An Empirical Study on Parallel Coordinates and Scatter Plots

Rassadarie Kanjanabose
Green Templeton College
University of Oxford

A thesis submitted for the degree of
Master of Science in Computer Science

September 2014

Acknowledgements

I would like to express my sincere gratitude to my project supervisor, Professor Min Chen, and his postdoctoral research associate, Alfie Abdul-Rahman, for their excellent guides and suggestions throughout the dissertation project. Without their supports and helps, I would not be able to complete this project with this successful result.

Besides, I would like to thank Professor Stephen Pulman, my supervisor in Michaelmas and Hilary terms, who have guided me through the MSc degree. Also, I would like to thank other professors and staffs at the Department of Computer Science, University of Oxford, for their lectures and academic assistance.

Additionally, I would like to thank all of my friends, colleagues, and other people who have participated in my user study. I would like to further thank all my friends in Oxford, with special thanks to Shijian, Helen, Denton, and Hillary for our academic discussion, their encouragement, and the experiences we have shared.

Lastly, I would like to thank my parents, my sister and brother, and my boyfriend for their endlessly supports throughout the academic year.

Abstract

Many visualisation techniques have been proposed and developed in recent years. Nevertheless, only a few researches have been conducted to evaluate these techniques. In this study, we empirically compare user performance, in terms of accuracy and response time, among three visualisation techniques, data table, scatter plots, and parallel coordinates plots, in four different visualisation tasks, value retrieval, clustering, outlier detection, and change detection. Results show that data table is the better technique in value retrieval task, while parallel coordinates plots outperform other techniques in change detection task. For clustering and outlier detection task, data table yields lower performance in all conditions, while parallel coordinates plots hold advantages over scatter plots in hard-level tasks. Subjective feedbacks also support the quantitative analyses. These results provide guidelines to select appropriate visualisation techniques for each visualisation task, which can be beneficial in information visualisation field.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Data analysis is a core process in discovering useful information and aids in decision making in various application domains. Currently, the explosion of data poses a major challenge to computer scientists and technology researchers in developing new algorithms and tools to perform effective analyses [YGX+09]. According to Schroeck *et al.* [SSS+12], as we are entering into the era of Big Data, visualisation is one of the valuable tools that can be used to convey information from this large amount of data. Visualisation provides better understanding and supports in problem solving, which increases the data analyses' performances. Thus, the information visualisation field has become one of the most important areas for current research studies.

In recent years, new visualisation techniques have been increasingly developed. Yet, these techniques are not equally suited for all types of data analysis [JFLC08]. The visualisation techniques may yield high performance in some visualisation tasks, but may not considerably support some other tasks. For this reason, it is important to find the methods of visualisations that are most suitable to each type of data analysis. As a consequence, the evaluation of each visualisation technique is required. Carpendale [Car08], emphasised that as research on visualisation techniques have been increased, it is becoming significantly important that these techniques are validated. The amount of these techniques' evaluation, however, does not proportionally increase [Car08].

The examples of the invented visualisation techniques include scatter plots and parallel coordinates plots, which are the two generally used techniques in data analysis on multivariate data [KZZM12]. These visualisation techniques have been developed and applied to many application areas [Weg90, KD09, TGS04, HNR05, AR11]. Nevertheless, only a few researches have been done to compare the user performance between these two types

of visualisation techniques [KZZM12]. Therefore, it is still questionable that which one performs better.

To our knowledge, the only previous comparison between these two techniques on multivariate data analysis was conducted by Kuang *et al.* [KZZM12], whose results indicated that parallel coordinates plots hold an advantage in low dimensionality and density dataset, whereas scatter plots perform better in the dataset with higher dimensionality and density. Yet, the research involved only value retrieval task, which is a simple and less critical task in data analyses [Pla04]. Several other complex tasks, such as clustering, correlation analysis, outlier detection, change detection, and classification tree generation, have not yet been evaluated.

1.2 Objectives

The goal of this project is **'to evaluate scatter plots (SCP) and parallel coordinates plots (PCP) in different visualisation tasks.'**

We will extend the previous research [KZZM12] by involving more tasks and also include data table (DT) to provide further comparison with the basic technique. We will focus our study on visualisation tasks for multivariate data as the data sets in the real world usually involve multiple variables. The method that we will use in our research is an empirical study, which is a commonly used method for evaluating visualisation techniques [Pla04, LBI+11]. The purpose of the empirical study is to compare user performance, in terms of effectiveness and efficiency, between DT, SCP, and PCP.

The objectives of the project include:

1. To study visualisation techniques in general, development of SCP and PCP, and the evaluation of visualisation techniques.

2. To formulate hypotheses of the empirical study, identify experiment's variables, indicate measurement metrics, and establish techniques for the result analyses.

3. To design visualisation tasks, stimuli, and software involved in the empirical study, and implement them accordingly.

4. To conduct an empirical study.

5. To analyse the empirical study's results.

To our knowledge, our research is the only empirical study that aims to compare DT, SCP, and PCP in various visualisation tasks of multivariate data, other than the value

retrieval task mentioned earlier. In conducting the study, we hope that our results will be beneficial in the information visualisation field, by providing evaluation of these three visualisation techniques and their relative advantages for different visualisation tasks.

1.3 Structure

This dissertation consists of seven chapters, including this introduction presenting the motivation behind the research study, the objectives of the project, and the outline of the dissertation structure.

Chapter 2 will provide background information on visualisation techniques in general, the literature reviews of SCP and PCP, and the approaches for evaluating visualisation techniques.

Chapter 3 will explain the methodology used in this research, covering the details on the study's hypotheses, variables in the experiment, measurement metrics, and result analyses' techniques.

Chapter 4 will discuss the user study design, consisting of task design, stimulus design, and software design. For task design, it will provide both general aspects and specific designs for each visualisation task.

Chapter 5 will describe the implementation of the study, starting from stimulus generation, software development, and followed by the experiment of the empirical study.

Chapter 6 will report the analysis of the empirical study's results, including the summary and the detailed analyses of each visualisation task.

Chapter 7 will summarise the research's results, project's evaluation, and suggestions for future works.

The appendices will further provide pictures of the visualisation techniques mentioned in the dissertation, different versions of stimuli, all stimuli used in the experiment, sample interfaces of the software, sample pictures of the experiment, data table for the result analyses, and schedules for our project.

Chapter 2

Background

Despite the popularity of scatter plots and parallel coordinates plots visualisation techniques in analysing multivariate data, their advantages in many visualisation tasks have not yet been evaluated.

This chapter begins by explaining what visualisation is, including the types of data involved, the techniques used in visualisation, and the tasks performed in visualisation. The discussion then moves to the development of scatter plots and parallel coordinates plots, followed by the evaluation approaches in information visualisation field, including previous researches on user performance comparison between scatter plots and parallel coordinates plots. In the last section of this chapter, we describe how an empirical study is conducted to evaluate different visualisation techniques.

2.1 Visualisation

The term *visualisation* has many definitions, but as mentioned by Ward *et al.* [WGK10, p. 1], it is 'the communication of information using graphical representations'.

Visualisation is commonly used to transform data to visual representations in order to increase the performance of data analyses [CFB13]. The goals of visualisation are to maximise human understanding and aid in problem solving [CFB13, WGK10].

Presently, people use various visualisation techniques to analyse different types of data.

2.1.1 Data Types in Science Measurement

In general, data can be broadly categorised into two types, quantitative data and qualitative data [Pri11]. Quantitative data (or numerical data) is a numerical measurement expressed in terms of numbers, whereas qualitative data (or categorical data) is a categorical measurement expressed by means of a description [Pri11].

In 1946, Stevens [Ste46] proposed that there are four primitive data types in all science measurement: *nominal*, *ordinal*, *interval*, and *ratio*.

Nominal refers to categorical data with no ordering of the categories [Pri11]. The examples of nominal data are different names such as types of cloth and names of schools. This type of data can be compared using $=$ and $\neq$ operators only.

Ordinal refers to categorical data that implies a natural ordering [Pri11]. The ranking of countries in Olympic Games, the grading scales, and the order of operations in a manufacturing chain are examples of ordinal data. Typical operators for ordinal include $>, <, =$, and $\neq$.

Interval refers to quantitative data where the intervals between each value are equal [Pri11]. The examples of interval data include calendar dates and temperature in Celsius. Interval data can be used with $+, -, >, <, =$, and $\neq$ operators.

Ratio refers to the interval data with a natural zero point; zero value has a unique and non-arbitrary meaning [Pri11]. For instance, Kelvin temperature scale is ratio since it has a unique zero point called *absolute zero*. Other examples are age, weight, and volume, whose zero value means none. Operators for ratio data are $+, -, \times, \div, >, <, =$, and $\neq$.

2.1.2 Multivariate Data

Multivariate data is a group of variables that are associated with the same object [WGK10].

These variables may represent different features of an object or represent the same feature in different conditions [WGK10]. The example of the former case is a group of colour, length, width, and thickness variables of a pencil case, while the example of variables in the latter case is the heights of a child in different years.

Objects in the real world usually have more than one variable. Therefore, visualisation of multivariate data is required.

2.1.3 Visualisation Techniques

Many visualisation techniques, targeting at both generic and specific domains, have been proposed to visualise multivariate data [LWZK08]. The following subsections will provide examples of some common techniques; the figure of each technique will be displayed in Appendix A.

2.1.3.1 Data Table

One of the basic techniques to present multivariate data is the data table.

A data table arranges data in rows and columns, where each row represents a data object (data point) and each column represents a variable. The intersection of a row and a

column is a cell. The value of a variable in a data point can be read from its corresponding cell.

Data table is widely used in all communication, including research and data analysis, as it can clearly display all data values in one place.

2.1.3.2 Scatter Plots

Scatter plot is another commonly used visualisation techniques for multivariate data.

In a scatter plot, two variables of each data object are used to plot a point in two-dimensional (2D) space, typically defined by horizontal and vertical axes, resulting in a scattering of points [FD05, YGX+09]. Each axis represents a variable, whereas each point represents a data object. A single scatter plot can show only two variables, yet it can be combined to visualise multivariate data with more than two dimensions [FD05].

Friendly and Denis [FD05] stated that scatter plot may be considered as one of the most useful invention among all forms of graphical representations. Scatter plot is an extremely effective visualisation technique used to discover clusters, trends, and relations between variables [FD05]. The development of scatter plots will be explained in Section 2.2.

2.1.3.3 Parallel Coordinates Plots

Another popular technique for transforming multivariate data into a 2D image is a *parallel coordinates plot*.

In a parallel coordinates plot, the axes are placed in parallel and each data point is represented as a series of line segments intersecting the axes at the corresponding values [ID90]. Each axis corresponds to a variable, while each line represents a data point.

Cuzzocrea and Zall [CZ13] summarised that parallel coordinates plots have several advantages. Parallel coordinates plots can provide continuous and comparative display for more than two variables with the ability to present different variable combinations [CZ13]. This technique enhances the discovery of data trend and relationship [CZ13]. The development of parallel coordinates plots will be described in Section 2.3.

2.1.3.4 Treemap

Treemap is another visualisation of multivariate data, using hierarchical structures [Shn92].

In a treemap, the hierarchical data, or tree-structured data, is presented as a set of nested rectangles, where each tree node is a rectangle and the node's children are its sub-rectangles [BHVW00]. Bruls *et al.* [BHVW00] clarified that the size of each leaf node determines the size of a sub-rectangle, while the size of each non-leaf node's rectangle is the sum of its children's sizes.

Bruls *et al.* [BHVW00] emphasised that the leaf nodes can also use colour coding and annotation, which supports the ability to recognise node patterns in different depth. In addition, treemap uses space efficiently, allowing it to present a large amount of data in a limited space [BHVW00]. Treemap is useful when the most interested variable of the data is the size feature [BHVW00], yet it does not show clearly advantages when applying to other features or when the data is not in hierarchical structures.

2.1.3.5 Star Glyph

The *star glyph* is also known as *radar chart* and *spider chart*.

In a star glyph, a data point is represented by a glyph, which is basically referred to symbol or icon [Mar07]. Each glyph consists of a centre point with equally angled rays; each rays corresponds to a variable and the lengths of the rays are proportional to the values of each variable [ANO12, Mar07]. Marghescu [Mar07] explained that the end points of these lines are usually connected together to form a polygon in order to reduce confusion when multiple glyphs overlap.

According to Rusu *et al.* [RSCT09], star glyphs provide effective performance for clustering and outlier detection tasks. Nevertheless, when the data set becomes significantly large, too many glyphs will be presented which decreases the ability to read and analyse data [FCI05, RSCT09].

2.1.4 Visualisation Tasks

For different data analyses, different visualisation tasks are involved. Following subsections will give some example of the commonly performed visualisation tasks in several researches and analyses.

2.1.4.1 Value Retrieval

Value retrieval task is the task to retrieve or read value of a data point. Amar *et al.* [AES05] listed value retrieval task as the first and simplest task in data analysis, which is a fundamental task to further perform complex task, for instance, sorting and outlier detection.

The value of each data can be read directly from the cell in a data table. On the other hand, in scatter plots and parallel coordinates plots, the value of each variable can be read from its corresponding axis.

2.1.4.2 Clustering

Clustering is one of the most important tasks for analytic activity. It is the task of grouping similar objects into the same group, called *cluster* [Bis06, Mur12]. One of the most popular algorithms used in clustering is k-means clustering, explained in the following subsection.

To partition a data set in a data table into clusters, we must observe the data points' values directly and group those with similar values together. However, in a scatter plot, the clustering task can be performed by looking at each data point's location; the data points located near each other can be grouped into a cluster. Meanwhile, the proximity of polylines at the axes of a parallel coordinates plot can also be used to determine the clusters; close lines belong to the same cluster. Furthermore, various techniques have been proposed to ease in cluster detection for parallel coordinates plots, such as using colour, opacity, curved line, and animation [HvW10].

2.1.4.2.1 *k*-means clustering

k-means clustering is based on two recursive observations [Bis06, Mur12]:

1. We can assign each data point to its nearest cluster if we know each cluster's centroid.
2. We can calculate each cluster's centroid if we know the cluster that each data point is assigned to.

The objective of k-means clustering is to provide optimal clustering, which minimises the distances from each cluster's centroid to each of its data point [Bis06, Mur12]. The technique has 5 main steps, as follows [Bis06, Mur12]:

1. Select the number of clusters, denoted as k.
2. Randomly pick k points from all data points as the centroids of the k clusters.
3. Assign each data point to the cluster whose centroid is the nearest. The distance between each data point and the clusters' centroids can be calculated using one of the distance measure, such as the *Manhattan distance* (l^1 *norm*), the *Euclidean distance* (l^2 *norm*), and the *Minkowski distance* (l^p *norm*). Each assignment is a binary variable r_{nk}, which denotes the assignment for n^{th} data point to k^{th} cluster, such that $\forall n \sum_k r_{nk} = 1$ as each data point can belong to only one cluster.
4. Recompute each cluster's centroid based on its assigned data points. Set each centroid equal to the mean of all data points within its cluster.

$$\mu_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}}$$

where μ_k is the centroid of cluster k and x_n is the nth data point.

5. If the centroids have changed significantly, repeat the process from step 3.

The algorithm for k-means clustering in this study will be provided in Section 5.3.

2.1.4.3 Correlation Analysis

Correlation analysis is the task to find *correlation* in a data set, which is the relationship between variables [Gar12]. It is another useful task as correlation can be used in data forecasting. Garson [Gar12] emphasised that when one variable is known, correlation supports the predictions of another variable that is related to it; the higher the correlation, the more accurate the predictions.

For a data table, correlation can be observed by looking at each data point's values and seeking for the relationship between one variable to another variable. Nonetheless, correlations can be discovered in a scatter plot by inspecting the patterns of data points. If the points resemble a line rising from lower left to upper right, it suggests a positive correlation between the two variables. Conversely, if the pattern falls from upper left to lower right, a negative correlation would exist. On the other hand, if the points are scattered with no pattern, there is no correlation. Regarding the parallel coordinates plots, a pattern with parallel line segments indicates a positive correlation, and a pattern with intersecting line segments in X-shapes represents a negative correlation. In contrast, if some line segments are in parallel and some are randomly intersecting, there is no correlation.

2.1.4.4 Outlier Detection

Outlier detection is the task to identify *outliers* in a data set. An outlier is an observation point that deviates significantly from other observations [Gru69]. Grubbs [Gru69] mentioned that outliers can suggest a data entry error or may indicate something scientifically interesting.

For a data table, the data points that have a large value differences from other data points are identified as outliers. Equivalently, the points that lie far away from the main data distribution in a scatter plot are considered as outliers. Similarly, in a parallel coordinates plot, line segments that are isolated from other dominating parts are outliers.

2.1.4.5 Change Detection

When analysing a data set with each variable representing a state of time, change detection is the task to detect changes in each data point. Change detection is a critical task in many

application areas. The change detection in land cover, for instance, can be used to evaluate the potential effects on erosion, flooding, and climate changes [Mas99].

Change detection in data table and scatter plots can be performed by looking at each data point's value and searching for a change of values across variables. While in parallel coordinates plots, the data lines that does not form a constant line denotes changes.

2.1.4.6 Classification Tree Generation

Classification tree, or *decision tree*, is a tree with decision nodes and leaf nodes. Each decision node is a variable, which has two or more branches representing its possible values, while each leaf node corresponds to a classification or decision [Mit97].

To generate a classification tree, one of the possible methods is to calculate *entropy* and *information gain* of each variable and manually construct a tree from the variable with the highest information gain [Mit97]. For parallel coordinates plots, we can also group close lines in each axis together and start constructing a tree from the axis that have most numbers of groups with members from only one classification.

2.2 Development of Scatter Plots

According to Friendly and Denis [FD05], there is no one that is widely credited with the invention of scatter plots.

One of the earliest graphical representations of bivariate relations was presented in the work of Francis Galton (1822-1911) on correlation, regression, and heritability [FD05]. Nevertheless, as mentioned by Friendly and Denis [FD05], Galton's graphical representations were actually the graphic transcriptions of tables by using numbers or count-symbols to represent each cell's frequency. This type of representation was later called *semi-graphic* displays, the term proposed by Tukey [Tuk70]. The example of Galton's semi-graphic scatter plots is included in Appendix A. In his later article, Galton [Gal90] also described the construction of the true scatter plots. However, scatter plots had been earlier presented in other works, thus, the invention of scatter plots cannot be attributed to Galton [FD05].

In 1833, Sir John Frederick W. Herschel (1792-1871) provided four figures in his work [Her33], which were identified by Friendly and Denis [FD05] as scatter plots in the modern sense. Example of Herschel's figures is displayed in Appendix A. Even though Herschel's scatter plots had time as the horizontal axis, which can be referred as a time-series graph, Friendly and Denis [FD05] argued that since the scatter plot is used to observe the relationship between position angle and separation distance, the time in the graph can be viewed as a variable for the distance measure.

Though scatter plots had been used in various works in the19th century, the first term, *scatter diagram*, was first introduced in between 1906 and 1920 [FD05]. Slightly later, the term scatter plot began to appear [FD05]; it was defined by the *Oxford English Dictionary (OED)* [OED89] as "a diagram having two variates plotted along its two axes and in which points are placed to show the values of these variates for each of a number of subjects, so that the form of the association between the variates can be seen."

As scatter plot was originally proposed to use with bivariate data, it has been later developed to display three or more variables [FD05]. Anderson [And28] presented the use of glyph symbols to denote additional variables; their values were represented by radial lines with varying length and angle. In the 1960s, the emergence of computer-generated graphics led to several enhancements of scatter plots, including the idea to plot many smaller scatter plots together in a single display [FD05]. This technique was later defined by Tufte and Graves-Morris [TGM83] as *small multiples*.

Friendly and Denis [FD05] stated that one example of small multiples is *coplots* (or *conditioning plots*), which illustrates two focus variables together for all combination of conditioning variables. Another example is *scatter plot matrix*, which presents multiple scatter plots with all pair of variables [FD05]. Apart from these two examples, various techniques have been recently developed to show a subset of scatter plots in a scatter plot matrix, organised in different layouts [KZZM12]. Qu *et al.* [QCX$^+$07] presented a row of scatter plot cells where each cell's horizontal axis is the same as consecutive cell's vertical axis; the technique was called *Weighted Complete Graph*. On the other hand, Viau *et al.* [VMCJ10] introduced a row of scatter plot cells with the same vertical axis, and another variant called *Scatterplot Staircase*, which shows the scatter plots in a staircase arrangement, where adjacent scatter plots share an axis along their common edge.

The variants of scatter plots discussed earlier are all static design. However, dynamic scatter plots were also proposed to display high-dimensional data in scatter plots [FD05]. In 1974, Fisherkeller *et al.* [FFT74] built *PRIM-9*, which is a system to display three-dimensional (3D) scatter plots through real time motion. PRIM-9 allows users to explore up to 9 dimensional data by rotating the 3D scatter plots to any desired orientation of the 36 two-dimensional projections [FFT74]. Later in 1985, Asimov [Asi85] introduced *grand tour*, which is a method to view a sequence of two-dimensional (2D) projections of multivariate data, based on the explicit criteria to make the sequence become dense in all 2D subspaces.

Apart from the computer-generated views of scatter plots, various interactive scatter plots have been developed due to the current researches on visualisations with direct interaction to the data [FD05]. Becker and Cleveland [BC87] provided four *brushing* operations (highlight, shadow highlight, delete, and label) on a scatter plot matrix, which are dynamic

methods occurred when the mouse cursor is moved over one of the scatter plots. Jourdan *et al.* [JPKM07] proposed a method to create scatter plots with interaction and selection of data points from a standard scatter plot matrix. Sadana and Stasko [SS14] implemented a dynamic scatter plots visualisation for multi-touch interactions on a tablet computer. Additionally, many software have been provided to support the creation of interactive scatter plots [FD05], including *D3 for Data-Driven Documents* (*D3.js*) [Bos], *R* software [Fou], IBM *Statistical Package for the Social Sciences (SPSS)* software [IBMa], IBM *Many Eyes* [IBMb], and *Tableau* software [Sof].

2.3 Development of Parallel Coordinates Plots

Parallel coordinates plots had been used before the 1980s, for instance in the *'General Summary, Showing the Rank of States, by Ratios, 1880'* by Gannett and Hewes [GH83], but it became popular after Inselberg's work [Ins85] in 1985.

Unlike scatter plots, parallel coordinates plots originally have the novel approaches to visualise multivariate data. Nevertheless, many techniques have been developed to enhance their performance using colour and opacity. Fua *et al.* [FWR99] proposed *Hierarchical Parallel Coordinates (HPC)* to facilitate the exploration of the very large multivariate data through cluster-based hierarchical enhancements. Artero *et al.* [AdOL04] introduced an opacity-based rendering method to construct interactive parallel coordinates frequency and density plots in order to support the ability to identify clusters. Johansson *et al.* [JLJC06] used *high-precision textures* to represent data and applied *transfer functions (TFs)* to obtain colour-based and opacity-based rendering for parallel coordinates plots.

Various researchers had also invented the techniques to use curves instead of polylines, as it is easier to visually follow smooth curves than lines [HvW10]. Theisel [The00] modified traditional parallel coordinates plots by connecting points on adjacent axes with free-form curves instead of polylines. Moustafa and Wegman [MW02] extended the concept with the use of smooth curves that cross the axes orthogonally. Similarly, Graham and Kennedy [GK03] replaced polylines with smooth curves, but these curves have no requirement on orthogonal crossing. Zhou *et al.* [ZYQ$^+$08] further optimised the curved edges' arrangement, while varying colour and opacity according to the density of the curves. Luo *et al.* [LWZK08] bundled curves with given cluster hierarchies to provide better visual separation between clusters and reduce visual clutter.

Apart from using colour and curves, parallel coordinates plots were developed to provide interactive displays. Hauser *et al.* [HLD02] provided interactive selection on parallel coordinates plots by using angular brushing. Barlow and Stuart [BS04] animated the movement of data with time-varying variables in parallel coordinates plots using *Animator software.*

Johansson *et al.* [JLJC06] applied a technique called *feature animation* to convey statistical properties of clusters in parallel coordinates plots.

Additionally, parallel coordinates plots had been extended into the 3D space. Honda and Nakano [HN06] presented 3D parallel coordinates plots by adding a spatial third orthogonal axis. Streit *et al.* [SEÖ+06b] developed 3D parallel coordinates plots to use in the analytical cytology field. Johansson *et al.* [JCJ05] introduced *3D clustered multi-relational parallel coordinates plots (CMRPC)* by using a technique called *relation spacing* to position the axes due to their interesting relations.

Furthermore, many methods had combined scatter plots into parallel coordinates plots, yielding the hybrid visualisation techniques. Holten and van Wijk [HvW10] embedded scatter plots between each pair of adjacent parallel coordinates plots' axes. In contrast, Yuan *et al.* [YGX+09] developed another visualisation that seamlessly integrates scattering point representation into the curves of parallel coordinates plots. Examples of these parallel coordinates plots' variants, including the hybrid visualisations, are demonstrated in Appendix A.

Presently, the available software to create parallel coordinates plots include D3.js [Bos], R software [Fou], IBM SPSS software [IBMa], *X-dimensional Data Analysis Tool (XDAT)* [xda], *GGobi* program [GGo], and Macrofocus *High-D tool* [Gmb].

2.4 Evaluation in Information Visualisation

Even though the evaluations in information visualisation field have been identified as important and needed over several years [LBI+11], only a few studies have been done as these evaluations are considered as challenging [Pla04, ED06]. Lam *et al.* [LBI+11] explained that apart from general evaluations' difficulties, the evaluations in information visualisation have further challenges as the data analysis process and its outputs are difficult to design and measure.

The following subsections first explain several evaluation approaches in information visualisation, followed by specific methods to evaluate visualisation techniques. The last subsection further provides studies that have been conducted to compare user performance between scatter plots and parallel coordinates plots.

2.4.1 Evaluation Approaches

There are various types of evaluation methodologies available, such as field and laboratory observation, interview and questionnaire, case study, usability test, controlled experiment, log analysis, and quality metric [LBI+11]. In information visualisation, these evaluation approaches have been classified in many several ways.

Plaisant [Pla04] categorised the evaluation into four areas:

1. *Controlled experiments comparing design elements*: compare specific components' design or the mappings of information to visual coding.

2. *Usability evaluation of a tool*: provide feedback on users' problems and redesign the visualisation technique.

3. *Controlled experiments comparing two or more tools*: compare different visualisation techniques.

4. *Case studies of tools in realistic settings*: report users' performances with real environment and tasks.

According to Plaisant [Pla04], the fourth evaluation is the least common type of studies, as it is time consuming and may not be generalisable to other context, whereas controlled experiment is one of the backbones of evaluation.

Barkhuus and Rode [BR07] characterised the evaluation according to the type of collected data (qualitative or quantitative), and the collection method whether it involves user or not (empirical or analytic).

As summarised by Lam *et al.* [LBI+11], other researches provided classification of the evaluation with respect to evaluation goal [And08, ED06, HR00], research strategies [Mcg95, IH01], design stages [Mun09, And08], and evaluation scope [TC05]. The diversity of these classifications demonstrates the complexity of the existing evaluation approaches, and thus indicates that it is hard to select the most appropriate approaches for each evaluation [LBI+11].

Lam *et al.* [LBI+11] further provided guidelines on how to choose the evaluation methodologies according to seven different evaluation scenarios:

1. *Evaluating environments and work practices*: to obtain formal requirements from a group of users in order to design a suitable visualisation technique for their work and information processing.

2. *Evaluating visual data analysis and reasoning*: to assess how a visualisation technique supports users in generating relevant information for a specific domain.

3. *Evaluating communication through visualisation*: to understand how a visualisation technique aids in conveying messages to relevant audiences.

4. *Evaluating collaborative data analysis*: to study how a visualisation technique allows for collaboration in order to provide joint conclusion or discovery.

5. *Evaluating user performance*: to test how specific features in a visualisation technique affect objectively measurable user performance.

6. *Evaluating user experience*: to identify users' subjective feedback for a visualisation technique in either a short or a long time span.

7. *Automated evaluation of visualisation*: to investigate aspects of a visualisation technique that can be measured through an automatic computational procedure.

Lam *et al.* [LBI+11] emphasised that the first four scenarios aim to understand the processes and roles of a visualisation technique, while the last three scenarios focus on the visualisation techniques itself with the goal to test the designs and usability issues. In any evaluation, the evaluation approaches should be selected based on its goal [LBI+11].

2.4.2 Evaluation of Visualisation Techniques

The evaluations of different visualisation techniques are required as their performances are not equally suited to all visualisation tasks [JFLC08]. These evaluations can be performed in two main ways: evaluate each visualisation technique to find advantages and disadvantages, and compare the user performance among different visualisation techniques.

For the first case, the evaluation falls into the second category of Plaisant's classification [Pla04] and the sixth category of Lam *et al.*'s classification [LBI+11]. Plaisant [Pla04] indicated that usability test should be used, while Lam *et al.* [LBI+11] identified other methods including informal evaluation, field observation, and questionnaire. Previous researches have been conducted using these methods.

Freitas *et al.* [FLC+02] evaluated *Bifocal Browser*, a visualisation technique to explore hierarchies in a node-edge diagram, using a usability test with their proposed criteria, and detected problems regarding the visual representation and interaction mechanisms. Pillat *et al.* [PVF05] performed a usability test to evaluate parallel coordinates plots and *Radviz*, and reported the usability problems of Radviz technique including location distinction and legibility problems. Cuzzocrea and Zall [CZ13] conducted a survey research involving technical experts who have experienced parallel coordinates plots to analyse its advantages and disadvantages. The results in [CZ13] showed that the parallel coordinates plots' advantages including integrity, connectivity, and consequences of results, while the disadvantages consist of over-plotting and lack of correlation analysis, which can be overcame by using several possible techniques, such as the use of high-precision textures and algorithms to filter out less relevant data.

In the second case, the evaluation falls into the third category of Plaisant's classification [Pla04] and the fifth category of Lam *et al.*'s classification [LBI+11]. Both researches [Pla04,

LBI+11] suggested that the suitable approach is to conduct a controlled experiment, which is an empirical method used to evaluate user performance. Lam *et al.* [LBI+11] mentioned that the most commonly objective measurement includes accuracy and response time, and the results are commonly tested by the analysis of variance to provide statistical evidence. This empirical research method, or user study, has been widely used and applied to many researches in different application areas.

Liston *et al.* [LFK00] used a user study to compare traditional paper-based method to view project information with the two visualisation techniques, highlight and overlay, and demonstrated that these visualisation techniques are useful. Lee *et al.* [LRB03] presented an empirical comparison between two glyph visualisations, *Chernoff faces* and star glyphs, and two spatial visualisations, common features model and distinctive features model, and concluded that the common spatial visualisation yields the best performance. Luo *et al.* [LWZK08] performed an empirical study comparing the user performance between parallel coordinates plots with traditional polylines and with the proposed bundled curve technique; the results indicated that the bundled curves support correlation analysis as effectively as in the polylines, but are more capable of revealing clusters. Johansson *et al.* [JFLC08] investigated the *visual quality* of standard 2D parallel coordinates plots and 3D parallel coordinates plots, and identified that the latter technique has lower level of noise in the data sets with about 11 variables or less. Sanyal *et al.* [SZB+09] empirically evaluated the user performance of four uncertainty visualisation techniques, traditional error bars, scaled size of glyphs, colour-mapping on glyphs, and colour-mapping on data surface, and found that scaled size glyphs and colour-mapping glyphs perform reasonably well, while error bars yield lower performance for all tasks.

Goodall [Goo09a] conducted an empirical evaluation in the cyber security area, comparing traditional textual interface and a visualisation application in order to analyse network packet captures; the results revealed that the visualisation technique is preferred as it yields higher accuracy, faster performance, and provide higher number of insights. Forsberg *et al.* [FCL09] demonstrated an empirical comparison among four visualisation techniques using variations in integral curve renderings (line and tube) and viewing conditions (monoscopic and stereoscopic). The results in [FCL09] indicated that the line methods with stereoscopic viewing yields faster performance and is preferred for participants, but its accuracy is not higher than other techniques. Azhar and Rissanen [AR11] empirically compared the user performance to filter alarm data in the typical alarm lists and an application of parallel coordinates plots, and reported that the parallel coordinates plots reduce alarm filtering time and human mistakes. Fu *et al.* [FNS14] used an eye tracking empirical study to compare user performance in the two ontology visualisation techniques, indented list and graph, and

concluded that indented lists yield higher performance to search for information but graph holds advantages in processing information.

2.4.3 Evaluation of Scatter Plots and Parallel Coordinates Plots

Scatter plots and parallel coordinates plots are the two common visualisation techniques used to represent multivariate data [KZZM12]. Even though several features have been proposed and developed to increase the performance of these two techniques, only a few researches have been evaluated their user performances. Many comparisons between the two techniques have not yet conducted, and thus the respective advantages of these two techniques for different visualisation tasks are still doubtful [KZZM12].

Henley *et al.* [HHB07] evaluated the two visualisation techniques for a specific application domain, genome comparison, and concluded that scatter plots are preferred in overall performance. Nonetheless, the results apply only with genome analysis and are not generalised to other domains. Li *et al.* [LMW08] revealed that scatter plots are more effective than parallel coordinates plots in supporting visual correlation analysis between two variables. However, the analysis was limited to only bivariate data. Holten and Van Wijk [HvW10] compared variants of parallel coordinates plots in cluster identification performance, and found that the variation with embedded scatter plots noticeably outperform other variants. They suggested that scatter plots are superior to parallel coordinates plots, though there was no comparison with the scatter plots alone.

To our knowledge, the only previous study on user performance for scatter plots and parallel coordinates plots on multivariate data analysis was performed by Kuang *et al.* [KZZM12]. They presented an experimental comparison of user performance in value retrieval tasks between three variants of scatter plots and the baseline parallel coordinates plots. The results indicated that parallel coordinates plots hold an advantage in low dimensionality and density data sets, whereas scatter plots perform better in the data sets with higher dimensionality and density [KZZM12]. Still, the research involved only one experiment, the retrieval task, which is a less critical visualisation task.

Although the advantages of parallel coordinates plots are still unclear according to the previous researches' results, several other important visualisation tasks have not been tested yet. Consequently, we will conduct an empirical study to compare the user performance in the more complex visualisation tasks. Our hypothesis is that parallel coordinates plots will show advantages in some of these tasks.

2.5 Empirical Study

Empirical study is a commonly used method to compare user performance between different visualisation techniques. It is a research method based on actual and objective observations or experiments in order to answer particular research questions [Goo09b].

Initially, a research question is formulated and then transformed to hypotheses, which can be empirically tested [Goo09b]. Then, the hypotheses lead to the design of the experiment, including the process to determine variables in the experiment and identify the methods to minimise confounding effects [Goo09b].

After the design step, an experiment is conducted, and data are collected and measured. To understand the research outcomes, statistical analyses, including descriptive statistics and inferential statistics, are performed [Goo09b]. These data analyses are used to evaluate the hypotheses.

Following subsections will explain these processes in detail, starting from formulating research questions and hypotheses, identifying variables in the experiment, minimising confounding effects, and performing statistical analyses.

2.5.1 Research Questions and Hypotheses

Goodwin [Goo09b] explained that an empirical study always begin with a research question, normally referred to as an *empirical question*, which is a question that is answerable with objective data. Research questions usually include terms that are precisely defined to allow specific predictions to be made [Goo09b].

The research question is then developed into a hypothesis, which is a prediction of the study's results, defining an expected relationship between variables [Goo09b].

2.5.2 Variables in Experiments

In an experiment, there are three involved variables, *independent variables*, *dependent variables*, and *controlled variables* [Goo09b].

Independent variables are the factor of interest; it will be studied whether it influences other factors or not [Goo09b]. Goodwin [Goo09b] expressed that independent variables must have at least two levels in order to make a comparison between them. **Dependent variables**, on the other hand, are the factors that will be measured during the study [Goo09b]. An experiment is conducted to investigate the effect of the independent variables on the dependent variables. For comparing user performance among visualisation techniques, the visualisation technique is the independent variable, while the user performance is the dependent variable.

Apart from these two variables, there are other factors that are not of interest but might affect the dependent variables, called **controlled variables** [Goo09b]. These variables must be held constant to avoid misleading results. If these variables are not controlled, they can influence the measured factors, leading to the *confounding effects* [Goo09b]. As described by Goodwin [Goo09b], the confounding effect occurs when any uncontrolled variable changes at the same time as the independent variable, and therefore the effects from the two variables cannot be separated. As a consequence, the results of the study cannot be determined whether they are due to independent variables or *confounding variables*, and thus cannot be interpreted [Goo09b].

2.5.3 Confounding Effects

The study comparing different visualisation techniques generally use a *within-subjects design (repeated-measures design)*, in which the same participants provide data to all levels of the independent variable (all visualisation techniques) [Goo09b].

According to Goodwin [Goo09b], a major problem in this type of design is the **sequence** or **order effect**, which is the effect where the first part of the study influences user performance in the later part. Earlier trials might enhance the participants' performance on later trials since they gain some experiences, creating some positive effects [Goo09b]. Meanwhile, repeated trials may create gradual fatigue or boredom, leading to the decreasing in performance on later trials, which is a negative effect [Goo09b]. As a result, Goodwin [Goo09b] summarised that the user performance may progressively changes from trial to trial, referred to as *progressive effect*, and different sequences of trials may yield different performance results, called *carryover effect*. Therefore, providing the first visualisation technique before the second might affect the user performance differently from displaying the second before the first [Goo09b].

The common solution to control the sequence effects is to use *complete counterbalancing*, in which every possible sequence will be used once [Goo09b]. The number of total sequences needed is $n!$ where n is the number of levels of the independent variable.

2.5.4 Statistical Analysis

Statistical analysis is an important method to analyse the study's results. Goodwin [Goo09b] stated that there are two main types of statistical analysis, *descriptive* and *inferential*. Descriptive statistics provide summary results of the data collected from the sample participants, while inferential statistics inform whether the results are due to chance factors or reflect genuine relationship that can be applied to wider population [Goo09b].

Descriptive statistics include measures of central tendency and variability [Goo09b]. Measures of central tendency consist of mean, median, and mode. *Mean* is the most common measure, which is calculated by summing all values and dividing by the number of data. *Median* is the value in the exact middle of the sorted data set. *Mode* is the value occurred most frequently in the data set. For measures of variability, there are range, standard deviation, and variance. *Range* is the difference between the lowest and highest values in the data set. *Standard deviation* is the average amount of deviation from each value to the mean of the data set. *Variance* is the standard deviation squared.

Inferential statistics usually involve *hypothesis testing* [Goo09b]. The first process in hypothesis testing is to create the *null hypothesis (H_0)*, which is the assumption that there is no difference in the performance mean between different independent variables. As described by Goodwin [Goo09b], an inferential analysis can only have two results, either rejecting the null hypothesis or failing to reject it. Failing to reject the null hypothesis can be inferred that any differences occurred in the study are due to chance, whereas rejecting the null hypothesis means that a real effect exists, and hence the results can be generalised [Goo09b].

Goodwin [Goo09b] emphasised that the null hypothesis can only be rejected with some degree of confidence called *significance level (α)*. The significance level is usually set at .05 ($\alpha = .05$), as according to Sir Ronald Aylmer Fisher [Fis25], the developer of *ANOVA*. Fisher [Fis25, p. 44] presented that "The value for which P =.05, or 1 in 20, is 1.96 or nearly 2 ; it is convenient to take this point as a limit in judging whether a deviation is to be considered significant or not".

To reject the hypothesis, the obtained value from the hypothesis testing called *p-value* must be less than the significance level [Sel14]. The p-value can be described as the probability that the outcome of the experiment is like the observed results or becomes more extreme (shows clearer difference), if the null hypothesis holds [Joh99]. Johnson [Joh99, p. 764] expressed that the p-value can be written as "p = Pr[observed or more extreme data | H_0]". As mentioned by Seltman [Sel14], if the p-value is less than the significance level, we should not say that the null hypothesis is true, and thus the null hypothesis should be rejected. To summarise, if the obtained p-value $< .05$, we should reject the null hypothesis, otherwise do not reject the null hypothesis. Furthermore, as the null hypothesis states that the mean of each group is equal, rejecting the null hypothesis leads us to the conclusion that there is a *statistically significant* difference among each group [Sel14].

The goal of the hypothesis testing is to reject the null hypothesis when a true difference occurs [Goo09b]. Nevertheless, the null hypothesis may be rejected even though there is no real effect (*Type I error*), and there may be no significant difference found in the study even

though a true effect exists (*Type II error*). These errors should be minimised through careful design of the experiment. The inferential statistical analyses include ANOVA analysis and *t-test analysis*, which will be described in the following subsections.

2.5.4.1 ANOVA Analysis

Analysis of Variance (ANOVA) is a statistical model that is used to test for significant differences among the means of several groups [Sel14]. There are many types of ANOVA analysis, but the suitable method for comparing user performance between different visualisation techniques is the one-way repeated-measures ANOVA analysis, which focuses on one quantitative dependent variable and one categorical independent variable that has two or more levels [KHK02, Sel14].

As a parametric test, ANOVA makes assumptions that each sample has been drawn from a normally distributed population, these samples have equal variance, and the dependent variable is interval or ratio [KHK02]. Besides, the repeated-measures ANOVA makes one additional assumption, which is referred to as *sphericity assumption* [KHK02]. For each participant, we can compute the performance differences between each pair of the groups, for instance between scatter plots and parallel coordinates plots. Then, for each pair of groups, the set of these differences of all participants can be used to calculate the variance. Sphericity assumption specified that these variances of all pairs must be equal. [KHK02].

According to Atkinson [Atk11], the violation of the sphericity assumption is serious for the repeated-measures ANOVA, as it will increase the Type I errors. Nevertheless, we can test the data for sphericity using a standard test called *Mauchly's Test of Sphericity*, and made corrections accordingly if the violations occur [Atk11].

Since the null hypothesis of the Mauchly's Test of Sphericity is that the variances of the differences are equal [Atk11], if the test shows $p > .05$, we do not reject the null hypothesis and report that the assumption of sphericity had been met. Otherwise, we need to use *Greenhouse-Geisser* or *Huynh-Feldt* Corrections, which give different value for the departure from the sphericity called *epsilon* (ϵ) [Atk11]. Atkinson [Atk11] mentioned that the epsilon has the value from 0 to 1, with 1 indicating the sphericity of the data. The Huynd-Feldt correction tends to overestimate epsilon, while the Greenhouse-Geisser correction tends to underestimate epsilon when the value is close to 1 [Atk11]. As a consequence, Girden [Gir91] suggested that we should use the Greenhouse-Geisser correction if its epsilon is less than 0.75, and use Huynd-Feldt correction otherwise.

Seltman [Sel14] explained that the p-value is obtained from the F ratio; p-value is the chance of getting an F-statistic greater than or equal to the computed F ratio when the null hypothesis holds. The F ratio in one-way repeated-measures ANOVA is defined as

$$F = \frac{MS_{between}}{MS_{error}}$$

where $MS_{between}$ denotes the mean square between groups and MS_{error} denotes the mean square error (MSE) [Sel14]. Each mean square is calculated from

$$MS = \frac{SS}{df}$$

where SS is the sum of squared deviations from the mean, and df is the degree of freedom [Sel14].

Seltman [Sel14] clarified that $df_{between}$ equals to $k - 1$, while df_{error} equals to $(n - 1)(k - 1)$ where k is the number of group and n is the number of samples in each group. If Greenhouse-Geisser or Huynh-Feldt Corrections are used, the degrees of freedoms are multiplied by its epsilon [Atk11]. The results from the ANOVA analysis is usually reported as *(F($df_{subjects}, df_{error}$) = F ratio , p = p-value).*

2.5.4.2 *t*-test Analysis

Even though ANOVA analysis can report the significant differences among several groups, it does not indicate that which group is the source of the main effects [Sel14]. Therefore, *t*-test analysis, a post-hoc test, should be performed to provide pairwise comparison among these groups.

The purpose of this technique is to test for significant differences between each pair of the visualisation techniques. Similar to ANOVA analysis, the null hypothesis of *t*-test analysis is that there is no significant difference between the performance mean of the two techniques in the pairs.

While performing *t*-test analyses for several pairs, the tests are performed multiple times on a single set of data, and thus the chance of obtaining Type I errors increases [Nap12]. To reduce these error rates, *Bonferroni correction*, the simplest and one of the most useful procedures [Sel14], must be applied. The adjusted p-value will identify whether the null hypothesis is rejected or not.

Chapter 3

Methodology

There are many approaches to evaluate the visualisation techniques, as described in Section 2.4, but the most common technique is an empirical study [Pla04, LBI+11]. Therefore, we will perform an empirical study to compare three visualisation techniques, including data table (hereinafter DT), scatter plots (hereinafter SCP), and parallel coordinates plots (hereinafter PCP).

We will first formulate four hypotheses for the four visualisation tasks, which consist of value retrieval, clustering, outlier detection, and change detection. To evaluate these hypotheses, we will design and conduct a controlled experiment with three independent variables (visualisation technique, visualisation task, and level of task difficulty) and two dependent variables (accuracy and response time). The results from the experiment will then be analysed using descriptive statistics and inferential statistics including ANOVA and t-test analyses to provide statistical evidence. Apart from these objective measures, we will also measure the subjective rating of each technique's effectiveness.

In this chapter, we will first describe our hypotheses, followed by the variables and measurement metrics involved in the experiment. The discussion will then move to the techniques for the analyses. The design of the study will be later explained in Chapter 4.

3.1 Research Question and Hypotheses

The research question in our study is *'Do DT, SCP, and PCP affect the user performance in different visualisation tasks?'*.

To answer this question, we have to formulate the hypotheses for each visualisation task, and later evaluate them during the analyses. The visualisation tasks in our study include value retrieval, clustering, outlier detection, and change detection task; the decision on task selection will be explained in Section 3.2.

We propose four hypotheses for the four visualisation tasks respectively:

H1: DT $\succ$ PCP $\succ$ SCP

H2: PCP $\succ$ SCP $\succ$ DT

H3: PCP $\succ$ SCP $\succ$ DT

H4: PCP $\succ$ SCP $\succ$ DT

where $\succ$ is used to represent that the technique on the left hand side of the operator yields higher performance than the technique on the right.

For the hypothesis in value retrieval task (H1), we state that PCP will be better than SCP, as according to Kuang *et al.* [KZZM12] since we will use low dimensionality and density data sets. However, we believe that DT will outperform these two visualisation techniques, as it directly provides values of the data.

The hypotheses for clustering task (H2), outlier detection task (H3), and change detection task (H4) are the same, due to our expectation that PCP will provide better results than the other two techniques, while DT will yield lowest performance in these tasks.

3.2 Variables in Experiments

Similar to most empirical studies, several variables may potentially affect the user performance. Therefore, it is common to focus a study on a small number of variables, while controlling other variables to remain constant. This is in order to reduce the confounding effects, as explained in Section 2.5, and provide comparable results. The following subsections will explain independent variables, dependent variables, and controlled variables involved in our study.

3.2.1 Independent Variables

This study has three independent variables: visualisation technique, visualisation task, and level of task difficulty.

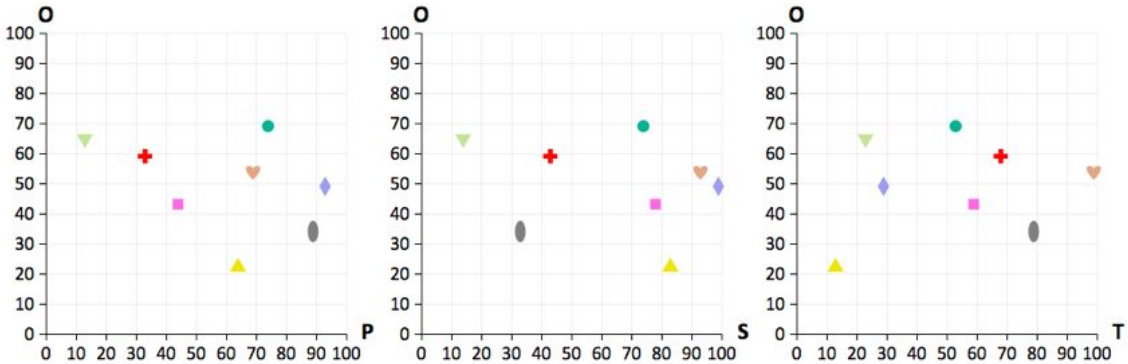
3.2.1.1 Visualisation Technique

Our experiment involves three visualisation techniques, which are DT, SCP, and PCP.

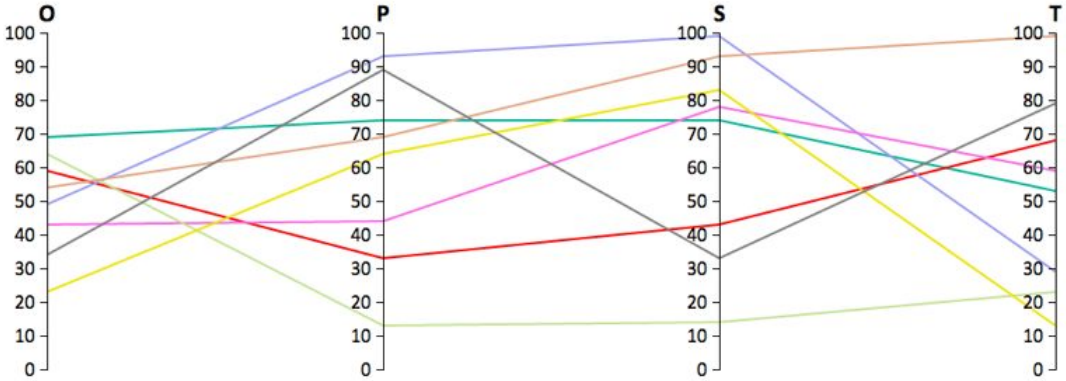
SCP and PCP have many variants, as introduced in Section 2.2 and Section 2.3. Nevertheless, the standard design of these two techniques have not been previously evaluated in many visualisation tasks. Therefore, we focus our study on the traditional non-interactive variants as these basic designs should be tested first.

O	P	S	T
69	74	74	53
59	33	43	68
49	93	99	29
43	44	78	59
64	13	14	23
23	64	83	13
54	69	93	99
34	89	33	79

(a) DT.



(b) SCP.



(c) PCP.

Figure 3.1: The three visualisation techniques.

Considering the static designs, DT and PCP both only have one layout (Figure 3.1(a) and Figure 3.1(c) respectively), whereas SCP provides several methods to present multivariate data. Since the space utilisation for DT and PCP are linear with the number of variables, we need the SCP variant that also has linear space requirements. This is in order to constrain the same space occupation for the three visualisation techniques to ensure a fair comparison analysis.

Kuang *et al.* [KZZM12] have compared user performance in value retrieval tasks between three variants of SCP, and the results indicated that *SCP-common* outperforms the other two techniques. Thus, we select SCP-common as the layout for the SCP in our study. SCP-common, the technique's name used in [KZZM12], is a SCP variant introduced by Viau *et al.* [VMCJ10] that presents a single row of SCP taken from the standard scatter plot matrix, providing a common vertical axis for all of its cells (Figure 3.1(b)). In our study, the first variable of each data is selected as the common vertical axis. Choosing this variant, all three visualisation techniques require the same $\mathcal{O}(NL^2)$ space, where N is the number of variables and L is the length of each axis [VMCJ10].

3.2.1.2 Visualisation Task

Different visualisation tasks may affect the performance of each visualisation technique [Pla04]. In an analytic activity, there are many visualisation tasks that are useful and commonly used, including but not limited to value retrieval, clustering, correlation analysis, outlier detection, change detection, and classification tree generation. Following Plaisant's suggestion [Pla04], our study includes some complex tasks and reports the results of each task separately.

Value retrieval task is one of the fundamental tasks. Kuang *et al.* [KZZM12] have compared user performance between SCP and PCP for this task. Nevertheless, we believe that DT may have advantages over these two visualisation techniques, and hence include this task in our experiment.

Clustering task is another important task in information visualisation. Holten and van Wijk [HvW10] have compared variants of PCP in this task performance, but there was no comparison with other visualisation techniques. Therefore, clustering task is another task in our study.

Correlation analysis is also a main task for analytic activity. However, it is mostly used to discover the relationship between two variables, and thus not suitable for our analysis on multivariate data.

Outlier detection is one of the most crucial visualisation tasks. There were many researches comparing the methods and algorithms for detecting outliers [CK10, Seo06a,

PJ01, MKMM07], but, to our knowledge, there was no comparison on using visualisation techniques to do this task. Consequently, outlier detection task is also involved in the study.

Change detection is another essential task in data analysis. Nonetheless, to our knowledge, there was no empirical study on the performance of each visualisation technique for this task. We then add change detection task in our experiment.

Classification tree generation is useful for data mining and decision analysis. Yet, it is not a common task for many users and may require a lot of learning efforts. For this reason, we decide not to include this task in the experiment.

In conclusion, our study consists of four visualisation tasks, which are value retrieval task, clustering task, outlier detection task, and change detection task.

3.2.1.3 Level of Task Difficulty

To do a visualisation task, the level of its difficulty may affect the user performance on each visualisation technique. It is possible that a participant may perform well using one visualisation technique on an easy task, but may find it better to use another technique on the more difficult tasks. Subsequently, we vary the task difficulty into three levels in our experiment, including easy level, medium level, and hard level.

Furthermore, if the task is too easy, most participants may be able to perform well with any visualisation technique provided, and hence it will be difficult to compare the performance between different techniques. On the other hand, if the task is too hard, the participants may start to guess instead of trying to find the correct answer, and thus the results will not be effective. In consequence, we have to design a proper range for each task difficulty.

The definition of each difficulty level will be defined separately for each visualisation task, and will be explained in Section 4.2 to Section 4.5.

3.2.2 Dependent Variables

Accuracy and response time are the two common dependent variables used in empirical studies. In this study, we use these two variables to measure the effectiveness and the efficiency of each visualisation technique. Accuracy relates to how useful the technique is to complete a task, and thus indicate its effectiveness. Response time, on the contrary, relates to how well the participants can perform a task, and hence determine each technique's efficiency.

3.2.2.1 Accuracy

Accuracy will be measured as a *Boolean* category of right and wrong. For each question, if the participants choose the correct answer then the accuracy will be recorded as 1, otherwise it will be 0.

For value retrieval task, we have an option to follow Kuang *et al.* [KZZM12] by allowing the participants to type in the answer and measure the accuracy as the absolute error difference. Nevertheless, this method can create additional time in typing, which varies among each individual and may affect the measured response time. Kuang *et al.* [KZZM12] solved this problem by measuring the time until the participants indicate that they have found the answer. The visualisation image will be then disappeared and the participants can take as much time to key in the value without affecting the measurement of response time [KZZM12]. Yet, it can create typographical error and the participants need to memorise the answer. Instead, we provide optional answers and record the accuracy in terms of Boolean variable. We also use this manner for other visualisation tasks to ensure consistency.

3.2.2.2 Response Time

Response time will be measured, in milliseconds, after the visualisation image is shown on the screen until the participants select an answer from the optional choices.

To avoid interference from the time spent on reading question and optional answers, we exclude these time from the measured response time by providing these information beforehand. The detail of the timing procedure will be later explained in Section 4.1.

3.2.3 Controlled Variables

There are three controlled variables in this study, including visualisation image, level of distractor difficulty, and position of optional answers.

3.2.3.1 Visualisation Image

Visualisation image is one of the most important variables in the study that we have to control to ensure that DT, SCP, and PCP have equal number of stimuli.

These stimuli are grouped into sets of *triple*; each triple consisting of one stimuli for each visualisation technique. The stimuli in each triple are controlled to have similar information, similar format, and similar complexity. Section 4.6 will go into detail of the stimulus design.

3.2.3.2 Level of Distractor Difficulty

Four optional answers will be provided for each question; one of them is a correct answer, and the other three are the distractors.

We control these distractors to have three different levels of difficulty in each question, which are easy level, medium level, and hard level. The definition of each distractor difficulty will be defined separately for each task, and will be explained in Section 4.2 to Section 4.5.

3.2.3.3 Position of Optional Answers

The positions of the correct answer and its distractors may lead to a confounding effect, since the participants need to move the mouse in different amount to different position.

For this reason, we control the stimuli in the same triple to have the same position of the four optional answers. This is in order to ensure consistency within each triple of stimuli.

3.3 Measurement Metrics

Our experiment involves two measurement metrics: objective measure and subjective measure.

3.3.1 Objective Measure

To analyse user performance, the objective measures in this study include accuracy and response time. Accuracy will be used to analyse effectiveness of the visualisation technique, whereas response time will be used to analyse its efficiency, as discussed in Section 3.2.

3.3.2 Subjective Measure

In this study, we will use effectiveness rating as the subjective measure for user performance. After completing the experiment, the participants will be asked to rate the effectiveness of each visualisation technique, according to each visualisation task.

For the rating technique, we will use the five-level *Likert scale*, which was introduced by Likert [Lik32]. The Likert scale is a scale that offers a fixed choice of responses to measure the levels of agreement with a particular statement [Bow97, BG97]. In our study, the Likert scale provides a choice of five responses including *not at all effective*, *slightly effective*, *moderately effective*, *very effective*, and *extremely effective*.

3.4 Techniques for Analyses

As stated earlier in this chapter, our study involves three independents variables (visualisation technique, visualisation tasks, and level of task difficulty), and two dependent variables (accuracy and response time).

To evaluate the four hypotheses provided in Section 3.1, we will analyse the user performance in the four visualisation tasks separately. Furthermore, to investigate the effect of the tasks' level of difficulty, the performance for each level will also be individually analysed. The hierarchy of the performance analyses is illustrated in Figure 3.2. Besides, in a single performance analysis, accuracy and response time will be analysed separately; the independent variable in each analysis is the visualisation technique, which consists of three levels: DT, SCP, and PCP.

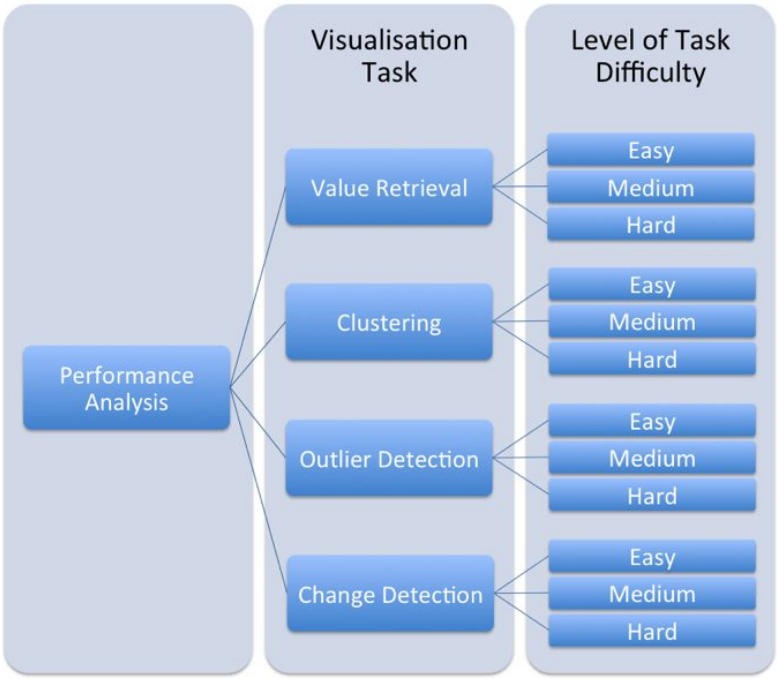


Figure 3.2: Hierarchy of performance analyses.

In order to ensure a fair comparison, we will apply a set of formal analyses as follows.

Firstly, we will use **one-way repeated-measures ANOVA analysis**, described in Section 2.5, to test whether there is a significant main effect of the visualisation techniques on the performance. The null hypothesis for our analysis is that there is no difference of effectiveness among the three visualisation techniques, which can be written as $H_0 : \mu_t = \mu_s = \mu_p$ where μ_t, μ_s, and μ_p are the performance mean for DT, SCP, and PCP respectively.

Secondly, if ANOVA reports that a significant effect exists, we will further perform ***t*-test analysis with Bonferroni correction** to examine the source of the main effect and

provide statistical evidence for the pairwise comparison results. The null hypotheses are $H_0 : \mu_t = \mu_s$, $H_0 : \mu_t = \mu_p$ and $H_0 : \mu_s = \mu_p$ for each pair of the three techniques.

The significance level of these two analyses will be set to $\alpha = .05$, as discussed in Section 2.5. In this study, we will use R [Fou], one of the free software, to perform both analyses.

We will report the study's results in pairwise comparison among the three techniques by using '[ ]' to enclose techniques with no significant difference in performance ($p > .05$), and use '$\succ$' to show that the technique on the left hand side of the operator is significantly better than the technique on the right ($p < .05$). Furthermore, the value in '( )' after each technique will indicate the average performance for the technique.

For the accuracy result, the performance will range from 0 to 6 in the all-level analyses, as there are 6 stimuli in total, whereas it will range from 0 to 2 in the analyses of each level of difficulty since there are 2 stimuli in each level. The number of stimuli in each task will be later explained in Section 4.1

Chapter 4

User Study Design

User study design is the most important process for an empirical study. We carefully design the study to reduce biases and confounding effects, which can mislead the experiment's results. The user study design consists of three main elements, task design, stimulus design, and software design.

In this chapter, we first describe the overview of the user study design, followed by the design for each visualisation task. The discussion then moves to the stimulus design and the software design respectively.

4.1 Design Overview

When conducting an empirical study, it is common that there are many confounding effects involving, as stated in Section 2.5. Though such confounding effects cannot be eliminated completely, the user study design should be developed in a way to reduce the confounding effects to the level that they will not remarkably affect the participants' performance.

4.1.1 Task Design Overview

In this study, all visualisation tasks will be performed on multivariate data, and all data in our experiment will have **four variables**. In general, it is possible for some variables to have nominal, ordinal, or interval data, as explained in Section 2.1. Nevertheless, we focus our study on **ratio data**, which is the most general case.

For each visualisation task, we classify each trial into three levels of difficulty, including easy level, medium level, and hard level. This is in order to observe the user performance on different levels of difficulty.

To collect user performance in each trial, we will use the **multiple choice** question-and-answer format since it is familiar to most participants, and thus require little learning

effort. Each trial will have one question and four optional answers, consisting of a correct answer, an easy-level distractor, a medium-level distractor, and a hard-level distractor.

The definition of these levels of difficulty, both for task and distractors, will be given separately for each visualisation task in Section 4.2 to Section 4.5.

4.1.2 Stimulus Design Overview

In order to minimise the confounding effect due to unexpected outliers in a stimulus design, we have to ensure that there is an **adequate number of stimuli** for each visualisation task. In consequence, each visualisation task will have 18 stimuli; 6 stimuli for DT, 6 stimuli for SCP, and another 6 stimuli for PCP. Furthermore, there will be 3 additional stimuli for each task, one for each visualisation technique, for the training process. Since the study has 4 visualisation tasks, there will be 84 stimuli in total; 72 stimuli for the main experiment and 12 stimuli for the training session.

We will reduce biases in the design by organising the stimuli design in a **structured manner**. In each visualisation task, the 18 stimuli will be divided into three groups of 6 stimuli each; 2 stimuli for DT, 2 stimuli for SCP, and 2 stimuli for PCP. The stimuli in each group will have the same level of task difficulty, either in easy, medium, or hard level. However, even though the stimuli are designed in a structure, they will be displayed in the actual trials through pseudo-randomisation; its scheme will be explained in Section 4.7.

Furthermore, all stimuli will be designed in **triples**, with one member from each type of visualisation techniques. In order to ensure that we conduct a fair comparison, the stimuli in each tuple will be designed to have similar information and complexity. Besides, all stimuli in the same triple will have the **same position of optional answers** in order to avoid order bias.

For each trial, one visualisation image will be shown on the screen, with a question and four optional answers displayed below. Each question will be provided with a task category in front. For example, Figure 4.1(a) has the word 'Values' in front of the question indicating value retrieval task. Other words include 'Clusters', 'Outliers', and 'Changes'. The four optional answers will be positioned in the same row beneath the question. On top of the visualisation image, there will be a progress bar and a countdown timer. The progress bar will be used to suggest what proportion of the time has been used, whereas the countdown timer will be used to exhibit the exact amount of time left, shown in seconds.

The information of the general design will be described in Section 4.6, while the detail of the design for each visualisation task will be explained in Section 4.2 to Section 4.5. All stimuli, together with questions and optional answers, can be found in Appendix C.

4.1.3 Software Design Overview

The software is designed to collect accurate user performance on different visualisation tasks.

For each trial, a question and four optional answers will be first shown on the screen. After the participants finish reading the questions and answers, they are required to click 'show picture' button. After which, a visualisation image will be displayed and the timer will be started. We exclude these **reading time** from the measured response time as different participants may use different time to read. Nevertheless, we will record the reading time separately.

As the timing continues, the progress bar and the countdown timer at the top of the screen will display the proportion of time used and the exact amount of time left, in seconds, respectively. The timers will be stopped when the participants select an answer from the four optional choices. After which, the amount of time used will be recorded as the **response time**, in milliseconds, along with the selected answer, which cannot be changed after one choice is selected. If the participants do not select an answer within the limited time, the visualisation image, the question, and its optional answers will disappear. The participants will then be asked to move on to the next trial.

The order of trials in the software will be pseudo-randomised in order to minimise the sequence effects, as introduced in Section 2.5. Additionally, between each section, the participants will be allowed to take a short break in order to reduce fatigue and boredom. The explanation on these techniques will be later discussed in Section 4.7.

4.2 Design for Value Retrieval Task

Following Kuang *et al.* [KZZM12], we define the value retrieval task in the context of multivariate data. Given a numerical value of one variable, the task for the participants is *'to find the numerical value of another variable of the same data point.'*

4.2.1 Task Design

This task can be performed by first locating the given variable and searching for the data point corresponding to the given value. Then, the participants may trace the data point to the questioned variable and read its corresponding value. For example, to perform the trials with question 'When O equals to 54, what is the value of S?', the participants may first locate the variable O. For all trials, each variable is displayed as a column in DT, while presented as an axis in SCP and PCP. After finding the column/axis of variable O, the participants may find the data point that has value 54 in that column/axis. The

participants may then locate variable S column/axis and read its corresponding value of the data point.

4.2.1.1 Searching Type and Searching Step

We categorise the searching method into two types: *forward searching* and *backward searching*. **Forward searching** occurs when the question asks for the value of the variable on the right hand side of the given variable. Therefore, participants need to search from left to right. **Backward searching**, on the other hand, happens when the question asks for the variable on the given variable's left hand side. Consequently, participants have to search from right to left. Considering that English is written from left to right, most people may find it easier to do forward searching after reading the English questions. We thus consider forward searching as the easier method.

The number of **searching steps** is determined by the distance from the given variable to the questioned variable. For instance, if the given variable is the first variable, the searching steps required to find the second variable, the third variable, and the fourth variable, are 1, 2, and 3 respectively. The searching step does not depend on the searching method. Hence, searching for the first or the third variable from the second variable require the same 1 searching step. The maximum searching step in the study is 3, as the data has 4 variables. Regarding minimum searching step, even though it is 1 in general, we set it to be 2 in order to prevent too easy tasks.

4.2.1.2 Level of Difficulty

The level of difficulty in value retrieval task is defined using searching method and the position of the involved variables.

Easy. For trials with easy level of difficulty, the given variable is the first variable, and thus the searching method is always forward searching. These trials are viewed as the easy tasks because the first variable is easy to locate, since most people are likely to search from left to right, as discussed above. There maybe 2 or 3 searching steps required for this task level.

Medium. Trials with medium level of difficulty require backward searching to the first variable. They are considered to be harder than the previous case because backward searching is thought to be more difficult, as explained above. Similar to the previous level, there maybe 2 or 3 searching steps required for these trials.

Hard. Hard-level trials require either forward searching or backward searching, but have no involvement with the first variable. We believe that they are hard as they require longer time to locate each variable. Especially in SCP technique, as the first variable is the

common vertical axis, searching from and to other variables add more steps to trace the data point. As an example, searching for the value of P when W equals 63 in Figure 4.1(b) is not straightforward. The participants may first locate the W axis and find the value 63, as usual. Subsequently, they may trace the data point to the vertical axis, which is variable L, then trace the data point across all three plots. In the last step, they may look at the leftmost plot and read the data point's corresponding value from the P axis, which is 64.

4.2.1.3 Distractor

In value retrieval task, the difficulty level of distractors depends on the value differences from the correct answer.

Easy. Distractors with easy level of difficulty differ from the correct answer from 40 to 80. If the correct answer is 60, for instance, distractors in this level must have the value in the range of -20 to 20 or 100 to 140. These distractors are easy to differentiate from the correct answer.

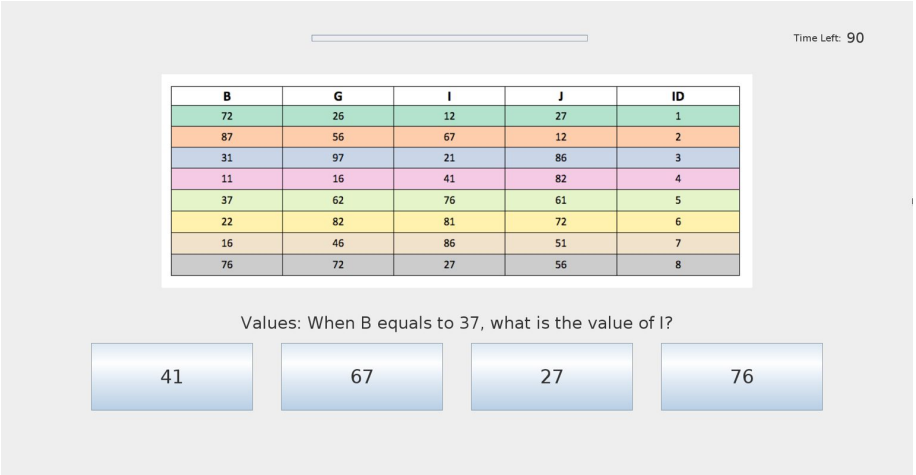
Medium. The value of medium-level distractors is different from the correct answer from 15 to 35. For the same correct answer as the previous example, 60, these distractors must be in the range of 25 to 45 or 75 to 95. As the distractors' values become closer to the correct answer, it creates more confusion, but still not that hard to distinguish between the two.

Hard. Hard-level distractors have 5 to 10 value differences from the correct answer. Using the same example, with 60 as the correct answer, the value of these distractors must be in the range of 50 to 55 or 65 to 70. We believe that these distractors have high potential to distract the participants, and consequently categorise them into the hard level.

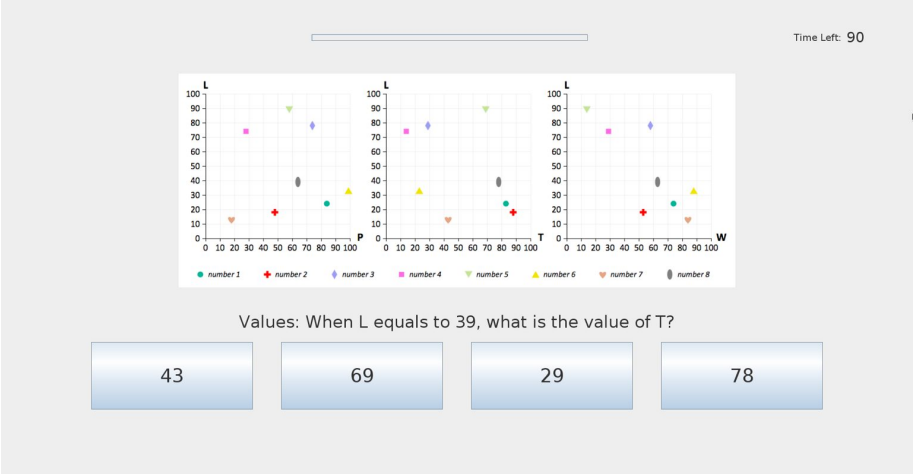
4.2.2 Stimulus Design

For the group of six stimuli having easy level of task difficulty, one triple requires 2 steps of forward searching from the first variable, while another triple requires 3 searching steps. Similar to the previous case, in the stimuli group with medium-level difficulty, one triple requires 2 steps of backward searching to the first variable, whereas another triple requires 3 searching steps. Regarding the hard-level group of stimuli, one triple requires 2 steps of forward searching while another triple requires 2 steps of backward searching; both have no involvement with the first variable.

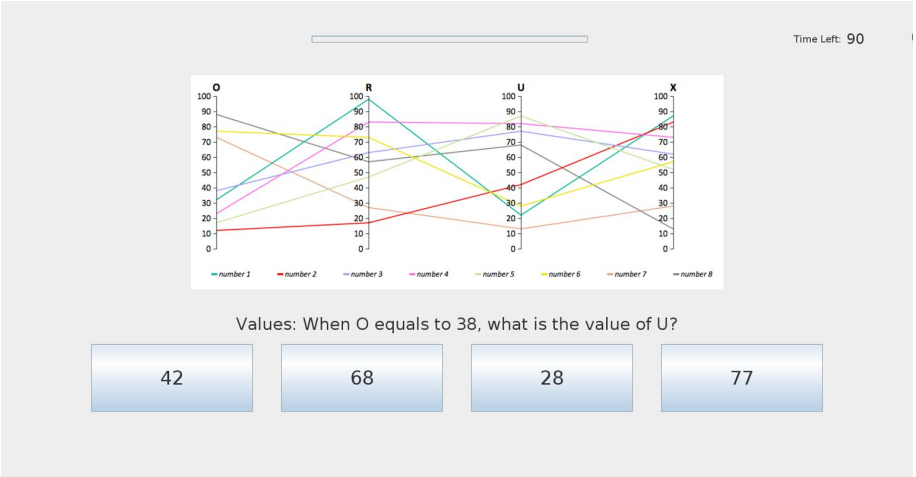
Figure 4.1 demonstrates the examples of the stimuli presented to the participants in this task. All other stimuli will be provided in Appendix C.



(a) Data Table.



(b) Scatter Plots.



(c) Parallel Coordinates Plots.

Figure 4.1: A triple of stimuli used in the experiment for value retrieval task.

4.3 Design for Clustering Task

For clustering task, participants are required *'to find the most appropriate cluster.'*

4.3.1 Task Design

The participants may perform this task by grouping data points according to their values in each variable; data points with close values should be grouped together. For instance, a data point with value 10, 60, 30, 80 in the four variables may be grouped with the data point with value 15, 20, 33, 75, whereas the data point with value 40, 55, 60, 20 may not belong to this cluster. To determine the closeness of the data points, each variable has equal weight. Therefore, having close value in many variables, the first two data points are grouped in the same cluster even though the values of second variable are quite different.

With SCP and PCP, clustering task can also be performed by observing data points' proximity and grouping closely located data points together. Again, to belong to the same cluster, data points in some variables can be located farther, as long as they have overall close proximity.

The correct answer is based on k-means clustering, which will be described below.

4.3.1.1 k-means clustering

k-means clustering, as discussed in Section 2.1, is used to partition observations into clusters.

In this study, we set K equal to 2. In addition, we control each cluster to have the same number of data points in order to reduce the task complexity, to ensure that the clustering tasks are not too hard. Hence, each trial has 2 clusters, each with 4 data points.

We use the Euclidean distance ($l^2 norm$), the most common metric, as the distance measure in our algorithm. The Euclidean distance is defined as

$$\|x_n - \mu_k\|_2 = \sqrt{\sum_{i=1}^{D}(x_{ni} - \mu_{ki})^2}$$

where x_n is the nth data point, μ_k is the centroid of cluster k, D is the number of variables, which is 4 in our study, x_{ni} is the value of variable i of the nth data point, and μ_{ki} is the value of variable i of the cluster k's centroid. Using this equation, we ensure that each variable has equal weight.

This algorithm, provided in detail in Section 5.3, is used to provide two clusters; one of which is selected as the correct answer in the optional choices.

4.3.1.2 Level of Difficulty

For clustering task, the level of difficulty depends on the number of variables that explicitly show the separation of the two clusters.

Easy. For trials with easy level of difficulty, values in 3 variables of data can be clearly grouped into two clusters. They are categorised as easy since the two clusters can be easily spotted and identified.

Medium. Medium-level trials have 2 variables that obviously display the two clusters. According to the decrease in number of these variables, it takes more time to investigate and separate the data points into two groups.

Hard. Trials with hard level of difficulty have only 1 variable that exhibits the two clusters distinctly. As a consequence, they require careful judgment in order to find the most appropriate cluster. We therefore classify them as hard.

4.3.1.3 Distractor

In clustering task, each optional answer consists of four data points. The correct answer is picked from one of the two most appropriate clusters calculated from k-means clustering technique, as introduced earlier. The distractors are obtained by swapping some data points in the two clusters, creating the incorrect answers. The difficulty level of distractors is based on the number of interchanged data points and their degree of closeness.

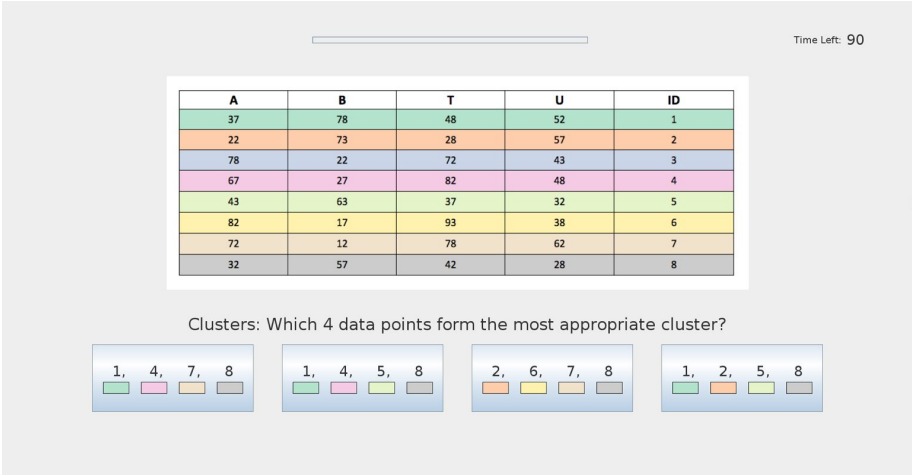
Easy. Easy-level distractors have 2 interchanged data points. In addition, the interchanged data points have noticeably different values from the other two. This is in order to ensure that these distractors will create not much confusion, and hence require just a few time for participants to realise that they are wrong.

Medium. Similar to easy level, distractors with medium level of difficulty also have 2 interchanged data points. Nevertheless, the data points selected have more proximity to the other two data points in the same choice. Thus, it requires more time and effort to make proper judgment.

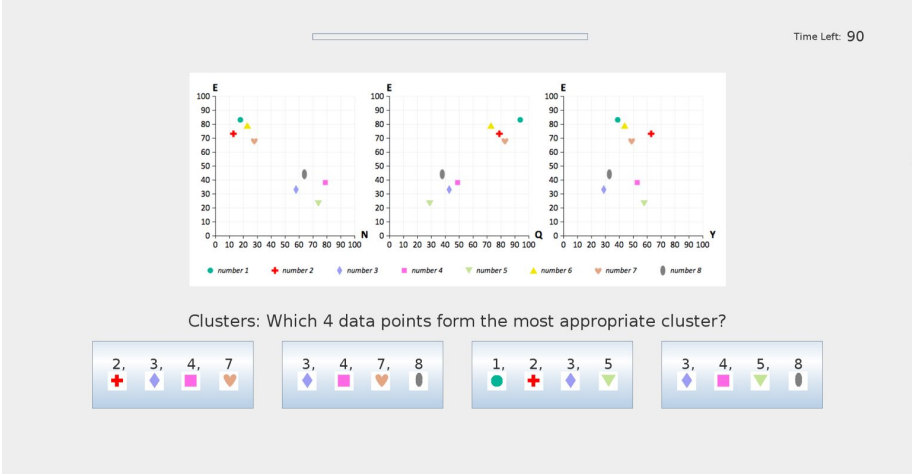
Hard. Hard-level distractors may have 1 or 2 interchanged data point(s). With 1 data point swapping, the distractors can largely distract the participants as they need to look at each data point in the choices more carefully. However, in some case, there may be 2 data points that have close proximity to another 2 data points in another cluster. In consequence, we may pick these four data points as the hard-level distractors.

4.3.2 Stimulus Design

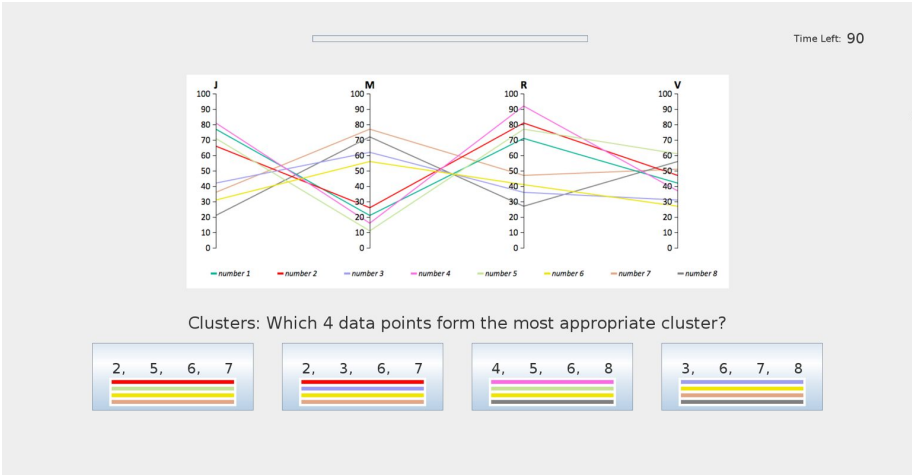
The stimuli group with easy level of difficulty has one triple displaying obvious two clusters in variable 1, 2, and 3, and another triple showing clear clusters in variable 2, 3 and 4. For



(a) Data Table.



(b) Scatter Plots.



(c) Parallel Coordinates Plots.

Figure 4.2: A triple of stimuli used in the experiment for clustering task.

the medium-level group of stimuli, the variables that distinctly exhibit the two clusters are 2 and 4 in one triple, and 3 and 4 in another triple. Regarding the stimuli group with hard level of difficulty, there is only one variable whose data can be clearly grouped into two clusters: variable 3 in one triple, and variable 4 in another triple.

The examples of the stimuli for clustering task are illustrated in Figure 4.2. All other stimuli will be provided in Appendix C.

4.4 Design for Outlier Detection Task

For outlier detection task, participants are required *'to find the data point(s) which is(are) outlier(s).'*

4.4.1 Task Design

In our design, outliers are based on two factors, which will be explained in the following subsection. For each trial, the outliers may exhibit only one factor or both factors. For this reason, this task can be performed by considering each factor separately.

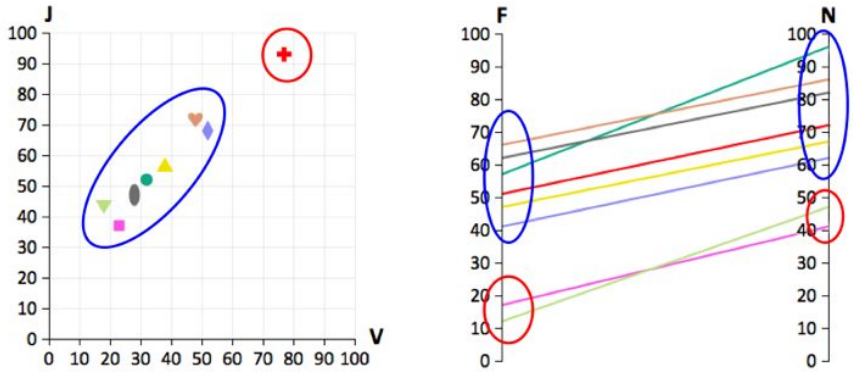
Besides, having only one outlier in each trial is too easy for the participants to detect. As a result, each trial may have one or two outlier(s). The participants are required to seek for all possible outliers.

4.4.1.1 Outlier Factors

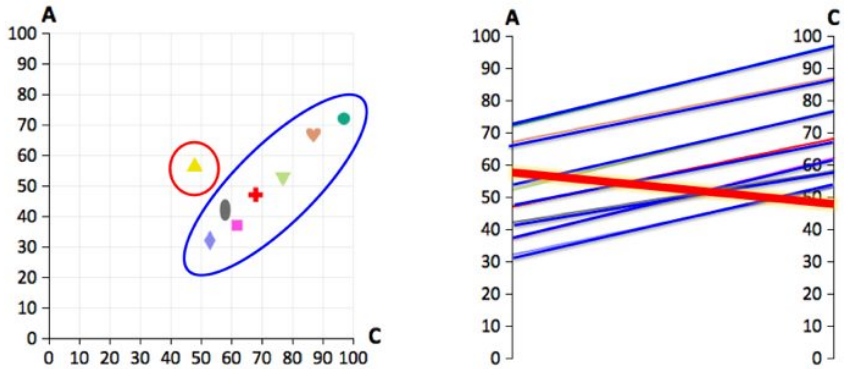
We define outliers by the two factors: different value range and different correlation.

Different Value Range. Outliers are the data points that are distant from other data. While most data points have close values and can form a cluster, the outliers lie outside of the cluster. As an example, suppose we take only one variable into account and the data set has data 80, 50, 70, 65, 20, 75 respectively. The fifth data point has different value range, and thus will be considered as the outlier. Figure 4.3(a) demonstrates outliers due to this factor.

Different Correlation. Correlation, as explained in Section 2.1, is the relationship between variables. Outliers are the data points that have different correlation from other data. If they are removed, the data in the data set will correlate better. For instance, considering only variable 1 and 2, and assume that each data point has value 30 and 50, 20 and 35, 40 and 10, and 50 and 75 for the two variables accordingly. The third data point, with value 40 and 10, has different correlation as the value in variable 2 is less than variable 1, which is different from other data points. Hence, this data point is identified as the outlier. Figure 4.3(b) illustrates outliers according to different correlation.



(a) Different Value Range.



(b) Different Correlation.

Figure 4.3: The two outlier factors.

4.4.1.2 Level of Difficulty

The level of difficulty in outlier detection task is defined by the type of outlier factors. In all cases, there may be 1 or 2 outlier(s) in each trial.

Easy. For trials with easy level of difficulty, the outliers are based on both factors. Therefore, they can be easily noticed and detected.

Medium. Outliers in medium-level trials are due to different value range factor only. As there is only one factor involved, they are harder to identify. Still, outliers with this factor are not much difficult to notice since the gap between outliers and other data can be easily seen.

Hard. Hard-level trials have outliers that are based on different correlation factor only. Outliers of this type are harder to detect as the participants have to observe the relationship between variables, not just looking at each variable separately. We therefore categorise them as in hard level.

4.4.1.3 Distractor

In outlier detection task, each optional answer may contain 1 or 2 data point(s). The distractors' difficulty level depends on the number of data points provided and their properties.

Easy. Easy-level distractors have the same number of data points as the correct answer, but none is the correct answer. For example, if the correct answer is the 4th data point, the distractors may consist of the 6th data point. In a similar manner, if the correct answer is the 4th and the 8th data points, the distractors may contain the 2nd and the 6th data points. Additionally, each data point is selected from the middle of the data set. As a consequence, it is easy to notice that the distractors are wrong.

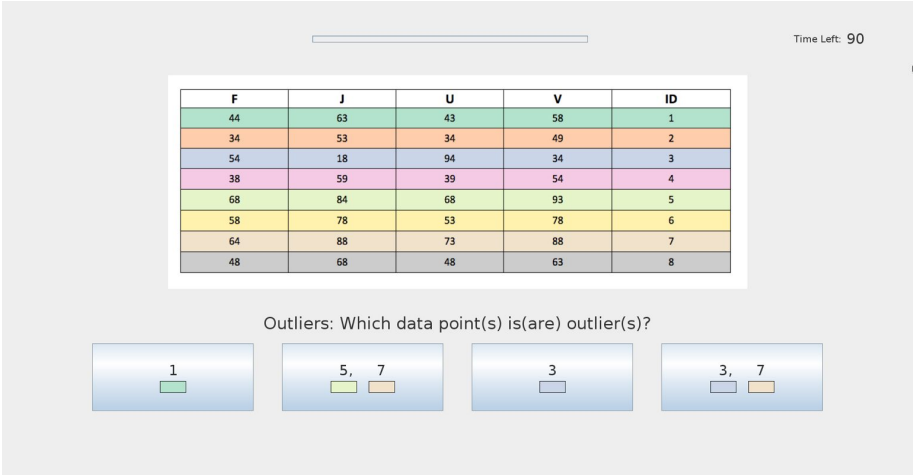
Medium. Distractors in medium level of difficulty have different number of data points from the correct answer, and none is right. Using the same example as the previous case, if the correct answer is the 4th data point, the distractors may include the 2nd and the 6th data points. Likewise, if the correct answer is the 4th and the 8th data points, the distractors may have the 5th data point. Moreover, in order to ensure that they have more potential to distract the participants then the easy-level distractors, each selected data point has some deviation from the overall data.

Hard. Similar to the previous case, hard-level distractors have different number of data points from the correct answer. Nevertheless, one data point is correct. Follow the same example, if the correct answer is the 4th data point, the distractors may contain the 4th and the 7th data points. In the same way, if the correct answer is the 4th and the 8th data points, the distractors may consist of the 4th data point. Even though the distractors in the latter case contain the correct answer, they are not considered as correct since they do not include all correct outliers. As this type of distractors has partial correction, it is defined as the hard distractor.

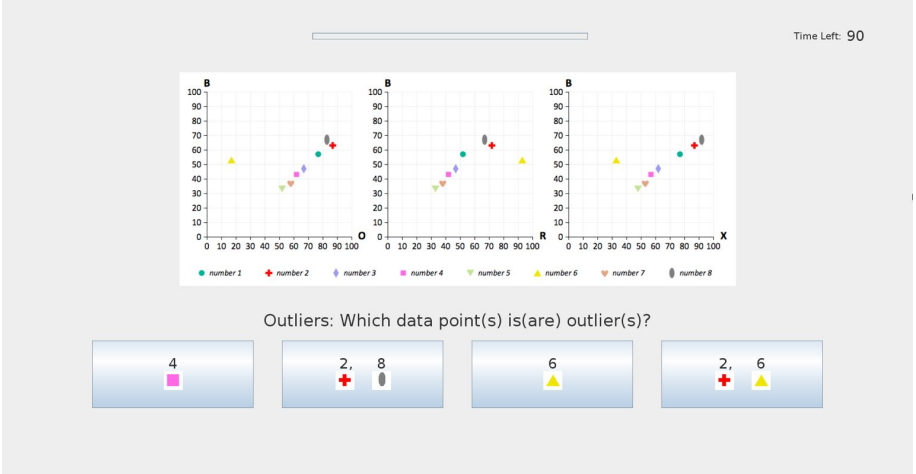
4.4.2 Stimulus Design

The two triples in easy-level group of stimuli have outliers due to both different value range and different correlation factors; one has 1 outlier while another has 2 outliers. Similar to the previous case, in the stimuli group with medium level of difficulty, one triple has 1 outlier whereas another triple has 2 outliers, but all outliers are based on different value range factor only. Following the same manner, one triple in hard-level group has 1 outlier and another has 2 outliers, but these outliers are due to different correlation factor only.

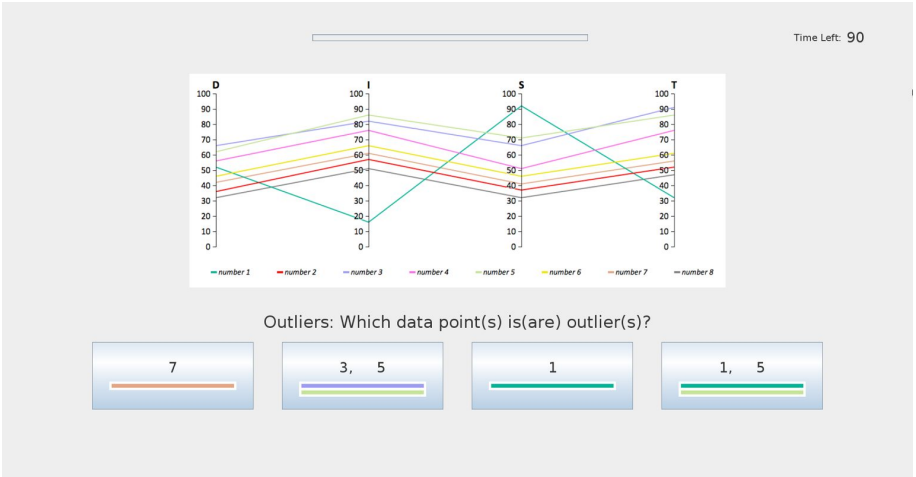
Figure 4.4 displays examples of the stimuli used for this task. All other stimuli will be provided in Appendix C.



(a) Data Table.



(b) Scatter Plots.



(c) Parallel Coordinates Plots.

Figure 4.4: A triple of stimuli used in the experiment for outlier detection task.

4.5 Design for Change Detection Task

For change detection task, participants are required *'to find the data point(s) with the most changes.'*

4.5.1 Task Design

Unlike data in other tasks, the provided data in change detection task is *temporal data*, whose variables represent states in time. Figure 4.5(a) shows an example of the temporal data, where the first variable, T1, represents the value at the first state, T2 represents the value at the second state, T3 represents the value at the third state, and T4 represents the value at the fourth state respectively. The participants may observe changes in data by looking at the value of each variable. The data point at the 5th row in Figure 4.5(a), for instance, changes from value 48 in state 1 to value 49 in state 2, then change back to 48 in state 3 and remain the same in state 4. Along all states of time, this example data point has changed by 2.

The following subsection explains how the participants may detect changes in each visualisation technique. Additionally, similar to outlier detection task, it is too easy to have only one data point with the most changes. Hence, each trial may have one or two answer(s) and participants are required to search for all possible answers.

4.5.1.1 Change Detection

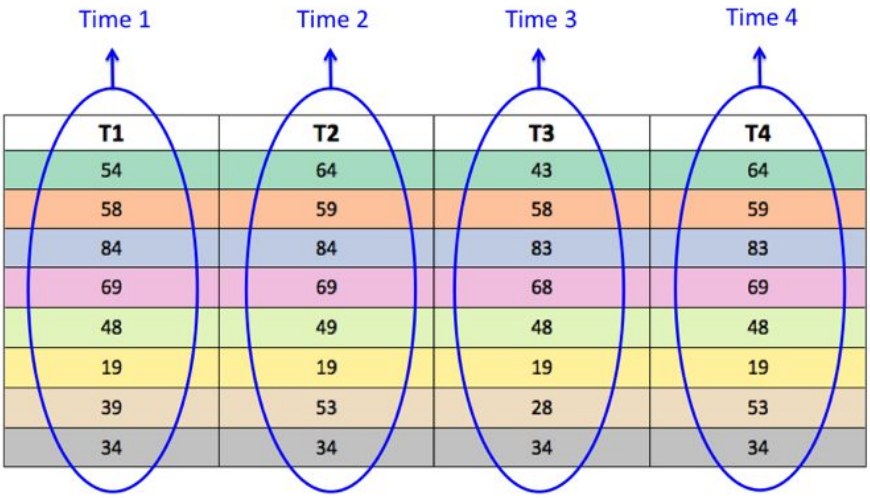
To detect changes in DT, as in Figure 4.5(a), the participants may observe the value of each data point and calculate, or preferably estimate, the changes across all states.

In SCP, if the value remains the same from one state to another, the data point will have equal values for horizontal and vertical axes of the two variables, and thus all data points with unchanged value will form a diagonal line with slope $m = 1$. Otherwise, if the value changes, the data point will divert from this diagonal line. Figure 4.5(b) illustrates a data set with two most changed data points.

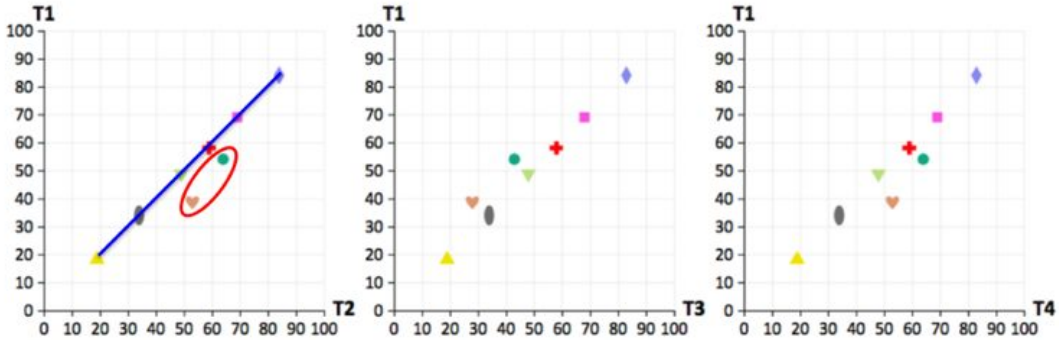
For PCP, if the value remains the same, the data point will be a constant line. Otherwise, if the value changes from one state to another, the data point's line will have some slope, either positive or negative. Figure 4.5(c) demonstrates a data set with two most changed data points, showing the green and the brown lines moving up and down across the axes.

4.5.1.2 Level of Difficulty

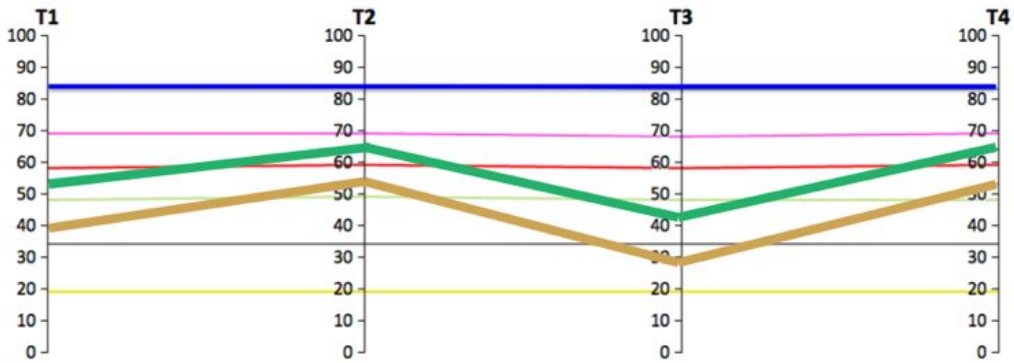
In change detection task, the level of difficulty stands on the value ranges of changes from one state to another. In all cases, there may be 1 or 2 answer(s) in each trial.



(a) Data Table.



(b) Scatter Plots.



(c) Parallel Coordinates Plot.

Figure 4.5: A data set for change detection task in the three visualisation techniques.

Easy. In trials with easy level of difficulty, the answer(s) change(s) in the range of 12, while other data points change in the range of 2. For example, the answer may have value

30, 40, 35, 30 in each state of time, whereas other data point may have value 35, 36, 36, 34 accordingly. Notice that the range of changes is defined for the changes from one state to another, not for the total changes across all states.

Medium. Medium-level trials have the range of changes at 16 for the answer(s) and 6 for other data points. For instance, the answer may have value 30, 43, 27, 35, while other data point may have value 35, 29, 32, 36. As the range of changes increases, it is more difficult to detect the data point(s) with the most changes.

Hard. The ranges of changes in hard-level trials are 20 for the answer(s) and 10 for other data points. As an example, the answer may have value 30, 48, 28, 42, whereas other data point may have value 35, 43, 36, 28. With the larger range of changes, the task becomes more complicated and thus it is classified as hard.

4.5.1.3 Distractor

In change detection task, each optional answer may contain 1 or 2 data point(s). Analogous to outlier detection task, the distractors' difficulty level is based on the number of data points provided and their properties.

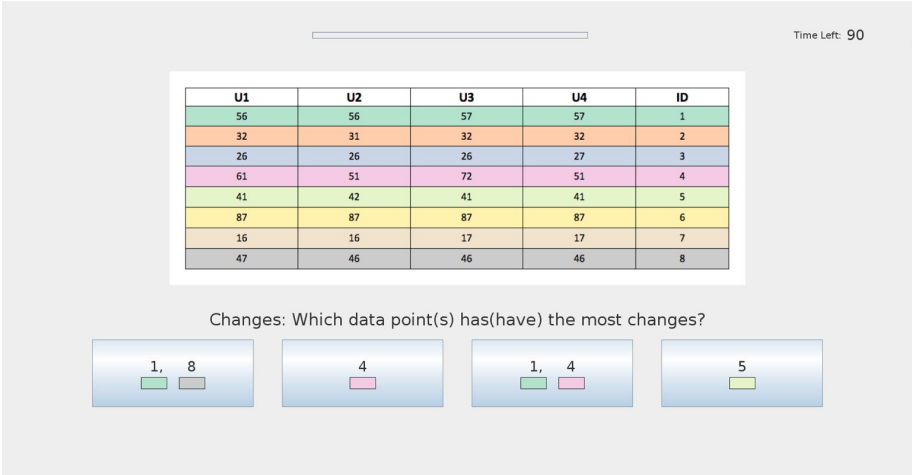
Easy. Distractors with easy level of difficulty have the same number of data points as the correct answer, but none is the correct answer. Furthermore, we select the data point that has only a few changes to ensure that it will not create much confusion.

Medium. Medium-level distractors have different number of data points from the correct answer, and none is right. Each data point in these distractors has more changes than other data, but still has less change than the answer(s). Since the data point(s) in this type of distractors contain(s) a lot of changes as well, the participants may believe that they are the correct answer. For this reason, these distractors are in medium-level difficulty.

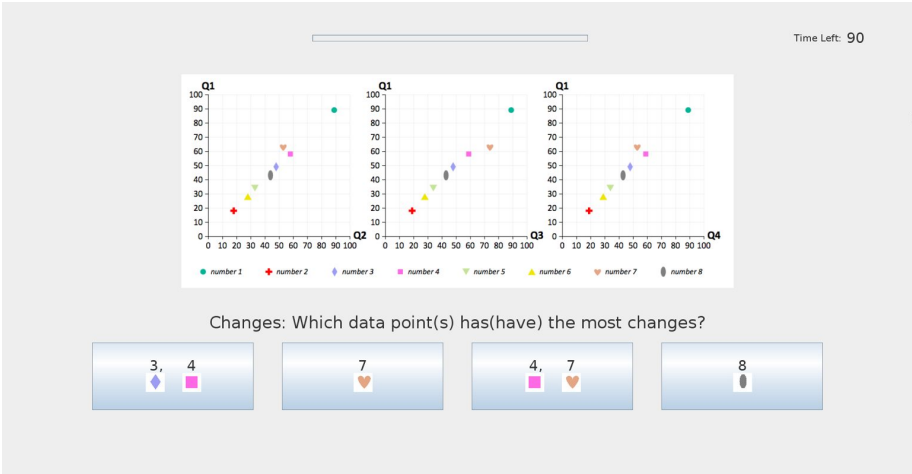
Hard. For hard-level distractors, they have different number of data points from the correct answer, but one data point is correct. Though these distractors contain partial correct answer, they are not defined as correct. The correct answer must include all correct data points. Yet, partial correction can create a huge distraction, and it is therefore categorised as hard.

4.5.2 Stimulus Design

For all three groups of stimuli in this task, one triple has 1 answer while another triple has 2 answers. Nevertheless, the stimuli in different groups have different properties. For the stimuli in the easy-level group, the answers change in the range of 12, while other data points change in the range of 2. In medium-level group of stimuli, the range of changes is



(a) Data Table.



(b) Scatter Plots.

(c) Parallel Coordinates Plots.

Figure 4.6: A triple of stimuli used in the experiment for change detection task.

16 for the answers and 6 for other data points. Regarding the hard-level group, the stimuli have the range of changes at 20 for the answers and 10 for other data points.

Figure 4.6 shows examples of the stimuli in change detection task. All other stimuli will be provided in Appendix C.

4.6 Stimulus Design

Stimulus design involves two important components, data design and visualisation image design, which are discussed in the following subsections.

4.6.1 Data Rules Design

In each stimulus, there are the total of **eight data points**. As discussed earlier, each data point in our study is a multivariate ratio data, consisting of **four variables**. In order to minimise the confounding effects, there are a number of rules that we apply to our data.

Firstly, we avoid negative values and floating-point numbers. We set the value range to 0 to 100 to ensure that it is large enough to have data variation, but not too large to display on the screen.

Secondly, some values should be avoided due to the confounding effect. These values include the values from 0 to 10, whose data points are displayed too close to the SCP's axes. Hence, it may be easier to read their corresponding value from the axes. Another value to avoid is 100 since it is the edge of the range. Besides, the values ending with 0 and 5 should be avoided as they are easier to notice.

Thirdly, since the participants may be able to memorise some data, we cannot use the same data for different stimuli. For this reason, we apply a minor variation with the data in the same triple to ensure that they still have similar data. Using one standard data set for each triple, the data set for each stimulus is varied by +0, +1, or -1. For example, if a data point in the standard data set has value 22, 57, 33, 78 for the four variables, the data set with +0 variation will have value 22, 57, 33, 78, whereas the data set with +1 variation will have value 23, 58, 34, 79, and another data set with -1 variation will have value 21, 56, 32, 77 accordingly. The three variations are assigned to each visualisation technique through randomisation process. Besides, in order avoid values ending with 0 and 5 in all variations, the standard data set must also avoid values ending with 1, 4, 6, and 9. As a consequence, the standard data will have values ending with 2, 3, 7, and 8 only.

Fourthly, to ensure that each data point is not located too close to each other in SCP and PCP, we define a minimum distance between each value in the same variables to be 4. For instance, the standard data set should not have values 22 and 23 in the same variable,

as same as not having both values 57 and 58 together. This is in order to ensure that each data point can be clearly distinguished from one another.

Lastly, in order to further reduce the ability to memorise data, we vary the variable names of each stimulus. Within each triple, the variable names are pseudo-randomised to ensure that there are no duplicate values. Nevertheless, the variable names in each stimulus are in alphabetical order. Apart from variable randomisation, we sort the data points in each stimulus differently to avoid the ability to remember the correct answer's data *identification number (ID)*. As an example, the correct data point in DT technique may have ID 3, whereas it may have ID 6 and 1 in SCP and PCP respectively.

The data rules are summarised below.

Rule 1: The values are integers between 0 and 100.

Rule 2: Standard data does not include the values between 0 and 10, the value 100, and the values ending with 0, 1, 4, 5, 6, and 9.

Rule 3: For each triple of stimuli, the data sets v are applied with a minor variation from the standard data s, where:

- $v_1 = s + 0$
- $v_2 = s + 1$
- $v_3 = s - 1$

Rule 4: The minimum distance between each value in the same variables is 4.

Rule 5: Within each triple of stimuli, the variable names are unique and the orders of data points are different.

4.6.2 Visualisation Image Design

The stimuli in the same triple are designed to have similar information, format, and complexity. Each stimulus is a visualisation image corresponding to a unique data set. Nevertheless, the data rules, as explained in the previous subsection, ensure that the stimuli in each triple have similar data. Other properties of each visualisation image are described below.

Labelling. To perform the visualisation tasks in our study, except for value retrieval task, the participants are required to identify each data point and select the point with the correct property: forms a cluster, be an outlier, or has the most changes. Hence, we need to

label each data point with an identification number (ID). For DT technique, we add another column to display each data point's ID. This ID column is added as the rightmost column of the table, in order to create less interference to the real data value. The participants may first observe the data points' values, then read their corresponding ID just before selecting the answer. In SCP and PCP, we add a legend to display the ID. These legends are placed below each plot to generate less distraction to the task performance. To differentiate each data point, colour is used as a visual channel for all three visualisation techniques, while SCP has shape as an additional channel for the encoding. The detail of colour and shape are explained in the following paragraphs.

(a) Colour scheme from ColorBrewer.

(b) Modified colour scheme for scatter plots and parallel coordinates plots.

(c) Lighter version of colour scheme for data table.

Figure 4.7: Different versions of the colour schemes.

Colour. The colours in our study are carefully selected to ensure that each colour has enough distance difference. We first pick '8-class Set2' colour scheme from *ColorBrewer* [Col13], then modify it to further maximise the differences. The lighter scheme, '8-class Pastel2' from ColorBrewer [Col13], is used in DT technique as the colours are used for background colour. For SCP and PCP, the colours are used for point colour and line colour respectively. Figure 4.7 presents different versions of these colour schemes.

Data Point Format. A data point is presented as a row in DT, whereas displayed as a point in SCP, and shown as a line in PCP. In the latter case, each line is designed to have 2-pixel width in order to allow the participants to comfortably recognise the lines. For SCP, the eight data points have eight different shapes: circle, cross, diamond, square, triangle-down, triangle-up, heart, and oval. Each shape has the same area of 100 square

pixels, except the oval shape which requires some more pixels. Since SCP has two visual channels, colour and shape, they are paired to maximise the differences between each data point; similar shapes are selected to have totally different colours. For instance, circle has green colour while oval is grey, and triangle-down has light green colour whereas triangle-up is in yellow colour.

Axis Scales and Tick Marks. All axes in SCP and PCP have the fixed scale from 0 to 100, with the tick marks at every 10 values. They are both in black colour. Besides, SCP has grid lines for both horizontal and vertical axes. The grid lines are in light grey colour with 0.7 opacity, to ensure that they will help in reading value, but not reducing the ability to view data points.

Font. All text in every stimuli use the same *Calibri* font to ensure consistency. The variable names are displayed in bold, with 22-pixel font size. The data values in DT, together with the values on the SCP and PCP's axes, all have the same font size of 18 pixels. The legends in the latter two techniques are presented in 15-pixel font size. Furthermore, all data values in DT, including the variable names, are aligned in the centre to enhance readability.

Size. For SCP and PCP, each plot cell has a fixed size of 300 pixels. As a result, with 4 axes, each PCP has the size of 900 pixels x 300 pixels, excluding the legend. SCP, on the other hand, has 50 pixels of gaps between adjacent plot cells, and therefore has the size of 1000 pixels x 300 pixels, excluding the legend. To provide analogous similarity, each DT has the total size of 900 pixels x 300 pixels.

4.7 Software Design

The software is used in the experiment to collect participants' performance on different visualisation tasks, together with their demographics information and feedbacks. The workflow of our software is illustrated in Figure 4.8. There are four main parts in the software's workflow, including demographics information and familiarity rating, training session, trial session, and effectiveness rating.

The following subsections first describe the workflow of the software, followed by the design of sequence and time scheme.

4.7.1 Software Workflow

In the first part of the software, the participants are required to give some demographics information and familiarity rating regarding of each visualisation technique: DT, SCP, and PCP.

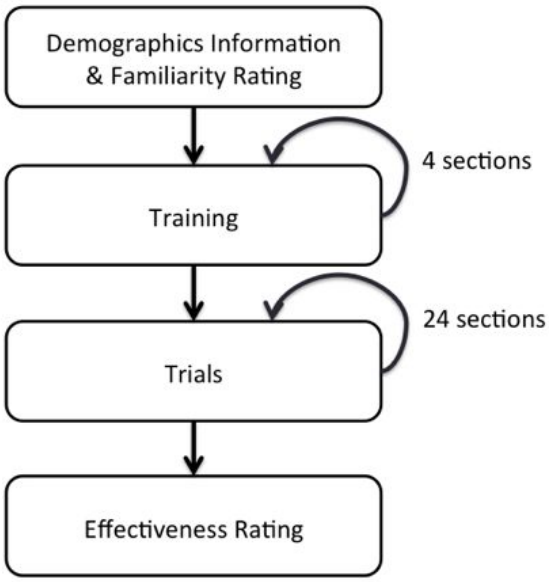


Figure 4.8: Workflow of the software in the experiment.

The second process is the training session, consisting of 4 sections. Each section is designed to familiarise each participant with instructions and how to perform each visualisation task. There are 3 trials in each section, providing one stimulus for each visualisation technique. To facilitate learning in each trial, the participants are given feedback whether the selected answer is correct or wrong, and the correct answer is also shown on the screen. The 4 training sections provide the training on value retrieval task, clustering task, outlier detection task, and change detection task respectively. Besides, within each section, the 3 training trials are ordered to display DT, SCP, and PCP respectively. In this process, the participants are required to perform 12 training trials in total.

The third step of the workflow is the main trial session. In this session, the total of 72 stimuli are presented to each participant in 24 sections of 3 stimuli each. Each section is designed to include 3 trials in the same task, one for each visualisation technique. However, unlike the training session, the visualisation techniques in different sections are not appeared in the same order. Furthermore, the 24 sections are not ordered from value retrieval task to change detection task as in the training session. All trials are pseudo-randomised, as explained in the following section.

In the last part of the software, the participants are required to provide effectiveness rating of each visualisation technique, as according to each visualisation task. The effectiveness ratings, including the familiarity rating in the first process, use five-level Likert

scale which consists of *not at all effective/familiar*, *slightly effective/familiar*, *moderately effective/familiar*, *very effective/familiar*, and *extremely effective/familiar* respectively.

4.7.2 Sequence Design

As explained in Section 2.5, stimuli provided in the earlier sections of the software may create positive and negative effects on the stimuli displayed later. In addition, the order of the optional choices may affect the time required to select the correct answer, since the participants need to move the mouse in different amount to different position.

Therefore, we reduce this order effect by using **pseudo-randomisation** so that participants would not be able to guess the order of the stimuli, or to reason the position of the correct answer. The pseudo-randomisation scheme uses complete counterbalancing technique to ensure that for each visualisation task, all three visualisation techniques have equal chance to be shown before one another, in order to cope with the learning effect. Furthermore, the scheme also ensures that the three stimuli in the same triple are separated from one another by at least 5 other stimuli. This is in order to reduce the participants' ability to recognise similar data in the same triple. Additionally, the scheme also ensures that the three stimuli in the same section are displayed from easy to hard level of difficulty. We arrange the hard-level stimuli as the last trial in each section to minimise the effect of participants' gradual fatigue, which may reduce their performance on the consecutive trials. Lastly, we use pseudo-randomisation to ensure that the correct answers have equal chance to appear in each of the four optional choices' position.

4.7.3 Time Design

Some participants may experience fatigue or boredom after repeated trials, which may affect the participants' performance in different sections of the study, as discussed in Section 2.5.

Therefore, we minimise this effect by introducing **a short break between each section**. Participants are allowed to take as much time as they want to ease the tiredness before continuing to the next sections. Our study does not limit the time for the break as different participants may need different time to recover from fatigue. Yet, we **limit the time for each trial**, according to two reasons. Firstly, without time limitation, if the participants cannot find the correct answer in some trials, they may choose to guess and select one of the optional choices, and thus the results will not be effective. Secondly, some participants may use a long time to do some difficult trials and become exhausted during the later part of the study. Hence, their performance may be reduced towards the end of the experiment. For these reasons, we set a time limitation for each trial.

To determine the amount of the suitable limited time, we conducted a pilot study with 2 participants, 1 female and 1 male. The average response time for the two participants is 16.63 seconds, while the maximum time used in a trial is 86.70 seconds. Besides, with the initial time limitation of 90 seconds, there is only one trial that a participant cannot answer in time. Consequently, we set the limited time to be 90 seconds as it is not too long but allow sufficient time to perform the task, on average.

Chapter 5

Implementation

The implementation process consists of three main components, stimulus generation, software implementation, and the experiment itself.

We iteratively perform stimulus generation and software implementation, together with the design process explained in Chapter 4, to assure the software quality. After which, we conduct a controlled experiment using the developed software to collect user performance on the four different visualisation tasks with the three different visualisation techniques.

In summary, there are the total of 3,024 trials included in our study (excluding training trials): 42 participants x 4 visualisation tasks (value retrieval, clustering, outlier detection, change detection) x 3 visualisation techniques (DT, SCP, PCP) x 6 stimuli (2 with easy level, 2 with medium level, 2 with hard level).

In this chapter, we first describe the development process used to develop the software for the experiment. The discussion then moves to planning and requirement analysis, followed by stimulus generation and software implementation respectively. In the last section, we provide the detail of the experiment.

5.1 Software Development Process

Our project use *Agile software development*, which is based on iterative, incremental, and evolutionary process.

Agile methods break the project down into smaller software increments; each increment follows a full software development life cycle, including *planning*, *requirement analysis*, *design*, *coding*, and *testing*. At the end of each cycle, an executable product is demonstrated to stakeholders, which are our research team members, to evaluate the software and get feedback for the next increments. These software increments are developed in iterations over short time frames. In our project, each iteration is a one-week period.

For software quality assurance, the testing in each iteration includes *unit testing*, *integration testing*, and *system testing*. Unit testing tests each component, integration testing tests functionally grouped components, while system testing tests the entire system. Software quality assurance is very important in our project since the software will be used in an experiment. If any software's mistake occur during the experiment, we cannot ask the same participants to do the experiment again as they have already seen the stimuli, and thus it will create some biases. As a consequence, we iterate the development process until there is no defect in the software and it can work perfectly as expected.

In total, we perform eight iterations, as follows:

Iteration 1:

- Design and generate data sets for value retrieval task.

Iteration 2:

- Define data rules and generate new data sets for value retrieval task.
- Design and generate visualisation images.

Iteration 3:

- Redesign and generate second version of visualisation images to ensure that all three visualisation techniques have similar formats.
- Design software workflow and develop software for pre-study and post-study questionnaires.

Iteration 4:

- Design and generate data sets for clustering task.
- Add gridlines to SCP.
- Design timing procedure and implement software for main trial session.

Iteration 5:

- Implement k-means clustering algorithm and regenerate data sets for clustering tasks.
- Label data points in DT, SCP, and PCP by using colours and shapes.
- Change interface for familiarity rating and effectiveness rating to use Likert scale.

Iteration 6:

- Redesign and generate new data sets for clustering task.
- Provide legends for SCP and PCP.
- Change data points' formats in the optional answers for each trial to match with the formats of its visualisation technique.

Iteration 7:

- Design and generate data sets for outlier detection and change detection tasks.
- Modify colours for SCP, and PCP to maximise their distance differences.
- Improve software to enhance usability and ensure that all needed results are recorded.

Iteration 8:

- Add progress bar and countdown timer.
- Implement software for training session.
- Prepare materials for introduction session.

The designs for tasks, stimuli, and software are explained in Chapter 4, whereas planning and requirement analysis, together with stimuli generation and software implementation, will be discussed in the following sections. Example stimuli for each iteration will be shown in Appendix B.

5.2 Planning and Requirement Analysis

There are eight iterations in our development process, and thus there are eight planning and requirement analyses involving in our project. Nevertheless, we summarise all eight components into two class diagrams, one for data generation (Figure 5.1) and one for software program (Figure 5.2).

For data generation, there are five main classes, one for each visualisation task, and another class for common methods. The *CommonMethods* class provides methods that will be used in all data generation, for instance the method to verify data rules and the method to write data sets into files. On the other hand, the four classes for the four tasks contain methods that will be specifically used for each task. For example, *Task1_GenerateData* class will involve a method to check for distractors' value differences, whereas *Task2_GenerateData* class will consist of a method to get the two most appropriate clusters. Section 5.3 will go into detail of these classes.

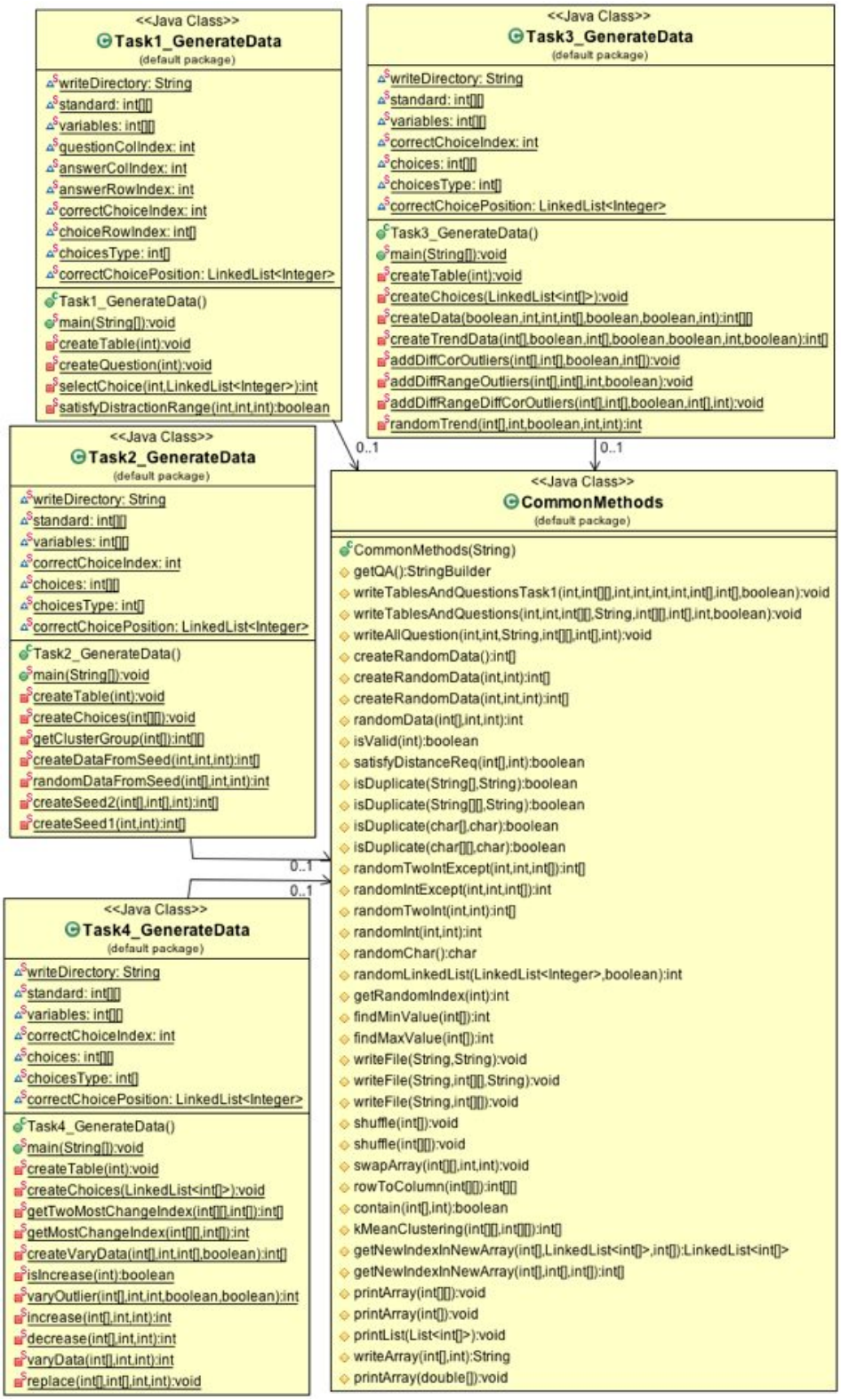


Figure 5.1: Class diagram for data generation.

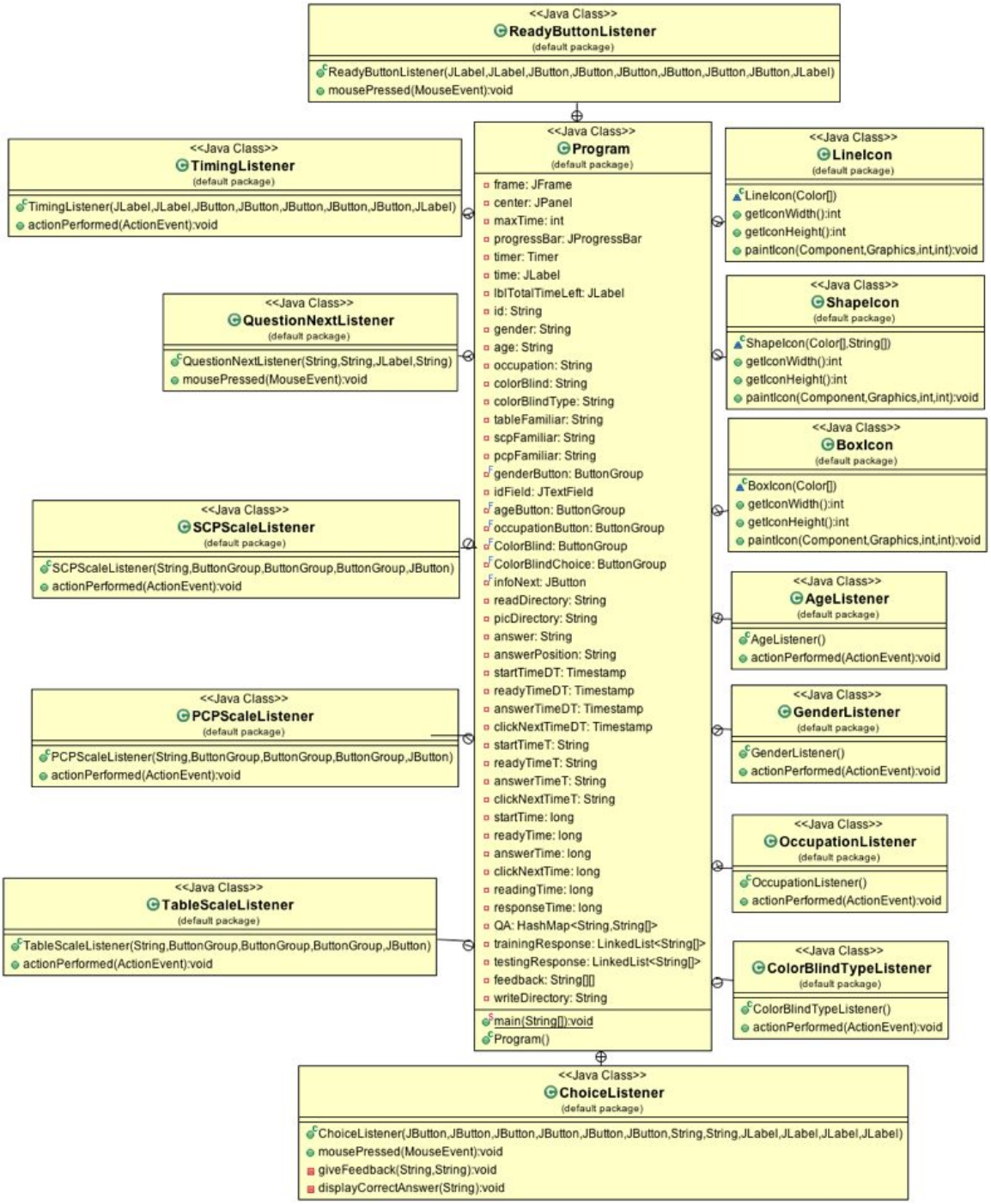


Figure 5.2: Class diagram for software program.

For the software program, there is one main class, with several *Action Listener* classes for handling button clicking events. There are also three classes for the three data points' formats, *BoxIcon*, *ShapeIcon*, and *LineIcon*, to be displayed in the optional choices for DT, SCP, and PCP respectively. Section 5.4 will explain these classes in detail.

5.3 Stimulus Generation

Stimulus generation consists of two main processes, data generation and visualisation image generation.

Firstly, the data sets are generated using custom software written in *Java*. Then, the generated data sets are used to create stimuli through custom software written in *JavaScript*. The stimuli are then saved as static visualisation images to be displayed in the software. The following subsections discuss these steps in detail.

5.3.1 Data Generation

There are five main classes for this process, *Task1_GenerateData*, *Task2_GenerateData*, *Task3_GenerateData*, *Task4_GenerateData*, and *CommonMethods* (Figure 5.1). For each class, we will explain the functions of the class, and will provide some essential methods.

Task1_GenerateData.java

This class is used to generate data sets for value retrieval task.

The method *createQuestion(stimuli)* generates given variable and questioned variable for each stimulus, according to its level of difficulty. For instance, stimuli in easy-level task will have the first variable as the given variable.

satisfyDistractionRange(range, answer, value) verifies whether the distractor's *value* differs from the *answer* in the appropriate *range* or not. The *range* variable in this method uses 1, 2, and 3, for easy, medium, and hard level of difficulty respectively. For each *range*'s value, the method will define minimum range and maximum range accordingly.

Task2_GenerateData.java

This class is used to generate data sets for clustering task.

For each data set, we first generate two *seeds*, which have the same value for some variables, and have significantly different value in other variables to display obvious clusters, as according to the stimulus design. We then generate the data set from these two seeds.

createDataFromSeed(seed1, seed2, range) randomly generates four data points around *seed*1 and another four data points around *seed*2 with the distance differences specified by *range* variable.

After all eight data points are generated, we use k-means clustering algorithm, provided in *CommonMethods* class, to assign data points to each cluster.

getClusterGroup(cluster) takes an array of *cluster* assignment obtained from k-means clustering algorithm, then outputs an array of the two clusters; one will be selected as the correct answer.

Task3_GenerateData.java

This class is used to generate data sets for outlier detection task.

addDiffRangeDiffCorOutliers generates outliers for easy-level trials, based on both different range and different correlation factors.

addDiffRangeOutliers generates outliers for medium-level trials, due to different range factor only.

addDiffCorOutliers generates outliers for hard-level trials, according to different correlation factor only.

Task4_GenerateData.java

This class is used to generate data sets for change detection task.

createVaryData(original, level, outlierIndex, change) generates values for each state of time. *original* represents data points' values in the first state, *level* identifies level of task difficulty. *outlierIndex* indicates the index of the outliers. *change* specifies that the values in this state change from the original state or not.

getMostChangeIndex and *getTwoMostChangeIndex* obtains the most, and the two most, changes data points from the data set, after excluding the answers of the data set. These data points will be used for hard-level distractors.

CommonMethods.java

This class is used to provide common methods for all data generation.

As defined in Section 4.6, there are five **data rules** for our data sets. The first rule is to have values between 0 and 100, which can be constrained through method *randomInt(min, max)* provided in this class. The method will random number from the specified *min* and *max* values, using method *nextInt* from *Random* class. Other methods for the other four rules are provided below.

isValid(number) verifies whether the *number* follows Rule 2 or not, which is to avoid values between 0 and 10, value 100, and values ending with 0, 1, 4, 5, 6, and 9.

getVariation(standard, variation) generates a data set with some *variation* from the *standard* data set, including 0, +1, and -1 variation, as according to Rule 3.

satisfyDistanceReq(array, value) identifies whether the *value* differs from other values in the same variable with at least 4 or not. *array* is the array of all values in the variable. This is in order to ensure Rule 4.

isDuplicate(array, value) checks whether the *value* is duplicate with the header variables in the *array* or not, to ensure Rule 5.

For clustering task, it involves ***k*-means clustering** algorithm method, which is provided in this class.

kMeanClustering(data, initialCentroids) takes *data* set and *initialCentroids* as input, and outputs the array of data points' cluster assignment.

The data set is defined as $\mathbf{x} = \{\mathbf{x_1}, ..., \mathbf{x_N}\}$ where $\mathbf{x}$ is a vector of data points, and $\mathbf{x_n} = \{x_{n1}, ..., x_{nD}\}$ where x_{nd} is the value of nth data point in d^{th} variable, N is the number of data points, and D is the number of variables.

The set of clusters' centroids is defined as $\boldsymbol{\mu} = \{\boldsymbol{\mu_1}, ..., \boldsymbol{\mu_K}\}$ where $\boldsymbol{\mu}$ is a vector of clusters' centroids, and $\boldsymbol{\mu_k} = \{\mu_{k1}, ..., \mu_{kD}\}$ where μ_{kd} is the value of centroid of the kth cluster in d^{th} variable, K is the number of clusters, and D is the number of variables.

The set of cluster assignments is defined as $\mathbf{r} = \{\mathbf{r_1}, ..., \mathbf{r_N}\}$ where $\mathbf{r}$ is a vector of data points' assignments to clusters, and $\mathbf{r_n} = \{r_{n1}, ..., r_{nK}\}$ where r_{nk} is a binary variable denoting the assignment of nth data point to k^{th} cluster, N is the number of data points, and K is the number of clusters.

The k-means clustering algorithm is provided in Algorithm 5.1.

After each data set is generated, it will be saved as a ***comma-separated values (CSV)* file**. Additionally, the questions and four optional choices for all data sets in a task are saved as another CSV file. The former files will be used to generate visualisation images, whereas the latter files will be used to display questions and answers in the software program.

writeFile(filename, array, columnName) creates a CSV file with specified $filename$, by writing $columnName$ as the header row of the file and adding data from the $array$.

5.3.2 Visualisation Image Generation

We use $HyperTextMarkupLanguage(HTML)$ and $JavaScript(JS)$ to generate the visualisation stimuli.

For each stimulus, a data set is loaded from the generated CSV file, explained in the previous section. *D3 for Data-Driven Documents (D3.js)* [Bos] is used as the JS library to draw DT and *Scalable Vector Graphics (SVG)* of SCP and PCP. A *Cascading Style Sheets (CSS)* file is used to define the standard format for size and colour of text fonts and data points for all visualisation techniques

Considering the eight different shapes in SCP, we use six standard symbol types (circle, cross, diamond, square, triangle-down, and triangle-up) from D3.js [Bos], and manually draw heart and oval shapes by specifying their paths.

The visualisation stimuli are displayed in web browsers and captured as static visualisation images in *Portable Network Graphics (PNG)* files. The PNG files are then used to present the visualisation images in the software program.

Algorithm 5.1 k-means clustering

Input: $\mathbf{x} = \{\mathbf{x_1}, ..., \mathbf{x_N}\}, \boldsymbol{\mu} = \{\boldsymbol{\mu_1}, ..., \boldsymbol{\mu_K}\}$
Output: $\mathbf{r} = \{\mathbf{r_1}, ..., \mathbf{r_N}\}$

```
function kMeanClustering(x, μ)
    initialise oldCentroids
    newCentroids ← μ
    while newCentroids has changed from oldCentroids do
        oldCentroids ← newCentroids
        clusters ← assignClusters(x, oldCentroids)
        newCentroids ← computeCentroids(x, clusters)
    end while
    return clusters
end function

function assignClusters(x, μ)
    initialise r                                          ▷ clusters assignment
    for all data points n do
        initialise distance[K]                ▷ Euclidean distance for all K clusters
        for all clusters k do
            initialise sumOfSquareDifference
            for all variables d do
                add sumOfSquareDifference with (x_nd − μ_kd)^2
            end for
            distance[k] = √(sumOfSquareDifference)
        end for
        assign r_n to cluster k with minimum distance
    end for
    return r
end function

function computeCentroids(x, vecr)
    initialise μ                                          ▷ clusters' centroids
    for all clusters k do
        for all variables d do
            initialise clusterPoints              ▷ number of data points in the cluster
            initialise sum                      ▷ sum of values in variable d of cluster k
            for all data points n do
                if r_nk = 1 then                ▷ the data point is assigned to this cluster
                    clusterPoints++
                    add sum with x_nd
                end if
            end for
            μ_kd ← sum/clusterPoints            ▷ mean of all data points in the cluster
        end for
    end for
    return μ
end function
```

5.4 Software Implementation

We implement the software program in *Java*, using *Swing* components to provide the *graphical user interface (GUI)* for the Java programs.

All classes for software implementation are displayed in Figure 5.2. The main class of the software is *Program* class, which is executable. Other classes include several Action Listener classes for button handling, and three classes for data points' formats in optional choices. The *Program* class contains critical functions, which will be described below.

Program.java

This class is the main class to run the software program.

It creates a *JFrame* with *BorderLayout* content panel, which divides the panel into five regions, *north*, *south*, *east*, *west*, and *centre*. The north region is used to display progress bar and countdown timer, whereas all other components are displayed in the centre region.

For centre region, we use *CardLayout* to store all panels of the software in a stack, and allow the participants to navigate from one panel to another by clicking on a button. All panels include a demographics information panel, a familiarity rating panel, 12 training trial panels, 72 main trial panels, and 4 panels for effectiveness rating.

Demographics information and familiarity rating panels are used to record the answers for pre-study questionnaires, including *id*, *gender*, *age*, *occupation*, *colorBlind*, *colourBlindType*, *tableFamiliar*, *scpFamiliar*, and *pcpFamiliar*.

Before creating **training trial and main trial** panels, we implement the software to first read the generated CSV files for questions and optional choices in each visualisation task, described in Section 5.3. All values are then stored in a *HashMap* variable, *QA*, where the HashMap's key is the stimuli name, and the HashMap's value is the array of seven information: question, first choice's values and type, second choice's values and type, third choice's values and type, fourth choice's values and type, correct answer's values, and correct choice position. The types of each choice include correct answer, easy-level distractor, medium-level distractor, and hard-level distractor. After all values are stored in the HashMap, the panel for each trial is created.

addQuestionSection(panelName, nextPanel, sectionHeader, task, pic1, pic2, pic3, trialType) creates panels for a section, including of one section title panel and three trial panels for DT, SCP, and PCP. *panelName* defines the panel name for this section, whereas *nextPanel* specifies the next panel section, which will be displayed after the participants click on '*next*' button at the end of this section. *task* identifies the type of visualisation task in this section, which can be 1 for value retrieval, 2 for clustering, 3 for outlier detection, and 4 for change detection. *pic*1, *pic*2, and *pic*3 give the stimuli name for the first, the

second, and the third trial panels respectively. $trialType$, either training or testing, tells whether the trials are for training or for main trials.

Using the same variables $pic1$, $pic2$, and $pic3$ to both load the visualisation images from the PNG files and retrieve the mapped values from the HashMap will ensure that we will get the stimuli images with their correct corresponding information. We also use these variables to specify the trial sequence as according to the pseudo-randomisation scheme, as explained in Section 4.7.

For **timing procedure**, we record the $startTime$ when each trial is appeared on the screen, then record the $readyTime$ when the participants click on '$showpicture'$ button. The time difference between these two recorded values is the time used to read question and choices; it is recorded as $readingTime$. Later on, after the participants click on an optional answer, the $answerTime$ is recorded. We then record $responseTime$ as the time difference between $readyTime$ and $answerTime$. Both $readingTime$ and $responseTime$ are recorded in milliseconds, while $startTime$, $readyTime$, and $answerTime$ are recorded in three formats: date and time in $Timestamp$, time in $Timestamp$, and time in milliseconds. This is in order to ensure that the recorded values are correct. We also record both selected choice's values and position.

After creating all trial panels, we create the panels for **effectiveness rating**. These panels are used to record the answers for post-study questionnaires, which consist of the effectiveness rating for DT, SCP, and PCP in value retrieval, clustering, outlier detection, and change detection tasks.

In all panels, the '$next'$ buttons will only appear on the screen after all forms or all questions in the current panel have been answered. This is in order to ensure that the participants will not be able to go to the next section if they do not give all required information. The values of these answers will be recorded after the participants click on these '$next'$ buttons. When the last panel is reached, all recorded values will be exported to a CSV file. These values include the given ID of the participants and their demographics information, familiarity rating, stimuli name and the four optional answers, selected answer, time for each step, reading time and response time, and effectiveness rating of each visualisation technique, as explained earlier.

The software program is exported to an executable *Java Archive (JAR)* file, which is runnable on any operating system. Besides, we implement the software to be run in full screen mode, in order to avoid the participants to quit the software during the experiment and to prevent any distractions from outside the software environment.

5.5 Experiment

The experiments were conducted in 5 sessions, 1 for the pilot study, and another 4 for the real experiments. All 5 sessions of experiments took place at the *Department of Computer Science, University of Oxford*.

For the pilot study, there were 2 participants, 1 female and 1 male, performing the experiment to evaluate our study design and its implementation. Our pilot study went well with no mistake occurred.

In the real experiments, each session consists of 8 to 15 participants. We set the maximum number of participants in each session to be 15 in order to provide sufficient support for each participant when needed. The following subsections describe the participants, apparatus, and procedure involved in the real experiments.

5.5.1 Participants

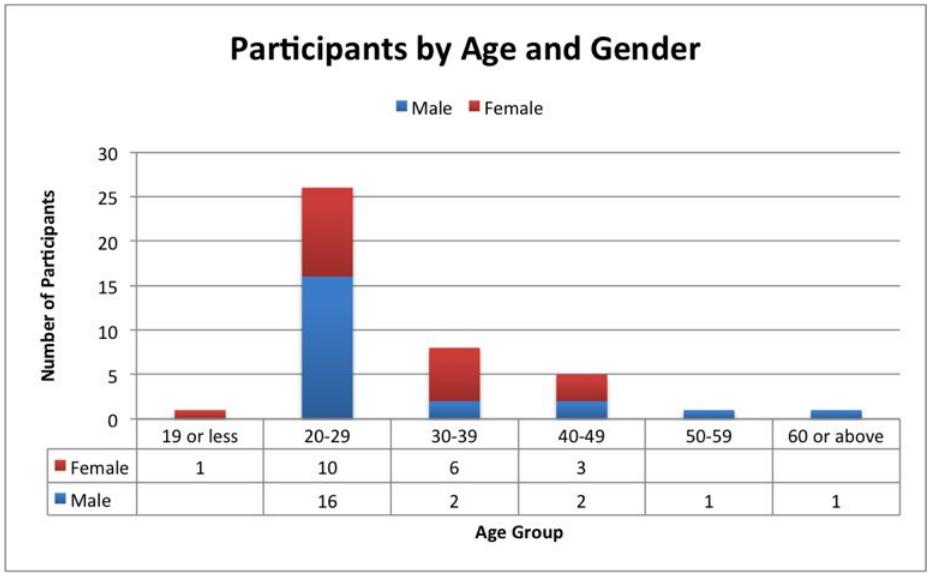
A total of 43 participants took part in this experiment in return for a £10 book voucher. One participant is colour-blind, and thus his data was removed.

This leaves to a total of 42 participants in our study. Among these, 22 participants are male, and 20 are female. All participants were recruited from the University of Oxford and related communities, with several disciplines including Computer Science, Engineering, Mathematics, Statistics, Zoology, Medical Sciences, Business, and Education. 27 of these participants are the university students, and another 15 participants are the university staffs. Out of the total participants, 26 participants are in 20-29 age range, 8 participants are in 30-39 age range, 5 participants are in 40-49 age range, and another 3 participants belong to each of the 3 age ranges: 19 or less, 50-59, and 60 or above, respectively. Figure 5.3(a) illustrates the participants' demographics information.

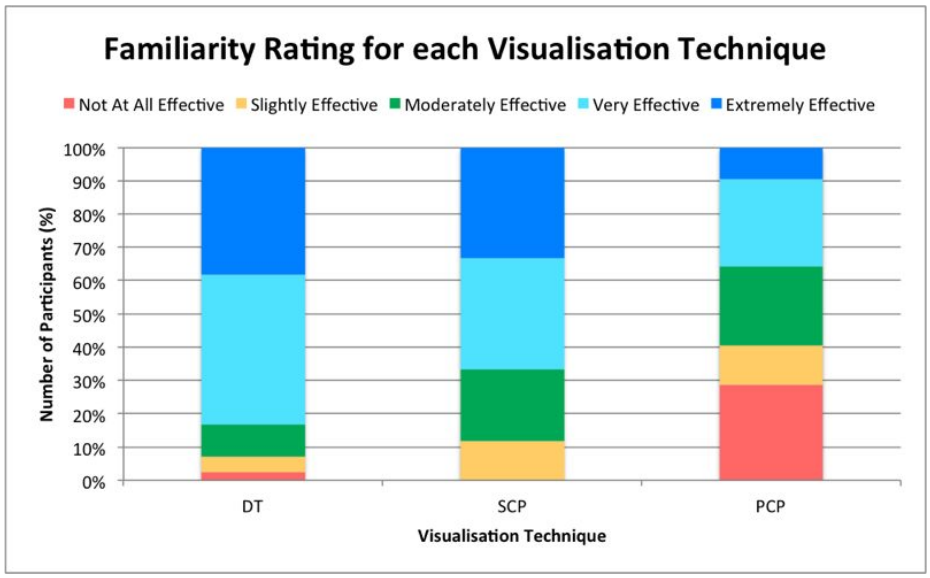
All participants have normal or corrected to normal vision. Most participants are familiar with DT and SCP technique, whereas only about half of the participants are familiar with PCP. Figure 5.3(b) displays the familiarity ratings for all participants, which are categorised by the five-level Likert scale as introduced in Section 3.3.

5.5.2 Apparatus

We generated the visual stimuli using custom software written in Java and JavaScript. The stimuli were then displayed to the participants through custom-made software program, written in Java. The software was run in full screen mode. The implementation process of stimuli generation and software development is described in Sections 4.6 and 4.7 respectively.



(a) Demographics information including age and gender.



(b) Familiarity rating for each visualisation technique.

Figure 5.3: Participants' demographics information and familiarity rating.

The experiments were run using computers with 3.7 GiB of RAM, 3.30 GHz quad-core Intel core i5-3550 processors running on Fedora, a Linux based operating system, with GNOME version 3.4.2. The total of 8 to 15 computers were used in each experiment session. Each computer had 24 inches Dell's LCD at 1920x1200 (16:10) resolution and sRGB colour mode display. We adjusted the monitors to the same brightness and level of contrasts. Each participant was required to interact with the software using the mouse at the desk.

5.5.3 Procedure

The time taken to complete this study was approximately 40 to 80 minutes in total, including 15 minutes of introduction. The time spent was varied according to the time that the participants used to read the instructions, perform each trial, and have a break.

Prior to the experiment, the experimenter gave a brief introduction, using the self-paced presentation slides, to familiarise the participants with the study. The introduction included the explanation on the three visualisation techniques, followed by the four visualisation tasks. During the presentation, after each technique or each task had been presented, the participants were allowed to ask questions. This was in order to ensure that the participants understand each section of the presentation. The participants were also instructed to finish each trial as accurately and as quickly as possible, and were informed that each visualisation image is independent to one another.

After the introduction session had finished, the experiment began. The participants must first provide their demographics information and familiarity rating in the software program. Then, the participants were required to undertake a training session with 12 training trials in total. Each trial was used to familiarise the participants with each visualisation task and each visualisation technique. After the participants completed the training trials, the main experiment was started.

The main experiment consisted of 72 trials, separated into 24 sections with 3 trials each. Before each section, an instruction was given on the screen indicating the visualisation task for the section. In each section, there were 3 trials for the three visualisation techniques, each had 90 seconds of time limitation. If the participants cannot select an answer in time, there would be no answer recorded for that trial, and the participants were asked to move on to the next trial.

After each section, the participants were allowed to take a short break before continuing to the next one. When all tasks had been completed, the participants were required to rate the effectiveness of each visualisation technique due to each visualisation task.

Sample pictures for the experiments can be found in Appendix E.

Chapter 6

Result Analysis

We use descriptive and inferential statistics to analyse the results for objective measures, accuracy and response time. Descriptive analysis identifies mean and standard deviation for each result, while inferential analysis provides statistical evidence for the conclusion. We also analyse the effectiveness rating for each visualisation technique and other relevant statistics.

In this chapter, we first give the summary of the result analyses, then explain the detail of each analysis for each visualisation task. In the last section, we further provide analyses for reading time and choice selection. All results are provided in bar charts in this chapter; the data table for the results can be found in Appendix F.

6.1 Result Summary

Our results **confirm the four hypotheses**, as stated in Section 3.1. The results for the four visualisation tasks can be summarised as follows:

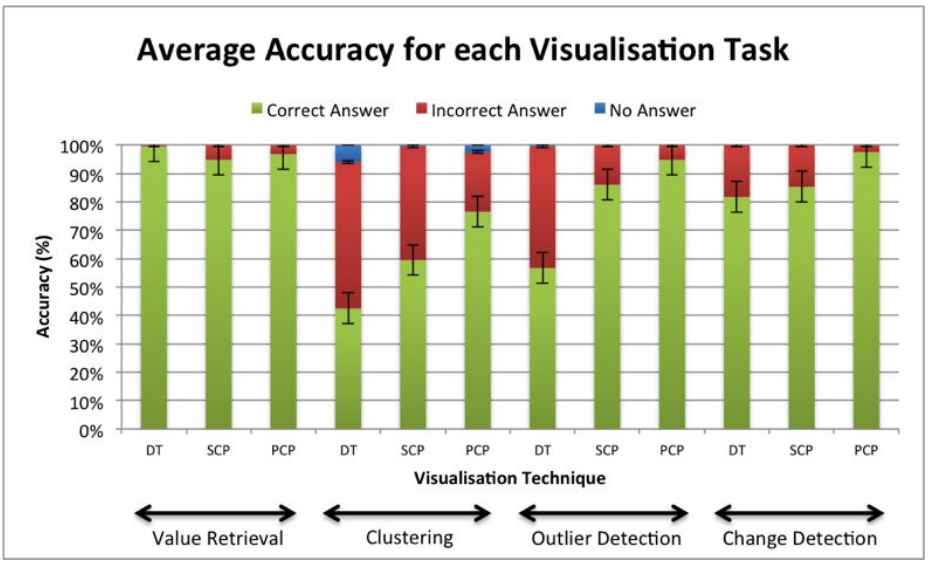
R1: DT $\succ$ PCP $\succ$ SCP

R2: PCP $\succ$ SCP $\succ$ DT

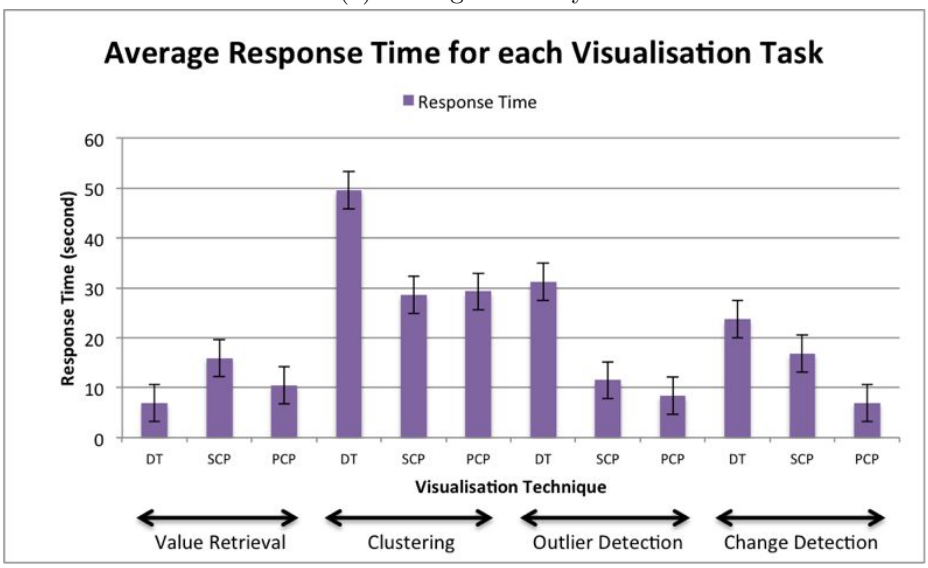
R3: PCP $\succ$ SCP $\succ$ DT

R4: PCP $\succ$ SCP $\succ$ DT

For value retrieval task (R1), DT is the obvious better technique, while PCP displays some advantages over SCP in terms of response time. In clustering task (R2), PCP is significantly better than DT and also yields higher accuracy performance than SCP when there is only 1 variable that exhibits obvious clusters. Considering the outlier detection task (R3), PCP is the better visualisation technique, providing higher user performance than DT in all cases, and higher than SCP when the outliers are based on different correlation factor



(a) Average accuracy.



(b) Average response time.

Figure 6.1: Performance analyses for each visualisation task.

only. For change detection task (R4), PCP holds advantages over the other two techniques in terms of both accuracy and response time, especially when all data points have a lot of changes in values.

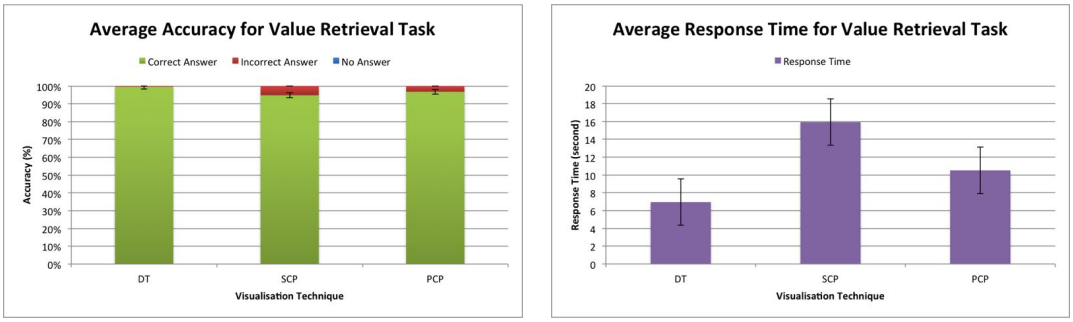
Overall, our results show that PCP is better than the other two techniques in three visualisation tasks: clustering, outlier detection, and change detection. For value retrieval task, PCP also holds advantages over SCP, which is complementary to the research paper of Kuang *et al.* [KZZM12], who concluded that PCP has advantages in the data set with low dimensionality and low density. Nevertheless, both visualisation techniques yield lower per-

formance than DT, which is more suitable for value retrieval task. Figure 6.1 demonstrates the performance analyses for each visualisation task.

6.2 Result Analyses for Value Retrieval Task

Following subsections describe the performance analyses for both objective and subjective measures in value retrieval task.

6.2.1 Accuracy and Response Time



(a) Average accuracy. (b) Average response time.

Figure 6.2: Performance analysis for value retrieval task.

Figure 6.2 shows average accuracy and average response time for each visualisation technique.

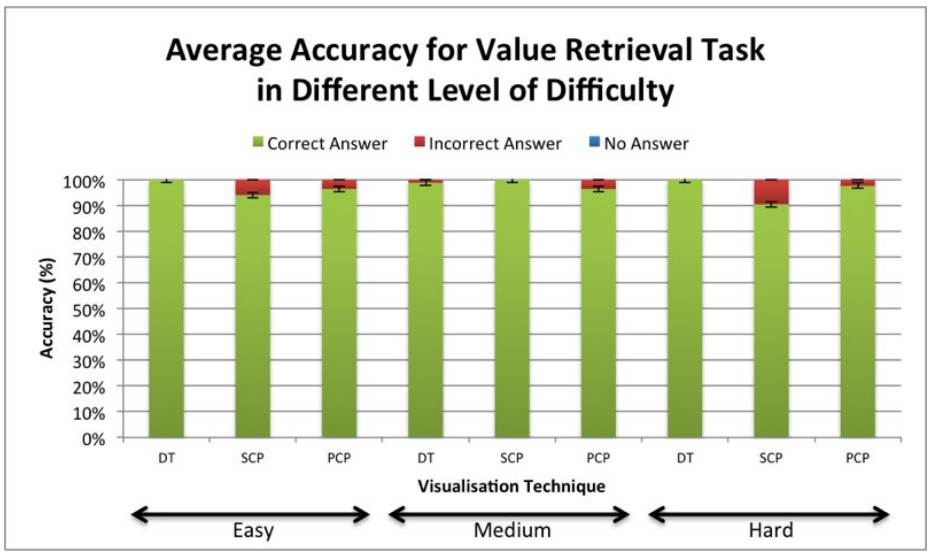
Accuracy. Mauchly's Test of Sphericity indicates that the assumption of sphericity had been violated ($p = .016$). Therefore, we use ANOVA analysis with Huynh-Feldt Corrections ($\epsilon = .875$), which suggests that there is a significant main effect of the techniques in accuracy ($F(1.750, 71.739) = 3.619, p = .037$).

The result prompts us to further perform t-test analysis, which reveals that DT yields significantly higher accuracy than SCP ($p = .037$). However, there is no significant effect between DT and PCP ($p = .153$), and between SCP and PCP ($p = 1$).

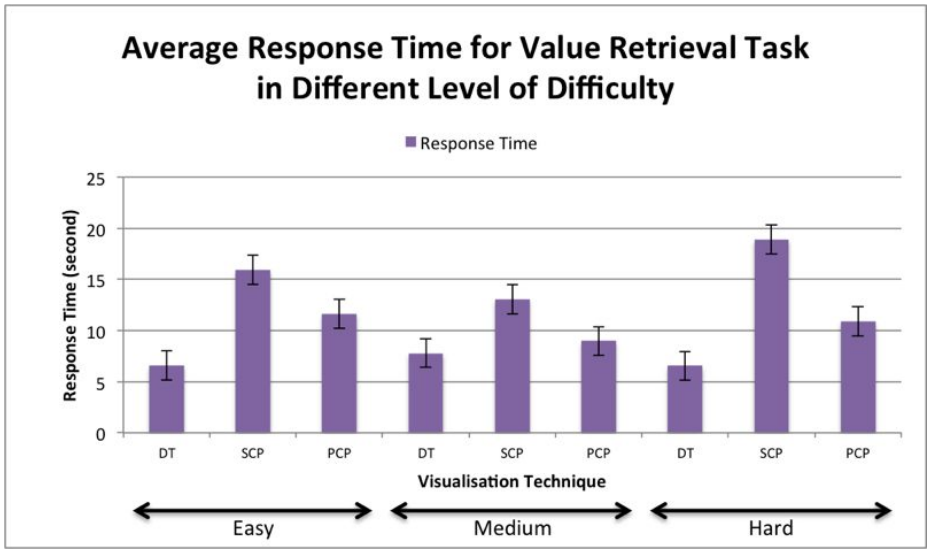
Response Time. Mauchly's Test of Sphericity reports that the assumption of sphericity had not been met ($p < .001$). As a consequence, the ANOVA analysis with Greenhouse-Geisser Corrections ($\epsilon = .607$) have been used. It shows a significant main effect in response time ($F(1.214, 49.784) = 111.12, p < .001$).

Further t-test analysis establishes that all techniques are significantly different from each other with DT yielding the fastest response time, followed by PCP and SCP respectively (all $p < .001$).

The user performances are additionally analysed for each level of task difficulty separately, as in the following subsections. Figure 6.3 summarises average accuracy and average response time for each level of difficulty in value retrieval task.



(a) Average accuracy.



(b) Average response time.

Figure 6.3: Performance analysis for value retrieval task in different level of difficulty.

6.2.1.1 Easy Level of Task Difficulty

Accuracy. Mauchly's Test of Sphericity shows that the assumption of sphericity had not been met ($p = .001$). The ANOVA analysis with Huynh-Feldt Corrections ($\epsilon = .804$) is then used. It reports that there is no significant difference in accuracy among the three visualisation techniques ($F(1.607, 65.905) = 2.170, p = .132$).

Response Time. Mauchly's Test of Sphericity indicates that the assumption of sphericity had been violated ($p < .001$). We therefore use ANOVA analysis with Greenhouse-Geisser Corrections ($\epsilon = .711$), which suggests that there is a significant main effect of the techniques in response time ($F(1.422, 58.288) = 59.358, p < .001$).

Further t-test analysis reveals that the response times in all techniques are significantly different. DT yields the fastest response time, followed by PCP and SCP respectively (all $p < .001$).

6.2.1.2 Medium Level of Task Difficulty

Accuracy. Mauchly's Test of Sphericity indicates that the assumption of sphericity had been violated ($p < .001$). Consequently, we use ANOVA analysis with Greenhouse-Geisser Corrections ($\epsilon = .692$), which suggests that there is no significant effect of the techniques in accuracy ($F(1.383, 56.714) = 1.783, p = .186$).

Response Time. Mauchly's Test of Sphericity reports that the assumption of sphericity had not been met ($p < .001$). The ANOVA analysis with Greenhouse-Geisser Corrections ($\epsilon = .662$) shows a significant main effect of the visualisation techniques in response time ($F(1.324, 54.276) = 26.083, p < .001$).

t-test analysis confirms that all three techniques have significant differences. SCP yields overall the slowest response time relative to the other two techniques (all $p < .001$), and DT yields faster response time than PCP ($p = .018$).

6.2.1.3 Hard Level of Task Difficulty

Accuracy. Mauchly's Test of Sphericity demonstrates that the assumption of sphericity had been violated ($p < .001$). As a result, we use ANOVA analysis with Greenhouse-Geisser Corrections ($\epsilon = .631$), which reports a significant main effect of the techniques in accuracy ($F(1.261, 51.717) = 4.717, p = .027$).

t-test analysis reveals that DT yields significantly higher accuracy than SCP ($p = .029$). However, no significant difference in accuracy is detected between DT and PCP ($p = .479$), and between SCP and PCP ($p = .25$).

Response Time. Mauchly's Test of Sphericity establishes that the assumption of sphericity had been violated ($p < .001$). The ANOVA analysis with Greenhouse-Geisser Corrections ($\epsilon = .587$) indicates a significant effect of the techniques in response time ($F(1.174, 48.114) = 98.97, p < .001$).

t-test analysis shows that all three visualisation techniques are significantly different from each other. DT yields the fastest response time, followed by PCP and SCP respectively (all $p < .001$).

6.2.2 Performance Summary

DT is clearly the better technique for value retrieval task.

Even though the three visualisation techniques provide similar accuracy, DT takes significantly shorter response time regardless of the level of difficulty. Comparing PCP and SCP, the former technique takes significantly shorter response time in every level of difficulty.

The relative performance relationships among the three techniques in accuracy and response time are summarised below:

Accuracy

All Levels: DT(5.98) $\succ$ SCP(5.69), [DT(5.98), PCP(5.81)], [PCP(5.81), SCP(5.69)]

Easy Level: [DT(2), PCP(1.93), SCP(1.88)]

Medium Level: [SCP(2), DT(1.98), PCP(1.93)]

Hard Level: DT(2) $\succ$ SCP(1.81), [DT(2), PCP(1.95)], [PCP(1.95), SCP(1.81)]

Response Time

All Levels: DT(6.97) $\succ$ PCP(10.49) $\succ$ SCP(15.95)

Easy Level: DT(6.58) $\succ$ PCP(11.63) $\succ$ SCP(15.93)

Medium Level: DT(7.78) $\succ$ PCP(8.97) $\succ$ SCP(13.03)

Hard Level: DT(6.54) $\succ$ PCP(10.88) $\succ$ SCP(18.90)

6.2.3 Effectiveness Rating

The subjective feedback for value retrieval task is consistent with the quantitative results.

More than 90% of the participants rate DT as an effective technique, with more than half specifying that it is extremely effective. PCP is the second preferred technique. Most participants agree that it is very effective. SCP, on the other hand, has the least effectiveness rating with only about half considering it as moderately effective and above. The rest of the participants judge SCP as a slightly effective technique.

Figure 6.4 illustrates the effectiveness rating for different techniques in value retrieval task.

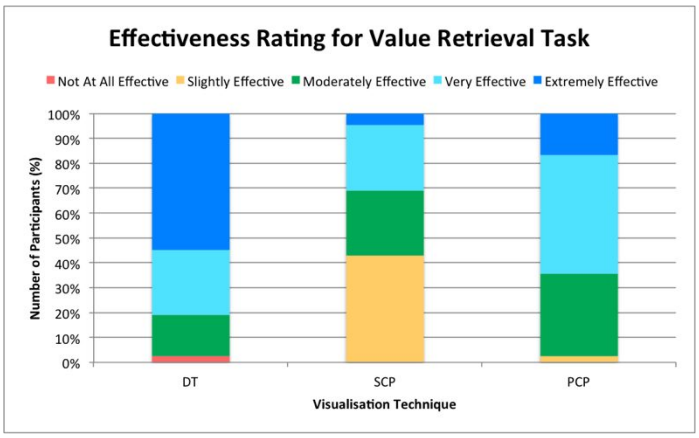
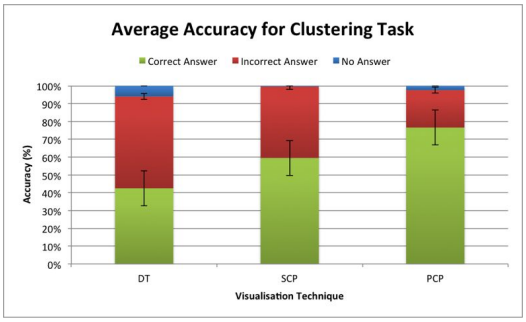


Figure 6.4: Participants' effectiveness rating for value retrieval task.

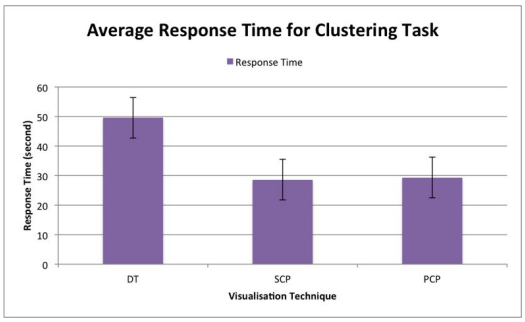
6.3 Result Analyses for Clustering Task

The performance analyses for clustering task, in terms of quantitative and subjective measures, is explained in the following subsections.

6.3.1 Accuracy and Response Time



(a) Average accuracy.



(b) Average response time.

Figure 6.5: Performance analysis for clustering task.

Figure 6.5 illustrates average accuracy and average response time for each visualisation technique in clustering task.

Accuracy. Mauchly's Test of Sphericity verifies that the assumption of sphericity had been met ($p = .921$). ANOVA analysis reports that there is a significant main effect of the techniques in accuracy ($F(2, 82) = 36.117, p < .001$).

Further t-test analysis demonstrates that the accuracies in all techniques are significantly different. PCP yields the overall highest accuracy, followed by SCP and DT respectively (all $p < .001$).

Response Time. Mauchly's Test of Sphericity identifies that the assumption of sphericity had been violated ($p = .027$). We thus use ANOVA analysis with Huynh-Feldt Corrections ($\epsilon = .892$), which shows a significant main effect of the techniques in response time ($F(1.785, 73.174) = 50.225, p < .001$).

t-test analysis reveals that the source of the main effect is DT technique. Response time in DT is significantly slower than the other two techniques (all $p < .001$), whereas there is no significant difference between SCP and PCP ($p = 1$).

The following subsections explain the analyses on user performance for each level of task difficulty. The summary of average accuracy and average response time for each difficulty in clustering task is displayed in figure 6.6.

6.3.1.1 Easy Level of Task Difficulty

Accuracy. Mauchly's Test of Sphericity shows that the assumption of sphericity had not been met ($p < .001$). The ANOVA analysis with Greenhouse-Geisser Corrections ($\epsilon = .694$) is then used. The analysis indicates that there is a significant main effect of the techniques in accuracy ($F(1.389, 56.932) = 17.771, p < .001$).

t-test analysis establishes that DT is the source of the main effect, yielding the lowest accuracy relative to the other two techniques (all $p < .001$). Nevertheless, SCP and PCP have no significant difference in accuracy ($p = 1$).

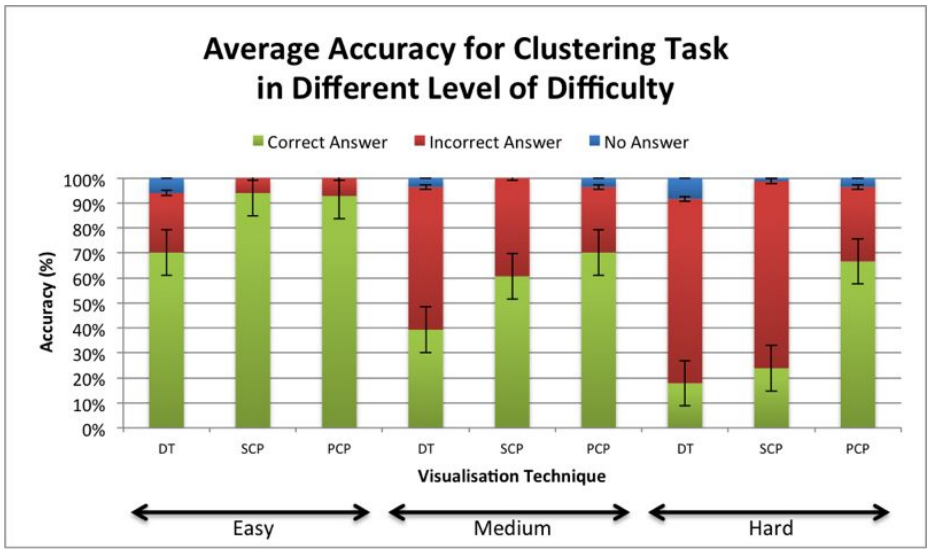
Response Time. Mauchly's Test of Sphericity reports that the assumption of sphericity had been violated ($p = .006$). The ANOVA analysis with Huynh-Feldt Corrections ($\epsilon = .844$) demonstrates a significant main effect of the techniques in response time ($F(1.689, 69.243) = 72.106, p < .001$).

t-test analysis confirms DT as the source of the main effect. Response time in DT is significantly slower than SCP and PCP (all $p < .001$), while there is no significant difference between the latter two techniques ($p = 1$).

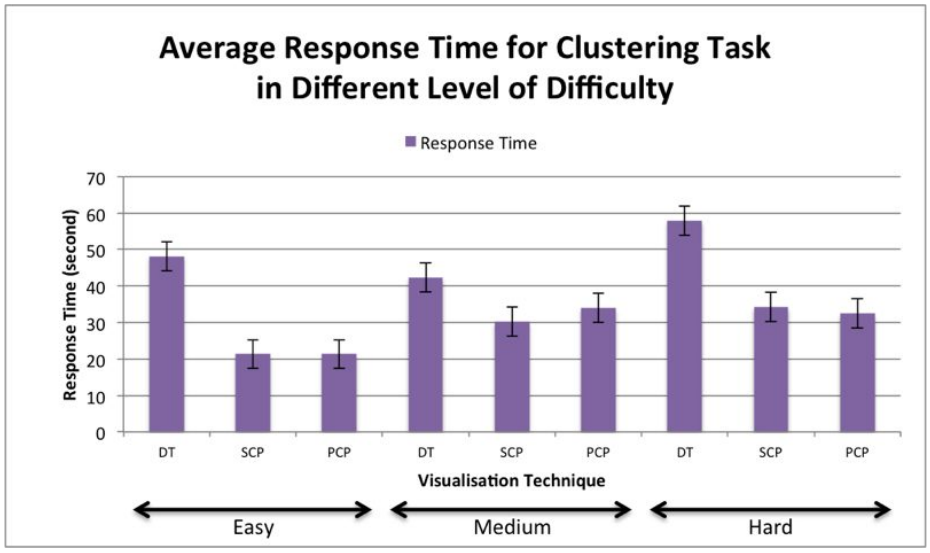
6.3.1.2 Medium Level of Task Difficulty

Accuracy. Mauchly's Test of Sphericity indicates that the assumption of sphericity had not been violated ($p = .590$). ANOVA analysis suggests that there is a significant main effect of the techniques in accuracy ($F(2, 82) = 9.219, p < .001$).

t-test analysis reveals that DT yields overall lower accuracy than SCP ($p = .028$) and PCP ($p < .001$). However, there is no significant difference in accuracy between SCP and PCP ($p = .629$).



(a) Average accuracy.



(b) Average response time.

Figure 6.6: Performance analysis for clustering task in different level of difficulty.

Response Time. Mauchly's Test of Sphericity shows that the assumption of sphericity had been violated ($p = .031$). We therefore use ANOVA analysis with Huynh-Feldt Corrections ($\epsilon = .897$), which establishes a significant main effect of the techniques in response time ($F(1.793, 73.515) = 5.288, p = .009$).

t-test analysis reports that DT yields significantly slower response time than SCP ($p = .005$). Yet, no significant difference in response time is detected between DT and PCP ($p = .213$), and between SCP and PCP ($p = .760$).

6.3.1.3 Hard Level of Task Difficulty

Accuracy. Mauchly's Test of Sphericity demonstrates that the assumption of sphericity had not been met ($p = .004$). For this reason, we use ANOVA analysis with Huynh-Feldt Corrections ($\epsilon = .833$), which shows that there is a significant main effect of the techniques in accuracy ($F(1.666, 68.313) = 33.87, p < .001$).

t-test analysis indicates that PCP is the source of the main effect, yielding the highest accuracy relative to the other two techniques (all $p < .001$), whereas DT and SCP have no significant difference in accuracy ($p = .69$).

Response Time. Mauchly's Test of Sphericity establishes that the assumption of sphericity had been met ($p = .506$). The ANOVA analysis reports a significant main effect of the techniques in response time ($F(2, 82) = 36.173, p < .001$).

Further t-test analysis reveals that DT is the source of the main effect. Response time in DT is significantly slower than SCP and PCP (all $p < .001$). Nevertheless, there is no significant difference in response time between SCP and PCP ($p = 1$).

6.3.2 Performance Summary

For clustering task, SCP and PCP are the two better visualisation techniques.

In terms of response time, there is no significant difference between the two techniques in every level of difficulty. Nevertheless, the accuracy performance in PCP becomes significantly better than SCP when there is only 1 variable that exhibits obvious clusters (hard level of task difficulty). The overall results of all-level tasks also indicates that PCP have the highest accuracy, followed by SCP and DT respectively.

As a conclusion, for clustering task, PCP have advantages over the other two techniques in terms of accuracy, particularly in the hard-level tasks.

The relative performance relationships among the three techniques in accuracy and response time are summarised below:

Accuracy

All Levels: PCP(4.60) $\succ$ SCP(3.57) $\succ$ DT(2.55)

Easy Level: [SCP(1.88), PCP(1.86)] $\succ$ DT(1.40)

Medium Level: [PCP(1.40), SCP(1.21)] $\succ$ DT(0.79)

Hard Level: PCP(1.33) $\succ$ [SCP(0.48), DT(0.36)]

Response Time

All Levels: [SCP(28.59), PCP(29.26)] $\succ$ DT(49.49)

Easy Level: [PCP(21.33), SCP(21.33)] $\succ$ DT(48.19)

Medium Level: SCP(30.27) $\succ$ DT(42.39), [SCP(30.27), PCP(34.05)], [PCP(34.05), DT(42.39)]

Hard Level: [PCP(32.41), SCP(34.16)] $\succ$ DT(57.90)

6.3.3 Effectiveness Rating

Conforming to the objective results, the subjective results conclude that DT is the least suitable technique for clustering task.

Majority of the participants state that DT is not at all effective, whereas more than 90% of the participants rate both SCP and PCP as moderately effective and above. The latter two techniques have similar effectiveness rating, which is conform to the quantitative results. More participants prefer SCP, but it is just a slightly difference.

The effectiveness ratings for each technique in clustering task are summarised in figure 6.7.

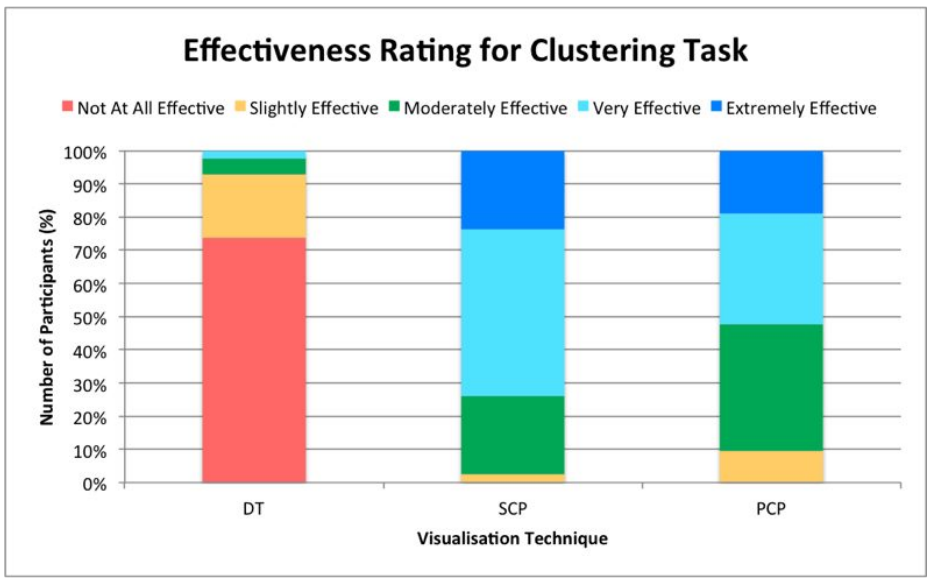
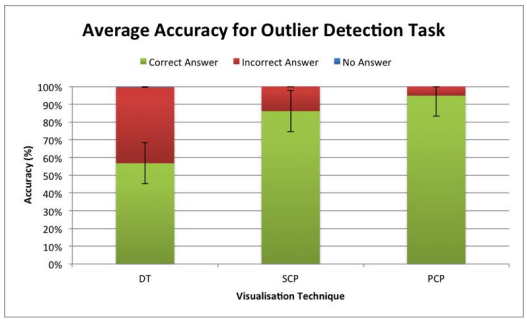


Figure 6.7: Participants' effectiveness rating for clustering task.

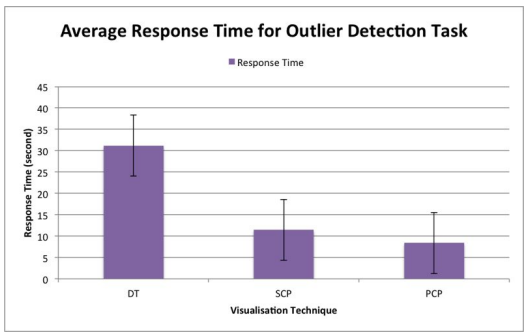
6.4 Result Analyses for Outlier Detection Task

Following subsections go into detail of the performance analyses for outlier detection task in both objective and subjective measures.

6.4.1 Accuracy and Response Time



(a) Average accuracy.



(b) Average response time.

Figure 6.8: Performance analysis for outlier detection task.

Figure 6.8 displays average accuracy and average response time for each visualisation technique in outlier detection task.

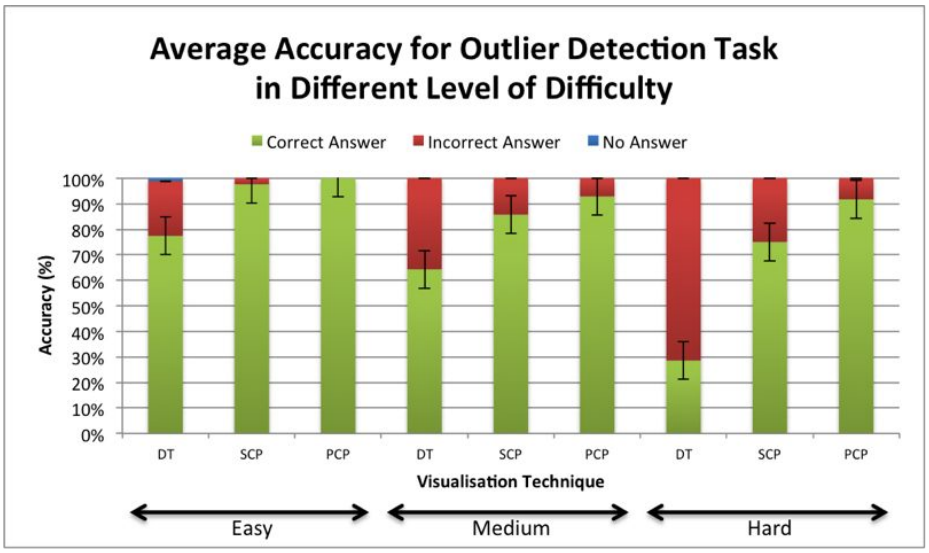
Accuracy. Mauchly's Test of Sphericity establishes that the assumption of sphericity had not been met ($p = .015$). As a result, we use ANOVA analysis with Huynh-Feldt Corrections ($\epsilon = .872$), which shows a significant main effect of the techniques in accuracy ($F(1.744, 71.522) = 58.173, p < .001$).

t-test analysis reports that the accuracies in all techniques are significantly different with DT yielding overall lower accuracy than the other two techniques (all $p < .001$), and SCP yielding lower accuracy than PCP ($p = .018$).

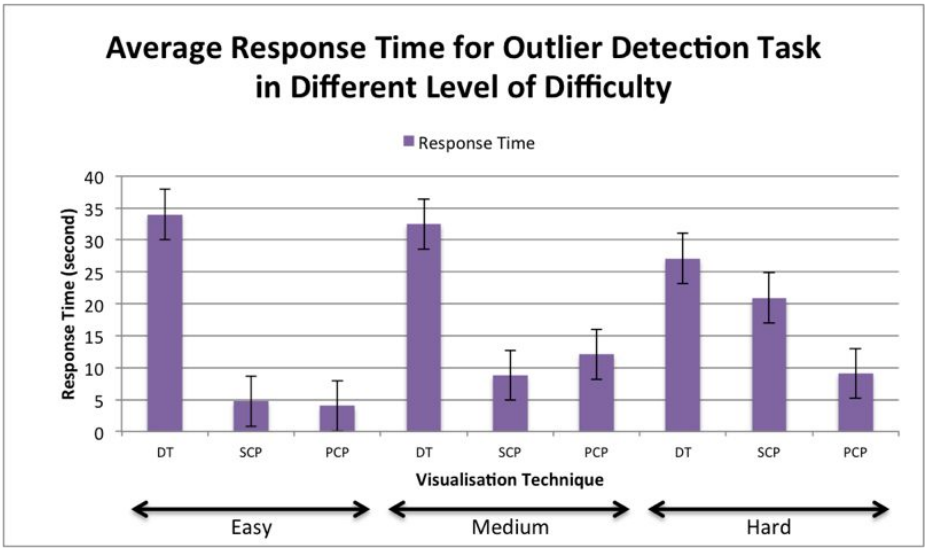
Response Time. Mauchly's Test of Sphericity demonstrates that the assumption of sphericity had been violated ($p = .002$). The ANOVA analysis with Huynh-Feldt Corrections ($\epsilon = .815$) identifies that there is a significant main effect of the techniques in response time ($F(1.630, 66.810) = 215.48, p < .001$).

Further t-test analysis confirms that all three visualisation techniques are significantly different from each other. DT yields the slowest response time relative to the other two techniques (all $p < .001$), while SCP yields slower response time than PCP ($p = .002$).

The analyses on user performance in different level of difficulty are reported in the following subsections. Figure 6.9 illustrates average accuracy and average response time for each level of difficulty in outlier detection task.



(a) Average accuracy.



(b) Average response time.

Figure 6.9: Performance analysis for outlier detection task in different level of difficulty.

6.4.1.1 Easy Level of Task Difficulty

Accuracy. Mauchly's Test of Sphericity shows that the assumption of sphericity had not been met ($p < .001$). We therefore use ANOVA analysis with Greenhouse-Geisser Corrections ($\epsilon = .574$), which identifies that there is a significant main effect of the techniques in accuracy ($F(1.148, 47.054) = 15.791, p < .001$).

t-test analysis reveals that DT is the source of the main effect. DT yields significantly lower accuracy than SCP ($p = .002$) and PCP ($p < .001$). Yet, there is no significant difference in accuracy between SCP and PCP ($p = .479$).

Response Time. Mauchly's Test of Sphericity indicates that the assumption of sphericity had not been met ($p < .001$). The ANOVA analysis with Greenhouse-Geisser Corrections ($\epsilon = .550$) reports a significant main effect of the techniques in response time ($F(1.100, 45.084) = 201.23, p < .001$).

t-test analysis demonstrates that response time in DT is significantly slower than the other two techniques (all $p < .001$), while SCP and PCP have no significant difference in response time ($p = .52$).

6.4.1.2 Medium Level of Task Difficulty

Accuracy. Mauchly's Test of Sphericity reports that the assumption of sphericity had not been violated ($p = .564$). The ANOVA analysis indicates that there is a significant main effect of the techniques in accuracy ($F(2, 82) = 13.904, p < .001$).

t-test analysis identifies DT as the source of the main effect with significantly lower accuracy than SCP ($p = .003$) and PCP ($p < .001$). However, no significant difference in accuracy is detected between SCP and PCP ($p = .675$).

Response Time. Mauchly's Test of Sphericity verifies that the assumption of sphericity had been met ($p = .822$). The ANOVA analysis shows a significant main effect of the techniques in response time ($F(2, 82) = 85.275, p < .001$).

t-test analysis reveals that DT yields significantly slower response time than the other two techniques (all $p < .001$). Nevertheless, there is no significant difference in response time between SCP and PCP ($p = .25$).

6.4.1.3 Hard Level of Task Difficulty

Accuracy. Mauchly's Test of Sphericity shows that the assumption of sphericity had been met ($p = .081$). The ANOVA analysis demonstrates that there is a significant main effect of the techniques in accuracy ($F(2, 82) = 52.449, p < .001$).

t-test analysis confirms that accuracies in all three visualisation techniques are significantly different from each other with DT yielding the lowest accuracy relative to the other two techniques (all $p < .001$). Besides, accuracy in PCP is significantly higher than accuracy in SCP ($p = .035$).

Response Time. Mauchly's Test of Sphericity establishes that the assumption of sphericity had been met ($p = .093$). The ANOVA analysis reports that there is a significant main effect of the techniques in response time ($F(2, 82) = 36.329, p < .001$).

t-test analysis reports that PCP yields the fastest response time relative to the other two techniques (all $p < .001$), whereas there is just a very small effect in response time between SCP and DT ($p = .05$).

6.4.2 Performance Summary

DT is undoubtedly not suitable for outlier detection task as it has lower accuracy and takes longer time, whereas PCP and SCP are the two better techniques.

Besides, while PCP seems to have comparable performance with SCP, it shows advantages when the outliers are based on different correlation factor only. In these hard-level tasks, PCP provides better performance than SCP in both accuracy and response time.

This result also holds in the all-level analysis, indicating that PCP is the better visualisation technique in outlier detection task.

The relative performance relationships among the three techniques in accuracy and response time are summarised below:

Accuracy

All Levels: PCP(5.69) $\succ$ SCP(5.17) $\succ$ DT(3.40)

Easy Level: [PCP(2), SCP(1.95)] $\succ$ DT(1.55)

Medium Level: [PCP(1.86), SCP(1.71)] $\succ$ DT(1.29)

Hard Level: PCP(1.83) $\succ$ SCP(1.50) $\succ$ DT(0.57)

Response Time

All Levels: PCP(8.40) $\succ$ SCP(11.49) $\succ$ DT(31.19)

Easy Level: [PCP(4.01), SCP(4.74)] $\succ$ DT(34.00)

Medium Level: [SCP(8.81), PCP(12.10)] $\succ$ DT(32.49)

Hard Level: PCP(9.10) $\succ$ [SCP(20.92), DT(27.09)]

6.4.3 Effectiveness Rating

SCP and PCP are considered as the two preferred techniques for outlier detection task in the subjective rating, as same as in the quantitative results.

About 80% of the participants evaluate DT as not at all effective and slightly effective, while more than 90% consider the other two techniques as moderately effective and above.

SCP and PCP have almost identical effectiveness rating, which is compatible with the objective results.

Figure 6.10 displays the effectiveness rating for outlier detection task.

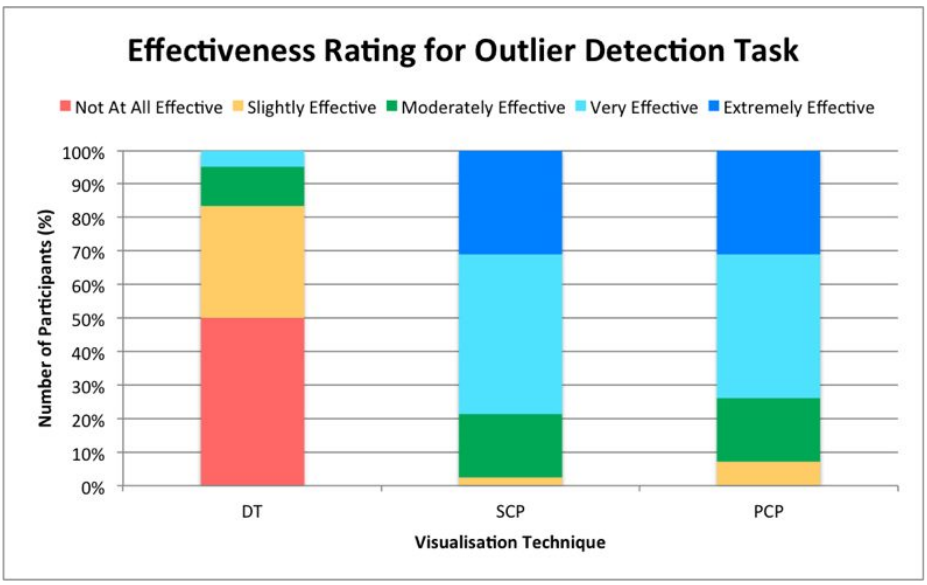
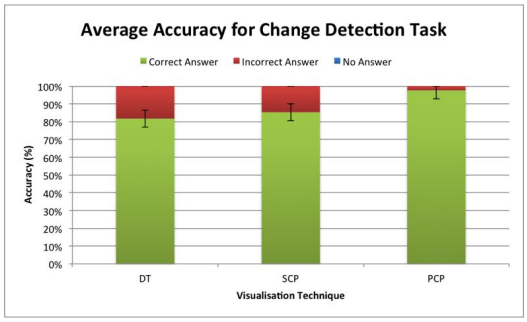


Figure 6.10: Participants' effectiveness rating for outlier detection task.

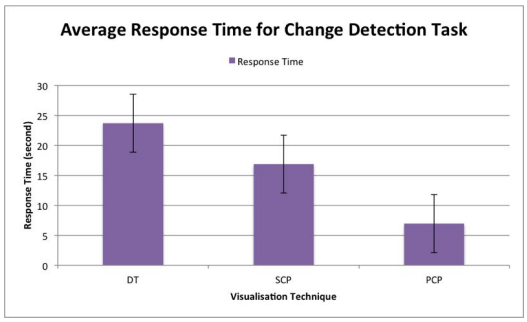
6.5 Result Analyses for Change Detection Task

The performance analyses in quantitative and subjective measures for change detection task are discussed in the following subsections.

6.5.1 Accuracy and Response Time



(a) Average accuracy.



(b) Average response time.

Figure 6.11: Performance analysis for change detection task.

Figure 6.11 shows average accuracy and average response time for each visualisation technique in change detection task.

Accuracy. Mauchly's Test of Sphericity reports that the assumption of sphericity had been met ($p = .693$). The ANOVA analysis indicates that there is a significant main effect of the techniques in accuracy ($F(2, 82) = 21.298, p < .001$).

t-test analysis reveals that PCP is the source of the main effect, yielding significantly higher accuracy than the other two techniques (all $p < .001$). Meanwhile, no significant difference in accuracy is detected between SCP and DT ($p = .423$).

Response Time. Mauchly's Test of Sphericity establishes that the assumption of sphericity had not been violated ($p = .286$). The ANOVA analysis shows a significant main effect of the techniques in response time ($F(2, 82) = 77.956, p < .001$).

t-test analysis indicates that the response times in all three visualisation techniques are significantly different from each other with PCP yielding the fastest response time, followed by SCP and DT respectively (all $p < .001$).

The following subsections explain the user performance analyses in different level of difficulty. Figure 6.12 summarises average accuracy and average response time for each level of difficulty in change detection task.

6.5.1.1 Easy Level of Task Difficulty

Accuracy. Mauchly's Test of Sphericity indicates that the assumption of sphericity had been met ($p = .053$). The ANOVA analysis suggests that there is a significant main effect of the technique in accuracy ($F(2, 82) = 6.017, p = .004$).

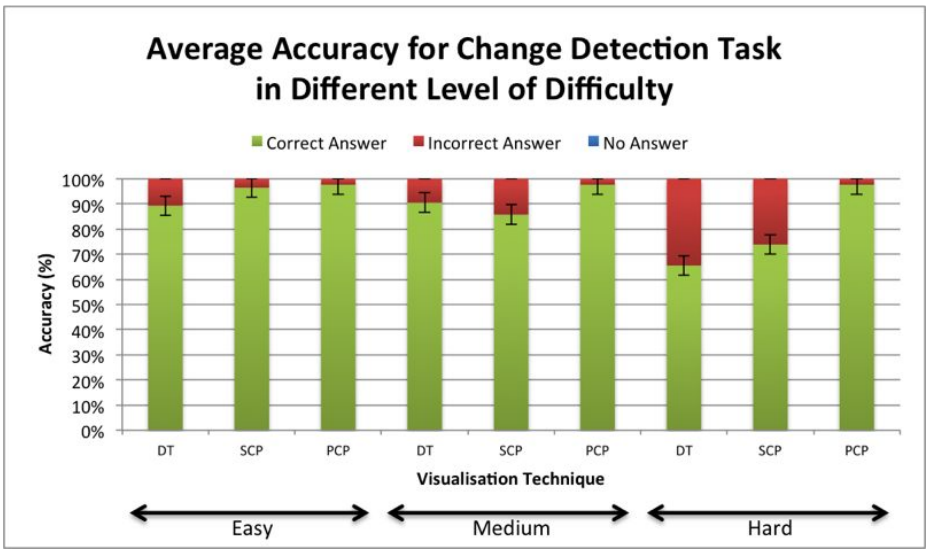
t-test analysis shows that DT yields significantly lower accuracy than SCP ($p = .037$) and PCP ($p = .02$), while there is no significant difference in accuracy between SCP and PCP ($p = 1$).

Response Time. Mauchly's Test of Sphericity reports that the assumption of sphericity had been violated ($p < .001$). Thus, we use ANOVA analysis with Greenhouse-Geisser Corrections ($\epsilon = .698$), which demonstrates a significant main effect of the techniques in response time ($F(1.395, 57.213) = 27.172, p < .001$).

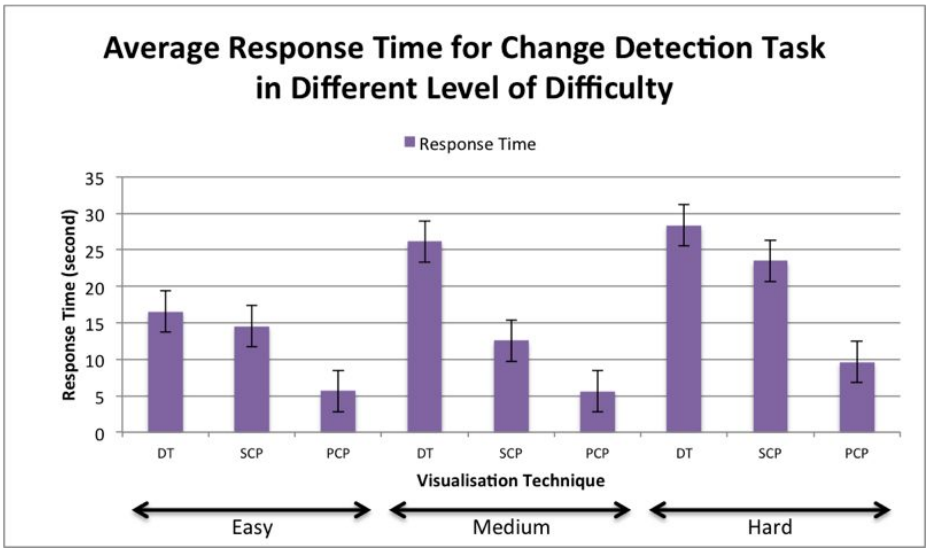
Further t-test analysis reveals that PCP is the source of the main effect, yielding the fastest response time relative to the other two techniques (all $p < .001$). Nevertheless, SCP and DT have no significant difference in response time ($p = .89$).

6.5.1.2 Medium Level of Task Difficulty

Accuracy. Mauchly's Test of Sphericity establishes that the assumption of sphericity had been met ($p = .158$). ANOVA analysis shows a significant main effect of the techniques in accuracy ($F(2, 82) = 4.079, p = .020$).



(a) Average accuracy.



(b) Average response time.

Figure 6.12: Performance analysis for change detection task in different level of difficulty.

t-test analysis reports that PCP yields significantly higher accuracy than SCP ($p = .008$). Yet, no significant difference in accuracy is detected between PCP and DT ($p = .0250$) and between DT and SCP ($p = .970$).

Response Time. Mauchly's Test of Sphericity indicates that the assumption of sphericity had not been met ($p < .001$). We therefore use the ANOVA analysis with Greenhouse-Geisser Corrections ($\epsilon = .734$), which suggests that there is a significant main effect of the techniques in response time ($F(1.468, 60.208) = 82.487, p < .001$).

t-test analysis confirms that all three techniques are significantly different from each other with PCP yielding the fastest response time, followed by SCP and DT respectively

(all $p < .001$).

6.5.1.3 Hard Level of Task Difficulty

Accuracy. Mauchly's Test of Sphericity indicates that the assumption of sphericity had been met ($p = .502$). ANOVA analysis reports that there is a significant main effect of the techniques in accuracy ($F(2, 82) = 19.181, p < .001$).

t-test analysis reveals that PCP is the source of the main effect. Accuracy in PCP is higher than the other two techniques (all $p < .001$), while there is no significant effect between SCP and DT ($p = .327$).

Response Time. Mauchly's Test of Sphericity shows that the assumption of sphericity had not been violated ($p = .342$). ANOVA analysis demonstrates that there is a significant main effect of the techniques in response time ($F(2, 82) = 47.279, p < .001$).

t-test analysis establishes that PCP yields the fastest response time relative to the other two techniques (all $p < .001$). However, SCP and DT have no significant difference in response time ($p = .068$).

6.5.2 Performance Summary

Regarding the change detection task, PCP is the noticeably better visualisation technique, taking significantly shorter response time in every level of difficulty.

In terms of accuracy, the performance difference between PCP and the other two techniques is small, but as the task gets harder, PCP seems to show the advantages. When all data points have a lot of changes in values, the accuracies to find the most changes data point(s) in SCP and DT are significantly less than in PCP.

As a consequence, PCP holds advantages in change detection task, especially when the task is hard.

The relative performance relationships among the three techniques in accuracy and response time are summarised below:

Accuracy

All Levels: PCP(5.86) $\succ$ [SCP(5.12), DT(4.90)]

Easy Level: [PCP(1.95), SCP(1.93)] $\succ$ DT(1.79)

Medium Level: PCP(1.95) $\succ$ SCP(1.71), [PCP(1.95), DT(1.81)], [DT(1.81), SCP(1.71)]

Hard Level: PCP(1.95) $\succ$ [SCP(1.48), DT(1.31)]

Response Time

All Levels: PCP(6.96) $\succ$ SCP(16.85) $\succ$ DT(23.68)

Easy Level: PCP(5.64) $\succ$ [SCP(14.53), DT(16.54)]

Medium Level: PCP(5.63) $\succ$ SCP(12.55) $\succ$ DT(26.15)

Hard Level: PCP(9.62) $\succ$ [SCP(23.49), DT(28.34)]

6.5.3 Effectiveness Rating

The effectiveness rating is agreeable with the quantitative results that PCP is the better technique in change detection task.

Only 1 participant evaluates PCP as slightly effective, whereas the rest rate it as moderately effective and above. Comparing SCP and DT, the former technique has overall higher effectiveness rating with no rating on not at all effective. DT, on the other hand, has above 20% of participants considering it as not at all effective.

The effectiveness ratings for each visualisation technique in change detection task are displayed in figure 6.13.

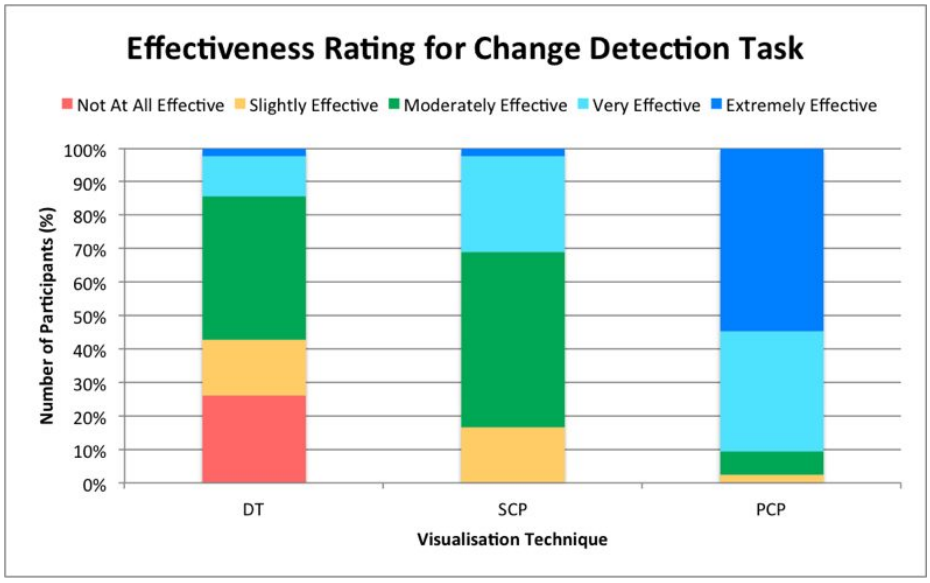


Figure 6.13: Participants' effectiveness rating for change detection task.

6.6 Other Relevant Statistics

Apart from the performance analysis, we also analyse other relevant statistics including the average time that each participant used to read questions (reading time), and the type of optional answers that the participants selected for each visualisation technique in the four different tasks (choice selection). The detail of each statistics is explained in the following subsections.

6.6.1 Reading Time

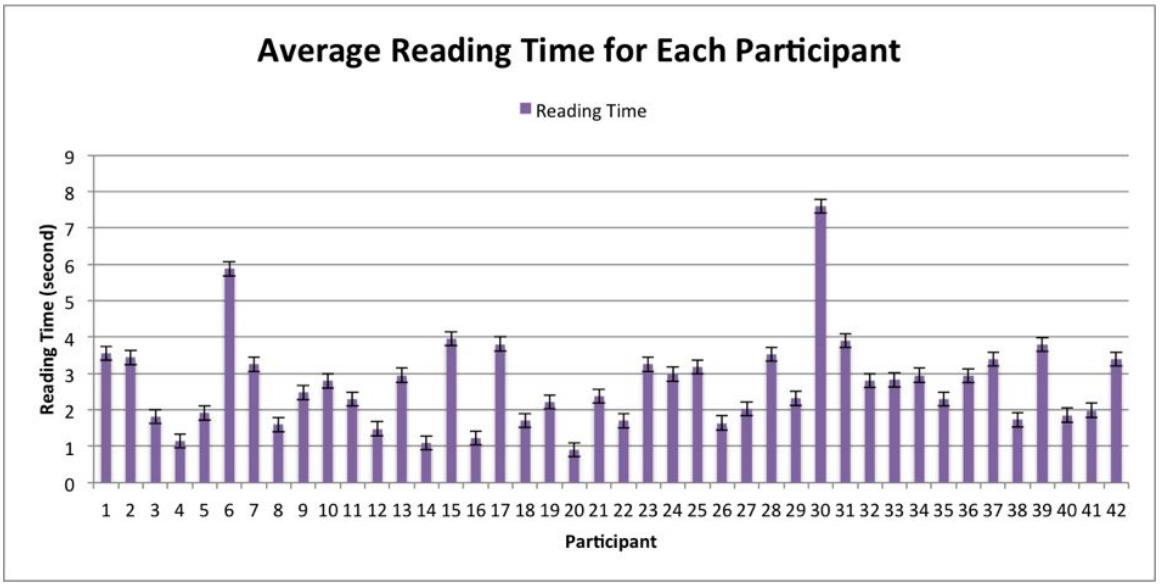


Figure 6.14: Average reading time for each participant.

Different participants used different amount of time to read the question in each trial. The average reading time for each participant is demonstrated in Figure 6.14.

Considering all 42 participants, the average reading time is 2.71 seconds with standard deviation of 1.26 seconds. The maximum time used is 7.59 seconds, whereas 0.91 seconds is the minimum reading time.

According to this result, if we measure the response time before giving the questions of the trials, the reading time can significantly affect the results as some participants took longer time to read than others. Excluding this reading time, as we did in our experiment, can help reducing this bias.

6.6.2 Choice Selection

Figure 6.15 summarises the choice selection for each visualisation task. The result for each task is reported as follows:

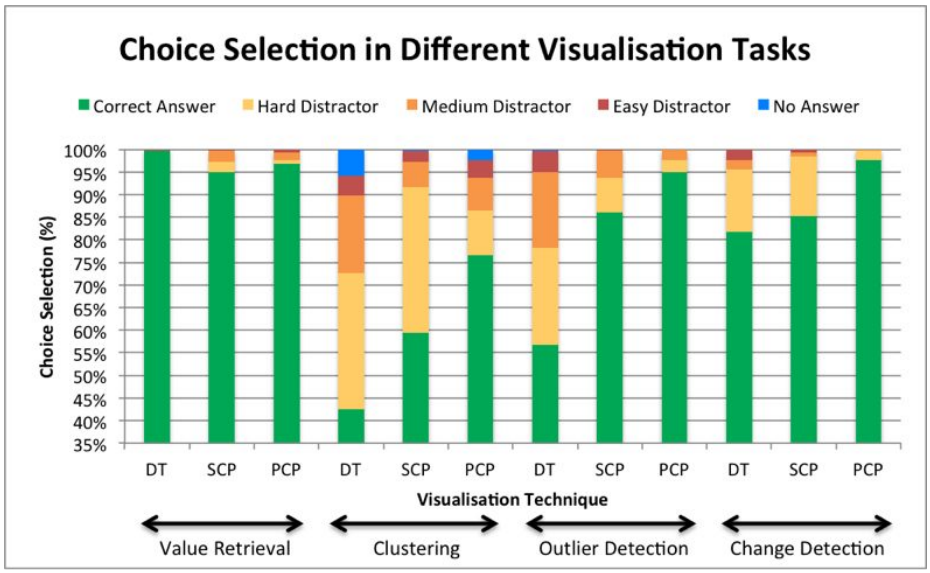


Figure 6.15: Choice selection for each visualisation task.

Value Retrieval Task. The participants select the correct answers in most trials for value retrieval task. More than 90% of the trials in all three visualisation techniques are answered correctly, and only less than 1% of the trials with DT technique are answered incorrectly. In SCP technique, 2% of the selected answers are hard distractors, another 2% are medium distractors, and less than 1% are easy distractors. For trials with PCP, hard distractors, medium distractors, and easy distractors, each occupy about 1% of the selected answers. Overall, most trials in value retrieval task are correctly answered and there is not much difference in choice selection among different visualisation techniques.

Clustering Task. Only approximately 40% of the trials in DT are answered correctly, and the number of correct answers increases for SCP and PCP respectively. Considering the trials with DT technique, 30% of the selected answers are hard distractors, 15% are medium distractors, and about 5% are easy distractors. For SCP, most of the incorrect answers are hard distractors, whereas around 10% of the selected answers are medium and easy distractors altogether. In PCP, hard distractors occupy roughly 10% of the selected answers, while medium and easy distractors each occupy about 5%. In general, hard-distractor choices have a large impact in clustering task, especially for DT and SCP techniques.

Outlier Detection Task. Trials in DT technique have roughly 55% of correct answers, whereas SCP and PCP have above 80% of trials with correct answers. For DT, hard distractors occupy approximately 20% of the selected answers, 15% are medium distractors, and around 5% are easy distractors. In SCP and PCP, about half of the incorrect answers are resulted from hard distractors and another half are from medium distractors. In summary, hard and medium distractors are the two choices that most distract the participants in

outlier detection task. They considerably affect user performance in DT technique, but create just a slightly effect on trials with SCP and PCP.

Change Detection Task. More than 80% of the trials in change detection task for all three visualisation techniques are answered correctly. In DT trials, another 15% of the selected answers are hard distractors, and roughly 5% are medium and easy distractors altogether. Considering the trials in SCP technique, most incorrect answers are hard distractors, and less than 2% of the selected answers are medium and easy distractors altogether. For PCP, approximately 2% of the selected answers are hard distractors, and less than 1% are medium distractors. Generally, most trials in change detection task are answered correctly, but hard distractors create slightly impact on trials in DT and SCP techniques.

Chapter 7

Conclusions

The goal of the project is to evaluate data table (DT), scatter plots (SCP), and parallel coordinates plots (PCP) in different visualisation tasks.

We have achieved our goal and have completely done the project as according to the planned schedule, with most processes starting well ahead of time. The planned and actual schedules are provided in Appendix G. Our completed works include:

1. Study the history and development of SCP and PCP, and gain understanding of how to perform a fair evaluation of these visualisation techniques.

2. Scope the empirical study for the evaluation by identifying hypotheses, variables, measurement metrics, and result analyses' methods for the controlled experiment.

3. Design and implement stimuli and software program for the experiment of four visualisation tasks, value retrieval, clustering, outlier detection, and change detection.

4. Conduct an empirical study using the developed software to collect user performance in the experiments.

5. Perform two statistical analyses including descriptive and inferential statistics to analyse the study's results and provide conclusion for the study.

The project summary, its evaluation, and possible future work will be given in the following subsections.

7.1 Summary

In this project, we conducted an empirical study comparing the user performance, in terms of accuracy and response time, between three visualisation techniques, DT, SCP, and PCP for four visualisation tasks, value retrieval, clustering, outlier detection, and change detection. The results show that DT is the better technique in value retrieval task, while PCP

yields higher performance in clustering, outlier detection, and change detection task. Regarding the clustering and outlier detection task, SCP provides comparable performance to PCP in easy and medium level of task difficulty, but its performance is lower than PCP in the hard-level tasks.

Subjective feedbacks from the participants are also consistent with the quantitative results, indicating that DT is preferred in value retrieval task, while PCP is preferred in change detection task. For clustering and outlier detection task, the effectiveness rating results identify SCP and PCP as the two preferred techniques, which is compatible with the analysed objective results. Both subjective and objective results confirm our four hypotheses that PCP holds advantages in complex tasks, while DT outperforms other techniques in performing basic task.

7.2 Evaluation

To our knowledge, this is the first study that empirically compares DT, SCP, and PCP in clustering, outlier detection, and change detection task, which are the complex and critical tasks in data analyses. For these tasks, our study demonstrates the advantages of PCP over DT and SCP, which had not yet been evaluated before.

The only other study that has found an advantage of PCP over SCP was conducted by Kuang *et al.* [KZZM12]. Nevertheless, their evaluation was performed for value retrieval task, which is a simple task where visualisation does not yield significant higher performance than traditional DT technique. Our research confirms that DT outperforms SCP and PCP in all levels of task difficulty, and thus it is more suitable for value retrieval task. This conclusion raises a question about the importance of the research paper of Kuang *et al.* [KZZM12].

The only limitation in our study is that we did not evaluate the three visualisation techniques for high dimensionality and high density data sets. Hence, we cannot compare the results with Kuang *et al.* [KZZM12] for this case.

With the current results, however, we believe that our study will be valuable and useful for visual analysts in order to select appropriate visualisation techniques for each visualisation task, and for other researchers to understand and further examine these advantages based on our findings.

7.3 Future Work

Many possible studies can be extended from our research.

Firstly, future studies can include other data types including nominal, ordinal, and interval data, or increase dimension and density of the data sets.

Secondly, other variants of SCP and PCP, including the hybrid techniques, can be further investigated and evaluated.

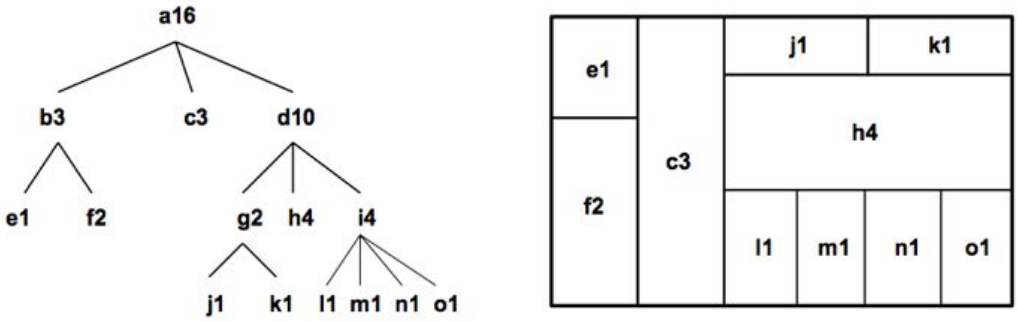
Thirdly, the performance of other visualisation techniques, for instance treemaps and star glyphs, can be compared in the future works.

Lastly, other visualisation tasks, such as classification tree generation, are extensible to our study.

In conclusion, the researches in information visualisation area still have many evaluations to be performed. Yet, we believe that our study, as a part of these evaluations, will be more or less beneficial in this field.

Appendix A

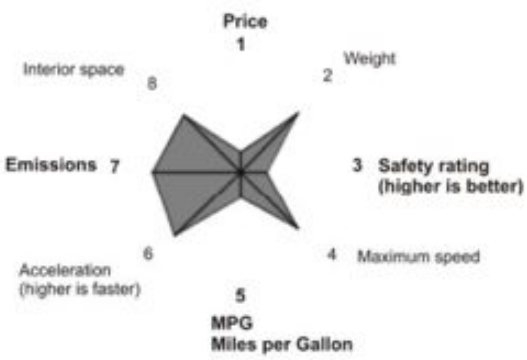
Different Visualisation Techniques



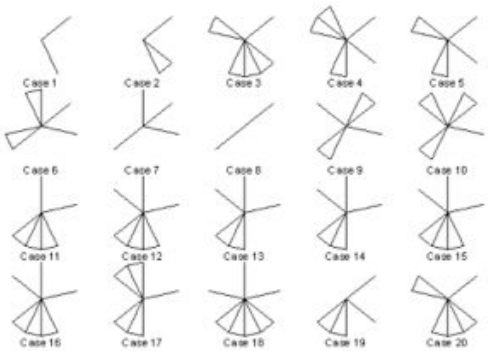
(a) Tree diagram.

(b) Treemap.

Figure A.1: An example of a treemap, constructed from a tree diagram. [BHVW00].

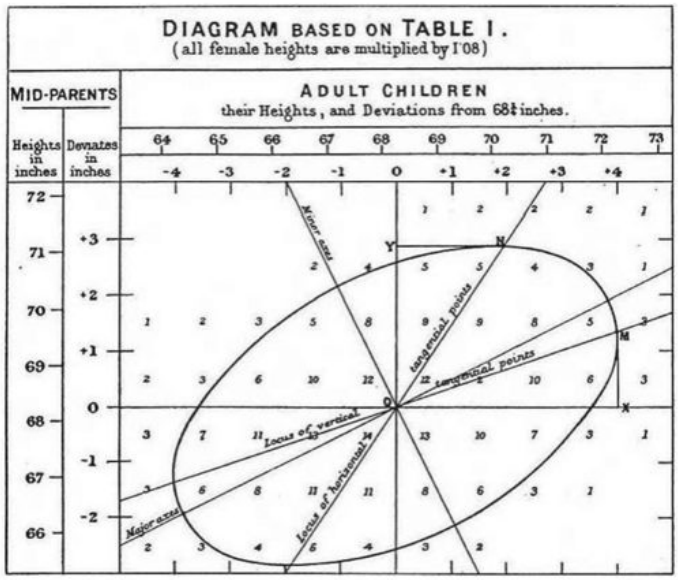


(a) A star glyph with 8 variables. [KHLW09].

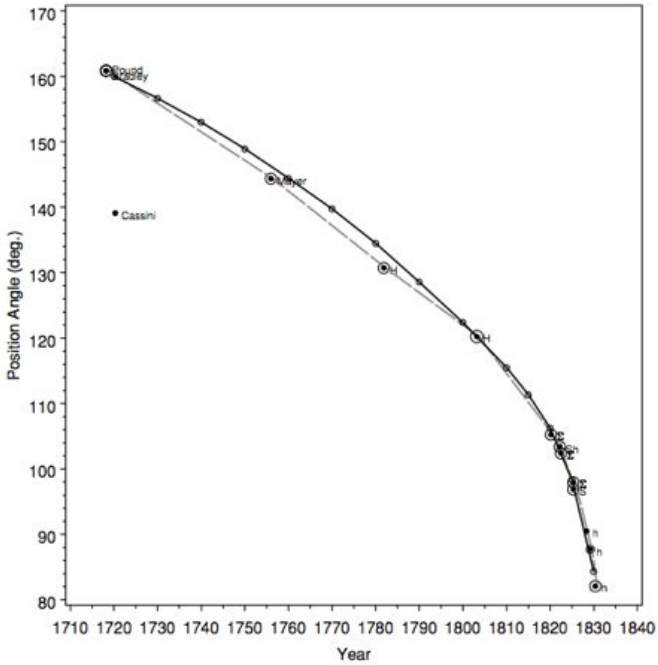


(b) The star glyphs of a data set. [LRB03].

Figure A.2: Examples of star glyph visualisation.

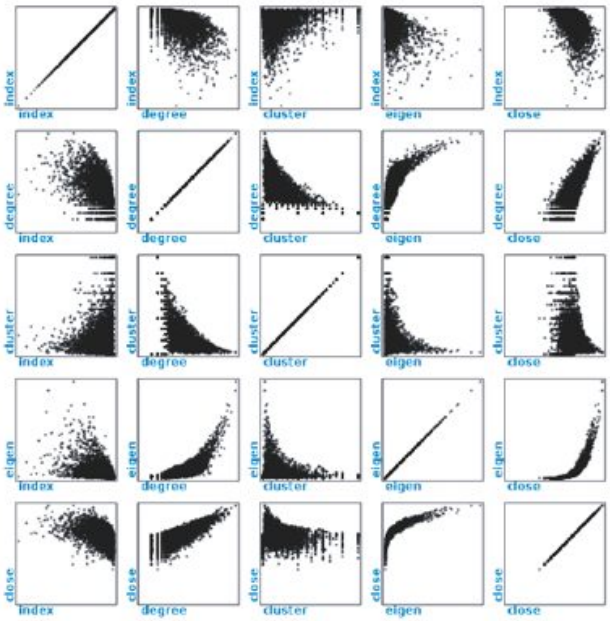


(a) Galton's semi-graphic scatter plot. [Gal86].

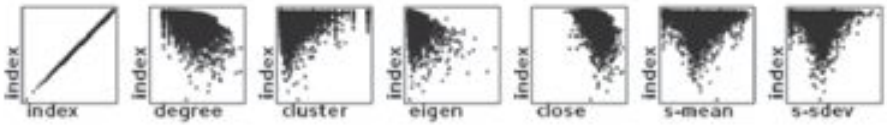


(b) Herschel's scatter plot. [FD05].

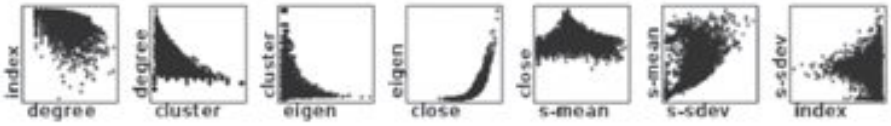
Figure A.3: Development of scatter plots.



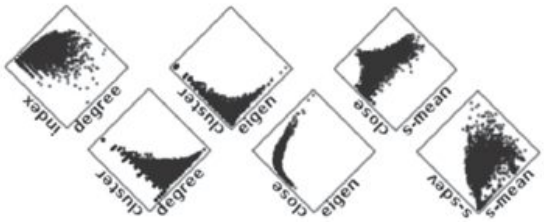
(a) Scatter plot matrix.



(b) A row of scatter plot cells with the same vertical axis.

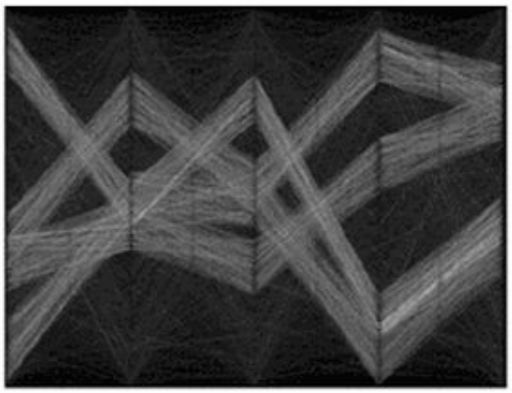


(c) Each cell's horizontal axis is the same as consecutive cell's vertical axis.

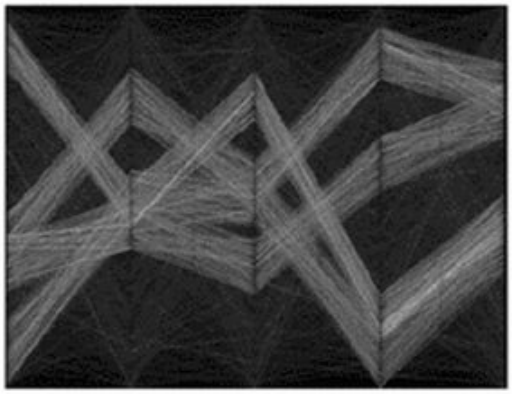


(d) Adjacent scatter plots share an axis along their common edge.

Figure A.4: Variants of scatter plots. [VMCJ10].



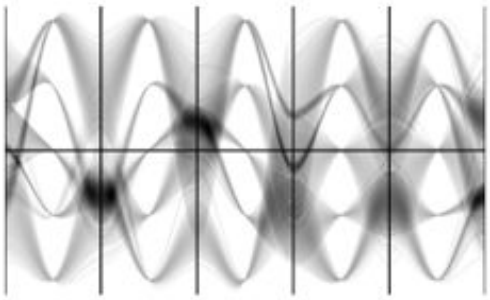
(a) Parallel coordinates frequency plots. [AdOL04].



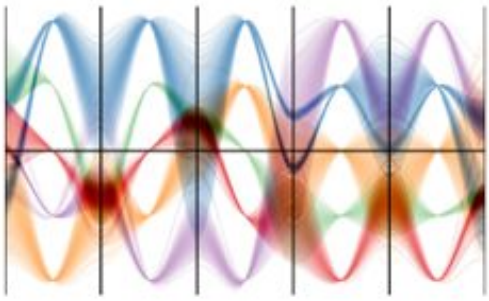
(b) Parallel coordinates density plots. [AdOL04].

(c) All clusters visualised using a linear TF. [JLJC06].

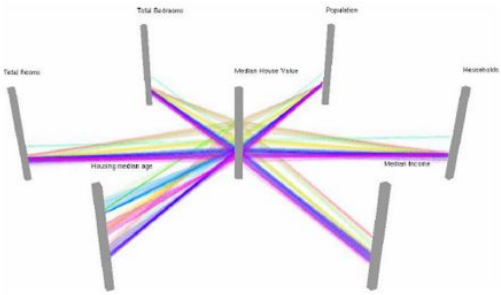
(d) Two similar clusters are selected and applied with a square root TF. [JLJC06].



(e) The use of smooth curves instead of polylines. [LWZK08].



(f) The use of smooth curves with colour coding. [LWZK08].



(g) An example of 3D parallel coordinates plots. [JCJ05].

Figure A.5: Variants of parallel coordinates plots.

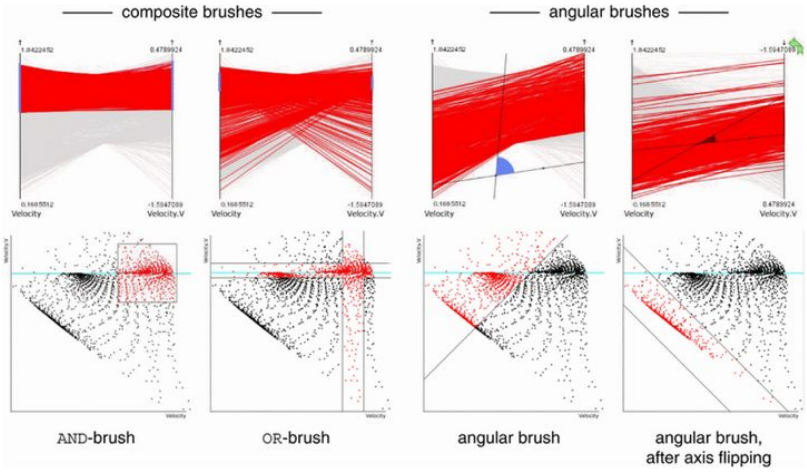
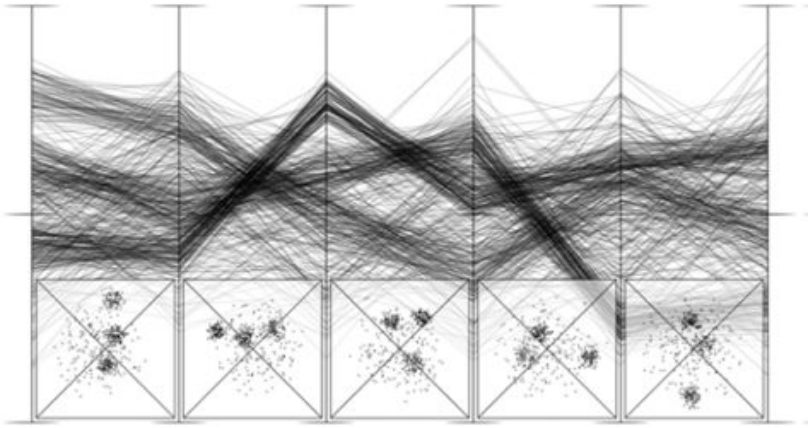
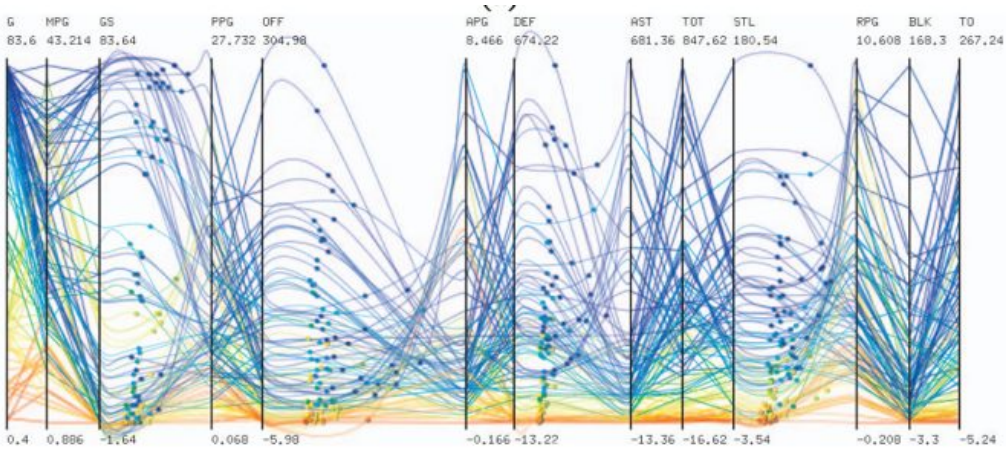


Figure A.6: Compare composite brushes and angular brushes using parallel coordinates plots and scatter plots. [HLD02].



(a) Scatter plots embedded between each pair of adjacent parallel coordinates plots' axes. [HvW10].



(b) Scattering Points in Parallel Coordinates Plots. [YGX+09]

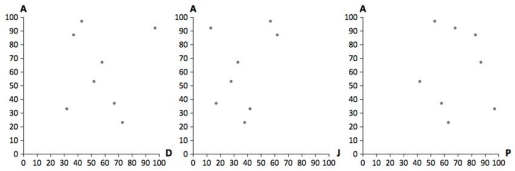
Figure A.7: Hybrid visualisation techniques of scatter plots and parallel coordinates plots.

Appendix B

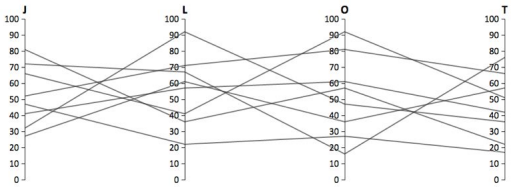
Different Versions of Stimuli

A	C	E	N
32	37	56	71
66	12	66	76
87	77	76	92
57	56	62	56
81	21	87	52
92	32	71	26
22	71	26	36
36	86	21	42

(a) Data Table.



(b) Scatter Plots.



(c) Parallel Coordinates Plots.

Figure B.1: Stimuli that have been modified to ensure the same formats for the three visualisation techniques.

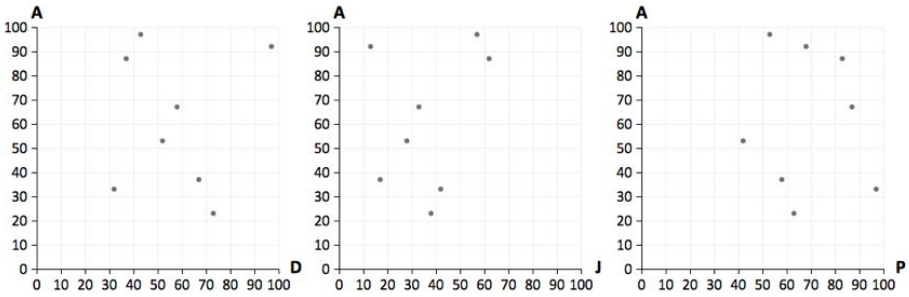
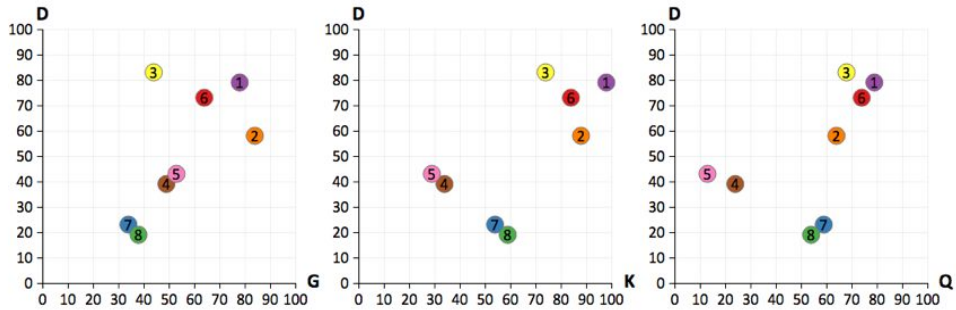


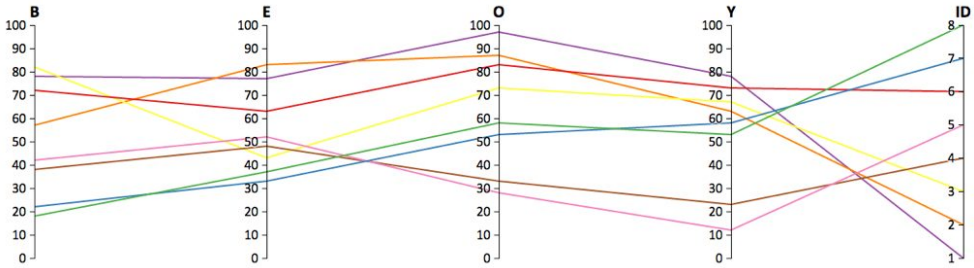
Figure B.2: Scatter plots with gridlines.

H	I	R	Z	ID
77	76	96	77	1
56	82	86	62	2
81	42	72	66	3
37	47	32	22	4
41	51	27	11	5
71	62	82	72	6
21	32	52	57	7
17	36	57	52	8

(a) Data Table.

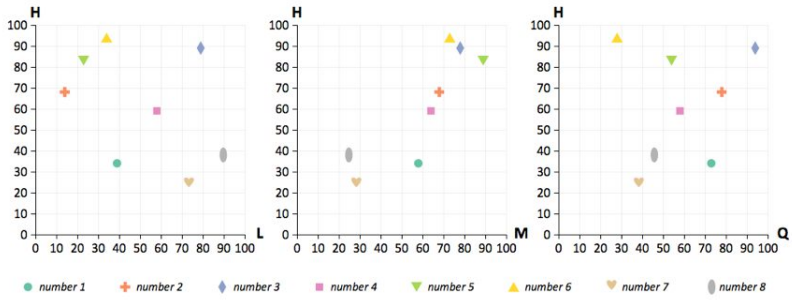


(b) Scatter Plots.

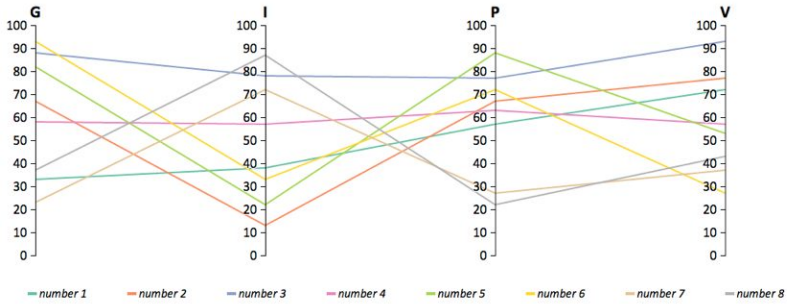


(c) Parallel Coordinates Plots.

Figure B.3: Stimuli that label data points using colour coding.

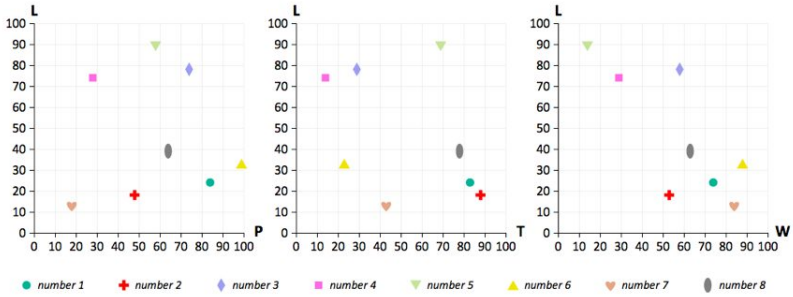


(a) Scatter Plots with Different Shapes.

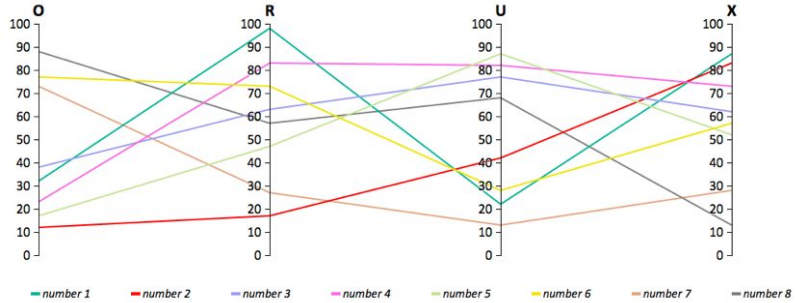


(b) Parallel Coordinates Plots.

Figure B.4: Stimuli that provide legends to ensure consistency between the two visualisation techniques.



(a) Scatter Plots.



(b) Parallel Coordinates Plots.

Figure B.5: Stimuli that have been modified to maximise colours' distance difference.

Appendix C

Stimuli in the Experiment

Our experiment involves four visualisation tasks, value retrieval, clustering, outlier detection, and change detection. Each task has 6 triples of stimuli; one triple consists of stimuli for data table, scatter plots, and parallel coordinates plots. The 6 triples are further divided into 3 groups, which have three levels of task difficulty: easy, medium, and hard. The structure of the stimuli in each task is given in figure C.1 and all triple of stimuli in our experiment are given in the following figures.

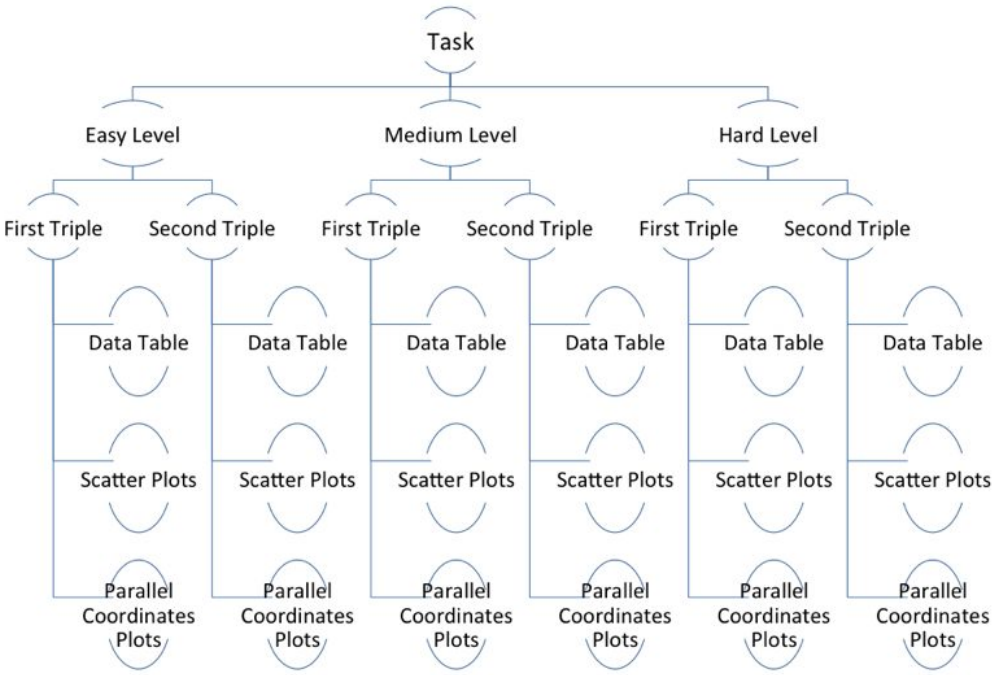
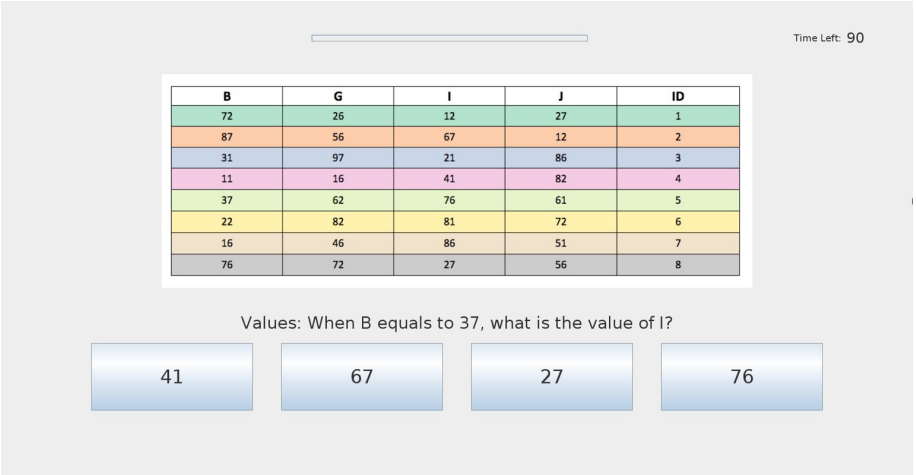
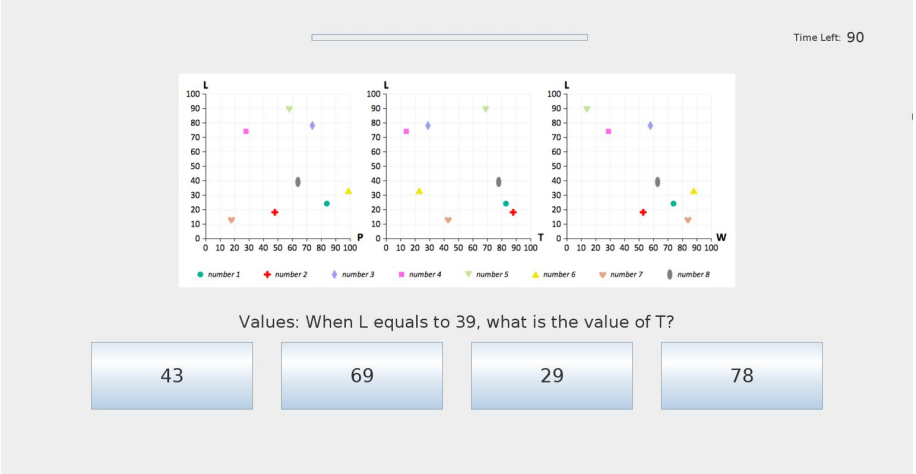


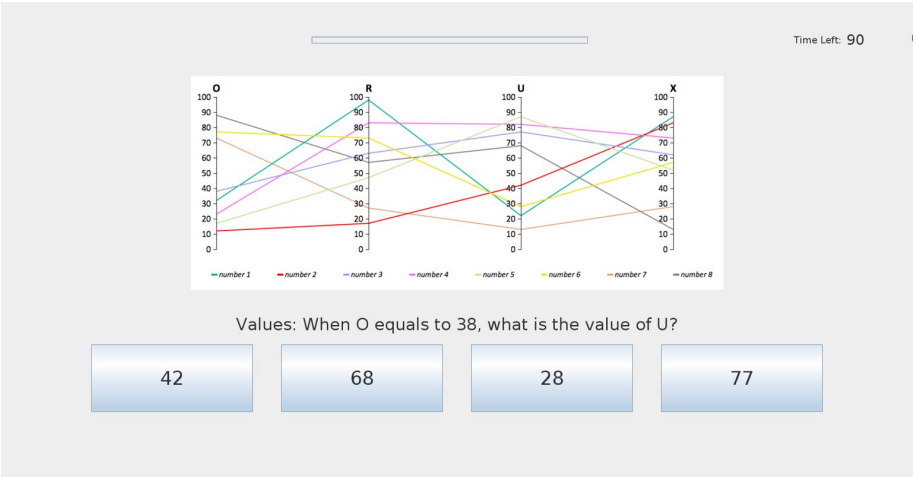
Figure C.1: Stimuli structure for each task.



(a) Data Table.

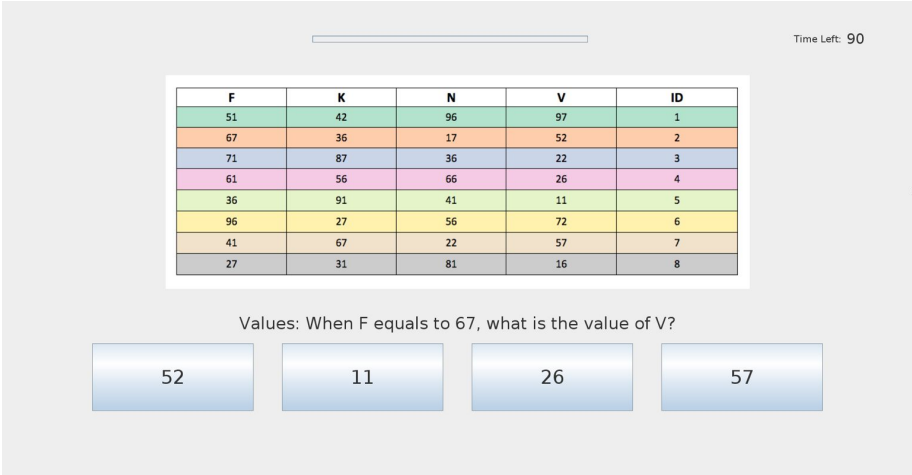


(b) Scatter Plots.

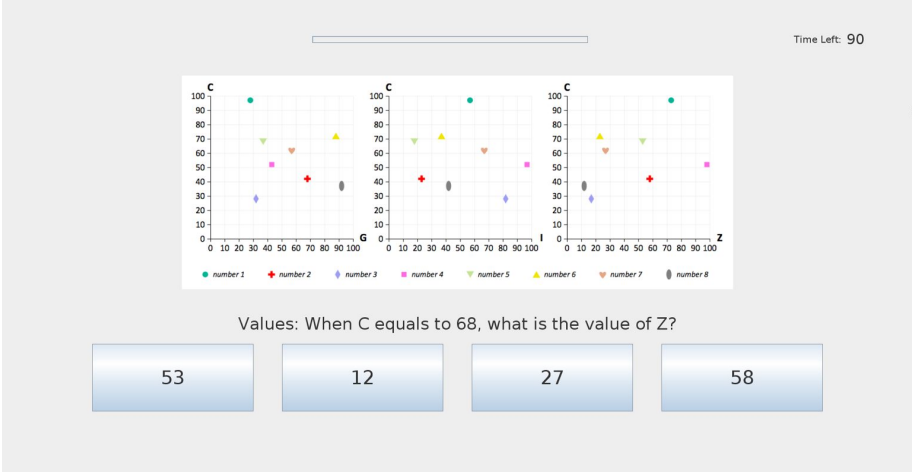


(c) Parallel Coordinates Plots.

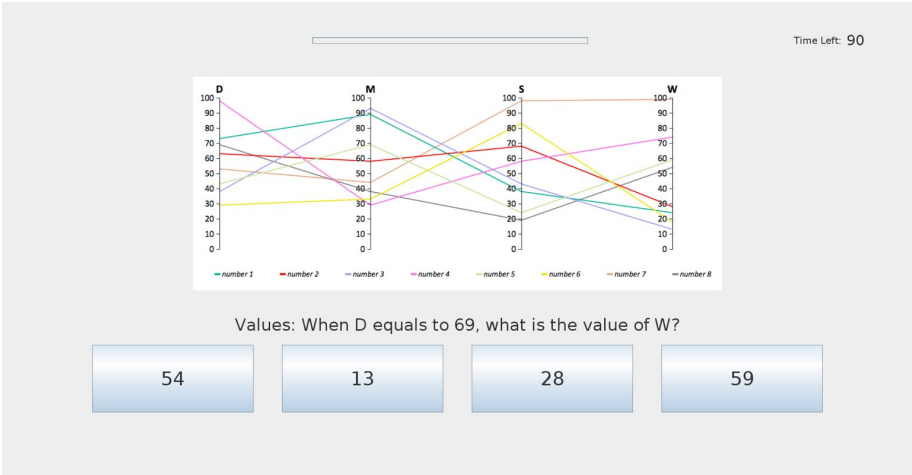
Figure C.2: First triple of stimuli for easy-level value retrieval task.



(a) Data Table.

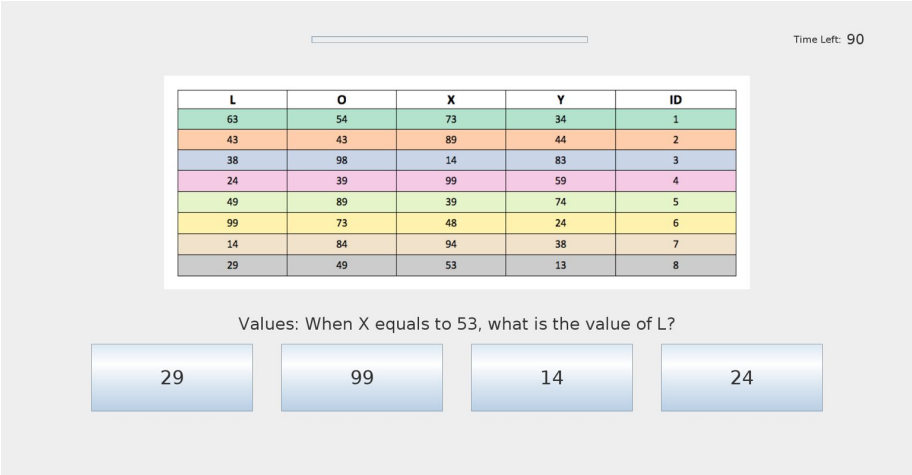


(b) Scatter Plots.

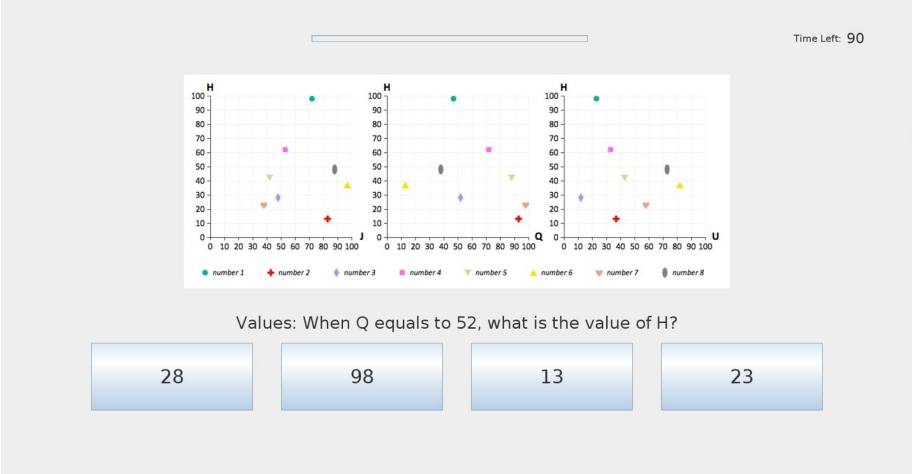


(c) Parallel Coordinates Plots.

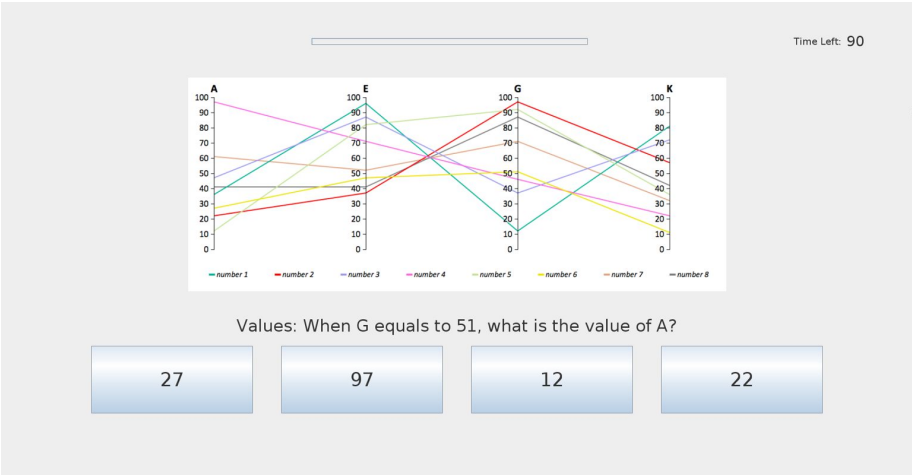
Figure C.3: Second triple of stimuli for easy-level value retrieval task.



(a) Data Table.

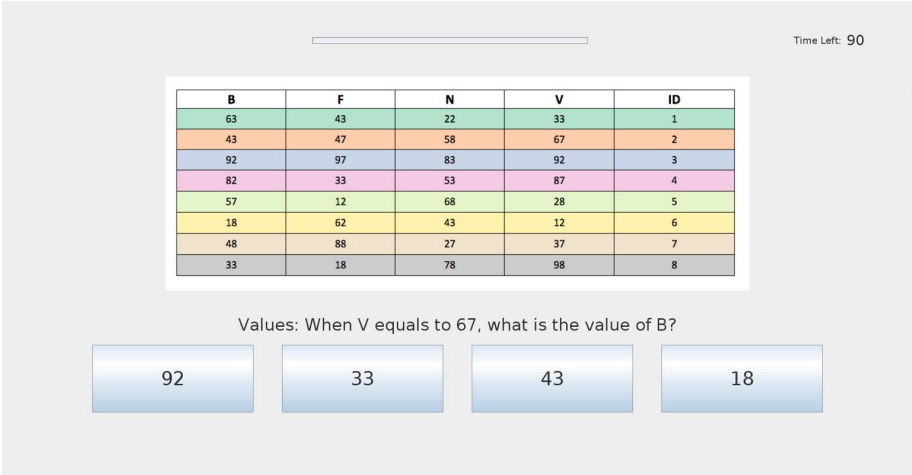


(b) Scatter Plots.



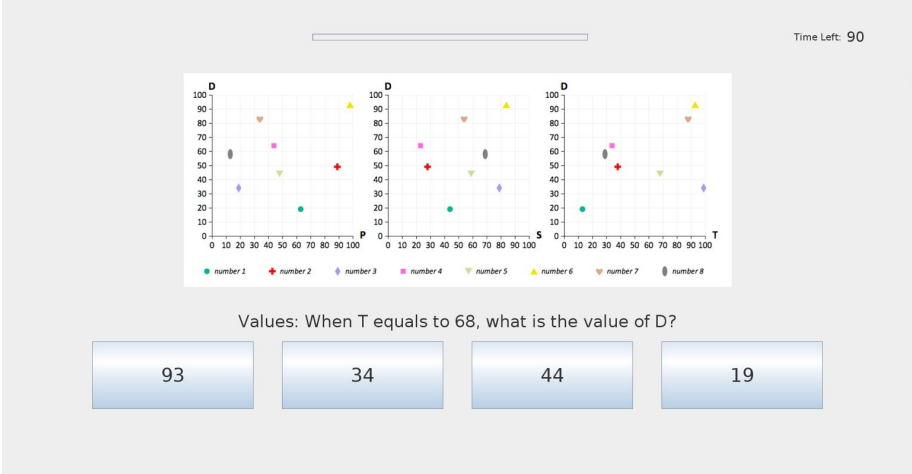
(c) Parallel Coordinates Plots.

Figure C.4: First triple of stimuli for medium-level value retrieval task.

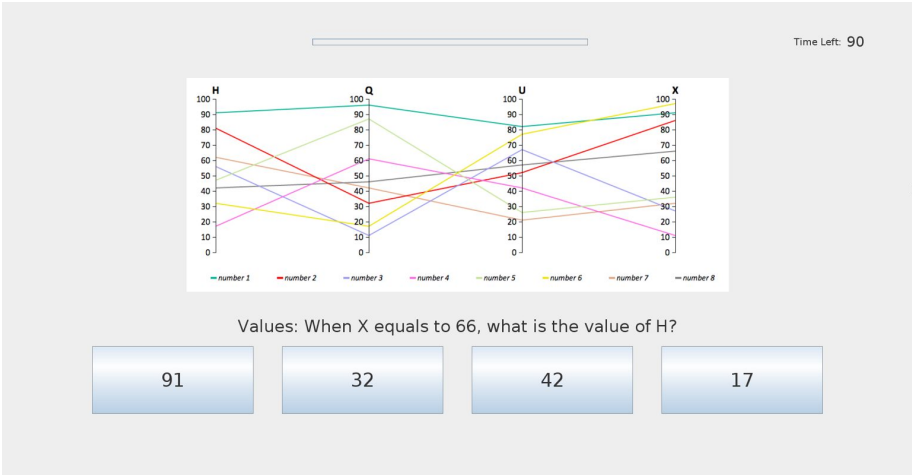


B	F	N	V	ID
63	43	22	33	1
43	47	58	67	2
92	97	83	92	3
82	33	53	87	4
57	12	68	28	5
18	62	43	12	6
48	88	27	37	7
33	18	78	98	8

(a) Data Table.

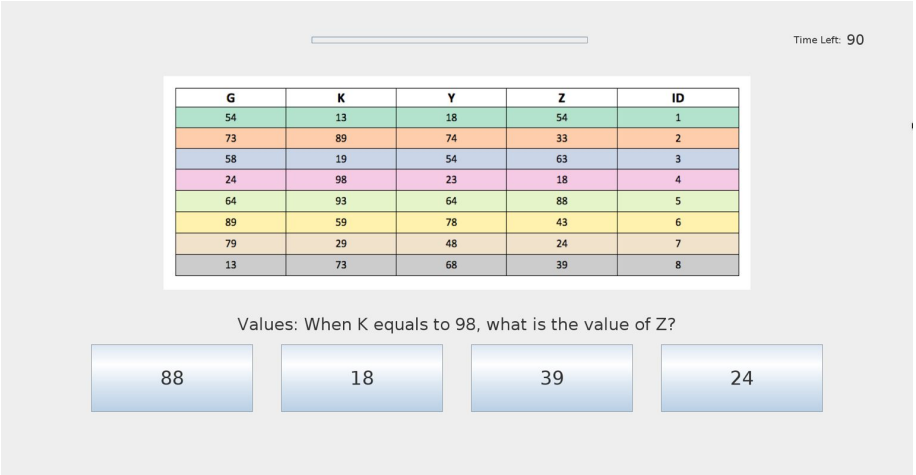


(b) Scatter Plots.

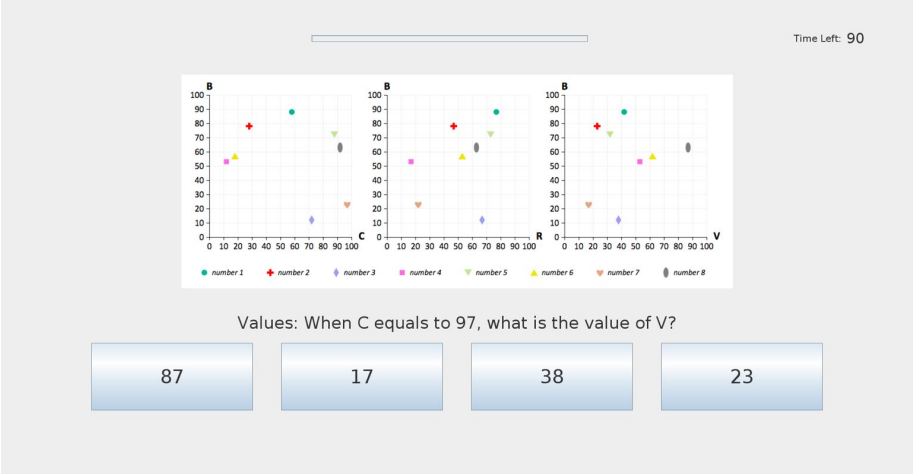


(c) Parallel Coordinates Plots.

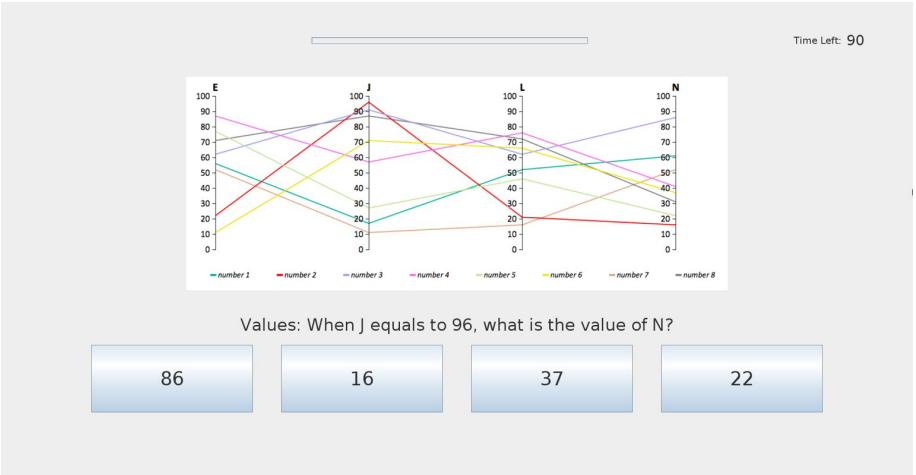
Figure C.5: Second triple of stimuli for medium-level value retrieval task.



(a) Data Table.

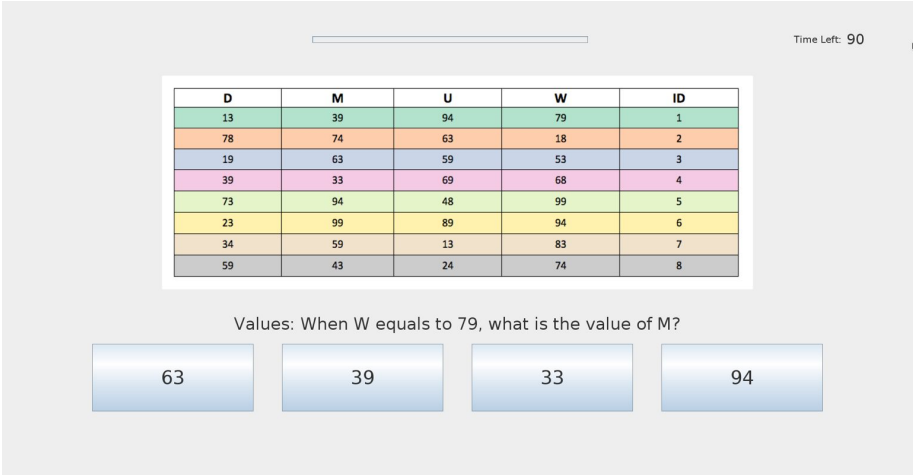


(b) Scatter Plots.

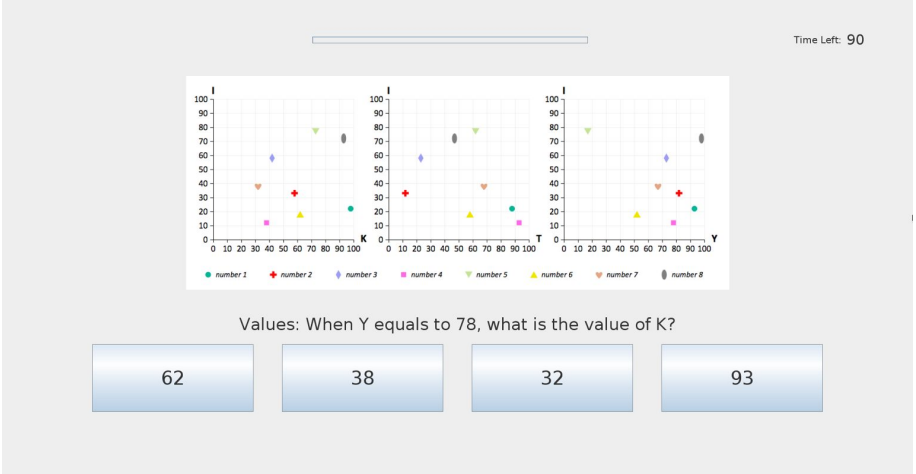


(c) Parallel Coordinates Plots.

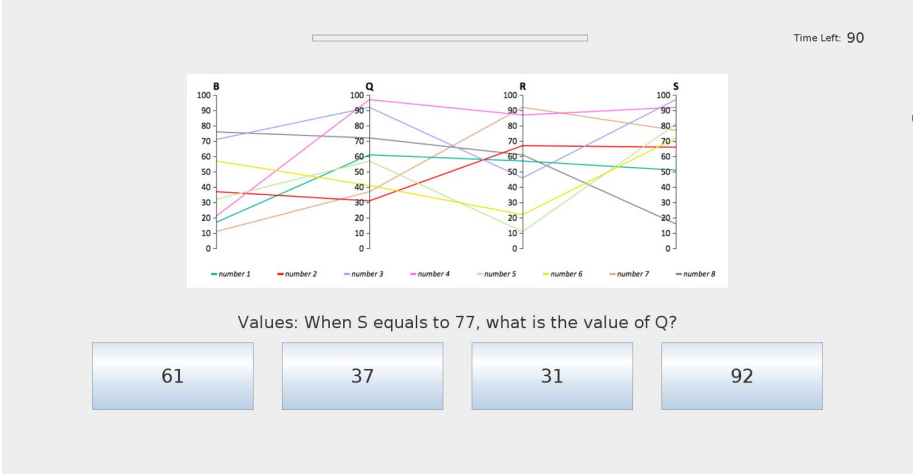
Figure C.6: First triple of stimuli for hard-level value retrieval task.



(a) Data Table.

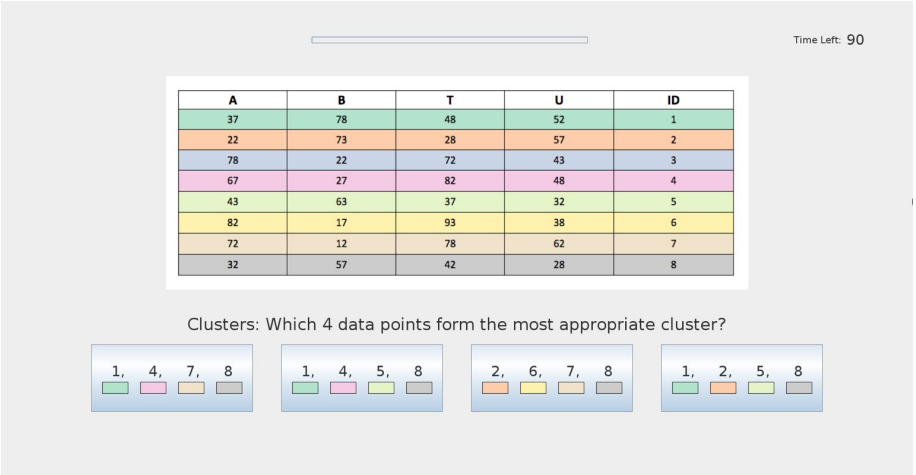


(b) Scatter Plots.



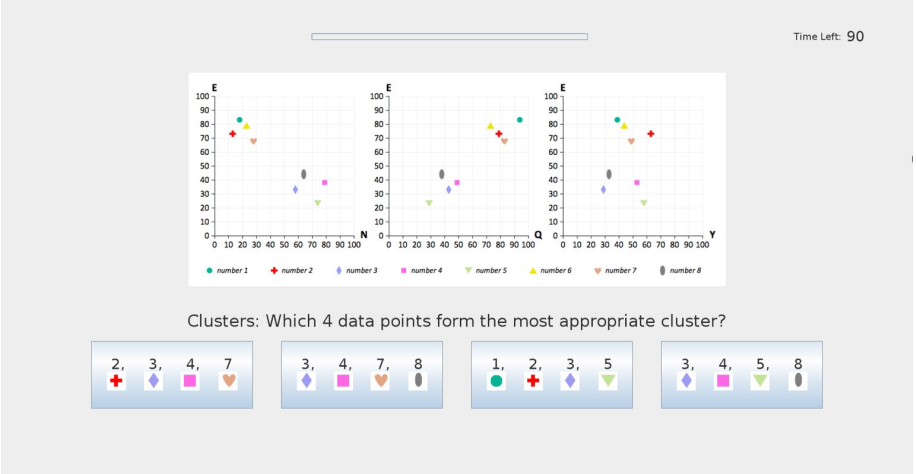
(c) Parallel Coordinates Plots.

Figure C.7: Second triple of stimuli for hard-level value retrieval task.

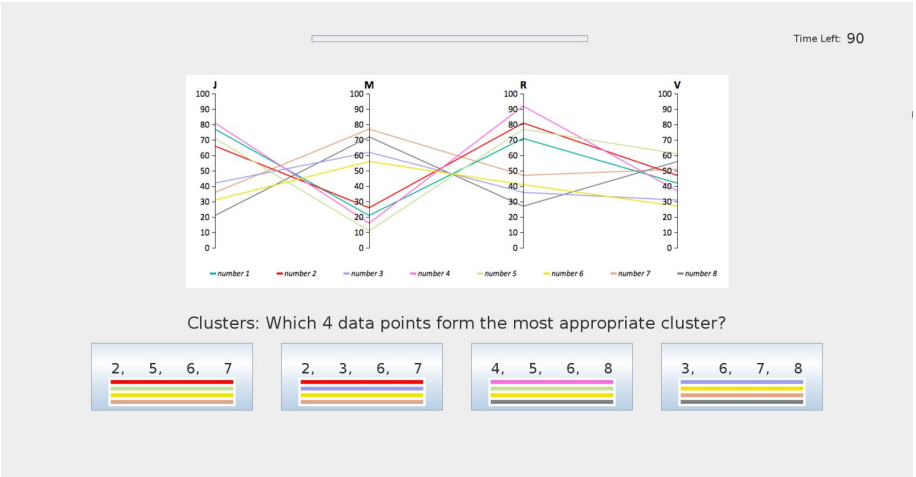


A	B	T	U	ID
37	78	48	52	1
22	73	28	57	2
78	22	72	43	3
67	27	82	48	4
43	63	37	32	5
82	17	93	38	6
72	12	78	62	7
32	57	42	28	8

(a) Data Table.

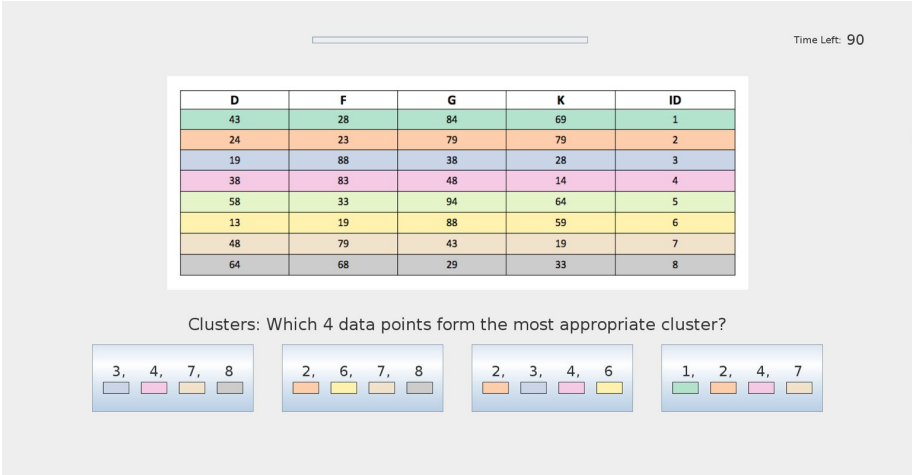


(b) Scatter Plots.

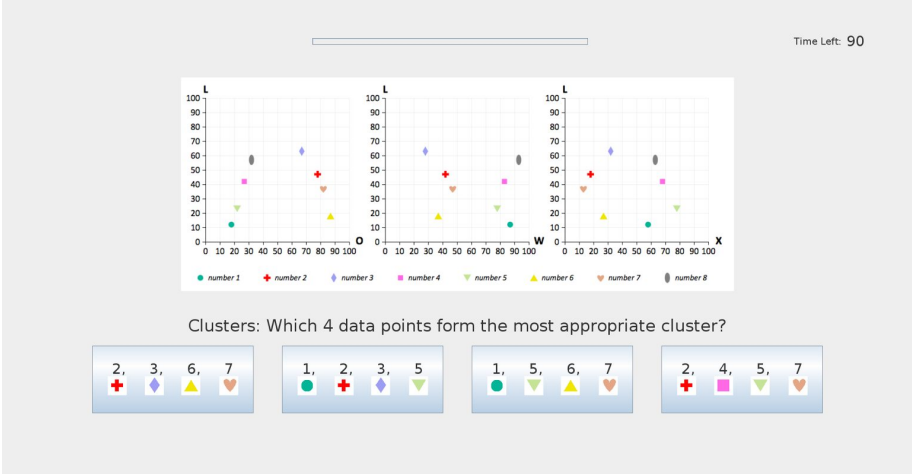


(c) Parallel Coordinates Plots.

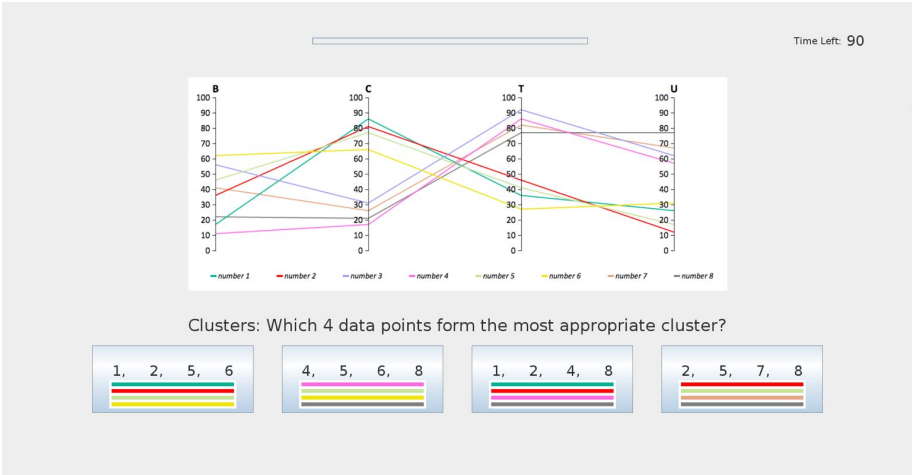
Figure C.8: First triple of stimuli for easy-level clustering task.



(a) Data Table.

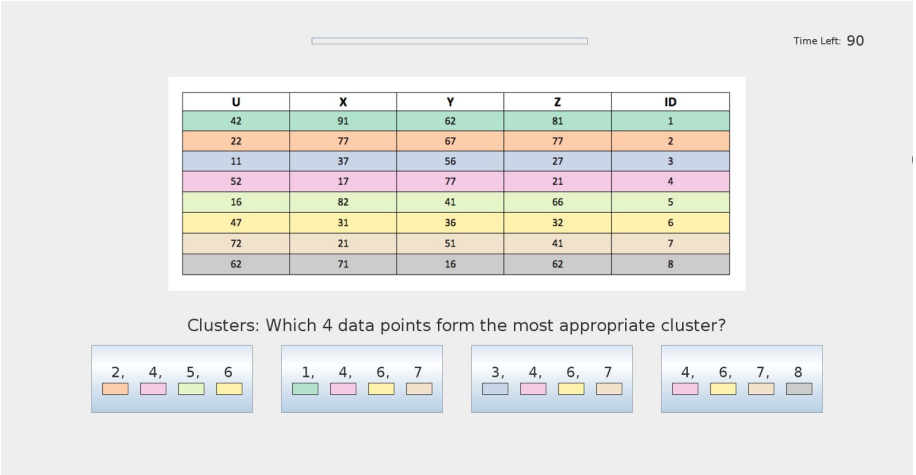


(b) Scatter Plots.



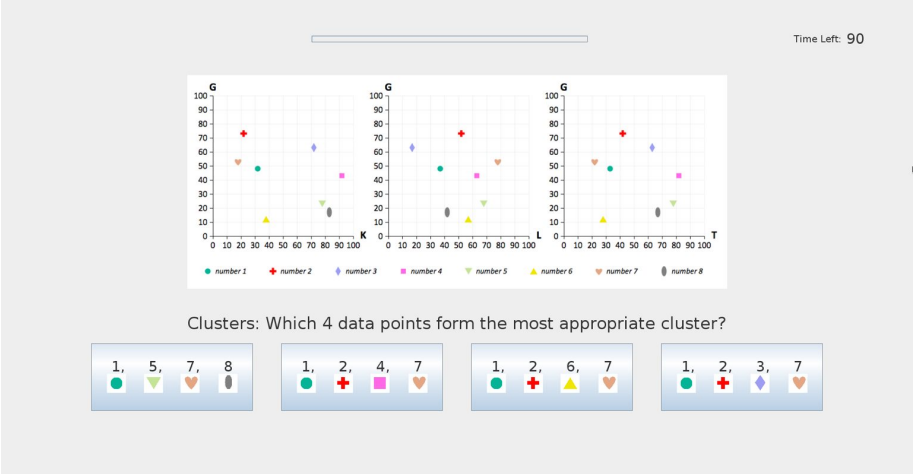
(c) Parallel Coordinates Plots.

Figure C.9: Second triple of stimuli for easy-level clustering task.

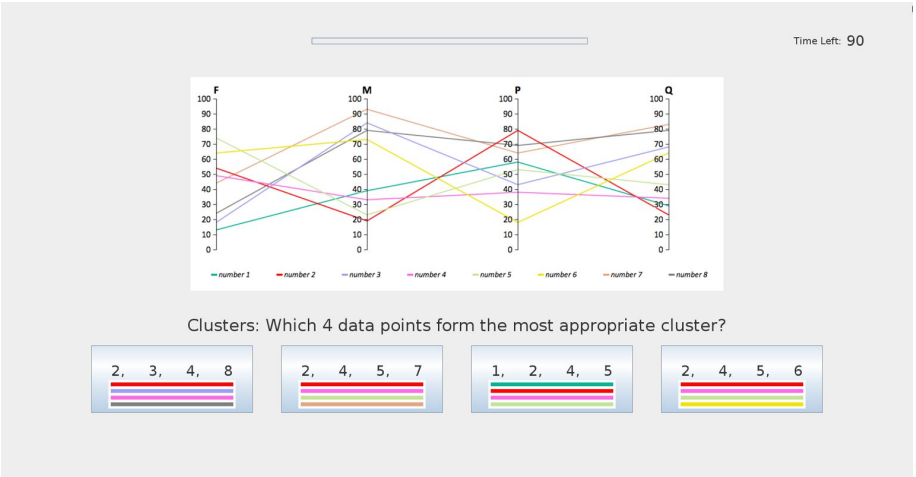


U	X	Y	Z	ID
42	91	62	81	1
22	77	67	77	2
11	37	56	27	3
52	17	77	21	4
16	82	41	66	5
47	31	36	32	6
72	21	51	41	7
62	71	16	62	8

(a) Data Table.

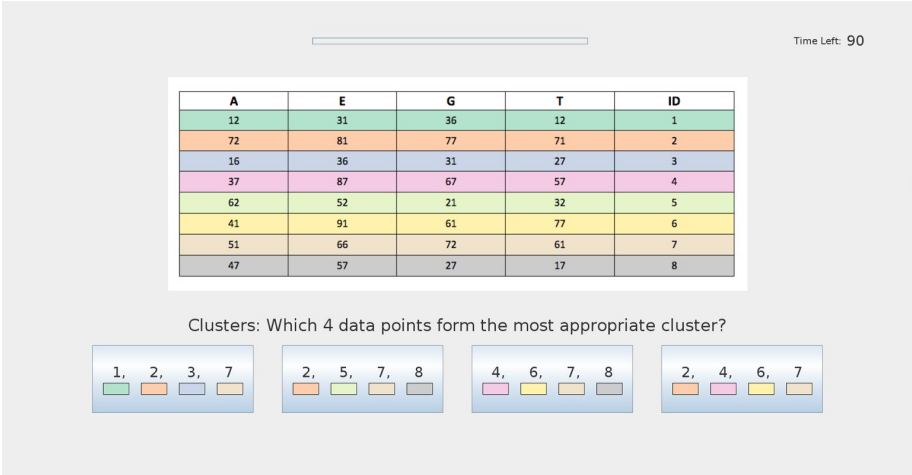


(b) Scatter Plots.

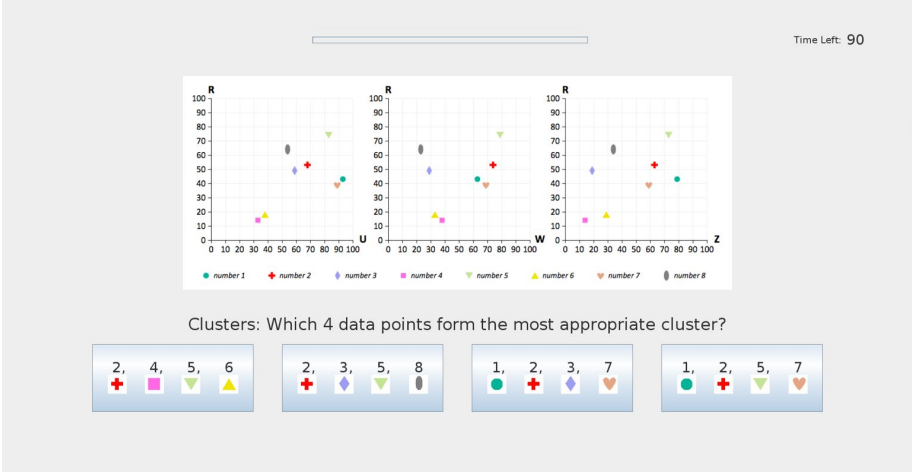


(c) Parallel Coordinates Plots.

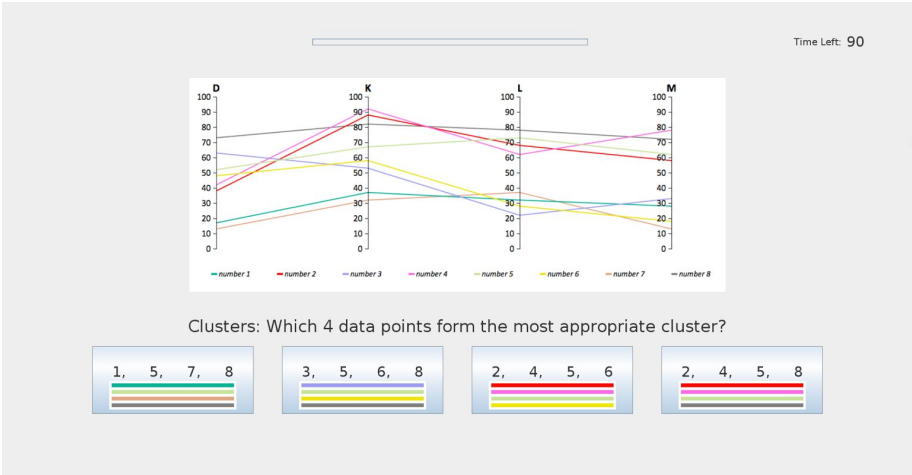
Figure C.10: First triple of stimuli for medium-level clustering task.



(a) Data Table.

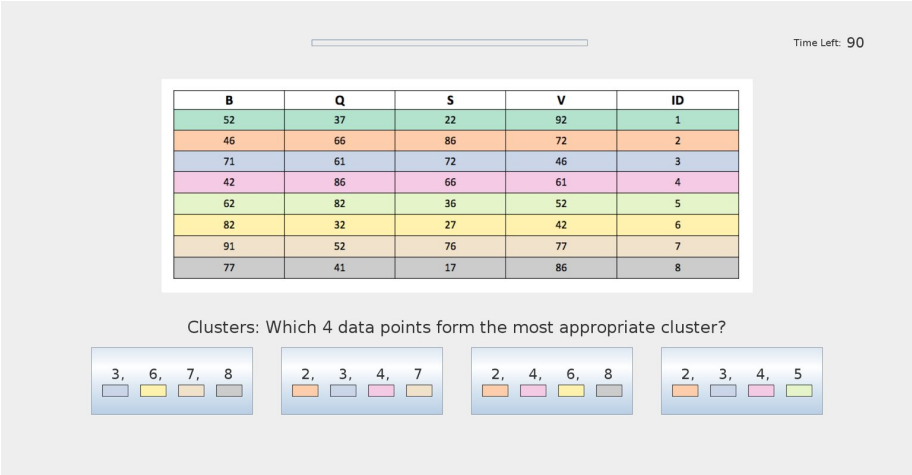


(b) Scatter Plots.



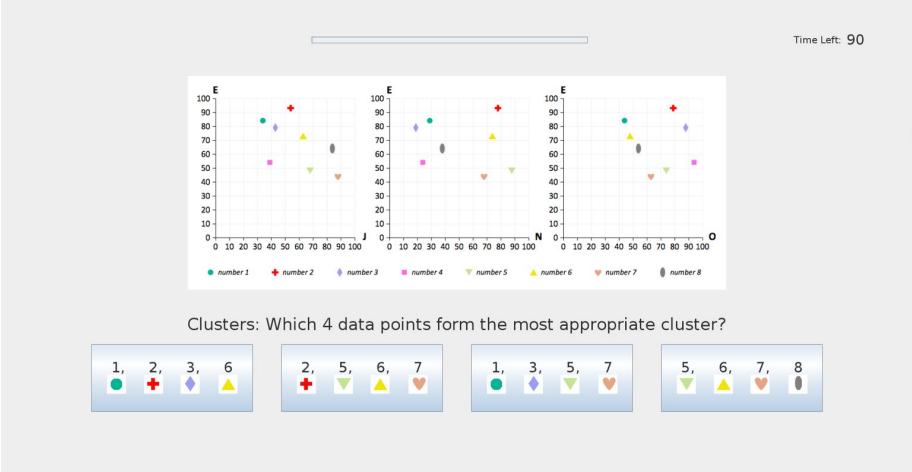
(c) Parallel Coordinates Plots.

Figure C.11: Second triple of stimuli for medium-level clustering task.

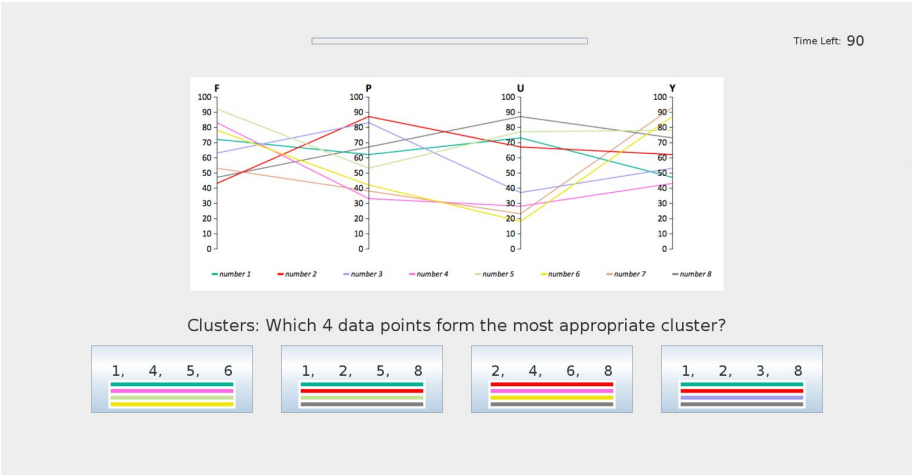


B	Q	S	V	ID
52	37	22	92	1
46	66	86	72	2
71	61	72	46	3
42	86	66	61	4
62	82	36	52	5
82	32	27	42	6
91	52	76	77	7
77	41	17	86	8

(a) Data Table.

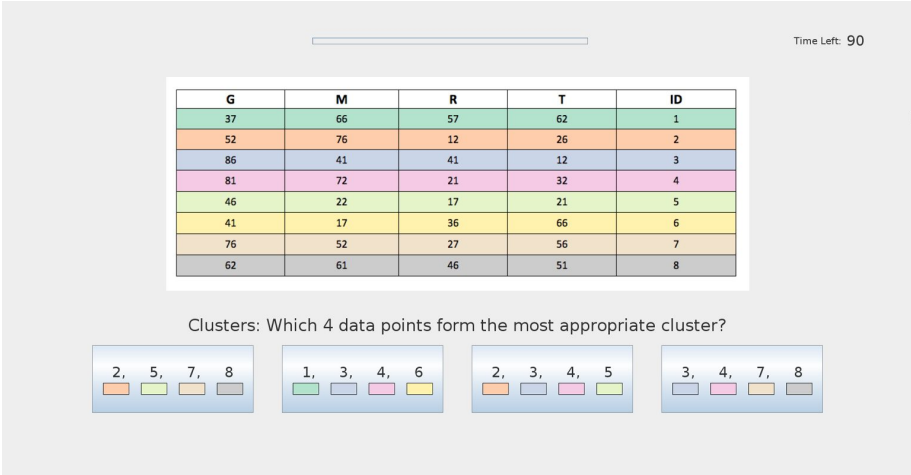


(b) Scatter Plots.

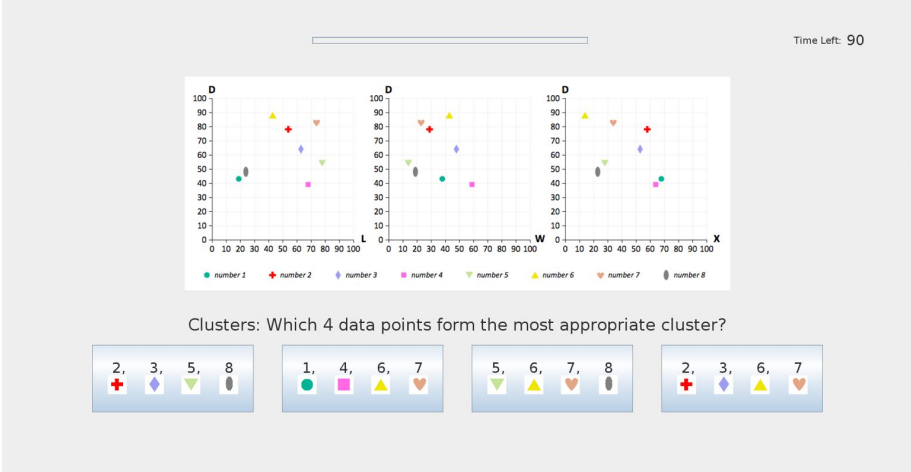


(c) Parallel Coordinates Plots.

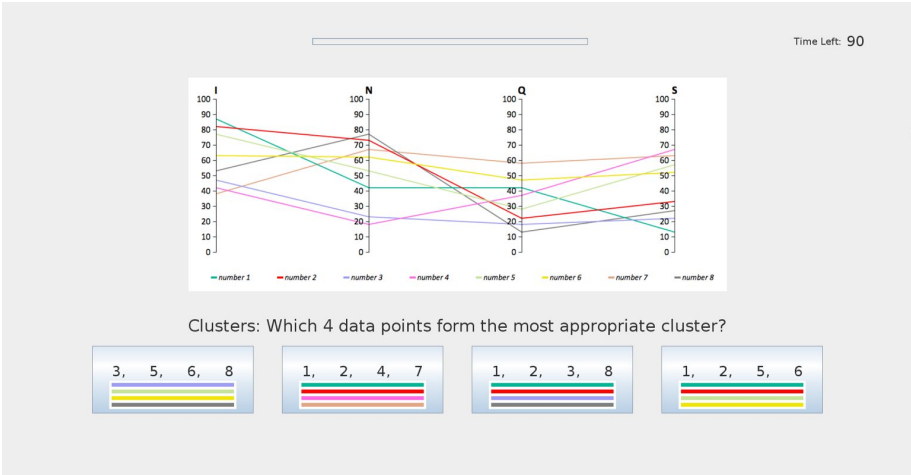
Figure C.12: First triple of stimuli for hard-level clustering task.



(a) Data Table.

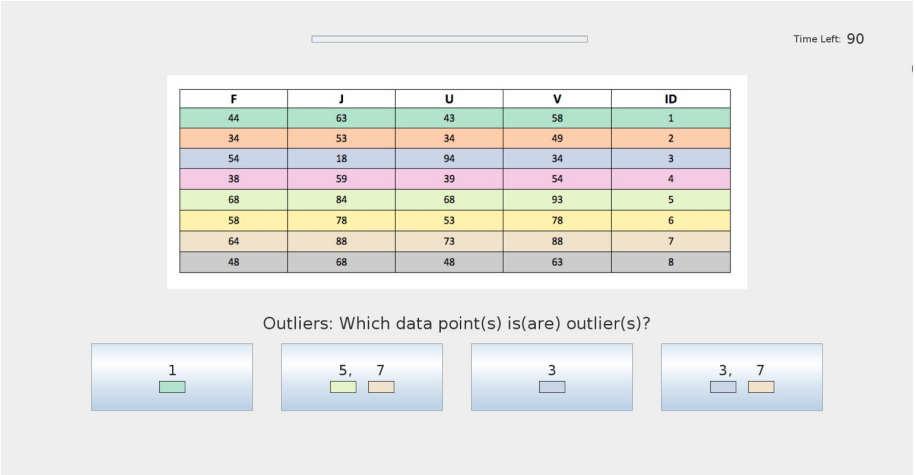


(b) Scatter Plots.



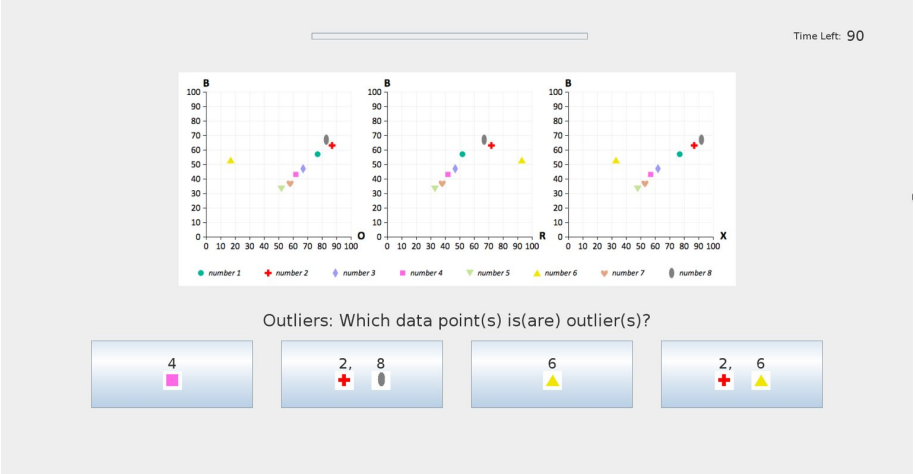
(c) Parallel Coordinates Plots.

Figure C.13: Second triple of stimuli for hard-level clustering task.

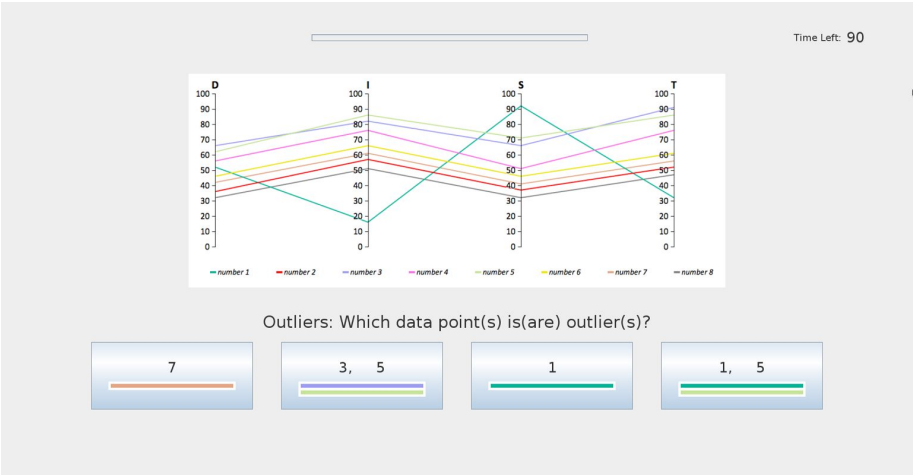


F	J	U	V	ID
44	63	43	58	1
34	53	34	49	2
54	18	94	34	3
38	59	39	54	4
68	84	68	93	5
58	78	53	78	6
64	88	73	88	7
48	68	48	63	8

(a) Data Table.

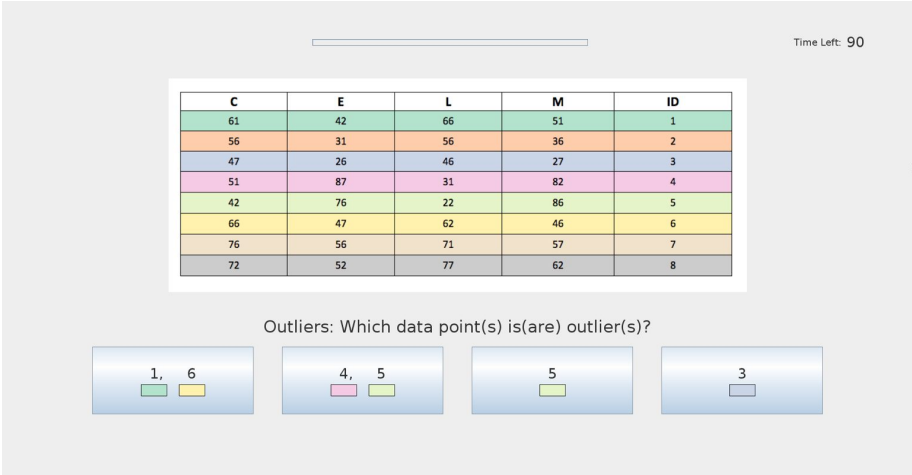


(b) Scatter Plots.



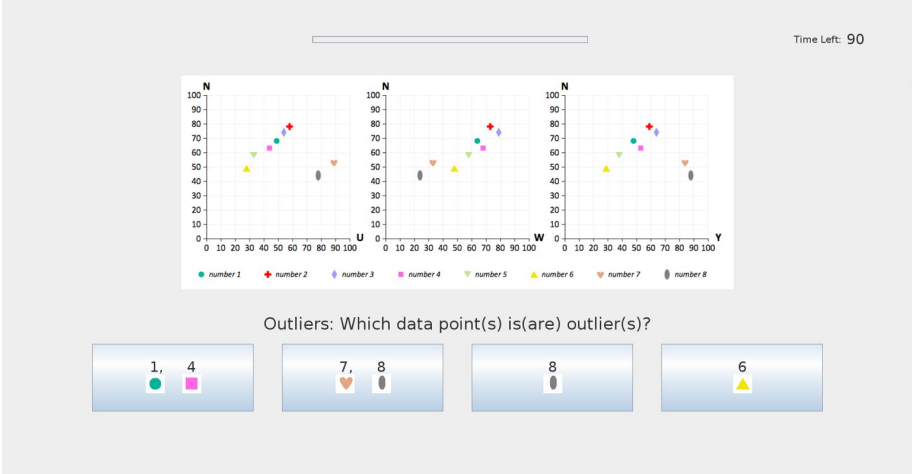
(c) Parallel Coordinates Plots.

Figure C.14: First triple of stimuli for easy-level outlier detection task.

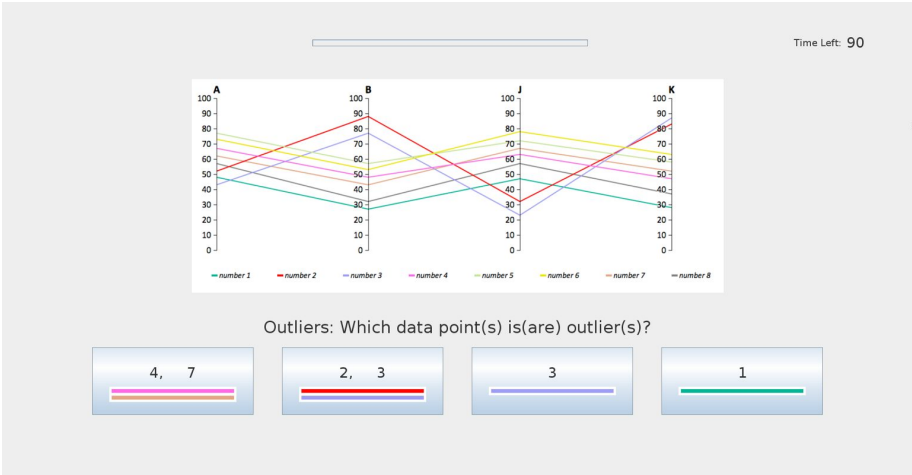


C	E	L	M	ID
61	42	66	51	1
56	31	56	36	2
47	26	46	27	3
51	87	31	82	4
42	76	22	86	5
66	47	62	46	6
76	56	71	57	7
72	52	77	62	8

(a) Data Table.

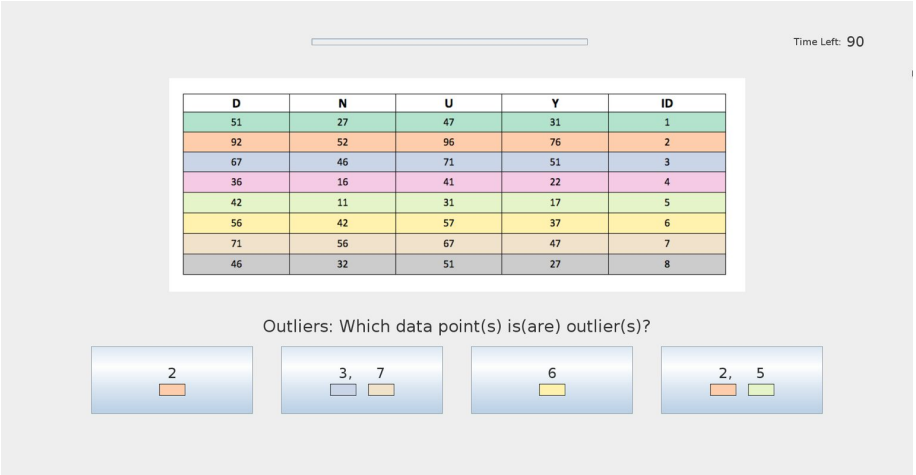


(b) Scatter Plots.

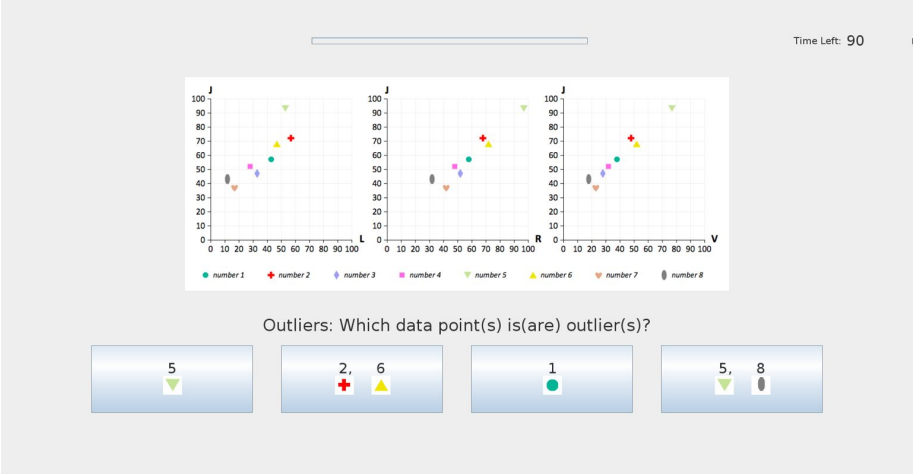


(c) Parallel Coordinates Plots.

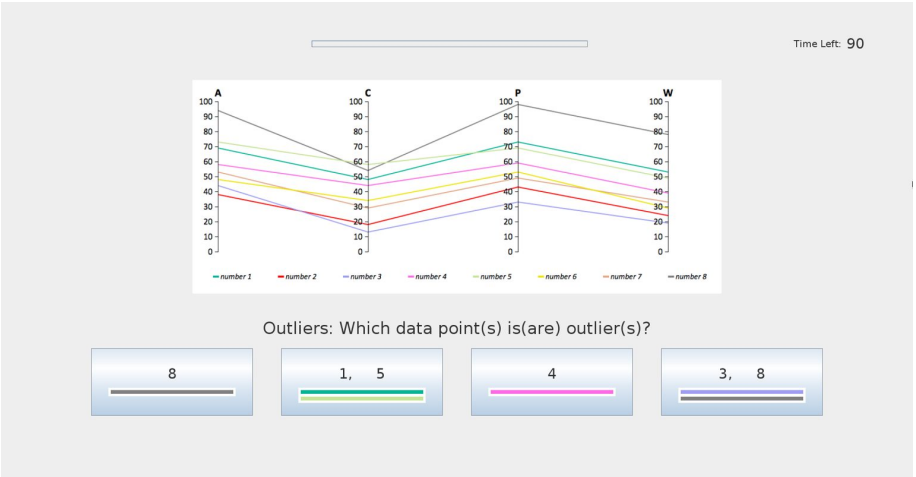
Figure C.15: Second triple of stimuli for easy-level outlier detection task.



(a) Data Table.

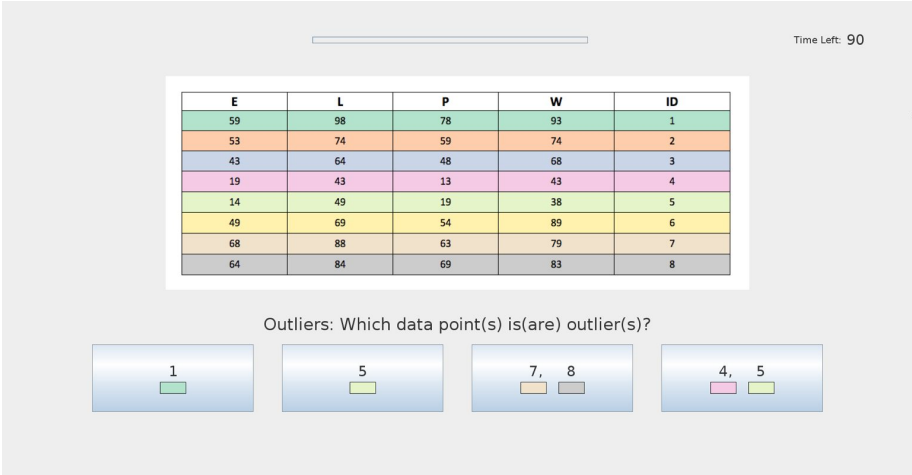


(b) Scatter Plots.



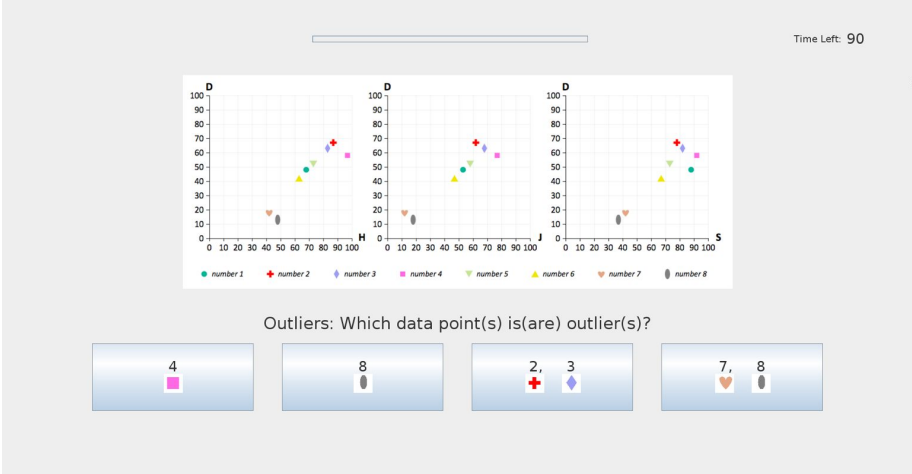
(c) Parallel Coordinates Plots.

Figure C.16: First triple of stimuli for medium-level outlier detection task.

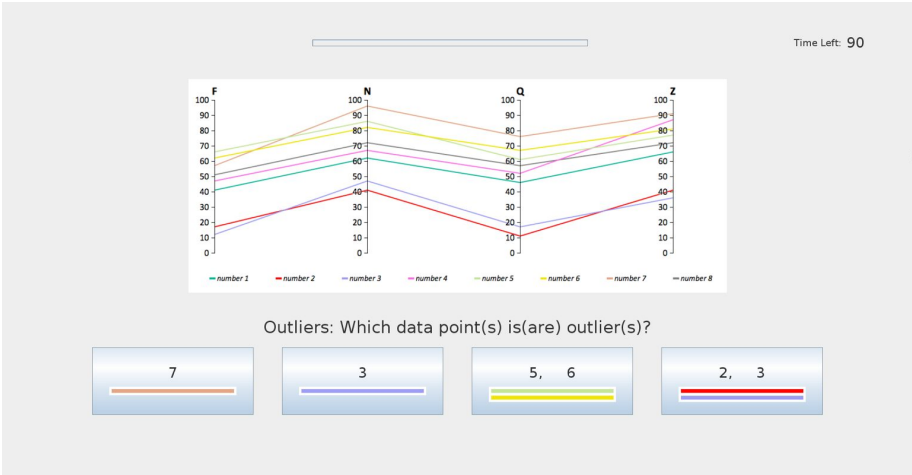


E	L	P	W	ID
59	98	78	93	1
53	74	59	74	2
43	64	48	68	3
19	43	13	43	4
14	49	19	38	5
49	69	54	89	6
68	88	63	79	7
64	84	69	83	8

(a) Data Table.

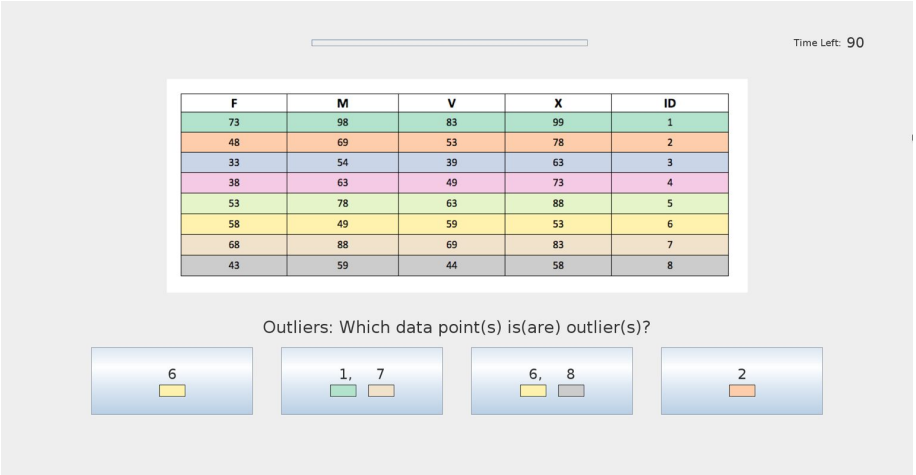


(b) Scatter Plots.



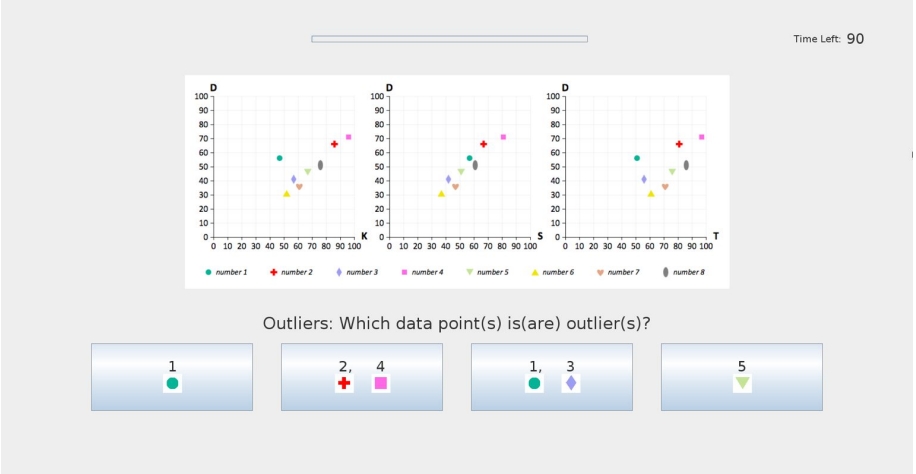
(c) Parallel Coordinates Plots.

Figure C.17: Second triple of stimuli for medium-level outlier detection task.



F	M	V	X	ID
73	98	83	99	1
48	69	53	78	2
33	54	39	63	3
38	63	49	73	4
53	78	63	88	5
58	49	59	53	6
68	88	69	83	7
43	59	44	58	8

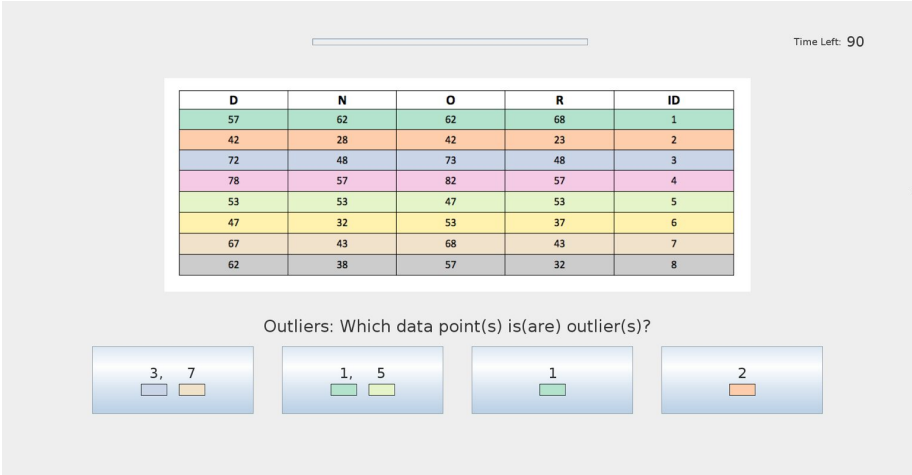
(a) Data Table.



(b) Scatter Plots.

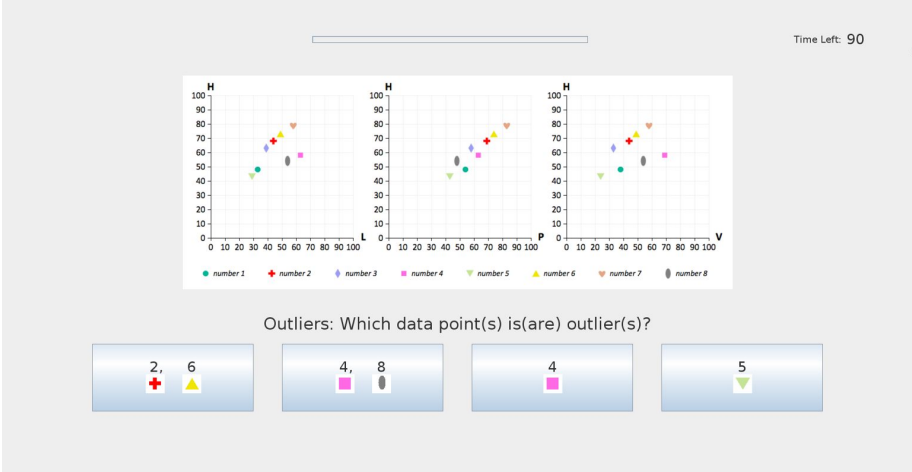
(c) Parallel Coordinates Plots.

Figure C.18: First triple of stimuli for hard-level outlier detection task.

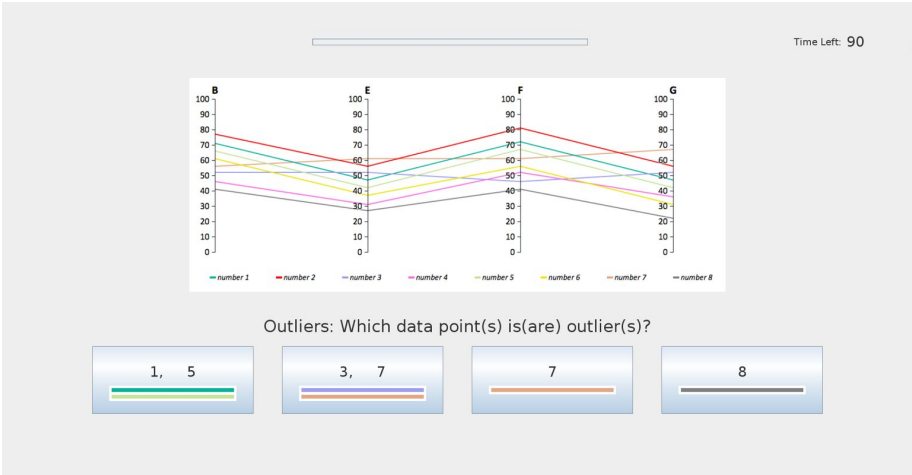


D	N	O	R	ID
57	62	62	68	1
42	28	42	23	2
72	48	73	48	3
78	57	82	57	4
53	53	47	53	5
47	32	53	37	6
67	43	68	43	7
62	38	57	32	8

(a) Data Table.

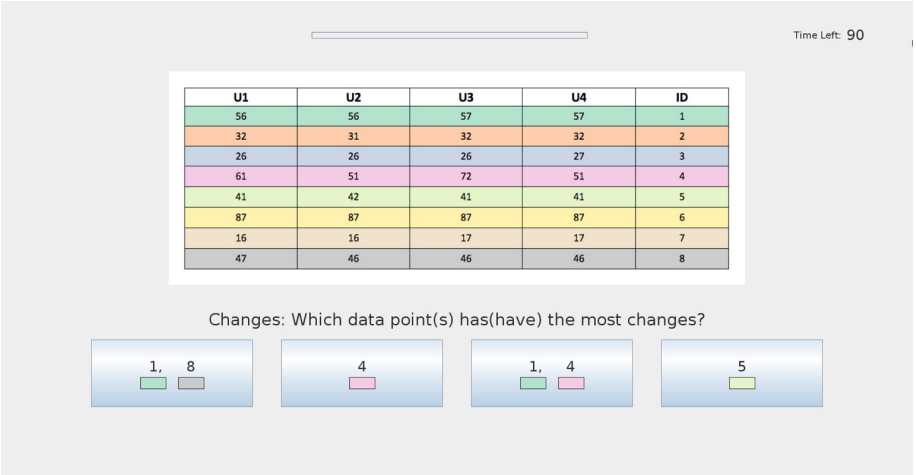


(b) Scatter Plots.

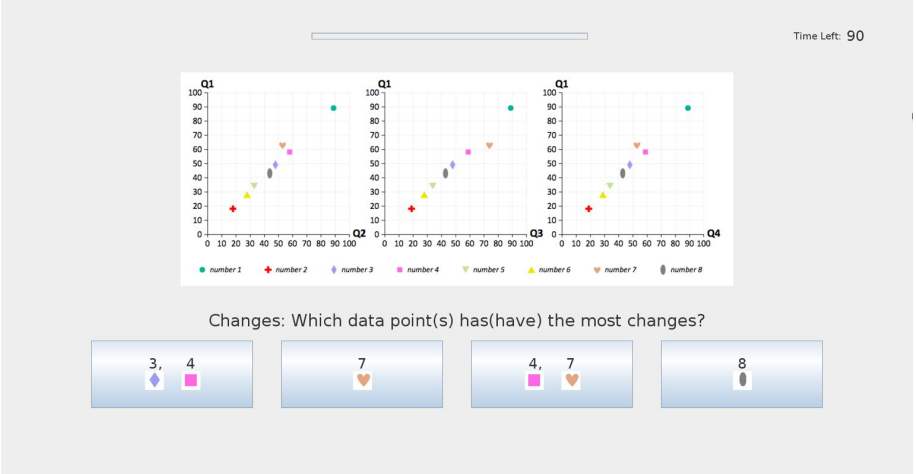


(c) Parallel Coordinates Plots.

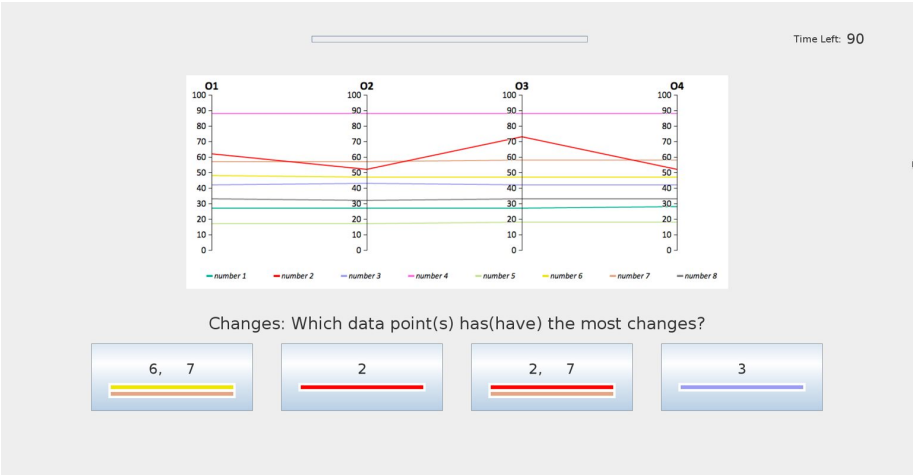
Figure C.19: Second triple of stimuli for hard-level outlier detection task.



(a) Data Table.

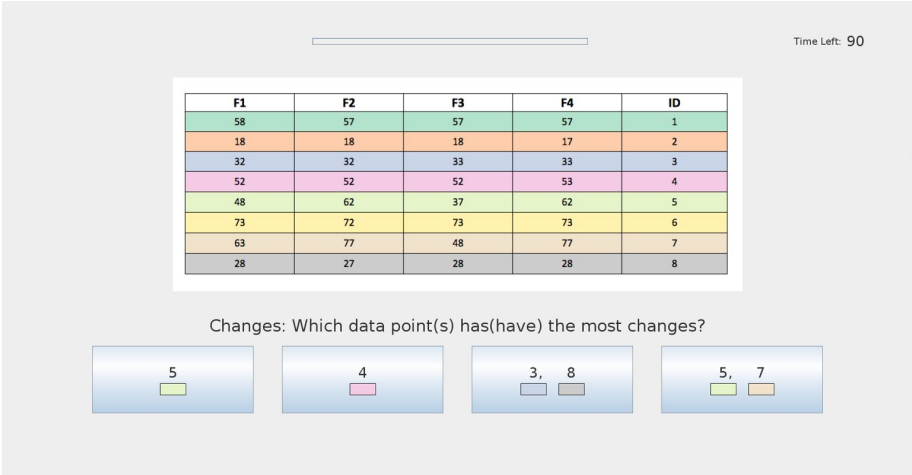


(b) Scatter Plots.



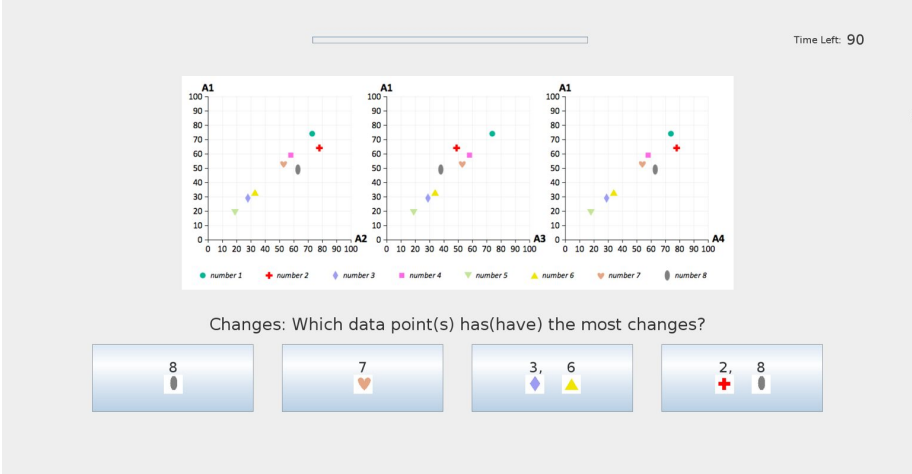
(c) Parallel Coordinates Plots.

Figure C.20: First triple of stimuli for easy-level change detection task.

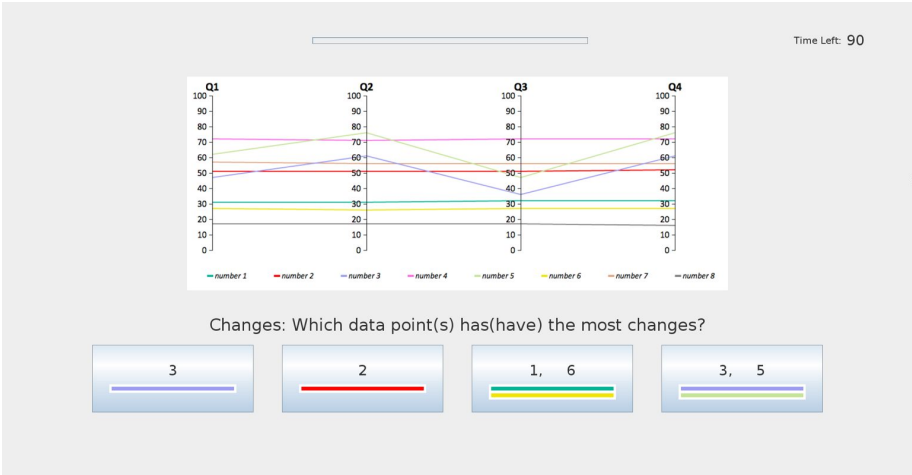


F1	F2	F3	F4	ID
58	57	57	57	1
18	18	18	17	2
32	32	33	33	3
52	52	52	53	4
48	62	37	62	5
73	72	73	73	6
63	77	48	77	7
28	27	28	28	8

(a) Data Table.

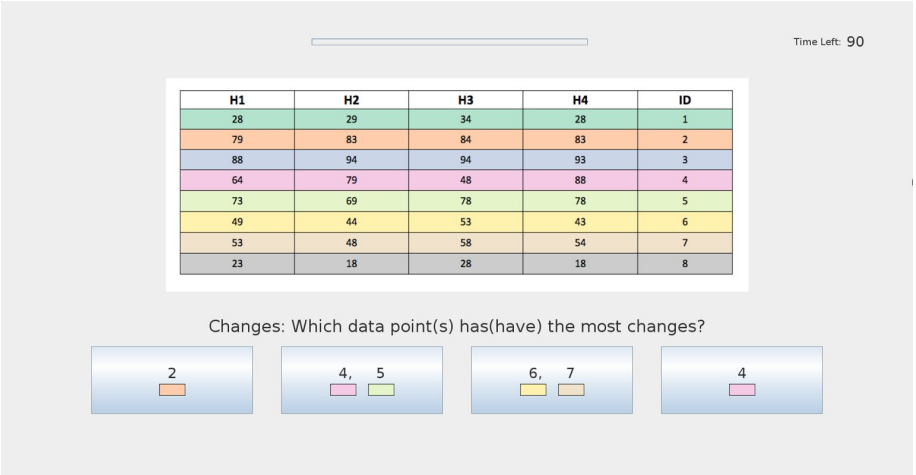


(b) Scatter Plots.



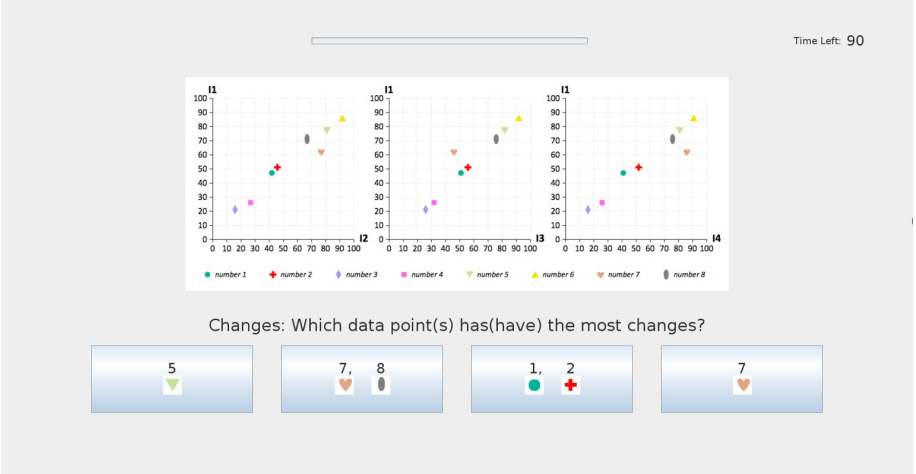
(c) Parallel Coordinates Plots.

Figure C.21: Second triple of stimuli for easy-level change detection task.

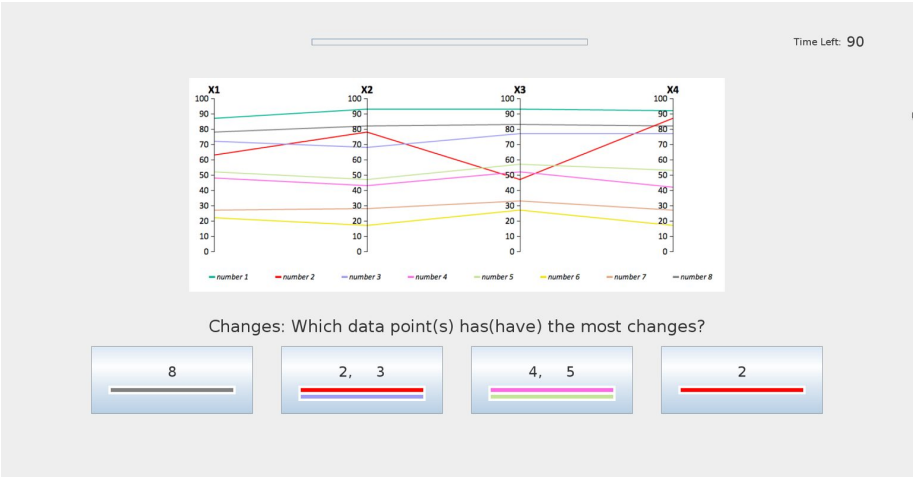


H1	H2	H3	H4	ID
28	29	34	28	1
79	83	84	83	2
88	94	94	93	3
64	79	48	88	4
73	69	78	78	5
49	44	53	43	6
53	48	58	54	7
23	18	28	18	8

(a) Data Table.

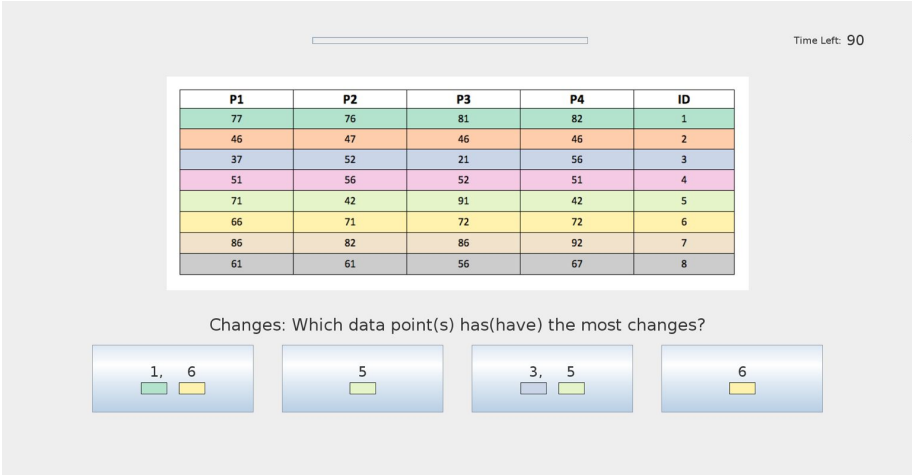


(b) Scatter Plots.



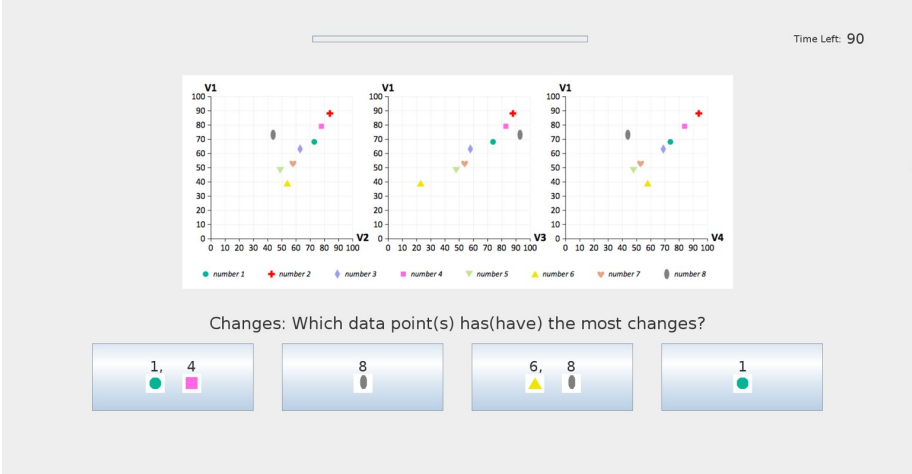
(c) Parallel Coordinates Plots.

Figure C.22: First triple of stimuli for medium-level change detection task.



P1	P2	P3	P4	ID
77	76	81	82	1
46	47	46	46	2
37	52	21	56	3
51	56	52	51	4
71	42	91	42	5
66	71	72	72	6
86	82	86	92	7
61	61	56	67	8

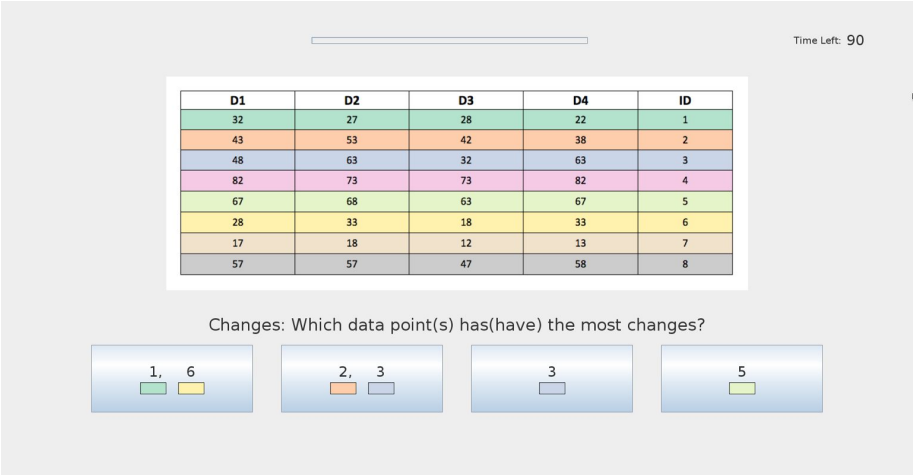
(a) Data Table.



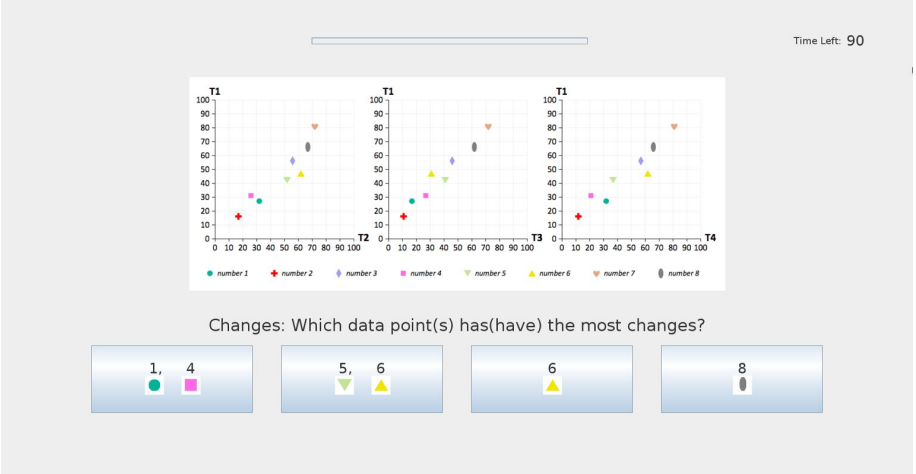
(b) Scatter Plots.

(c) Parallel Coordinates Plots.

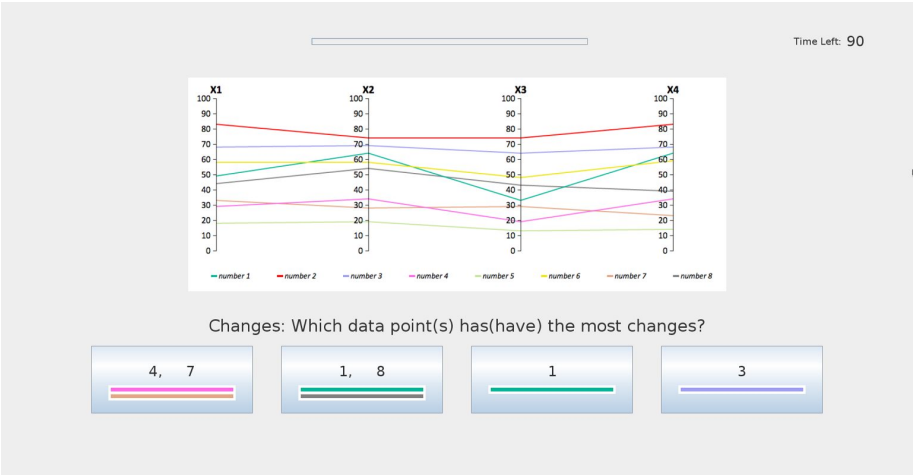
Figure C.23: Second triple of stimuli for medium-level change detection task.



(a) Data Table.

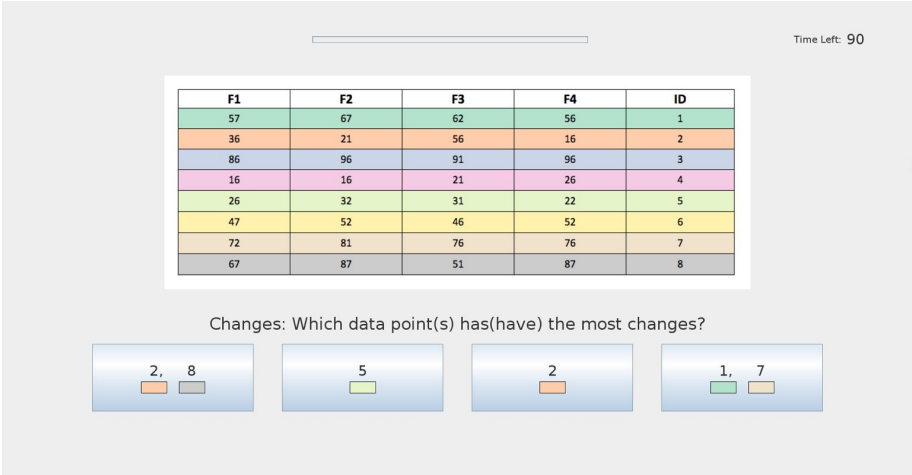


(b) Scatter Plots.



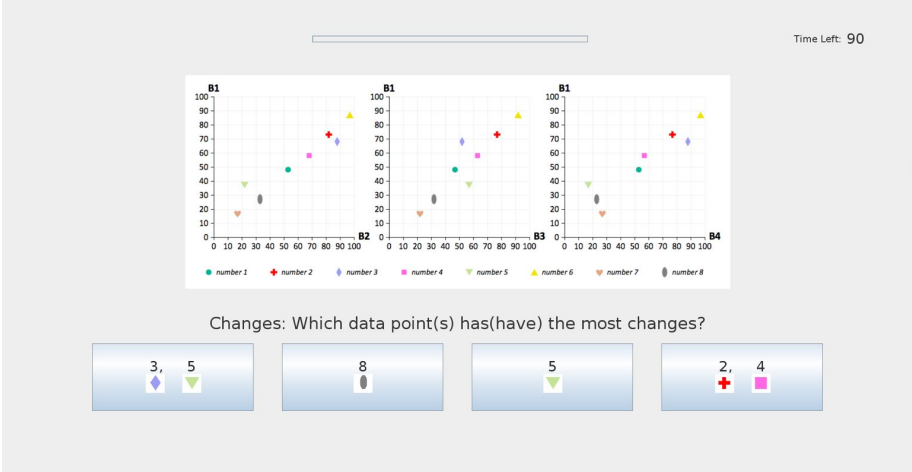
(c) Parallel Coordinates Plots.

Figure C.24: First triple of stimuli for hard-level change detection task.

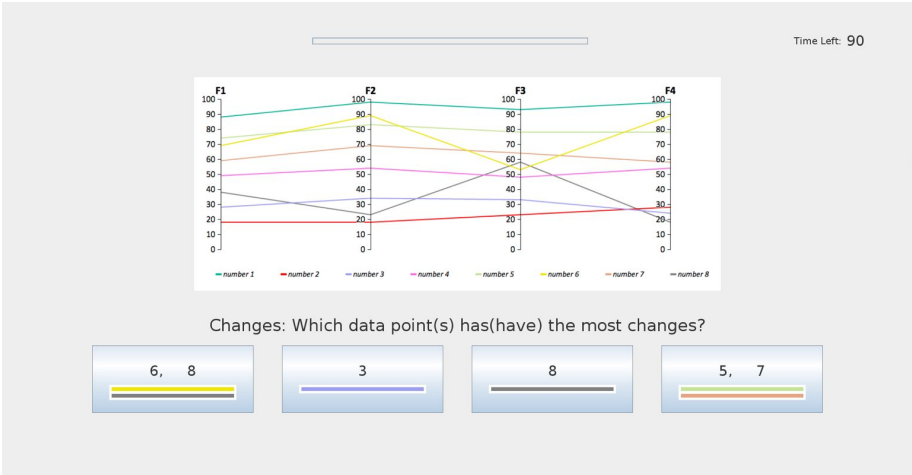


F1	F2	F3	F4	ID
57	67	62	56	1
36	21	56	16	2
86	96	91	96	3
16	16	21	26	4
26	32	31	22	5
47	52	46	52	6
72	81	76	76	7
67	87	51	87	8

(a) Data Table.



(b) Scatter Plots.

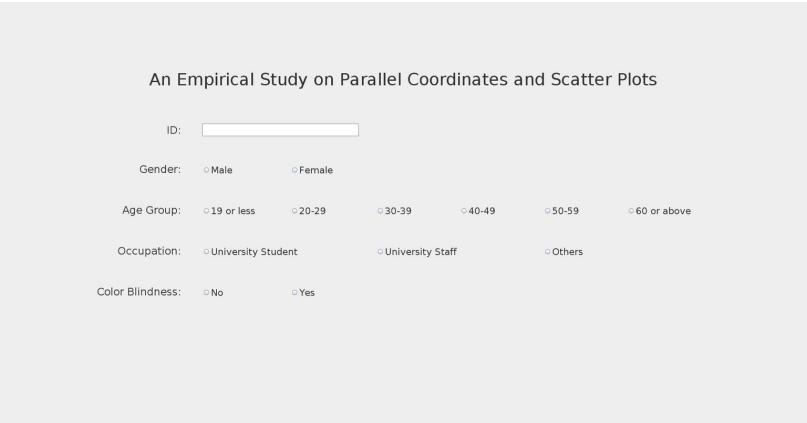


(c) Parallel Coordinates Plots.

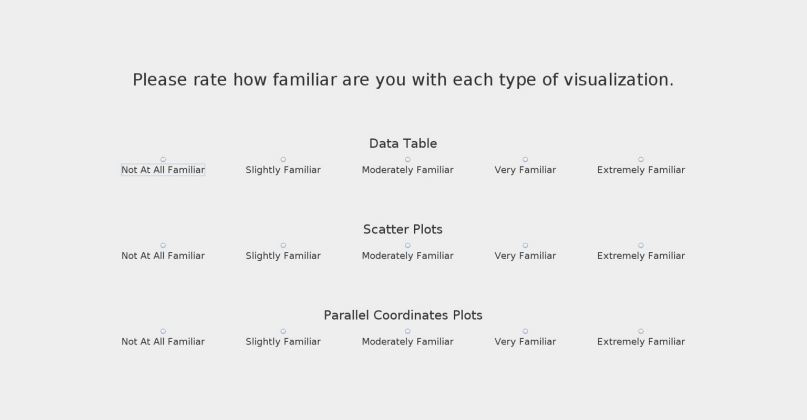
Figure C.25: Second triple of stimuli for hard-level change detection task.

Appendix D

Interfaces of the Software

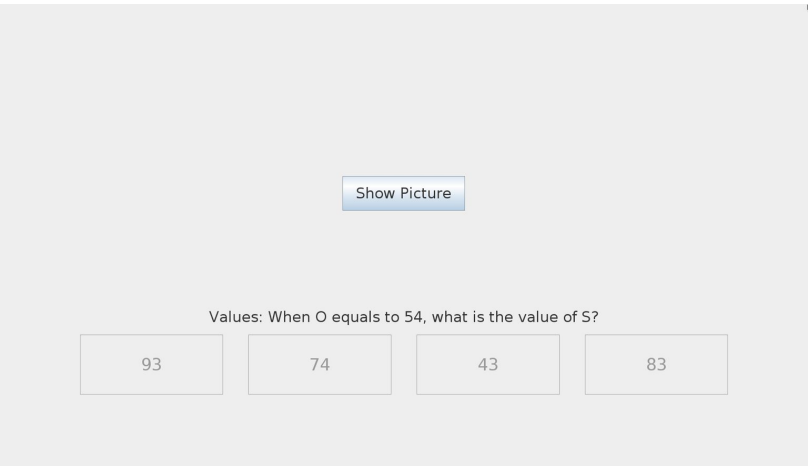


(a) Demographics information.

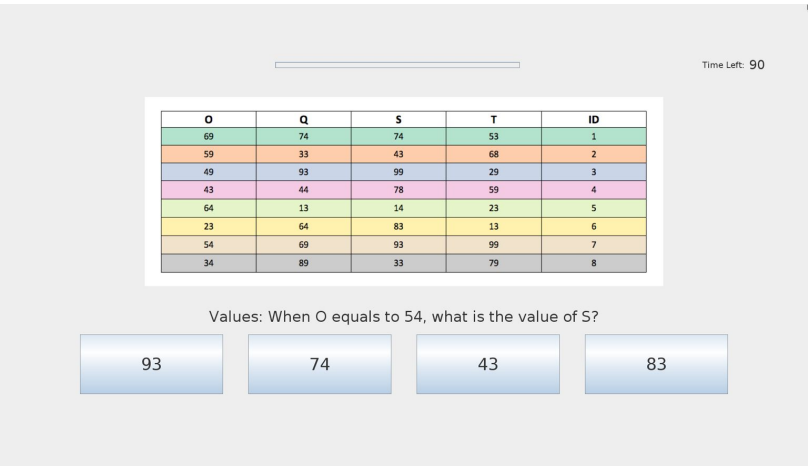


(b) Familiarity rating.

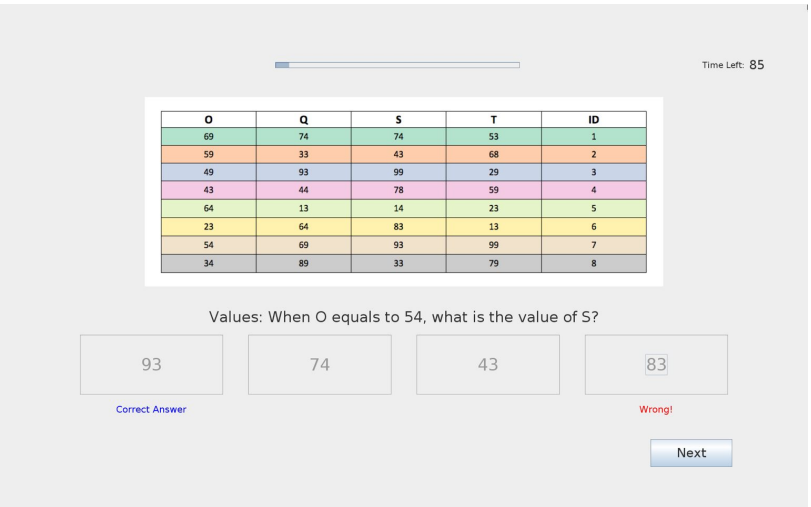
Figure D.1: Interfaces for the pre-study questionnaire.



(a) Question and four optional answers for the training trial are shown on the screen.

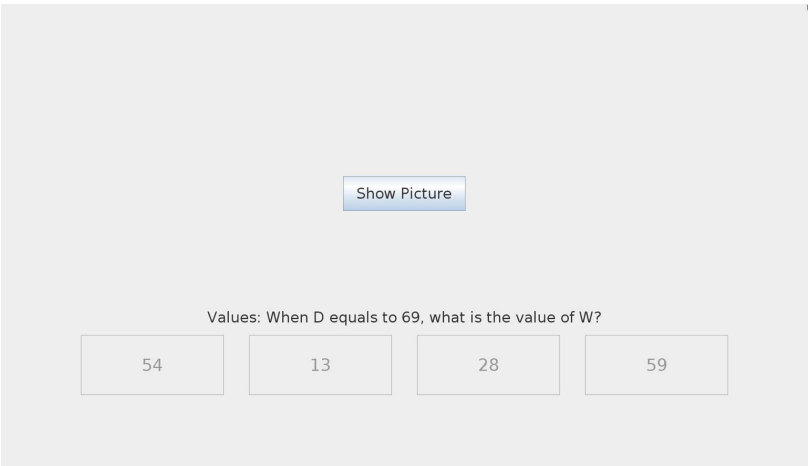


(b) A stimuli is shown after 'show picture' button is clicked.

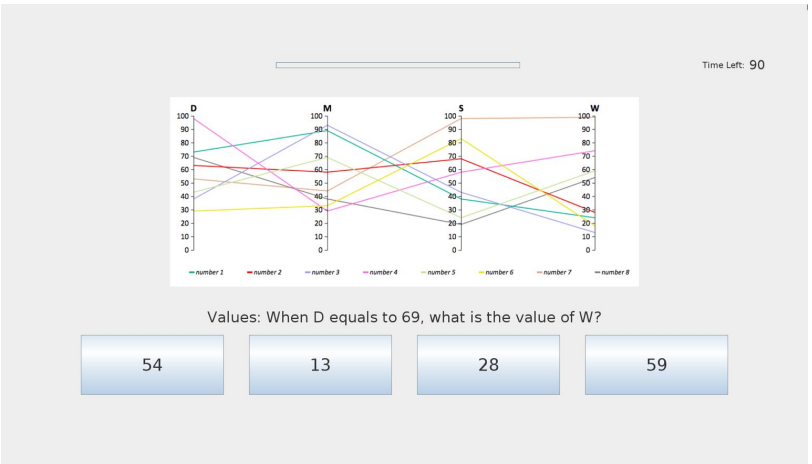


(c) Feedback is given after a choice is selected.

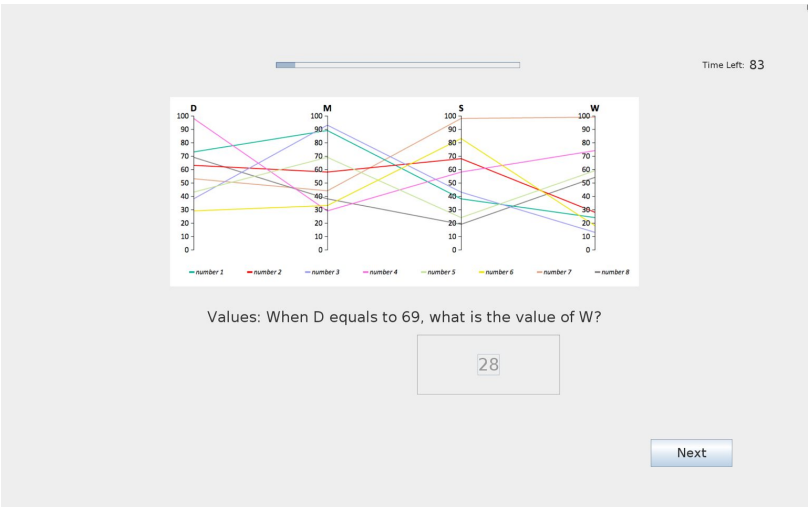
Figure D.2: Interfaces for the training sections.



(a) Question and four optional answers for the trial are shown on the screen.

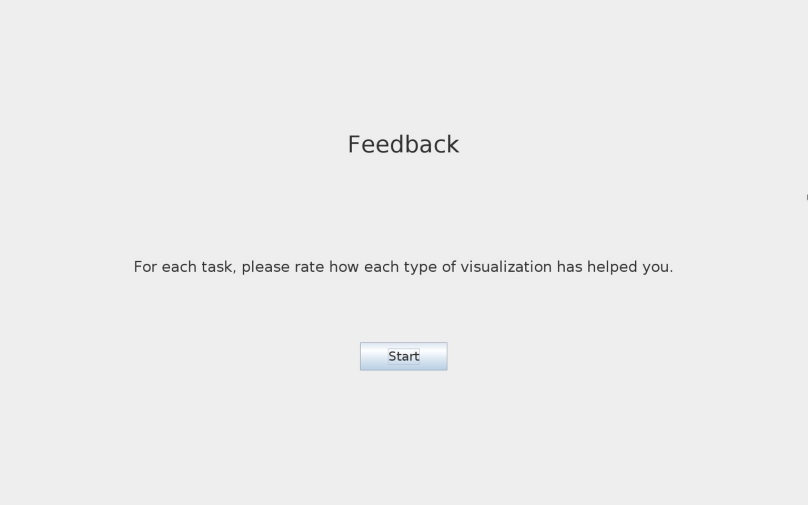


(b) A stimuli is shown after 'show picture' button is clicked.



(c) No feedback is given in the main trial section.

Figure D.3: Interfaces for the main trial sections.



(a) Feedback section page.

(b) Effectiveness rating for value retrieval task.

Figure D.4: Interfaces for the post-study questionnaire.

Appendix E

The Experiments

Figure E.1: Sample pictures for the experiments.

Appendix F

Data Tables for Result Analyses

Task	Technique	Effectiveness Rating				
		Not At All Effective	Slightly Effective	Moderately Effective	Very Effective	Extremely Effective
Value Retrieval Task	DT	1	-	7	11	23
	SCP	-	18	11	11	2
	PCP	-	1	14	20	7
Clustering Task	DT	31	8	2	1	-
	SCP	-	1	10	21	10
	PCP	-	4	16	14	8
Outlier Detection Task	DT	21	14	5	2	-
	SCP	-	1	8	20	13
	PCP	-	3	8	18	13
Change Detection Task	DT	11	7	18	5	1
	SCP	-	7	22	12	1
	PCP	-	1	3	15	23

(a) Effectiveness rating of the three visualisation techniques for all 42 participants.

Task	Reading Time			
	Mean	Standard Deviation	Min	Max
Value Retrieval	2.96	1.17	0.92	6.38
Clustering	3.00	2.27	0.85	14.49
Outlier Detection	2.32	1.01	0.78	5.77
Change Detection	2.56	1.03	1.07	5.52
All Tasks	2.71	1.26	0.91	7.59

(b) Descriptive statistical analysis for reading time

Task	Technique	Choice Selection				
		Correct Answer	Hard Distractor	Medium Distractor	Easy Distractor	No Answer
Value Retrieval Task	DT	251	-	-	1	-
	SCP	239	6	6	1	-
	PCP	244	2	4	2	-
Clustering Task	DT	107	76	43	11	15
	SCP	150	81	14	6	1
	PCP	193	25	18	10	6
Outlier Detection Task	DT	143	54	42	12	1
	SCP	217	19	15	1	-
	PCP	239	7	6	-	-
Change Detection Task	DT	206	35	5	6	-
	SCP	215	33	2	2	-
	PCP	246	5	1	-	-

(c) Choice selection for all 252 trials of each technique in each task.

Figure F.1: Analyses for effectiveness rating, reading time, and choice selection.

Task	Task Difficulty	Technique	Accuracy				Response Time			
			Mean	Standard Deviation	Min	Max	Mean	Standard Deviation	Min	Max
Value Retrieval Task	All Levels	DT	5.98	0.15	5	6	6.97	2.42	3.03	14.59
		SCP	5.69	0.72	3	6	15.95	5.90	5.82	31.94
		PCP	5.81	0.51	4	6	10.49	3.00	5.67	19.16
	Easy Level	DT	2.00	0.00	2	2	6.58	2.35	3.03	15.63
		SCP	1.88	0.40	0	2	15.93	6.92	5.37	36.40
		PCP	1.93	0.26	1	2	11.63	3.94	5.28	23.14
	Medium Level	DT	1.98	0.15	1	2	7.78	3.08	3.33	17.75
		SCP	2.00	0.00	2	2	13.03	6.29	5.40	38.55
		PCP	1.93	0.26	1	2	8.97	2.34	4.90	16.77
	Hard Level	DT	2.00	0.00	2	2	6.54	3.07	2.72	20.27
		SCP	1.81	0.45	0	2	18.90	7.96	6.69	38.87
		PCP	1.95	0.22	1	2	10.88	4.16	4.95	27.09
Clustering Task	All Levels	DT	2.55	1.52	0	6	49.49	14.91	14.17	80.10
		SCP	3.57	1.02	1	5	28.59	11.21	11.13	49.44
		PCP	4.60	1.59	1	6	29.26	14.90	9.35	59.97
	Easy Level	DT	1.40	0.63	0	2	48.19	18.06	15.03	88.20
		SCP	1.88	0.40	0	2	21.33	14.61	7.02	60.81
		PCP	1.86	0.42	0	2	21.33	15.33	5.08	75.72
	Medium Level	DT	0.79	0.78	0	2	42.39	19.11	1.52	89.66
		SCP	1.21	0.56	0	2	30.27	13.90	9.69	65.08
		PCP	1.40	0.77	0	2	34.05	20.66	9.28	89.61
	Hard Level	DT	0.36	0.58	0	2	57.90	19.70	16.78	90.00
		SCP	0.48	0.51	0	1	34.16	15.20	9.61	78.02
		PCP	1.33	0.85	0	2	32.41	17.42	7.28	70.80
Outlier Detection Task	All Levels	DT	3.40	1.56	0	6	31.19	9.43	13.93	50.09
		SCP	5.17	0.91	3	6	11.49	6.48	3.39	30.78
		PCP	5.69	0.60	4	6	8.40	4.08	2.91	20.30
	Easy Level	DT	1.55	0.67	0	2	34.00	13.92	9.60	61.65
		SCP	1.95	0.22	1	2	4.74	3.12	2.11	20.75
		PCP	2.00	0.00	2	2	4.01	1.97	1.65	10.74
	Medium Level	DT	1.29	0.64	0	2	32.49	11.82	11.25	59.20
		SCP	1.71	0.60	0	2	8.81	9.30	1.90	46.40
		PCP	1.86	0.42	0	2	12.10	9.53	3.30	46.38
	Hard Level	DT	0.57	0.74	0	2	27.09	12.66	8.55	59.84
		SCP	1.50	0.63	0	2	20.92	12.82	5.36	56.00
		PCP	1.83	0.38	1	2	9.10	6.61	2.25	41.49
Change Detection Task	All Levels	DT	4.90	1.08	2	6	23.68	8.11	11.28	45.23
		SCP	5.12	1.09	2	6	16.85	9.91	5.29	47.54
		PCP	5.86	0.52	3	6	6.96	2.22	3.32	11.39
	Easy Level	DT	1.79	0.47	0	2	16.54	6.77	8.45	38.72
		SCP	1.93	0.26	1	2	14.53	11.79	2.91	64.70
		PCP	1.95	0.31	0	2	5.64	3.31	2.10	16.42
	Medium Level	DT	1.81	0.45	0	2	26.15	11.12	12.00	65.63
		SCP	1.71	0.46	1	2	12.55	8.04	4.13	44.97
		PCP	1.95	0.22	1	2	5.63	2.40	2.32	12.55
	Hard Level	DT	1.31	0.72	0	2	28.34	11.35	11.52	57.54
		SCP	1.48	0.63	0	2	23.49	14.79	7.26	69.81
		PCP	1.95	0.22	1	2	9.62	3.46	2.95	18.43

Figure F.2: Descriptive statistical analysis for accuracy and response time.

Appendix G

Planned and Actual Schedules

Our actual schedule of the project have steadily progress as according to the planned schedule, with most processes starting well ahead of time.

The project was done during the period of April to September. It consisted of six processes: information gathering and literature review, iterative development process, experiment, result analysis, and dissertation writing.

In April, we did information gathering and literature review, which is the first and the most important procedure in research study. We studied previous experiments and investigations to obtain guidelines and sources, in order to provide relatively fair experimental methods.

Then, we developed the software for the experiment by iteratively performing the five procedures: planning and requirement analysis, user study design, stimuli generation, software development, and testing. This development process has been iteratively performed until the software program can completely perform as expected, which has ended around the end of June.

Later in July, we conducted the experiments for the empirical study. A pilot study was also conducted prior to the main experiments, in order to ensure the software quality.

Then, roughly in the mid of July, we performed the result analyses to compare whether which visualisation techniques yield better user performance, and provide statistical evidence.

The dissertation writing started at the beginning of May and continued throughout the research until the beginning of September.

Figure G.1 summarises the actual schedule described above, and provides the planned schedule for comparison.

WBS	Tasks	Planned		Actual	
		Start	End	Start	End
1	**Information Gathering and Literature Review**	**21-Apr-14**	**4-May-14**	**14-Apr-14**	**2-May-14**
2	**Iterative Development Process**	**5-May-14**	**29-Jun-14**	**28-Apr-14**	**29-Jun-14**
2.1	Planning and Requirement Analysis	5-May-14	29-Jun-14	28-Apr-14	29-Jun-14
2.2	User Study Design	5-May-14	29-Jun-14	28-Apr-14	29-Jun-14
2.3	Stimuli Generation	5-May-14	29-Jun-14	28-Apr-14	29-Jun-14
2.4	Software Development	5-May-14	29-Jun-14	28-Apr-14	29-Jun-14
2.5	Testing	5-May-14	29-Jun-14	28-Apr-14	29-Jun-14
3	**Experiment**	**30-Jun-14**	**13-Jul-14**	**3-Jul-14**	**11-Jul-14**
4	**Result Analysis**	**14-Jul-14**	**3-Aug-14**	**12-Jul-14**	**25-Jul-14**
5	**Dissertation Writing**	**5-May-14**	**4-Sep-14**	**3-May-14**	**3-Sep-14**

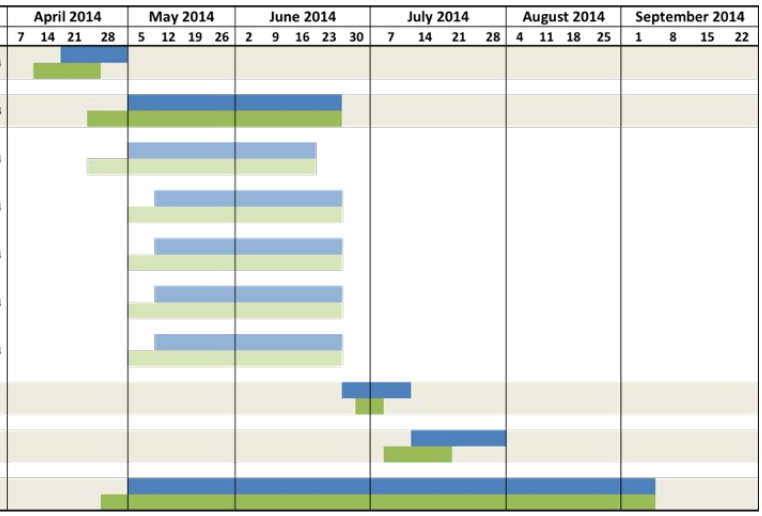


Figure G.1: Planned (blue) and actual (green) schedules for our project.

Bibliography

[AdOL04] A. O. Artero, M. C. F. de Oliveira, and H. Levkowitz. Uncovering clusters in crowded parallel coordinates visualizations. In *Proceedings of the IEEE Symposium on Information Visualization*, INFOVIS '04, pages 81–88, Washington, DC, USA, 2004. IEEE Computer Society.

[AES05] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *Proceedings of IEEE Symposium on Information Visualization*, INFOVIS 2005, pages 111–117, October 2005.

[And28] E. Anderson. The problem of species in the northern blue flags, iris versicolor l. and iris virginica l. *Annals of the Missouri Botanical Garden*, 15(3):241–332, 1928.

[And08] K. Andrews. Evaluation comes in many guises. In *Proceedings of AVI Workshop on BEyond time and errors (BELIV) Position Paper*, 2008.

[ANO12] M. Abdelaziz, R. Nancy, and S. Olivier. Survey of multidimensional visualization techniques. In *Proceedings of CGVCVIP'12: Computer Graphics, Visualization, Computer Vision and Image Processing Conference*, 2012.

[AR11] S. B. Azhar and M. J. Rissanen. Evaluation of parallel coordinates for interactive alarm filtering. In *Proceedings of 15th International Conference on Information Visualisation*, pages 102–109. IEEE, 2011.

[Asi85] D. Asimov. The grand tour: a tool for viewing multidimensional data. *SIAM Journal on Scientific and Statistical Computing*, 6(1):128–143, 1985.

[Atk11] G. Atkinson. Analysis of repeated measurements in physical therapy research. *Physical Therapy in Sport*, 2(4):194–208, November 2011.

[BC87] R. A. Becker and W. S. Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, May 1987.

[BG97] N. Burns and S. K. Grove. *The Practice of Nursing Research Conduct, Critique, and Utilization.* W.B. Saunders and Co, Philadelphia, 1997.

[BHVW00] M. Bruls, K. Huizing, and J. J. Van Wijk. *Squarified treemaps.* Springer, 2000.

[Bis06] C. M. Bishop. *Pattern Recognition and Machine Learning.* Springer, New York, 2006.

[Bos] M. Bostock. D3.js - data-driven documents. `http://d3js.org/`. [Online; Accessed 5 May 2014].

[Bow97] A. Bowling. *Research Methods in Health.* Open University Press, Buckingham, 1997.

[BR07] L. Barkhuus and J. A. Rode. From mice to men - 24 years of evaluation in chi. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, New York, NY, USA, 2007. ACM.

[BS04] N. Barlow and L. Stuart. Animator: a tool for the animation of parallel coordinates. In *Proceedings Eighth International Conference on Information Visualisation*, INFOVIS '04, pages 725–730, July 2004.

[Car08] S. Carpendale. Evaluating information visualizations. In A. Kerren, J. T. Stasko, J.-D. Fekete, and C. North, editors, *Information Visualization*, pages 19–45. Springer-Verlag, Berlin, Heidelberg, 2008.

[CFB13] M. Chen, L. Floridi, and R. Borgo. What is visualization really for? *ArXiv e-prints*, 2013.

[CK10] T. M. Chromiński Kornel. Comparison of outlier detection methods in biomedical data. *Journal of Medical Informatics and Technologies*, 16, 2010.

[Col13] ColorBrewer. Colorbrewer: Color advice for maps. `http://colorbrewer2.org/`, 2013. [Online; Accessed 30 May 2014].

[CZ13] A. Cuzzocrea and D. Zall. Parallel coordinates technique in visual data mining: Advantages, disadvantages and combinations. In *Proceedings of Information Visualisation (IV), 2013 17th International Conference*, pages 278–284, July 2013.

[DDK+13] B. Duffy, A. Dasgupta, R. Kosara, S. Walton, and M. Chen. Measuring visual complexity of cluster-based visualizations. *ArXiv e-prints*, February 2013.

[ED06] G. Ellis and A. Dix. An explorative analysis of user evaluation studies in information visualisation. In *Proceedings of the 2006 AVI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization*, BELIV '06, pages 1–7, New York, NY, USA, 2006. ACM.

[FCI05] E. Fanea, S. Carpendale, and T. Isenberg. An interactive 3d integration of parallel coordinates and star glyphs. In *Proceedings of IEEE Symposium on Information Visualization, 2005*, INFOVIS 2005, pages 149–156, Washington, DC, USA, October 2005. IEEE Computer Society.

[FCL09] A. Forsberg, J. Chen, and D. H. Laidlaw. Comparing 3d vector field visualization methods: A user study. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1219–1226, 2009.

[FD05] M. Friendly and D. Denis. The early origins and development of the scatterplot. *Journal of the History of the Behavioral Sciences*, 41(2):103–30, 2005.

[FFT74] M. Fisherkeller, J. Friedman, and J. Tukey. *Prim-9: An Interactive Multidimensional Data Display and Analysis System.* Stanford Linear Accelerator Center, 1974.

[Fis25] R. A. Fisher. *Statistical methods for research workers.* Oliver and Boyd, Edinburgh, 1925.

[FLC+02] C. M. D. S. Freitas, P. R. G. Luzzardi, R. A. Cava, M. Winckler, M. S. Pimenta, and L. P. Nedel. On evaluating information visualization techniques. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '02, pages 373–374, New York, NY, USA, 2002. ACM.

[FNS14] B. Fu, N. F. Noy, and M.-A. Storey. Eye tracking the user experience-an evaluation of ontology visualization techniques. May 2014.

[Fou] T. R. Foundation. The r project for statistical computing. `http://www.r-project.org/`. [Online; Accessed 5 May 2014].

[FWR99] Y.-H. Fua, M. O. Ward, and E. A. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *Proceedings of the Conference on Visualization '99: Celebrating Ten Years*, VIS '99, pages 43–50, Los Alamitos, CA, USA, 1999. IEEE Computer Society Press.

[Gal86] F. Galton. Regression towards mediocrity in hereditary stature. *Journal of the Anthropological Institute of Great Britain and Ireland*, pages 246–263, 1886.

[Gal90] F. Galton. Kinship and correlation. *The North American Review*, pages 419–431, 1890.

[Gar12] G. D. Garson. *Correlation*. Number 3 in Statistical Associates Blue Book Series. Statistical Associates, April 2012.

[GGo] GGobi. Ggobi data visualization system. `http://www.ggobi.org/`. [Online; Accessed 5 May 2014].

[GH83] H. Gannett and F. W. Hewes. *Statistical atlas of the United States*. Charles Scribner's Sons, New York, 1883.

[Gir91] E. R. Girden. *ANOVA: Repeated Measures*. Number 84 in Sage university paper series on quantitative applications in social sciences. SAGE Publications Ltd, Newbury Park, November 1991.

[GK03] M. Graham and J. Kennedy. Using curves to enhance parallel coordinate visualisations. In *Information Visualization, 2003. IV 2003. Proceedings. Seventh International Conference on*, pages 10–16, July 2003.

[Gmb] M. GmbH. Parallel coordinates interactive visualization tool by macrofocus. `http://www.high-d.com/`. [Online; Accessed 5 May 2014].

[Goo09a] J. Goodall. Visualization is better! a comparative evaluation. In *Proceedings of 6th International Workshop on Visualization for Cyber Security*, VizSec 2009, pages 57–68, October 2009.

[Goo09b] C. J. Goodwin. *Research in psychology: Methods and design*. John Wiley & Sons, USA, 2009.

[Gru69] F. E. Grubbs. Procedures for detecting outlying observations in samples. *Technometrics*, 11(1):1–21, February 1969.

[Her33] J. F. Herschel. On the investigation of the orbits of revolving double stars. *Memoirs of the Royal Astronomical Society*, 5:171, 1833.

[HHB07] M. Henley, M. Hagen, and R. D. Bergeron. Evaluating two visualization techniques for genome comparison. In *Proceedings of 11th International Conference Information Visualization (IV '07)*, pages 551–558. IEEE, 2007.

[HLD02] H. Hauser, F. Ledermann, and H. Doleisch. Angular brushing of extended parallel coordinates. In *IEEE Symposium on Information Visualization*, INFOVIS 2002, pages 127–130, 2002.

[HN06] K. Honda and J. Nakano. 3 dimensional parallel coordinates plot and its use for variable selection. In *Proceedings in Computational Statistics*, pages 187–195. Springer, 2006.

[HNR05] M. Hemmje, C. Niederée, and T. Risse, editors. *From Integrated Publication and Information Systems to Information and Knowledge Environments*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[HR00] D. M. Hilbert and D. F. Redmiles. Extracting usability information from user interface events. *ACM Computing Surveys (CSUR)*, 32(4):384–421, December 2000.

[HvW10] D. Holten and J. J. van Wijk. Evaluation of cluster identification performance for different pcp variants. *Computer Graphics Forum*, 29(3):793–802, August 2010.

[IBMa] IBM. Ibm spss software. `http://www.spss.com`. [Online; Accessed 5 May 2014].

[IBMb] IBM. Many eyes. `http://www-958.ibm.com/software/analytics/manyeyes/`. [Online; Accessed 5 May 2014].

[ID90] A. Inselberg and B. Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proceedings of the First IEEE Conference on Visualization: Visualization '90*. IEEE Computer Society Press, 1990.

[IH01] M. Y. Ivory and M. A. Hearst. The state of the art in automating usability evaluation of user interfaces. *ACM Computing Surveys (CSUR)*, 33(4):470–516, December 2001.

[Ins85] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(2):69–91, 1985.

[JCJ05] J. Johansson, M. Cooper, and M. Jern. 3-dimensional display for clustered multi-relational parallel coordinates. In *Proceedings Ninth International Conference on Information Visualisation*, pages 188–193, July 2005.

[JFLC08] J. Johansson, C. Forsell, M. Lind, and M. Cooper. Perceiving patterns in parallel coordinates: determining thresholds for identification of relationships. *Information Visualization*, 7(2):152–162, April 2008.

[JLJC06] J. Johansson, P. Ljung, M. Jern, and M. Cooper. Revealing structure in visualizations of dense 2d and 3d parallel coordinates. *Information Visualization*, 5(2):125–136, June 2006.

[Joh99] D. H. Johnson. The insignificance of statistical significance testing. *Journal of Wildlife Management*, 63(3):763–772, 1999.

[JPKM07] F. Jourdan, A. Paris, P.-Y. Koenig, and G. Melançon. Multiscale scatterplot matrix for visual and interactive exploration of metabonomic data. In *Pixelization Paradigm*, pages 202–215. Springer, 2007.

[KD09] R. Kincaid and K. Dejgaard. Massvis: Visual analysis of protein complexes using mass spectrometry. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology*, pages 163–170. IEEE, 2009.

[KHK02] A. W. Kerr, H. K. Hall, and S. A. Kozub. *Doing Statistics With SPSS*. SAGE Publications Ltd, London, 2002.

[KHLW09] A. Klippel, F. Hardisty, R. Li, and C. Weaver. Colour-enhanced star plot glyphs: Can salient shape characteristics be overcome? *Cartographica: The International Journal for Geographic Information and Geovisualization*, 44(3):217–231, 2009.

[KZZM12] X. Kuang, H. Zhang, S. Zhao, and M. J. McGuffin. Tracing tuples across dimensions: A comparison of scatterplots and parallel coordinate plots. *Computer Graphics Forum*, 31(3):1365–1374, June 2012.

[LBI+11] H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale. Seven guiding scenarios for information visualization evaluation. Technical Report 2011-992-04, University of Calgary, 2011.

[LFK00] K. Liston, M. Fischer, and J. Kunz. Designing and evaluating visualization techniques for construction planning. In *Proc. of the 8th International Conference on Computing in Civil and Building Engineering (ICCCBE-VIII), Stanford University, Stanford, CA*, pages 1293–300, 2000.

[Lik32] R. Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 22(140):1–55, June 1932.

[LMW08] J. Li, J.-B. Martens, and J. J. v. Wijk. Judging correlation from scatterplots and parallel coordinate plots. *Information Visualization*, pages 1–18, 2008.

[LRB03] M. D. Lee, R. E. Reilly, and M. E. Butavicius. An empirical evaluation of chernoff faces, star glyphs, and spatial visualizations for binary data. In *Proceedings of the Asia-Pacific symposium on Information visualisation*, volume 24 of *APVis '03*, pages 1–10, Adelaide, Australia, 2003. Australian Computer Society, Inc.

[LWZK08] Y. Luo, D. Weiskopf, H. Zhang, and A. E. Kirkpatrick. Cluster visualization in parallel coordinates using curve bundles. *IEEE Transactions on Visualization and Computer Graphics*, 2008.

[Mar07] D. Marghescu. Multidimensional data visualization techniques for financial performance data: A review. Technical Report 810, Turku Centre for Computer Science, February 2007.

[Mas99] J.-F. Mas. Monitoring land-cover changes: a comparison of change detection techniques. *International journal of remote sensing*, 20(1):139–152, 1999.

[Mcg95] E. Mcgrath. Methodology matters: Doing research in the behavioral and social sciences. In *Proceedings of Readings in Human-Computer Interaction: Toward the Year 2000*, pages 152–169. Morgan Kaufman, 1995.

[Mit97] T. M. Mitchell. *Machine Learning*. McGraw-Hill Series in Computer Science. McGraw-Hill Science/Engineering/Math, 1 edition, March 1997.

[MKMM07] S. Matsumoto, Y. Kamei, A. Monden, and K.-i. Matsumoto. Comparison of outlier detection methods in fault-proneness models. In *Proceedings of the First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)*, pages 461–463. IEEE, 2007.

[Mun09] T. Munzner. A nested model for visualization design and validation. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):921–928, November 2009.

[Mur12] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, London, 2012.

[MW02] R. E. Moustafa and E. J. Wegman. On some generalizations of parallel coordinate plots. In *Seeing a Million-A Data Visualization Workshop*, pages 41–48, 2002.

[Nap12] M. A. Napierala. What is the bonferroni correction? *AAOS Now, Journal of the AAOS*, 2012.

[OED89] *The Oxford English Dictionary (OED).* Oxford University Press., Oxford, 2 edition, 1989.

[PJ01] K. I. Penny and I. T. Jolliffe. A comparison of multivariate outlier detection methods for clinical laboratory safety data. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 50(3):295–307, September 2001.

[Pla04] C. Plaisant. The challenge of information visualization evaluation. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '04, pages 109–116, New York, NY, USA, 2004. ACM.

[Pri11] G. J. Privitera. *Statistics for the behavioral sciences.* Sage Publications, 2011.

[PVF05] R. M. Pillat, E. R. A. Valiati, and C. M. D. S. Freitas. Experimental study on evaluation of multidimensional information visualization techniques. In *Proceedings of the 2005 Latin American Conference on Human-computer Interaction*, CLIHC '05, pages 20–30, New York, NY, USA, 2005. ACM.

[QCX+07] H. Qu, W.-Y. Chan, A. Xu, K.-L. Chung, K.-H. Lau, and P. Guo. Visual analysis of the air pollution problem in hong kong. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1408–15, 2007.

[RSCT09] A. Rusu, C. Santiago, A. Crowell, and E. Thomas. Enhanced star glyphs for multiple-source data analysis. In *Proceedings of Information Visualisation, 2009 13th International Conference*, pages 183–190, July 2009.

[Sel14] H. J. Seltman. *Experimental Design and Analysis.* Carnegie Mellon University, June 2014.

[Seo06a] S. Seo. A review and comparison of methods for detecting outliers in univariate data sets. Master's thesis, University of Pittsburgh, August 2006.

[SEÖ+06b] M. Streit, R. C. Ecker, K. Österreicher, G. E. Steiner, H. Bischof, C. Bangert, T. Kopp, and R. Rogojanu. 3d parallel coordinate systems-a new data visualization method in the context of microscopy-based multicolor tissue cytometry. *Cytometry Part A*, 69(7):601–611, 2006.

[Shn92] B. Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transactions on Graphics*, 11(1):92–99, January 1992.

[Sof] T. Software. Business intelligence and analytics — tableau software. `http://www.tableausoftware.com/`. [Online; Accessed 5 May 2014].

[SS14] R. Sadana and J. Stasko. Designing and implementing an interactive scatterplot visualization for a tablet computer. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*, AVI '14, pages 265–272, New York, NY, USA, 2014. ACM.

[SSS+12] M. Schroeck, R. Shockley, J. Smart, D. Romero-Morales, and P. Tufano. Analytics: The real-world use of big data. Executive report, IBM Institute for Business Value In collaboration with Saïd Business School at the University of Oxford, 2012.

[Ste46] S. S. Stevens. On the theory of scales of measurement. *Science (New York, N.Y.)*, 103(2684):677–80, June 1946.

[SZB+09] J. Sanyal, S. Zhang, G. Bhattacharya, P. Amburn, and R. Moorhead. A user study to compare four uncertainty visualization methods for 1d and 2d datasets. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1209–1218, November 2009.

[TC05] J. J. Thomas and K. A. Cook. *Illuminating the path: The research and development agenda for visual analytics*. IEEE Computer Society Press, 2005.

[TGM83] E. R. Tufte and P. Graves-Morris. *The visual display of quantitative information*, volume 2. Graphics press Cheshire, CT, 1983.

[TGS04] J. Tyman, G. Gruetzmacher, and J. Stasko. Proceedings of infovisexplorer. In *IEEE Symposium on Information Visualization*. IEEE, 2004.

[The00] H. Theisel. Higher order parallel coordinates. In *Proceedings of VMV*, pages 415–420, 2000.

[Tuk70] J. W. Tukey. Some graphic and semi-graphic displays. In *Technometrics*, volume 12, page 205, Alexandria, VA, 1970. American Statistical Association (ASA).

[VMCJ10] C. Viau, M. J. McGuffin, Y. Chiricota, and I. Jurisica. The flowvizmenu and parallel scatterplot matrix: hybrid multidimensional visualizations for network exploration. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1100–8, 2010.

[Weg90] E. J. Wegman. Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association*, 85(411):664–675, September 1990.

[WGK10] M. O. Ward, G. Grinstein, and D. Keim. *Interactive Data Visualization: Foundations, Techniques, and Applications*. A K Peters, Ltd., Natick, Massachusetts, 2010.

[xda] xdat.org. Xdat - a free parallel coordinates software tool. `http://www.xdat.org/`. [Online; Accessed 5 May 2014].

[YGX+09] X. Yuan, P. Guo, H. Xiao, H. Zhou, and H. Qu. Scattering points in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1001–8, January 2009.

[ZYQ+08] H. Zhou, X. Yuan, H. Qu, W. Cui, and B. Chen. Visual clustering in parallel coordinates. *Computer Graphics Forum*, 27(3):1047–1054, 2008.