Visualizing Confusion Matrices for Multidimensional Signal Detection Correlational Methods and Semantic Cluster Based Visualization in Virtual Environments

Yue Zhou
Wright State University

Follow this and additional works at: https://corescholar.libraries.wright.edu/etd_all

Part of the Computer Engineering Commons

Repository Citation
https://corescholar.libraries.wright.edu/etd_all/758
Visualizing Confusion Matrices for Multidimensional Signal Detection Correlational Methods and Semantic Cluster based Visualization in Virtual Environments

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Engineering

by

YUE ZHOU
M.S., Wright State University, 2013

2013
Wright State University

Thomas Wischgoll, Ph.D.
Thesis Director

Mateen Rizki, Ph.D.
Chair, Department of Computer Science and Engineering

Committee on Final Examination

Thomas Wischgoll, Ph.D.

Lang Hong, Ph.D.

Jack Jean, Ph.D.

R. William Ayres, Ph.D.
Dean, School of Graduate Studies
ABSTRACT


General Recognition Theory (GRT) is a multidimensional signal detection theory framework for capturing sources of perceptual and decisional dependence. The primary type of data for GRT models is an identification-confusion matrix derived in a complete factorial identification task. This confusion matrix plots the responses of study participant for a given signal. The responses may reveal that participants were unable to recognize the signal properly. Such violations of any type of independence in the GRT framework result in response patterns that reflect some form of correlation in the GRT space. While an individual confusion matrix is rather small and relatively easy to visualize, similar studies may be repeated a lot of times resulting in thousands if not millions of confusion matrices that have to be visualized. This thesis describes methods that adapt the web-based D3 visualization framework combined with pre-processing tools for the raw data to identify ways of visualization methodologies that enable domain specialists to more easily interpret their data. As the D3 framework utilizes Javascript and scalable vector graphics (SVG) to generate the visualizations it can run readily within the web browser to directly enable deployment of the visualization algorithms by the domain specialists. Parallel coordinate plots and heat maps were developed for the confusion matrix data, and the results were shown to a GRT expert for an informal evaluation of their utility. A second part is the visualization of database for semantic cluster which come from DBpedia. DBpedia is a crowd-sourced community effort to extract structured information from Wikipedia. Its database is served as Linked Data on the Wikipedia. VTK as a generic C++ class library has a wide range of scientific and engineering usage.
It can be used with another C++ based virtual environment library VRUI (virtual reality user interface). VTK and VRUI based semantic cluster database visualizations with different layout approaches were developed. There is a clear benefit to model interpretation from these visualizations when researchers need to interpret larger amounts of simulated data.
## Contents

1. **Introduction** ......................................................... 1

2. **Related work** .......................................................... 3

   2.1 Multidimensional Information Visualization ........................................ 3

   2.2 Parallel Coordinates ........................................................................ 6

   2.3 Heat Map ..................................................................................... 8

   2.4 Data Visualization by D3 ............................................................... 10

   2.5 Visualization of hierarchical information ........................................... 12

   2.6 Document and text information visualization .................................. 13

   2.7 Web Visualization ......................................................................... 15

   2.8 VTK ............................................................................................ 16

3. **Methodology** ............................................................... 18

   3.1 Motivation .................................................................................... 18

   3.2 Data structures ............................................................................. 21

   3.3 Step of generating Parallel Coordinates Data .................................... 27

   3.4 Step of Generating Heat-map data .................................................. 32
3.5 Data visualizations .................................................................................................................. 41

4. Results and discussion .................................................................................................................. 45

4.1 Parallel coordinates ................................................................................................................. 45

4.2 Heat map visualization .............................................................................................................. 47

4.3 Semantic cluster based visualization .......................................................................................... 52

5. Conclusions .................................................................................................................................. 61

6. Future works ................................................................................................................................. 63

Bibliography ..................................................................................................................................... 64
# List of Figures

2.1 “Worlds within worlds” coordinate system .................................................4

2.2 An example of parallel coordinates ................................................................6

2.3 Heat map with two main different colors .......................................................8

2.4 Web-based JavaScript library D3 ....................................................................10

3.1 Sample data file for model ...........................................................................24

3.2 Dictionary structure ....................................................................................25

3.3 Some active content has been disabled .........................................................27

3.4 Some content need to be enabled ..................................................................27

3.5 The Macros option .......................................................................................28

3.6 Edit the parallel macro ................................................................................28

3.7 Locate the file path ......................................................................................29

3.8 Generate the data .......................................................................................29

3.9 Locate the data file ......................................................................................30

3.10 Copy the html files ...................................................................................30

3.11 Use the html file to see parallel coordinates .............................................31

3.12 Consolidation program ...............................................................................32
3.13 Use the command to do the consolidation ....................................................33
3.14 The consolidation has been finished ............................................................33
3.15 Locate the file with all matrixes .................................................................34
3.16 VBA programs ............................................................................................34
3.17 Get the data from a file .................................................................................35
3.18 The break lines .............................................................................................35
3.19 Sort the data ..................................................................................................36
3.20 Deleting of unused row ...............................................................................36
3.21 The Macros option .......................................................................................37
3.22 The heatmap Macro ......................................................................................37
3.23 Locate the output file ....................................................................................38
3.24 Generate the data ........................................................................................38
3.25 The generated data file ...............................................................................39
3.26 Words replacement ......................................................................................39
3.27 Words replacement 2 ..................................................................................40
3.28 The html files ...............................................................................................40
3.29 Add the generated data ...............................................................................40
3.30 Check for the last data ................................................................................41
4.1 Parallel coordinate plot with 9 different GRT model simulation datasets. ............46

4.2 Layout example of heat map .................................................................................48

4.3 Heat map layout in which values are lumped by their position within the confusion matrix. ..................................................................................................................49

4.4 The same data set as shown as in Figure 4.2 using a circular layout for a heat map. 50

4.5 Circular layout showing the same data set as Figure 4.3, lower right. ..............51

4.6 Random position nodes ...........................................................................................53

4.7 Nodes start to move using force directed ..............................................................54

4.8 Become a force directed graph ..............................................................................55

4.9 Rotation to a better direction ..................................................................................56

4.10 Fast algorithms .......................................................................................................57

4.11 Simple algorithms ..................................................................................................57

4.12 Force directed algorithms ......................................................................................58

4.13 Clustering algorithms ...........................................................................................58

4.14 Span tree algorithms .............................................................................................59

4.15 Zoom out only shows the nearest labels ...............................................................59

4.16 Zoom in shows details ...........................................................................................60

6.1 Heat maps of the same data-set with Figure 4.3 (lower right) .........................63
Acknowledgment

This thesis would not have been possible without the help of many other people. At first I express my thanks to my advisor Dr. Thomas Wischgoll for giving me this opportunity and support to work on this project. I also would like to thank Dr. Lang Hong, and Dr. Jack Jean for serving on the committee and advice. Thanks Dr. Leslie M. Blaha, Dr. Ross Smith, and Dr. Rhonda J. Vickery for working on the same project. Also thank to Dr. Pascal Hitzler for the help and data. Thank you to Dr. Mateen Rizki, Chair for the Department of Computer Science and Engineering and the department faculty and staff for providing me the opportunity to complete my degree at Wright State University.
1. INTRODUCTION

Visualization is the process of transferring data, information, and knowledge into a visual form that takes advantage of people's natural ability to quickly identify visual changes. Visualization combines the capabilities of the human brain and the most powerful modern computer information processing system to form a powerful symbiosis. Using an appropriate visual interface allows us to understand and easily interact with large-scale data and observe differences while exploring information. When applying the right technique it can be extremely effective in detecting hidden information inside the characteristics of a regular pattern. In our increasingly information-rich life, visualization technology research and application development has fundamentally changed the way we express and understand large, complex data.

Information visualization on the other hand combines scientific visualization, human-computer interaction, data analysis, image technology, graphics technology, cognitive science, and many other disciplines, theories and methods. The ability to interact with the visualized data is particularly important in this case.

In this thesis, information visualization principles are applied to so-called identification-confusion matrix (IDCM) from the area of General Recognition Theory (GRT) which summarizes the frequency with which participants of a particular study respond to a set of specific stimuli. Traditionally, these matrices are visualized using four elliptical representations divided by lines to capture the results of the experiments. While being well-accepted by domain specialist, these representations are successful in providing an over-
view of the experiment. However, they are not capable of showing the entire detail present in the data. Especially with large series of experiments, the data can result in millions of IDCMs that need to be visualized. Hence, tools are needed that encode as much detail as possible of the original data within the visualization. This thesis describes the application of different information visualization techniques to such large number of IDCMs to help domain specialists to better and more easily analyze their data. These techniques were specifically adapted and expanded to improve the visualization of the data at hand.
2. RELATED WORK

Information visualization is a relatively new area of computer science. Many approaches have successfully been applied to numerous of data sets. The following subsections describe some of these techniques.

2.1 Multidimensional Information Visualization

We live in a world of three-dimensional physical space. Our visual perception is difficult from the front, left, up and down, the three-dimensional space-set. The fact that our surrounding is three-dimensional makes it difficult to intuitively understand higher-dimensional information. The vast majority of abstract information is more than three-dimensional information, such as financial information, stock information, and databases. Therefore, the visualization of multidimensional information is an important goal of information visualization.

We were able to show on the screen spatial information in two-dimensional, because our visual habits have left in our minds a three-dimensional space of the mark. Can we say that we can only see the three-dimensional space? Then how does three-dimensional visualization of multidimensional information work above it?
Figure 2.1: “Worlds within worlds” coordinate system

Feiner and Beshers present a multi-dimensional visualization with coordinates nested called “worlds within worlds” [1]. Input variables are mapped to a plurality of axes of the coordinate system. The high-dimensional function by a coordinate system nested to another coordinate system to be visualized. The mobile of internal coordinate system in the external coordinate system can cause changes in the surface. So at this time, the surface’s three variables (respectively by the three axes of the external coordinate system represented) have been changed. When the two coordinate systems in the same location, it will result in a nested closure. In this method, the virtual reality technology for manipulating visual structure has been used to reduce closures. This approach represents a very limited information dimension. Also the coordinate system in order to eliminate visual clutter caused due to nesting must use a complex interaction. So this technology is very difficult to use.
For information in the data base and analytics, large-scale data table representation is a key issue. Basically large data in the database is multidimensional. How to analyze multidimensional data sheet’s implied characteristics and the relationship between data in an intuitive way is crucial for information analytics. Xerox Palo Alto Research Center user interface Study Group, Rao and Card proposed table lens technology. It is a visualization and understanding of a large-scale data sheets technology [2]. This technology put the symbols and graphics into a single steerable focus + context layout. Some simple operations (such as sorting) are supporting a large data space browsing, and can be easily separated focus characteristics or patterns. It combines statistical analysis and the easy use of spreadsheet. It is suitable for large-scale data analysis areas, such as financial data, insurance data and drug analysis. It should be noted that table lens is a good visualization tool for multidimensional information. At present, there is a corresponding product based on this technology by Inxight Software.
2.2 Parallel Coordinates

Parallel coordinates are one of the first two-dimensional visualization techniques for high-dimensional data [3]. The basic idea is to put $N$ dimensional data of the attribute space by mapping $N$ equidistant parallel axis to the two-dimensional space. The axis of each dimension represents a property. The axis of the corresponding attribute values range from minimum to maximum uniform distribution. Thus, each data item can be used in accordance with its property values in a line segment on $N$ parallel axes. Parallel coordinates have been proven to be a powerful visualization tool. But when large data sets are visualized using parallel coordinates, it can cause confusion due to large numbers of lines overlapping. To improve on this issue, many parallel coordinate improvements were suggested in the literature. For example, Peng defines visual clutter as any aspect of the visu-
alization that interferes with the viewer's understanding of the data, and presents the concept of clutter-based dimension reordering. Dimension order is a technique that can significantly affect the visualization’s expressiveness. By varying the dimension order in a display, it is possible to reduce clutter without reducing information content or modifying the data in any way[4]. Hauser et al. present angular brushing for parallel coordinates as a new approach to highlighting rational data-properties, i.e., features which - in a non-separable way - depend on two data dimensions. It also demonstrate smooth brushing as an intuitive tool for specifying non-binary degree-of-interest functions (for focus+context visualization)[5]. Siirtola introduced two web browser based novel techniques to manipulate parallel coordinates. The first technique, polyline averaging, provides the capability of dynamically summarizing a set of polylines and can hence replace computationally much more demanding methods. The other new technique interactively visualizes correlation coefficients between polyline subsets, helping the user to discover new information in the data set[6]. Zhao et al. propose a technique based on parallel coordinates visualization that utilizes an edit-distance-based technique to rearrange variables so that interesting patterns can be easily detected. This system, V-Miner, includes both automated methods for visualizing common patterns and a query tool that enables the user to describe specific target patterns to be mined/displayed by the system [7]. Johansson et al. introduced a method to allow the user to simultaneously examine the relationships of a single dimension with many others in the data. The single dimension can then be interactively changed to allow the user to quickly examine all possible combinations. This method is achieved by extending the standard parallel coordinate approach to a 3D clustered multi-relational parallel coordinate representation [8].
Another simple, but very effective information visualization tool is a heat map. The basic idea of a heat map is essentially a two-dimensional visualization that utilizes a palette of a predefined color gradient to encode the individual data values by color mapping. For example, the closer the data value gets to a range associated with hot, the brighter the color used to represent that data value [9]. A single data point is rendered with a single color value from the color palette. However, using these colors directly to draw a large number of points can cause discretization artifacts. To avoid these artifacts, Dylan Vester proposes a method based on the superposition of gray and the color will gradually change, and finally fade to black. It starts with a gray canvas, and then takes the entire canvas as a bitmap. At last it maps the 256 gray scale map to a 256-color palette to generate the heat map [10]. Nierman et al. introduced Genes with significantly differentiated...
expression grouped into ten clusters. Three clusters of interest are shown with centroid graphs and heat map. Moon et al. used heat maps to visualize altered concentrations of urinary steroids found and heat maps produced from a 70-compound study that showed differences between the ratios of steroid precursors and their metabolites [11]. Špakov and Miniotas present another version of a heat map visualization technique. They propose a technique to facilitate visualization of eye gaze data gathered during both basic experimental studies on eye movements and usability studies on products and displays. The technique is an extension of the heat-map based visualization method [12]. Merico, D. et al used heat-maps to display gene-set expression patterns in the estrogen treatment experiment. The heat map view in their software (Enrichment Map) enables the user to zoom in and explore an enriched gene-set in more detail and the query set analysis facilitates exploration of relations to known disease genes or regulatory modules.
2.4 Data Visualization by D3

The main purpose of data visualization is effective and clearly information delivery by graphical data. Data visualization does not have to be a boring function or design that turns the data into a functional diagram, nor is it meant to be designed for an aesthetic form and dazzling gorgeous screen. In order to effectively convey information, aesthetic form and function needs go hand in hand through the vast amounts of complex information for analysis in a very intuitive visual means of expression. However, designers are often unable to obtain aesthetic balance between form and function to create some gor-
geous visual charts. These are the expense of their primary purpose - effective transfer of information.

Data visualization as a means to convey information, in addition to the use of a chart as visual form of expression, but also can add art and design through visual elements to more clearly communicate information to the audience. In the description on the basis of objective information to convey the author's while feeling and combine with the movement in the form of concept or image of the data. For this, Michael Bostock et al. proposed a web-based JavaScript library called D3 (Data-Driven Documents) [13]. When people use this kind of web-based visualization as a means of communication, it not only changed the design semantics, but also broaden the possibilities of bar, charts and diagrams. Michael Bostock et al developed eight kinds of data visualization techniques within D3: calendar view, chord diagram, choropleth map, hierarchical edge bundling, scatterplot matrix, grouped & stacked bars, force-directed graph clusters and Voronoi tessellation. In addition to the basic library, they also provided numerous sample programs on D3’s website. In August 2012, The New York Times started to use D3’s visualization for many articles [14]. Also Scott Murray’s book “Interactive Data Visualization for the Web” [15] describes several expansions of his works. For example, a Downton Abbey-inspired lorem ipsum text generator, which can generate complete, grammatically correct sentences; an interactive data map to communicate the results of the 2012 Global Peace Index, an annual measurement of levels of 158 countries’ peacefulness; an interactive data map to communicate the results of the Global Terrorism Index, and a new measurement of the impact of terrorism in 158 countries.
D3 as a web-based data visualization library is closely related to information graphics, information visualization, scientific visualization and statistical graphics. So visualization by using D3 can be a very active and important subject in the area of research, education, and product development.

2.5 Visualization of hierarchical information

Between the abstract information the most common type is the hierarchical relationships, such as a HDD directory structure, document management, library classification and etc... Hierarchical relationships almost everywhere, also in some cases, arbitrary graph can be transformed into hierarchical relationships [16]. The most intuitive way in visualization of hierarchical information is a tree structure. However, the traditional tree structure has a significant disadvantage: when increasing the structure or increasing the level of nodes, the structure needs to occupy a lot of visual space. The visual space on a computer screen is very limited, so the user must scroll through the way to grasp the entire hierarchy, which find a node or to obtain information on the entire structure is very difficult.

Different tree layout and visualization algorithms are available. George G. Robertson et al. proposed a technique called cone tree method [17]. Cone tree the hierarchy are arranged in three-dimensional space equalizer. Placed on top of the hierarchy of the top of the visualization space, each vertex cone apex represents the layer structure. Its child nodes (D) are uniformly arranged in the bottom of the cone. The cone diameter of the bottom of the hierarchy is gradually reduced with increasing depth in order to ensure that the bottom of the structure in the visualization space can be effectively represented. Between each cone transparent cover, which can ensure that each cone can easily be per-
ceived, it will not interfere with the back cone. Meanwhile accompanied by rotation, drag, etc. to facilitate interaction techniques, people can easily achieve the grasp of complex hierarchical relationships.

Xerox Palo Alto Research Center User Interface Research Group John Lamping, who proposed an approach based on hyperbolic geometry visualize and manipulate large hierarchies of focus + context technique, called Hyperbolic tree [18]. This technology will allow more space for the visualization of the current hierarchy part of current interest, and at the same time to display the entire hierarchy. The technology will be a specification of the algorithm display in a hierarchical relationship hyperbolic plane, and then the hyperbolic plane will be mapped to the display area. The selected mapping provides a smooth transition between a fisheye distortion focus and context. Hyperbolic tree through convenient interactive tools solves the problem of smooth transition between hierarchy focus and context. At present, this technology has been mature products from Inxight Software Company, for hard disk directories, site structure, electronic library catalogs and other related visualization.

2.6 Document and text information visualization

People are faced with a lot of information; the vast majority is text information. Such as email, Internet documents, scientific papers, newspaper, articles and so forth. Document information is an extension of our memory; people need to communicate regularly with document information. A variety of document information have large amount, visualization can help us quickly get the information from the document content and knowledge which people need. Document information visualization can be divided into two catego-
ries: one is the visualization of a single document itself, and the other is a large collection of documents visualization.

Stephen G. Eick et al. introduced Seesoft, a visualization system for visualizing software source code [19]. The approach utilizes the length of each line of code, which is mapped to a short straight line. People can achieve simultaneous analysis of up to 50,000 lines of code. The color of each term can be used to express some concern the statistical characteristics, such as the red is recently modified code, blue indicates the least recently modified code, and so on. Seesoft can be used to visualize a variety of data sources. For example, a version control system that can track versions time, programmers, code, purpose, and etc. Static analysis includes the location of the function call. Dynamic analysis includes the characteristics of the data. The use of flexible interaction technology, users can easily manipulate a simplified representation of the code to discover features of interest. A further detailed observation of an additional window displays the actual code, which the user can implement based on that piece of code. Seesoft can be used for knowledge discovery, project management, code tuning, development methodology analysis and many other areas.

For large document collection, the topic or content between documents relevant for users is very important. For example, we search for information on the Internet, it is so important to quickly grasp the thousands of search results and the search criteria. Also the correlations between the searches and the results can help people to quickly find the information you really need. PNNL (Pacific Northwest National Laboratory) scientists made a series of information access and visualization analysis tools. These technologies are collectively known as SPIRE (Spatial Paradigm for Information Retrieval and Explo-
ration). SPIRE can be used for almost any type of large document collections to determine the relationships between documents, and display them as a very natural visualization form for humans [20]. For example, James A. Wise, proposed a large collection of documents on the relationship between the methods visualization called “Themespace” [21]. In Themespace, the document topic is displayed on the computer screen as a natural topographic map. Themespace the peaks imply the dominant topic, and the valley says the part of topic is relatively weak. The mountain, valley shape, large range of hills, or high peaks, shows that topic as well as how information is distributed between documents and the association. This visualization is a way to avoid language processing and saving the user's mental work. So it for the information retrieval and knowledge processing is very useful.

2.7 Web Visualization

It can be stated that the information explosion was triggered by the Internet. Currently the information size of each website may more than TB, and the information distributed throughout the world's millions of different websites. The website through hyperlinks between documents intertwined with each other. Also no matter how large the WWW (World Wide Web) is, one thing is for sure, it will continue to expand faster and faster. How to easily use information on the website has become an urgent problem to be solved. However, the current information access method is far from satisfactory. Information Visualization in helping people understand the structure of information space, quickly find the required information and other aspects will play an important role in convincing.
HTML is only one component of WWW; therefore a single website in the Web visualization occupies an important position. Xerox Palo Alto Research Center, User Interface Research Group Ed H. Chi et al proposed a visualization method of the changing in the website [22]. It put Xerox website’s more than 7000 nodes re-organized into a tree. When considering the time range in each node of the tree has a corresponding position. The color and thickness of each node connection is according to the decision of visits. Different site history page is using different tree representation. This makes it easy to find web content changes and access volume changes.

The web is an information space; how to visualize its structure is the most important task. Web space structure is actually a network, the current research in this area focus on how to effectively visualize the information space of the network structure.

2.8 VTK

VTK (Visualization ToolKit) is an open source and free firmware. In the world's thousands of researchers and developers are using it for 3D computer graphics, image processing, and visualization. VTK includes a c + + class libraries, numerous interface layers, including Tcl, Tk, Java, Python. Visualization Toolkit is an application for visualizing structure and operation support environment. It is based on the 3D OpenGL library and provides an object-oriented design method [23]. It will shield the visual development process encounter details and put some commonly used algorithms package together. For example Visualization Toolkit surface reconstruction of the common Marching Cubes algorithm package into the form of class to support the user, so that user in the three-dimensional surface reconstruction lattice data will not have to repeat the writing of
MarchingCubes algorithm code. The user can directly use the VTK-provided vtkMarchingCubes classes. VTK is engaging visualization application development researchers to provide direct technical support to develop a powerful visualization tool. It has the ease of use and flexibility as the main principle and the following characteristics:

(1) A strong three-dimensional graphics functions. Visualization Toolkit supports both voxel-based rendering also retains the traditional surface rendering, resulting in greatly improved visual effects at the same time can make full using of existing graphics library and graphics hardware.

(2) Its architecture has a very good flow streaming and caching ability in handling large amounts of data without worrying about the memory resource constraints.

(3) It support Web-based tools such as Java and VRML. As Web and Internet technology development it has great prospects for development.

(4) Capable of supporting multiple coloring.

(5) A device-independent code so that it has good portability.

(6) It defines a number of macros that greatly simplifies the programming effort and reinforce a consistent object behavior.

(7) Richer data types, support for processing a variety of data types.

(8) It can work in both the Windows operating system and the Unix operating system which greatly broadens its user base.
3. METHODOLOGY

3.1 Motivation

The goal of the visualization process is for the data to be presented in a form (ex. an adjustable map) that is optimized for the human visual perception system. This process resembles a standard information visualization pipeline. The pipeline describes the transition from the original data to the final representation. The data may need to undergo a series of data conversions first. The process of converting the original data to the final visualization is likely to be expressed in a series of transformations. Each transformation may include a user operation to fine-tune these transformations. Here is an example: the data is a conversion of the original data with associations to a data table with data correlation descriptions. Visual mapping of that data table into a visual structure is a combination of space-based marking and graphical attributes. View transformations defining the position, scale, cropping and other graphical parameters create a visual view of the data representation. The user can interactively control the operational parameters of the transformations within the visualization pipeline, for example, binding to a specific view of the range of data, or change the attributes of the transformation. Information visualization paradigms are applied to solve the main problem of mapping, conversion, and interactive control.

In this information visualization pipeline model, the core of the pipeline is the data structure table for the visualization mapping. The data structure tables are based on mathemat-
ical relationships, whereas the visualization structure is based on being able to deal effectively with the visual attributes. Scientific visualization focuses on real data sets, for example from specific experiments or simulations, whereas information visualization research generally focuses on more abstract information. In many cases, the information itself cannot automatically be mapped to geometry or physical space. This means that typically different types of information with no natural or obvious physical representation are common place. Therefore, a key question is to discover appropriate visual metaphors and structure to represent the information, combined with an understanding of the information supported by an analysis tasks. In visualization there are three main components: special grouping, annotations, and graphical properties of annotations. In the visualization process, the data table is mapped into a visual representation. Also visual representation is encoded by graphical properties and annotations. In order to get a good visual representation, the mapping of the data onto visual elements has to preserve the original information and only the data in the table are displayed within the visual representation. To find a good mapping is not an easy task. The visual structure is prone to contain less important data, and the data encoded in the visual representation needs to be easily comprehensible by the user. When using a mapping that cannot be understood as easily, misinterpretations of the results can be caused. Hence, finding a good visualization of the data is a key issue for information visualization.

View transformations interactively change the view onto the visual representation. Through the use of graphical attributes to create a visual representation, the representation becomes visualization. The visualization exists temporally in time. These view transformations are very common to navigate a larger visual representation to create different
viewpoints to investigate specific details encoded within the visualization of the date. Another viewpoint control technique commonly used encodes both an overview and a detailed representation. The overview provides a detail view of the context, and acts as a control unit to change the detailed view. The detailed view is used to magnify the selected area or focus. Such a focus + context visualization can use separate areas to depict the overview and focused representation or both can be combined within a single view [24]. Typically, the screen real-estate that can be used for the visualization is very limited. To compensate for the imbalance between screen real-estate and space required by the visualization specifically designed deformations, such as hyperbolic projections, are used. If such a deformation can allow users to perceive a larger visual representation, then such modifications are effective.

Interaction and control of information visualization are dependent on the form of encoding of the data within the visualization. Based on the previous information visualization reference model, the user needs to be able to control the visual mapping by tweaking specific parameters. These controls can be in the form of simple user interface elements, such as various control buttons, scroll bars, etc. Many interactive information visualization techniques, in essence, are a form of selection method by selecting subsets of the data table objects to retrieve the desired visual representation. These interactive techniques can be used to position data elements, reveal patterns within the data, select the transform parameters, etc, such as “Details-on-demand” technology [25] , “Brushing” technology [26] etc. These interactive techniques not only improve the speed at which the user interacts with the information, but also can avoid the user taking a wrong path, or get lost in solving a specific problem. Furnas introduced the "universal fisheye view" [27]. This
work first studied the fish-eye view, which is a zoom in of a small local area combined with a lens-type technique. The surrounding information around the enlarged area is pushed toward the outer perimeter, but is still visible. Later, many researchers added further improvements to this technique. That allows the user to observe a small central focus area while maintaining a greater visibility of the surrounding area, which is the meaning of “focus + context” view. This technique can be a collection of information detail view of a specific portion (focus), and the overall structural view of a set of information (context) is mixed together. Or the layout can be considered many large information spaces (context) at the same time, one of it with more details as part of the layout (focus).

3.2 Data structures

The data to be visualized in this thesis stems from researchers in the area of General Recognition Theory (GRT), a multidimensional generalization of the theory of signal detection applied to human perceptual processes [28]. The GRT framework is designed to tease apart perceptual and decisional sources of bias between stimulus dimensions. To use GRT in cognitive modeling, experiments are conducted that present a limited set of stimuli to a human observer who is asked to make a complete identification response (i.e. to identify which of N possible objects was presented on a given trial). This typically results in a so-called identification-confusion matrix (IDCM) [29] that summarizes the frequency with which the subject gives each of the N responses to each of the N stimulus options. Correct responses are tabulated along the diagonal cells of the IDCM, while confusions (incorrect responses) are tracked in the off-diagonal cells. Although GRT research is typically done with human participants in order to model perceptual phenomena, the data to be examined in this thesis comes from a GRT modeling project in which
large numbers of IDCMs (over 2 million data files) are simulated for various GRT theoretical constructs; the purpose of these simulations is to develop and test new statistical analyses for the GRT approach. However, the simulation of such a large number of IDCMs makes data and analyses comparison a hard problem, which could be helped by appropriate visualization tools [2,3].

For example, if we assume that the subjects are supposed to recognize a specific signal that can be on or off the subjects can respond in terms of whether they recognized the signal at that point in time. As a result the confusion matrix consists of a two-by-two configuration with the signal being on or off in one column and the response by the subjects to the signal being on or off. Obviously, the subject can elect to recognize a signal even if there was no signal present, they can fail to recognize the presence of the signal, or they can perceive the signal correctly. Since such experiments can be conducted multiple times with various subjects this can result in a large amount of data that needs to be analyzed.

In the present thesis, the goal was to preprocess the data and then provide a web-based user interface and visualization of the data to provide easy access to the visualization by the application domain specialists. There are different options for web-based visualizations nowadays, including WebGL- and SVG-based methods. In order to keep the visual representation of the data simple to make it easier for the domain specialists to analyze the data, a 2D-based visualization seemed more appropriate. As a result, WebGL seemed to create more overhead and be more difficult to integrate specifically with all sorts of flavors of web browsers. Hence, an SVG-based approach was selected using the D3 JavaScript library.
To develop the techniques described in this thesis, a subset of the overall modeling study was selected by choosing one stimulated model size (total simulated trials = 400). This subset even was a large-scale data set that needed to be imported into D3 for visualization. Within the subset, there were more than 1350 individual model simulations which each contained 1000 IDCMs. For example, Boxes 1 and 2 show sample data files from two models with different types of simulated GRT theoretical constructs. As they illustrate, the IDCM matrix essentially captures the stimulus against the responses of the simulated study participants. As such, one would expect higher values along the main diagonal as that indicates that participants correctly recognized the signal. A deviation from the main diagonal indicates confusion between the actual stimulus shown and the response made. It is hard to see in numerical format, but these two IDCMs show different patterns of confusion in the off-diagonal cells; Box 1 has fairly evenly spread confusion, while Box 2 shows most of the confusion responses in the middle two columns and few confusions in columns 1 and 4.

<table>
<thead>
<tr>
<th></th>
<th>Rx1y1</th>
<th>Rx1y2</th>
<th>Rx2y1</th>
<th>Rx2y2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dx1y1</td>
<td>29</td>
<td>32</td>
<td>34</td>
<td>5</td>
</tr>
<tr>
<td>Dx1y2</td>
<td>17</td>
<td>56</td>
<td>21</td>
<td>6</td>
</tr>
<tr>
<td>Dx2y1</td>
<td>16</td>
<td>10</td>
<td>62</td>
<td>12</td>
</tr>
<tr>
<td>Dx2y2</td>
<td>2</td>
<td>27</td>
<td>35</td>
<td>36</td>
</tr>
</tbody>
</table>

(a)
This matrix essentially plots the stimulus against the responses of the study participants. As such, one would expect higher values along the main diagonal as that indicates that participants correctly recognized the signal. A deviation from the main diagonal indicates confusion amongst the study participants.

There is support for a lot of different kinds of data formats available in the D3 library. Due to the size of the overall data set it is impractical to convert the data by hand. Instead, Visual Basic (VBA) was used to convert the data into a format readable by D3. Once the data was imported into D3 properly, there were different approaches that were tested to find appropriate ways to visually analyze the data, namely parallel coordinates and heat maps following various different layout paradigms.

For a parallel coordinates layout, it is possible to read a csv format, making importing the data easy. To increase the processing speed, the `Scripting.Dictionary` in VBA is used, as a conventional loop to open each data file tends to be very slow. `Scripting.Dictionary` can also add the paths for all files to its dictionary. To add all paths the data files and including all subfolders one can use a call like this one: `Dic.Add ("root folder path\", ")`. By
using a Do While loop, all paths of the subfolders can be checked. Note that each Scripting.Dictionary has its own keys to store Dic.keys. The Dir command also works in VBA, but in order to work with Scripting.Dictionary, it needs the Dic.keys. To look for each sub folder directory in the dictionary one can use a construct such as Dir(Dic.keys (0), vbDirectory). The output will be the same as with the “Dir” command, so the sub folder directory will be followed by “.” And “..”. After this, another Do While loop has to be executed to add all sub folder directory into the dictionary: Dic.Add (Dic.keys (i) & output & "," ). Once completed, the dictionary has all of the folder’s paths stored. Figure 3.2(a) shows an example of the dictionary structure.

```
<table>
<thead>
<tr>
<th>Dic.keys</th>
<th>Sub folder path 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0, root folder path\</td>
<td>0, data file 0</td>
</tr>
<tr>
<td>1, sub folder path1\</td>
<td>1, data file 1</td>
</tr>
<tr>
<td>2, sub folder path2\</td>
<td>2, data file 2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
```

(a) (b)

Figure 3.2: Dictionary structure of the sub folders (a) and the selected files (b)

The next step is to get a dictionary of all the data files under each sub folder. When using a For Each loop with a dictionary, it will use each value in the dictionary within the loop. By using For Each path In Dic.keys, the path variable will go through each sub folders’ path in the loop. To identify all of the data files under its folder, just like determining each sub folder as described before, something like Dir(path & "*.txt") can be used. Because the data files all have the suffix “txt”, this will only select files of that type. Finally, the second dictionary contains the entire data list, which will have a structure as shown in figure 3.2(b):
Then, using the file list, each of the data files can be read into another dictionary. When opening the file, VBA needs the particular path of each file. Hence, it needs a variable to combine the `Dic.keys (0)` and each data file list. Since the dictionary can only add one line at a time, `Trim(fso.OpenTextFile(file).ReadLine())` will read each line in the data file. To detect the end of the file, one can use a `Do While Not fso.OpenTextFile(file).AtEndOfStream` loop construct.

In some of the layouts, duplicated data headers are a big problem. From the figure 3.1(a) example, the text: “Mean(dPrime): 1 AnyZeros: 0 vPI_Rho: 0.00 vPS_Shift: 0.00 vDSX_piecewiseShift_or_continousSlope: 0.00 SampleSize: 100 Rx1y1 Rx1y2 Rx2y1 Rx2y2 ” will be duplicated data in all 1000 IDCMs simulated for the same set of model parameters. Also some rows will have duplicated values, which will slow down the process and display in the web browser. In the worst case, the D3 library will simply crash the web browser in our tests if there are too many duplicated values. With the parallel coordinates layout, there was no difference whether the duplicated values were included or not. However, the duplicated values slowed down the rendering speed significantly when the user is trying to select a range or change the position of the rows. This performance issue can be prevented by using the dictionary. VBA can look for the new line and check if it exists in the dictionary or not by using `Did.Exists(stringLine)`. It will return a true or false, so in order to avoid duplicated values, one can add the new line only if `Did.Exists(stringLine)` evaluates to false.
3.3 Step of generating Parallel Coordinates Data

In order to generate the data so that it can be imported into D3 Excel is used. First, open the .xlsm file which includes the VBA scripts. If Excel disabled the active content, choose “Options”.

![Security Warning]

Figure 3.3: Some active content has been disabled

Then select “Enable this content” in both sections.

![Security Alerts - Multiple Issues]

Figure 3.4: Some content need to be enabled
Choose View -> Macros -> view macros

Figure 3.5: The Macros option

Select “parallel” and hit edit

Figure 3.6: Edit the parallel macro
In the line 9, find the code `Dic.Add ("***"), ""` and change the *** to the folder you want to make it to the parallel view. Be sure to include the “\” after the path. For example:

`Dic.Add ("D:\d3\test\"), ""`

```vba
Set Did = CreateObject("Scripting.Dictionary")
Set fso = CreateObject("Scripting.FileSystemObject")

Dic.Add ("C:\Users\RFID\Downloads\DPrime1\DPrime1\vEI2\"), ""
i = 0
Do While i < Dic.Count
```

Figure 3.7: Locate the file path

Choose Run in the menu -> Run sub/userForm. This will start the script to process all sub-folders in the path. Once it finished, it will show up a message and tell you how long it takes.

```
Figure 3.8: Generate the data
```

Open the source folder, which will show a file called “data.csv”.
Copy all files in the folder called “parallel” into the dataset’s folder you want to see.

Open the “parallel2.html” with Firefox to see the parallel coordinates plot.
Figure 3.11: Use the html file to see parallel coordinates
3.4 Step of Generating Heat-map data

Copy the Consolidation.exe into the path of the matrix dataset’s folder.

![Figure 3.12: Consolidation program](image)

Figure 3.12: Consolidation program
Run Consolidation.exe and type “type *.txt >> all.txt”.

Figure 3.13: Use the command to do the consolidation

Figure 3.14: The consolidation has been finished
The “all.txt” will be the file that includes all matrixes in the dataset.

Figure 3.15: Locate the file with all matrixes

Open both “heatmap.xlsm” and “heat3.xlsm”.

Figure 3.16: VBA programs
Choose Data-> from text, then select the “all.txt” in which data-set folder you want to covert the heat-map view.

Figure 3.17: Get the data from a file

Choose next and set the break line, then hit finish.

Figure 3.18: The break lines
Choose Data -> Sort & filter -> sort A to Z.

![Spreadsheet screenshot showing sorting options](image1.png)

Figure 3.19: Sort the data

Select the first row and right click -> delete.

![Spreadsheet screenshot showing delete option](image2.png)

Figure 3.20: Deleting of unused row
Choose View -> Macros -> view macros.

Figure 3.21: The Macros option

Select the “heatmap” and hit edit.

Figure 3.22: The heatmap Macro
In the last row, change **Filename;="***"** to the path and output file name:

```vba
Range("FD100").Select
ActiveCell.FormulaR1C1 = "159"]"
Range("FD401").Select
ActiveWorkbook.SaveAs Filename:="C:\heatmap.csv", _
    FileFormat:=xlCSV, CreateBackup:=False
End Sub
```

Figure 3.23: Locate the output file

Choose Run in the menu -> run sub/userForm.

Figure 3.24: Generate the data
Open the folder, there is a file called “heat.csv” will be in the folder. Use a text editor to open, like the Notepad.

Figure 3.25: The generated data file

Replace all of “[,“ with “[“.

Figure 3.26: Words replacement
Then replace all of “[,]” with “,”.

![Image of replace tool]

Figure 3.27: Words replacement 2

Copy all files in the folder called “heat-map” into the dataset’s folder you want to see.

![Image of folder contents]

Figure 3.28: The html files

Open data.js using Notepad, and copy all data from heatmap.csv to “var data = [ ]”.

```
var maxData = 90;
var minData = 0;
var data = [[],[]];
var cols = [];
var rows = ['Dx1y1', 'Dx1y2', 'Dx1v2', 'Dx2v1', 'Dx2v2'];
```

Figure 3.29: Add the generated data
Delete the last comma, it should be like this. Save the file.

```
[11, 99, 150], [18, 99, 151],
[11, 99, 154], [14, 99, 155],
[17, 99, 158], [12, 99, 159]]
```

Figure 3.30: Check for the last data

Open the “heatmap.html” with any web browser to see the heat-map view.

### 3.5 Data visualizations with D3

After preprocessing the data to a format readable by the D3 JavaScript library, the next step is to use D3 to provide a web-based user interface and visualization of the data so the application domain specialists and the users can get easy access to the data through the visualization. The D3 library can read many types of data, however it can only preprocess very limited kinds of data formats, such as the csv format or three-dimensional arrays. The csv format is very suitable as it can make editing the data easy and reviewing it at a later time as there are many tools that can read and edit the csv format, such as Microsoft Excel and VBA. Thus, the csv format was chosen in this project to import data into the D3 library.

In order to develop the techniques describing different elements in this thesis, several heat maps and parallel coordinate plots were created using D3. In the parallel coordinates layout, 5 different lines have been set first, to capture the four individual sets of data and the average data across IDCMs, and CSS (Cascading Style Sheets) was used to set the styles. The rows Dx1y1 in the confusion matrices data sets is set to brown by using
Dx1y1 { stroke: #80 00 00;}; Dx1y2 is green so Dx1y2 { stroke: #00 88 00;} and Dx2y2 { stroke: #FF A5 00;}, respectively. The last line style illustrates the average value for the IDCM data set and is over-drawn on other lines, so it need to be set to black by using .avg { stroke: #00 00 00;}. The width of each line has been set to 2 px (pixel) by using line { stroke-width: 2px; }. Because of the large number of lines overlapping each other, the translucence has been set by adding a stroke-opacity: 0.5; parameter. The next step is to read the data by using the D3 library: d3.csv ("data.csv", function(layout)). It also stores the data in such way that it is accessible via a function called layout, so the data can be reached anywhere by a simple function call. D3 can read the csv data file and preprocess them to different traitsarrays. In this confusion matrices data set, Dx1y1, Dx1y2, Dx2y1, and Dx2y2 are different traitsarrays.

The values of those arrays need to be transformed into different lines for the parallel coordinate plot and plotted at the correct height positions. In order to show different values in the arrays, a larger value will be set to a higher position in this layout. The height of each position shows the difference of each value. Now it is simple to show only this dataset because the minimum is 0 and the maximum is 100. If the maximum height in this layout is 50 pixels, a value of 20 can be set to 20% of 50 pixels and a value of 80 can be set to 80% of 50 pixels. However, some data sets have a maximal value of up to 1000. Hence, a scale function within D3 is deployed to rescale the values appropriately. In order to use this function, “domain” and “range” need to be provided as input parameters. The domain describes the range of input values whereas the range is the range of output values. In this case it should output a pixel unit to set the position. For example, if the
minimum value is 0 then the position is 50 pixels; if the maximum value is 100, then the position is 200 pixels; the range of the output is 50 to 200. In this case, the center of the range is 125, so if the input value is 50, then the output will be 125 pixels. In order to use this function, \( y[d] = d3.scale.linear() \) is used to access it and applies a linear scaling. To set the domain input, \( .domain(d3.extent(layout, function(p) { return +p[d]; })) \) will determine the range of input values in the array. The function called \( layout \) has been generated before, so it provides access to all the values in this data set. The \( d3.extent \) function is used to determine the minimum and maximum value in that array. To set the range of output values, \( .range([h, 0]) \) will set the range of output pixel. At this point, the position of each value has been set. At last, the line for each path will be generated via an SVG object. An SVG function in D3 is then creates a new path: the function call \( svg.append("svg:path") \) is used to create a new HTML element and \( .attr("d", path) \). \( .attr("class", function(d) { return d.rows; }) \) is called to set the attributes. Because the style of the path’s CSS is already set, the lines generated as new HTML elements will use different colors for each array as specified in the CSS.

### 3.6 Data visualizations in virtual environments

VTK as a generic C++ class library has a wide range of scientific and engineering usage. It can be used with another C++ based virtual environment library called VRUI (virtual reality user interface) [30]. One of the most important and common applications is in the medical field, such as Dr. Wichgoll’s "Visualizing Vascular Structures in Virtual Environments" project is using VRUI. The goal for this thesis is to make a 3D semantic clus-
ter based visualization in virtual environments. One important aspect is the three-dimensional reconstruction. So, using VTK and VRUI to three-dimensional reconstruction of a semantic cluster database is a suitable application example. Using VTK and VRUI for three-dimensional reconstruction algorithms and data structures from the data, the most difficult part is reading data and the graphical display. The database for semantic cluster is using a tlp format which is defined within DBpedia project. DBpedia is a crowd-sourced community effort to extract structured information from Wikipedia. Its database is served as Linked Data on the Wikipedia [31]. Because the database conforms with the tlp standard, reading that format is not a big problem. VTK has already support for this kind of data: the vtkTulipReader class subclasses can be used to perform this task. In our own program - as long as the parameters are specified correctly - it is capable of reading the linked data. According to the data pipelined in VTK, the original data can be used for various filters to generate a variety of data conversion. In this case, the linked data has been extracted, mapped, and drawn. Because it is a three-dimensional reconstruction, it should be added using an interactor so that VRUI can take care of interacting with the drawing window. Regarding the reconstruction algorithm, I used some mature and basic algorithm for reconstruction, such as clustering algorithms, force directed, and spanning tree. I used the VTK development platform to complete the above work, such as read database file, reconstruction and VRUI for display, interactive features such as integrated into the 3D monitors, in order to provide a convenient user interface in virtual environments.
4. RESULTS AND DISCUSSION

4.1 Parallel coordinates

Parallel coordinates are a very useful layout to compare each row and each column in every matrix. In the IDCM datasets, the rows "Dx1y1", "Dx1y2", "Dx2y1", and "Dx2y2" are drawn with different colors, in order to make it much easier to compare. The columns, "Rx1y1", "Rx1y2", "Rx2y1", and "Rx2y2", are then plotted as the axes of the parallel coordinate system to show the values. This allows the user to compare a value with other rows using a different color, or compare with different columns using the same color. If the lines become bolder, that means the overall values in the matrix tend to be noisier; and if the lines become thinner, that means the values are much closer the each other. The black lines in the centers of each different color are showing the average values, which enables the user to easier understand the difference of the spread of values encoded within the matrix. Also this layout is capable of showing a lot of different data sets together within a single visualization, so comparing different dataset can be done easily.
The nine parallel coordinate plots in Figure 4.1 illustrate nine different GRT theoretical model situations; researchers are trying to understand how different combinations of GRT model construct violations influence the pattern of responses within the IDCMs. Figure 4.1 shows a possible way to quickly compare the rates of responding to the different stimulus conditions (the different colors) across multiple models. Notably, the yellow lines, representing the Dx2y2 stimulus, show variability across the plots; this was the stimulus modified by changes in the simulation parameters. The center plot shows very little confusion for the Dx2y2 stimulus, because the response rates are low for "Rx1y1", "Rx1y2", and "Rx2y1", and high "Rx2y2". This is very different from the models in the
bottom right and bottom left corners, which show a large amount of confusion of Dx2y2 with the other stimuli. Thus, those models produced more confusion and might be harder to analyze in a human perception experiment. Domain experts see much potential for using these parallel coordinate plots for model comparison based on the IDCM patterns alone (without additional statistical analyses).

### 4.2 Heat map visualization

A heat map is another way of visualizing the matrix data. Generally, heat maps, while not depicting the exact absolute value encoded in the confusion matrix, are useful for providing an overview of the values of the confusion matrices relative to each other. The goal here was to utilize the heat maps to visualize the data from multiple individual IDCM data sets, including layouts that might enable the visualization of all 1000 individual IDCMs for each simulated model. Different layouts for heat maps were tested to identify the more suitable ones for looking within and across datasets. Figure 4.2 shows an example of a heat map layout; it is a combination of each matrix in the dataset. It highlights each individual value within the entire data set using color coding ranging from white (low value) to red (large value). Also for each exact value, the user can hover the mouse over the heat map which will automatically show the exact values in that confusion matrix the mouse cursor is currently hovering over (Figure 4.2, right hand side). This basic heat map allows for several IDCMs to be displayed together, which can help identify some patterns of confusion. However, with 16 cells for each IDCM, this particular layout can get unwieldy rather quickly.
Figure 4.2: Layout example of heat map - a combination of each matrix in the dataset

The image on the right illustrates the details on demand offered by a mouse hovering over a single IDCM.

Another heat map layout assembles all values in the dataset to one matrix layout, to better combine the large number of data files. This results in specific cells within the IDCMs being lumped together in the same heat map block. For example, the values of the upper left cell (row 1, column 1) of each IDCM gets plotted in a single block where the value of the first confusion matrix is used for the first cell within the block, the second value is used for the second cell, and so on. This better preserves the overall layout of the confusion matrix (4-by-4 grid), which makes it appear more familiar with the domain specialists. It also tends to better show the overall confusion for the entire study, because all 1000 IDCMs are included. Figure 4.1 shows four examples of this layout. Comparing the heat maps in the top row of Figure 4.3, both show the darkest blocks are all along the main diagonal, which means the responses overall were correct (this is important for ensuring that the model or human participant are doing the task as expected). However, the heat map in the upper left has darker blocks surrounding the main diagonal compared to the upper right, which indicates greater confusion in response to the stimulus for the up-
per left model. The bottom left heat maps shows an example with even less confusion during the study, as many of the off-diagonal blocks are close to white in color (very low numbers of responses). Similar to the previous layout, the user can hover the mouse over the heat map which will automatically select each value within a single block, thereby showing the exact values of that block (Figure 4.3, bottom right). Similar to the parallel coordinate plots in Figure 4.1, the domain experts will be able to use these blocked heat maps to quickly compare multiple stimulated model conditions for their influence on the overall pattern of confusions.

Figure 4.3: Heat map layout in which values are lumped by their position within the confusion matrix. Each of the four heat maps illustrates a different simulated set of data (i.e. different underlying model) but all capture the 1000 IDCMs simulated per model. This is to show how this layout also highlights differences in the overall confusion patterns (color changes in the off-diagonal blocks). The mouse over action illustrated in the bottom
right corner now shows all the values from the 1000 IDCMS for the selected cell (here, the row 1, column 1 cells of all 1000 IDCMs).

Figure 4.4: The same data set as shown as in Figure 4.2 using a circular layout for a heat map, but using less screen space to show the same information.

A circular heat map is another way of arranging the values in this type of visualization. It provides the same functionality as the previous heat map layouts, but uses a different basic shape to accommodate a different layout. Instead of arranging the individual values in a rectangular layout a circular metaphor is used. Figure 4.4 shows an example using the circular layout using the same data as depicted in Figure 4.1. Note that in comparison to Figure 4.2, it uses less space to show the same amount of information. Figures 4.5 and 8 show the same data set as Figure 4.4, but Figure 6.1 breaks each block into a ring in the
circle, that will use much less space to show each value in each block. Hence, it follows a similar paradigm as Figures 4.4 in that it lumps together the values from a specific cell of all confusion matrices.

![Circular layout showing the same data set as Figure 4.3, lower right](image)

Figure 4.5: Circular layout showing the same data set as Figure 4.3, lower right

Across these circular layouts, domain experts like Figure 4.5 the most for quick grasp of the IDCM confusion patterns. While Figure 4.4 does track the individual rows as rings of the circle, it is harder to separate the columns of each individual IDCM; the data in the innermost ring (Dx2y2) is also quite small and harder to see. The rings in Figure 8 capture patterns within each ring well, but experts find it hard to compare the same response ring across the different stimuli (Dx1y1-Rx1y1, and Dx2y2-Rx1y1, for example) because
of the extra rings in between them. With the quadrant layout to Figure 4.5, domain experts are reminded of the traditional means of illustrating GRT model space, which involves plotting a bivariate normal curve in each of 4 quadrants representing the 4 stimuli. Thus, Figure 4.3 may lend itself to easy integration with other GRT plotting tools.

4.3 Semantic cluster based visualization in virtual environments

There are many algorithms I can use to visualize the semantic cluster database, but their functions are limited. So I made my own algorithms to do the data reconstruction. At first, it will put each node randomly in the space and connect them with edges. Then it can use any basic reconstruction algorithms to move each node. In this case I used force directed algorithms. After about 150 times of moves, it will finish the reconstruction and become a force directed graph. At last it will rotate to a better direction.
Figure 4.6: Random position nodes
Figure 4.7: Nodes start to move using force directed
Figure 4.8: Become a force directed graph
VTK also has some other reconstruction algorithms in the library which I can use directly. For reconstruction, such as fast, simple, clustering, force directed, and spanning tree algorithms.
Figure 4.10: Fast algorithms

Figure 4.11: Simple algorithms
Figure 4.12: Force directed algorithms

Figure 4.13: Clustering algorithms
When it is showing the text information of each node, it will only show the labels in that zoom in such a way that overlap between the labels is avoided.
4.4 Accomplishments

Within this thesis, pre-processing tools for the raw data were developed to identify ways of visualization methodologies that enable domain specialists to more easily interpret their data. A parallel coordinates with five different layout approaches and modified heat maps were implemented to determine those methods that are most suitable for visualizing the data from the IDCMs simulated for GRT models. VTK and VRUI based semantic cluster database visualizations in virtual environments with six different layout approaches were developed as well.
5. CONCLUSIONS

Information visualization as a tool has matured into the analytics context. It is based on computer graphics, general graphics principles and methods. It converts massive data into graphics, images, and representations in visual form. It involves computer graphics, image processing, computer vision, computer-aided design, graphical user interface and many other research areas. Humans obtain information retrieved from external data mostly through the visual channel. Visual information is processed by humans at high-speed, high-capacity, and parallel features. Hence the sayings "seeing is believing" and "a picture is worth a thousand words". With the rapid development of human society, people in more and more scientific research projects and production practices results in more and more data. To cope with the information overload present in today’s society, appropriate visualization techniques are needed. Within this study, several different visualization approaches were developed to determine those methods that are most suitable to visualize the data from the IDCMs simulated for GRT models, namely parallel coordinates and modified heat maps using different layout approaches. As is expected, some are better suited than others depending on the task the domain specialists are trying to achieve as pointed out during the discussion of the results. Also VTK and VRUI based semantic cluster database visualizations in virtual environments were developed as well, namely fast, simple, force directed, clustering, span tree and my own layout using different algorithms. Overall, the visualization methods were very successful in visualizing aggregated IDCMs and semantic cluster database in a way that allowed the domain specialists to eas-
ily interpret the changes in the patterns of confusion across different models. Today, most design is using paper media to express creativity, and drawing is the basis of modern industrial production. Visualization method still continues with the tradition of expressing human intentions. From this research, the visualization technology generated diagram is a new form of the data. The emergence of visualization technology has a profound historical background, which is a huge demand of society and technology progress. Visualization techniques have long time of research, so the original charts and statistics visualization techniques were applied to many scientific data analysis. The birth and popularization of computers is taking the human society into an information age. It provides new scientific computing and data acquisition to human society. This also means to take human society into a sea of data. Scientific research’s purposes are not only in order to obtain data, but through analyzing the data to explore the laws from data. The traditional visualization techniques and data analysis mean inefficiency, and have severely restricted the scientific and technological progress. As computer software and hardware performance continues to improve and the rapid development of information visualization, they are prompting people to apply this new technology in scientific data analysis.
6. FUTURE WORKS

While the domain specialists were very pleased with the current results, a more formal user study will have to be performed as part of the future work.

Figure 6.1: Heat maps of the same data-set with Figure 4.3 (lower right), but it breaks each block into each ring in the circle, that will use much less space to show each value in each block.

Also the VTK and VRUI based virtual environments software could benefit from improved interaction methods based on, for example, Microsoft’s Kinect, Wii remote, or a data glove. These methods would have to be tested within a user study.
BIBLIOGRAPHY


[7] Zhao, K., Liu, B., Tirpak, T. M., & Schaller, A. “Detecting patterns of change using enhanced parallel coordinates visualization”. In Data Min-


[31] DBpedia http://wiki.dbpedia.org/About