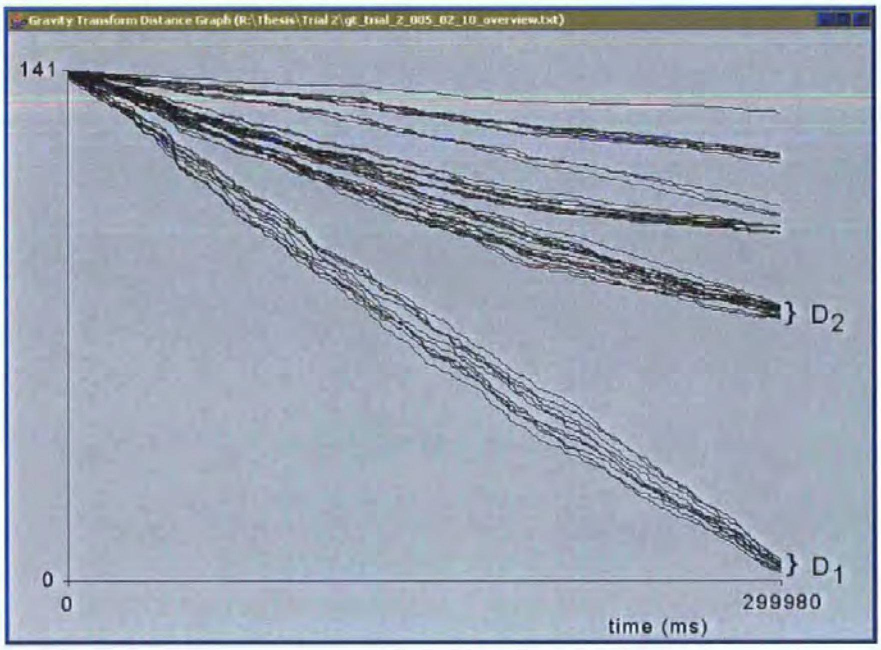


Figure 6.25: Euclidian distance graph of the gravity transform for dataset of case study two, with constants $a = 0.2$, $\tau = 0.05$ and $b = 10$, note all distance pairs are plotted.

From the distance graph, it is possible to observe some separation and grouping of distance pairs. The most striking feature of this distance graph is the highly separated group D_1 as shown in figure 6.25. This group consists of the distances between particles: 1 with 3, 5, 7 and 9; 3 with 5, 7 and 9; 5 with 7 and 9; and 7 with 9. In addition, a second group of separated distance pairs is notable, D_2

in figure 6.25, consisting of the distances between particles: 1 with 4, 8 and 10; 3 with 4, 8 and 10; 4 with 5, 7 and 9; 5 with 8; 7 with 8 and 10; 8 with 9; and 9 with 10.

From this highly distinct separation of distance pairs, it is possible to hypothesis that the spike train dataset was generated from a neuronal assembly with some internal functional separation. Moreover, it is likely that neurons 5, 7, 8 and 9 form a functional group, due to the particles representing these neurons being attached in a similar manner. In addition, it is possible to suggest that neurons 1, 3, 4, and 10 form a second functional group.

6.2.1.2 PCA

In addition to viewing the results of the Gravity Transform as a distance graph, PCA was applied to this data. PCA was used to form a projection of this multi-dimensional dataset onto a 2-dimensional sub-plane. The result of this analysis was plotted in MatLab and is shown in figure 6.26.

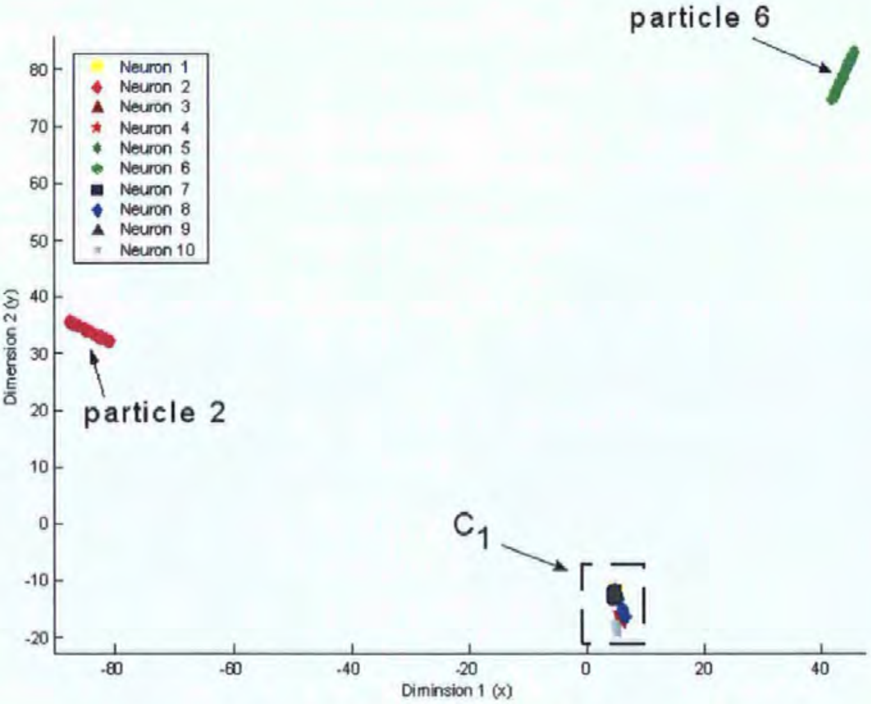


Figure 6.26: This plot shows the result of the gravity transform for trial two dataset, with constants $a = 0.2$, $\tau = 0.05$ and $b = 10$, projected onto a 2-dimensional sub-plane using PCA.

First note the cluster of particles, indicated as C_1 in figure 6.26. Although there is considerable overlapping, by a process of elimination using the legend, it is possible to deduce that this cluster consists of the particles 1, 3, 4, 5, 7, 8, 9 and 10. Also note the separation of particles 2 and 6 from the cluster (C_1). It is possible to hypothesis from this separation that the neurons represented by particles 2 and 6 exhibit a different spiking behaviour to the remainder of the group. Thus, it is possible that these neurons form a separate functional group, as they are approximately equidistant from the main cluster. However, they have a strong tendency to cluster with the group, thus it is also possible they are related to part, or all, of the main cluster.

Closer examination of the cluster C_1 in figure 6.26, enlarged in figure 6.27 for clarity, reveals two separate groups, labelled as C_2 and C_3. It is clear that C_3 consists of particles 4, 8 and 10. Also, despite overlapping of data, it is possible to observe that C_2 is made up of particles 1, 3, 5, 7 and 9.

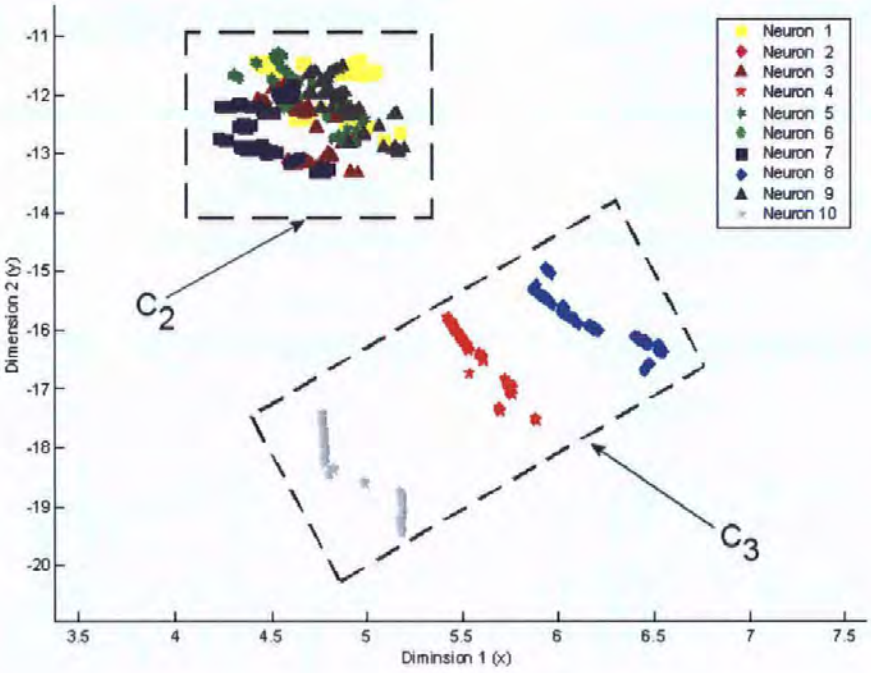


Figure 6.27: Enlargement of tight cluster shown in figure 6.26.

As with the separation of distance pairs in the graph shown in figure 6.25, the separation of particles tends to suggest the presence of more than one functional group of neurons, in the underlying assembly.

However, the constituent members of the groups differ in the PCA representation and the distance graph representation. From the enlarged PCA plot, figure 6.27, particles 1, 3, 5, 7 and 9 and 4, 8 and 10 form groups. From the overall PCA plot, figure 6.26, particles 2 and 6 seem to related, but the manner is unclear.

6.2.1.3 Summary of Gravity Transform analysis

From this first stage of analysis, it is clear that the underlying neuronal structure consists of at least two functional groups. In addition, the behaviour of particles 2 and 6 is significantly different from the remainder of the group. However, the exact structure of these groups is unclear, as there is an ambiguity between the results gleaned from the distance graph and the PCA plot.

Combining the results of this analysis the following hypotheses can be offered.

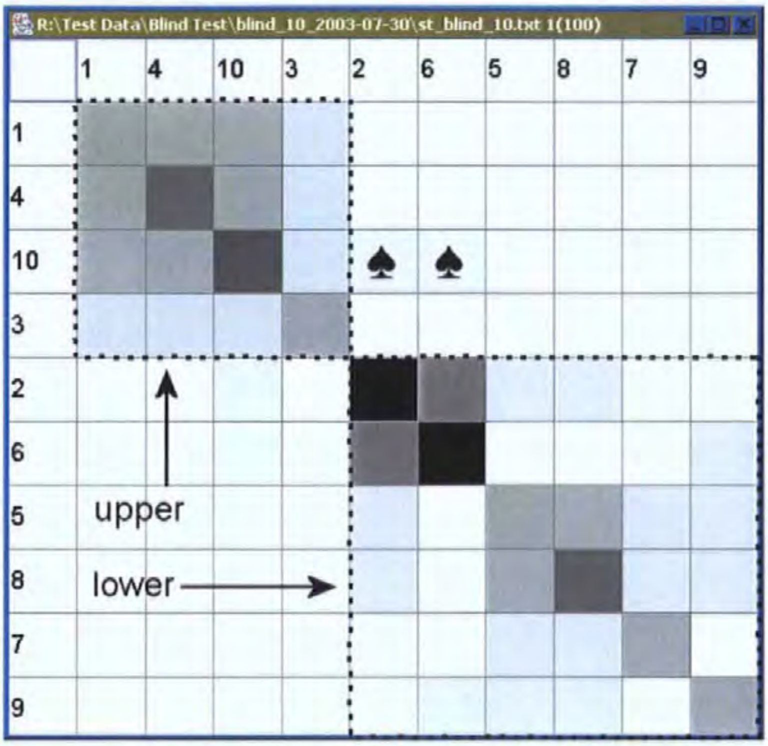
1. Neurons 5, 7 and 9 show a strong functional relationship.
2. Neurons 1 and 3 show a strong functional relationship.
3. Neurons 4 and 10 show a strong functional relationship.
4. It is possible that neurons (1 and 3) and (4 and 10) are related.
5. The relationship of neurons 2, 6 and 8 is unclear.

Further analysis is required to test these hypotheses and to define the connections of the groups of inter-connected neurons. The second stage of the analysis of trial two uses the Correlation Grid to achieve this.

6.2.2 Stage 2: Correlation Grid analysis

6.2.2.1 The Correlation Grid for trial two

The Correlation Grid for the original trial two multi-dimensional spike train dataset was generated with a correlation bin size of 1ms and a correlation window of 100 bins (100ms). The Grid was subsequently filtered and clustered; the resultant Correlation Grid is shown in figure 6.28.



R:\Test Data\Blind Test\blind_10_2003-07-30\st_blind_10.txt 1(100)

Figure 6.28: The final Correlation Grid depicting the two main groups of the trial two dataset.

Initial inspection of the Grid reveals that two main groups exist. These are referred to as the upper and lower groups, as indicated in figure 6.28. The upper group consists of the spike trains of neurons 1, 3, 4 and 10, which were identified as having a function relationship during the Gravity Transform analysis. The lower group consists of the spike trains of neurons 2, 5, 6, 7, 8 and 9. A functional relationship for part of the lower group (neurons 5, 7 and 9) was discovered during stage 1 of this analysis. These groups are examined in detail in §6.2.2.2 and §6.2.2.3.

In addition, there appears to be a link between the upper and lower groups. Note the correlations between spike trains 2 and 10 and spike trains 6 and 10, indicated by the spade symbols (♠) in figure 6.28.

Closer inspection of the cross-correlograms, for each of these pairs, shows a peak that is barely higher than the upper bound of the confidence interval. See figure 6.29 for the cross-correlogram of spike trains 6 and 10. Recall, values which fall inside the confidence interval are not statically signification, and thus they indicate that no synchrony exists between the spike trains represented. In addition, peaks that fall barely outside the confidence interval can be considered a random occurrence and thus not indicating synchrony of spiking. For this reason, these correlations can be discounted. Thus, it is possible to conclude that two independent groups exist.

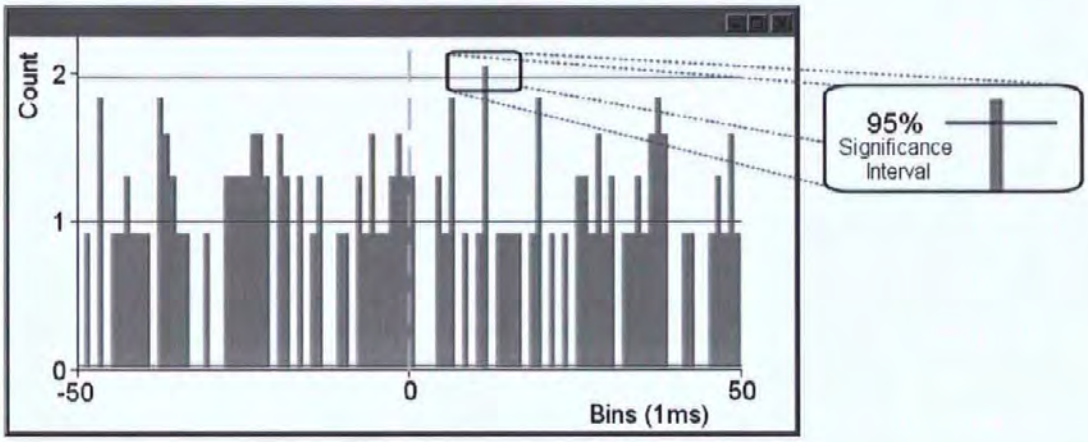


Figure 6.29: The cross-correlogram of spike trains 6 and 10 of trial two. Note that there are no truly signification peaks in this cross-correlogram.

6.2.2.2 The upper group of trial two

In order to interpret the upper group of trial two, the top portion of the Correlation Grid has been enlarged and is shown in figure 6.30.

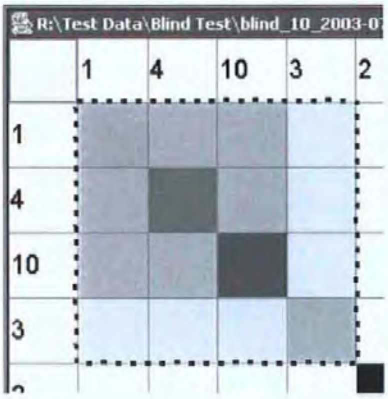

R:\Test Data\Blind Test\blind_10_2003-0

Figure 6.30: Enlargement of the upper group of the Correlation Grid of trial two data.

Note, the upper group consists of the spike trains of neurons 1, 4, 10 and 3, as ordered in the grid. Further, there are stronger correlations between the spike trains of neurons 1, 4 and 10. In addition, spike train 3 correlates, although more weakly, with spike trains 1, 4 and 10.

Closer inspection of the cross-correlograms for this group reveals that spikes tend to occur on trains 3, 4 and 10 after a spike on train 1. In addition, neurons 3, 4, and 10 tend to spike synchronously.

Thus it is possible to deduce that neuron 1 is a common input source to neurons 3, 4 and 10, as shown in figure 6.31.

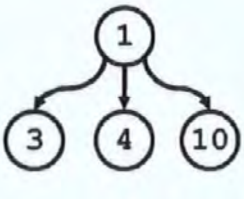
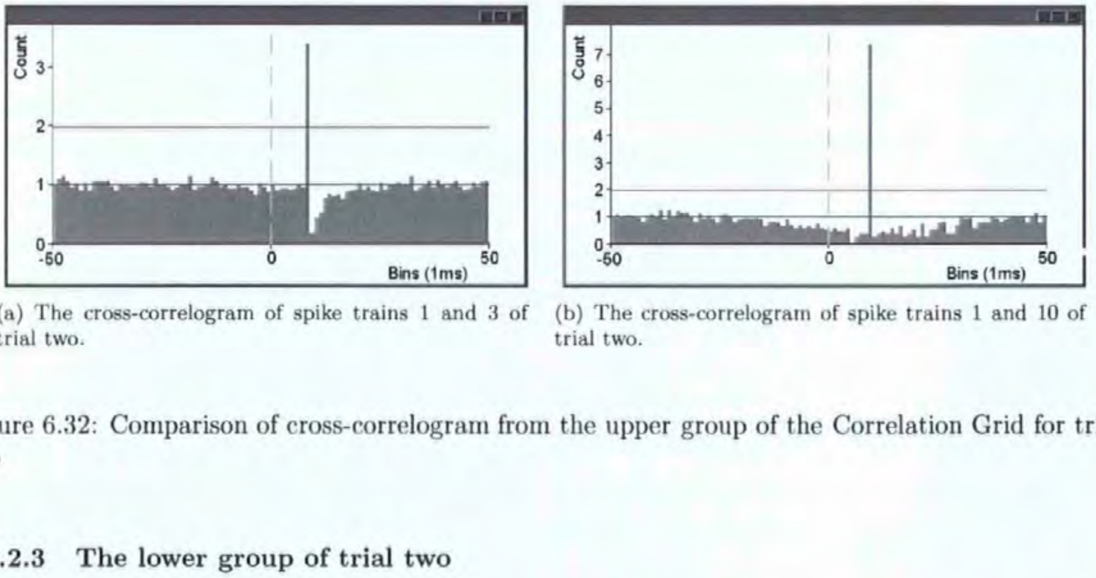


Figure 6.31: The neuronal assembly of the upper group of trial two.

In addition, it is possible to observe that the strength of the connection between neurons 1 and 3, see figure 6.32(a) for the cross-correlogram, is weaker than that of the connection between the neuron pair (1, 10), see figure 6.32(b). Note that the cross-correlogram for spike trains (1, 4) is similar to that of the cross-correlogram of neuron pair (1, 10).



(a) The cross-correlogram of spike trains 1 and 3 of trial two.

(b) The cross-correlogram of spike trains 1 and 10 of trial two.

Figure 6.32: Comparison of cross-correlogram from the upper group of the Correlation Grid for trial two

6.2.2.3 The lower group of trial two

As with the upper group, to aid interpretation of the lower portion of the Correlation Grid it has been enlarged and is shown in figure 6.33.

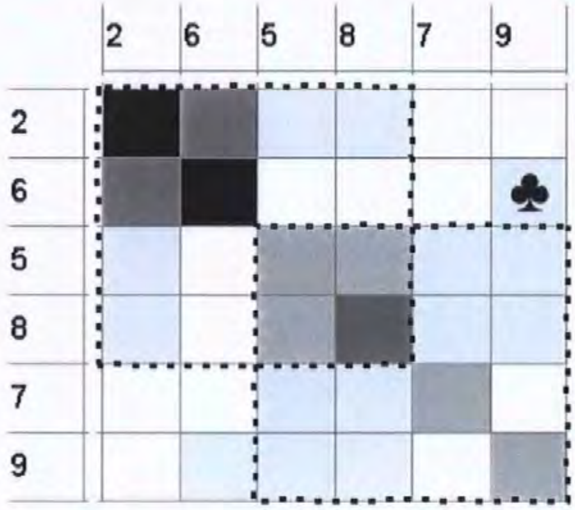


Figure 6.33: Enlargement of the lower group of the Correlation Grid of trial two data.

From this figure, two overlapping groups are apparent. The top group, of this portion of the grid, consists of the spike trains of neurons 2, 6, 5 and 8; and the bottom group consists of the spike trains for neurons 5, 8, 7 and 9. Recall, from the analysis of the first trial (§6.1.2.4), that overlapping groups tend to indicate connected hierarchies.

The interpretation of the bottom group is relatively straightforward, the pattern is similar to that of the upper group of this trial (§6.2.2.2). The bottom group represents a common input circuit, with neuron 5 providing common input to neurons 7, 8 and 9. This hypothesis can be confirmed by inspecting the individual cross-correlogram for this portion of the grid. Note that the cross-correlograms for each of the spike train pairs (5, 7), (5, 8) and (5, 9) exhibit a significant peak with positive delay. Moreover, the cross-correlograms between spike trains 7, 8 and 9 exhibit a significant peak with zero delay, indicating that the neurons tend to spike coincidently.

The top group depicts neuron 2 connecting to both neurons 5 and 6. In addition, a connection exists from neuron 2 to neuron 8. It is likely that this is attributable to the connection from neuron 2 to neuron 8 via neuron 5. By examining the details of the cross-correlogram, between spike trains 2 and 8, this hypothesis was verified. The cross-correlogram displays a significant peak with a positive delay approximately twice that of the peak in the cross-correlogram for spike trains 2 and 5. The link between the top and bottom groups, for this portion of the Grid, is clearly via neuron 5.

In addition to these relationships, the Grid shows a second link between the two groups, from neuron 6 to neuron 9, indicated in figure 6.33 by a club symbol (♣).

Therefore it is possible to propose the assembly shown in figure 6.34 was the functional structure of the lower group shown in the Correlation Grid.

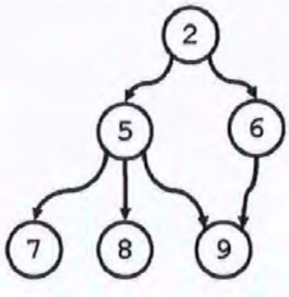


Figure 6.34: Proposed assembly for the lower group of the Correlation Grid of trial two data.

6.2.2.4 Summary of the Correlation Grid observations

From this second stage of analysis of the dataset for trial two, it is possible to propose the neuronal assembly depicted in figure 6.35.

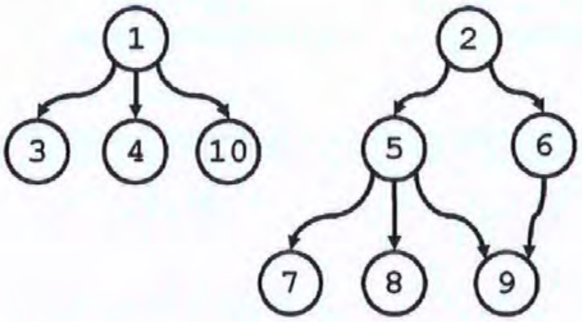


Figure 6.35: The proposed neuronal assembly of trial two.

The proposed structure supports the first four hypotheses generated during the first stage of analysis. The functional relationship of neurons 5, 7 and 9 is clear. In addition, the feasible relationship between neurons 1, 3, 4 and 10 is shown. Moreover, the ambiguity surrounding neurons 2, 6 and 8 is resolved.

After completion of the second stage analysis, the Spike Train Tunnel is used to examine the finer details of the temporal relationships between the spike trains.

6.2.3 Stage 3: Spike Train Tunnel analysis

To provide confirmation of the neuronal assembly proposed in figure 6.35, the dataset for trial two was further analysed using the Spike Train Tunnel. As previously, the spike trains in the Tunnel representation have been reordered according to the results of the cluster analysis of this dataset.

The Tunnel analysis was broken into three sections, based on the functional relationships found during the previous analysis stages. These groups are:

- the sub-group of neurons 1, 3, 4 and 10
- the sub-group of neurons 5, 7, 8 and 9
- the sub-group of neurons 2, 5, 6 and 9

6.2.3.1 The sub-group of neurons 1, 3, 4, and 10

To aid clarity, the proposed architecture of this common input structure, from the stage two analysis, is reproduced in figure 6.36.

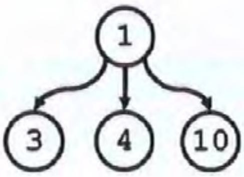


Figure 6.36: The neuronal assembly proposed for neurons 1, 3, 4 and 10 of trial two.

Figure 6.37 shows a snapshot of the Spike Train Tunnel generated for this dataset. Note that trains 1, 3, 4 and 10 are highlighted for clarity.

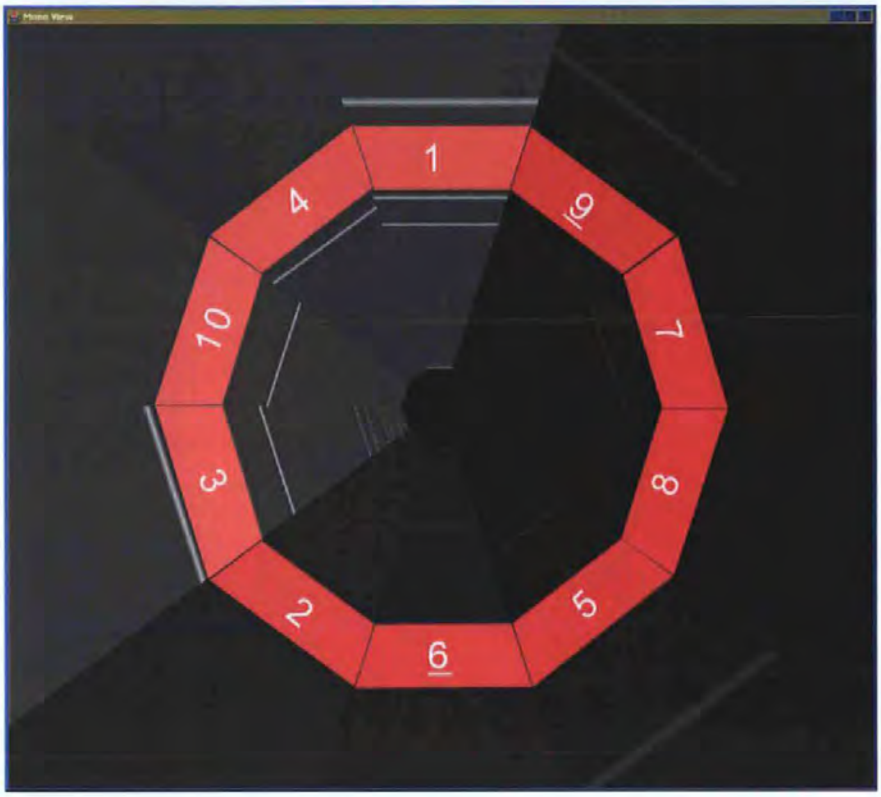


Figure 6.37: A snapshot of the Spike Train Tunnel for trial two, with spike trains 1, 3, 4 and 10 highlighted.

Note the first two spikes on spike train 1. These spikes are followed (with differing delays) by spikes on spike trains 3, 4 and 10. Moreover, this arrangement is prevalent throughout the whole Tunnels length.

It appears, that to generate a spike on spike trains 4 and 10 requires an accumulation of charge greater than that of a single spike from neuron 1. In contrast it is common for neuron 3 to spike after each spike from neuron 1.

Also note the first spike visible on spike train 3. This spike is ignored during the previous analysis as it does not form a sequence. Recall that neurons are modelled as point Poisson processes. Therefore, a neuron will randomly generate spikes according to a Poisson distribution. Thus, the first visible spike on spike train 3 can be considered as a randomly generated spike and not part of a spiking sequence.

6.2.3.2 The sub-group of neurons 5, 7, 8 and 9

For clarity the proposed architecture for this group of inter-connected neurons, identified in the stage two analysis, is shown in figure 6.38. Figure 6.39 shows a snapshot of the Tunnel for the trial two dataset, with the spike trains 5, 7, 8 and 9 highlighted.

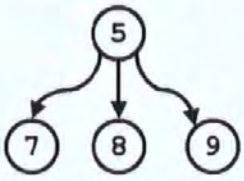


Figure 6.38: The neuronal assembly proposed for neurons 5, 7, 8 and 9 of trial two.

With this common input structure it would be normal to observe a set of near synchronous spikes on spike trains 7, 8 and 9, preceded by a spike on spike train 5. Note this pattern is apparent in figure 6.39, starting with the first spike on spike train 5.

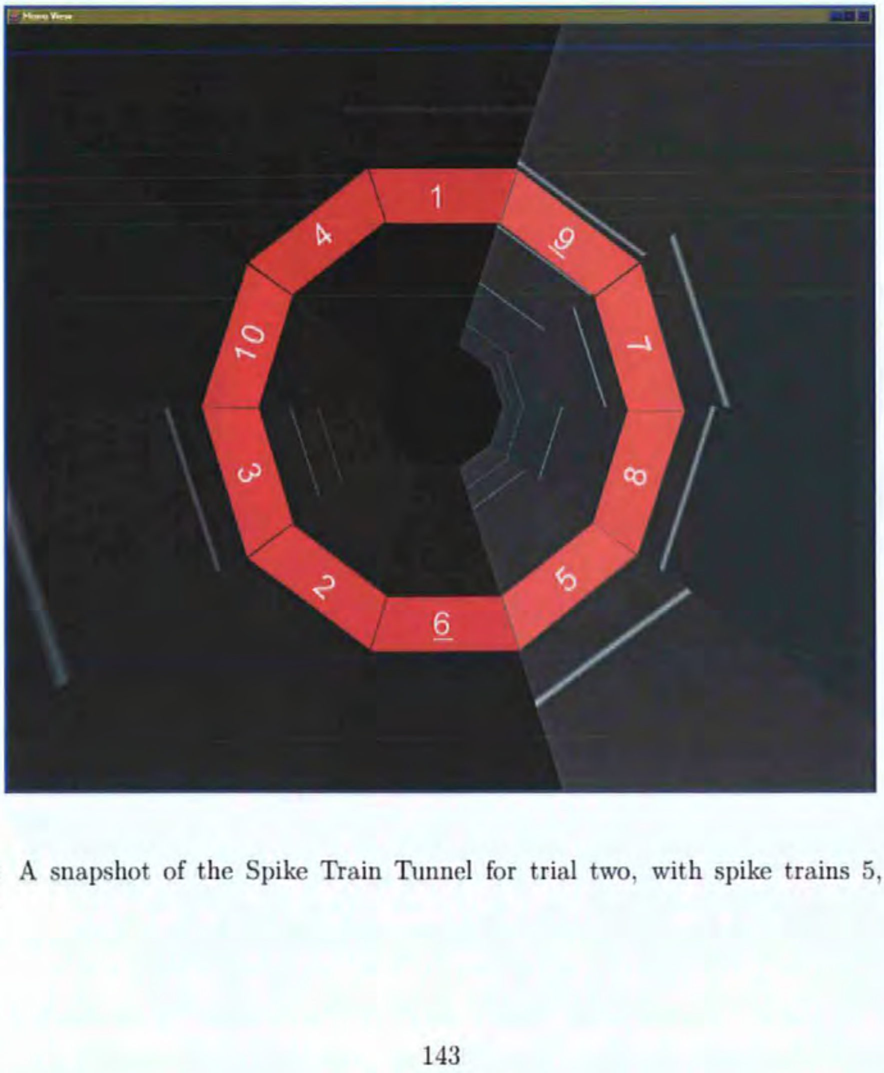


Figure 6.39: A snapshot of the Spike Train Tunnel for trial two, with spike trains 5, 7, 8 and 9 highlighted.

6.2.3.3 The sub-group of neurons 2, 5, 6 and 9

For clarity the proposed architecture for this group of inter-connected neurons, identified in the stage two analysis, is shown in figure 6.40. Figure 6.41 shows a snapshot of the Tunnel with the spike trains 2, 5, 6 and 9 highlighted.

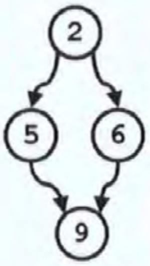


Figure 6.40: The neuronal assembly proposed for neurons 2, 5, 6 and 9 of trial two.

With this structure of neurons, a near synchronous spike on both spike trains 5 and 6, following a spike on spike train 2 is likely to be observed. Moreover, this sequence of spikes is likely to generate a spike on spike train 9, with some delay.

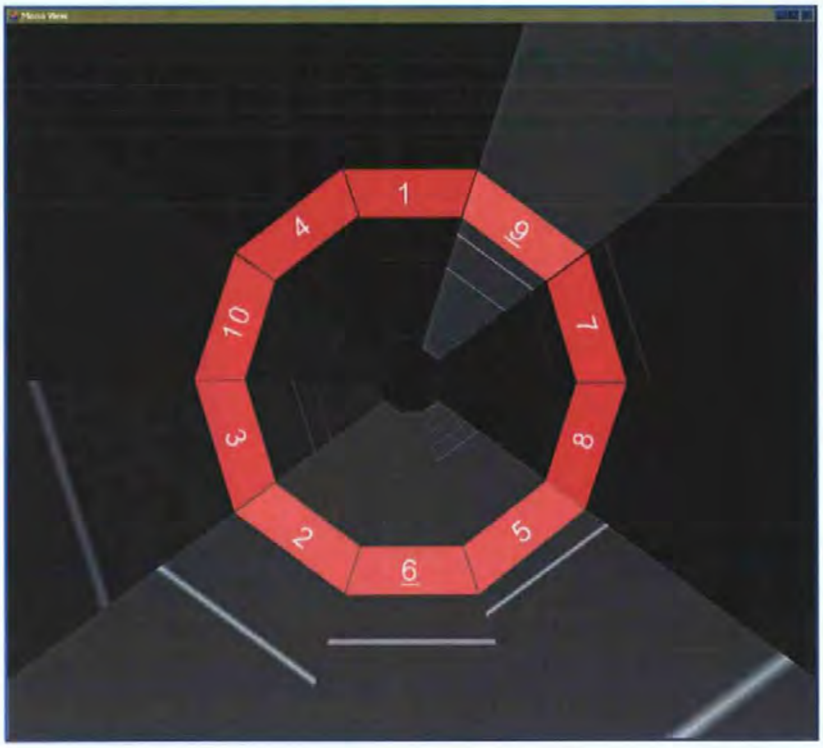


Figure 6.41: A snapshot of the Spike Train Tunnel for trial two, with spike trains 2, 5, 6 and 9 highlighted.

Note the first spike on spike train 2, which is followed by a spike on both spike trains 5 and 6; these spikes are subsequently followed by a spike on spike train 9. Also note, the lower spiking rate of both spike trains 2 and 6, which is obvious throughout the Tunnel.

Recall the slower aggregation of the particles representing neurons 2 and 6 during the Gravity Transform analysis; these two particles did not form part of the main cluster. This lower spiking rate may account for the slower aggregation.

6.2.3.4 Summary of the Spike Train Tunnel analysis

The results of the third stage analysis of the trial two dataset, in the Spike Train Tunnel, reinforces the neuronal coupling structure proposed in figure 6.35. Moreover, additional features have been discovered. The lower firing rates of neurons 2 and 6 are evident in the Tunnel representation of this dataset, this feature helps to explain the slower aggregation, of the related particles, shown in the Gravity Transform.

6.2.4 Summary of the analysis of Trial Two

From the Gravity Transform analysis of this dataset, it was possible to suggest that the underlying neuronal structure consisted of two (or more) separate functional groups. In addition, it was possible to hypothesise that neurons 5, 7 and 9 formed part of one functional group and that neurons 1, 3, 4 and 10 formed part of a second. However, the relationship of these neurons was unclear. Likewise, the relationship of neurons 2, 6 and 8 was ambiguous.

The Correlation Grid analysis confirmed the existence of two separate functional groups, consisting of neurons 1, 3, 4 and 10 and neurons 2, 5, 6, 7, 8 and 9. This reinforced the observations from the Gravity Transform and clarified the relationship of neurons 2, 6 and 8.

Moreover, it was possible to propose a connection architecture for the neurons in these functional groups. Further analysis, in the Spike Train Tunnel, confirmed this structure. In addition, the Tunnel highlighted additional information regarding the firing properties of neurons 2 and 6 which clarifies their clustering pattern in the Gravity Transform.

This process of analysis permitted a functional coupling structure to be proposed for the neurons underlying the spike train dataset in this trial. In addition, details regarding the delays and firing rates of the neurons were also deduced.

Note, the architecture of connections of the neurons in this assembly was unknown to the analyst prior to the investigation. Furthermore, the results of the analysis yielded an entirely accurate neural architecture.

6.3 Trial Three

For this trial, an assembly of 10 neurons were simulated for a period of 100000ms. Note, the architecture of connections of the neurons in this assembly was unknown to the analyst prior to the investigation.

The raster plot for the first 3000ms of this dataset is shown in figure 6.42.

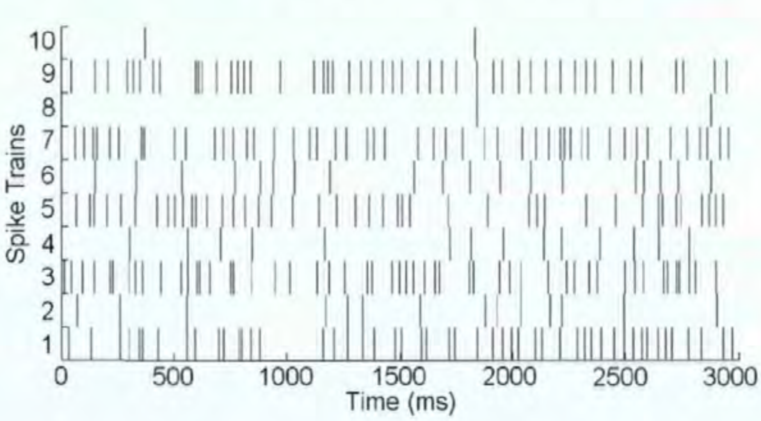


Figure 6.42: Raster plot of trial two spike trains.

6.3.1 Stage 1: Gravity Transform analysis

In this first stage of the analysis of the Gravity Transform of the trial three dataset was computed using a charge increment, $a = 0.2$, charge decay, $\tau = 0.05$ and overall aggregation, $b = 25$.

Recall, in the Gravity Transform, each neuron is represented by a 'particle' and the behaviour of each particle is governed by the corresponding neuron's spike train. In the following analysis, of the Gravity Transform results the particles representing the neurons are analysed.

6.3.1.1 Distance Graph

The Euclidian distance graph for this computation is shown in figure 6.43. Note, all distance pairs are shown.

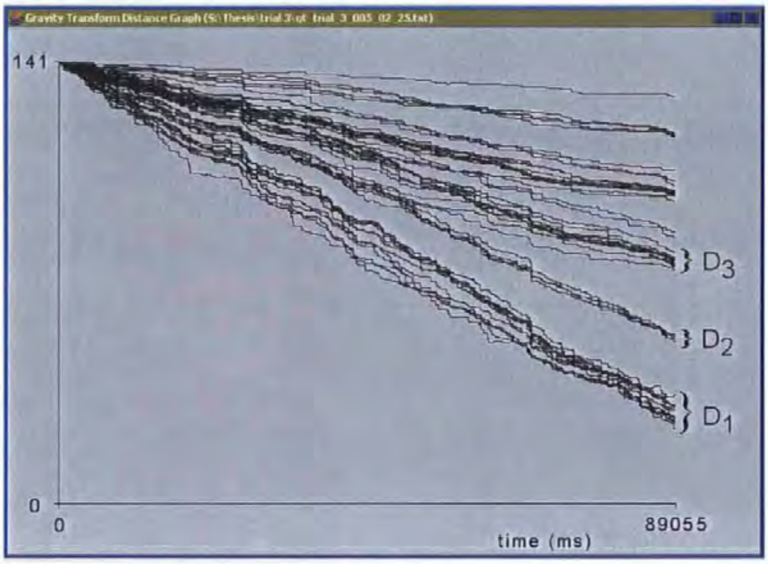


Figure 6.43: Euclidian distance graph of the Gravity Transform for the trial three dataset, with constants $a = 0.2$, $\tau = 0.05$ and $b = 25$, note all distance pairs are plotted.

From the distance graph shown in figure 6.43 it is possible to see a separation of distance pairs. The three groups of most closely related distance pairs have been labelled on the graph. D$_1$ represents distance pairs for particle 1 with 3, 5, 7 and 9; 2 with 3; 3 with 5, 7 and 9; 5 with 7 and 9; and 7 with 9. Group D$_2$ represents the distance pairs of particle 1 with 2; and 2 with 5, 7 and 9. Finally, D$_3$ represents the distance pairs of particle 1 with 4 and 6; 3 with 4 and 6; 4 with 5, 7, and 9; 5 with 6; and 6 with 7 and 9.

From these initial observations, no discernible groups are evident. It appears that all particles are attracted to each other with similar bias. This effect would tend to suggest that the spike trains of the neurons represented by the particles are highly correlated. In turn, the neuronal assembly has a high level of interconnection.

6.3.1.2 PCA

In addition to viewing the results of the Gravity Transform as a distance graph, PCA was applied to this dataset to aid the investigation. A 2-dimensional PCA projection of the Gravity Transform analysis of this dataset is shown in figure 6.44.

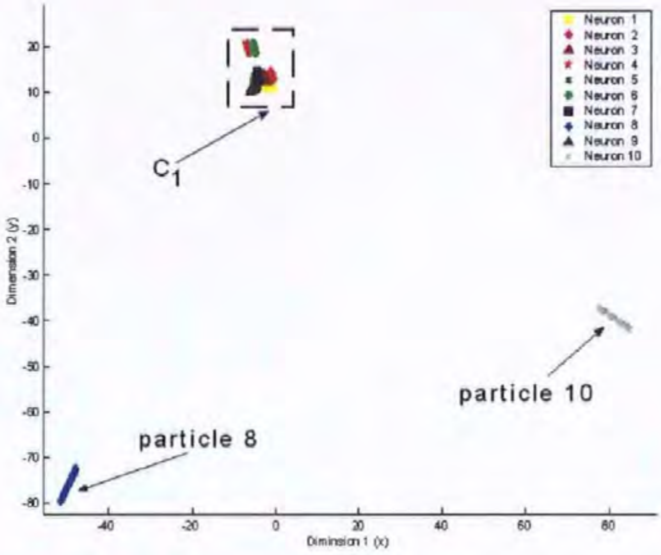


Figure 6.44: Plot of the result of the Gravity Transform for dataset of trial three, with constants $a = 0.2$, $\tau = 0.05$ and $b = 25$, projected onto a 2-dimensional sub-plane using PCA.

From the PCA plot the follow is evident: eight of the ten particles form a cluster, marked as C_1 in the figure; particles 8 and 10 are approximately equidistance from the main cluster. Enlargement of the cluster yields no addition hint to the underlying neuronal grouping represented by the particles.

However, it would appear that both particles 8 and 10 have a lower tendency to cluster with the other particles. It is therefore reasonable to hypothesise that the spike trains for these particles are less strongly correlated with the other spike trains in the dataset. Thus, it is likely that the underlying neurons represented by particles 8 and 10 are less comprehensibly connected with the other neurons.

6.3.1.3 Summary of Gravity Transform analysis

From the results this first stage of analysis of the trial three dataset, it is possible to hypothesise that the spike train dataset was generated from an assembly of neurons with a high level of interconnection. However, it is not possible to draw any conclusions about the internal structure; apart from the fact that it would appear that the spike trains of neurons 8 and 10 are less highly correlated with the others in the dataset.

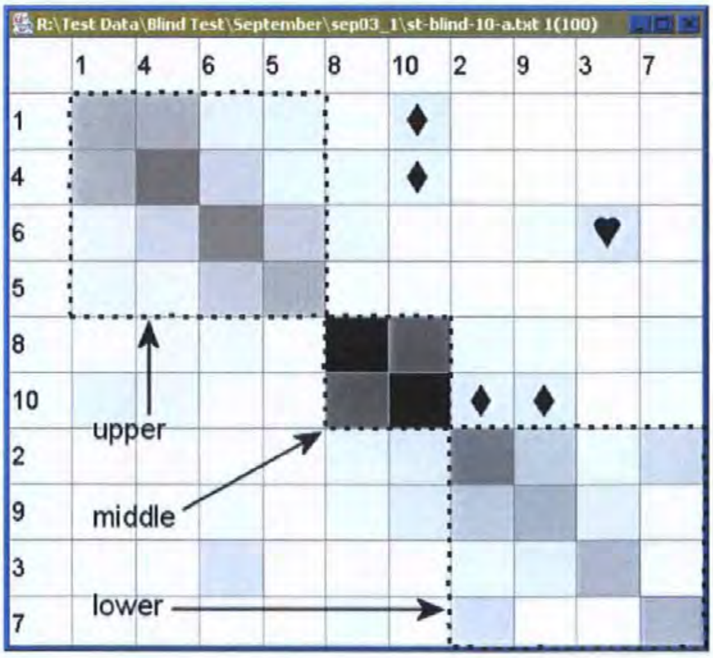
Further detailed analysis is required to extract any information regarding the functional relationship within this dataset. To achieve this, the trial three dataset was further analysed using the Correlation Grid.

6.3.2 Stage 2: Correlation Grid analysis

6.3.2.1 Creating the Correlation Grid for trial three

The Correlation Grid for the original trial three multi-dimensional spike train dataset was generated, with correlation bin size of 1ms and a correlation window of 100 bins (100ms). Subsequently, the Grid was filtered and clustered. The resultant Correlation Grid is shown in figure 6.45.

Initial inspection of the Grid reveals that there is considerable interconnection within this data set; this is consistent with the findings of the first stage of analysis, using the Gravity Transform. However, it is possible to partition this Grid into three groups: the upper, middle and lower groups as depicted in figure 6.45. The upper and lower groups of this case study are discussed in detail in §6.3.2.2 and §6.3.2.3 respectively.



R:\Test Data\Blind Test\September\sep03_1\st-blind-10-a.txt 1(100)

	1	4	6	5	8	10	2	9	3	7
1						♦				
4						♦				
6									♥	
5										
8										
10							♦	♦		
2										
9										
3										
7										

Figure 6.45: The final Correlation Grid depicting the three main groups of the trial three dataset.

The most dominant feature of the Correlation Grid, shown in figure 6.45, is the strong correlation between spike trains 8 and 10, the middle group as indicated in the diagram. Generally, this feature infers a direct connection between the underlying neurons. Examination of the cross-correlogram for this pair confirms the assertion that there is a connection from neuron 10 to neuron 8 as the cross-correlogram exhibits a significant peak with positive delay, indicating a time delayed spike propagation from neuron 10 to neuron 8. Recall, from the Gravity Transform analysis, that the particles representing neurons 8 and 10 were separated from the cluster of particles.

Another dominant feature of this Correlation Grid is that all three groups are somehow linked to neuron 10. This is indicated by the shading in all of the cells labelled with a diamond (◇). This indicates that neuron 10 is likely to be the root of the assembly.

6.3.2.2 The upper group of trial three

In order to interpret the upper group of trial three, the top portion of the Correlation Grid is enlarged and shown in figure 6.46.

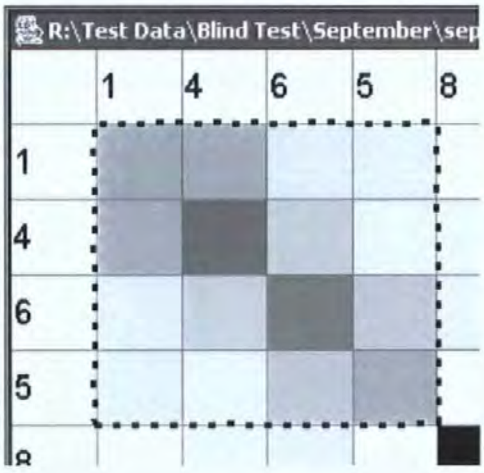


Figure 6.46: Enlargement of the upper group of the Correlation Grid of trial three data.

The upper group consists of neurons 1, 4, 6, and 5. Note that, there is a stronger correlation between spike trains 1 and 4; 4 and 6; 6 and 5 in comparison to the remainder. Therefore, it is likely that these neuron pairs are connected. Examination of their respective cross-correlograms confirms this hypothesis.

The correlation between spike trains 1 and 6 is significant but the reason for this correlation is unclear. The cross-correlogram for this pair is shown in figure 6.47. Note, in this cross-correlogram the main correlation peak occurs around 15ms, this is approximately twice that of the correlation delays observed in cross-correlograms of spike trains (1, 4), (4, 6) and (5, 6). Thus, there is a time delay in the spike propagation from neuron 1 to neuron 6. There is also a second (less significant) peak with a slightly larger time delay.

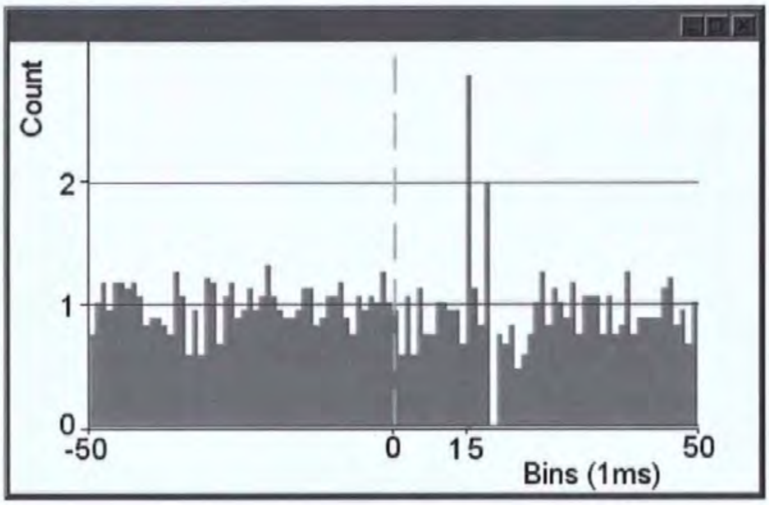


Figure 6.47: The cross-correlogram of spike trains 1 and 6 of trial three.

The increased time delay indicates that the corresponding neurons are connected via an intermediate neuron. The presence of two peaks in the cross-correlogram suggests that two different paths exist between neurons 1 and 6.

Figure 6.48 shows the cross-correlgoram of spike trains 4 and 5 of trial three. The strength of correlation between neurons 4 and 5 also indicates a significant relationship. Upon closer inspection, the cross-correlogram of spike trains 4 and 5 shows a correlation peak close to zero. This indicates that the neurons spike synchronously. Thus, suggesting that the two neurons are receiving correlated input.

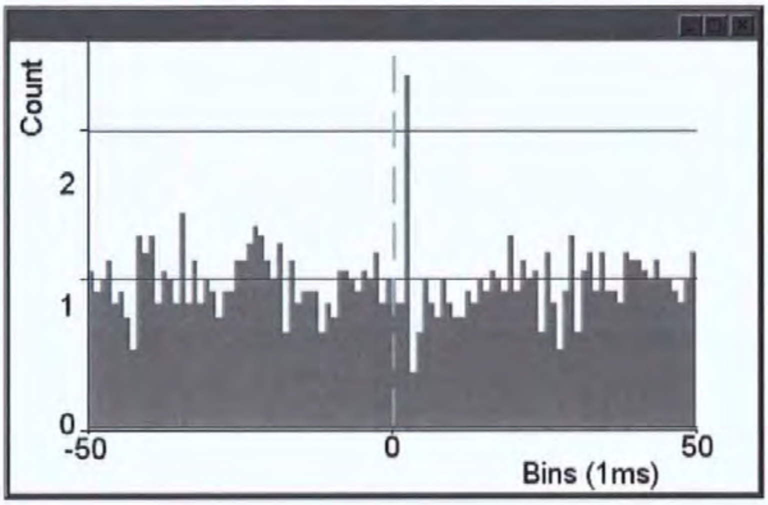


Figure 6.48: The cross-correlogram of spike trains 4 and 5 of trial three.

From these observations, it is possible to deduce the connection scheme shown in figure 6.49. This architecture shows that neuron 1 is a common source to neurons 4 and 5 and that they both have connections to neuron 6.

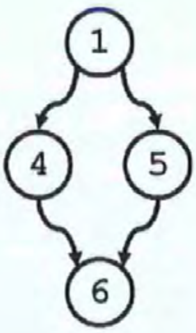


Figure 6.49: The neuronal assembly of the upper group of trial three.

6.3.2.3 The lower group of trial three

In order to interpret the lower group, the lower portion of the Correlation Grid is enlarged and shown in figure 6.50.

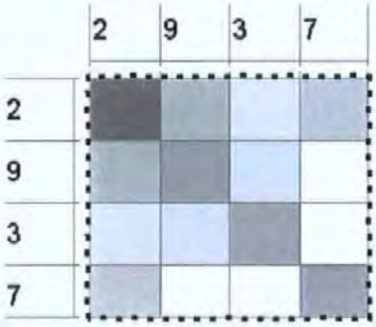


Figure 6.50: Enlargement of the lower group of the Correlation Grid of trial three data.

From this figure, it is apparent that spike train 2 correlates strongly with both spike trains 7 and 9. Likewise, spike train 9 correlates with spike train 3. In addition, spike train 2 correlates, less strongly, with spike train 3. It is possible to hypothesise from these observations that a connection exists between neuron 9 and both neurons 2 and 3. Similarly, that there is a connection between neuron 7 and 3. Inspection of the cross-correlograms for these pairs confirms this hypothesis.

It is also possible to specify the direction of these connections, as shown in figure 6.51. This structure also explains the weaker correlation between spike trains 2 and 3, as these neurons both receive input from neuron 9.

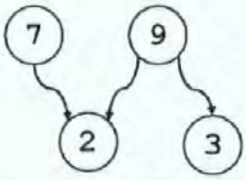


Figure 6.51: The neuronal assembly of the lower group of trial three.

6.3.2.4 Interconnection of upper and lower groups in trial three

As hypothesised previously, neuron 10 links both the upper and lower groups. By investigating the cross-correlograms of spike train 10 with each of the upper and the lower groups, it is possible to deduce that neuron 10 connects to both neurons 9 and 1.

From the grid shown in figure 6.45, it is also possible to observe another link, between the upper and lower groups. This correlation is between spike trains 3 and 6 and is indicated by the heart symbol

(♡).

Note that the cross-correlogram of neurons 3 and 6 showed a single peak, slightly delayed, thus indicating a direct connection between neurons 3 and 6.

6.3.2.5 Summary of the Correlation Grid observations

From the stage two analysis of the trial three dataset, it is possible to propose a coupling structure for the neuronal assembly underlying the spike trains. This structure is depicted in figure 6.52. The structure of this assembly also explains the weak correlations, shown in the grid, which have not yet been considered.

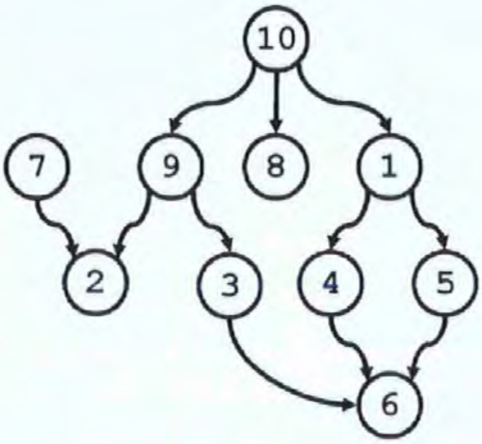


Figure 6.52: The neuronal assembly of trial three.

The correlation between spike trains 10 and 4 is clear; neuron 10 is connected to neuron 4 via neuron 1. Likewise, the correlation between spike trains 8 and 9 is due to common input. The correlations between spike trains 4 and 8; 6 and 8; 2 and 8 is not immediately obvious. However, these correlations are attributable to the fact that all inputs are governed by neuron 10.

6.3.3 Stage 3: Spike Train Tunnel analysis

To further confirm the neuronal assembly proposed from the stage 2 analysis of the trial three dataset, the dataset was further analysed using the Spike Train Tunnel. The spike trains in the Tunnel representation were re-ordered according to the results of the cluster analysis performed on the dataset.

The analysis of this dataset in the Tunnel was broken into three stages. These stages are based on the partitioning of the Correlation Grid and are as follows:

1. the sub-group of neurons 1, 4, 5 and 6;

2. the sub-group of neurons 2, 3, 7 and 9;

3. the sub-group of neurons 10, 1, 8 and 9.

6.3.3.1 The sub-group of neurons 1, 4, 5 and 6

For ease of reference, the proposed connection architecture for neurons 1, 4, 5 and 6 is shown again in figure 6.53. This connection structure shows that the output of neuron 1 is the input of both neurons 4 and 5; and thus, the output of neurons 4 and 5 is the input of neuron 6.

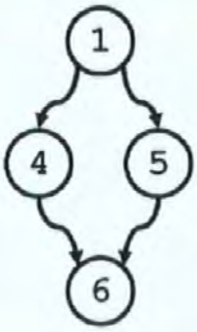


Figure 6.53: Proposed connection structure for neurons 1, 4, 5 and 6

A snapshot of the Tunnel is shown in figure 6.54, note trains 1, 4, 5 and 6 have been highlighted to aid investigation.

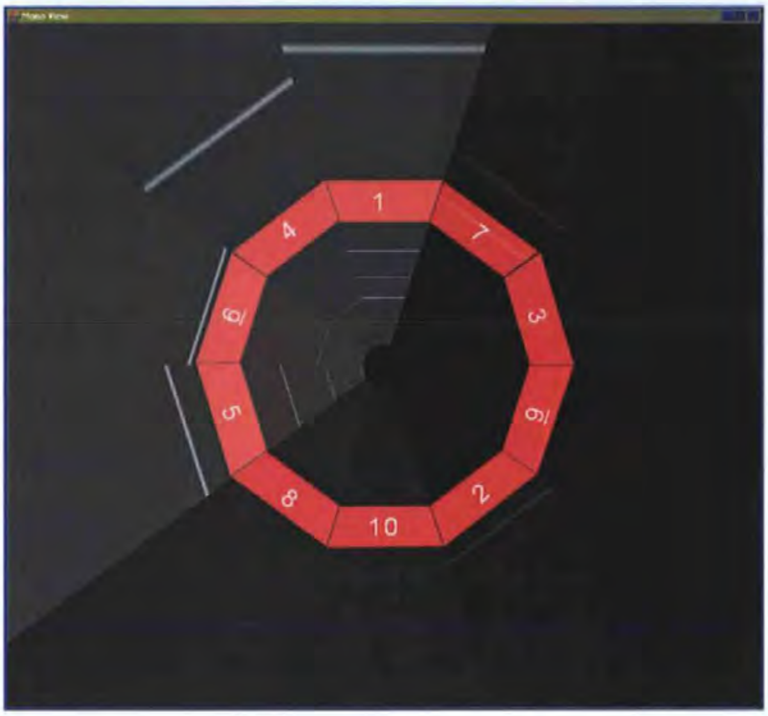


Figure 6.54: Snapshot of the Spike Train Tunnel for trial three, with spike trains 1, 4, 5 and 6 highlighted

Note the first spike on spike train 1, shown in figure 6.54, this spike is subsequently followed by a spike on spike train 4 and then a spike on spike train 5. The spikes on spike trains 4 and 5 are followed by a spike on spike train 6. This sequence of propagated spikes is also evident further along the Tunnel. This pattern of firing supports the proposed structure for these four neurons.

6.3.3.2 Sub-group of neurons 2, 3, 7 and 9

To aid clarity, the proposed structure of neurons 2, 3, 7 and 9 is shown again in figure 6.55.

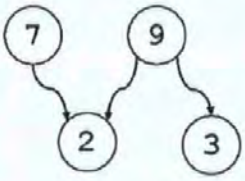


Figure 6.55: The neuronal assembly proposed for the sub-assembly of neurons 2, 3, 7 and 9.

Firstly, the common input relationship between neurons 9, 2 and 3 was examined. Figure 6.56 shows a snapshot of the Spike Train Tunnel, with spike trains 2, 3, and 9 highlighted, which displays a firing pattern that supports the proposed relationship.

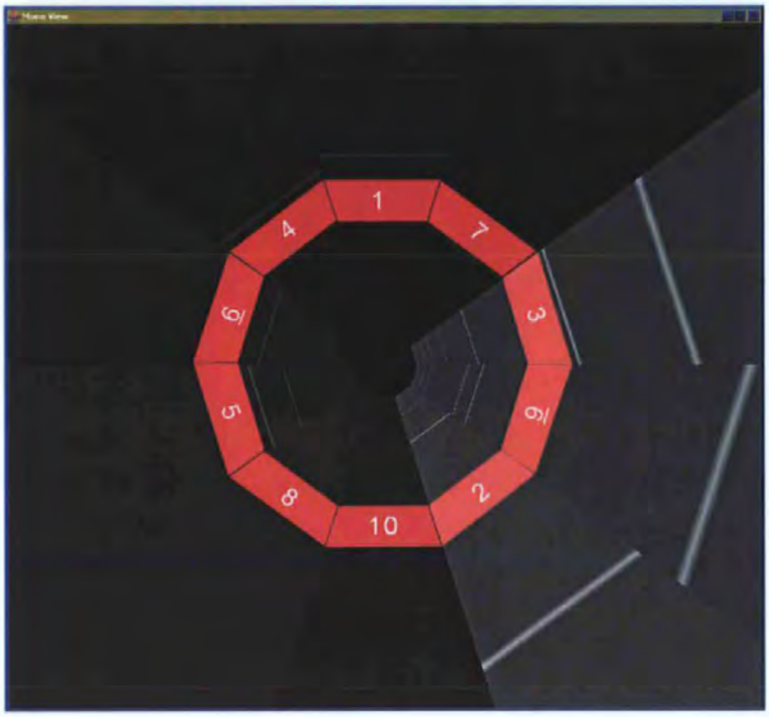


Figure 6.56: Snapshot of the Spike Train Tunnel with trains 2, 3 and 9 highlighted

In figure 6.56, note the coincident spikes on spike trains 2 and 3, the first spike on each train, which are preceded by a spike on spike train 9. Also note the 2^{nd} and 3^{rd} spike on spike train 9, which are both followed by near coincident spikes on spike trains 2 and 3.

Secondly, the common output relationship between neurons 2, 7 and 9 was investigated. For this analysis another snapshot, from further along the Tunnel, is used. This snapshot is shown in figure 6.57, note the relevant spike trains are highlighted.

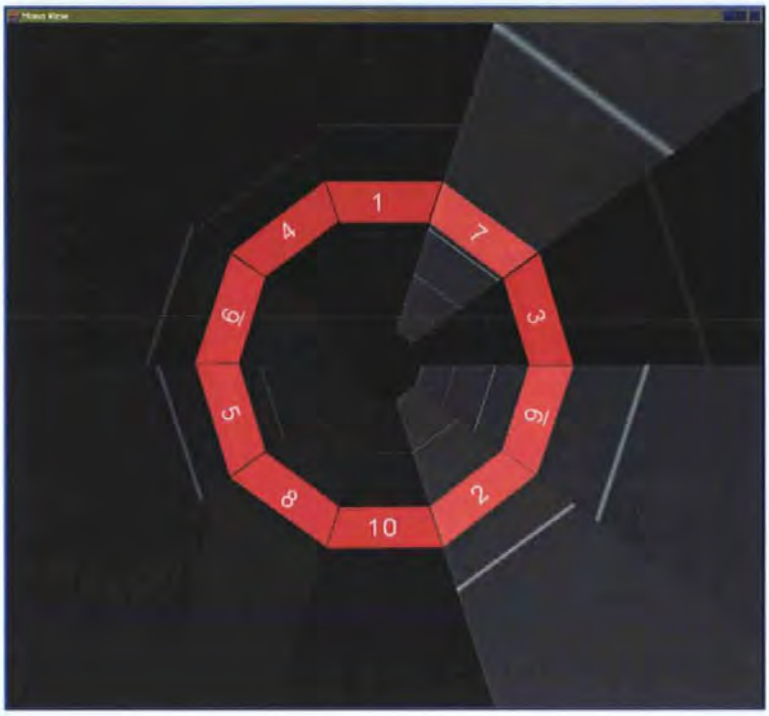


Figure 6.57: Snapshot of the Spike Train Tunnel with trains 2, 7 and 9 highlighted

Observe the first and second spikes visible on spike train 2 in this figure. Note, that both these spikes are preceded by spikes on spike trains 7 and 9.

It is possible to observe these spiking patterns in a number of segments of the Tunnel for this dataset. These observations reinforce the proposed neuronal structure shown in figure 6.55.

6.3.3.3 The sub-group of neurons 1, 8, 9 and 10

The assembly proposed for the neurons in this dataset, from the Correlation Grid analysis, denotes neuron 10 as the root of the main assembly. In addition, neuron 10 supplies common input to neurons 1, 8 and 9. To investigate this structure of the assembly another snapshoot of the Tunnel is used, shown in figure 6.58, note the relevant spike trains are highlighted.

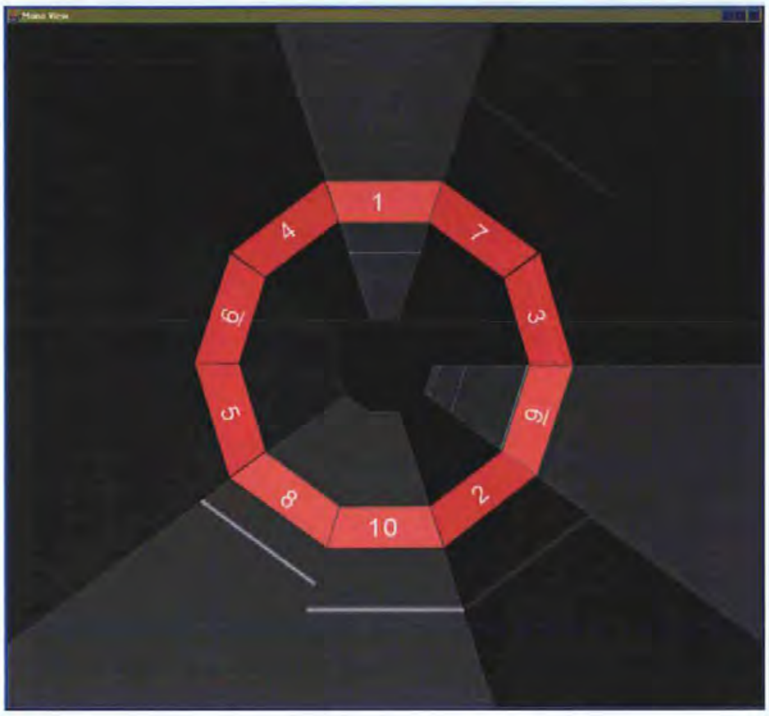


Figure 6.58: Snapshot of the Spike Train Tunnel with spike trains 1, 8, 9 and 10 highlighted

The first feature that is apparent from observing spike trains 8 and 10 (in this segment and in general throughout the Tunnel) is that they both have a much lower firing rate, compared to the rest of the spike trains in the Tunnel.

In addition, it is common for spikes on spike train 8 to be preceded by spikes on spike train 10, as demonstrated by first spike on spike trains 8 and 10 as shown in figure 6.58.

The common input relationship of spike trains 1 and 9 with 10 is noted by the common presence of a spike on train 1 and 9 following a spike on spike train 10. However, it should be noted that there appears to be a significant difference in the propagation delays of this common input set, as it is rare to find coincident spikes on spike trains 1, 8 and 9.

6.3.3.4 Summary of the stage three analysis

The results of the analysis of the trial three dataset in the Spike Train Tunnel help to confirm the neuronal coupling structure proposed in figure 6.52. Moreover, an additional feature has been discovered. The lower firing rate of spike trains 8 and 10 is evident in the Tunnel representation. This behaviour helps to explain the slower aggregation of their related particles shown in the Gravity Transform analysis.

In addition, the variety of propagation delays is evident. For example, from figure 6.58 it is possible to observe that a spike from neuron 10 generates spikes from neurons 1, 8 and 9 with varying delays. The delay for a spike from neuron 8 is relatively short; however, the time taken for a spike to result from neuron 9 is approximately five times longer. Likewise, the propagation delay between neurons 10 and 1 appears to be approximately twice that of the delay between neurons 10 and 9.

6.3.4 Summary of the analysis of Trial Three

The initial analysis of the dataset with the Gravity Transform yielded little information regarding the internal functional structure of the dataset. However the results of the Gravity Transform analysis support the theory that the dataset was generated from a group of highly connected neurons. However, it was apparent that neurons 8 and 10 exhibited a slightly different behaviour in contrast to the rest of the group. The Tunnel analysis showed that this difference was due to the significantly lower firing rates of neurons 8 and 10 compared to the other neurons.

The Correlation Grid analysis supported the theory of a highly connected group of neurons. However, it was possible to segment this group into a number of smaller sub-assemblies. In addition, each of the sub-assemblies was inter-connected. From this analysis, it was possible to propose a plausible functional connection architecture.

This architecture was further tested using the Spike Train Tunnel representation. The results from the Tunnel analysis supported the proposed architecture and also highlighted additional characteristics. The Tunnel representation revealed that neurons 8 and 10 had relatively low spiking rates in comparison to the other neurons; furthermore, it revealed the variation in propagation delays in the root group.

This process of analysis permitted a functional coupling structure to be proposed for the neurons underlying the spike train dataset. In addition, details regarding the delays and firing rates of the neurons were revealed.

Note, the architecture of connections of the neurons in this assembly was unknown to the analyst prior to the investigation. Furthermore, the results of the analysis yielded an entirely accurate neural architecture.

Chapter Summary

This chapter has shown that by combining the analysis and Visualization techniques implemented in the Toolbox, the underlying neuronal assembly of datasets can be extracted. In particular, the second and third studies demonstrated the extraction of a neuronal assembly solely from the multi-dimensional spike train dataset. No other information was available prior to the analysis.

Chapter 7

Evaluation and The Way Ahead

7.1 Introduction

The identification of synchronous spiking activity between the neurons, within multi-dimensional spike train recordings, is paramount to understanding many of the brain's functions. A limited number of methods exist for the analysis of synchronicity between groups of neurons, such as the Gravity Transform. However, the quantity of resultant data from these methods can pose their own analysis problems. Therefore, new innovative techniques are clearly required in order to reformat and present data in a manner that can be explored more easily. It is clear that techniques from the field of Information Visualization have a valuable contribution to make towards solving these problems.

This thesis has presented new contributions to the Gravity Transform algorithm and the visualization of its output. In addition, it has presented a novel visualization technique in which a tunnel is used to represent the individual spikes on spike trains. Finally, the most significant contribution of this thesis, based on the experimental evidence, is the Correlation Grid.

Subsequently, each of these contributions are evaluated and the future direction of each is discussed.

Each of the analysis and Visualization techniques discussed in Chapter 5 is evaluated based on the experience gained from their development and subsequent use in many case studies (including the three case studies represented in Chapter 6). These are:

1. Visualization techniques for use with the Gravity Transform algorithm

2. The Spike Train Tunnel

3. The Correlation Grid

7.1.1 Visualization techniques for use with the Gravity Transform algorithm

The Gravity Transform, developed by Gerstein et al. [GA85], is a very useful tool for the analysis of synchronous spiking within multi-dimensional spike train recordings. Within the Toolbox, several methods have been implemented for visualizing the output of the Gravity Transform[SWB02]. In addition, the original algorithm was adapted to increase the speed and accuracy of calculation[SWB02]. These are:

1. the interactive distance graph
2. PCA plots
3. ICA plots

Case studies one and two described in Chapter 6 demonstrated the usefulness of the Gravity Transform visualization techniques. In particular, they demonstrated how effective the Euclidian distance graph and PCA visualization, of the Gravity Transform output were in identifying the functional separation of the dataset. Thus, it was possible to infer some of the internal relationships.

In contrast, the visualization of the Gravity Transform results of the third case study yielded little information. Note that this case study consisted of a highly interconnected dataset. Thus it is possible that the error induced by reducing the data to two-dimensions may have excluded critical information. Thus, it is likely that if the data was examined in its original high-dimensional state, further details may have been extracted.

Additionally, the use of high dimension representations, such as Parallel Coordinates would be useful in the analysis of these datasets. In particular, versions of Parallel Coordinates that support the animation of datasets, over time, and the identification of clusters, such as the Animator[BS04] tool (also developed by the Visualization Lab, University of Plymouth), have already been shown to be useful.

7.1.2 The Spike Train Tunnel

From the experimental evaluation of the Spike Train Tunnel, as detailed in the case studies in Chapter 6, it has been demonstrated that this Visualization can aid in identifying synchronous firing within multi-dimensional spike train datasets. In particular, this visualization is useful for identifying repeating spiking patterns within the dataset. Moreover, it can be used to infer details regarding the connection delays between neurons.

Evaluation of the Tunnel has identified two possible improvements that could be made to the Tunnel representation. Firstly, the ability to view multiple non-consecutive segments of the dataset simultaneously; and secondly the ability to highlight spikes that occur within a specific time-frame of each other.

7.1.2.1 Multi-Segment Tunnel

It is common for analysts to identify repeating pattern of spikes (sometimes at regular intervals) throughout a dataset. Currently, the Tunnel environment only supports the display of a contiguous section of the dataset. Thus, this limitation can make the comparison of spiking patterns difficult, unless the repeating patterns are very close together.

Therefore, the next version of the Tunnel should facilitate the simultaneous display of a number of different segments of the dataset. This segmented display would permit the analyst to select a number of portions of the dataset in order to compare patterns and investigate repetition.

7.1.2.2 Correlation time-frame indication

The identification of synchronous spikes is relatively straightforward when the number of spike trains is relatively small, up to approximately 20 spike trains. However, as the number of spike trains increases this comparison is increasing difficult, particularly when the synchronous trains are on opposite sides of the Tunnel representation.

To reduce the complexity of this comparison task, it is proposed that the analyst should have the capability to highlight spikes that occur within a specified correlation time-frame. Then, the analyst would be able to position a static overlay onto the Tunnel. This overlay would colour code the spike as the spike trains animate underneath it. This colour coding would aid the analyst in inferring the temporal sequence of spikes within the Tunnel.

7.1.3 The Correlation Grid

The Correlation Grid presents the analyst with an interactive overview of all of the cross-correlations of a multi-dimensional spike train dataset using a simple 2D plane (image). Initially, the order of the spike trains in this display is the order in which they appear in the data file.

The case studies in Chapter 6 demonstrated the usefulness of this method. In all cases, it was possible to identify the underlying neuronal assembly solely from using the Correlation Grid visualization of the multi-dimensional spike train recordings.

The experimental evaluation of this technique has shown it to be extremely effective in the specification of functional relationships within these datasets.

Thus, this visualization technique constitutes a major contribution to the field. It is clear the most significant of the visualization techniques presented in this thesis. This technique has been presented to the user community and the initial responses have been very positive. Further analysis is underway.

Despite the success of this technique, initial feedback has led to the identification of a number of improvements that could be made to the Correlation Grid. These include the introduction of:

1. a selection of spike train ordering algorithms

2. a facility to identify and annotate sub-groups of the Correlation Grid by marking and labelling

3. a peak height encoding method

4. a peak delay encoding method

Once these improvements are made to the current version of the Correlation Grid it is likely to become a key method used by neurophysiologists.

7.1.3.1 Ordering algorithms

Currently the only method to order the spike trains within the Grid is via the cluster analysis algorithm. Despite the effectiveness of this method, demonstrated during the case studies, it is useful to fine-tune the order manually.

Furthermore, the Toolbox should enable the user to specify their own spike train ordering algorithm. And finally, it may be useful to enable the user to fine-tune the order of the spike train manually.

7.1.3.2 Annotation

In the example Correlation Grids previously shown, the sub-groups of spike trains under investigation were marked and labelled on each of the figures for clarity. All of this annotation was performed manually. A dynamic form of this mark-up would enhance the technique and increase efficiency.

7.1.3.3 Peak Height Encoding

Currently, the comparison of cells that are not adjacent in the Grid is problematic due to the limited encoding scope of the grey scale system used. An improvement to the display would be the use of an RGB colour value to encode the height(s) of the peaks(s). Moreover, the analyst should have the capability to partition this encoding scale in a non-linear manner to effectively group peaks of different heights.

In order to support the analyst in this partitioning of the correlation peak heights, a histogram of all significant peak heights could be used to display the distribution of peaks. The analyst could then interactively select partitions of this histogram to encode as different cell colours or sizes. This would significantly improve efficiency in the analysis stage.

7.1.3.4 Peak Delay Encoding Method

Whenever a cross-correlogram of two spike trains exhibits a significant peak this indicates that the corresponding neurons are likely to be connected. However, this connection can be one of three general types:

1. neuron A connected to neuron B with a positive peak delay – this is where the peak is to the right hand side of the origin of the histogram

2. neuron B connected to neuron A with a negative peak delay – this is where the peak is to the left hand side of the origin of the histogram

3. neuron A and neuron B both having a connection from a third neuron, possibly via an intermediate neuron with no peak delay– this is where the peak is at (or very close to) the origin of the histogram

In order to resolve this ambiguity the delay of the main correlation peak must be taken into account. Thus, in addition to encoding the height of the main correlation peak, the delay of this peak should also be encoded in the Grid display.

7.2 Software Testing and User Group Testing

During the development of the various methods of the Toolbox, the algorithms have been tested to evaluate their correctness and to assess their usefulness. To evaluate correctness white box testing was conducted to ensure that the various components performed their specified tasks correctly and efficiently.

In order to assess the usefulness of the techniques a small user group was formed. This comprised the research student undertaking the work, the supervisor and the studies advisor, an eminent Neuroscientist. Due to time limitations, it was not possible to expand this group to include other users. However, the group membership does reflect the user community to which this work contributes.

During training, many different datasets were analysed. Initially, simple, known, assemblies were analysed. This initial testing was undertaken to develop the users capabilities in using each technique.

Subsequently, the capabilities obtained from initial analysis were used in the analysis of more complex assemblies. Furthermore, the testing progressed to the analysis of 'blind' data, where the underlying neuronal structure was unknown to the analyst. The datasets for all of these blind trials were generated by Neuroscientists, to maintain a close relationship with experimentally recorded data.

Three of the key case studies undertaken during testing have been described in detail in chapter 6. This experimental testing has demonstrated that the techniques are extremely useful for investigating firing synchrony within multi-dimensional spike train datasets. These techniques facilitated the identification of functional groups contained in the datasets. Moreover, in most cases the exact neuronal connection architecture used to generate the dataset was recovered.

Within the field of Information Visualization a current and very important principle, is that of user group testing. This has been highlighted at the recent IV04[BBC$^+$04] and InfoVis2003[MN03] conferences. Thus, further detailed testing must be undertaken with domain users, such as Neuroscientists. The results of this testing will in turn drive future development.

7.3 Future Directions

In addition to the improvements to the current tool, the development and testing of these methods has generated ideas for other novel analysis techniques and research directions.

7.3.1 Multiple-View System

One of the main aims of the Visualization of Inter-Spike Association project [SWB] is to produce an integrated investigation environment for Neurophysiologists. In this environment investigators should have the ability to perform multiple analyses on their datasets and view the results in various visualizations simultaneously, thereby producing a multiple-view system.

The usefulness of combining the methods described in Chapter 5 into a multi-view system has been highlighted during the case studies in Chapter 6. In these studies it was necessary to compare the results of different visualization to extract the details of the underlying structures.

Currently, one of the key issues of effective multi-view investigation is the linking of actions between views [Nor01, Rob03]. In order to achieve this, each visualization must have similar, or comparable, interaction primitives. For example, the direct user annotation proposed for the Correlation Grid (see §7.1.3.2) could link with the current filtering processes in the Tunnel (train dimming) and Distance Graph (selective display).

Without this linkage the methods are still useful, but their true power as analysis tools is in their linkage.

A comparative study of the effectiveness of the Toolbox methods, both individually and in a multi-view environment, should be conducted; to examine the benefits of desktop, multiple monitor and large scale displays. In addition, new interaction techniques should be developed to support improved interaction within these environments.

7.3.2 Sonic Model

The Sonic model was inspired by the disciplines of sound design and engineering for theatre, television and film. These fields exploit the ability of the human audio system to integrate disparate sound sources and to mentally produce a 'sound field'. This field permits the sound designer to create the effect of sounds occurring from specific points around the audience. For example, the sound of an explosion from behind an audience, or the sound of a gun shot from the left of a cinema projection screen.

In addition, it is possible to create sound effects that are perceived to move with an object. For example, in a cinema this means that the sound effect is perceived to occur from the same position as the object appear in the picture and the sound appears to move (or track) with the object. Further, when watching a film, with stereo or surround sound, suppose a car enters the picture from the left of the picture, travels across the picture, and leaves the picture from the right. The audio effect of the car 'tracks' the location of the car. This sound tracking is achieved by emitting the sound effect of the car from the left, centre and right speakers with differing delays. The effect is created by the delayed correlation of the sound effect waveforms emitted from the various speakers.

This principle could also be used to develop an innovative method for analysing the synchrony of spiking within multi-dimensional datasets.

It is proposed that the spike trains would be encoded as frequency modulated waveforms. Each waveform would be the output of a point sound source, positioned around a circular perimeter within a modelling space. The interaction of these point sound sources would then be modelled over time.

When two similar waveforms in the modelling space are correlated an amplified waveform would be produced. Thus, two spike trains that exhibit synchronous spiking activity will produce two waveforms that would be correlated within the modelling space. By recording the amplitude of the signals within the modelling space, the synchrony of the spike trains could be examined. Two perfectly correlated spike trains would produce two waveforms that correlated and produced a maximum; amplitude equal to the sum of the two waveforms involved.

The size of the circular perimeter, on which the point sound sources are located, defines the correlation timeframe similar to the correlation window used with the cross-correlation function. Maxima at the centre of this circle would indicate the presence of spike trains that were correlated with no delay. Maxima away from the centre would indicate the presence of correlated spike trains with varying delays. By examining the maxima, the correlation of the underlying spike trains could be evaluated.

Some initial implementation and testing has been undertaken with this method. However, the main implementation is yet to be developed in order to demonstrate usefulness of this method.

7.3.3 Display Walls

During the experimental evaluation of the current methods, the usefulness of collaborative exploration has been highlighted. Moreover, the value of large-scale displays, such as display walls, has also been shown. In particular, the Correlation Grid and Spike Train Tunnel work well on large scale displays.

However, despite the prevalence of large scale displays, the user interaction techniques for these systems are still based on the techniques of standards desktop environments. Where the mouse is an effective, and now ubiquitous, interface for the single display desktop computer, it poses considerable user interaction overheads when used on multiple-display or large scale display systems. The time taken to traverse elongated displays with a traditional mouse is time consuming, compared to the traditional single monitor system for which the mouse is intended.

7.4 Conclusions

Retrieving insight and understanding from multi-dimensional spike train recordings is absolutely paramount to our comprehension of how the Human brain functions. The accuracy and efficiency with which this data can be recorded has increased dramatically over recent years. Thus, a huge amount of experimental data recorded from the Human brain is now recorded and stored, but analysis of this data is still extremely limited. Scientists have only begun to ‘scratch the surface’ in understanding this data and thus, how the brain functions.

The lack of appropriate methods to efficiently analyse this large body of data is a significant limitation in the fight to comprehend the secrets of the Human brain. The techniques proposed in this thesis begin to alleviate this problem, by providing more efficient and effective displays for current analysis techniques; new methods that provide interactive access to the data; and methods that support the analysis of larger datasets. This is the beginning of an important and significant area of research which is capable of improving the treatment of serious brain diseases such as Alzheimer's and Parkinson's.

Bibliography

[3Dm] 3Dmer. How 3d movies work. http://www.mindspring.com/ dmerriman/Imaxwrk.htm.

[AB92] Steve Aukstakalnis and David Blatner. *Silicon Mirage: The Art and Science of Virtual Reality*. Peach Pit Press, 1992.

[AG85] A. M. H. J. Aertsen and George L. Gerstein. Evaluation of neuronal connectivity: Sensitivity of cross-correlation. *Brain Research*, 340:341–354, 1985.

[And00] Keith Andrews. Information visualisation, 2000. http://www.iicm.edu/ivis/.

[Ans01] Josephine Anstey. Building the low-cost vr system, 2001. http://www.ccr.buffalo.edu/anstey/VR/LOW_COST_VR/system.html.

[AQ98] Bouguettaya A and Le Viet Q. Data clustering analysis in a multidimensional space. *Information Sciences*, 112(1-4):267–295, 1998.

[Awi88] F. Awiszus. Continuous functions determined by spike trains of a neuron subject to stimulation. *Biol. Cybern.*, 58:321–327, 1988.

[Awi97] F. Awiszus. Spike train analysis. *Journal of Neuroscience Methods*, 74, 1997.

[Bau] Robert Baumann. Cortex window suite. http://cog.nimh.nik.gov/CORTEX/cortexwc/cortexws.htm

[BB97] R/ Borisyuk and G/ Borisyuk. Information coding on the basis of synchronisation of neuronal activity. *BioSystem*, 40:3–10, 1997.

[BBC+03] Ebad Banissi, Katy Börner, Chaomei Chen, Gorden Clapworthy, Carsten Maple, Amy Lobben, Chris Moore, Jonathan Orberts, Anna Ursyn, and Jian Zhang, editors. *Rendering Symposium*, 2003.

[BBC+04] Ebad Banissia, Katy Börner, Chaomei Chen, Muhammad Dastbaz, Gordon Clapworthy, Anthony Faiola, Ebroul Izuierdo, Carsten Maple, Jonathan Roberts, Chris Moore, Anna Ursyn, and Jian J. Zhang, editors. *Proceedings of the Eighth International Conference on Information Visualization - IV'04*. IEEE Computer Society, 2004.

[BC87] Richard A. Becker and William S. Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, 1987.

[BE96] Thomas Ball and Stephen G. Eick. Software visualization in the large. *IEEE Computer*, 29(4):33–43, 1996.

[BG00] S. N. Baker and G. L. Gerstein. Improvements to the sensitivity of the gravitational clustering for multiple neuron recordings. *Neural Computation*, 12:2597 – 2620, 2000.

[BHvW00] M. Bruls, K. Huizing, and J. van Wijk. Squarified treemaps. In *In Proc. of Joint Eurographics and IEEE TCVG Symp. on Visualization (TCVG 2000)*, pages 33–42, 2000.

[BJD81] Gautam Biswas, Anil K. Jain, and Richard C. Dubes. Evaluation of projection algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(6):701–708, 1981.

[BKM04] Emery N. Brown, Robert E. Kass, and Partha P. Mitra. Multiple neural spike train data analysis: state-of-the-art and future challenges. *Nature Neuroscience*, 7:456 – 461, 2004.

[BLP68a] P. Bessou, Y. Laporte, and B. Pagès. Frequencygrams of spindle primary endings elicited by stimulation of static and dynamic fusimotor fibres. *The Journal of Physiology*, 196(1):47–63, May 1968.

[BLP68b] P. Bessou, Y. Laporte, and B. Pagès. A method of analysing the responses of spindle primary endings to fusimotor stimulation. *The Journal of Physiology*, 196(1):37–45, May 1968.

[BM86] William Buxton and Brad A. Myers. A study in two-handed input. In *Proceedings of CHI'86*, pages 321–326, 1986.

[Bor02] R. M. Borisyuk. Oscillatory activity in the neural network of spiking elements. *BioSystems*, 67:3–16, 2002.

[Bri79] David R. Brillinger. Confidence intervals for the crosscovariance function. *Selecta Statistica Canadiana*, V:1–16, 1979.

[BS04] N. Barlow and L. J. Stuart. Animator: A tool for the animation of parallel coordinates. In *Information Visualization, IV04*, pages 725–730, 2004.

[Bux86] William Buxton. There's more to interaction than meets the eye: Some issues in manual input. In *User Centered System Design: New Perspectives on Human-Computer Interaction*, pages 319–337, 1986.

[BWK00] Michelle Q. Wang Baldonado, Allison Woodruff, and Allan Kuchinsky. Guidelines for using multiple views in information visualization. In *Advanced Visual Interfaces*, pages 110–119, 2000.

[Car] Peter A. Cariani. Temporal coding of sensory information in the brain. http://homepage.mac.com/cariani/CarianiWebsite/CarianiTempCodes.pdf.

[CCNL$^+$97] D. Cook, C. Cruz-Neira, Uli Lechner, Laura Nelson, Anthony Olsen, Sue Pierson, and Juergen Symanzik. Using dynamic statistical graphics in a highly immersive virtual reality environment to understand multivariate (spatial) data. *International Statistical Institute*, August, 1997.

[Che03] Hong Chen. Compound brushing. In *IEEE Symposium on Information Visualization*, pages 181–188, 2003.

[CHLS03] Christoph Csallner, Marcus Handte, Othmas Lehmann, and John Stasko. Fund explorer: Supporting the diversification of mutual fund portfolios using context treemaps. In *IEEE Symposium on Information Visualization*, pages 203–208, 2003.

[CMS88] Micheal Chen, S. Joy Mountford, and Abigail Sellen. A study in interactive 3-d rotation using 2-d control devices. *Computer Graphics*, 22(4):121–129, 1988.

[Cor] IMAX Corperation. Imax website. http://www.imax.com/.

[COS80] R. Cooper, J. W. Osselton, and J. C. Shaw. *EEG Technology*. Butterworths & Co (Publishers) Ltd, third edition, 1980.

[Dat] Official website of the datamunch project. http://www.novl.indiana.edu/ dmunch/.

[DFB00] Andrew Davison, Jianfeng Feng, and David Brown. Spike synchronization in a biophysically-detailed model of the olfactory bulb. *Neurocomputing*, 38-40:515–521, 2000.

[EA95] C. ERE and J. AND. Interacting with huge hierarchies: Beyond cone trees. In *In Proceedings of InfoViz'95, IEEE Symposium on Information Visualization*, pages 74–78, 1995.

[Ell78] P. H. Ellaway. Cumulative sun technique and its application to the analysis of peristimulus time histograms. *Electroencephalogr. Clin. Neurophysiol.*, 45:302–304, 1978.

[Enc99] Jones Telecommunications & Multimedia Encyclopedia. Virtual reality, 1994-99. http://www.digitalcentery.com/encyclo/update/vr.html.

[FNE$^+$99] P Fries, S Neuenschwander, A K Engel, R Goebel, and W Singer. Rapid feature selective neuronal synchronization through correlated latency shifting. *Nature Neuroscience*, 4(2):194–200, 1999.

[Fri87] Jerome H. Friedman. Exploratory projection pursuit. *Journal of the American Statistical Association*, 82(397):249–266, 1987.

[FT74] Jerome H. Friedman and John W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, c-22(9):881–890, 1974.

[FvDFH96] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics Principles and Practice.* Addison-Wesley Publishing Company, second edition, 1996.

[FWR99] Y Fua, M Ward, and E Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *In Proceedings of Visualization'99*, pages 58–64, 1999.

[GA85] George L Gerstein and A M Aertsen. Representation of cooperative firing activity among simultaneously recorded neurons. *Journal of Neurophysiology*, 54(6):1513–1528, 1985.

[Gib77] J. Gibson. The theory of affordances. In R. Shaw and J. Bransford Erlbaum, editors, *Perceiving, Acting and Knowing: Toward an Ecological Psychology*, 1977.

[GK60] G. L. Gerstein and N. Y. S. Kiang. An approach to the quantitative analysis of electroohysiological data from single neurons. *Journal of Biophys*, 1:15–28, 1960.

[GPE85] George L. Gerstein, D. H. Perkel, and Dayhoff J. E. Cooperative firing activity in simultaneously recorded populations of neurons: Detection and measurement. *Journal of Neuroscience*, 5(4):881–889, 1985.

[GW69] R Gnanadesikan and M B Wilk. Data analytic methods in multivariate statistical analysis. *Multivariate Analysis*, 2:593–638, 1969.

[HBMS03] Harry Hochheiser, Eric H. Baehrecke, Stephen M. Mount, and Ben Shneiderman. Dynamic querying for pattern identification in microarray and genomic data. In *Proceedings IEEE International Conference on Multimedia and Expo*, 2003.

[Hic01] Martin Hicks. A helix metaphor for customer behaviour visualisation. In *IV2001 Information Visualisation*, pages 22–28, 2001.

[HO00] A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13:411–430, 2000.

[Hub85] Peter J. Huber. Projection pursuit. *The Annals of Statistics*, 13(2):435–475, 1985.

[ID90] A Inselberg and B Dimsdale. Parallel coordinates: A tool for visualising multidimensional geometry. In *In Proceedings of Visualization'90*, pages 361–378, 1990.

[Isd93] Jerry Isdale. What is virtual reality?, 1993. http://www.cms.dmu.ac.uk/ cph/VR/whatisvr.html.

[IT00] Hiroyuki Ito and Satoshi Tsuji. Model dependence in quantification of spike interdependence by joint peri-stimulus time histogram. *Neural Computation*, 12(1):195–217, 2000.

[JKM03] T.J. Jankun-Kelly and Kwan-Liu Ma. Moiregraphs: Radial focus+contect visualzation and interaction for graphs with visual nodes. In *Proceedings of InfoVis 2003, IEEE Symposium on Information Visualization*, pages 59–66, 2003.

[JM95] B. D. Johnson and K. W. J. Malafant. Visualisation techniques for geoscience data and modelling. In *Proceedings of the Third National Conference on the Management of Geoscience Information and Data*, pages 18.1–4, 1995.

[JS87] M. C. Jones and Robin Sibson. What is projection pursuit? *Journal of the Royal Statistical Society*, 150(1):1–36, 1987.

[KS96] Eser Kandogan and Ben Shneiderman. Elastic windows: Improved spatial layout and rapid multiple window operation. In *Asvanced Visual Interface, ACM AVI'96*, pages 27–29, 1996.

[LN03] Qing Li and Chris North. Empirical comparison of dynamic query sliders and brushing histograms. In *IEEE Symposium on Information Visualization*, pages 147–153, 2003.

[LRP95] J. Lamping, R. Rao, and P. Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *ACM Conference on Human Factors in Software (CHI '95)*, 1995.

[Ltd01] Macmillian Magazines Ltd. Windows on the brain. *Nature*, 412:266–268, July 2001.

[McA93] David F. McAllister, editor. *Stereo Computer Graphics and Other True 3D Technologies*. Princeton University Press, 1993.

[MCLA98] Salzman M., Dede C., B. Loftin, and K. Ash. Vr's frames of reference: A visualization technique for mastering abstract information spaces. In *Third International Conference on Learning Sciences*, pages 249–255, 1998.

[MDB87] B. H. McCormick, T. A. Defanti, and M. D. Brown. Visualisation in scientific computing. *Computer Graphics*, 21(6), November 1987.

[MDH$^+$03] Alan MacEachren, Xiping Dai, Frank Hardisty, Diansheng Guo, and Gene Lengerich. Exploring high-d spaces with multiform matrices and small multiples. In *IEEE Symposium on Information Visualization*, pages 31–38, 2003.

[MFH95] Sougata Mukherjea, James D. Foley, and Scott Hudson. Visualizing complex hypermedia networks through multiple hierarchical views. In *Proceedings of ACM CHI'95*, pages 331–337, 1995.

[MGT$^+$03] Tamara Munzner, Francois Guimbretiere, Serdar Tasiran, Li Zhang, and Yunhong Zhou. Treejuxtaposer: Scalable tree comparison using focus+context with guaranteed visibility. *ACM Transactions on Graphics*, 22(3):453–462, 2003.

[MHSG02] Krešimir Mathović, Helwig Hauser, Reinhard Sainitzer, and M. Eduard Gröller. Process visualization with level of detail. In *Proceedings of InfoVis 2002, IEEE Symposium on Information Visualization*, pages 67–70, 2002.

[MK97] Kenneth Mitchell and Jessie Kennedy. The perspective tunnel: An inside view on smoothly integrating detail and context. In *8th Eurographics Workshop on Visualization in Scientific Computing*, 1997.

[MMKM01] Ciaran McDonnell, Hugh McAtamney, Cathal Keegan, and Catherine McMahon. Aspects of virtual reality and visualisation. *International journal of Modern Physics C*, 12(4):581–587, 2001.

[MN03] Tamara Munzner and Stephen North, editors. *Proceedings of InfoVis 2003, IEEE Symposium on Information Visualization*, 2003.

[MUL] University of Pennsylvania Multiple Unit Laboratory. Mulab. Website. http://mulab.physiol.upenn.edu/index.html.

[NCCN98] L. Nelson, D. Cook, and C. Cruz-Neira. Xgobi vs the c2: Results of an experiment comparing data visualization in a 3-d immersive virtual reality environment with a 2-d workstation display. *Computational Statistics*, 1998.

[net04] netViz Corporation, 2004. http://www.netviz.com/tech_overview/o_default.asp.

[neu] Official website of the neuroexplorer package. http://www.neuroexplorer.com/.

[NHa] Neuro-Heuristics. Data structure. http://www-lnh.unil.cd/Appl/tsan2.html.

[NHb] Neuro-Heuristics. Data structure of spike data files in a standard format. http://www-lnh.unil.cd/alnh/dataformat.html.

[Nor01] Chris North. Multiple views and tight coupling in visualization: A language, taxonomy, and system. In *Proc. CSREA CISST 2001 Workshop of Fundamental Issues in Visualization*, pages 626–632, 2001.

[Now97] Lucy Terry Nowell. Graphical encoding in information visualization. In *CHI97, Conference on Human Factors in Computing Systems*, 1997.

[NPAG] University of Illinois Neuronal Pattern Analysis Group, Beckman Institute. Ntsa workbench website. http://soma.npa.uiuc.edu/isnpa/isnpa.html.

[oF] University of Freiburg. Meatools website. http://www.brainworks.uni-freiburg.de/projects/mea/meatools/overview.htm.

[Osa01] Noritaka Osawa. A multiple-focus graph browsing technique using heat models and force-directed layout. In *IV2001 Information Visualisation*, pages 277–283, 2001.

[OT99] Takashi Oshiba and Jiro Tanaka. Three-dimensional modeling environment "claymore" based on augmented direct manipulation technique. In *The 8th International Conference on Human-Computer Interaction (HCI International '99)*, volume 2, pages 1075–1079, 1999.

[Pala] Brad Paley. Digital image design incorporated website. website. www.didi.com.

[Palb] W. Bradford Paley. Textarc website. http://www.textarc.org/.

[Pal02] W. Bradford Paley. Textarc: Showing word frequency and distribution in text, 2002. Interactive poster presented at InfoVis 2002.

[PGST75] Donald H. Perkel, George L. Gerstein, Mark S. Smith, and William G. Tatton. Nerve-impulse patterns: a quantitative display technique for three neurons. *Brain Research*, 100:271–296, 1975.

[PRS$^+$94] Jenny Preece, Yvonne Rodgers, Helen Sharp, David Benyon, Simon Halland, and Tom Carey. *Human-Computer Interaction*. Addison-Wesley, 1994.

[PRS02] Jenny Preece, Yvonne Rodgers, and Helen Sharp. *Interaction Design*. Wiley, 2002.

[PT95] Ken Pimentel and Kevin Teixeira. *Virtual Reality Through the New Looking Glass*. McGraw-Hill, Inc., 2 edition, 1995.

[QHDD01] Brigitte Quenet, David Horn, Gérard Dreyfus, and Rémi Dubois. Temporal coding in an olfactory oscillatory model. *Neurocomputing*, 38-40:831–836, 2001.

[RBR02] Jonathan Roberts, Nadia Boukhelifa, and Peter Rodgers. Multiform Glyph Based Search Result Visualization. In *Proceeding Information Visualization 2002*, pages 549–554. IVS, IEEE, July 2002.

[RCM93] George G. Robertson, Stuart K. Card, and Jock D. Mackinlay. Information visualization using 3d interactive animation. *Communications of the ACM*, 34(4):57–71, 1993.

[RMC91] G.G. Robertson, J.D. Mackinlay, and S.K. Card. Cone trees: Animated 3d visualizations of hierarchical information. In *Human Factors in Computing Systems, CHI '91 Conference Proceedings*, pages 189–194, 1991.

[Rob98a] Jonathan C. Roberts. On encouraging multiple views for visualization. In Ebad Banissi, Farzad Khosrowshahi, and Muhammad Sarfraz, editors, *IV'98 – Proceedings International Conference on Information Visualization*, pages 8–14. IEEE Computer Society, July 1998.

[Rob98b] David Robinson, editor. *Neurobiology*. Springer, The Open University, 1998. ISBN: 3-540-63546-7.

[Rob03] Jonathan C. Roberts, editor. *2003 International Conference on Coordinated and Multiple Views in Exploratory Visualization*. IEEE Computer Society, July 2003.

[Sal99] David Salomon. *Computer Graphics & Geometric Modeling*. Springer, 1999.

[Sam69] J W Sammon, Jr. A nonlinear mapping for data structure analysis. *IEEE Trans. On Computers*, C-18(5), 1969.

[SB03] Christian Seeling and Andreas Becks. Exploiting metadata for ontology-based visual expolration of weakly structured text documents. In *Proceedings of IV'03, IEEE Symposium on Information Visualization*, pages 652–567, 2003.

[SCK+97] J. Symanzik, D. Cook, B. D. Kohlmeyer, U. Lechner, and C. Cruz-Neira. Dynamic statistical graphics in the c2 virtual environment. *Computing Science and Statistics*, 29(2):35–40, 1997.

[SCKCN96] J. Symanzik, D. Cook, B. D. Kohlmeyer, and C. Cruz-Neira. Dynamic statistical graphics in the cave virtual environment. In *Dynamic Statistical Graphics Workshop*, 1996.

[SGK03] Kenneth Summers, Timothy Goldsmith, and Steve Kubica. An experimental evaluation of continuous semantic zooming in progam visualization. In *IEEE Symposium on Information Visualization*, pages 155–162, 2003.

[Shn96] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proc. IEEE Symp. Visual Languages, VL*, pages 336–343, 3–6 1996.

[Shn98] Ben Shneiderman. *Designing the User Interface*. Addison-Wesley, 3 edition, 1998.

[SML97] Will Schroeder, Ken Martin, and Bill Lorensen. *The Visualization Toolkit, An Object-Oriented Approach to 3D Graphics*. Prentice Hall PTR, 2nd edition, 1997. Special Contributors: Lisa Sobierajski Avila, Rick Avila and C. Charles Law.

[SPPC01] J. Z. Simon, S. Parameshwaran, T. Penney, and C.E. Carr. Temporal coding in the auditory brainstem of the barn owl. *Neuroscience Research*, 2001.

[SWB] L Stuart, M Walter, and R Borisyuk. The visualization of inter-spike association (visa) official website. website. www.plymouth.ac.uk/infovis.

[SWB02] L. Stuart, M. Walter, and R. Borisyuk. Visualisation of synchronous firing in multi-dimensional spike trains. *Bio Systems Special Edition*, 67:265–279, 2002.

[SWB04] L. Stuart, M. Walter, and R. Borisyuk. The correlation grid: Analysis of synchronous spiking in multi-dimensional spike train data and identification of feasible connection architectures. *BioSystems*, 2004.

[Tay02] Russell Taylor. Visualization viewpoints. *IEEE Computer Graphics and Applications*, pages 6–10, May/June 2002.

[Tec] DataWave Technologies. Datawave technologies website. http://www.dwavetech.com/.

[Ter73] A Y Terekhina. Methods of multi-dimensional data scaling and visualization (survey). *Avtom. Telemekh.*, 7:80–94, 1973.

[Tuf97] Edward Rolf Tufte. *Visual Explanations*. Graphics Press, 1997.

[Vic00] Jonathan D. Victor. How the brain uses time to represent and process visual information. *Brain Research*, 886:33–46, 2000.

[Vin95] John Vince. *Virtual Reality Ssytems*. Addison-Wesley, 1995.

[VN00] Frëdëric Vernier and Laurence Nigay. Modifiable treemaps containing variable-shaped units, 2000. citeseer.nj.nec.com/vernier00modifiable.html.

[vWN03] Jarke van Wijk and Wim Nuij. Smooth and efficient zooming and panning. In *IEEE Symposium on Information Visualization*, pages 15–22, 2003.

[VWTDS99] B.P. Vos, M. Wijnants, S. Taeymans, and E. De Schutter. Miniature carrier with six independently moveable electrodes for recording of multiple single-units in the cerebellar cortex of awake rats. *Journal of Neuroscience Methods*, 94, 1999.

[Wan] Ruye Wang. Nerual signaling III - neural coding. http://mulan.eng.hmc.edu/ rwang/e180/handouts/signal3/.

[Weg90] E J Wegman. Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association*, 411(85):664–675, 1990.

[Wil92] William H. Press, et al. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1992.

[Wil96] Graham J. Wills. Selection: 524,288 ways to say "this is interesting", 1996. http://www.bell-labs.com/user/gwills/HomeTalk95/apper.html.

[WK97] Matthew O. Ward and Daniel Keim. Screen layout methods for multidimensional visualization. In *Proc. CODATA Euro-American Workshop on Visualization of Information and Data*, 1997.

[WM93] M.A. Wilson and McNaughton. Dynamics of the hippocampal ensemble code for space. *Science*, 261:10551058, 1993.

[WTP+95] J. A. Wise, J. J. Thoma, K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow. Visualizing the non-visual: Spatial anaylsis and interaction with information from text documents. In *Proceedings of InfoViz'95, IEEE Symposium on Information Visualization*, pages 51–58, 1995.

[YPWR03] Jing Yang, Wei Peng, Matthew O. Ward, and Elke A. Rundensteiner. Interactive hierarchical dimension ordering, spacing and filtering or exploration of high dimensional; datasets. In *Proceedings of InfoVis 2003, IEEE Symposium on Information Visualization*, pages 105–112, 2003.

[Zah71] C T Zahn. Graph-theoreical methods for detecting and describing gestalt clusters. *IEEE Trans. Comput.*, C20:68–86, 1971.

Publications

Visualisation of multi-dimensional Spike Trains

E. Stuart, M. Walter and R. Borisyuk

Centre for Neural and Adaptive Systems, School of Computing,
University of Plymouth, Plymouth, Devon, UK, PL4 8AA
lstuart@plymouth.ac.uk
http://www.tech.plym.ac.uk/soc/research/neural/members.html#stuart

1. Introduction

Temporal coding has an important role in the debate on information encoding by spike trains. It establishes that information is encoded in the seemingly random patterns of spikes, even in the exact temporal arrangement of inter-spike intervals. Subsequently, one of basic principles that underlie information processing in the brain is the principle of synchronisation of neural activity [1] [2]. Vast quantities of experimental data and mathematical models indicate that the synchronisation principle may be useful in devising various systems of information processing.

The experimental evidence that is currently available requires analysis in order to extract inherent information. Analysis of multidimensional spike trains using standard tools such as cross-correlograms is increasingly complex due to the quantity of data involved. Hence, new methods of dealing with this data are needed. In 1985, one such analysis tool called the "Gravity Transformation" [3][4] was developed at the Multiple Unit Laboratory at Department of Neuroscience in the University of Pennsylvania [5].

It is based on the principle of gravitational interaction of particles where each neuron is represented by a particle and the movement of that particle is described in an n-dimensional space, where n is the number of neurons under investigation. All particles start equidistant from one another and the gravitational force (or charge) exerted by a particle is proportional to the spike train of the corresponding neuron. Each spike contributes charge and this charge decays exponentially over time. Thus, should two or more neurons spike coincidently, their corresponding particles will have an attractive force that causes the particle to move closer together. Let us suppose that two neurons have an above average synchrony of firing. Over time this would result in a strong attractive force between their corresponding particles. This would result in aggregation in n-dimensional space. Since, significant synchrony can indicate synaptic coupling [6] the aggregation of the particles can show the assemblies represented by the spike train data.

The gravity transformation has made a significant contribution to the field however there are some difficulties with the display of output data for large numbers of particles.

2. Parallel Coordinates

The use of parallel coordinates, originally pioneered in the 1980's, is a technique used to represent diverse sets of multidimensional data. In 1990, Inselberg [7] [8] renewed the use of parallel coordinates for the analysis of large quantities of multidimensional data and introduced some new representation features that have led to a marked increase in their utilization.

Inselberg's representation of parallel coordinates denotes data points as y-axis coordinate values distributed along the x-axis. In this scheme, a specific point in n-dimensional Euclidean space is represented by n y-axis values distributed along the x-axis. In the last decade much research has focused on the development of parallel coordinates in order to analyse even greater quantities of data. An example of this is the concept of hierarchical parallel coordinates {9].

3. Visualisation Tool

It has been established that parallel coordinates can be used to identify correlations between variables and to convey aggregation information. Subsequently, this focuses on the application of parallel coordinates to the visualization of data produced by the gravity transformation in order to support the investigation of greater numbers of neurons. Naturally, the advantage of parallel axes over orthogonal axes is the fact that their limitations are based on the size of the display area available. Note that since there is no loss of data when using parallel coordinates that there is no "cost" to be considered for the gains achieved.

This paper presents a software analysis tool, VISA, used for the Visualization of Inter-Spike Associations that supports the analysis of multidimensional spike trains using both the gravity transformation and parallel coordinates. In additional to this, it provides additional functionality such as animation of the parallel coordinates display over time thus in this case depicting the aggregation of particles in the gravity transformation data. There is also the capability to view the display output in a static mode. Most significantly, the tool supports the display of any subset of particles for closer inspection. Since the range of values represented by each parallel axis is dependent upon the group of particles viewed on that axis, this is effectively a zoom facility.

Currently, the parallel coordinates may be used for relatively larger values of *n* than the standard output display of the gravity transformation. Indeed, provided that scrolling windows are deemed acceptable to the user, there is no theoretical limit to the number of neurons that could be displayed in this manner. However, in practice, significant demands for interaction with the graphical user interface reduce the overall effectiveness of the software tool since user perception is a significant factor in the process. Note that the use of hierarchical parallel coordinates offers additional significant opportunities for future development of the VISA.

References

[1] Borisyuk RM & Borisyuk GN (1997), "Information coding on the basis of synchronisation of neuronal activity" BioSystem, 40, 3-10.

[2] Fries P, Neuenschwander S, Engel AK, Goebel R & Singer W (2001) "Rapid feature selective neuronal synchronization through correlated latency shifting" Nature Neuroscience, 4(2), 194-200.

[3] Gerstein GL & Aertsen AM (1985) "Representation of Cooperative Firing Activity Among Simultaneously Recorded Neurons" Journal of Neurophysiology, 54(6), 1513-1528.

[4] Gerstein GL et al (1985), "Cooperative Firing Activity in Simultaneously Recorded Populations of Neurons: Detection and Measurement" Journal of Neuroscience, 5(4), 881-889.

[5] http://mulab.physiol.upenn.edu/index.html

[6] Baker SN & Gerstein GL (2000) "Improvements to the Sensitivity of the Gravitational Clustering for Multiple Neuron Recordings" Neural Computation, 12, 2597 – 2620.

[7] Inselberg A & Dimsdale B (1990), "Parallel Coordinates: A tool for visualising multidimensional geometry" In Proceedings of Visualization'90, 361-378.

[8] Wegman EJ (1990), "Hyperdimensional Data Analysis Using Parallel Coordinates" Journal of the American Statistical Association, 411(85), 664-675.

[9] Fua Y, Ward M & Rundensteiner E (1999), "Hierarchical parallel coordinates for exploration of large datasets" In Proceedings of Visualization'99, 58-64.

Visualisation of Synchronous Firing in Multi-dimensional Spike Trains

L. Stuart*, M. Walter* and R. Borisyuk*†

*Centre for Neural and Adaptive Systems, School of Computing, University of Plymouth, Plymouth, Devon, UK

†Institute of Mathematical Problems in Biology, Russian Academy of Sciences, Pushchino, Moscow Region 142 290, Russia

Abstract

The gravity transform algorithm is used to study the dependencies in firing of multi-dimensional spike trains. The pros and cons of this algorithm are discussed and the necessity for improved representation of output data is demonstrated. Parallel coordinates are introduced to visualise the results of the gravity transform and principal component analysis is used to reduce the quantity of data represented whilst minimising loss of information.

1 Introduction

Solution of many problems in the field of Neuroscience is associated with the theoretical comprehension of a large body of experimental data. More specifically, investigation of information processing in the nervous system is associated with the analysis of vast quantities of simultaneously recorded multi-dimensional spike train data. Much of this analysis is based on the principle of synchronisation of neural activity (Borisyuk and Borisyuk (1997); Fries et al. (2001)).

The experimental evidence that is currently available requires further, in-depth analysis in order to extract inherent information. Analysis of multi-dimensional spike trains using traditional tools such as cross-correlograms is increasingly complex due to the vast number of pairs involved. Hence, new methods of analysing this data are required. Among other methods, the 'Gravity Transform' algorithm, developed by Gerstein and Aertsen (1985); Gerstein et al. (1985), has a lot of potential.

In this paper, an implementation of the original Gravity Transform algorithm is presented alongside simulation results. Three trials, based on different neuronal assemblies, are presented. This includes a trial where n is large, relative to previously published results for the method. Subsequently, conclusions are drawn regarding the advantages and disadvantages of the method.

As the size of n increased in these trials, the output of data became increasingly complex. In addition to traditional representations of gravity transform data, Parallel Coordinates, were also used to support interpretation of the data. These are an innovative means of representing n-dimensional data sets.

In addition to investigating new methods of visualising these large quantities of data, reduction of data sets was also investigated. Thus, this paper presents the results of using Principal Component Analysis (PCA) to create more manageable data sets whilst maintaining the most significant characteristics of the data. An additional two trials, based on different neuronal assemblies, are discussed to highlight the effectiveness of using PCA. In these trials, PCA is used to reduce data sets, created by the gravity transform, which are subsequently plotted and analysed.

2 The Gravity Transform

The gravity transform is a method of analysis of spike train dependencies and synchronisation based on the principle of gravitational interaction of particles. Each neuron is represented by a "virtual particle; the movement of those particles is described in n-dimensional space, where n is the number of spike trains under investigation. All particles start equidistant.

The gravitational force resulting from the charge of a particle is calculated on the basis of the spike train of its corresponding neuron. Each spike contributes into the charge and this contribution decays exponentially over time. If two or more neurons spike coincidently, their corresponding particles will exert an attractive force that causes the particles to move closer together.

Let us suppose that several neurons have an above average synchrony. Over time this would result in a strong attractive force between their corresponding particles. In turn, this would cause the particles to aggregate into specific patterns in n-dimensional space. Gerstein specifies that over time all particles will eventually aggregate together into a single cluster due to these attractive forces. Since significant synchrony can indicate synaptic coupling, (Baker and Gerstein, 2000), the aggregation of the particles can show the assemblies representing the neuronal interactions. In fact, the aggregation reflects neuronal activity.

Note that all spike trains used for experimentation were generated using an enhanced Integrate and Fire generator defined by Borisyuk and Borisyuk (1997).

2.1 Description of the Gravity Transform Algorithm

Let us consider n simultaneously recorded spike trains with epoch $[0,T]$. Suppose that the i^{th} spike train is represented by spikes at times $T_1, T_2, ..., T_k$. The 'charge' of the i^{th} particle corresponding to the spike train is described by the following procedure. Each spike contributes a quantity of charge a, which decays exponentially over time with constant τ. Thus the charge on particle i at time t depends on the sum of all preceding spikes and is given by

$$q_i(t) = \sum_{m=1}^{k} Q(t - T_m) - \lambda_i$$

$$Q(x) = \begin{cases} a & \textit{if} \ \ x = 0 \\ ae^{-x/\tau} & \textit{if} \ \ x > 0 \\ 0 & \textit{otherwise} \end{cases}$$

where $\lambda_i = a\tau k / T$ is the average firing rate of neuron i. The charge function $q_i(t)$ of the particle is pre-calculated with a time step $\Delta\,(\Delta = 1 \text{ ms})$, thus $q_i(t_j), t_j = j\Delta,\ j = 0,1,...$, and intermediate charge values are linearly interpolated from the stored values. The dynamics of interactive particles in n-dimensional space is governed by the equations:

$$\frac{dx_i^k(t)}{dt} = bq_i(t)\sum_{j=1}^{n} q_j(t)\frac{x_j^k(t) - x_i^k(t)}{R_{ij}(t)}$$

where $(x_i^1(t),..., x_i^n(t))$ is the position of particle i at time t; $k=1,2,...,n$; $i=1,2,...,n$; $R_{ij}(t)$ is the Euclidean distance between particles i and j; a, b and τ are constants. All particles in the system are initialised at time $t = 0$ to be equidistance:

$$x_i^m(0) = 100 \ \textit{if} \ i = m \ \textit{and} \ x_i^m(0) = 0 \ \textit{otherwise}. \quad (1)$$

Integration within this implementation of the algorithm is achieved by an adaptive step-size Runga-Kutta 4th order algorithm (Press et al., 1992). The use of this ODE solver permits the algorithm to progress through the integration with an optimal time step whilst maintaining a low cumulative error compared to an algorithm using a fixed time step.

2.2 Distance Graphs

In the original implementation of the gravity transform method, output is represented by a 'distance graph'. This graph depicts the Euclidean distance between each pair of particles in the system over time.

Using the neuron circuit shown in Figure 2.1, depicting three groups of excitatory neurons and a solitary neuron, spike train data was generated. This trial lasted 5s and neurons had a firing rate of approximately 0.2670 spike/s. A raster plot of a portion of this data is shown in Figure 2.2.

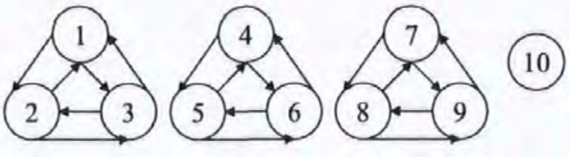


Figure 2.1 Specification of the connections between the 10 neurons used as input to the spike train generator.

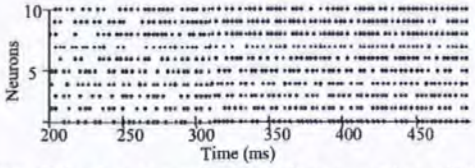


Figure 2.2 A raster plot depicting a portion of the 10 spike trains generated for the neuron assembly shown in Figure 2.1.

The cross-correlograms of four neuron pairs are shown in Figure 2.3.

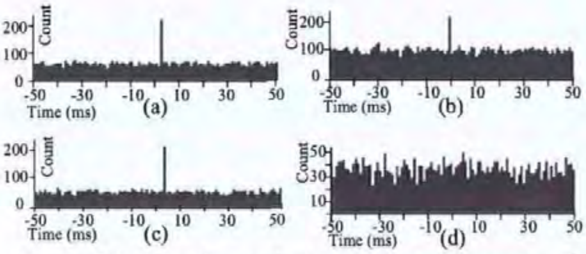


Figure 2.3 This figure depicts cross-correlograms of (a) neurons 1 & 2, (b) neurons 2 & 3, (c) neurons 1 & 3, and (d) neurons 1 & 4.

The correlation of synchronous activity within the first group is shown in Figure 2.3(a)–(c), which clearly shows pair-wise interdependencies (the high peak on each graph depicts synchronous firing). Additionally, the cross-correlogram (different scale) for neurons 1 and 4 is shown in Figure 2.3(d). This shows that no dependency exists between these neurons.

The same data was input to the gravity transform and the distance graph shown in Figure 2.4 resulted. This graph has been annotated, using set theory notation, in order to identify individual distance pair plots. Let d_{ij} represent the Euclidean distance between the particles i and j.

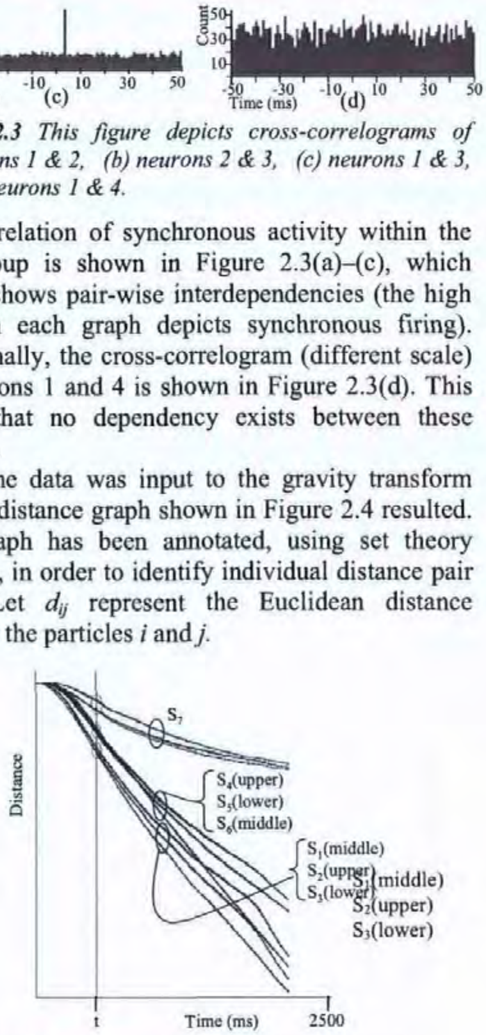


Figure 2.4 Distance graph of the results of the gravity transform where a=0.3, τ=0.4 and b=1.

At time *t* three distinct groups are apparent. The top group represents the distances between neurons 1 to 9 and neuron 10. The lower, and middle, groups represent all intra-group, and inter-group, distances respectively. Note that overlapping occurs.

The final intra-group distances, $s_1=\{d_{12,}\ d_{13,}\ d_{23}\}$, $s_2=\{d_{45,}\ d_{46,}\ d_{56}\}$ and $s_3=\{d_{78,}\ d_{79,}\ d_{89}\}$ show the three groups of neurons aggregating. The final inter-group distances, $s_4=\{d_{ij}{:}i=1,...,3, j=4,...,6\}$, $s_5=\{d_{ij}{:}\ i=1,...,3, j=7,...,9\}$ and $s_6=\{d_{ij}{:}\ i=4,...,6, j=7,...,9\}$, show the distances between the three aggregating groups. The final distances between neurons 1 to 9 and neuron 10, $s_7=\{d_{ij}{:}\ i=10, j=1,...,9\}$, show that the solitary neuron has no tendency to group with any of the other neurons.

This interpretation of a distance graph highlights the usefulness of the technique in identifying groups of neurons within assemblies. However, it should be noted that it is generally difficult to produce a distance graph of similar quality to the one shown in Figure 2.4. The quality of the distance graph is highly influenced by the number of neurons in the assembly and the choice of parameter values: (i) the increment of charge per spike, *a,* (ii) the charge decay rate, τ, (iii) the overall aggregation of the system, *b*. For Figure 2.4, a=0.3, τ=0.4 and b=1.

The gravity transform is sensitive to the "*appropriate*" specification of the constants that represent the decay, increment and aggregation. Inappropriate choice of these values may result in all particles becoming relatively coincidental within *n*-dimensional space, before any useful information about neuronal groups is discovered. Alternatively, it could result in the particles aggregating at such a low rate that they appear unrelated. Hence, it is sometimes necessary for the investigator to 'fine-tune' the specification of these constants in order to gain useful distance graphs.

To demonstrate this problem, first of all, consider the neuron assembly in Figure 2.5.

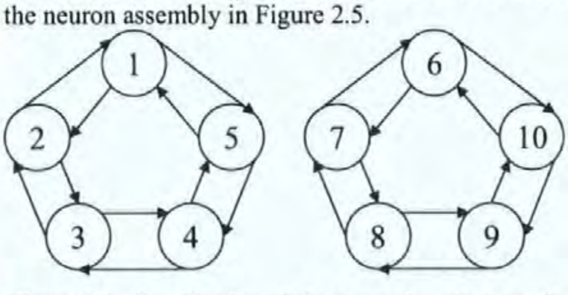


Figure 2.5 *Specification of the connections between the 10 neurons used as input to the spike train generator.*

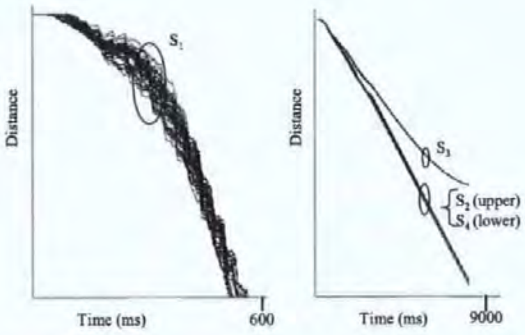


Figure 2.6 *Distance graphs of the results of the gravity transform where n=10, a=0.5, τ=0.5, b=5(left) and b=0.1(right). Note the differing scales.*

This assembly was used to produce spike train data for a trial lasting 20s. This data set was input to the gravity transform with parameters a=0.5, τ=0.5 and b=5. The resultant distance graph is shown on the left in Figure 2.6 where $s_1=\{d_{ij}{:}\ \ i=1,...,10, j=2,...,10, i<j\}$.

Note that it is not possible to distinguish the two groups in the assembly, even though the scale is relatively small .

Using the same input data set, the gravity transform was executed with another set of parameters: a=0.5, τ=0.5 and b=0.1. The resultant distance graph is shown on the right in Figure 2.6 where $s_2=\{d_{ij}{:}\ i=6,...,10, j=7,...,10, i<j\}$, $s_3=\{d_{ij}{:}\ i=1,...,5, j=6,...,10, i<j\}$ and $s_4=\{d_{ij}{:}\ i=1,...,5, j=2,...,5, i<j\}$. From this graph, the structure of the assembly can be derived. Hence, the groups of neurons 1 to 5 and 6 to 10 are notable as is the separation between the groups.

Limited simulation results have shown that appropriate ranges for the parameter values are $0.1<a<0.5$; $0.3<\tau<0.5$ and $0.1<b<$ 2. A method of automatic specification of these parameters, based on the distribution of inter spike intervals, is under investigation.

2.3 Increasing the number of particles

In the relevant publications, the maximum number of particles used within the gravity transform is relatively small, $n \leq 10$. This section reports on trials using the gravity transform based on relatively large values of *n*. In these trials, where *n* is relatively large, the use of distance graphs to display the output proved to be less useful, due to the vast number of individual pair plots that exist.

Consider an assembly of 50 neurons. The connected portion of this assembly, involving 20 neurons, is shown in Figure 2.7. The remaining 30 neurons of this assembly are not shown as they are unconnected.

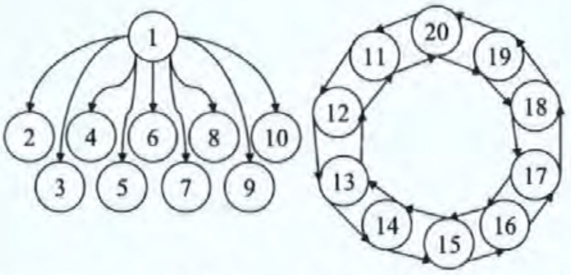


Figure 2.7 Specification of the circuit used by the 20 connected neurons. The remaining 30 neurons are not shown as they are unconnected. The complete assembly was used as input to the spike train generator.

This assembly was used to produce spike train data for a trial lasting 20s. This data was input to the gravity transform, with a=0.4, τ=0.3 and b=1.

The resultant distance graph is shown is Figure 2.8 where $s_1=\{d_{ij}: i=11,...,20, j=12,...,20, i<j\}$, $s_2=\{d_{ij}: i=1, j=2,...,10\}$ and $s_3=\{d_{ij}: i=2,...,10, j=3,...,10, i<j\}$. The different clusters are notable, however, detail is obscured by the quantity of data displayed.

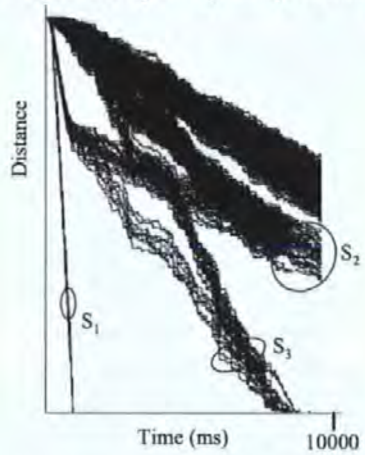


Figure 2.8 Distance graph of the results of the gravity transform where n=50, a=0.4, τ=0.3 and b=1.

Since, this problem increases with the size of n, the interpretation and analysis of the large quantities of output from the gravity transform requires more sophisticated methods of representation. In the remainder of this paper, two solutions to the problem of representing this data are proposed: (a) Utilisation of parallel coordinates to visualise the information, and; (b) Reduction of the quantity of data dimensions displayed whilst minimising loss of information.

3 Data Presentation

With the use of large numbers of particles, the gravity transform results in vast multi-dimensional data sets which represent the position of each particle in every dimension at every time point. The visual representation of the result should accurately convey the position of the particles particularly their direction. This poses a significant analysis problem which may be handled by introducing an alternative method of data representation.

The use of parallel coordinates, originally pioneered in the 1980's, is a technique used to represent diverse sets of multi-dimensional data. In 1990, Inselberg and Dimsdale (1990) and Wegman (1990) renewed the use of parallel coordinates for the analysis of large quantities of multi-dimensional data and introduced new representation features that led to a significant increase in their use.

Inselberg's representation of parallel coordinates denotes data points as vertical axis coordinate values distributed along a horizontal axis. In this scheme, a specific point in n-dimensional Euclidean space is represented by n vertical axes values distributed along the horizontal axis. For example, suppose we have the points a and b in 3-dimensional (p,q,r) space: a(2,0,0) and b(1,2,2). Figure 3.1 depicts these as parallel coordinates using three vertical axes p, q and r.

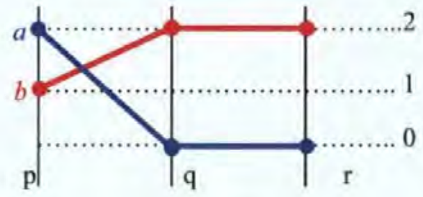


Figure 3.1 Illustration of a 3-dimensional parallel coordinate plot. Representation of points a(2,00) and b(1,2,2), using parallel coordinates.

Parallel coordinates can be used to identify correlations between variables and to convey aggregation information. In this paper we, propose the idea of using parallel coordinates as a simple means of representing n-dimensional coordinates in a 2-dimensional plane. This would represent a snapshot of the gravity transform. It is proposed that these coordinates are animated over time to represent the changing position of particle within the gravitational system.

3.1 Parallel Coordinates: Example One

The assembly of 10 neurons, depicted in Figure 2.5, is used to generate spike train data for 20s. This data is input to the gravity transform and its output is visualised using parallel coordinates. In total there are 20000 intervals (time step=1ms). Hence, the animation is made up of 20000 snapshots. Additionally, since n=10, each snapshot of the parallel coordinates depicts the position of all 10 particles in 10-dimensional space. The legend for all this data is shown in Figure 3.2 and Figure 3.3 depicts snapshot 2000 of the animation

● Particle 1
● Particle 2
● Particle 3
● Particle 4
● Particle 5
● Particle 6
● Particle 7
● Particle 8
● Particle 9
● Particle 10

Figure 3.2 Legend of the animated parallel coordinates.

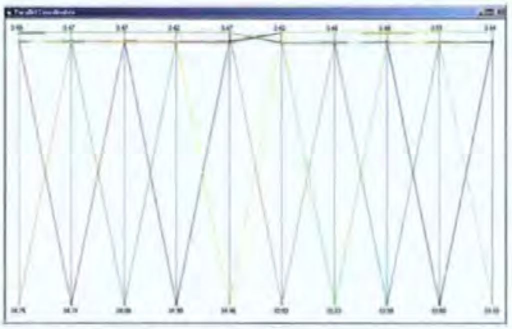


Figure 3.3 Representation of particle positions using parallel coordinates. Snapshot 2000 showing all 10 parallel coordinates in all 10 dimensions.

Recall that all particles in the system are initialised at time $t = 0$ to be equidistance (1).
Hence, the initial range of each vertical axis will be from 0 to 100. In Figure 3.3 some change, from this initial position, is noted in the plot. However, change is more obvious in snapshot 6000, shown in Figure 3.4, where a separation of the data in two groups is noted.

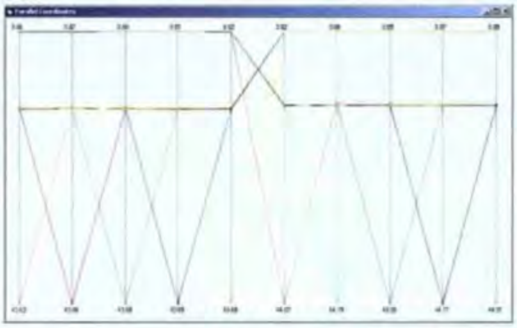


Figure 3.4 Snapshot 6000 showing all 10 parallel coordinates in all 10 dimensions.

Note that snapshot 10000 of this animation, given in Figure 3.5, shows two very distinct groups. Closer examination reveals that all five particles, in the first group, are at the same position denoted in the diagram by the overlap of their parallel coordinates. All particles from the second group are also coincident at a different location to the first group.

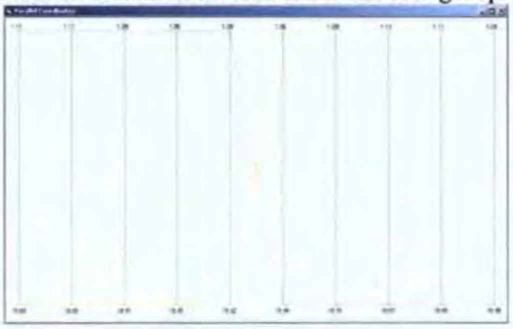


Figure 3.5 Snapshot 10000 showing all 10 parallel coordinates in all 10 dimensions.

Figures 3.3 to 3.5 reveal more information than the distance graph shown on the left in Figure 2.6 even though they were generated from the same data set. Thus, parallel coordinates can be significantly less sensitive to change than distance graph displays.

3.2 Parallel Coordinates: Example Two

The use of distance graphs to view the output from the gravity transform is somewhat limited to smaller values of n. In this section, the data used to create the distance graph in Figure 2.8 is viewed using parallel coordinates. Recall that the assembly of connected neurons is given in Figure 2.7 and that n=50. The trial lasted 20s leading to 20000 snapshots in the animation (time step=1ms).
Note that due to the large number of particles a legend is in not included but grouping is described in detail. Additionally, due to the limitations of the equipment available for displaying output, only the most significant 29 axes of the overall 50 are captured in each snapshot. This is satisfactory to analyse this example as the major activity is confined within neurons 1 to 20.

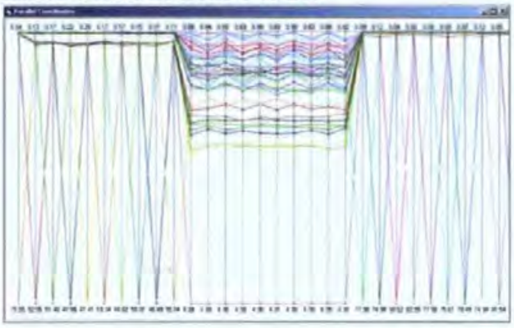


Figure 3.6 Snapshot 5000 of the animation showing all 50 parallel coordinates in 29 dimensions.

Figure 3.6 shows snapshot 5000 of the animation. Close inspection reveals that the group of particles denoted by the straight line along the bottom of the plot includes all the particles 11-20. Note that a zoom facility is used to isolate groups and identify these groups accurately. It is also possible to suggest that two more groups may exist in the top left of the plot.
Figure 3.7, showing snapshot 7500, confirms the conjecture that two more groups exist. Closer inspection reveals that the upper group throughout this plot relates to particles 31 to 50. The lower group that zigzags at the left denotes particles 2 to 10. This directly reflects the neuron assembly. Note that a zoom facility was used, to include/exclude particles in order to accurately identify the members of each group.

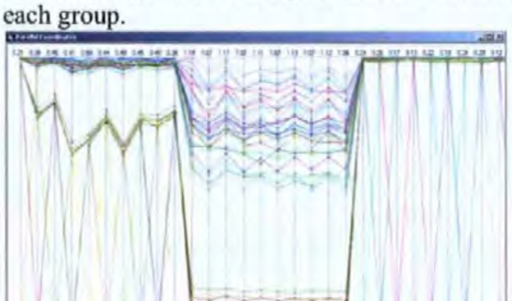


Figure 3.7 Snapshot 7500 of the animation showing all 50 parallel coordinates in 29 dimensions.

3.3 Future use of Parallel Coordinates

One of the most significant benefits of using parallel coordinates in comparison to distance graphs, is the fact that, evidence to date suggests, they are significantly less sensitive to the specification of aggregation, decay and increment parameters. Numerous trials were performed in which the data used to generate 'poor' distance graphs was represented in parallel coordinates. In general, this data was analysed successfully using parallel coordinates.

One of the main disadvantages of parallel coordinates is the impact that standard display equipment limitations has on their use. This was demonstrated in §3.2, where the assembly of 50 neurons, highlighted the fact that the maximum number of vertical axes, easily viewed in parallel coordinates, is approximately 30 (based on a standard monitor size). Future research will incorporate alternative projection facilities to overcome these limitations.

Recently, within the domain of "Information Visualisation" much research has focused on the development of parallel coordinates in order to analyse even greater quantities of data. An example of this is the concept of hierarchical parallel coordinates (Fua et al., 1990). Future work will investigate using hierarchical parallel coordinates for representation of larger data sets.

4 Dimension Reduction

In addition, to alternative methods of data representation, it is also possible to reduce the quantity of data represented whilst maximising the amount of information portrayed. PCA is one method of achieving this type of reduction.

PCA (Biswas et al., 1981; Gnanadesikan and Wilk, 1969), is used to find the "best" subspace for the projection of multi-dimensional data. This method achieves a higher degree of representation accuracy by maintaining as much of the overall data structure as possible. The PCA method used in these trials was achieved by analysing the covariance matrix of a selected time slice of the output from the gravity transform. A co-variance matrix depicts the position of each particle in each dimension at a chosen time t. Note that t is chosen to be a point after which useful aggregation has occurred. The eigenvalues and eigenvectors are derived using Householder's reduction and an Implicit QL algorithm, (Press et al., 1992; StatLib). Subsequently, the same eigenvectors are used to project each time slice of the output from the gravity transform.

4.1 PCA Example One

Using the assembly of 15 neurons given in Figure 4.1, spike train data was generated. This trial lasted 20s and neurons had a firing rate of approximately 0.3231 spike/s.

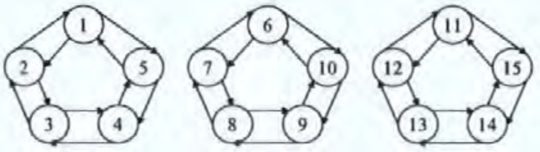


Figure 4.1 Specification of the connections between the 15 neurons used as input to the spike train generator.

Figure 4.2 shows this data using a standard raster plot representation.

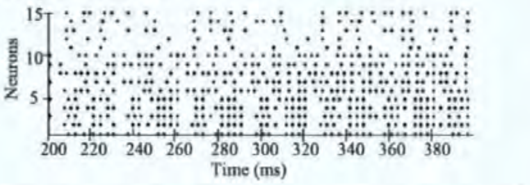


Figure 4.2 A raster plot depicting a portion of the 15 spike trains generated for the neuron assembly shown in Figure 4.1.

Subsequently, the data was input to the gravity transform where a=0.1, τ=0.2 and b=0.1. The resultant distance graph is shown Figure 4.3 where $s_1=\{d_{ij}$: i=1,...,10, j=2,...,10, $i<j\}$, $s_{2=}\{d_{ij}$: i=1,...,10, j=11,...,15$\}$ and $s_3 = \{d_{ij}$: i=11,...,15, j=12,...,15, $i<j\}$.

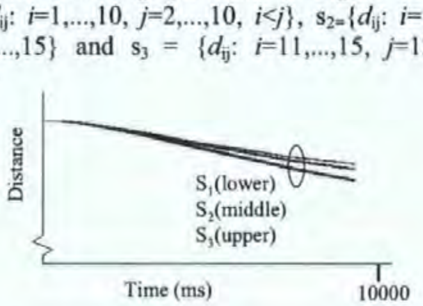


Figure 4.3 Distance graph of the results of the gravity transform where n=15, a=0.1, τ=0.2 and b=0.1.

The data set output from the gravity transform, shown in Figure 4.3 as a distance graph, was reduced to two dimensions using PCA. The output of this method is displayed in Figure 4.4.

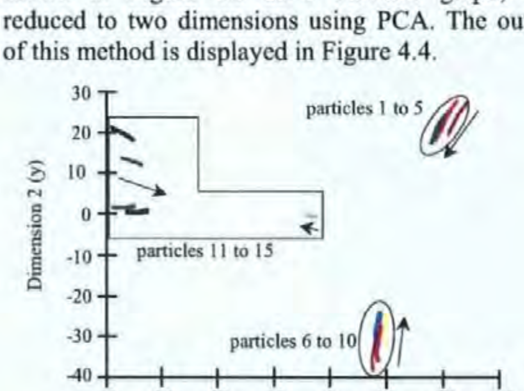


Figure 4.4 This plot depicts the trajectories of all 15 particles over time, after the output of the gravity transform was reduced from 15 dimensions to 2 dimensions using PCA.

Note that the trajectory of each particle is represented as a number of discrete points in space over time. In this representation, neurons 1 to 5 and neurons 6 to 10 can be seen as two distinct groups. In addition, note that neurons 11 to 15 are beginning to form a third group. Recall that only two of the original 15 projected dimensions are portrayed.

4.2 PCA Example Two

In Figure 4.5, an assembly of 20 neurons is shown. This assembly was used to generate spike train data for a trial lasting 20s where neurons had a firing rate of approximately 0.0438 spike/s.

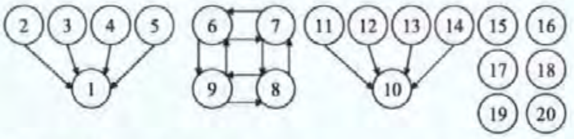


Figure 4.5 *Specification of the connections between the 20 neurons used as input to the spike train generator.*

Figure 4.6 shows the data set generated using a raster plot.

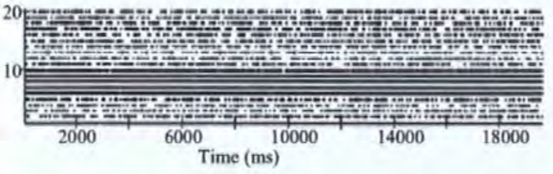


Figure 4.6 *A raster plot depicting a portion of the 20 spike trains generated for the neuron assembly shown in Figure 4.5.*

This data set was subsequently used as input to the gravity transform with a=0.5, τ=0.7and b=1. The distance graph shown in Figure 4.7 was produced. Note that $s_1=\{d_{ij}: i=6,...,9, j=7,...,9, i<j\}$, $s_2=\{d_{ij}: i=1, j=2,...,5\}$ and $s_3=\{d_{ij}: i=1, j=10,...,13\}$.

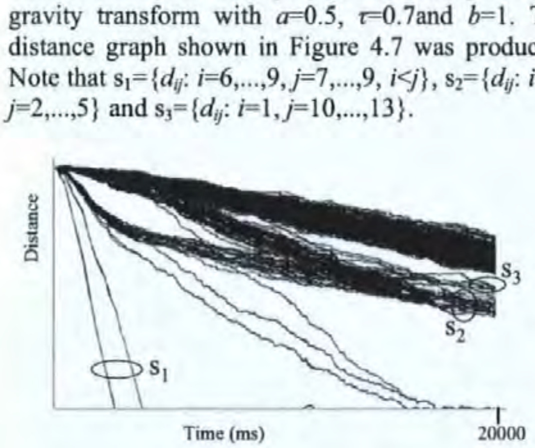


Figure 4.7 *Distance graph of the results of the gravity transform where n=20, a=0.5, τ=0.7 and b=1.*

In Figure 4.7, neurons 6 to 9 (s_1) are easily identified whilst neurons 1 to 5 (s_2) and 10 to 14 (s_3) are masked by the other distances pairs plots.

The data set output from the gravity transform, shown as a distance graph in Figure 4.7, was reduced to 2 dimensions using PCA. The output of this method is displayed in Figure 4.8.

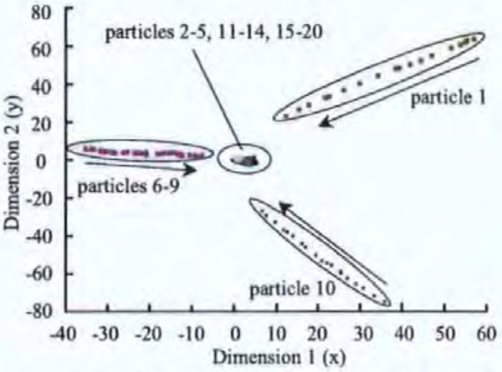


Figure 4.8 *This plot depicts the trajectories of all 20 particles over time, after the output of the gravity transform was reduced from 20 dimensions to 2 dimensions using PCA.*

The distance pairs relating to the solitary neurons cluster at the centre of the display, indicating that they have no tendency to cluster with other neurons. A zoom of this cluster is shown in Figure 4.9.

Figure 4.8 also shows another group, neurons 6 to 9 clearly distinct from the rest. However, it is difficult to draw any further conclusions from this plot. Note that the arrows indicate the general direction of the trajectory.

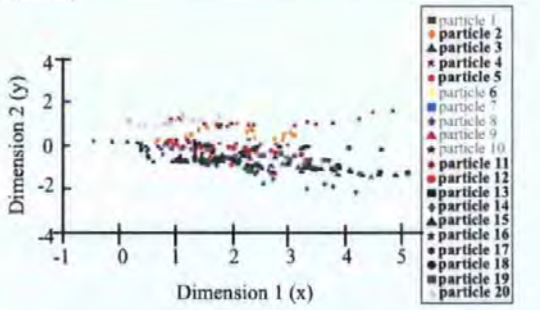


Figure 4.9 *This plot depicts the trajectories of particles 2-5 and 11-20, after the output of the gravity transform was reduced from 30 dimensions to 2 dimensions using PCA.*

4.3 Limitations of PCA

As demonstrated, dimension reduction can be used to convey the majority of the information contained in the original *n*-dimensional data set. Moreover, it can result in an improved representation of the data. However, a balance needs to be maintained between dimension reduction and loss of information. A comparison of Figure 4.7 and 4.8 highlights the loss of information that can result from dimension reduction. Note that the use of a non-linear reduction may ease this problem (Sammon (1969)).

In addition to the reduction of dimensions, investigating innovative methods of representing large quantities of data, such as parallel coordinates, can alleviate the problem of representing vast data sets.

5 Conclusions

This paper reports on the use of visualisation in the analysis of synchronous activity in multi-dimensional spike train data derived from the gravity transform algorithm. On the basis of our simulations we derived approximate values for the algorithm parameters: aggregation, charge and delay. Numerous trials were run using relatively large numbers of spike trains. These were successful whilst highlighting the limitations of using distance graphs to output the data.

Parallel coordinates and animation techniques were used to support the analysis of these vast data sets. In addition to these innovative methods of display, concentration of the data can also alleviate the problem of representing vast data sets.

PCA was used to reduce the quantity of data whilst maximising the quality of the data retained. This method is very useful in creating manageable data sets, yet limited due to display. Hence, future work will incorporate the use of parallel coordinates for the display of PCA output data in addition to output directly from the gravity transform.

In conclusion, no single method will overcome the problems of analysing synchronous activity in multi-dimensional data sets. However, the combination of many diverse methods from domains such as mathematics and statistics, information visualisation and graphics will provide a very practical platform on which to analyse this quantity of data.

6 Acknowledgements

This research is supported by the Engineering and Physical Sciences Research Council grant number GR/N04904. Additionally, the research of RB is supported in part by the Russian Foundation of Basic Research (grant 99-04-49112) and by EPSRC (grant GR/N63888/01).

7 References

Baker, S.N., Gerstein, G.L., 2000. Improvements to the Sensitivity of the Gravitational Clustering for Multiple Neuron Recordings. Neural Computation 12, 2597 - 2620.

Borisyuk, R.M., Borisyuk, G.N., 1997. Information coding on the basis of synchronisation of neuronal activity. BioSystems 40, 3-10.

Fries, P., Neuenschwander, S., Engel, A.K., Goebel, R., Singer, W., 2001. Rapid feature selective neuronal synchronization through correlated latency shifting. Nature Neuroscience 4(2), 194-200.

Fua, Y., Ward, M., Rundensteiner, E., 1999. Hierarchical parallel coordinates for exploration of large datasets. In Proceedings of Visualization'99, 58-64.

Biswas, G., Jain, A.K., Dubes, R.C., 1981. Evaluation of Projection Algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence 12(6), 701-708.

Gerstein, G.L., Aertsen, A.M., 1985. Representation of Cooperative Firing Activity Among Simultaneously Recorded Neurons. Journal of Neurophysiology 54(6), 1513-1528.

Gerstein, G.L., Perkel, D.H., Dayhoff, J.E., 1985. Cooperative Firing Activity in Simultaneously Recorded Populations of Neurons: Detection and Measurement. Journal of Neuroscience 5(4), 881-889.

Gnanadesikan, R., Wilk, M.B., 1969. Data Analytic Methods in Multivariate Statistical Analysis. Multivariate Analysis 2, 593-638.

Inselberg, A., Dimsdale, B., 1990. Parallel Coordinates: A tool for visualising multi-dimensional geometry. In Proceedings of Visualization'90, 361-378.

Mulab. http://mulab.physiol.upenn.edu/index.html

Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannerty, B.P., 1992. Numerical Recipes in C: The Art of Scientific Computing (2nd Edition). Cambridge University Press, New York.

Sammon Jr, J.W., 1969. A Nonlinear Mapping for Data Structure Analysis. IEEE Transactions on Computers C-18(5), 401-409.

StatLib. http://lib.stat.cmu.edu/multi/

Wegman, E.J., 1990. Hyper-dimensional Data Analysis Using Parallel Coordinates. Journal of the American Statistical Association 411(85), 664-675.

Interactive Poster: Visualisation of Neurophsyiological Data

E. Stuart, M. Walter and R. Borisyuk
Centre for Neural and Adaptive Systems, School of Computing,
University of Plymouth, Plymouth, Devon, UK, PL4 8AA
lstuart@plymouth.ac.uk
http://www.tech.plym.ac.uk/soc/research/neural/members.html#stuart

1. Introduction

Temporal coding has an important role in the debate on information encoding by spike trains. It establishes that information is encoded in the seemingly random patterns of spikes, even in the exact temporal arrangement of inter-spike intervals. Subsequently, one of basic principles that underlie information processing in the brain is the principle of synchronisation of neural activity [1] [2]. Vast quantities of experimental data and mathematical models indicate that the synchronisation principle may be useful in devising various systems of information processing.

The experimental evidence that is currently available requires analysis in order to extract inherent information. Analysis of multidimensional spike trains using standard tools such as cross-correlograms is increasingly complex due to the quantity of data involved. Hence, new methods of dealing with this data are needed. In 1985, one such analysis tool called the "Gravity Transformation" [3][4] was developed at the Multiple Unit Laboratory at Department of Neuroscience in the University of Pennsylvania [5].

It is based on the principle of gravitational interaction of particles where each neuron is represented by a particle and the movement of that particle is described in an *n*-dimensional space, where *n* is the number of neurons under investigation. All particles start equidistant from one another and the gravitational force (or charge) exerted by a particle is proportional to the spike train of the corresponding neuron. Each spike contributes charge and this charge decays exponentially over time. Thus, should two or more neurons spike coincidently, their corresponding particles will have an attractive force that causes the particle to move closer together. Let us suppose that two neurons have an above average synchrony of firing. Over time this would result in a strong attractive force between their corresponding particles. This would result in aggregation in *n*-dimensional space. Since, significant synchrony can indicate synaptic coupling [6] the aggregation of the particles can show the assemblies represented by the spike train data.

The gravity transformation has made a significant contribution to the field however there are some difficulties with the display of output data for large numbers of particles.

2. Representation of n-dimensional data

Parallel coordinates, originally pioneered in the 1980's, is a technique used to represent diverse sets of multidimensional data. In 1990, Inselberg [7] [8] renewed the use of parallel coordinates for the analysis of large quantities of multidimensional data and introduced some new representation features that have led to a marked increase in their utilization.

Inselberg's representation of parallel coordinates denotes data points as y-axis coordinate values distributed along the x-axis. In this scheme, a specific point in *n*-dimensional Euclidean space is represented by *n* y-axis values distributed along the x-axis. In the last decade much research has focused on the development of parallel coordinates in order to analyse even greater quantities of data. An example of this is the concept of hierarchical parallel coordinates [9].

3. Visualisation Software

It has been established that parallel coordinates can be used to identify correlations between variables and to convey aggregation information. Subsequently, this focuses on the application of parallel coordinates to the visualization of data produced by the gravity transformation in order to support the investigation of greater numbers of neurons. Naturally, the advantage of parallel axes over orthogonal axes is the fact that their limitations are based on the size of the display area available. Note that since there is no loss of data when using parallel coordinates that there is no "cost" to be considered for the gains achieved.

This paper presents a software analysis tool, VISA, used for the Visualization of Inter-Spike Associations that supports the analysis of multidimensional spike trains using both the gravity transformation and parallel coordinates. In additional to this, it provides additional functionality such as animation of the parallel coordinates display over time thus in this case depicting the aggregation of particles in the gravity transformation data. There is also the capability to view the display output in a static mode. Most significantly, the tool supports the display of any subset of particles for closer inspection.

Since the range of values represented by each parallel axis is dependent upon the group of particles viewed on that axis, this is effectively a zoom facility.

Currently, the parallel coordinates may be used for relatively larger values of *n* than the standard output display of the gravity transformation. Indeed, provided that scrolling windows are deemed acceptable to the user, there is no theoretical limit to the number of neurons that could be displayed in this manner. However, in practice, significant demands for interaction with the graphical user interface reduce the overall effectiveness of the software tool since user perception is a significant factor in the process. Note that the use of hierarchical parallel coordinates offers additional significant opportunities for future development of the VISA.

10. References

[1] Borisyuk RM & Borisyuk GN (1997), "Information coding on the basis of synchronisation of neuronal activity" BioSystem, 40, 3-10.

[2] Fries P, Neuenschwander S, Engel AK, Goebel R & Singer W (2001) "Rapid feature selective neuronal synchronization through correlated latency shifting" Nature Neuroscience, 4(2), 194-200.

[3] Gerstein GL & Aertsen AM (1985) "Representation of Cooperative Firing Activity Among Simultaneously Recorded Neurons" Journal of Neurophysiology, 54(6), 1513-1528.

[4] Gerstein GL et al (1985), "Cooperative Firing Activity in Simultaneously Recorded Populations of Neurons: Detection and Measurement" Journal of Neuroscience, 5(4), 881-889.

[5] http://mulab.physiol.upenn.edu/index.html

[6] Baker SN & Gerstein GL (2000) "Improvements to the Sensitivity of the Gravitational Clustering for Multiple Neuron Recordings" Neural Computation, 12, 2597 – 2620.

[7] Inselberg A & Dimsdale B (1990), "Parallel Coordinates: A tool for visualising multidimensional geometry" In Proceedings of Visualization'90, 361-378.

[8] Wegman EJ (1990), "Hyperdimensional Data Analysis Using Parallel Coordinates" Journal of the American Statistical Association, 411(85), 664-675.

[9] Fua Y, Ward M & Rundensteiner E (1999), "Hierarchical parallel coordinates for exploration of large datasets" In Proceedings of Visualization'99, 58-64.

A Compact Visualisation for Neurophysiological Data

M. Walter[*], L. Stuart[*] and R. Borisyuk[*][†]
[*]Centre for Neural and Adaptive Systems, School of Computing, University of Plymouth, Plymouth, Devon, UK
[†]Institute of Mathematical Problems in Biology, Russian Academy of Sciences, Pushchino, Moscow Region 142 290, Russia

Abstract

The current ability to record neural activity within the brains of mammals has led to the production of a large body of experimental data. The analysis and comprehension of this data is key to the understanding of many basic brains functions, for example learning and memory.

The main constituent of this data is multi-dimensional spike train recordings. As the analysis of these datasets, by traditional means, becomes more complex and time consuming the need for better methods of data analysis increases.

This paper presents an innovative method for analysis of the relationships within large multi-dimensional spike train datasets. This method, called the 'Correlation Grid,' is based on the Information Visualisation principles; overview the data, filter and zoom the data and obtain details-on-demand [1]. The features of the Correlation Grid are described, including filtering and statistical sorting methods.

1 Introduction

Explanations to many questions in the field of Neuroscience are dependent on the theoretical understanding of a large body of experimental neural data. Specifically, this understanding is fundamental to the exploration of information processing within the nervous system. A primary component of this data is simultaneously recorded multi-dimensional spike trains. Significant research in this area is steered towards the principle of synchronisation of neural activity [2][3].

Further, in-depth, analysis of the available experimental evidence is required in order to extract inherent information. The analysis of neural data, such as multi-dimensional spike trains, is increasingly complex using traditional tools, like cross-correlograms, due to the vast quantity of data involved. Consequently, new analysis methods are required to deal with this data.

The specific computer science field of Information Visualisation is focused on innovations in the representation of vast quantities of data. A guiding principle of Information Visualisation is that the investigator should have control over the data representation that they are using. Moreover, the investigator should be able to manipulate the data by applying relevant techniques, in order to steer that direction of the analysis. For example, it may be appropriate to use statistical or other mathematical routines to sort and organise the data.

The "information–seeking mantra", introduced by Shneiderman [1] in 1996, highlighted user requirements in this area. It proposed that users should have the ability to overview data, zoom and filter this data and to obtain details-on-demand. This mantra was widely adopted throughout the Information Visualisation community as a basis for defining user requirements. Frequently different levels of detail are viewed using different visualizations. Resulting in a number of different views of the data. For consistency, these multiple views should be linked [4][5][6].

In this paper, a method of dealing with the analysis of relatively large numbers of spike trains, involving cross-correlation, is proposed.

2 Neurophysiological data

Within the mammalian nervous system, there are many different types of neurons, each of which performs a different task. . These neurons communicate via small electrical impulses.

A neuron accumulates electrical charge from other neurons attached to it. When this accumulation reaches an internal threshold, the neuron will initiate an action potential. When a neuron initiates action potentials over time, it is said that the neuron is firing. For example, the application of pressure to the skin causes pressure-sensitive neurons to fire.

Note that action potentials are often referred to as spikes and a series of the spikes, over time, is called a spike train.

2.1 Spike trains

Spike train data is the primary data recorded during experimental Neurophysiology. This data is a record of the spiking activity of a collection of neurons under investigation. Figure 2-1 shows a section, from 200ms to

700ms, of a typical spike train recording for three neurons. In this figure, a horizontal plot represents the spike train of each neuron. Each horizontal plot denotes the occurrence of spikes, at specific times, by a vertical line.

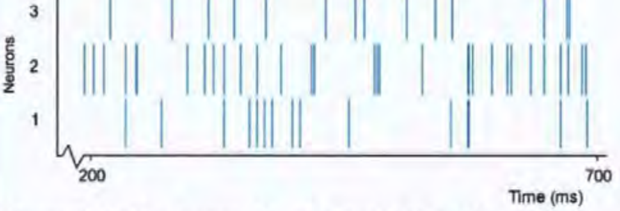


Figure 2-1: An example of a typical spike train recording for three neurons over a period of 500ms.

It has been established that information is encoded in this data. However, it has also been shown that spikes from a single neuron are identical [7]. Thus, the form of individual spikes is believed to carry little information. Instead, it is the spiking frequency and thus, inter-spike-intervals that carry information. Moreover, the synchronisation of spikes within this data is of prime interest. Research [2] has shown that spike synchrony is key to information processing within the brain.

2.2 Coupling

Connections between neurons fall into to general groups; these are *Direct Synaptic* coupling and *Common Input* coupling. In both cases the firing patterns of the neurons will be synchronised.

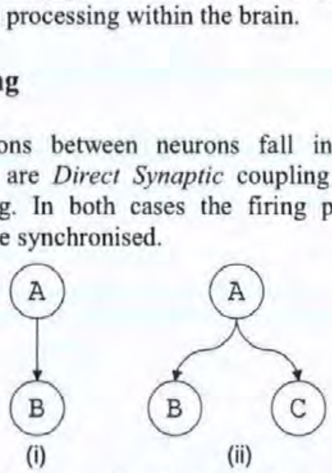


Figure 2-2: An example of (i) direct synaptic coupling and (ii) common input coupling

Figure 2-2 (i) illustrates Direct Synaptic coupling, where neuron A is coupled so that it stimulates neuron B. If neuron A fires, then neuron B has an increased probability of firing. Figure 2-2 (ii) illustrates Common Input coupling, where neuron A stimulates both neurons B and C, resulting in the correlation of their input. Thus, if neuron A fires then both neurons B and C have increased probabilities of firing.

2.3 Multi-dimensional spike train data

Explanations to many questions in the field of Neuroscience are dependent on the comprehension of large sets of experimental data. For example, many fundamental brain processes, such as memory, learning and attention, are still unclear in many respects. One of the central theories explaining how the brain processes information is the synchronisation of neural activity. Research [2] suggests that the synchronisation principle may be useful in describing various brain systems.

Much experimental data has been recorded, from the mammalian brain. The majority of this data is in the form of multi-dimensional spike train recordings. Investigation of this data centres on the synchronisation of spikes between spike trains, to derive the coupling of the underlying neurons.

Neurophysiologists still utilise traditional analysis methods due to the absence of more substantial software support. However, due to the quantity of data involved these methods are both time consuming and complex.

3 Traditional methods of analysis

Currently there are a variety of methods to analyse multi-dimensional spike train data. The *cross-correlogram* is one of the most common and measures the temporal correlation of one spike train with another. Like the cross-correlogram, most current methods are designed for use with only a pair of neurons, and these methods do not scale-up to deal with larger assemblies.

3.1 Pair-wise correlation

The cross-correlogram [8][9] is used to measure the synchronisation between the spike trains of two neurons. One spike train is designated to be the reference train. The other is known as the target. A time frame for correlation must be specified, this is to account for the inherent delay in neural circuits. The time frame, or correlation window, consists of a number of equal time segments, called bins.

The correlation window is centred over the first spike of the reference train. The number of target train spikes that fall within each bin is calculated. This is repeated for each subsequent spike in the reference train.

The result of each comparison is added to the previous to give the overall correlation; an example of this is illustrated in Figure 3-1.

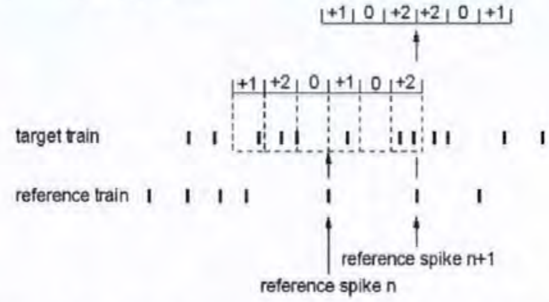


Figure 3-1 An example of a Cross Correlogram calculation using a six-bin window

The overall correlation is then plotted using a cross-correlogram.

3.2 The cross-correlogram

The resultant cross-correlation, plotted as the cross-correlogram, shows the correlation of the target train with respect to the reference train. This is illustrated in, Figure 3-2, for two spike trains recorded from neurons coupled as shown in Figure 2-2(i).

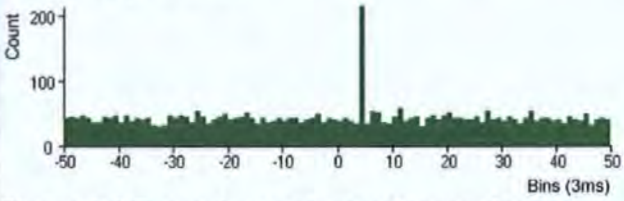


Figure 3-2 Example cross-correlogram for two connected neurons

Should a significant peak [10], as shown in Figure 3-2, be evident in this plot a correlation exists between the two trains. In this case the two neurons are likely to be connected. In contrast when no peaks are evident from the cross-correlogram, see Figure 3-3, it is likely that the neurons are not connected.

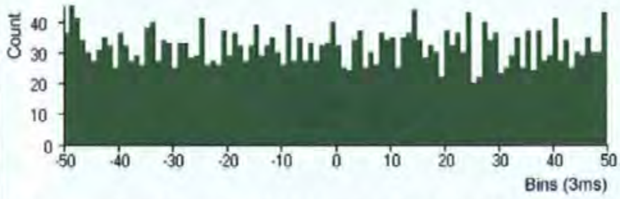


Figure 3-3 Example cross-correlogram for two un-connected neurons

It takes one cross-correlogram to represent the relationship between the spike trains of two neurons. Thus, for any significantly sized neuronal assembly, numerous cross-correlograms would be required to analyse the connectivity of the assembly. New problems are posed by this large quantity of data. This is because all pair-wise results must be analysed to understand any relationships between the underlying neurons.

3.3 Peak significance

It is possible for a peak in the output of a cross-correlogram to be 'false'. These false peaks are random occurrences and due to the high spike frequency of the train. A number of methods have been proposed to deal with this problem including the Brillinger normalisation and confidence interval [10].

The Brillinger method normalises the results of the cross-correlogram, giving each bin, in the output, a mean value of one. This normalisation supports the comparison of cross-correlograms with different train lengths, as the quantity of spikes on each train is taken into account. In addition, a confidence interval, based on the data, is calculated. Any peaks that are within this interval are defined to be random. In contrast, peaks greater than this interval are considered to show significant correlation.

Figure 3-4 illustrates a Brillinger normalised cross-correlogram, for a pair of common input neurons. Note the majority of bins have values around one, falling within the Brillinger confidence interval. Thus, these bins do not show statistically significant correlation. In contrast, the central peak is much greater than the confidence interval, showing a significant correlation between the input spike trains. Thus indicating a high likelihood of the neuronal coupling.

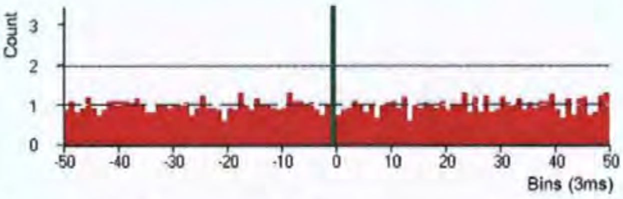


Figure 3-4 Example Brillinger normalised cross-correlogram with confidence interval

4 Current methods of analysis

A number of innovative methods exist for the analysis of spike trains from large groups of interconnected neurons, called *assemblies*. The 'Gravity Transform', originally developed by Gerstein and Aertsen [11][12][13] is one notable method. The Gravity Transform algorithm can be used to study the firing dependencies of multi-dimensional spike trains. Recent work by Stuart et al. [14] has enhanced the output of the original Gravity Transform algorithm using visualisation techniques including parallel coordinates to support the display of the Gravity Transform output.

However, a great deal work is still necessary in this area in order to fully support exploration of large multi-dimensional data sets. For this reason, the spike train 'Correlation Grid' has been developed.

5 Correlation grid

The Correlation Gird presents users with an overview of cross-correlogram results, for a number of spike trains.

For a given dataset, of *n* spike trains, all unique cross-correlograms are generated, where the user specifies the bin and window size. Subsequently, the cross-correlograms are normalised using the Brillinger method. Finally, the results of these cross-correlograms are displayed as an *n*-by-*n* grid of grey scale cells, representing the individual correlations between all pairs of spike trains.

The grid encodes the 'height' of the largest peak in each cross-correlogram. The peaks are in coded from white, representing no peak, to black, representing the largest peak in the grid.

The user can select whether to view 'all peaks' or just significant peaks. Significant peaks are those that fall outside of the Brillinger confidence interval specified for the grid. In addition, the individual cross-correlograms can be view by selecting the corresponding cell in the grid.

To demonstrate the Correlation Grid, a dataset of ten spike trains was generated, over 2000ms, for the assembly of neurons shown in Figure 5-1.

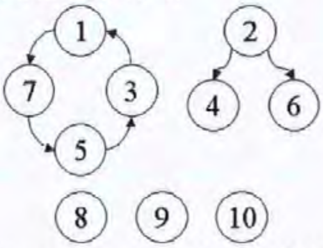


Figure 5-1 Neuron assembly for test data set

In this assembly, the spike trains of neurons one, three, five and seven will be correlated, as their corresponding neurons are connected. Like wise the spike trains of neurons two, four and six will correlate. In contrast, the spike trains of neurons eight, nine and ten will not correlate with any others as they are unconnected, thus have a random firing pattern.

A Correlation Grid for this dataset was generated, with a bin size of 2ms and window size of 100. The resultant grid, showing al peaks, is shown in Figure 5-2.

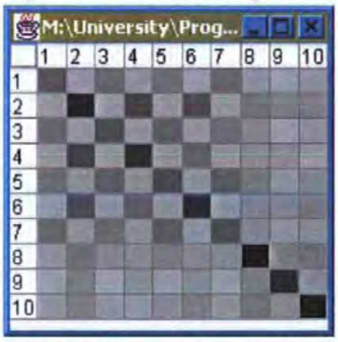


Figure 5-2 Example Correlation Grid showing all peaks (bin size 2ms, window size 100)

On close examination of this Grid, it is possible to deduce that spike trains one, three, five and seven are correlated. Note the higher peaks (represented by darker greys) between these columns/rows. Likewise, it is possible to conclude that a relationship exists between the spike trains of neurons two, four and six.

The clarity of these relationships is greatly improved by filtering the grid so it only containing significant peaks. This filtered grid is shown in Figure 5-3. Recall the correlation between spike trains one, three, five and seven which is more prevalent in this filter Grid. Like wise the correlation between train two, four and six.

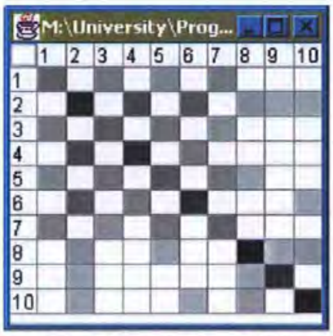


Figure 5-3 Example Correlation Grid showing only significant peaks (bin size 2ms, window size 100)

With the removal of the non-significant peaks the grid is less cluttered. It is now easier to identify the correlation between the connected neurons, shown in Figure 5-1.

Moreover the lack of correlation between spike trains is more apparent. From an examination of Figure 5-3 it is possible to see that trains eight, nine and ten have minimal, or no, correlation with other trains.

5.1 Cluster analysis

The identification of groups, or clusters, of correlations is key to understanding the relationships between the underlying neurons. It is possible to identify these clusters visually, however this is not easily expanded for problems with larger datasets.

To aid with the identification of correlation clusters, a statistical cluster analysis method has been implemented. This method uses the height of the most significant peak, if any exist, of each cross-correlogram to build a dendrograph for the correlation gird. This in turn is used to generate the initial display order of the spike trains.

The effect of clustering is shown in Figure 5-4. This grid was generated from the same data as used to create the grid in Figure 5-3, based on the neural assembly in Figure 5-1. Cluster analysis was performed on the grid, to determine the optimal sequence of spike trains in the grid. This analysis was performed using significant peaks only.

With the aid of filtering and clustering it is possible to clearly identify the groups in the neuronal assembly shown in Figure 5-1.

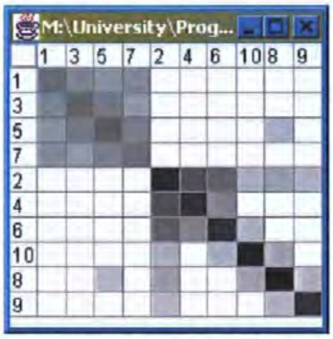


Figure 5-4 Example Correlation Grid, showing only significant peaks and clustered (bin size 2ms, window size 100)

From Figure 5-4 it is possible to identify the correlations in the neuron 'ring'; neurons one, three, five and seven. They are shown in the top left portion of the grid. Additionally, the common input group; neurons two, four and six; are also grouped together in the centre of the diagram. Finally, from the same figure, the independent neurons, eight, nine and ten, are all in the lower right portion of the grid.

From closer examination of the cross-correlograms of the final cluster, of unconnected neurons, it is apparent that the peaks are relatively small. See Figure 5-6 (i), which shows the cross-correlation for neurons two and ten.

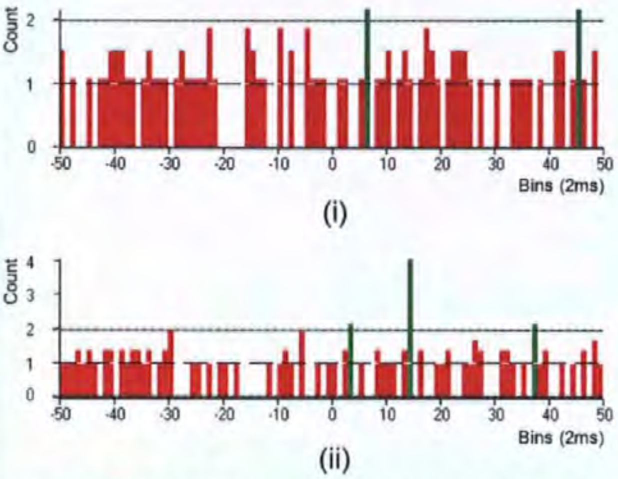


Figure 5-5 Cross-correlograms for (i) neurons two and ten (ii) neurons two and four from the previous Grid

From the display it is possible to see that the peaks are relatively small, only slightly greater than the confidence interval. In contrast the correlations of connected neurons, shown in Figure 5-6 (ii) for neurons two and four, show relatively large peaks. Thus, for neurons eight, nine and ten, it would be possible to infer that these peaks do not represent true correlation between the spike trains. Hence, these neurons are not coupled to any others in the assembly.

This overview of cross-correlograms facilitates improved fine-tuning of the correlation parameters. This fine-tuning is commonly needed to obtain the highest clarity of result.

Increasing the bin size, for example, means more spikes are taken into consideration. Thus, a peak must be higher to be classed as significant. This principle is illustrated in Figure 5-6 where the bin size for the grid has been changed to 3ms. The same dataset and previous window size were used to create this grid.

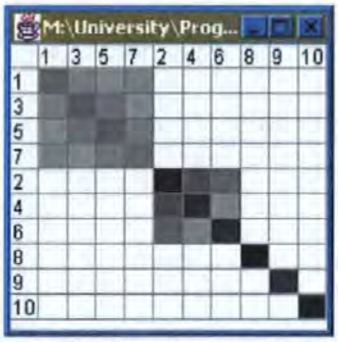


Figure 5-6 Example Correlation Grid showing only significant peaks (bin size 3ms, window size 100)

The grid in Figure 5-6 has been filtered to solely show significant peaks and clusters. Note that correlation only exists between the spike trains of the coupled neurons. For example, spike trains eight, nine and ten show no correlation with any other spike trains.

6 Future work

The work presented in this paper is part of an Information Visualisation project, at the Centre for Neural and Adaptive Systems, called Visualisation of Inter-Spike Associations (VISA) [15]. Specifically, this paper has presented work on an innovative method for analysis of large neural assemblies using cross-correlation. However, a number of areas still require attention.

Currently, users are unable to alter the order of the spike trains. The ability to reorder the gird and to fine-tune the display will be implemented.

Empirical testing is underway to evaluate the usability of this visualization method. Subsequently, it is likely that this tool will be integrated into the VISA Toolbox [15].

7 Acknowledgments

This research is supported by the Engineering and Physical Sciences Research Council (EPSRC) grant number GR/N04904. Additionally, the research of Roman Borisyuk is supported in part by the Russian Foundation of Basic Research (grant 99-04-49112) and by EPSRC (grant GR/N63888/01).

8 References

[1] Shneiderman B, 1996. The eyes have it: A task by data type taxonomy for information visualizations. Proc. IEEE Symposium Visual Language, VL, pages 336-343, 3-6.

[2] Borisyuk R.M., Borisyuk G.N., 1997. Information coding on the basis of synchronisation of neuronal activity. BioSystems 40, 3-10

[3] Fries P., Neuenschwander S., Engel A.K., Goebel R. and Singer W., 2001. Rapid feature selective neuronal synchronization through correlated latency shifting. Nature Neuroscience 4(2), 194-200.

[4] North C, Multiple views and tight coupling in visualization: A language, taxonomy, and system. Proc. CSREA CISST 2001 Workshop of fundamental Issues in Visualization, pages 626-632

[5] Baldonado M. Q. W, Woodruff A., and Kuchinsky A 2000. Guidelines for using multiple views in information visualization. Advanced Visual Interfaces, pages 110-119.

[6] Roberts J. C. 1998. On encouraging multiple views for visualization. IV'98 – Proceedings International Conference on Information Visualization, pages 8-14 IEEE Computer Society, July 1998

[7] Robinson D. editor 1998. Neurobiology. Springer, The Open University. ISBN: 3-540-63546-7

[8] Awiszus F 1997. Spike train analysis. Journal of Neuroscience Methods, 74.

[9] MuLab http://mulab.physiol.upenn.edu/index.html

[10] Brillinger D. R. 1979. Confidence intervals for the crosscovariance function. Selecta Statistica Canadiana, V, pages 1-16.

[11] Gerstein, G.L., Aertsen, A.M., 1985. Representation of Cooperative Firing Activity Among Simultaneously Recorded Neurons. Journal of Neurophysiology 54(6), 1513-1528

[12] Gerstein, G.L., Perkel, D.H., Dayhoff, J.E., 1985. Cooperative Firing Activity in Simultaneously Recorded Populations of Neurons: Detection and Measurement. Journal of Neuroscience 5(4), 881-889.

[13] Baker, S.N., Gerstein, G.L., 2000. Improvements to the Sensitivity of the Gravitational Clustering for Multiple Neuron Recordings. Neural Computation 12, 2597 - 2620.

[14] Stuart L., Walter M. and Borisyuk R. 2002. Visualization of synchronous firing in multi-dimensional spike trains. BioSystem 67:265-279

[15] Visualization of Inter-spike Association (VISA) project website http://www.tech.plym.ac.uk/soc/research/neural/staff/lstuart/VISA/index.html

Spike Train Correlation Visualization

Martin A. Walter* mwalter@plymouth.ac.uk
Liz J. Stuart* lstuart@plymouth.ac.uk
Roman Borisyuk*† rborisyuk@plymouth.ac.uk

*Centre for Neural and Adaptive Systems, School of Computing, University of Plymouth, Plymouth, Devon, UK
†Institute of Mathematical Problems in Biology, Russian Academy of Sciences, Pushchino, Moscow Region 142 290, Russia

Abstract

The current ability to record neural activity within the brains of mammals has led to the production of a large body of experimental data. The analysis and comprehension of this data is key to the understanding of many basic brains functions, for example learning and memory.

The main constituent of this data is multi-dimensional spike train recordings. As the analysis of these datasets, by traditional means, becomes more complex and time consuming the need for better methods of data analysis increases.

This paper presents an innovative method for analysis of the relationships within large multi-dimensional spike train datasets. This method, called the 'Correlation Grid,' is based on the Information Visualisation principles; overview the data, filter and zoom the data and obtain details-on-demand [1]. The features of the Correlation Grid are described, including filtering and statistical sorting methods.

1 Introduction

Explanations to many questions in the field of Neuroscience are dependent on the theoretical understanding of a large body of experimental neural data. Specifically, this understanding is fundamental to the exploration of information processing within the nervous system. A primary component of this data is simultaneously recorded multi-dimensional spike trains. Significant research in this area is steered towards the principle of synchronisation of neural activity [2][3].

Further, in-depth, analysis of the available experimental evidence is required in order to extract inherent information. The analysis of neural data, such as multi-dimensional spike trains, is increasingly complex using traditional tools, like cross-correlograms, due to the vast quantity of data involved. Consequently, new analysis methods are required to deal with this data.

The specific computer science field of Information Visualisation is focused on innovations in the representation of vast quantities of data. A guiding principle of Information Visualisation is that the investigator should have control over the data representation that they are using. Moreover, the investigator should be able to manipulate the data by applying relevant techniques, in order to steer that direction of the analysis. For example, it may be appropriate to use statistical or other mathematical routines to sort and organise the data.

The "information-seeking mantra", introduced by Shneiderman [1] in 1996, highlighted user requirements in this area. It proposed that users should have the ability to overview data, zoom and filter this data and to obtain details-on-demand. This mantra was widely adopted throughout the Information Visualisation community as a basis for defining user requirements. Frequently different levels of detail are viewed using different visualizations. Resulting in a number of different views of the data. For consistency, these multiple views should be linked [4][5][6].

In this paper, a method of dealing with the analysis of relatively large numbers of spike trains, involving cross-correlation, is proposed.

2 Neurophysiological data

Within the mammalian nervous system, there are many different types of neurons, each of which performs a different task. . These neurons communicate via small electrical impulses.

A neuron accumulates electrical charge from other neurons attached to it. When this accumulation reaches an internal threshold, the neuron will initiate an action potential. When a neuron initiates action potentials over time, it is said that the neuron is firing. For example, the application of pressure to the skin causes pressure-sensitive neurons to fire.

Note that action potentials are often referred to as spikes and a series of the spikes, over time, is called a spike train.

2.1 Spike trains

Spike train data is the primary data recorded during experimental Neurophysiology. This data is a record of the spiking activity of a collection of neurons under investigation. Figure 2-1 shows a section, from 200ms to 700ms, of a typical spike train recording for three neurons. In this figure, a horizontal plot represents the spike train of each neuron. Each horizontal plot denotes the occurrence of spikes, at specific times, by a vertical line.

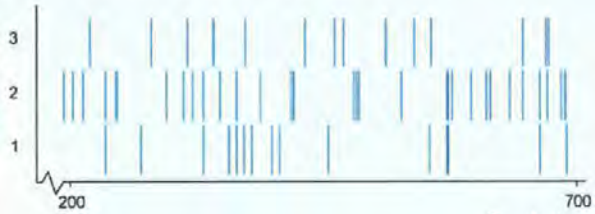


Figure 2-1: An example of a typical spike train recording for three neurons over a period of 500ms.

It has been established that information is encoded in this data. However, it has also been shown that spikes from a single neuron are identical [7]. Thus, the form of individual spikes is believed to carry little information. Instead, it is the spiking frequency and thus, inter-spike-intervals that carry information. Moreover, the synchronisation of spikes within this data is of prime interest. Research [2] has shown that spike synchrony is key to information processing within the brain.

2.2 Coupling

Connections between neurons fall into to general groups; these are *Direct Synaptic* coupling and *Common Input* coupling. In both cases the firing patterns of the neurons will be synchronised.

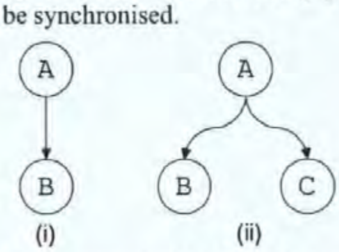


Figure 2-2: An example of (i) direct synaptic coupling and (ii) common input coupling

Figure 2-2 (i) illustrates Direct Synaptic coupling, where neuron A is coupled so that it stimulates neuron B. If neuron A fires, then neuron B has an increased probability of firing. Figure 2-2 (ii) illustrates Common Input coupling, where neuron A stimulates both neurons B and C, resulting in the correlation of their input. Thus, if neuron A fires then both neurons B and C have increased probabilities of firing.

2.3 Multi-dimensional spike train data

Explanations to many questions in the field of Neuroscience are dependent on the comprehension of large sets of experimental data. For example, many fundamental brain processes, such as memory, learning and attention, are still unclear in many respects. One of the central theories explaining how the brain processes information is the synchronisation of neural activity. Research [2] suggests that the synchronisation principle may be useful in describing various brain systems.

Much experimental data has been recorded, from the mammalian brain. The majority of this data is in the form of multi-dimensional spike train recordings. Investigation of this data centres on the synchronisation of spikes between spike trains, to derive the coupling of the underlying neurons.

Neurophysiologists still utilise traditional analysis methods due to the absence of more substantial software support. However, due to the quantity of data involved these methods are both time consuming and complex.

3 Traditional methods of analysis

Currently there are a variety of methods to analyse multi-dimensional spike train data. The *cross-correlogram* is one of the most common and measures the temporal correlation of one spike train with another. Like the cross-correlogram, most current methods are designed for use with only a pair of neurons, and these methods do not scale-up to deal with larger assemblies.

3.1 Pair-wise correlation

The cross-correlogram [8][9] is used to measure the synchronisation between the spike trains of two neurons. One spike train is designated to be the reference train. The other is known as the target. A time frame for correlation must be specified, this is to account for the inherent delay in neural circuits. The time frame, or correlation window, consists of a number of equal time segments, called bins.

The correlation window is centred over the first spike of the reference train. The number of target train spikes that fall within each bin is calculated. This is repeated for each subsequent spike in the reference train.

The result of each comparison is added to the previous to give the overall correlation; an example of this is illustrated in Figure 3-1.

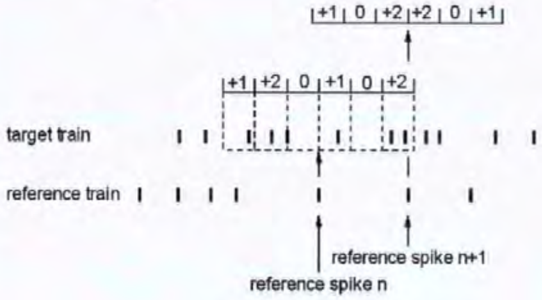


Figure 3-1 An example of a Cross Correlogram calculation using a six-bin window

The overall correlation is then plotted using a cross-correlogram.

3.2 The cross-correlogram

The resultant cross-correlation, plotted as the cross-correlogram, shows the correlation of the target train with respect to the reference train. This is illustrated in, Figure 3-2, for two spike trains recorded from neurons coupled as shown in Figure 2-2(i).

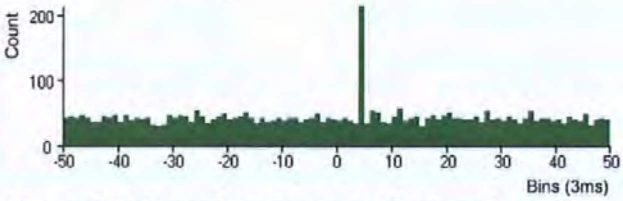


Figure 3-2 Example cross-correlogram for two connected neurons

Should a significant peak [10], as shown in Figure 3-2, be evident in this plot a correlation exists between the two trains. In this case the two neurons are likely to be connected. In contrast when no peaks are evident from the cross-correlogram, see Figure 3-3, it is likely that the neurons are not connected.

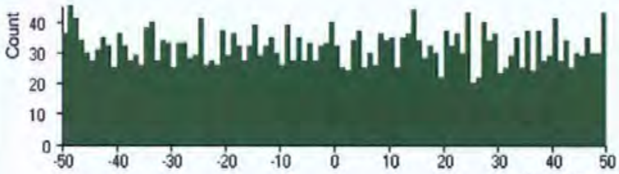


Figure 3-3 Example cross-correlogram for two un-connected neurons

It takes one cross-correlogram to represent the relationship between the spike trains of two neurons. Thus, for any significantly sized neuronal assembly, numerous cross-correlograms would be required to analyse the connectivity of the assembly. New problems are posed by this large quantity of data. This is because all pair-wise results must be analysed to understand any relationships between the underlying neurons.

3.3 Peak significance

It is possible for a peak in the output of a cross-correlogram to be 'false'. These false peaks are random occurrences and due to the high spike frequency of the train.

A number of methods have been proposed to deal with this problem including the Brillinger normalisation and confidence interval [10].

The Brillinger method normalises the results of the cross-correlogram, giving each bin, in the output, a mean value of one. This normalisation supports the comparison of cross-correlograms with different train lengths, as the quantity of spikes on each train is taken into account. In addition, a confidence interval, based on the data, is calculated. Any peaks that are within this interval are defined to be random. In contrast, peaks greater than this interval are considered to show significant correlation.

Figure 3-4 illustrates a Brillinger normalised cross-correlogram, for a pair of common input neurons. Note the majority of bins have values around one, falling within the Brillinger confidence interval. Thus, these bins do not show statistically significant correlation. In contrast, the central peak is much greater than the confidence interval, showing a significant correlation between the input spike trains. Thus indicating a high likelihood of the neuronal coupling.

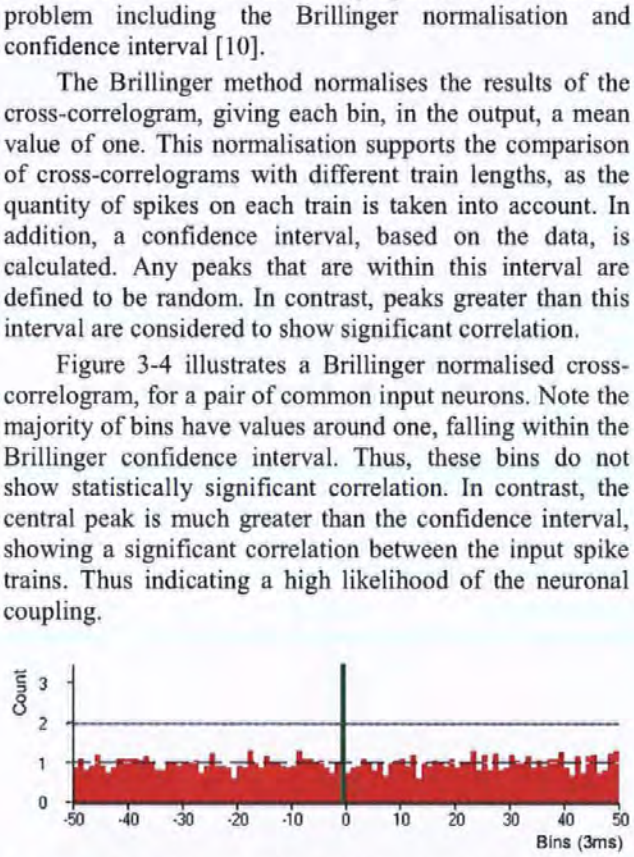


Figure 3-4 Example Brillinger normalised cross-correlogram with confidence interval

4 Current methods of analysis

A number of innovative methods exist for the analysis of spike trains from large groups of interconnected neurons, called *assemblies*. The 'Gravity Transform', originally developed by Gerstein and Aertsen [11][12][13] is one notable method. The Gravity Transform algorithm can be used to study the firing dependencies of multi-dimensional spike trains. Recent work by Stuart et al. [14] has enhanced the output of the original Gravity Transform algorithm using visualisation techniques including parallel coordinates to support the display of the Gravity Transform output.

However, a great deal work is still necessary in this area in order to fully support exploration of large multi-dimensional data sets. For this reason, the spike train 'Correlation Grid' has been developed.

5 Correlation grid

The Correlation Gird presents users with an overview of cross-correlogram results, for a number of spike trains.

For a given dataset, of *n* spike trains, all unique cross-correlograms are generated, where the user specifies the bin and window size. Subsequently, the cross-correlograms are normalised using the Brillinger method. Finally, the results of these cross-correlograms are displayed as an *n*-by-*n* grid of grey scale cells, representing the individual correlations between all pairs of spike trains.

The grid encodes the 'height' of the largest peak in each cross-correlogram. The peaks are in coded from white, representing no peak, to black, representing the largest peak in the grid.

The user can select whether to view 'all peaks' or just significant peaks. Significant peaks are those that fall outside of the Brillinger confidence interval specified for the grid. In addition, the individual cross-correlograms can be view by selecting the corresponding cell in the grid.

To demonstrate the Correlation Grid, a dataset of ten spike trains was generated, over 2000ms, for the assembly of neurons shown in Figure 5-1.

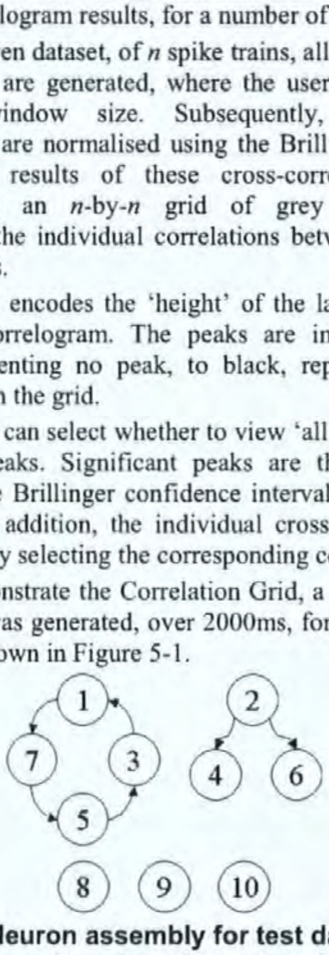


Figure 5-1 Neuron assembly for test data set

In this assembly, the spike trains of neurons one, three, five and seven will be correlated, as their corresponding neurons are connected. Like wise the spike trains of neurons two, four and six will correlate. In contrast, the spike trains of neurons eight, nine and ten will not correlate with any others as they are unconnected, thus have a random firing pattern.

A Correlation Grid for this dataset was generated, with a bin size of 2ms and window size of 100. The resultant grid, showing al peaks, is shown in Figure 5-2.

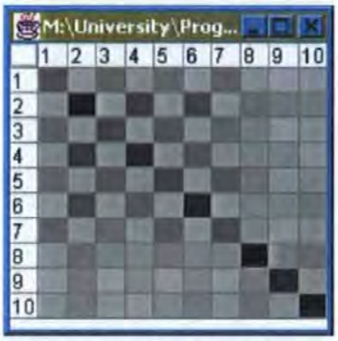


Figure 5-2 Example Correlation Grid showing all peaks (bin size 2ms, window size 100)

On close examination of this Grid, it is possible to deduce that spike trains one, three, five and seven are correlated. Note the higher peaks (represented by darker greys) between these columns/rows. Likewise, it is possible to conclude that a relationship exists between the spike trains of neurons two, four and six.

The clarity of these relationships is greatly improved by filtering the grid so it only containing significant peaks. This filtered grid is shown in Figure 5-3. Recall the correlation between spike trains one, three, five and seven which is more prevalent in this filter Grid. Like wise the correlation between train two, four and six.

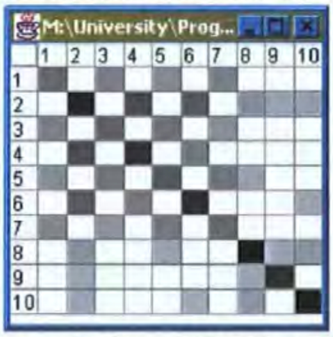


Figure 5-3 Example Correlation Grid showing only significant peaks (bin size 2ms, window size 100)

With the removal of the non-significant peaks the grid is less cluttered. It is now easier to identify the correlation between the connected neurons, shown in Figure 5-1.

Moreover the lack of correlation between spike trains is more apparent. From an examination of Figure 5-3 it is possible to see that trains eight, nine and ten have minimal, or no, correlation with other trains.

5.1 Cluster analysis

The identification of groups, or clusters, of correlations is key to understanding the relationships between the underlying neurons. It is possible to identify these clusters visually, however this is not easily expanded for problems with larger datasets.

To aid with the identification of correlation clusters, a statistical cluster analysis method has been implemented. This method uses the height of the most significant peak, if any exist, of each cross-correlogram to build a dendrograph for the correlation gird. This in turn is used to generate the initial display order of the spike trains.

The effect of clustering is shown in Figure 5-4. This grid was generated from the same data as used to create the grid in Figure 5-3, based on the neural assembly in Figure 5-1. Cluster analysis was performed on the grid, to determine the optimal sequence of spike trains in the grid. This analysis was performed using significant peaks only.

With the aid of filtering and clustering it is possible to clearly identify the groups in the neuronal assembly shown in Figure 5-1.

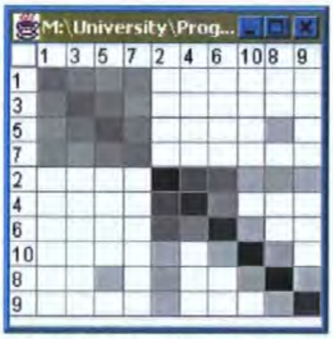


Figure 5-4 Example Correlation Grid, showing only significant peaks and clustered (bin size 2ms, window size 100)

From Figure 5-4 it is possible to identify the correlations in the neuron 'ring'; neurons one, three, five and seven. They are shown in the top left portion of the grid. Additionally, the common input group; neurons two, four and six; are also grouped together in the centre of the diagram. Finally, from the same figure, the independent neurons, eight, nine and ten, are all in the lower right portion of the grid.

From closer examination of the cross-correlograms of the final cluster, of unconnected neurons, it is apparent that the peaks are relatively small. See Figure 5-6 (i), which shows the cross-correlation for neurons two and ten.

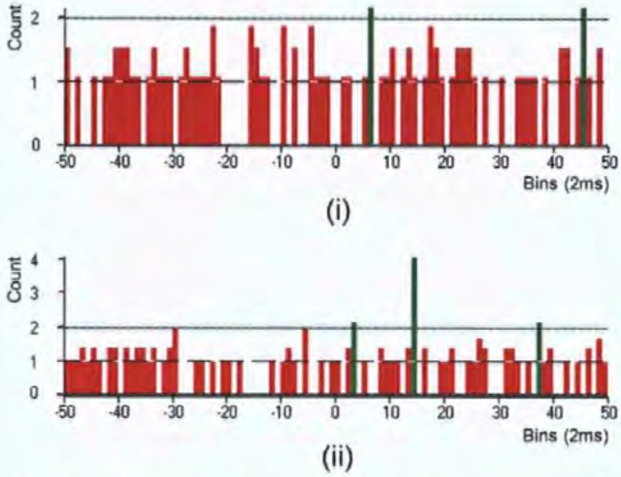


Figure 5-5 Cross-correlograms for (i) neurons two and ten (ii) neurons two and four from the previous Grid

From the display it is possible to see that the peaks are relatively small, only slightly greater than the confidence interval. In contrast the correlations of connected neurons, shown in Figure 5-6 (ii) for neurons two and four, show relatively large peaks. Thus, for neurons eight, nine and ten, it would be possible to infer that these peaks do not represent true correlation between the spike trains. Hence, these neurons are not coupled to any others in the assembly.

This overview of cross-correlograms facilitates improved fine-tuning of the correlation parameters. This fine-tuning is commonly needed to obtain the highest clarity of result.

Increasing the bin size, for example, means more spikes are taken into consideration. Thus, a peak must be higher to be classed as significant. This principle is illustrated in Figure 5-6 where the bin size for the grid has been changed to 3ms. The same dataset and previous window size were used to create this grid.

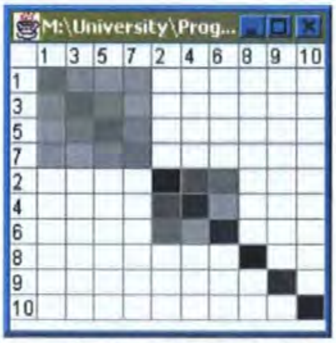


Figure 5-6 Example Correlation Grid showing only significant peaks (bin size 3ms, window size 100)

The grid in Figure 5-6 has been filtered to solely show significant peaks and clusters. Note that correlation only exists between the spike trains of the coupled neurons. For example, spike trains eight, nine and ten show no correlation with any other spike trains.

6 Future work

The work presented in this paper is part of an Information Visualisation project, at the Centre for Neural and Adaptive Systems, called Visualisation of Inter-Spike Associations (VISA) [15]. Specifically, this paper has presented work on an innovative method for analysis of large neural assemblies using cross-correlation. However, a number of areas still require attention.

Currently, users are unable to alter the order of the spike trains. The ability to reorder the gird and to fine-tune the display will be implemented.

Empirical testing is underway to evaluate the usability of this visualization method. Subsequently, it is likely that this tool will be integrated into the VISA Toolbox [15].

7 Acknowledgments

This research is supported by the Engineering and Physical Sciences Research Council (EPSRC) grant number GR/N04904. Additionally, the research of Roman Borisyuk is supported in part by the Russian Foundation of Basic Research (grant 99-04-49112) and by EPSRC (grant GR/N63888/01).

8 References

[1] Shneiderman B, 1996. The eyes have it: A task by data type taxonomy for information visualizations. Proc. IEEE Symposium Visual Language, VL, pages 336-343, 3-6.

[2] Borisyuk R.M., Borisyuk G.N., 1997. Information coding on the basis of synchronisation of neuronal activity. BioSystems 40, 3-10

[3] Fries P., Neuenschwander S., Engel A.K., Goebel R. and Singer W., 2001. Rapid feature selective neuronal synchronization through correlated latency shifting. Nature Neuroscience 4(2), 194-200.

[4] North C, Multiple views and tight coupling in visualization: A language, taxonomy, and system. Proc. CSREA CISST 2001 Workshop of fundamental Issues in Visualization, pages 626-632

[5] Baldonado M. Q. W, Woodruff A., and Kuchinsky A 2000. Guidelines for using multiple views in information visualization. Advanced Visual Interfaces, pages 110-119.

[6] Roberts J. C. 1998. On encouraging multiple views for visualization. IV'98 – Proceedings International Conference on Information Visualization, pages 8-14 IEEE Computer Society, July 1998

[7] Robinson D. editor 1998. Neurobiology. Springer, The Open University. ISBN: 3-540-63546-7

[8] Awiszus F 1997. Spike train analysis. Journal of Neuroscience Methods, 74.

[9] MuLab http://mulab.physiol.upenn.edu/index.html

[10] Brillinger D. R. 1979. Confidence intervals for the crosscovariance function. Selecta Statistica Canadiana, V, pages 1-16.

[11] Gerstein, G.L., Aertsen, A.M., 1985. Representation of Cooperative Firing Activity Among Simultaneously Recorded Neurons. Journal of Neurophysiology 54(6), 1513-1528

[12] Gerstein, G.L., Perkel, D.H., Dayhoff, J.E., 1985. Cooperative Firing Activity in Simultaneously Recorded Populations of Neurons: Detection and Measurement. Journal of Neuroscience 5(4), 881-889.

[13] Baker, S.N., Gerstein, G.L., 2000. Improvements to the Sensitivity of the Gravitational Clustering for Multiple Neuron Recordings. Neural Computation 12, 2597 - 2620.

[14] Stuart L., Walter M. and Borisyuk R. 2002. Visualization of synchronous firing in multi-dimensional spike trains. BioSystem 67:265-279

[15] Visualization of Inter-spike Association (VISA) project website http://www.tech.plym.ac.uk/soc/research/neural/staff/lstuart/VISA/index.html

Analysis of multi-dimensional Spike Trains using VISA

L. Stuart, M. Walter and R. Borisyuk

Interactive and Adaptive Systems Group, School of Computing,
University of Plymouth, Plymouth, Devon, UK, PL4 8AA
lstuart@plymouth.ac.uk
http://www.tech.plym.ac.uk/soc/research/neural/members.html#stuart

ABSTRACT

1. Introduction

In general, experts involved in research on neural coding agree that information in nervous system is encoded in the spatio-temporal patterns of spikes. However, there are two distinct opinions about the manner in which this encoding takes place. Some experts consider that temporal coding is based on the exact timing of spiking activity of a single neuron whilst others believe that this encoding is solely based on the firing rate of a single neuron [1]. The authors subscribe to the former opinion. Substantial quantities of simultaneously recorded spike train data exist, in addition to the numerous models developed for the generation of this data. However, current software systems to support the analysis and exploration of these large datasets are not adequate. The main focus of the research presented in this paper is the development of improved software systems to study the spatio-temporal patterns in these datasets.

Experimental data has shown that spatio-temporal patterns are variable within these datasets. Thus, when the same stimulus is presented to a subject twice, the resulting patterns can vary. However, when a larger neural population is studied, it is still possible to identify a sub-population, under variable conditions, that exhibits synchronous activity.

This paper describes several techniques based on Information Visualization that support the exploration of multidimensional spike train datasets and the identification of sub-populations of synchronously active neurons. Traditional measures of assessing synchrony, such as pair-wise cross correlation functions are becoming increasingly time-consuming and complicated when the number of simultaneously recorded spike train increases. Therefore, new, computer methods of dealing with these vast data sets are essential.

2. The VISA Visualisation Tool

In the previous workshop on Neural Coding, 2001, an initial prototype of the visualization tool was presented [2, 3]. This tool is called, VISA, Visualization of Inter-Spike Associations, and it supports the analysis of multidimensional spike train data. It supported the use of the gravity transformation algorithm [4, 5] and the display of its output data using parallel coordinates [6, 7]. Additionally, these parallel coordinates could be animated over time.

Much software development in the area of Information Visualization is now designed upon the much-cited "Information Seeking mantra" introduced by Shneiderman [8]. This mantra states the basic requirements of any useful information visualization system as "Overview first, zoom and filter, then details on demand". This mantra has been adopted as the fundamental premise upon which the VISA tool has been designed. The main aim of this tool is to enable users to view their data at different levels of detail, from abstract representations of the complete data set to specific representations that enable inspection of individual data items. The latest version of VISA includes additional numerical methods and visualization algorithms.

3. Additional functionality

Users require a wide variety of different ways to look at the same data set. This is largely due to the fact that representations accentuate some aspects of data sets but rarely do they expose all features of a data set. Additionally, the increasing size of data sets currently under investigation also means that multiple, and often synchronized, views of data are now required. To support this requirement for a range of representations of the same data set, additional functionality has been incorporated into VISA. This includes the addition of two new visualization tools: the "Correlation Grid"[9] and the "Tunnel"[10] representations.

The Correlation Grid is a technique used to represent the synchronous activity exhibited by clusters of spiking neurons. Let us consider a dataset of *n* spike trains. To quantify the similarity of spike trains, the cross correlation function of each pair of neurons is calculated. Subsequently, each of these functions are normalized using the Brillinger method [11]. This normalization is used to define the distance between two spike trains subsequently used for cluster analysis. The results of cluster analysis are represented using an *n*n* grey-scale grid. The scale goes from white, representing no significant correlation between the two corresponding spike trains, to black, representing the most significant correlation within the current dataset. The Correlation Grid allows clearly visualize the clusters of synchronous spike trains.

The "Tunnel" also allows to find clusters of synchronous activity and this method uses 3D representation of information and immersion possibilities, such as those used in Virtual Reality, to locate neurons that are spiking synchronously. Multiple spike trains are represented as bands or strips that make up the walls of a long tunnel. This representation supports movement of the user through the Tunnel. Additionally, it support the reordering of spike trains, to aid the identification of trains that are synchronized. For example, the cross correlation based clustering method can be used to reorder the spike trains within the tunnel, so that trains that are most similar are adjacent to each other.

4. Results of synchronization analysis supported by visualization

In this section, a simplified case study is described. This case study illustrates how the use of visualization representations supports the investigation of synchrony in artificially simulated neural activity.

4.1 The neuronal assembly

In order to illustrate the new visualization techniques included in the VISA toolbox, a data set was generated, for 20000 msec with a time step of 1msec, using an enhanced integrate-and-fire generatorc[12]. This dataset was based on an assembly of ten neurons with the connection scheme shown in Figure 1.

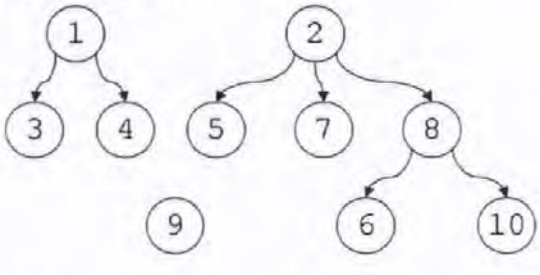


Figure 1: The neuronal assembly used to generate data depicted in Figures 2 and 3.

4.2 The Correlation Grid

The Correlation Grid is a representation that provides an overview of all of the individual cross correlograms for a given data set. An example of this representation for the generated data is shown in Figure 2.

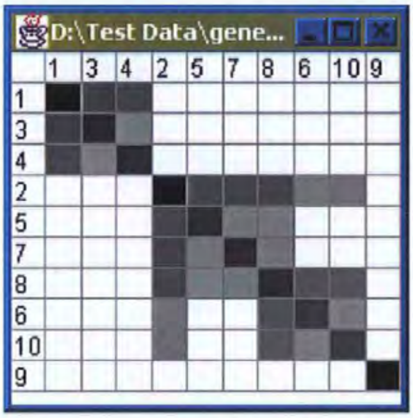


Figure 2: An example of the Correlation Grid

As explained in section 3, this grid is based on the cross correlation functions of neuron pairs. Note that the cluster analysis based on the Brillinger method, inherent in this representation, reorders the spike trains in order to highlight similarity between train pairs. This significantly increases the usefulness of this representation. The Correlation Grid shows two clusters of synchronously working neurons (#1, 3; and 4; # 2, 5, 7, 8, 6, 10) and one independent neuron (#9). Also, the second cluster is consists of two overlapping sub-clusters: (#2, 5, 7, 8; and 8, 6, 10). All these findings correspond to connection scheme for spike train generator (Fig. 1).

4.3 The Tunnel visualization

The "Tunnel" visualization is a method of analyzing the firing patterns of multiple neurons. This environment presents different views of the same dataset and an additional overlay that encodes spike coincidence.

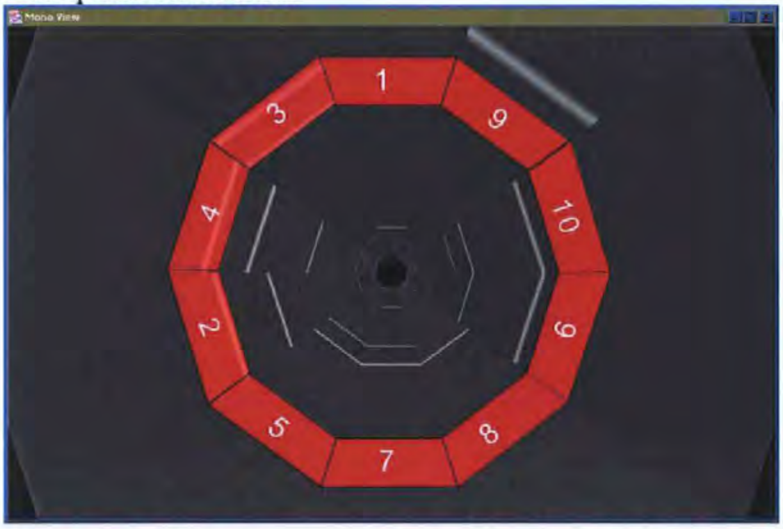


Figure 3: A snapshot of the Tunnel representation of the generated dataset over 500ms.

The Tunnel is a cylindrical virtual environment that has been designed to provide useful user interaction. Figure 3 shows the Tunnel visualization of the generated dataset from 1000ms to 1500ms. Each of the numbered horizontal bands, that comprise this Tunnel visualization, encodes the spike train of the corresponding neuron. The two 'end' bands, band 1 and band 9 in Figure 3, are adjacent to each other, thus forming the tunnel environment.

Note that in figure 3 the order of the spike trains within the Tunnel is based on the cluster analysis shown in Figure 2. Thus, it is possible for the user to reorder spike trains in order to highlight synchrony between trains. Additionally, inside the tunnel, the user is able to navigate or "fly" through the tunnel at different speeds, in order to identify "interesting" features in the dataset at different positions in the tunnel.

5. Conclusion

Since the VISA tool is now equipped with the fundamental functionality required to investigate neural assemblies, this paper will describe a series of analyses that are currently underway. The paper shall report on the usefulness of the visualization techniques currently available, and the results obtained by using a combination of all the techniques available using VISA.

Note that this research and the development of the VISA software is an ongoing project. It is anticipated that further use of the tool will also lead to further requirements definition.

References

[1] Shalden MN and Newsome, WT, (1998), "The variable discharge of cortical neurons: Implications for connectivity, computation, and information coding", Journal of Neuroscience, 18, 3870-3896.

[2] Stuart L, Walter M and Borisyuk R, (2001), "Visualization of multi-dimensional Spike Trains", In Proceedings of 4th International workshop Neural Coding'2001, 47-48, (2001).

[3] Stuart L, Walter M & Borisyuk R (2002), "Visualization of Synchronous Firing in Multi-dimensional Spike Trains", BioSystems, 67, 265-279.

[4] Gerstein GL & Aertsen AM (1985), "Representation of Cooperative Firing Activity Among Simultaneously Recorded Neurons", Journal of Neurophysiology, 54(6), 1513-1528.

[5] Gerstein GL et al. (1985), "Cooperative Firing Activity in Simultaneously Recorded Populations of Neurons: Detection and Measurement", Journal of Neuroscience, 5(4), 881-889.

[6] Inselberg A & Dimsdale B (1990), "Parallel Coordinates: A tool for visualising multidimensional geometry", In Proceedings of Visualization'90, 361-378.

[7] Wegman EJ (1990), "Hyperdimensional Data Analysis Using Parallel Coordinates", Journal of the American Statistical Association, 411(85), 664-675.

[8] Shneiderman B (1996), "The eyes have it: A task by data type taxonomy for information visualizations", In Proceedings of IEEE Symposium on Visual Languages, VL, 3-6, 336-343.

[9] Walter M, Stuart L and Borisyuk R, (2003), "A Compact Visualization for Neural Data", In Proceedings of the 7th IEEE International Conference on Information Visualization, IV03, *to appear.*

[10] Walter M, Stuart L and Borisyuk R, (2003), "The Representation of Neural Data using Visualization", Submitted to the 9th IEEE International Symposium on Information Visualization, InfoVis'03, *submitted.*

[11] Brillinger D R, (1979), "Confidence intervals for the crosscovariance function", *Selecta Statistica Canadiana, V*, 1-16.

[12] Borisyuk RM & Borisyuk GN (1997), "Information coding on the basis of synchronisation of neuronal activity" BioSystems, 40, 3-10.

Evaluation of Spike Train Analysis using Visualization

Martin A. Walter*
mwalter@plymouth.ac.uk

Liz J. Stuart*
lstuart@plymouth.ac:uk

Roman Borisyuk*†
rborisyuk@plymouth.ac.uk

*Centre for Neural and Adaptive Systems, School of Computing, University of Plymouth, Plymouth, Devon, UK
†Institute of Mathematical Problems in Biology, Russian Academy of Sciences, Pushchino, Moscow Region 142 290, Russia

1 Introduction

Many unresolved issues in the field of Neuroscience are dependent upon the comprehension of vast quantities of neural data. Exploration of information processing within the nervous system depends upon the comprehension of this data. This research focuses on simultaneously recorded multi-dimensional spike train data that is used in the investigation of the principle of synchronisation of neural activity [Borisyuk and Borisyuk 1997].

In order to "mine" this data for inherent information, these data sets require thorough and diverse analysis. Since these data sets are large, conventional means of analysis, such as the use of cross-correlograms, are insufficient on their own.

Thus, advanced techniques are being developed to exploit new and traditional analysis methods for larger data sets.

This paper presents the initial evaluation of a method of dealing with the analysis of relatively large numbers of spike trains, based upon the cross-correlogram, called the Correlation Grid [Walter et al. 2003].

2 The VISA Tool

An initial prototype of the visualization tool was presented at the Neural Coding Workshop in 2001[Stuart et al. 2002(a)]. This tool is called; VISA, Visualization of Inter-Spike Associations, and it supports the analysis of multidimensional spike train data. It supported the use of the gravity transformation algorithm [Gerstein and Aertsen 1985] and the display of its output data using parallel coordinates [Inselberg and Dimsdale 1990]. Additionally, these parallel coordinates could be animated over time [Stuart et al. 2002(b)].

Much software development in the area of Information Visualization is now designed upon the much-cited "Information Seeking mantra" introduced by Shneiderman [1996]. This mantra states the basic requirements of any useful information visualization system as "Overview first, zoom and filter, then details on demand". This mantra has been adopted as the fundamental premise upon which the VISA tool has been designed. The main aim of this tool is to enable users to view their data at different levels of detail, from abstract representations of the complete data set to specific representations that enable inspection of individual data items. The latest version of VISA includes additional numerical methods and visualization algorithms, including the Correlation Grid [Walter et al. 2003] and cluster analysis.

3 The Cross-Correlogram

A cross-correlogram is used to visually represent the synchrony between the spike trains of two neurons. This representation is plotted as a histogram and represents the spiking activity of one neuron, designated to be the 'target' neuron, with respect to a second neuron, designated the 'reference' neuron.

The cross-correlogram is analysed for the existence of 'significant' peaks as defined by Brillinger[1979]. The height, position (with respect to zero) and the number of peaks all help to determine the type and number of connections, if any, between the neurons.

A cross-correlogram with a significant peak after zero indicates that the target neuron has a tendency to spike after the reference neuron. In contrast, a significant peak before zero would indicate that the reference neuron tends to spike after the target neuron. A peak at or near zero indicates the neurons tend to spike coincidently.

4 The Correlation Grid

The Correlation Gird presents users with an overview of the cross-correlogram results, for a number of spike trains.

So, for a given dataset, of *n* spike trains, all unique cross-correlograms are generated, for specified correlation parameters of bin and window size. Subsequently, the cross-correlograms are normalised using the Brillinger[1979] method. Finally, the results of these cross-correlograms are displayed as an *n*-by-*n* grid of grey scale cells, representing the individual correlations between all pairs of spike trains.

The grid encodes the 'height' of the largest peak in each cross-correlogram. The peaks are in coded from white, representing no peak, to black, representing the largest peak in the grid.

The user can select whether to view 'all peaks' or just significant peaks. Significant peaks are those that lie outside of the Brillinger confidence interval specified for the grid. In addition, the individual cross-correlograms can be view by simply selecting the corresponding cell in the grid.

The identification of groups, or clusters, of correlations is key to understanding the relationships between the underlying neurons. It is possible to identify these clusters visually, however this is not easily expanded for problems with larger datasets.

To aid with the identification of correlation clusters, a statistical cluster analysis method has been implemented. This method uses the height of the most significant peak, if any exist, of each cross-correlogram to build a dendrograph for the correlation gird. This in turn is used to generate the initial display order of the spike trains.

5 Correlation Grid Trial

Note that all spike trains used for experimentation were generated using an enhanced Integrate and Fire generator defined by Borisyuk and Borisyuk [1997].

For this example, a dataset of fifteen spike trains was generated, over 2000ms, for the assembly of neurons shown in Figure 5-1.

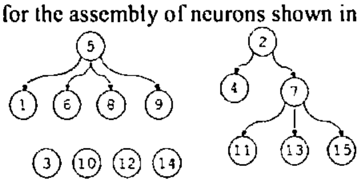


Figure 5-1 The assembly of 15 neurons

A Correlation Grid for this data was generated, with bin size 3ms and window size of 100 bins, and cluster analysis applied. This Grid is shown in Figure 5-2.

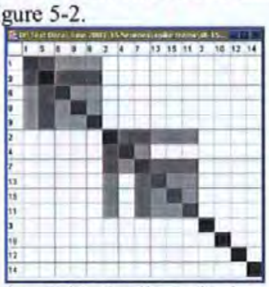


Figure 5-2 Correlation Grid of spike train data in Figure 5-1, with bin size 3ms, window 100

On initial inspection of the Grid, three groups of spike trains are identifiable, the first, spike trains 1, 5, 6, 8 and 9. The second, containing spike trains 2, 4, 7, 13, 15 and 11. The third, spike trains 3, 10, 12 and 14. Based upon experimental experience, it is possible to infer there are two separate neuronal assemblies, corresponding to the spike trains in the first and second groups, and several separate, unrelated neurons, the third group.

On closer inspection of the first group, it is possible to see that the correlations between spike train 5 and trains 1, 6, 8 and 9 are the stronger, darker grey, compared to the rest of this group. Note, auto-correlations, correlation of a spike train with itself, are shown in the grid and are generally the darkest greys of the whole grid.

It is possible to hypothesize from this that neuron 5 connects to neurons 1, 6, 8 and 9. In addition, the correlations between spike trains 1, 6, 8 and 9 are due to these connections, rather than a separate connection. More over, this hypothesis can be confirmed by inspecting the cross-correlations of these spike trains. Figure 5-3 shows the correlation between spike trains (a) 5 and 6, (b) 1 and 5 and (c) 1 and 6.

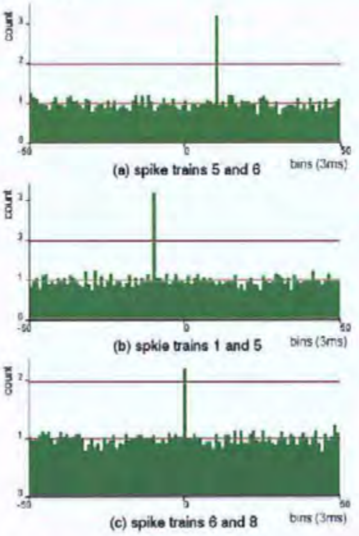


Figure 5-3 Cross-correlograms from grid shown in Figure 5-2 for spike trains (a) 5 and 6, (b) 1 and 5 and (c) 1 and 6

In Figure 5-3 (a), observe that the correlation between spike trains 5 and 6 is high. Additionally, this correlation peak is to the right of zero, thus indicating a "positive" delay for the correlation. This shows that the connection is from neuron 5 to neuron 6, as neuron 6 has a tendency to spike after neuron 5. Likewise, there is a strong correlation between neurons 1 and 5, see Figure 5-3 (b). However this correlation has a "negative" delay, indicating a connection from 5 to 1. Contrasting to these correlations, spike trains 1 and 6, see Figure 5-3 (c), is not relatively strong (note differing scales) and the peak is around zero. This indicates that the two neurons have a tendency to spike at the same time, thus indicating that they are likely to have the same input. In this scenario, it is highly likely that they both have connections from neuron 5. The resultant correlations for the other neuron pairs in the first group are similar to these examples. Therefore, it is possible to deduce the neuronal configuration for the first group as that shown in Figure 5-1.

A similar, more extended, yet equally successful approach is taken to the analysis of the second group. However, due to space limitations, this analysis is not included here.

From these initial trials, the Grid appears to be useful. Its overview, filtering and sorting functionality in addition to the details of individual cross-correlations makes it possible to "recover" the assembly of neurons from the spike train dataset. This is very significant to our users as many neurophysiologists record "real" data in their laboratories that requires the type of in-depth analysis that we have presented in this paper.

6 Future Work

The work presented in this paper is part of a Visualisation project, led by Dr. Liz Stuart, called Visualisation of Inter-Spike Associations [VISA]. Specifically, this paper has presented the preliminary evaluation of an information visualisation method developed for the analysis of large neural assemblies using cross-correlation.

Significant development of this visualization method, particularly with respect to user interaction, is planned. Further empirical testing is planned and underway.

7 References

BORISYUK R.M., AND BORISYUK G.N., 1997. Information coding on the basis of synchronisation of neuronal activity. *BioSystems 40*, 3-10

BRILLINGER D. R. 1979. Confidence intervals for the crosscovariance function. *Selecta Statistica Canadiana, V*, pages 1-16.

GERSTEIN, G.L. AND AERTSEN, A.M., 1985. Representation of Cooperative Firing Activity Among Simultaneously Recorded Neurons. *Journal of Neurophysiology 54(6),* 1513-1528

INSELBERG A. AND DIMSDALE B. 1990 Parallel Coordinates: A tool for visualising multidimensional geometry, in: *Proceedings of Visualization'90*, pp. 361-378.

SHNEIDERMAN B, 1996. The eyes have it: A task by data type taxonomy for information visualizations. *Proc. IEEE Symposium Visual Language, VL*, pages 336-343, 3-6.

STUART L., WALTER M. AND BORISYUK R. 2002 (a). Visualization of synchronous firing in multi-dimensional spike trains. *BioSystem 67*:265-279

STUART, L., WALTER, M. AND R. BORISYUK 2002 (b), Visualisation of Neurophsyiological Data, Presented at the 8th IEEE International Conference on Information Visualization.

VISA Visualization of Inter-spike Association project website http://www.plymouth.ac.uk/infovis/

WALTER M., STUART L. AND BORISYUK R. 2003 Spike Train Correlation Visualization, in: Proceedings of 7th International Conference on Information Visualization, pp. 555-560

The Correlation Grid: Analysis of Synchronous Spiking in Multi-dimensional Spike Train Data and Identification of Feasible Connection Architectures

L. Stuart[*], M. Walter[*] and R. Borisyuk[†‡]

[*]The Visualization Lab, Centre for Intelligent Interactive Systems, University of Plymouth, Plymouth, UK

[†]Centre for Theoretical and Computational Neuroscience, University of Plymouth, Plymouth, UK

[‡]Institute of Mathematical Problems in Biology, Russian Academy of Sciences, Pushchino, Moscow Region 142 290, Russia

Abstract

This paper presents a visualization technique specifically designed to support the analysis of synchronous firings in multiple, simultaneously recorded, spike trains. This technique, called the Correlation Grid, enables investigators to identify groups of spike trains, where each pair of spike trains has a high probability of generating spikes approximately simultaneously or within a constant time shift. Moreover, the correlation grid was developed to help solve the following reverse problem: identification of the connection architecture between spike train generating units, which may produce a spike train dataset similar to the one under analysis. To demonstrate the efficacy of this approach, results are presented from a study of three simulated, noisy, spike train datasets. The parameters of the simulated neurons were chosen to reflect the typical characteristics of cortical pyramidal neurons. The schemes of neuronal connections were not known to the analysts. Nevertheless, the correlation grid enabled the analysts to find the correct connection architecture for each of these three data sets.

1 Introduction

Synchronisation of neural discharges is considered to be an important principle of information processing by cortical neural circuits (Nase et al. 2003; Neuenschwander et al., 2003, Schmidt 2003). Analysis of synchronisation of simultaneously recorded spike trains is usually based on the calculation of a counting function, such as a cross-correlation function or a cross-correlogram (Gerstein and Kirkland, 2001). Cross-correlograms are a common and useful means of representing the relationship between pairs of spike trains, recorded in this way.

However, for any significantly sized neuronal architecture, numerous cross-correlograms would require in-depth analysis in order to identify synchronous activity in and between groups of spike trains. Indeed, investigation into the functional connectivity of neuron groups is a very important area of research.

The objective is the identification of a feasible architecture of connections between elements that could account for the original spike train data and subsequent correlation functions. Typically in mathematics, the reverse problem is very difficult to solve and non-unique solutions may exist. Nevertheless, the results of our blind testing are promising.

Furthermore, new problems are posed by the increasingly large neural assemblies that are currently recorded.

In this paper, the role that information visualization may play in alleviating some of these problems is discussed. A visualization technique, called the Correlation Grid (Walter et al. 2003), is used to analyse a simultaneously recorded dataset of *n* spike trains. Subsequently, this grid is used to identify clusters of synchronous spike trains. Thus, it supports the proposal of a scheme of functional connectivity based on the fact that high correlation between spike trains corresponds to significant functional connection.

In order to imitate the experimental data of multi-spike train recordings, a biologically-inspired generator of spike trains with interconnections according to a predefined connection scheme, (Borisyuk 2002) is used. Each spike train is generated on the basis of an enhanced integrate and fire model and some specified connection architecture with particular synaptic weights. Note that all results in this paper are obtained in the regime of blind testing. Only the spike trains were made available for analysis. These spike trains are used to generate the Correlation Grid, which is analysed and used to identify the underlying connection architecture of neurons. Finally, this proposed architecture is assessed with respect to the original used to generate the data.

Progressively, the number of correct connection identifications has increased during empirical testing, resulting in improvements to the methodology used. Currently, it is possible to obtain a completely accurate connection scheme. In this paper, three trials are presented, which correspond to a different number of spike trains and/or connection schemes.

2 Brillinger normalisation of cross-correlogram

The cross-correlogram for each pair of spike trains is calculated in order to create the correlation grid. To create the cross-correlogram of a 'target' spike train relative to a 'reference' spike train, a standard counting function (Gerstein 2001, MuLab) is used. This function counts the number of spikes in the reference spike train relative to the spikes in the target spike train. Figure 2-1(i) depicts a typical cross-correlogram for a pair of connected neurons with a time shift corresponding to the delay in spike propagation from one neuron to another. In contrast, a typically flat cross-correlogram for a pair of unconnected neurons is shown in Figure 2-1(ii).

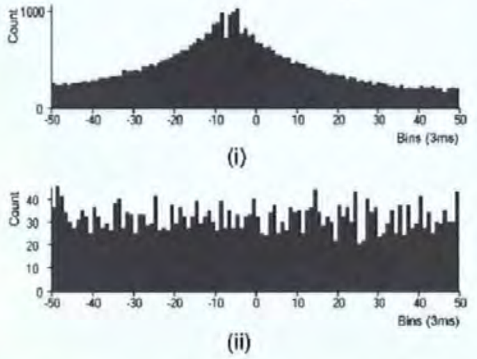


Figure 2-1 An example of a cross-correlogram of (i) two correlated spike trains and (ii) two uncorrelated spike trains

In order to make statistically significant judgements regarding the nature of the peaks, in the cross-correlogram, the Brillinger normalisation is applied to the data (Brillinger 1979) and the peaks exceeding the higher bound of the 95% confidence interval are considered to be significant peaks. Figure 2-2 shows a normalised cross-correlograms for the same data as shown in Figure 2-1 and boundaries of 95% confidence interval. The highest significant peak is referred to as the 'main' peak. Thus, the main peak can be considered to be a measure of the proximity of two spike trains.

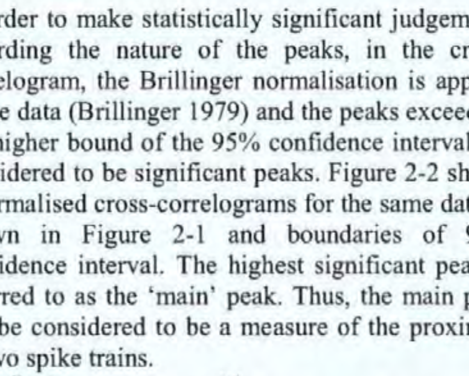


Figure 2-2 An example of a Brillinger normalised cross correlogram with confidence interval of (i) two correlated spike trains and (ii) two uncorrelated spike trains.

Subsequently, the value of this main peak is used to quantify the distance between pairs of spike trains in the clustering algorithm as well as the density of shading in the corresponding cell of the correlation grid.

3 The correlation grid

The correlation grid is an overview of multiple cross-correlograms for a number of spike trains. Hence, for a given dataset, of *n* spike trains, all pair wise cross-correlograms are generated and normalised using the Brillinger method, and the main peaks $c_{i,j}, (i=1,...,n, j=1,...,n)$ are calculated for all pairs. Finally, the results are displayed as an *n*-by-*n* symmetrical grid of grey scale cells, representing correlations between all pairs of spike trains. Thus, the magnitudes of main peaks are encoded from white, representing a non-significant peak, to black, representing the largest peak in the grid. The user has the flexibility to view 'all peaks' or solely significant peaks. Significant peaks are those that exceed the higher bound of the confidence interval. In this paper, we analyse neural circuits with positive connection strengths only. When the main peak of the cross-correlogram exceeds the higher bound of the Brillinger interval, a positive functional connection exists between the neurons generating these spike trains. In principle, negative connections should also be considered as they will result in peaks that are lower than the low bound of the significance interval.

Additionally, it is useful to reorder the rows and columns of correlation grid in order to highlight the inherent relationships between multiple spike trains. The method used to accomplish this reordering is the cluster analysis algorithm.

3.1 The clustering algorithm

Let us consider an *n*-by-*n* correlation grid of cells as a matrix of similarities between objects (spike trains). Thus, the pair of spike trains who cross-correlogram has the largest main peak is considered to be the most similar (the most correlated) pair of spike trains.

In the trials, different algorithms were used to cluster spike trains. These included the following methods: nearest neighbour (the minimum of measures between objects in two groups), furthest neighbour (the maximum of measures between objects in two groups), and a centroid clustering algorithm. In conclusion, the most efficient algorithm is the furthest neighbour method. Intuitively this algorithm creates tight clusters and all objects inside the cluster have limited dissimilarity.

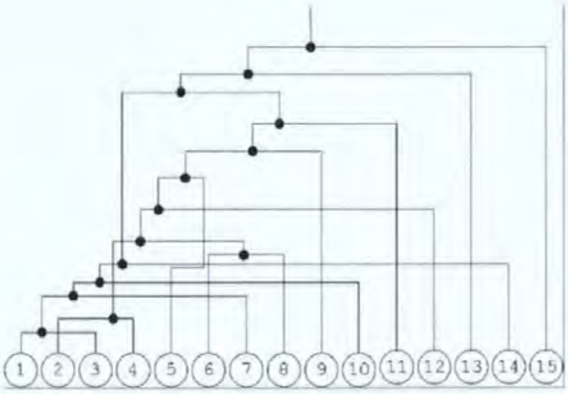


Figure 3-1 Cluster analysis dendrogram for trial one data, also shown as the correlation grid in Figure 4-3.

Figure 3-1 shows a cluster analysis dendrogram for the data from trial one. This data is also depicted as the 'initial' correlation grid shown in Figure 4-3. In order to enhance the perception of groups within this correlation grid, a left-hand recursive algorithm is used to redefine the ordering of spike trains.

This algorithm descends this binary tree (see the dendrogram, Figure 3-1) and at each node, it initially follows the leftmost branch. Thus, the algorithm recursively follows the leftmost branch until it reaches a leaf node. Upon finding a left node, the algorithm then traverses the rightmost branch of the current sub-tree before ascending back up the tree to find the next rightmost branch to be followed.

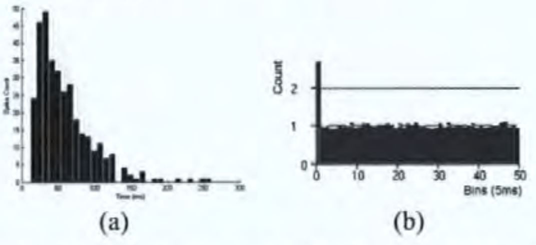
Subsequently, this new ordering is used to re-order the spike trains in the correlation grid. The resultant correlation grid is shown in Figure 4-5. This shows a significant improvement to the visualization of these groups of synchronously spiking trains. From this grid, it is immediately possible to infer that all spike trains are arranged into two separate clusters: (1, 3, 7, 10 and 14) and (2, 4, 6, 8, 12, 5, 9, and 11) and also that spike trains 13 and 15 are independent.

Additionally, it is immediately notable that the second main cluster has some overlapping connections that require further analysis.

4 Data analysis and connection structure identification

In this section, the results of three trials are presented. Each dataset was generated using simulations of integrate and fire neurons with particular coupling between elements. The parameters of the model were chosen to mimic the general neurophysiological characteristics of cortical neurons. All connection strengths in these simulations were chosen to be positive.

For example, in trial one, an assembly of 15 neurons was simulated for a period of 20000 ms. The mean inter-spike interval (ISI) was 75 ms, the standard deviation of the ISI of the dataset was 53 ms and its coefficient of variation was 0.7. The ISI Histogram for spike train number 6 is shown in Figure 4-1 (a) and its autocorrelation is shown in Figure 4-1. This figure depicts (a) the ISI histogram of spike train 6 and (b) the autocorrelation of spike train 6 from the trial one dataset.



(a) (b)

Figure 4-1 This figure depicts (a) the ISI histogram of spike train 6 and (b) the autocorrelation of spike train 6 from the trial one dataset.

The raster plot of the first 3000ms of this data is also shown in Figure 4-2.

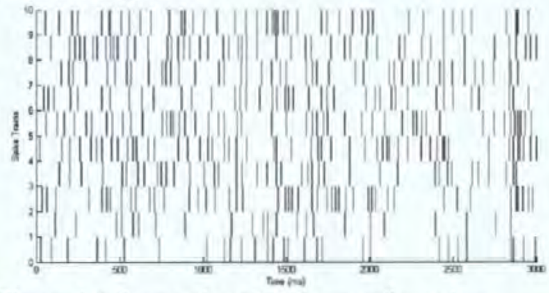


Figure 4-2 Raster plot of trial one spike trains

We call this analysis "blind trials" because the only spike trains have been available for analysis.

4.1 Trial One. Creating the correlation grid

Initially, the cross-correlogram for each pair of fifteen spike trains is calculated with a bin size of 1ms and a time-window of 100ms (100 bins). The correlation grid for the first data set is shown in Figure 4-3.

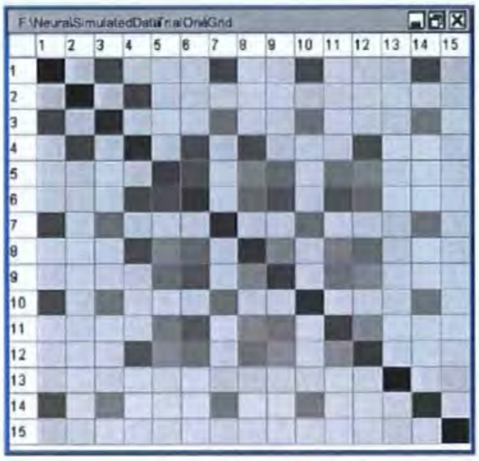


Figure 4-3 The initial correlation grid for trial one data

For step two, all peaks that do not exceed the higher bound of the confidence interval of the Brillinger normalization are ignored. Thus, Figure 4-4 solely shows the significant values of the main peaks as insignificant peaks are represented in white.

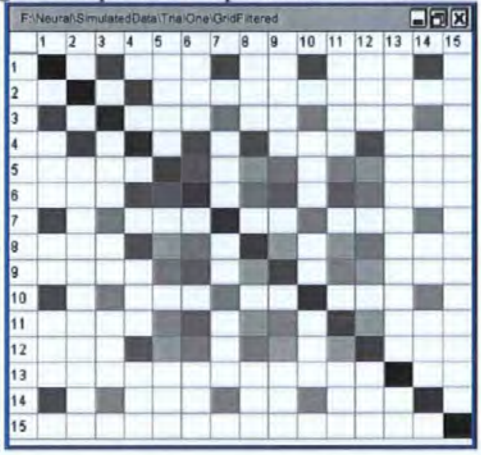


Figure 4-4 The filtered correlation grid for trial one data

The final stage of creating the correlation grid is the application of the clustering algorithm. Figure 4-5 shows the striking affect that clustering may have on a correlation grid. This improved ordering supports the identification of groups and their position within any hierarchies that exist. The interpretation of the correlation grid is discussed in the following section.

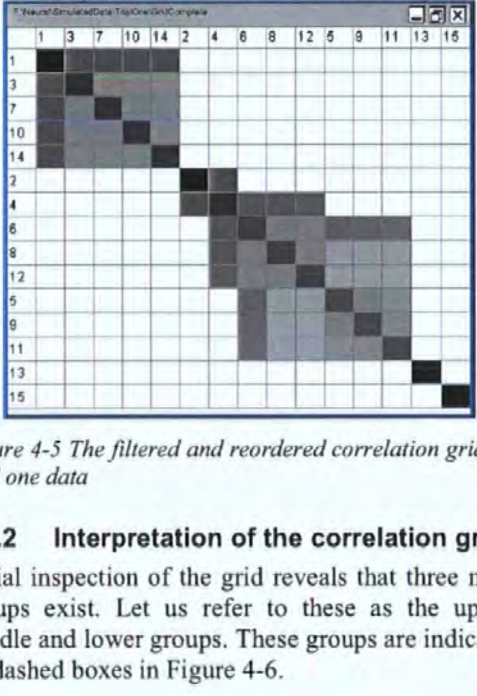


Figure 4-5 The filtered and reordered correlation grid for trial one data

4.1.2 Interpretation of the correlation grid

Initial inspection of the grid reveals that three main groups exist. Let us refer to these as the upper, middle and lower groups. These groups are indicated by dashed boxes in Figure 4-6.

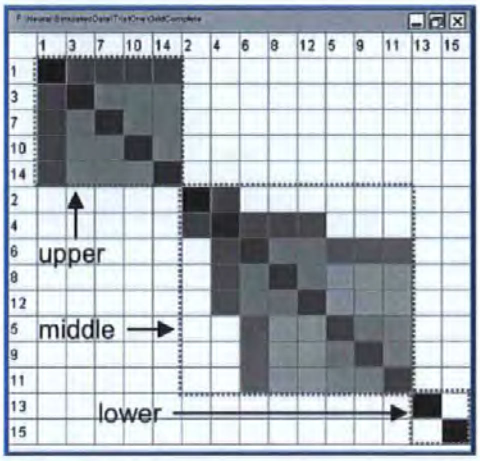


Figure 4-6 The final correlation grid depicting the three main groups in the trial one dataset

The top and middle groups are examined in significant detail in sections 4.1.3 and 4.1.4.

The lower group is very clear. As there is no indication of any significant relationships of either neuron with any other neuron in the assembly, it is feasible to conclude that both neurons 13 and 15 are completely unconnected.

4.1.3 The upper group of trial one

In order to interpret the upper group, the top portion of the correlation grid is enlarged and is shown in Figure 4-7.

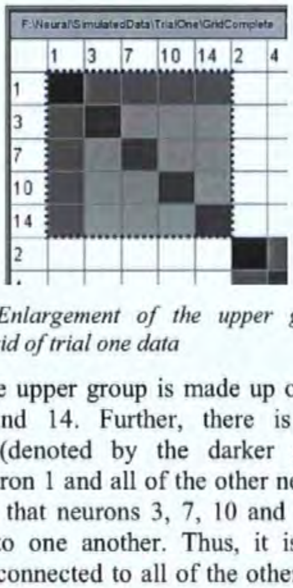


Figure 4-7 Enlargement of the upper group of the correlation grid of trial one data

Note that the upper group is made up of neurons 1, 3, 7, 10 and 14. Further, there is a stronger correlation (denoted by the darker grey shade) between neuron 1 and all of the other neurons in the group. Note that neurons 3, 7, 10 and 14 exhibit a correlation to one another. Thus, it is likely that neuron 1 is connected to all of the other neurons: 3, 7, 10 and 14. Thus, it is likely that the correlation between neurons 3, 7, 10 and 14 is due to the fact that they have a common input.

This hypothesis is confirmed by closer inspection of the cross-correlograms of the neuron pairs (1, 3) and (3, 7), shown in Figure 4-8 and Figure 4-9, respectively.

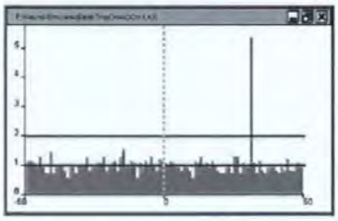


Figure 4-8 The cross-correlogram of spike trains 1 and 3 of trial one

Figure 4-8 depicts the correlation of neurons 1 and 3. As anticipated, there is a time delay between the spikes of these neurons. From the diagram, it is possible to deduce that there is a high probability that a spike will be generated by neuron 3 approximately 25-30ms after neuron 1 spikes. Thus, there is an excitatory connection from neuron 1 to neuron 3.

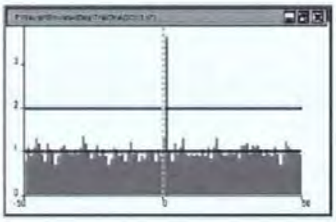


Figure 4-9 The cross-correlogram of spike trains 3 and 7 of trail one

Figure 4-9 depicts the correlation of neurons 3 and 7. As anticipated, there is direct synchronous spiking activity between these two neurons due to the fact that both are stimulated by neuron 1.

All of these observations support the hypothesis that the sub-assembly of the upper group is a circuit with a common source. This is depicted in Figure 4-10.

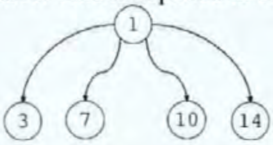


Figure 4-10 The neuronal assembly of the upper group of trial one

4.1.4 The middle group of trial one

In order to interpret the middle group, the middle portion of the correlation grid is enlarged and shown in Figure 4-11.

From this Figure, three main groups are apparent. The 2-group of neurons 2 and 4, the 4-group made up of neurons 4, 6, 8 and 12 and the 6-group made up of 6, 8, 12, 5, 9 and 11. Note that the 2-group and 4-group overlap, as do the 4-group and 6-group. This overlapping indicates that there are connections between these groups and the overlapping neurons in principle could provide these couplings.

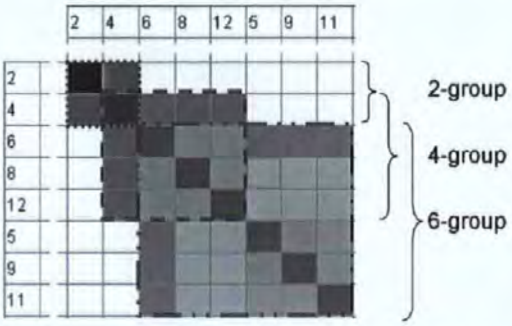


Figure 4-11 Enlargement of the middle group of the correlation grid of trial one data

The 2-group is relatively straightforward. It depicts neuron 2's influence on the spike trains of neuron 4. The 4-group has a similar structure depicted in Figure 4-7. Thus, it is possible to infer that neuron 4 is a common input to neurons 6, 8 and 12.

The 6-group requires further analysis. Thus, it is shown again in Figure 4-12. In this figure, the different values of the main peaks within the 6-group are highlighted using solid lines. Very high correlation exists between neuron 6 and neurons 5, 9 and 11. High correlation exists between these neurons 5, 9 and 11. Finally, there is a low correlation between neurons 8 and 12 with neurons 5, 9 and 11.

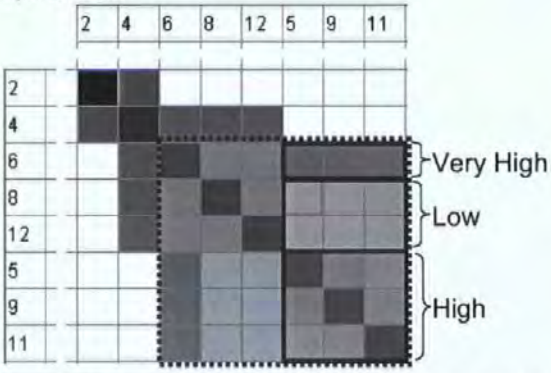


Figure 4-12 Enlargement of the middle group of the correlation grid of trial one data with further classification

The higher correlations form a familiar hierarchical pattern. Thus, neuron 6 connects to 5, 9 and 11. However, this is still some ambiguity regarding the connections between neurons 5, 9 and 11 with neurons 8 and 12.

It is probable that this low correlation is attributable to the fact that neuron 4 is a common input to neurons 8 and 12 and also to neurons 5, 9 and 11 via neuron 6. To verify this assertion, the cross-correlogram of neurons 8 and 11, shown in 4-13, is examined. As anticipated, there is a delayed correlation due to the fact that neuron 4 excites both neurons 8 and 11; however the excitation of neuron 11 is via neuron 6 accounting for the delayed correlation peak in Figure 4-13.

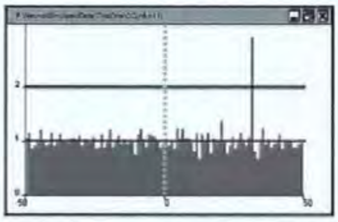


Figure 4-13 The cross-correlogram of spike trains 8 and 11 of trial one

4.1.5 Summary of trial one observations

From this analysis, it is now possible to re-create the coupling structure of this assembly of neurons based on the correlation grid and details from some of the cross-correlograms. This is depicted in Figure 4-14. After completion of the analysis we have compared the result obtained with connection scheme used in simulations and have found their complete correspondence.

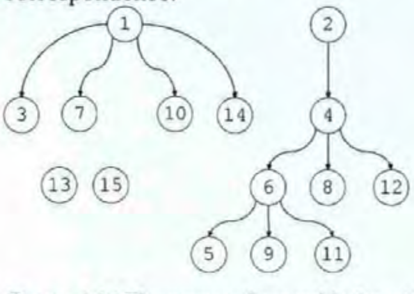


Figure 4-14 The neuronal assembly for trial one

4.2 Trial two

For this trial, an assembly of 10 neurons were simulated for a period of 100000ms. The raster plot for a portion of this data is shown in Figure 4-15. Note that the architecture of connections of the neurons in this assembly was unknown to the analysts prior to the investigation. Furthermore, the results of the analysis yielded an entirely accurate neural architecture.

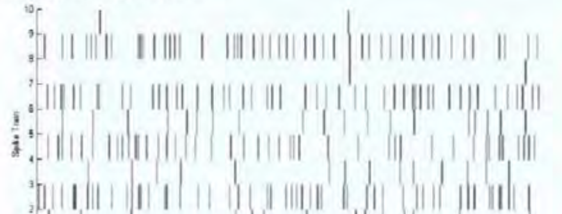


Figure 4-15 Raster plot of trial two spike trains

4.2.1 The correlation grid for trial two

Initially, the correlation grid was generated for this data, based on calculations of cross-correlograms with a bin size of 1ms and a time-window of 100ms (100 bins). Subsequently, the grid was filtered and clustered according to algorithms described above and the resultant correlation grid is shown in Figure 4-16.

Initial inspection of the grid reveals that there is considerable interconnection within this data set. In particular, note the distribution of grey cells in the grid. This grid can be partitioned into three groups: the upper, middle and lower groups as depicted in Figure 4-16. The upper and lower groups of trial two are discussed in detail in sections 4.2.2 and 4.2.3 respectively.

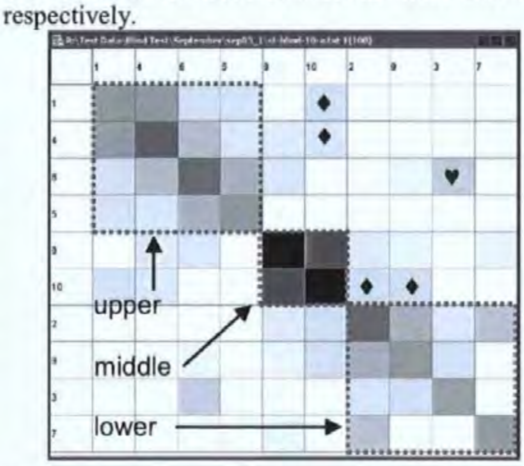


Figure 4-16 The final correlation grid depicting the three main groups of the trial two dataset

The most dominant feature of the correlation grid shown in Figure 4-16 is the correlation between spike trains 8 and 10, the middle group as indicated in the diagram. Generally, this feature infers a direct relationship between the two neurons. Further examination of the cross-correlogram of this pair (time delay in spike propagation from neuron 10 to 8), confirms that there is a connection from neuron 10 to neuron 8.

Another dominant feature of the correlation grid shown in Figure 4-16 is that all three groups are somehow linked to neuron 10. This is confirmed by the shading in all of the cells labelled with a diamond (♦). This indicates that neuron 10 is likely to be the root of the assembly.

4.2.2 The upper group of trial two

In order to interpret the upper group of trial two, the top portion of the correlation grid is enlarged and shown in Figure 4-17.

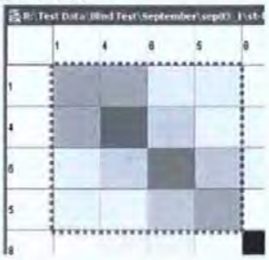


Figure 4-17 Enlargement of the upper group of the correlation grid of trial two data

Note that the upper group consists of neurons 1, 4, 6, and 5. Further, there is a stronger correlation between spike trains 1 and 4; 4 and 6; 6 and 5 in comparison to the remainder. Therefore, it is likely that these pairs are connected. Examination of their respective cross-correlograms confirms this hypothesis.

The correlation between spike trains 1 and 6 is significant but unclear. Therefore, the cross-correlogram for this pair, shown in Figure 4-18, is examined. Note that in this cross-correlogram the main peak occurs around 15ms. Thus, there is a time delay in the spike propagation from neuron 1 to neuron 6. There is also a second (less significant) peak with a slightly larger time delay.

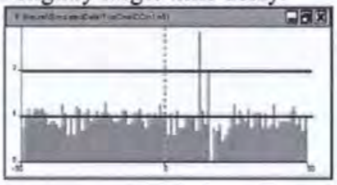


Figure 4-18 The cross-correlogram of spike trains 1 and 6 of trial two

These time delays indicate that the corresponding neurons are connected via some intermediate neuron(s). The presence of two peaks in the cross-correlogram suggests that two different paths exist between the neurons 1 and 6.

The strength of correlation between neurons 4 and 5 also indicates significant correlation. Upon closer inspection, the cross-correlogram of spike trains 4 and 5 shows a correlation peak close to zero (see Figure 4-19). This indicates that these neurons spike synchronously. Thus, suggesting that the two neurons are receiving common input.

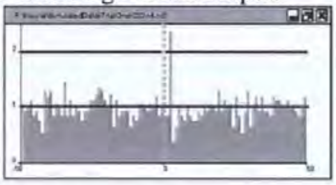


Figure 4-19 The cross-correlogram of spike trains 4 and 5 of trial two

From these observations, it is possible to deduce the connection scheme shown in Figure 4-20. This architecture shows that neuron 1 is a common source to neurons 4 and 5 and that they both have connections to neuron 6.

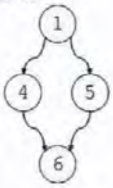


Figure 4-20 The neuronal assembly of the upper group of trial two

4.2.3 The lower group of trial two

In order to interpret the lower group, the lower portion of the correlation grid is enlarged and shown in Figure 4-21.

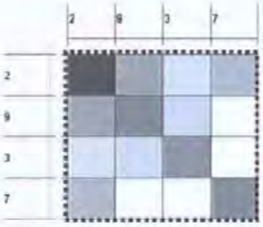


Figure 4-21 Enlargement of the lower group of the correlation grid of trial two data

From this figure, it is apparent that spike train 2 correlates strongly with both spike trains 7 and 9. Likewise, spike train 9 correlates with spike train 3. In addition, spike train 2 correlates, less strongly, with spike train 3. It is possible to hypothesise from these observations that a connection exists between neuron 9 and both neurons 2 and 3. Similarly, that there is a connection between neurons 7 and 3. Inspection of the cross-correlograms for these pairs confirms this hypothesis.

It is also possible to specify the direction of these connections, as shown in Figure 4-22. This structure also explains the weaker correlation between spike trains 2 and 3, as these neurons both receive input from neuron 9.

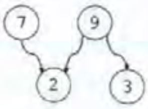


Figure 4-22 The neuronal assembly of the lower group of trial two

4.2.4 Interconnection of upper and lower groups in trial two

As hypothesised previously, neuron 10 links both the upper and lower groups. By investigating the cross-correlograms of spike train 10 with each of the upper and the lower groups (not shown here due to space limitations), it is possible to deduce that neuron 10 connects to both neurons 9 and 1.

From the grid, it is also possible to observe another link, between the upper and lower groups. This correlation is between spike trains 3 and 6 and is indicated by the heart symbol (♥) in Figure 4-16.

Note that the cross-correlogram of neurons 3 and 6 showed a single peak, slightly delayed, thus indicating a direct connection between neurons 3 and 6.

4.2.5 Summary of trial two observations

From this analysis, the overview and details of the correlation grid, it is possible to deduce the underlying neuronal assembly of the data set, as depicted in Figure 4-23. The structure of this assembly also explains the weak correlations, shown in the grid, which have not yet been considered.

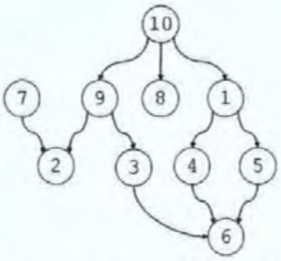


Figure 4-23 The neuronal assembly of trial two

The correlation between spike trains 10 and 4 is clear; neuron 10 is connected to neuron 4 via neuron 1. Likewise, the correlation between spike trains 8 and 9 is due to common input. The correlations between spike trains 4 and 8; 6 and 8; 2 and 8 is not immediately obvious. However, these correlations are attributable to the fact that all inputs are governed by neuron 10.

This assembly was completely and correctly reassembled based on the data available from the cross-correlograms and the correlation grid.

4.3 Trial three

For this trial, an assembly of 10 neurons were simulated for a period of 300000ms. Note that the architecture of connections of the neurons in this assembly was unknown to the analysts prior to the investigation. Again, the results of the analysis yielded an entirely accurate neural architecture.

4.3.1 The correlation grid for trial three

A correlation grid was generated for this data, with a bin size of 1ms and a time window of 100 bins. The grid was subsequently filtered and clustered; the resultant correlation grid is shown in Figure 4-24.

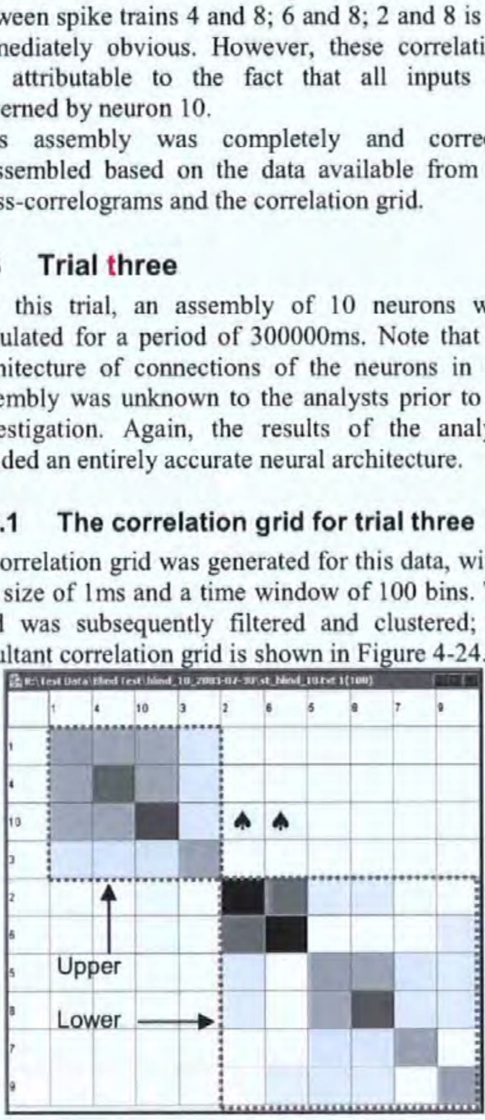


Figure 4-24 The final correlation grid depicting the two main groups of the trial three dataset

Initial inspection of the grid reveals that two main groups exist. Let us refer to these as the upper and lower groups as indicated in Figure 4-24. These groups are examined in detail in sections 4.3.2 and 4.3.3.

In addition, there appears to be a link between the upper and lower groups. Note the correlations between spike trains 2 and 10 and spike trains 6 and 10, indicated by the spade symbols (♠) in Figure 4-24.

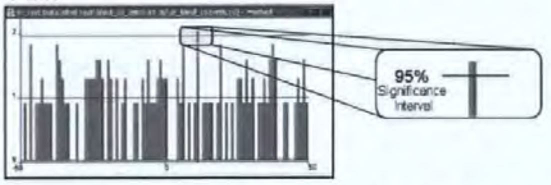


Figure 4-25 The cross-correlogram of spike trains 6 and 10 of trail three. Note that there are no truly signification peaks in this cross-correlogram

Closer inspection of the cross-correlograms, for these pairs, shows a peak barely higher than the confidence interval, see Figure 4-25. For this reason, these correlations can be discounted. Thus, it is possible to conclude that two independent groups exist.

4.3.2 The upper group of trial three

In order to interpret the upper group of trial three, the top portion of the correlation grid is enlarged, as shown in Figure 4-26.

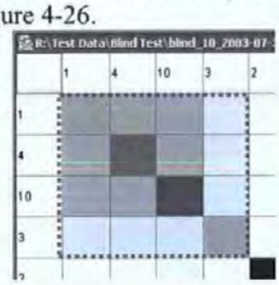


Figure 4-26 Enlargement of the upper group of the correlation grid of trial three data

Note that the upper group consists of neurons 1, 4, 10 and 3, as ordered in the grid. Further, there are strong correlations between spike trains 1, 4 and 10. Also note that spike train 3 correlates, relatively weakly, with spike trains 1, 4 and 10. Further to closer inspection of their cross-correlograms, it is possible to deduce that neuron 1 connects to neurons 3, 4 and 10, as shown in Figure 4-27.

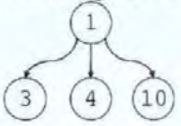


Figure 4-27 The neuronal assembly of the upper group of trial three

In addition, it is possible to observe that the strength of the connection between neurons 1 and 3, see Figure 4-28 for the cross-correlogram, is weaker than that of the connections between neurons pairs (1, 4) and (1, 10), see Figure 4-29.

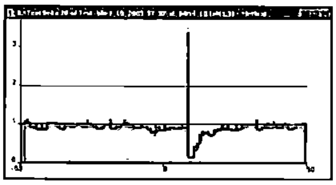


Figure 4-28 The cross-correlogram of spike trains 1 and 3 of trail three

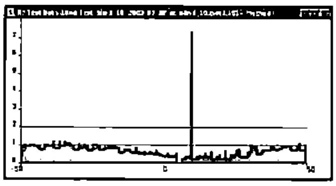


Figure 4-29 The cross-correlogram of spike trains 1 and 10 of trail three

4.3.3 The lower group of trial three

In order to interpret the lower group of trial three, the lower portion of the correlation grid is enlarged and is shown in Figure 4-30.

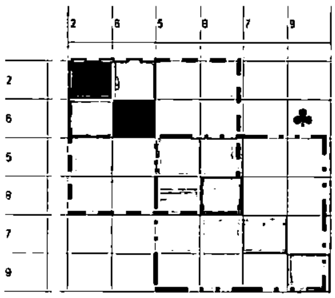


Figure 4-30 Enlargement of the lower group of the correlation grid of trial three data

From this figure, two main, overlapping groups are apparent. The top group consists of neurons 2, 6, 5 and 8; and the bottom group which consists of neurons 5, 8, 7 and 9. Recall from section 4.1.4, that overlapping groups tend to indicate different, connected hierarchies.

The interpretation of the bottom group is relatively straight-forward, it has a similar pattern and structure to the upper group of this trial. The bottom group depicts neuron 5, one level higher than neurons 7, 8 and 9.

The top group depicts neuron 2 connecting to both neurons 5 and 6. In addition, a connection exists from neuron 2 to neuron 8. It is likely that this is attributable to the connection from neuron 1 to neuron 8 via neuron 5. By examining the details of the cross-correlogram (not shown due to space limitations), between spike trains 2 and 8, it is possible to verify this hypothesis. The link between the top and bottom groups is clearly via neuron 5.

In addition to these relationships, the grid shows a second link between the two groups, from neuron 6 to neuron 9, indicated in Figure 4-30 by the following symbol (♣).

4.3.4 Summary of trial three observations

From this analysis, the overview and details of the correlation grid, it is possible to deduce the underlying neuronal assembly of the data set, as depicted in Figure 4-31.

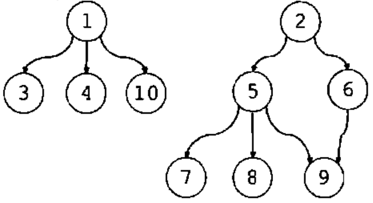


Figure 4-31 The neuronal assembly of trial three

5 Conclusions

This correlation grid, which is based on information visualisation, has proven to be an effective tool in supporting the study of synchronous spiking in multi-dimensional neuronal systems. This method has helped us to define the unknown structure of connections between neurons. At this stage, only a small number of simulated data sets have been analysed. However, these initial empirical studies have yielded successive correct connection architectures.

These empirical studies are continuing with a variety of datasets from larger and more diverse assemblies. The results from these studies will be published in the future.

6 Future Work

The Correlation Grid will be developed to support software functionality such as (i) greater direct user manipulation, (ii) the facility to "zoom" and "hide" data in the grid, (iii) the capability to vary the thresholds of "significance" which underlie the colour coding, (iv) encoding other information such as the number of peaks & peak delay in each individual cross correlogram and (v) multiple threshold colour coding will be introduced.

7 Acknowledgements

This research is supported by the Engineering and Physical Sciences Research Council (EPSRC) grant number GR/N04904. Additionally, the research of Roman Borisyuk is supported in part by the Russian Foundation of Basic Research (grant 03-04-48482).

8 References

Awiszus F., 1997. Spike train analysis. Journal of Neuroscience Methods, 74.

Borisyuk R. M., 2002. Oscillatory activity in the neural network of spiking elements. BioSystems, 67:3-16

Brillinger D. R., 1979. Confidence intervals for the crosscovariance function. Selecta Statistica Canadiana, V:1-16.

Gerstein G. L., Kirkland K. L., 2001. Neural assemblies: technical issues, analysis, and modelling. Neural Networks, 14(6-7):569-598

MuLab http://mulab:physiol.upenn.edu/index.html

Nase G., Singer W., Monyer H., Engel A. K., 2003. Features of neuronal synchrony in mouse visual cortex. Journal of Neurophysiology, August 90(2):1115-1123

Neuenschwander S., Castelo-Branco M., Baron J., Singer W., 2002. Feed-forward synchronization: propagation of temporal patterns along the retinothalamocortical pathway. Philos Trans R Soc Land B Boil Sci December, 29;357(1428):1869-1876

Schmidt M. F., 2003. Pattern of interhemispheric synchronization in HVc during singing correlates with key transitions in the song pattern. Journal of Neurophysiology, August 27

Walter M., Stuart L., Borisyuk R., 2003. Spike Train Correlation Visualization. Proceedings of the 7th IEEE International Conference on Information Visualization (IV03):555-560

The Representation of Neural Data using Visualization

Mr. Martin Walter* mwalter@plymouth.ac.uk

Dr. Liz Stuart* lstuart@plymouth.ac.uk

Prof. Roman Borisyuk[†‡] rborisyuk@plymouth.ac.uk

*School of Computing, Communications and Electronics, University of Plymouth, Plymouth, Devon, UK
[†]Plymouth Institute of Neuroscience, University of Plymouth, Plymouth, Devon, UK
[‡]Institute of Mathematical Problems in Biology, Russian Academy of Sciences, Pushchino, Moscow Region 142 290, Russia

Corresponding Author:

Mr Martin Walter

Room B310, Portland Square

School of Computing, Communications and Electronics

University of Plymouth

Plymouth

Devon

PL4 8AA

United Kingdom

Telephone: +44 (0) 1752 23 2620

Fax: +44 (0) 1752 23 2540

Email: mwalter@plymouth.ac.uk

Acknowledgments

This research is supported by the Engineering and Physical Sciences Research Council (EPSRC) grant number GR/N04904. Additionally, the research of Roman Borisyuk is supported in part by the Russian Foundation of Basic Research (grant 99-04-49112) and by EPSRC (grant GR/N63888/01).

Abstract

Currently, the focus of research within Information Visualization is steering towards genomic data visualization [27] due to the level of activity that the Human Genome Project [12] has generated. However, the Human Brian project [11], renowned within Neuroinformatics, is equally challenging and exciting. Its main aim is to increase current understanding of brain function such as memory, learning, attention, emotions and consciousness. It is understood that this task will require the "integration of information from the level of the gene to the level of behaviour".

The work presented in this paper focuses on the visualization of neural data. More specifically, the data being analyzed is multi-dimensional spike train data. Traditional methods, such as the 'raster plot' and the 'cross-correlogram', are still useful but they do not scale up for larger assemblies of neurons.

In this paper, a new innovative method called the Tunnel is defined. Its design is based on the principles of Information Visualization; overview the data, zoom and filter data, data details on demand [22]. The features of this visualization environment are described. This includes data filtering, navigation and a 'flat map' overview facility. Additionally, a 'coincidence overlay map' is presented. This map washes the Tunnel with colour, which encodes the coincidence of spikes.

Keywords

Information visualization, neural data, spikes, spike trains, analysis, visualization environment.

1 Introduction

Solution of many problems in the field of Neuroscience is associated with the theoretical comprehension of a large body of experimental neural data. Specifically, investigation of information processing in the nervous system is associated with the analysis of these vast quantities of neural data. More specifically, this data is simultaneously recorded multi-dimensional spike train data. Much of the research effort in this area is steered towards the principle of synchronization of neural activity [4][7].

The experimental evidence that is currently available requires further, in-depth analysis in order to extract inherent information. Analysis of neural data such as multi-dimensional spike trains using traditional tools, like raster plots and cross-correlograms, is increasingly complex due to the vast amount of data involved. Hence, new methods of analysing this data are required. Information Visualization is one of the fields of computer science that deals with innovations in the representation of vast quantities of data. This field is already recognised for its current and potential contribution to large scale projects such at the Human Genome Project [12] and the Human Brain Project [11]. Information Visualization has also led the development of many useful visual representations for hierarchical and temporal data. These visualizations include techniques such as treemaps [21], space filling visualizations based on radial layouts [23] and the use of helical structures [13]. Other fundamental techniques such as parallel coordinates [14], have already been successfully applied to the analysis of multi-dimensional spike train datasets [24], as well as the innovative use of other basic geometric primitives[26].

One of the fundamentals of the field of Information Visualization is the ability of the investigator to interact with the data being analyzed, in order to achieve greater insight. Thus, the investigator should be able to navigate throughout the data, to identify and explore specific subsets of interest. When visualising large datasets, the issue of efficient navigation is amplified. It is important for the user to be able to move quickly to points of interest without becoming disoriented within the data. By limiting the ways in which the user can navigate, this problem can be alleviated. The user can be constrained to follow predetermined paths throughout the data space. Subsequently, reversing or re-tracing your steps [17] becomes trivial. In addition, providing the user with different frames of reference can also help [20].

In addition to navigation capabilities, the investigator should also have control over the representation itself. Thus, in order to steer the analysis, they should be able to manipulate the data by applying appropriate techniques. For example, it may be appropriate to sort or organise the data using statistical or other mathematical routines.

Traditionally, analysis of multi-dimensional spike train data has not supported real-time user interaction. In 1996, Shneiderman [22] identified user interaction as one of the essential components of Information Visualization. Shneiderman also introduced the "information–seeking mantra" that highlighted user requirements in this area. This mantra specified that users should have the capability to overview data, zoom and filter this data and to obtain details-on-demand. This mantra was widely adopted throughout the Information Visualization community as a basis for defining user requirements. In many cases, different visualizations are utilised to represent data at differing levels of detail. Thus, resulting in the creation of a number of different views of the same data. Ideally, these multiple views should be linked for consistency [3][16][17].

In this paper, an interactive method for exploring neural data is presented. This representation of data is based on current Information Visualization and virtual reality principles. It supports multiple views of the neural data as well as real time interaction by means of a 'toolbox' that facilitates zooming, filtering and manipulation of the data.

2 Neural Data

There are many different types of neurons in the mammalian nervous system, each of which performs a different task. For example, excitation of motor neurons controls muscle fibres resulting in the contraction of muscles. These neurons communicate via relatively weak electrical impulses.

2.1 Spikes and Spike Trains

In general, a neuron accumulates electrical stimulus from other neurons coupled to it, until some internal threshold is reached. Once its threshold is reached, the neuron initiates an action potential. When a neuron initiates action potentials over time, we say that the neuron is firing. Note that action potentials are more commonly referred to as spikes and a series of these spikes over time is known as a spike train. Spike train data is one of the main types of data collected during neurophysiological experimentation. It is a record of the activity of a collection of neurons under investigation. Figure 1 shows a section, from 300ms to 800ms, of a typical spike train recording for three neurons. In this figure, a horizontal plot represents the spike train of each neuron. This horizontal plot denotes the occurrence of spikes, at specific times, by a vertical line.

It is well established that information is encoded in this data but it has also been established that each spike from a single neuron is identical [19]. Hence, the form of individual spikes is believed to carry little information. Instead, it is the spiking frequency and thus, inter-spike-intervals that carry information. Thus, research is focused on the analysis of multi-dimensional spike train data to reveal information about the synchronization of spike trains and the coupling of neurons.

2.2 Coupling

Direct Synaptic coupling and *Common Input* coupling are the general cases of coupling where synchronization may occur between the firing of two neurons.

Direct Synaptic coupling is illustrated in Figure 2 (i) where neuron A is coupled so that it stimulates neuron B. If neuron A fires, then neuron B has an increased probability of firing. Common Input coupling is illustrated in Figure 2 (ii), where neurons A and B are both coupled so that they receive stimulation from a third neuron C, resulting in the correlation of their input. Thus, if neuron C fires then both neurons A and B have increased probabilities of firing.

2.3 Multi-Dimensional Spike Train Data

One of the basic principles that underlie information processing in the brain is the principle of synchronization of neural activity. Research [4] indicates that the synchronization principle may be useful in devising various systems of information processing.

The current capability to record neural activity has led to the production of large quantities of experimental data. This data is in the form of multi-dimensional spike train recordings. Investigation of this data focuses on the synchrony between spike trains and the coupling of neurons.

Traditional methods of analysis are still employed by Neurophysiologists in the absence of more substantial software support. However, this type of analysis is both time consuming and complex due to the quantity of data currently available.

3 Traditional Methods of Analysis

A number of methods exist to analyze multi-dimensional spike train data. Two of the most commonly used methods are the *raster plot*, which directly plots the spike train data, and the *cross-correlogram* used to analyze the correspondence between spike trains. Most current methods are designed for use with a pair of neurons and do not scale-up to deal with larger numbers of neurons.

3.1 The Raster Plot

The raster plot [1] is one of the original methods for viewing and analysing spike train recordings. Each train is displayed as a line of dots, where each dot representing the presence of a spike at that time from the stimulus. Raster plots can be used to compare a number of recordings from the same neuron. This aids in the identification of similarities between these trials. In addition, raster plots can be used to view a number of spike trains from different neurons.

Figure 3 shows two raster plots each displaying spike train data from two neurons. From Figure 3 (i) it can be deduced that the spike trains of neurons a, and b, are synchronised and thus, neurons a, and b, are likely to be coupled. From Figure 3 (ii), note that the spike trains of neuron a and b are not obviously correlated and thus neurons a and b are less likely to be coupled.

3.2 The Cross-Correlogram

The cross-correlogram [1][15] quantifies the synchronization between the spike trains of two neurons. One spike train is designated to be the 'reference' train. The other is known as the 'target'. Due to the inherent delay in neural circuits, a time frame for correlation must be specified. This time frame, or correlation window, consists of a number of equal time segments, called 'bins'.

The correlation window is centred over the first spike of the reference train. The number of target train spikes that fall within each bin is calculated. This process is repeated for each subsequent spike in the reference train. The results of individual comparisons are summed up to give the overall correlation.

This overall correlation is then plotted as the 'cross-correlogram' (see Figure 4) for the two spike trains and shows the correlation of the target train with the reference train.

If the cross-correlogram has a significant peak [6] a correlation exists between the two trains. Consequently, it is likely that the two neurons are connected. In Figure 4 (i) the cross-correlogram of two spike trains, from connected neurons, clearly shows a significant peak. In contrast, noting the different scale, the cross-correlogram in Figure 4 (ii) shows no significant peaks, indicative of neurons that are not connected.

The cross-correlogram is a very useful and commonly used tool for representing the relationship between the spike trains of a pair of neurons. However, for any significantly sized neuronal assembly, numerous pair-wise results would be generated. This large quantity of resultant data poses its own analysis problem, as all pair-wise results must be analyzed to understand any relationship between the underlying neurons.

4 Current Methods of Analysis

Other innovative methods also exist for the analysis of the spike trains of large groups of interconnected neurons, also known as assemblies. One notable method is the 'Gravity Transform', originally developed by Gerstein and Aertsen [2][8][9]. The gravity transform algorithm can be used to study the dependencies in firing of multi-dimensional spike trains. Recent work by Stuart et al. [24] has enhanced the output from the original gravity transform algorithm using visualization techniques including parallel coordinates. However, much work is still needed in this area in order to fully support interactive exploration of these large multi-dimensional data sets.

5 An incremental approach to providing support

The ultimate system for this type of problem would be an intelligent adaptive system that built up knowledge and expertise based on experimentation that provided positive and negative feedback to the system. This ideal system would receive a dataset from the user as input and would produce an assembly, perhaps even two or three, with a certainty value or percentage associated with each topology suggested.

However, the main milestone separating the current support available for this type of analysis from this "ideal" system is a reliable human-centred approach that accurately specifies the topology of a network of neurons from a multidimensional spike train dataset.

This human-centred approach will not comprise a single, perfect representation that is suddenly discovered. On the contrary, this human-centred analysis approach will be a toolbox of many different representations, each with individual strengths and weaknesses. However, the combined functionality of this suite of tools will enable the experienced user to accurately identify the topology of a network of neurons from multi-dimensional spike train datasets as required by Neurophysiologists.

Hence, our work is focused on the design, implementation and testing of individual representations which, when coherently combined together, truly support the analysis of multi-dimensional spike train data. This approach is a unique and significant development in this area as it is based on the principles of Information Visualization and Software Engineering. This paper presents a new representation called the spike train 'Tunnel', that enables the user to investigate synchrony in the datasets.

6 The Tunnel Environment

This environment presents different views of the dataset and an additional overlay that encodes spike coincidence. It enables the user to focus on a specific subset of the dataset using a set of interaction tools. Different frames of reference are provided to enable investigators to track their location within the data space.

6.1 The Tunnel Visualization

The Tunnel is a cylindrical environment that supports user interaction. Figure 5 shows the Tunnel visualization of a randomly generated dataset over 200ms.

Each of the numbered horizontal bands, that comprise this Tunnel visualization, encodes the spike train of the corresponding neuron. The two 'end' bands, band 1 and band 10 in Figure 5, are adjacent to each other, thus forming the cylindrical environment. Note that, time is represented down through the Tunnel.

Overall, illumination inside the environment represents the firing of neurons in the currently displayed portion of the dataset. Synchrony is detected by perception of the position, intensity and frequency of light sources at different parts of the Tunnel.

In the Tunnel visualization, the investigator is able to 'fly' through the Tunnel to arrive at sections of the Tunnel (subsets of the data) that are of specific interest. The user has control over both, the speed and direction of the flight. However, to minimise the possible side-affects of disorientation during navigation, motion is restricted to being along the Tunnels length. Thus, the user is restricted to forward and reverse motion.

6.2 Filtering Data

To meet the user requirements previously discussed in section 1, the Tunnel has filtering functionality knowing as 'dimming'. Whilst in the filtering mode, dimming may be switched, on or off, for each of the spike trains individually. Figure 6 illustrates filtering of another 200ms dataset. In this dataset, the spike trains of neurons 4, 6 and 8 are identical. The remaining spike trains were all randomly generated. In this figure, all spike trains are dimmed with the exception of spike trains 4, 6 and 8. This filtering is designed to enable investigators to highlight spike trains of interest while maintaining context within the dataset.

6.3 Coincidence Sorting

As stated previously, in section 2.1, a key concept of spike train analysis is the identification of coincidence between spikes. Within the Tunnel environment, the investigator is able to progressively sort the order of the spike trains to view spike coincidence. The user selects a spike on a reference train. Subsequently, the spike trains in the Tunnel are reordered, so trains with spikes coincident to the selected spike are adjacent. Trains that do not have any coincident spikes are inherently moved away from the reference train.

To illustrate coincidence sorting, another 200ms dataset, based on an assembly of ten neurons, was generated. In this assembly, neurons three and ten fire every 12ms and neuron eight fires every 7ms. To illustrate the progressive sorting feature of the Tunnel, neurons four and six fire every 7ms and 12ms.

The Tunnel visualization of this unsorted dataset is illustrated in Figure 7, where the selected spike on the reference train is indicated by the arrow. Note the first coincident spikes (at 12ms) on spike train three and four, and the following non-coincident spike (at 14ms), solely on train four. Subsequent to sorting, spike trains four and eight are moved adjacent to the reference train, six, due to coincidence with the selected spike. This reordering is shown in Figure 8.

The Tunnel visualization provides a progressive sorting facility. When a successive sort is applied, the order of sorted spike trains is initially preserved. However, as this ordering is applied, any train with a spike correlated to the currently selected spike overrides this order. This supports 'fine-tuning' of the spike train order within the Tunnel.

As a result of this type of successive sort, highly correlated spike trains are nearer to each other. This is demonstrated in Figure 9. Spike train three, four and ten are near to train six due to their correlation with the currently selected spike. Note that, the previous ordering of train four adjacent to train six persists as it correlated to both the first and second selected spikes. Since train three only correlated to the currently selected spike, it cannot displace train four. In contrast, note that train ten displaces spike train eight.

6.4 Coincidence Summary

In addition to individual spike coincidences, the overall spike coincidence of the dataset is also of interest. Thus, the Coincidence Summary was developed. This representation derives a summary of neuron firing and colour codes this data.

Each spike train in the dataset is divided into a number of equal time slots, *n*, commonly referred to as "bins". The size of bin is specified by the user, but is usually relative to the neural transmission delay time.

Each spike train has an associated array made up of *n* elements, where each element of the array is associated with a bin. Each bin is inspected and if one, or more, spikes occur inside the bin, the corresponding element in the associated array is set equal to one, otherwise zero. Note, the total number of spikes in each bin is not important, but the presence, or absence, of a spike within that bin.

When all of the associated arrays have been calculated, an intermediate summary array, of *n* elements, is created. It is computed from the associated arrays, such that, the *i*th element of this intermediate summary array is equal to the sum of the *i*th element of each associated array. Subsequent to computation, if an element of the intermediate summary array is less than two, there is no spike coincidence. Thus, these elements are set equal to zero in the final summary array. Note that the maximum value of an element in this final summary array is equal to the total number of spike trains in the dataset. This array is used to create the Coincidence Summary Visualization.

The dataset used to create the visualization in Figure 11 comprised ten spike trains each lasting 200ms. Each of the spike trains were created by appending four, 50ms trains together, such that the first and third segments were low in spike frequency in comparison to the second and fourth segments. Thus, for this 200ms dataset, the final summary array is made up of 67 elements where there are sixty-six 3ms bins and one 2ms bin. Due to the way in which the dataset was generated, elements 1-17 and 35 –51 of the final summary array will be relatively low in value with respect to elements 18-34 and 52-67.

This data is encoded for the Coincidence Summary Visualization using colour based on the Hue map shown in Figure 10.

The Coincidence Summary Visualization (CSV) for the Tunnel is illustrated in Figure 11. Recall, that the labels (red) of the trains bear no relation to coincidence, they have been added to this paper for clarity.

Note that the majority of the visualization viewed at this position in the Tunnel primarily displays the first segment of the final summary array. Due to the relatively low values of the first segment, this is represented by different hues of blue.

It is possible to distinguish the second segment of the final summary array, at the centre of the visualization. Due to the relatively high values in this segment, it is represented by mainly red and yellow hues.

6.5 Combining them Together

In addition to viewing the CSV, it is also possible to superimpose the Tunnel visualization onto the CSV. An example of this is shown in Figure 12 where the data used to generate the CSV in Figure 11 has been superimposed by the corresponding Tunnel visualization of this dataset.

This combination of the CSV and the Tunnel visualization increases the complexity of the display but also helps identify coincidence between spike trains.

Note that in this combined visualization, the colour used to represent a spike is defined by the corresponding colour in the CSV. This relates to the overall firing activity at that time in the Tunnel.

6.6 The 'Flat Map' Representation

Investigators can also overview the data using the 'Flat Map' representation as shown in Figure 13. This is similar to the raster plot discussed in section 3.1 however, it has additional functionality. This functionality enables the user to select a 1000ms subset of the data for analysis. It is this selected 1000ms subset that is subsequently displayed in greater detail within the Tunnel representation.

To define the required subset the user interacts directly with the Flat Map resulting in a broad black "Boundary indicator" which defines the current portion to be viewed in detail in the tunnel. Note that this detailed viewing is a dynamic process, effectively the user flies through the segment. The "progression indicator" on the Flat Map indicates to the user their progression within the segment.

Figure 13, depicts the flat map for the demonstration dataset used in section 5.3. Although the tunnel is capable of displaying 1000ms at any given time, this demonstration dataset is only 200ms long in order to clarify the concepts presented. Thus, it is the majority of this 200ms dataset that is shown in the tunnel representation shown in Figure 7.

6.7 Undo/Redo Facility

In addition to user interaction and navigation, the Tunnel supports a comprehensive undo/redo facility. Shneiderman asserted that the ability to back track adaptation to the visualization was key to the refinement of understanding [22]. Thus, the user should be able to easily return to previous states of the visualization.

Towards this end, the environment tracks all changes to the spike train order enabling the user to selectively undo/redo refinements to aid understanding.

7 Empirical Testing

In this section, the results of two trials are presented. Each of the trials use a dataset that was generated using an advanced integrate and fire model of neurons with particular coupling between elements [4]. The parameters of this model were chosen to imitate the general neurophysiological characteristics of cortical neurons. All connection strengths in these simulations were chosen to be positive.

7.1 Trial One

In this trial, an assembly of 10 neurons, as shown in Figure 14, was simulated for a period of 20000 ms. The mean inter-spike interval (ISI) was 70ms, the standard deviation of the ISI of the dataset was 53ms and its coefficient of variation was 0.76.

7.1.1 Clustering

The random ordering of spike trains within the Tunnel poses a number of problems when attempting to analyze their interconnection(s). Thus, it is preferable to have trains with a high temporal correlation adjacent to each other. To achieve this, the stripes of the tunnel (representing each spike train) are reordered using a clustering algorithm.

A range of different mathematical clustering algorithms were analyzed to perform the clustering of the spike trains. These algorithms included a nearest neighbour algorithm (the minimum of measures between objects in two groups), a furthest neighbour algorithm (the maximum of measures between objects in two groups), and a centroid clustering algorithm [5]. The most suitable algorithm is the furthest neighbour method as this algorithm creates tight clusters between objects and all objects inside clusters have limited dissimilarity.

7.1.2 The smaller group of trial one

A Tunnel representation for the trial one data was compiled and subsequently sorted using the selected clustering algorithm. This is shown in Figure 15, in which the trains 1, 3, 5 and 7 have been highlighted, for the segment of the dataset between 3000ms and 4000ms. This is a useful feature of the tunnel environment.

Observe the simultaneous spiking activity on trains 3, 5 and 7. Further, note that these spikes are preceded by a spike on train 1. It was also possible to see this spiking pattern occurring further along the tunnel. Although it is not easy to see this from Figure 15, it is possible to see a pattern of spikes around the second and third spikes on train 1. Subsequently, it is reasonable to infer that a functional relationship exists between these neurons. Furthermore, since the spikes on train 1 precede those of the other trains, it is likely that neuron 1 is a common input to neurons 3, 5 and 7. This is confirmed by referring back to Figure 14.

7.1.3 The larger group of trial one

Figure 16 shows the same snapshot as Figure 15, however, in Figure 16 the spike trains 4, 6, 8, 9 and 10 are illuminated. Observe the simultaneous spiking activity on trains 6, 8, 9 and 10. Furthermore, note that these spikes are also preceded by a spike on train 4. Again, it is reasonable to infer that neurons 6, 8, 9 and 10 are connected and that neuron 4 is a common input to them. This is also confirmed by referring back to Figure 14.

7.1.4 Unconnected neuron

By observing train 2 along a number of sections in the tunnel, it is possible to see that there is no notable synchrony with any other trains. This is due to the fact that neuron 2 is unconnected in the assembly in Figure 14.

7.1.5 Summary

Overall the simple assembly depicted in Figure 14 can be deduced solely using the Tunnel representation.

7.2 Trial Two

In this trial, an assembly of 15 neurons, as shown in Figure 17, was simulated for a period of 20000 ms. The mean inter-spike interval (ISI) was 75ms, the standard deviation of the ISI of the dataset was 53ms and its coefficient of variation was 0.7. The ISI histogram for spike train 6 is shown in Figure 19(a) and its autocorrelation is shown in Figure 19(b). The raster plot of the first 3000ms of this dataset is also shown in Figure 18. The initial spike train tunnel for these 15 spike trains is shown in Figure 20, in which the order of the stripes is based on the default order of the spike trains in the data file. Recall that the user sees up to 1000ms of the dataset at a time in this representation, hence, Figure 20 displays a snapshot of the first 1000ms of the dataset. Note, that from this random ordering of the trains it is difficult to extract useful information. However, when viewing several snapshots, or 'flying' through this section of the tunnel, it is possible to identify similarity between trains.

Figure 21 shows the same snapshot of the Tunnel as Figure 20 except that in this figure the trains 5, 6, 9, and 11 are highlighted. Note the spike-pair, the first two spikes, on train 6. This spike-pair is followed by a similar spike-pair on trains 5, 9 and 11, each with different short delays. Thus, it is possible to infer that a relationship exists between spike trains 5, 6, 9 and 11. This synchrony of spiking is attributable to the connections from neuron 6 to neurons 5, 9 and 11; the lower sub-group in the assembly shown in Figure17. In Figure 21, the trains 5, 6, 9 and 11 have been highlighted to aid clarity in this diagram. Even though identification of relationships in this manner is possible, it is difficult and time consuming. The result of clustering the spike trains of this dataset, and subsequently, reordering the tunnel stripes, is shown in Figure 22.

7.2.1 The smaller group of trial two

Subsequent to clustering, it is easier to identify trains that are related. In Figure 23, spike trains 3, 7, 10 and 14 are highlighted. Note how the clustering algorithm has assigned trains 3, 7, 10 and 14 adjacent to one another. Also note the coincident spikes on all four trains, mid way down through the tunnel. Additionally, the occurrence of synchronous spikes on trains 3, 7, 10, 14, was observed in other parts of the tunnel. As these neurons tend to spike coincidently, it is reasonable to suggest that these neurons are all receiving a similar input. Further note that it is possible to see this synchronous firing is commonly preceded by the occurrence of a spike on train 1 as shown in Figure 23.

Indeed, from Figure 17, it is noted that spike trains 3, 7, 10 and 14 are generated by a group of common input neurons; namely neurons 3, 7, 10 and 14 which receive common input from neuron 1.

7.2.2 The larger group of trial two

In Figure 24, spike trains 6, 8 and 12 are highlighted to make it easier for the user to view the relationship between these trains. Closer examination also shows that this synchronous firing is commonly preceded by the occurrence of a spike on train 4. Thus, it is likely that neuron 4 is common input to neurons 6, 8 and 12.

In Figure 25, spike trains 5, 9 and 11 are highlighted and it is very clear that a relationship exists between these trains. From Figure 26, it is possible to note that the synchronous spiking in trains 5, 9 and 11 is preceded by a spike in train 6. As it is already known that neuron 4 is a common input to neurons 6, 8 and 12, it is deduced that this is a three level hierarchy of neurons.

Inspection of train 2 using Figure 26 and additional snapshots of the tunnel also revealed that a relationship exists between trains 2 and 4. Therefore, it is possible that neuron 2 connects to neuron 4 which subsequently connects to neurons 6, 8 and 12.

However, these relationships are not as clear from the Tunnel representation and it is likely that an additional representation, called the Correlation Grid [26] which is accurate in the identification of hierarchies, would be required to reinforce this aspect of the investigation.

7.2.3 Unconnected neurons

With the clustered order of the spike trains in the Tunnel representation, it is clear that spike trains 13 and 15 have no correlation with any of the other trains in this dataset. This is notable from the lack of temporal relationships shown for these spike trains in Figure 22, as opposed to the other groups.

7.2.4 Summary of trial two

From the clustered version of the Tunnel representation, it is possible to extract the information relating to the inter-neuron and inter-group relationships. However, it may be necessary, to support such an inferred definition of a neural assembly with additional evidence such as data from the correlation grid technique. This will become more apparent from subsequent empirical trials.

8 Future Work

The work presented in this paper is part of an Information Visualization project, at the Centre for Interactive Intelligent Systems, University of Plymouth, called Visualization of Inter-Spike Associations (VISA) [25]. Specifically, this paper has described an innovative visualization technique for the analysis of multi-dimensional spike train data called the Tunnel visualization. Testing of this representation to date has concentrated on datasets of approximately 20 spike trains over 20000ms. In these cases, the assemblies of the neurons that produced the spike train data were known to the investigator. This initial feasibility testing has been beneficial, reinforcing the efficacy of the method. However, this testing has also highlighted a number of limitations to the current version of the Tunnel representation. Each of these weaknesses is identified and future plans to strengthen these areas are described.

In this paper, all of the snapshots of the Tunnel have spike train numbers appended to aid identification. This is a trivial problem which will be addressed by simply developing a subtle labelling system that does not infer with data representation. A number of identification methods exist and currently two are under consideration. To identify a specific train, the user could "mouse over" the train to see its number. Alternatively, a semi transparent overlay of the train number could be developed. Moreover, it is likely to be more appropriate to implement both methods, allowing the user to select between them.

Currently, the 'flat map' overview of the Tunnel identifies the section of the data that the user has chosen to zoom in upon, but it does not track the user's position. This is not helpful to the user and thus, an exocentric frame of reference [20] will be developed for the Tunnel, to enable the user to identify their current position within the Tunnel.

Currently, the user may only select one, continuous, area of the tunnel to zoom. However, this representation does not provide the user with the facility to zoom in on a number of disjoint sections of the Tunnel, simultaneously. This is problem as the user is largely interested in synchrony and repeated patterns of activity within the whole dataset. Thus, the user may need to view and compare a number of sections (subsets of the whole dataset) of interest. An efficient method of shrinking areas of less interest in order to focus on areas of greater interest will be investigated.

Hitherto the weaknesses of the Tunnel representation described are changes which need to be implemented to improve the general use of this representation specifically.

However, there is a more significant issue that also needs to be addressed. It is understood that the human-centred analysis system, see section 5, that will provide the overall combined functionality required by Neurophysiologists, will comprise a range of representations. However, it also well established that the usefulness of multiple views is significantly increased if views are linked [16]. As the Tunnel representation is an integral part of the VISA system, its contribution to the overall analysis problem will be significantly advanced if it is used in connection with the other representations of VISA.

Thus, it is envisaged that future effort will be focused on meaningfully linking the Tunnel representation with other representations within the VISA system. In the future, it is likely that the VISA project will not only link representations together but it will provide a visual programming interface to enable users to “build” their own analyzes interactively.

Finally, with respect to empirical testing, further testing is currently underway to evaluate the usability of this visualization method with larger numbers of neurons. Subsequent to this, significant testing will begin on real data recorded by Neurophysiologists who are keen to collaborate on this project.

9 References

[1] Awiszus F. Spike train analysis. *Joumal of Neuroscience Methods* 1997; **74**:155-166.

[2] Baker, S.N., and Gerstein, G.L. Improvements to the Sensitivity of the Gravitational Clustering for Multiple Neuron Recordings. *Neural Computation* 2000; **12**: 2597-2620.

[3] Baldonado M. Q. W, Woodruff A., and Kuchinsky A. Guidelines for using multiple views in information visualization. *Advanced Visual Interfaces* 2000; 110-119.

[4] Borisyuk R.M., and Borisyuk G.N.. Information coding on the basis of synchronisation of neuronal activity. *BioSystems* 1997; **40**: 3-10.

[5] Bouguettaya A and Le Viet Q. Data Clustering Analysis in a Multidimensional Space, *Information Sciences* 1998; **112**:1-4: 267-295.

[6] Brillinger D. R. Confidence intervals for the crosscovariance function. *Selecta Statistica Canadiana* 1979; **V**:1-16.

[7] Fries P., Neuenschwander S., Engel A.K., Goebel R. and Singer W. Rapid feature selective neuronal synchronization through correlated latency shifting. *Nature Neuroscience* 2001; **4(2)**: 194-200.

[8] Gerstein, G.L. and Aertsen, A.M. Representation of Cooperative Firing Activity Among Simultaneously Recorded Neurons. *Joumal of Neurophysiology* 1985; **54(6)**: 1513-1528.

[9] Gerstein, G.L., Perkel, D.H., and Dayhoff, J.E. Cooperative Firing Activity in Simultaneously Recorded Populations of Neurons: Detection and Measurement. *Joumal of Neuroscience* 1985; **5(4)**: 881-889.

[10] Grün S., Diesmann M., Grammont F., Riehle A. and Aertsen A. Detecting unitary events without discretization of time. *Joumal of Neuroscience Emthods* 1999; **94**: 67-79.

[11] HBP Human brain project web site [WWW Document] http://www.nimh.nih.gov/neuroinformatics/index.cfm (accessed 04 August 2003).

[12] HGP Human genome project [WWW Document] http://www.ornl.gov/TechResources/Human_Genome/ (accessed 04 August 2003).

[13] Hicks M. *A Helix Metaphor for Customer Behaviour Visualisation*. IEEE International Conference on Information Visualization 2001 (London, UK, 2001), IEEE Computer Society Press; 22-28.

[14] Inselburg A. and Dimsdale B. *Parallel coordinates: A tool for visualising multidimensional geometry*. Conference on Visualization 1990: 361-378.

[15] MuLab [WWW Document] http://mulab.physiol.upenn.edu/index.html accessed 04 August 2003).

[16] North C. *Multiple views and tight coupling in visualization: A language, taxonomy, and system*. Workshop of fundamental Issues in Visualization 2001; 626-632

[17] Roberts J. C. *On encouraging multiple views for visualization*. IEEE International Conference on Information Visualization 1998 (London, UK, 1998), IEEE Computer Society Press: London; 8-14.

[18] Robertson G. G., Card S. K. and Mackinlay J. D. Information visualization using 3D interactive anaimation. *Communications of the ACM* 1993; **34(4)**: 59-71.

[19] Robinson D. (editor) *Neurobiology*. Springer, The Open University; 1998.

[20] Salzman M., Dede C., Loftin B. and Ash K. *VR's frames of reference: A visualization technique for mastering abstract information spaces*. Third International Conference on Learning Sciences 1998; 249-255.

[21] Shneiderman, B. Tree visualization with treemaps: a 2-d space-filling approach. *ACM Transactions on Graphics* 1991; **11:1**: 92-99.

[22] Shneiderman B, 1996. *The eyes have it: A task by data type taxonomy for information visualizations*, IEEE Symposium on Visual Languages 1996 (Las Alamos, USA, 1996), IEEE Computer Society Press: Chicago; 336-343.

[23] Stasko J. and Zhang E. *Focus+Contect Display and Navigation Techniques for Enhancing Radial, Space-Filling Hierarchy Visualizations*. IEEE Symposium on Information Visualization 2000, IEEE Computer Society Press; 57-65.

[24] Stuart L., Walter M. and Borisyuk R. Visualization of synchronous firing in multi-dimensional spike trains. *BioSystem* 2002 **67**: 265-279.

[25] Visualization of Inter-spike Association project website [WWW Document] http://www.plymouth.ac.uk/infovis (accessed 04 August 2003).

[26] Walter M., Stuart L. and Borisyuk R. 2003 *Spike Train Correlation Visualization*. IEEE International Conference on Information Visualization 2001 (London, UK, 2001), IEEE Computer Society Press; 555-560

[27] Wong P. C and Andrews K. *Proceeding of the IEEE Symposium on Information Visualization 2002*. IEEE Computer Society; 2002

[28] Zahn C. T. Graph-Theoretical Methods of Detecting and Describing Gestalt Clusters. *IEEE Transactions on Computers* 1971 **C-20:1**: pages 68-86

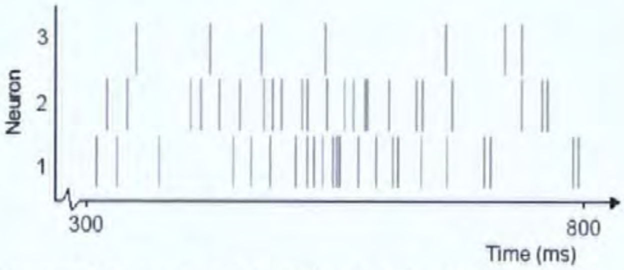


Figure 1: An example of a typical spike train recording for three neurons over a period of 500ms

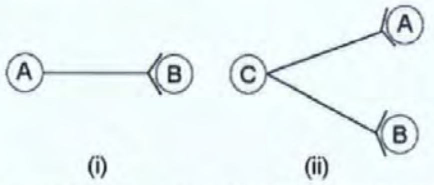


Figure 2: An example of (i) direct synaptic coupling and (ii) common input coupling

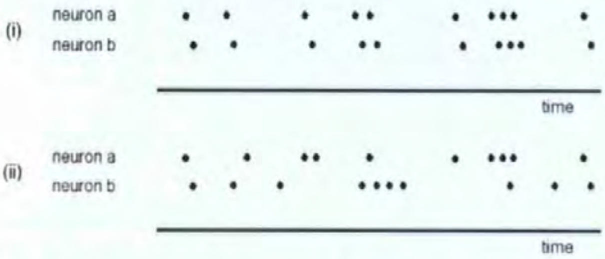


Figure 3: An example of two raster plots denoting (i) correlated spike trains, and (ii) spike trains with no apparent correlation

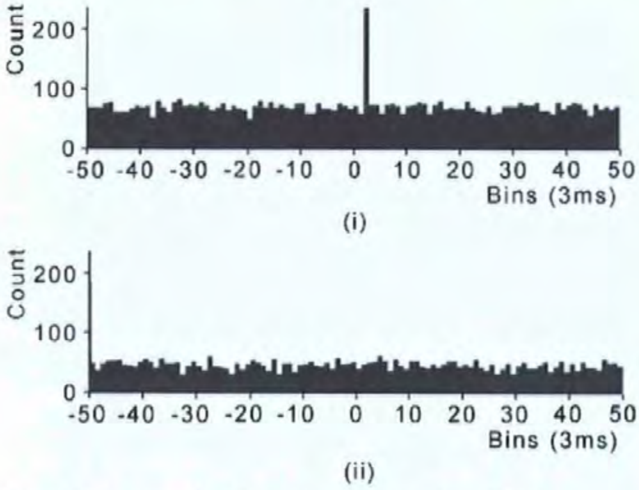


Figure 4: An example of a cross-correlogram from (i) two connected neurons and (ii) two detached neurons

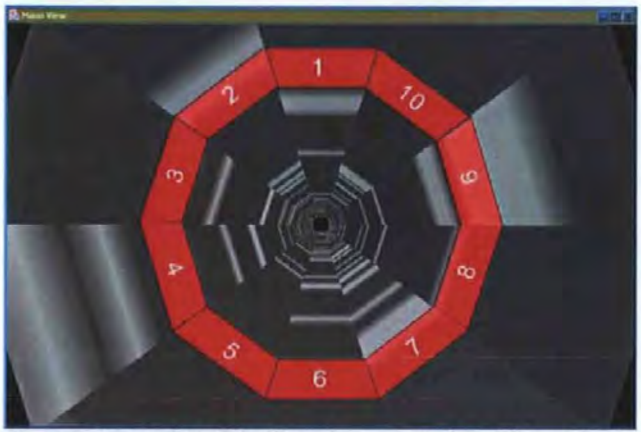


Figure 5: A snapshot of the Tunnel representation of the randomly generated dataset over 200ms

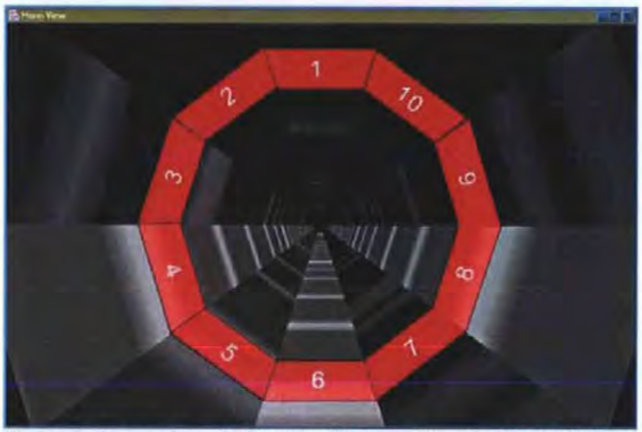


Figure 6: A snapshot of dimming (filtering functionality) within the Tunnel environment of a 200ms dataset

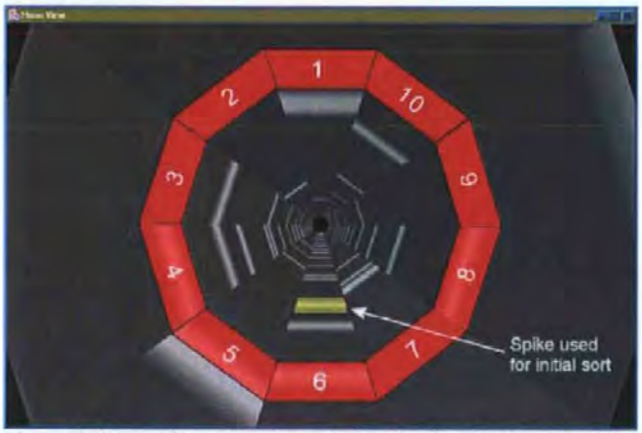


Figure 7: A snapshot of the Tunnel visualization for the unsorted dataset

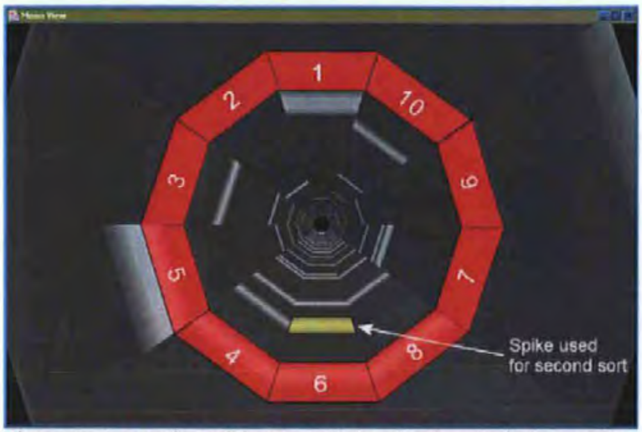


Figure 8: A snapshot of the Tunnel visualization depicting sorting

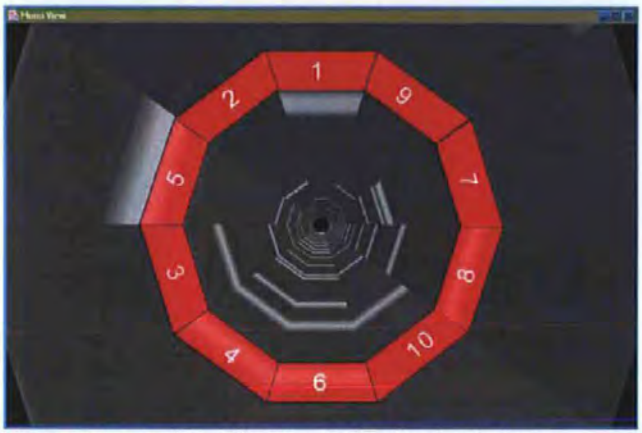


Figure 9: A snapshot of the Tunnel visualization depicting progressive sorting

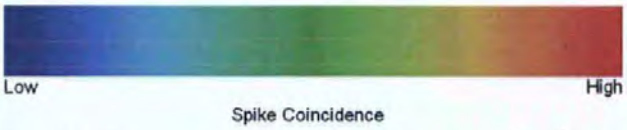


Figure 10: Colour coding for Coincidence Summary visualization

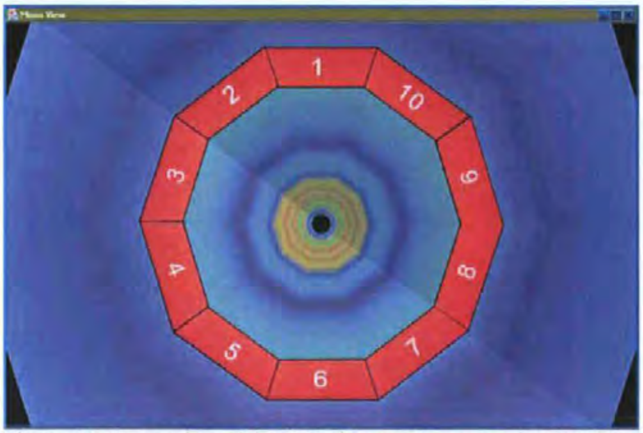


Figure 11: A snapshot of the Coincidence Summary visualization for a 200ms dataset

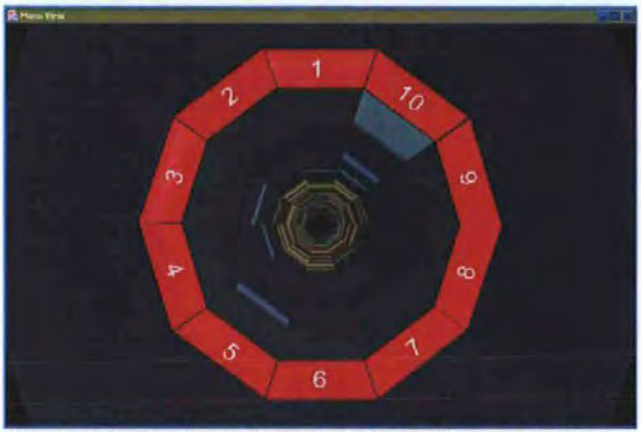


Figure 12: A snapshot of the Tunnel visualization superimposed onto the CSV

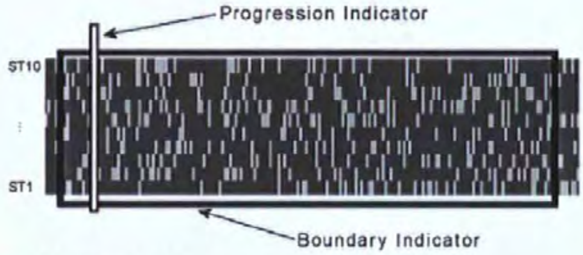


Figure 13: The Flat Map representation of the data segment depicted in the Tunnel representation of Figure 7

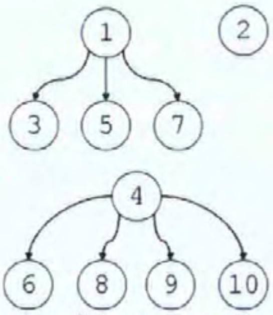


Figure 14: The assembly of ten neurons used to generate the spike train datasets used in trial one

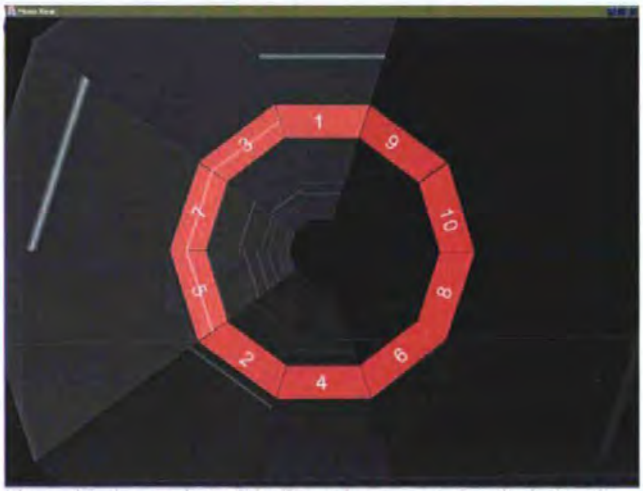


Figure 15: A snapshot of the Tunnel representation depicting the 10 spike trains of trial one, in which the order of the trains is

defined by the clustering algorithm and spike trains 1, 3, 5 and 7 are highlighted

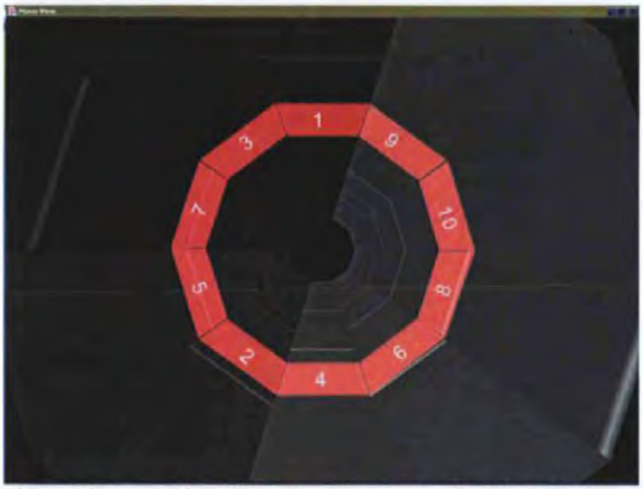


Figure 16: A snapshot of the Tunnel representation depicting the 10 spike trains of trial one, in which the order of the trains is defined by the clustering algorithm and spike trains 4, 6, 8, 9 and 10 are highlighted

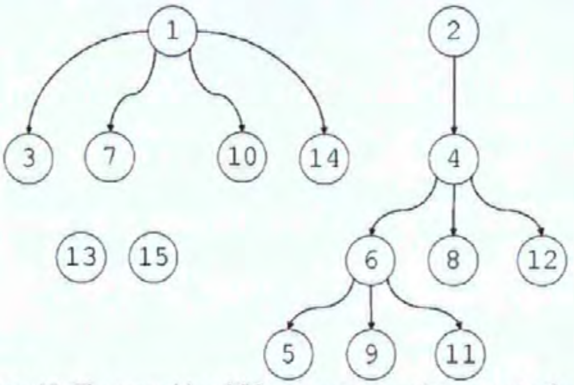


Figure 17: The assembly of fifteen neurons used to generate the spike train datasets used in trial two

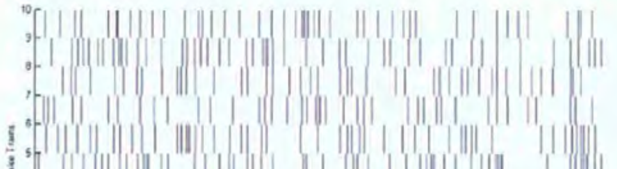
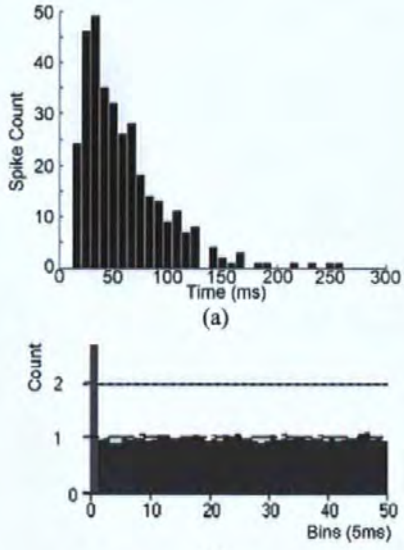


Figure 18: Raster A raster plot depicting the first 3000ms of the dataset generated for trial two



(a)

(b)

Figure 19: This figure depicts (a) the ISI histogram of spike train 6 and (b) the autocorrelation of spike train 6 from the dataset used in trial two

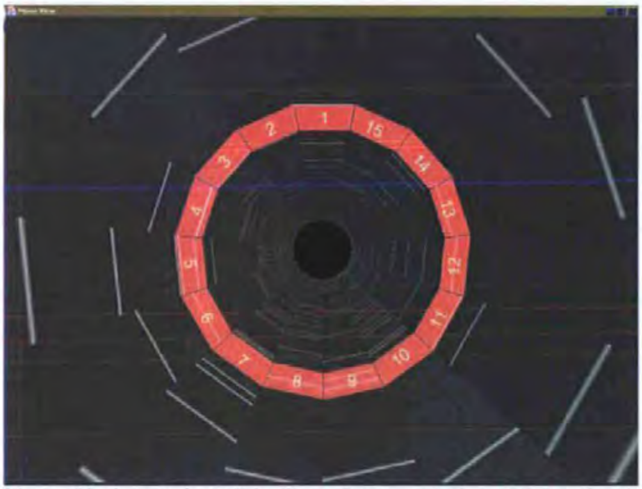


Figure 20: A snapshot of the Tunnel representation with 'random' ordering of the 15 spike trains of the second trial

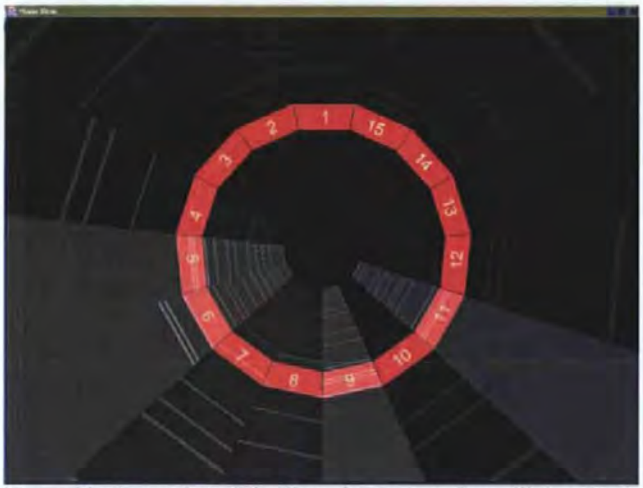


Figure 21: A snapshot of the Tunnel representation with 'random' ordering of the 15 spike trains of the second trial, in which trains 5, 6, 9 and 11 are highlighted

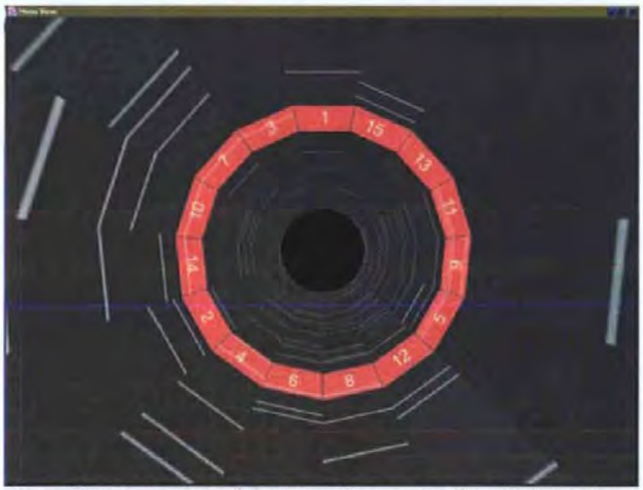


Figure 22: A snapshot of the Tunnel representation depicting the 15 spike trains of trial two, in which the order of the trains is defined by the clustering algorithm

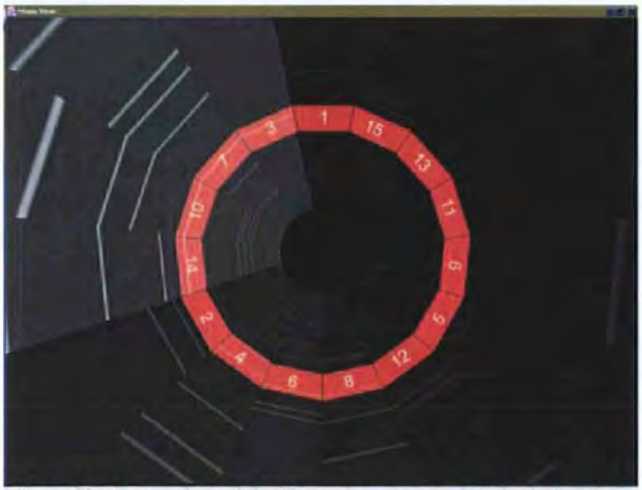


Figure 23: A snapshot of the Tunnel representation depicting the 15 spike trains of trial two, in which the order of the trains is defined by the clustering algorithm and spike trains 3, 7, 10 and 14 are highlighted

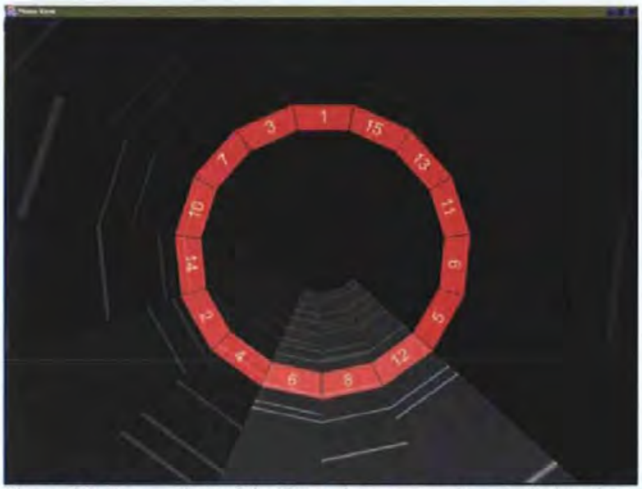


Figure 24: A snapshot of the Tunnel representation depicting the 15 spike trains of trial two, in which the order of the trains is defined by the clustering algorithm and spike trains 6, 8 and 12 are highlighted

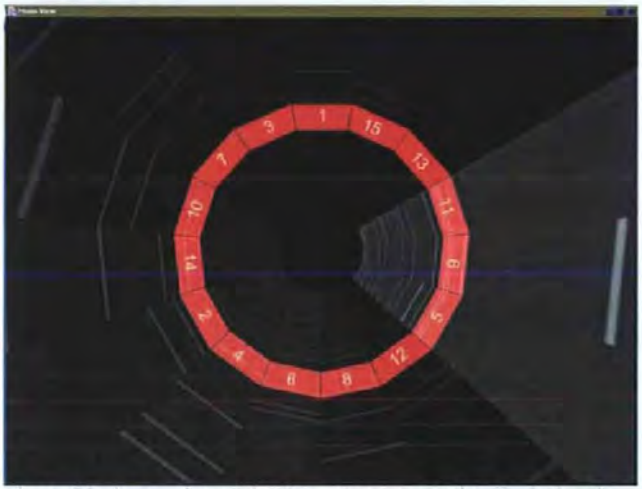


Figure 25: A snapshot of the Tunnel representation depicting the 15 spike trains of trial two, in which the order of the trains is defined by the clustering algorithm and spike trains 5, 9 and 11 are highlighted

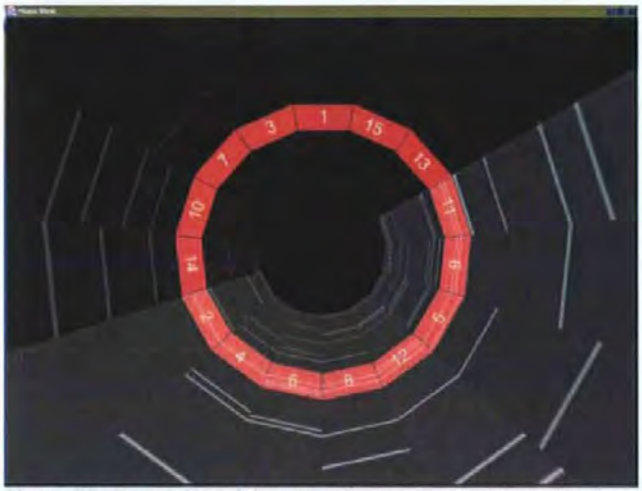


Figure 26: A snapshot of the Tunnel representation depicting the 15 spike trains of trial two, in which the order of the trains is

defined by the clustering algorithm and spike trains 2, 4, 6, 8, 12, 5, 9 and 11 are highlighted