Interactive Visual Data Query & Exploration

Techniques for visual data analytics through visual query modelling and multidimensional data interaction

Phi Giang Pham

Supervisor: Associate Professor Dr. **Mao Lin Huang**

A thesis submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

in

the Faculty of Engineering and Information Technology

University of Technology Sydney

Sydney, Australia 2018

Certificate of Original Authorship

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature of candidate: Production Note:
Signature removed prior to publication.

Phi Giang Pham

Date: 22 – Jan – 2018

Acknowledgement

Today is, for me, the day of a beautiful memory which would be unforgettable during my life time. This is because I am here and writing the last but not least of the significant parts of my dissertation that is about the acknowledgement expression for the completing stage of my interesting Ph.D. study. Four years ago, I had not believed and imagined what I could and have reached as today until there was a person who appeared and changed my mind.

Absolutely, the man with the role of my supervisor is Associate Professor Mao Lin Huang, to whom I would like to express my genuine gratefulness firstly. Thanks to his advanced academic guidance, mental encouragement, especially free and active working style deployment, I have learned and experienced plenty of self-study and research methodologies and optimized the strength of mine in order to overcome the research challenges and reach the excellent achievement of today.

Additionally, I would like to thank all of my colleges who greatly supported me during the candidate in the sharing of knowledge, solving the technical problems and dealing with the life issues. I also would like to thank all of the staffs who are working in the school of Software, FEIT, UTS for their help in the administrative and financial procedure.

Finally, it is unexplainable by words and languages actually that I would like to thank all of my family members, who are always beside me, look after me, and love me, especially the meaningful accompany of my wife Le Thu Trang Ho and my daughter Mai Thanh Pham. Without all of them, my Ph.D. study could not be started and completed successfully.

Thanks for all.

Table of Contents

List of Algorithms viii

List of Figures ix

List of Tables xv

Abstract xvi

Chapter 1 Introduction **1**

1.1 From Information Visualization to Visual Queries 1

1.1.1 Information Visualization and Scientific Visualization 2

1.1.2 Visual Queries 5

1.2 Problem Statement 6

1.3 Challenges and Goals 7

1.4 Contributions 8

1.5 Skeleton 10

Chapter 2 Background **12**

2.1 Terminology Definitions 12

2.2 Relational Data Visualization 13

2.2.1 Relational Data Model Visualization 13

2.2.2 Relational Data Mapping 14

2.2.3 Relational Data Cleaning Processes 16

2.2.4 Discussion 17

2.3 Multiple Dimensional Visualization 17

2.3.1 Parallel Coordinates 18

2.3.2 Scatterplots and Scatterplot Matrices 19

2.3.3 Star Coordinates 21

2.3.4 TableLens 22

2.3.5 Discussion 23

Chapter 3 A Framework of Visual Data Exploration **24**

3.1 Introduction 24

3.2 A Multiple-Visual-Context Framework of Data Exploration ... 26

3.2.1 Data-Model Context ... 26

3.2.2 Multiple-Dimension Context ... 27

3.2.3 Pairwise-Dimension Context ... 27

3.3 Discussion ... 27

Chapter 4 A New Interactive Visual Query for Relational Data Models ... 29

4.1 Revisiting Visual Queries for Relational Data ... 29

4.2 Data Model Visualization ... 34

4.2.1 Coordinating Context Views ... 34

4.2.2 Node-Link Graph Design for Relational Data Models ... 36

4.2.3 Data Mapping for Visualization ... 42

4.2.4 Discussion ... 47

4.3 Query Interaction ... 47

4.3.1 Interaction Model for Coordinating Context Views ... 49

4.3.2 Interaction Model for Node-Link-Based Queries ... 50

4.3.3 Incremental Data Exploration ... 56

4.3.4 Discussion ... 57

4.4 Visual Navigation Methods ... 58

4.4.1 Focus + Context ... 58

4.4.2 Zooming and Filtering ... 59

4.5 Query Implementation ... 60

4.5.1 A System Framework for Visual Queries with Coordinating Contexts ... 60

4.6 Summary ... 62

Chapter 5 New Interactive Visual Queries for Multi-Dimensional Data ... 63

5.1 Revisiting Manipulation on Multi-Dimensional Data ... 64

5.1.1 Parallel coordinate Interaction ... 64

5.1.2 Scatterplot Interaction ... 67

5.2 Quantitative Visualization with SumUp ... 69

5.2.1 Double Layer Views 71

5.2.2 Parallel Coordinate and Stacked Bar Integration 72

5.2.3 Data Mapping for Visualization 73

5.2.4 Discussion 75

5.3 Query Interaction on Parallel Coordinates with Quantitative Approach 75

5.3.1 Quantitative Visual Queries by Brushing 77

5.3.2 Discussion 81

5.4 Query Interaction on Scatterplots by FigAxis 82

5.4.1 Quantitative Visual Queries by Zooming 82

5.4.2 Data Mapping 88

5.5 Visual Enhancement Methods 88

5.5.1 Parallel Coordinate Scalability 88

5.5.2 Scatterplot Navigation 91

5.6 Query Implementation 94

5.6.1 A System Framework for Visual Queries with Quantitative Approach ... 94

5.7 Summary 96

Chapter 6 Case Studies 97

6.1 Case Study 1: Visual Data Exploration with Relational Models 97

6.1.1 Scenario 1 97

6.1.2 Scenario 2 99

6.1.3 Scenario 3 101

6.1.4 Scenario 4 104

6.1.5 Discussion 105

6.2 Case Study 2: Visual Analysis of Multiple-Dimensional Data 106

6.2.1 Multiple-Attribute Comparison 106

6.2.2 Correlative Analysis 108

6.2.3 Flexible Data Support 109

6.2.4 Discussion 110

6.3 Case Study 3: Interactive Data Exploration of Multiple Visual Contexts 110

6.3.1 In-depth Exploration on Multiple Dimensions and Pairwise Comparison111

6.3.2 Multiple-Context Queries of Data Models and Multi-Dimensional Data115

6.3.3 Discussion 120

6.4 Summary 121

Chapter 7 Evaluations 122

7.1 Space-Efficient Visualization 122

7.1.1 Dynamic Representation 122

7.1.2 Layering Display + Sharing Axes 125

7.2 Distinctive Features 127

7.2.1 Relational Query Making through Node-Link Graphics 127

7.2.2 Quantitative Query Making through Parallel Coordinates 128

7.2.3 Quantitative Query Making through Scatterplots 129

7.3 Friendliness of Techniques 129

7.3.1 Query Making with Node-Link Graphphics 129

7.3.2 Query Making with Parallel Coordinates 132

7.3.3 Query Making with Scatterplots 134

7.4 Discussion 136

Chapter 8 Extended work 137

8.1 Introduction 137

8.2 Review of Tag Visualization 138

8.3 Ranking Visualization Technique 139

8.3.1 Basic Design and Interaction 139

8.3.2 Grouped-Score Rankings 141

8.4 Visual Enhancement 142

8.5 Data Mapping 144

8.6 Case Study 144

8.6.1 Overall Contribution Rankings 144

8.6.2 Topic Contribution Rankings 147

8.7 Evaluation 148

8.7.1 Study Setup and Procedure 149

8.7.2 Result 149

8.8 Discussion 150

Chapter 9 Conclusion 151

9.1 Summary 151

9.2 Final Conclusion 152

List of Publications 154

Bibliography 155

List of Algorithms

Algorithm 4.1 The algorithm of setting relationships from the data schema information 43

Algorithm 4.2 The algorithm of checking a foreign key from the data schema information. 44

Algorithm 4.3 The algorithm of setting nodes from the data schema information. 45

Algorithm 4.4 The algorithm of the visualization mapping from the data of a query result. 47

Algorithm 5.1 The procedure of recording the interaction on the parallel coordinate dimensions. 80

Algorithm 5.2 The procedure of displaying the query results for the stacked bar interaction. 81

Algorithm 5.3 The procedure of recording the interaction on FigAxis.......... 87

Algorithm 6.1 Scenario completion procedures of MCquery and a data query tool.................. 105

List of Figures

Figure 1.1 The visualization of car crashing experiment 2

Figure 1.2 A tree map sample 3

Figure 1.3 The focus layer display in a biological study 4

Figure 1.4 The icon-based representations for a control query.......................6

Figure 2.1 A sample of the ER application 13

Figure 2.2 An entire employment of a simple ER diagram 14

Figure 2.3 The impact of the attributed visualization on node-link graphs ..15

Figure 2.4 The relational context visualization for entity resolution 16

Figure 2.5 The visual calculation of the y-coordinate of a basic data point (di) in parallel coordinates 18

Figure 2.6 An application of parallel coordinates with nine dimensions and around four hundred instances 19

Figure 2.7 The scatterplots with different shapes of data points..................20

Figure 2.8 The scatterplot matrix with 4 dimensions...................................21

Figure 2.9 The star coordinates with 8 dimensions......................................22

Figure 2.10 A sample layout of the TableLens employment 23

Figure 3.1 The proposed multiple-visual-context framework of relational data exploration...26

Figure 4.1 The relational data query interface of MS Access 2010 30

Figure 4.2 The relational data query interface of QGraph 31

Figure 4.3 The relational data query interface of Dataplay..........................31

Figure 4.4 The relational data query interface based on HV........................32

Figure 4.5 The relational data query interface based on a node-link graph ..33

Figure 4.6 The main user interface for the coordinating visual contexts of data models and query results by technique MCquery. 35

Figure 4.7 The relational schema of six tables Payments, Customers, Countries, Orders, OrderDetails, and Products used in the samples of this chapter. 37

Figure 4.8 A sample of the data model representation for six tables. 38

Figure 4.9 A sample of the result graph corresponding to three tables countries (a blue node), customers (an orange node), and products (the green nodes) 39

Figure 4.10 A query formulation model..48

Figure 4.11 The query interaction model proposed for the query module of Microsoft Access and MySQL..49

Figure 4.12 The new model for the data exploration by interaction on the coordinating visual contexts of data models and query results. 49

Figure 4.13 The instance of a query formulation with the finding component in the data model context 51

Figure 4.14 The instance of a query formulation with the condition component in the data model context..52

Figure 4.15 The instance of removing a link from a query..........................53

Figure 4.16 The filtering feature in the data model context.........................54

Figure 4.17 The instance of query interaction in the query result context....55

Figure 4.18 The logical-frame-based exploration of a huge graph 56

Figure 4.19 The proposed system framework for visual queries with the coordinating context views..........60

Figure 5.1 The direct manipulation with rectangle drawing..........64

Figure 5.2 The density based filtering in parallel coordinates..........65

Figure 5.3 A focus+context visualization model in parallel coordinates......66

Figure 5.4 A dimensional tree of visual hierarchical dimensional reduction66

Figure 5.5 The scatterplots with dynamic queries..........67

Figure 5.6 The scatterplots with data point selection optimization..........68

Figure 5.7 The scatterplots with display space transformation..........68

Figure 5.8 The bar and stacked bar layout with various arrangements for visual rankings..........70

Figure 5.9 The TableLens layout..........70

Figure 5.10 The main layout design of the SumUp user interface..........71

Figure 5.11 An instance of the SumUp query comparing the car models of three representatives Toyota of Japan (green), Volkswagen of Europe (orange), and Ford of USA (red)..........73

Figure 5.12 The data structure for considered polyline ranges..........74

Figure 5.13 The data matrix for stacked bar representation..........74

Figure 5.14 The parallel coordinates with box-plot embedded for data instance summary..........77

Figure 5.15 The interaction model of query interaction on the double layer views..........79

Figure 5.16 The linked views of scatterplots with various graphs..........82

Figure 5.17 The procedure of multiple-view matching between scatterplots and other graphics 83

Figure 5.18 The overview of the FigAxis layout design..........84

Figure 5.19 The layout of a FigAxis application for the correlative comparison of new car model delivery in term of Horsepower and Weight with the targets of original brands from USA, Europe, and Japan..........85

Figure 5.20 The zooming level measurement background 86

Figure 5.21 The filtering feature of SumUp applied in the visual analysis of Census income data concerning Income, Hourperweek, Age, and Sex towards Occupation 90

Figure 5.22 The zooming and panning feature of FigAxis in the proximity navigation..........91

Figure 5.23 The colour-based highlight of the car models delivered by USA (the green plotted points and the green stacked bars). 92

Figure 5.24 The extended FigAxis layout of the visual comparison of the car model delivery in term of Year and Cylinder 93

Figure 5.25 The system framework proposed for the visual query deployment of SumUp 95

Figure 6.1 The query interaction on the relationship representations of store, staff, city, and country..........98

Figure 6.2 The flexible interaction on the coordinating visual contexts of the data model and query results in Scenario 2..........100

Figure 6.3 The filtering feature applied for countries China and India..........101

Figure 6.4 The query interaction with the highlighted dimensions of film, category, and actor 103

Figure 6.5 The relationship recognition in the query result of categories Children and Comedy..104

Figure 6.6 The multiple attribute comparison for the number of models of six Japanese brands based on Cylinder, Horsepower, Weight, and Year 107

Figure 6.7 The correlative analyses of Weight and MPG, Weight and Horsepower, and Weight and Displacement. 108

Figure 6.8 The data summary with flexible data support to explore Income and Workclass towards the ages of the population in United States...........109

Figure 6.9 The statistical parallel coordinates of the United States Census income data towards Sex and Education..110

Figure 6.10 The leve-1 summary of Age and 40-and-over Hoursperweek.112

Figure 6.11 The leve-6 summary of Age and 40-and-over Hoursperweek.113

Figure 6.12 The quantitative plotting comparison between Occupation and 40-50 HoursPerWeek 114

Figure 6.13 The data-model context of customer, film, category, actor, and store 115

Figure 6.14 the multi-dimension context of categoryname, filmreleaseyear, and storeaddress 116

Figure 6.15 The pairwise comparison of categoryname and filmreplacementcost 118

Figure 6.16 The name of the impacted films of Drama and Family in the comparison of costs 10.99 and 24.99 visualized in the data-model context of MCquery..120

Figure 7.1 The space-saving rates in term of the link display of MCquery 124

Figure 7.2 The space-saving rates in term of the node display of MCquery125

Figure 7.3 The detailed comparison between the IT and BA groups on the MCquery feedbacks...130

Figure 7.4 The task completion time of the MCquery usage for the IT and BA groups 131

Figure 7.5 The feedback result of the friendliness comparison between the FigAxis layout and the multi-view layout...135

Figure 8.1 The basic layout design of Qstack 140

Figure 8.2 An instance of the grouped-score visual ranking......................141

Figure 8.3 A stacked bar chart with multi-tag filtering..............................143

Figure 8.4 The scaling components of Qstack 143

Figure 8.5 The overall contribution ranking for price A............................145

Figure 8.6 The overall contribution ranking for price B 146

Figure 8.7 The sorting view by tag park (red bars) and tag flower (green bars) with multi-baselines and flexible scales..148

Figure 8.8 The brushing across categories 4, 5, and 6 148

List of Tables

Table 4.1 The data of the query result visualized in Figure 4.9. 40

Table 4.2 The 10-categorical colour scheme of d3js library 41

Table 4.3 The 20-categorical colour scheme of d3js library 42

Table 5.1 The FigAxis data support summary 88

Table 7.1 The parameter values chosen for the space-saving assessment in term of the node display of MCquery 124

Table 7.2. The feature comparison of MCquery, the visual query tool of Microsoft Access, and Ploceus 127

Table 7.3 The feature comparison of SumUp, the tool of Siirtola (2002), and Ho et al. (2011) 128

Abstract

The direct data manipulation through visualization and associated navigation techniques has been implemented for many years. However, these methods are not uniformly discussed in the context of user interface design. During the history of user interface development, the interaction between humans and computers is almost to be done through software widgets. Since in the last decade, many advanced data visualization and interaction techniques have been developed, now it is the time to bring them into the formal discussion about the context of user interface design, data queries, and data manipulation. The dissertation attempts to fulfill the gap between visual user interface design and interactive data visualization.

In relational data queries, many visualization techniques have featured advanced interactive operation; however, a majority of those would concentrate on the traditional style, instead of a modern approach. This is the reason why today in visual analytics truly direct manipulation is highly encouraged, instead of the conventional methods.

This dissertation focuses on the investigation of modern data query approaches. It attempts to model the new data query methods that apply those advanced visualization and interaction techniques to facilitate the data analysis procedures. The second contribution of the dissertation is the design of new interaction methods for multi-dimensional data visualization.

We first introduce a new framework which includes straightforward manipulation techniques for relational data discovery. These novel techniques, named *MCquery*, *SumUp,* and *FigAxis*, are exclusively developed for the key characteristics of relational data such as *data models* and *data dimensions*. The core methodology is about interactive visual query design based upon node-link graphics, parallel coordinate geometries, and scatterplot visualization, where the direct interaction is performed by friendly action such as clicks and brushes. The tools materialized from these techniques can help to reduce users' cognitive and behavioral effort efficiently in dealing with the issues of information search-retrieval, quantitative data analysis, and correlation examination.

Chapter 1 Introduction

We all, human, have been living in a world of information, the world of variety and complexity. Thanks to the achievements of science and technology, especially computer development, information is collected, stored, processed, etc. in order to serve human needs, which is increasing quickly and steadily. Of those, how to support people in learning and mining information through the process of human-computer interaction is a critical topic and a vital challenge in computer science. One of the most concerned solutions is about making information items to become graphical elements to be visible and interactive by software programs and user behaviours. In other words, this action is called Visualization. In point of fact, the role of Visualization has become not trivial in modern cognitive systems due to its information transferring characteristics to human senses, as people sometimes said

"The eyes…the window of the soul"

(Leonardo 1452 – 1519)

and

"Use a picture. It's worth a thousand words"

(Speakers Give Sound Advice 1911).

1.1 From Information Visualization to Visual Queries

There are heterogeneous definitions of Visualization; however, a classical one proposed by Card, Mackinlay and Shneiderman (1999) -

"The use of computer-supported, interactive visual representations of abstract, non-physically based data to amplify cognition"-

is very popularly accepted. It simply means that Visualization is to use computer graphics to acquire knowledge and understanding. In actuality, visualization activities are able to bring various advantages to both front-end users and back-end developers such as supporting to understand an enormous amount of data, enabling perception of unanticipated elements, allowing quick displaying of the problems associated with data, etc. (Ware 2012). The other definitions and descriptions of Visualization can also be found in Little et al. 1972. A sample about the visualization of car crashing experiment is shown in Figure 1.1.

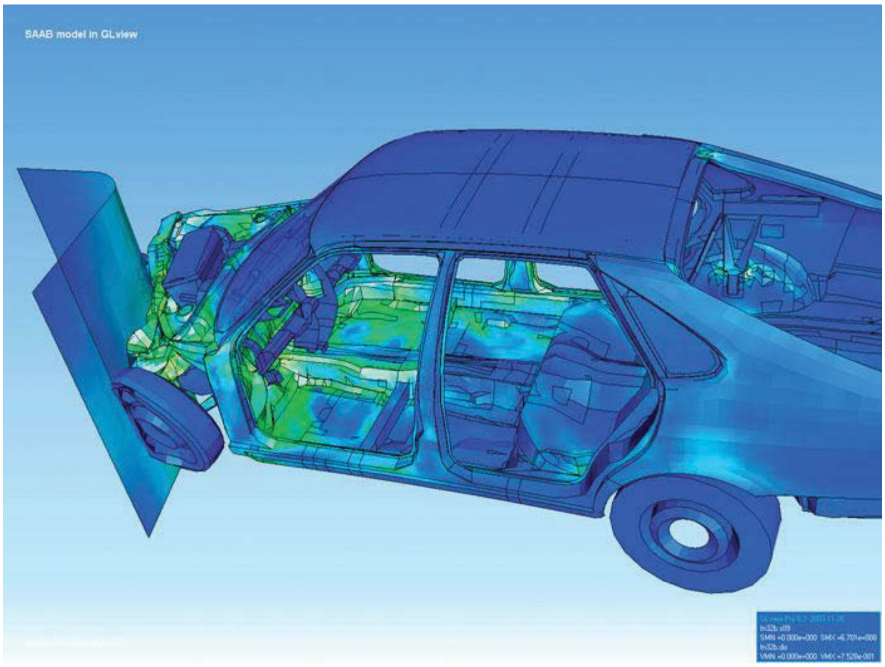

SAAB model in GLview
SMN =0.000e+000 SMX =6.701e+008
VMN =0.000e+000 VMX =7.529e-001

Figure 1.1 The visualization of car crashing experiment. (Source: https://en.wikipedia.org/wiki/Visualization_(graphics) 2017)

The definition by Card (1999) was, additionally, offered to describe term Information Visualization at that time. However, nowadays, the widely-known concept of Visualization refers to two sub branches including Information Visualization and Scientific Visualization. Although in some areas the border of these two fields seems to be not very clear, on numerous occasions they still possess the typical characteristics of their own.

1.1.1 Information Visualization and Scientific Visualization

Term Information Visualization was first used by Robertson, Card, and Mackinlay in 1989 (Robertson, Card & Mackinlay 1989). Ten years later, international conferences on the topic had officially started such as *IEEE Visualisation*, *CHI'XX*, and *UIST'XX* conferences, and many well-known Information Visualization articles were also delivered such as *Worlds within worlds* (Feiner & Beshers 1990), *Tree-maps* (Johnson & Shneiderman 1991), *Information visualizer* (Card, Robertson & Mackinlay 1991), etc.

Information visualization techniques aim to reduce complexity in acquiring knowledge and analyse information on computer applications by using visual representation and interactive behaviours on the abstract data which do not have explicit spatial references and do not have natural mapping features such as textual, tabular, or hierarchical data. The mentioned data often comprise a high number of dimensions and a large number of attributes, which the standard two-dimensional (2D) model with axis X and axis Y is not able to represent adequately. For dealing with such challenges, novel visualization techniques such as Parallel Coordinates (Inselberg & Dimsdale 1991), Treemaps (Shneiderman 1992) (see Figure 1.2), and hierarchical-graph-based techniques were proposed respectively. Besides, with the rapid development of big data employment and mining, information visualization often requires automated analysis techniques such as classification and clustering in data preparation at pre-processing stages for handling a huge volume of data. Today, due to the critical benefits of information visualization, it widely appears in many fields and areas such as software applications in education, business, government, entertainment, etc.

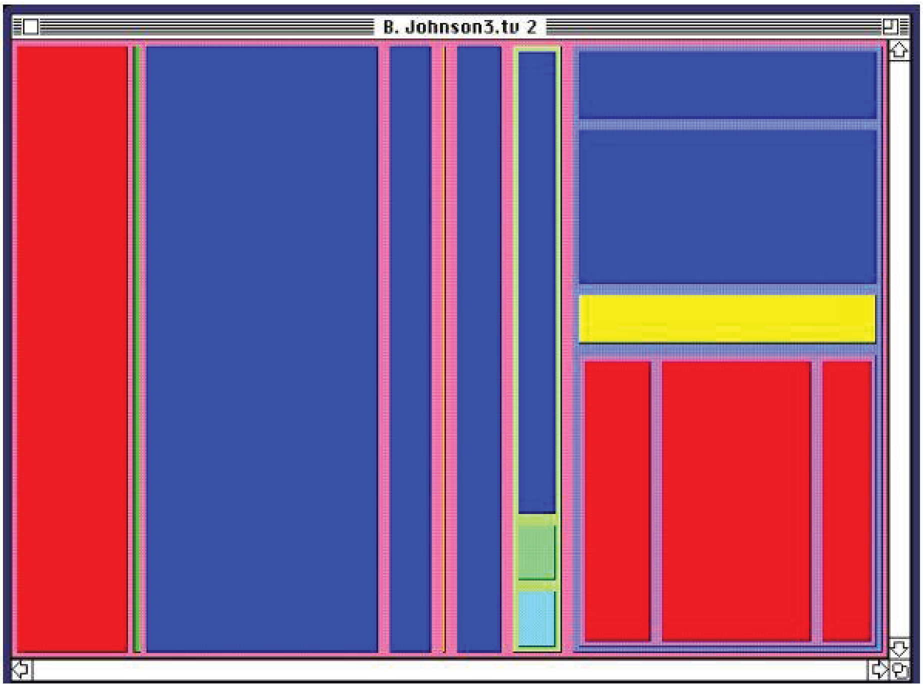


Figure 1.2 A tree map sample (Shneiderman 1992).

On the other hand, scientific visualization techniques center on visualizing complex structures of real objects in the physical world for scientific study, typically in three-dimensional (3D) geometries, and on being explicit references to time and space such as the objects of medicine, biology, geography, etc. Widespread approaches were, typically, based on direct volume rendering such as iso-surfaces (Engel et al. 2004), Glyph (Forsell, Seipel & Lind 2005), flow visualization (Merzkirch 1987), etc. Due to the complexity of such displaying, the deployment of those was commonly integrated with interaction techniques, especially focus+context technique, to enhance navigation effectiveness in 3D-based representation (Kruger, Schneider & Westermann 2008) (see Figure 1.3).

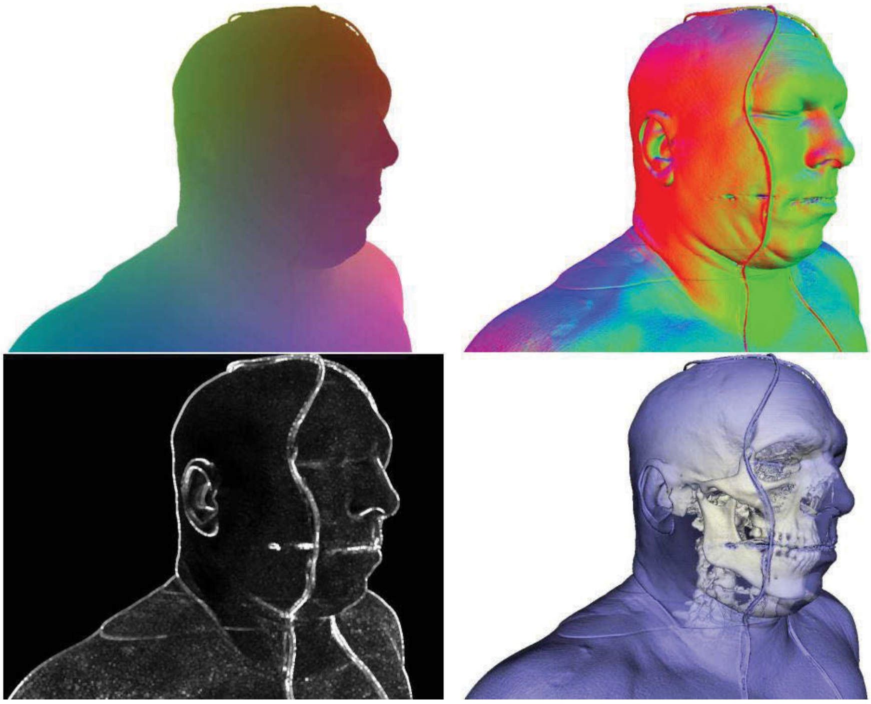


Figure 1.3 The focus layer display in a biological study (Kruger, Schneider & Westermann 2008).

Nowadays, the contributions of visualization are becoming more famous in many fields for its significant benefits, which was admired by many works such as

"The success of information visualization is often an interplay between an expert's meta-knowledge and knowledge of other sources as well as information from the visualization in use." (Carpendale 2008),

"The ultimate goal of interactive visualization design is to optimize applications so that they help us perform cognitive work more efficiently.", and

"*The_user_ benefits_from_ visualization =*

*the_ cognitive_ work_ done * the_ value_ of_ the_ work.*" (Ware 2012).

1.1.2 Visual Queries

In addition to the benefits mentioned above, another important contribution which visualization supplies to data mining through the activities of information search and retrieval is the visual support for querying. In the past, when visualization had not developed, to interact and explore data there was almost only the method of using text-based commands as the query languages. However, because of the complication of such traditional query languages, the method was appropriate for experienced users, but not recommended for novices and general users. At the present time, query-supporting visualization has primarily changed the conventional way of data exploration by the appearance of term Visual Query.

The definitions of term Visual Query can be found in Angelaccio, Catarci and Santucci 1990; Ware 2005; Caschera, D'Ulizia and Tininini 2008, etc. In most of the cases, Visual Query was widely understood as the queries that are made by using visual elements or objects on computer interfaces for data and information discovery. By this way, instead of using text-based queries, users would use computer graphics such as forms, tabular objects, diagrammatic objects, iconic objects, etc. (see Figure 1.4) to create the requests of search, retrieval, and analysis of desired information and data. Making queries by using visual objects is evaluated to be more accessible to general users and novices since it supports them in reducing the mental effort in query performance without the requirement of handling text-based commands. Moreover, this approach is assessed to be friendlier in term of intelligent interaction interface than the traditional one. As a result of its exclusive strengths, recently, the number of visual query applications has been increasing day by day, and sometimes people said,

"Most of the visual queries we make of the world seem literally effortless, so much so that we are not even aware that we are making them." (Ware 2005).

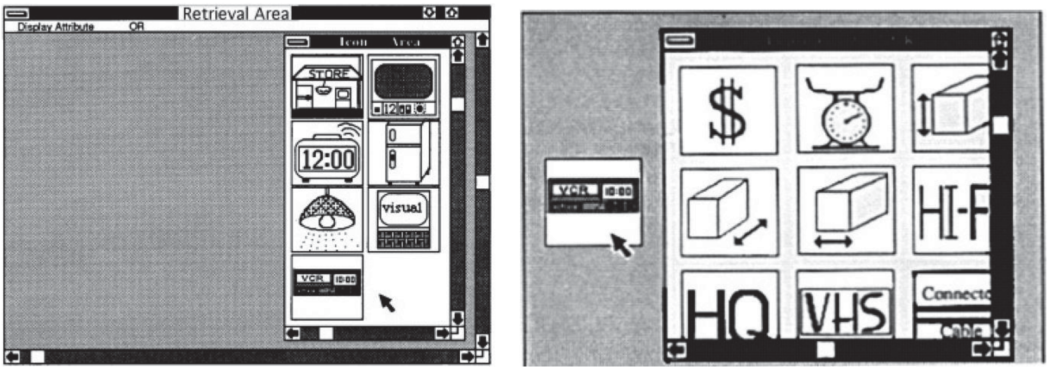


Figure 1.4 The icon-based representations for a control query (Tsuda et al. 1990).

1.2 Problem Statement

The direct data manipulation in visualization has been considered insufficiently. During the history of user interface development, the interaction between humans and computers is almost to be done through software widgets. The traditional and common widgets in general software are about the form or table layouts with the columns and rows of data and commands, which are widely deployed in visualization tools as well. Therefore, this event incidentally makes people think that interactive data visualization is just the simple interaction performed by playing with text fields, data cells, buttons, etc. Although these tasks are theoretically basic for most of interaction procedure, in order to explore data visually and coherently, using such functions only is not sufficient empirically. For example, *clicking a button* is to activate or run a function rather than to manipulate and interact with visual objects directly. In other words, such conventional interaction is often operated separately from representation objects, which can result in the increase of work load in information transformation between functional components and graphical elements. For visual queries of relational data, many visualization techniques have featured advanced interactive operation; however, a majority of those would concentrate on the traditional style, instead of a modern approach. This event can support the reason why today in visual analytics straightforward and direct manipulation is significantly recommended and encouraged.

The aim of this research is to improve the direct-interaction mechanism of relational-data visual queries such as the exploration based upon data models and data dimensions. The existing visual query approaches that are currently employed in relational data visualization have not greatly supported direct manipulation on the characteristics of such data, particularly for data models and data dimensions. In fact, relational data representation is usually formatted in tabular or table based layouts, which can be easily interactive with text-based queries. These queries always require expert knowledge or much technical understanding in practical usage, which is a not small barrier to novices and general users. Additionally, although existing visual analytics suites offer various interactive controls for data exploration, they also require the technical configuration during using procedures that would seem only beneficial for well-trained or senior users.

Without direct and friendly manipulation, the visual data exploration tools would be hard to be used by ordinary users, who are securing the majority of target customers of information technology services.

1.3 Challenges and Goals

The widespread deployment of visual data solutions was one of the most interesting challenges (Keim et. al. 2006). As a matter of fact, there are a large number of visual analytics techniques and tools, but not many people can access them. Particularly, relational data can be found easily, yet general users cannot mine and explore them, due mainly to lack of right visual query tools. Here, *right tools* refer *easy-to-use tools*, which could not be the tools to be exclusive for experts definitely.

The overall goals of this dissertation are:

- To investigate novel visual query models that can effectively merge the latest data visualization, interaction, and former data query requirements into one framework which could friendly support analytical reasoning processes,
- To investigate novel interaction techniques that can support information search and retrieval in multi-dimensional data visualization,

- To build relational data exploration tools that materialize and synchronize developed visual queries in term of space efficiency, novelty, and friendliness,
- To apply analytical reasoning processes into our proposed interactive visual query designs in order to evaluate the effectiveness and efficiency of new approaches.

1.4 Contributions

In general, the contributions of this dissertation are listed below, but not limited to:

1. A new framework for synchronous and continuous exploration through multiple visual contexts of relational data (Chapter 3).

Visual analytics play a fundamental role in bringing insights to the audiences who are interested and dedicated to data exploration. In the area of relational data, not a small number of advanced visualization tools and frameworks are proposed in order to deal with such data features. However, a majority of those have not significantly considered the whole process from data-model mining to dimension-based querying, which might enhance the flexibility of continuous and further exploration. This chapter presents a new interactive exploration framework for relational data analysis through the automatic interconnection of data models and data dimensions. The approach is to enable users to make relative queries flexibly on desired contexts at any stage of exploration for in-depth data understanding.

2. A new interactive visual query technique named MCquery and based upon data model representation for information search and retrieval (Chapter 4).

The use of visual queries for relational data search and retrieval is a well-known approach, and various advanced visualization methods have been proposed to improve the quality of such queries. Nevertheless, most current methods pay attention to constructing queries in a single visualization context, which causes the isolation of data models and query results in exploration procedures. Consequently, the cognitive effort of users for query formulation and adjustment is increased, significantly in visual matching the query details with their contexts. How to visually and synchronously represent all data models, query formulation, and query results in a single screen of multi-contexts is a concerned challenge in order to make great convenience for data exploration.

This chapter presents a novel visual method called MCquery for dealing with the above challenge. This method allows query specification in the coordinating visual contexts of data models and query results by interaction on node-link graphs of relational data representation. MCquery enables relational data to be analysed with relative retrieval from the incremental exploration of data models, queries, and query results.

3. A new interactive visual technique for quantitative queries called SumUp and based upon multi-dimensional visualization, particularly in parallel coordinate geometries (Chapter 5).

Parallel coordinates and scatterplots are the visual techniques widely used for representing multivariate and multi-dimensional data. While parallel coordinates is well suited to provide the general display of a vast number of attribute values given by a large number of dimensions, scatterplots is a right choice in the detailed comparison of pairs of dimensions. One of the most considered issues in this area is how to quantitatively analyze the data density caused by polyline growth in the limited space of the parallel coordinate axes and resulted by dot increasing in the plotting space of scatterplots. The existing visualization techniques can successfully support the comparison of data summary among single dimensions and independent focus zones. Nevertheless, these might face challenges in complex analytics, which require the flexibility and multiplicity of such comparison.

This chapter introduces two new visual query techniques named SumUp, for the statistical analysis of the multiple attributes of dimensions with the multiple ranges of the polylines of parallel coordinates, and FigAxis, for the quantitative analysis of the focus zones of scatterplots. The methodology of SumUp is primarily based on developing dynamic queries by using brushing operation to deliver summary stacked bars on the parallel coordinate axes. Users can quickly retrieve quantitative information from the data patterns and flexibly make the multiple-attribute comparison with minimum effects of the polyline density. SumUp is able to enhance direct manipulation on parallel coordinates in term of statistical discovery and scalability. In FigAxis, zooming operation is adapted for data plot density measurement with dynamic stacked bars on the scatterplot axes. Users are enabled to quantitatively explore plotted data instances in the same space and at the same time of correlation observing.

Other additional contributions include:

4. An advanced scatterplot method named FigAxis for statistical data analytics (Chapter 5), and

5. A new dynamic query system for visual rankings (Chapter 8).

1.5 Skeleton

This dissertation is organized as follows:

- Chapter 2 Background: This chapter introduces an overview of basic knowledge about the visualization of relational data models and data dimensions.
- Chapter 3 A Framework of Visual Data Exploration: This chapter presents a new interactive exploration framework for relational data analysis through the automatic interconnection of data models, multiple dimensions and pairwise dimensions.
- Chapter 4 A New Interactive Visual Query for Relational Data Models: This chapter presents a novel visual method, called MCquery, for dealing with the challenge of how to visually and synchronously represent all data models, query formulation, and query results in a single screen.
- Chapter 5 New Interactive Visual Queries for Data Dimensions: This chapter introduces two new visual query techniques, including SumUp for statistical analysis of multiple attributes of dimensions and ranges of polylines in parallel coordinates and FigAxis for quantitative analysis of focus zones in scatterplots.
- Chapter 6 Case Studies: This chapter illustrates and discusses typical case studies for demonstrating the implementation and usefulness of our interactive visual

queries, which is introduced and synthesized in Chapter 3, Chapter 4, and Chapter 5.

- Chapter 7 Evaluations: This chapter illustrates the procedures and results of evaluations for our techniques in term of space-efficient visualization, distinctive features, and friendly techniques.
- Chapter 8 Extended Work: This chapter presents a dynamic and interactive query method which is extended from the previous query techniques. Although this additional technique is not a primary contribution to the goals of this dissertation, its theory and practice are closely related to the employment of visual queries for data exploration.
- Chapter 9 Conclusion: This chapter concludes this dissertation.

Chapter 2 Background

This chapter provides some terminology definitions and an overview of basic knowledge about visualization of relational data models and data dimensions, which are relatively scoped in this dissertation. We classify the background into relational data visualization and multi-dimensional visualization as the details below.

2.1 Terminology Definitions

We is kindly referred to the author, himself, and the parties who have significantly contributed to the completion and success of the studies of this dissertation.

Data model is referred to as a conceptual description about data organization, and in this dissertation, we only focus on the model of **Entity-Relationship**.

Entity-Relationship model is a name of an abstract data model, which considers objects and their connections in real life as entities and relationships in theory for data collection, organization, and mining.

Relational data is referred to data which are organized in a relational schema and stored in a relational database.

Dimension is referred to a data vector, commonly meaning an attribute or a variable of data in related scientific areas.

Multi-dimensional data is referred to a data collection which is organized, structured, and stored with many dimensions or a large number of dimensions.

Data instance is referred to as a record or a row of a data table.

G = {V, E} is a graph, defined as a pair (V, E), where *V* is a set of *vertices*, and *E* is a set of *edges* between the *vertices* $E = \{(u, v) \mid u, v \in V\}$. In this dissertation, *nodes* and *links* are preferred and used as the visual representation of *vertices* and *edges*.

2.2 Relational Data Visualization

Visualization and relational data are inter-cooperated in various angles. On account of the objectives of this dissertation, we focus on revisiting relational data visualization in term of data model visualization, data mapping, and data cleaning processes. These are the significant aspects of not only the role of visualization in such data but also the core preparation stages for effective data visualization.

2.2.1 Relational Data Model Visualization

Ordinarily, all data types are compulsory to be formatted in a particular model for usage, and relational data is unexceptional. Relational data are well-known with the model named Entity-Relationship (ER) and firstly proposed by Chen (1976). ER model enables mapping real world objects into conceptual diagrams that can serve information system analysis and development. Besides of mathematic norms, graphical approach defines the characteristic of ER methodology. A diagram of ER model is the meaningful connection of rectangles, lozenges, lines, arrows, etc., where rectangles and lozenges represent real-life objects and their relations. For example, to express a real context of that employees work in projects, the corresponding ER diagram is drawn as Figure 2.1, and Figure 2.2 illustrates a sample of an entire simple ER diagram.

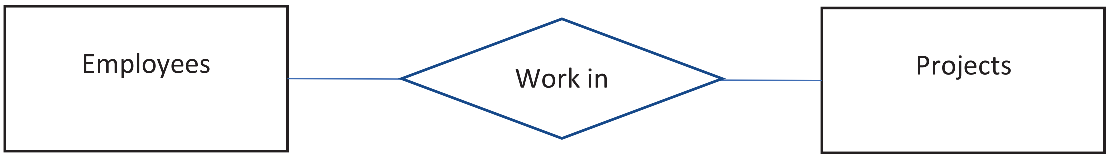


Figure 2.1 A sample of the ER application. This conceptual diagram illustrates the relationship *"Employees work in projects"* in real life.

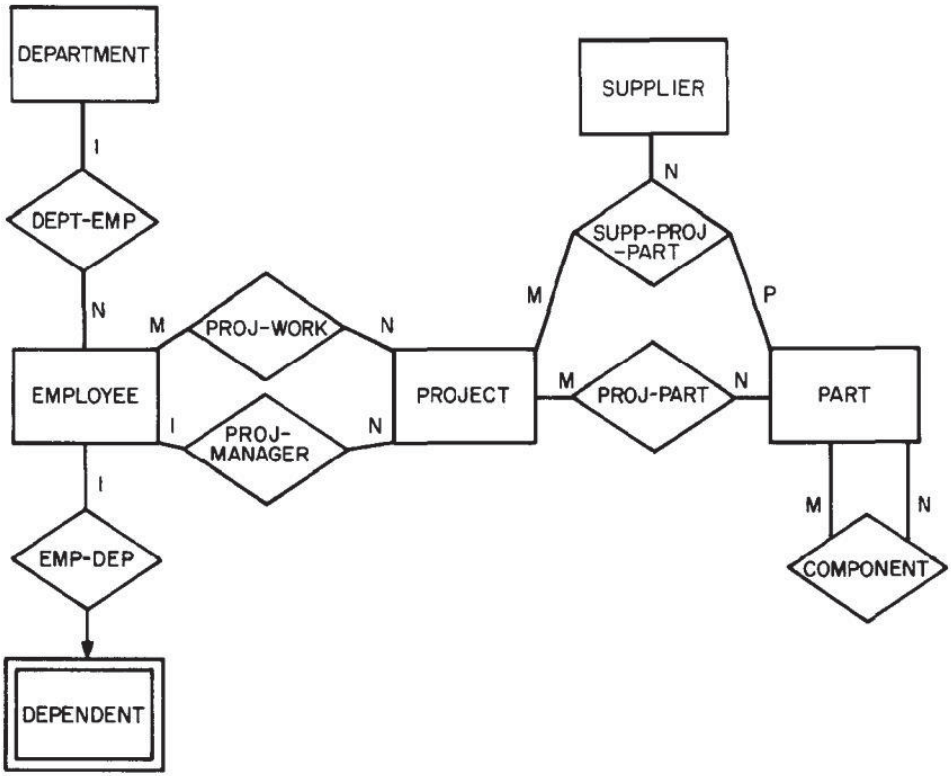
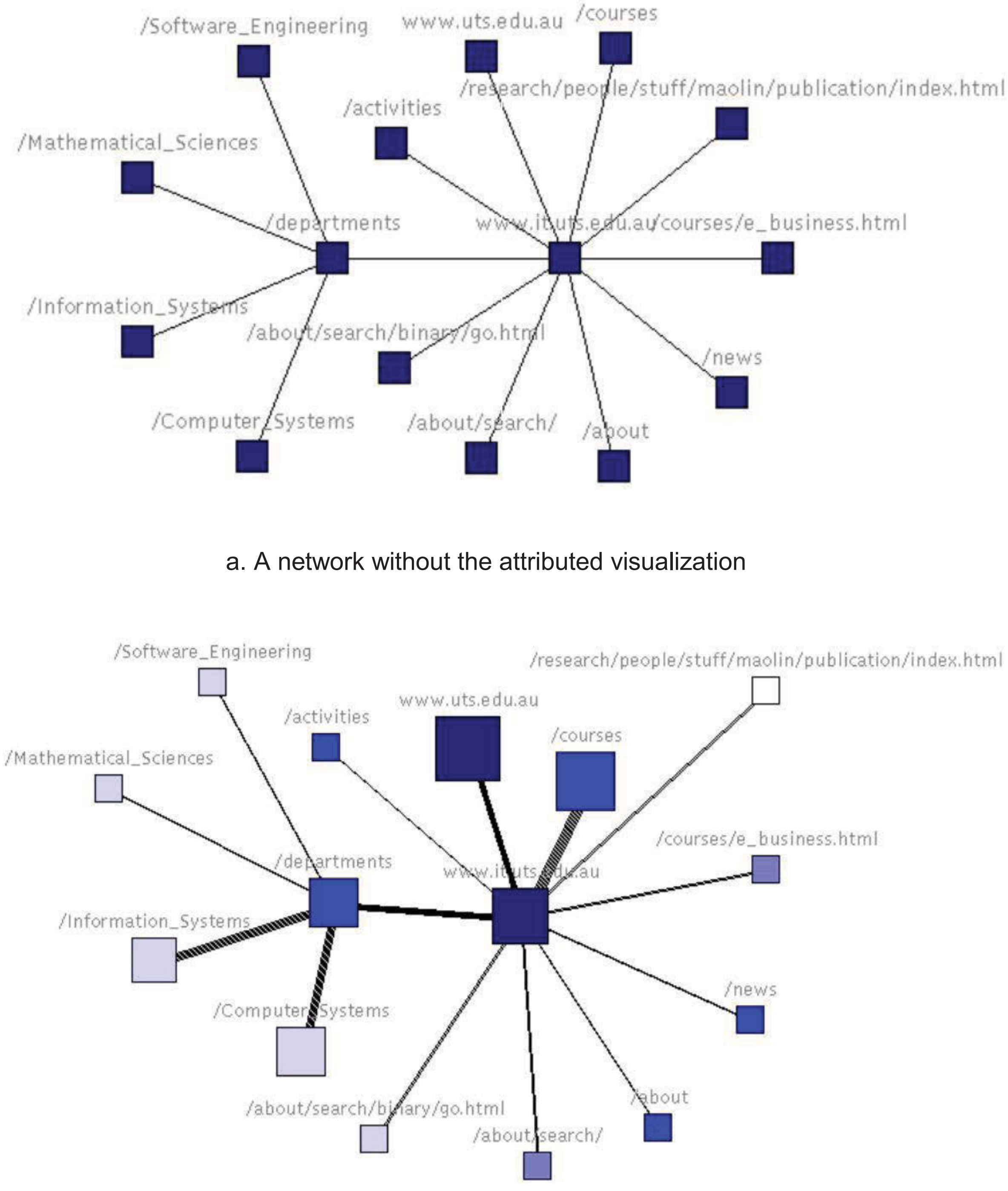


Figure 2.2 An entire employment of a simple ER diagram (Chen 1976).

Nowadays, Entity-Relationship diagram visualization can be found in many commercial modeling systems such as Rational Rose (IBM, http://www-03.ibm.com/software/products/en/ratirosefami) and PowerDesigner (Sybase, http://www.sybase.com/products/modelingdevelopment/powerdesigner).

2.2.2 Relational Data Mapping

In visualization, data mapping is a mandatory procedure for converting data attributes to visual element attributes. For reviewing attributes associated with relational data, Huang (2001) introduced an attributed visualization model through geometric and graphical mapping. The main idea includes the procedures of transforming conceptual relational structure to node-link structure and transforming data object attributes to graphical attributes, which is a basic guide for relational structure data visualization. Later, Dastani (2002) proposed structure-preserving mapping for effective visualization of relational data with the idea *"The intended structure of data should coincide with the perceptual* *structure of its visualization”*. Figure 2.3 illustrates an application of attributed visualization.



a. A network without the attributed visualization

b. The network applying the attributed visualization

Figure 2.3 The impact of the attributed visualization on node-link graphs (Huang 2001).

2.2.3 Relational Data Cleaning Processes

In information visualization mechanism, before data mapping, the given data, normally, need to be under noise removal for clean and precise data pattern of further visual representation. For node-link graph application, dealing with entity resolution is always a greatly considered topic. Entity resolution helps to unify redundant references in order to make entity objects distinct and identified. Kang (2008) proposed a typical entity resolution technique based on relational context visualization. Duplicated references are displayed, clustered, and highlighted in a relational network that allows users to make a decision on the cleaning process visually and coherently. In addition to these benefits, the technique employment might bring relational-data users the concept of node-link graphs through graphical interaction activities. Figure 2.4 illustrates a use case of the technique.

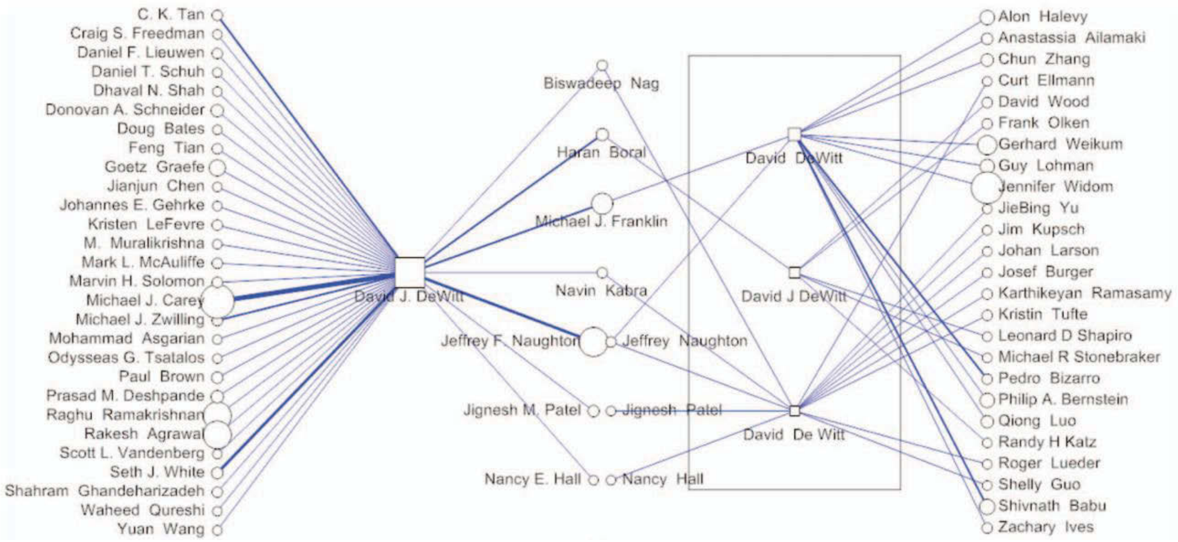


Figure 2.4 The relational context visualization for entity resolution (Kang 2008).

2.2.4 Discussion

Relational data are one of the most popular data types, and they can be dealt by various visualization techniques. The mentioned-above background is only the principles which are the closest to our methodology based on node-link graphics. Moreover, these works play a not trivial role in the instruction of our further design about relational visual queries in this dissertation.

Currently, the node-link graph analytic features for relational data can be found in many software libraries such as Prefuse (Heer, Card & Landay 2005), Orion (Deer & Perer 2011), and Tulip 3 (Auber et al. 2012).

2.3 Multiple Dimensional Visualization

Multivariate and multi-dimensional data analytics are critically essential activities for retrieving and understanding a large number of complicated information in term of types and contents. While the variety of data types is resulted by different information-generating sources, the complexity of contents is along with the increase of data dimensions and data instances. Once organized and structured, such data almost always require distinctive techniques and equipment for dealing with high-dimensional challenges.

According to Keim and Kriegel (1996) and Keim (2002), the taxonomies of visualization techniques could be categorized into pixel oriented, geometric projection, icon based, hierarchical based, and graph based techniques. In respect of the primary methodology applied throughout this dissertation, we concentrate on revisiting the geometric projection based techniques, including parallel coordinates, scatterplots, scatterplot matrices, star coordinates, and TableLens.

2.3.1 Parallel Coordinates

Parallel coordinates, originally proposed by d'Ocagne (1885) and Inselberg (1985), is a technique based on geometric projection. Its layout is drawn by combining crossing polylines and parallel vertical axes, where the polylines encode records or instances denoted as $P_i = \{d_1, d_2, d_3, \ldots, d_n\}$, and the axes encode dimensions denoted as $X = \{x_1, x_2, x_3, \ldots, x_n\}$. A data point d_i of P_i is mapped to a display space as the following formula.

$$y_{di} = y_{max\_X} + \frac{max\_X - d_i}{max\_X - min\_X} H$$

where y_{di}, y_{max_X}, max_X, min_X, and H are the *y-coordinates* of d_i and max_X, maximum value of X, minimum value of X, and the height of the display space. Figure 2.5 illustrates the detail of visual calculation, and Figure 2.6 is an application of parallel coordinates with nine dimensions and around four hundred instances.

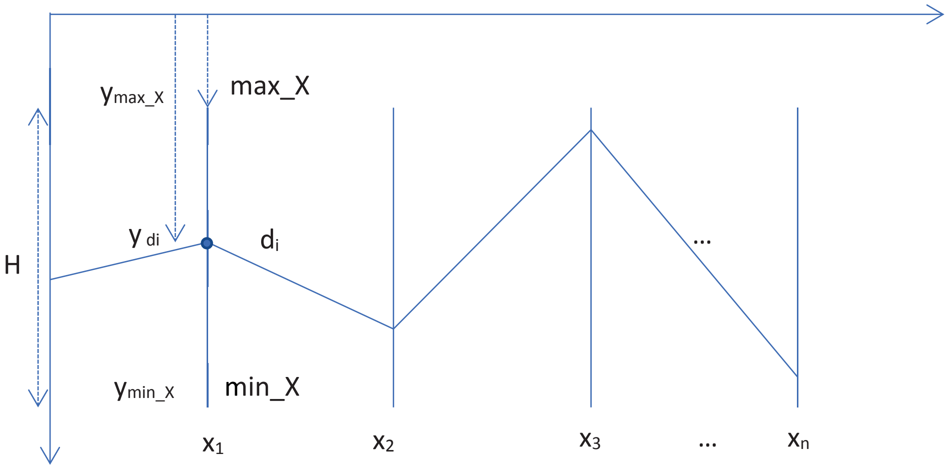


Figure 2.5 The visual calculation of the y-coordinate of a data point (d_i) in parallel coordinates.

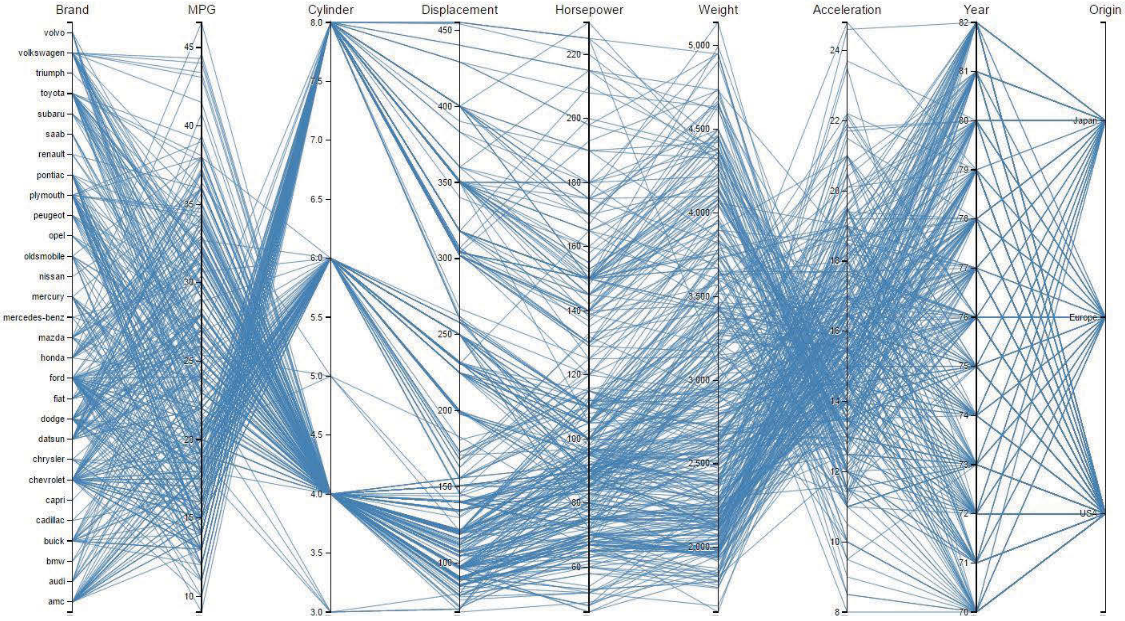


Figure 2.6 An application of parallel coordinates with nine dimensions and around four hundred instances.

2.3.2 Scatterplots and Scatterplot Matrices

For data with high dimensions, it is not always all of them to be considered simultaneously. In that case, scatterplots, also a geometric projection technique, is an appropriate option to show the data pattern of a couple of variables (Keim & Kriegel 1996). Scatterplots visually represents the correlation between two variables by plotting data instances as data points between a vertical axis Y and a horizontal one X. The data instances plotted by scatterplots might be encoded with shape, size, and color in order to enhance the number of represented dimensions. Figure 2.7 illustrates a sample of a standard scatterplot application.

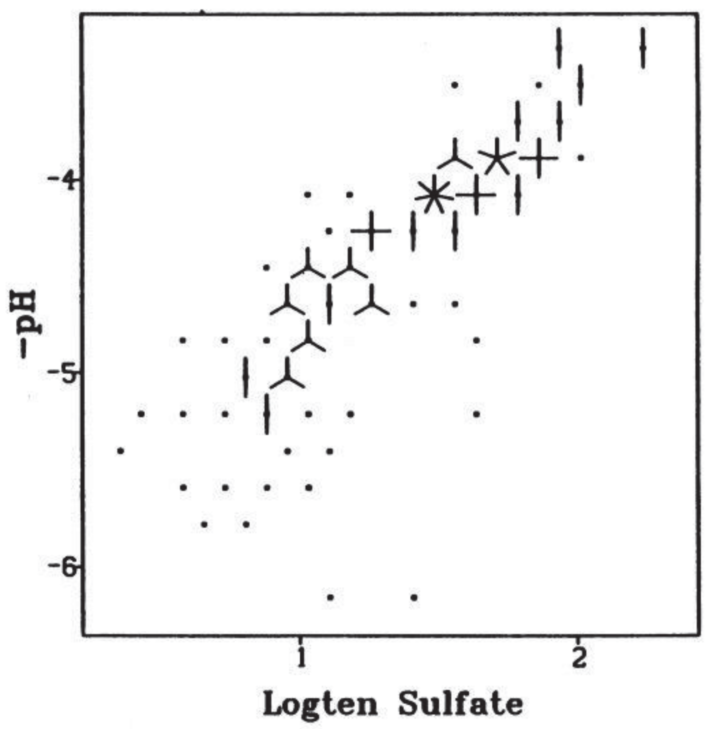


Figure 2.7 The scatterplots with different shapes of data points (Carr et al. 1987).

For dealing with correlation analysis of many variables concurrently, an extension of conventional scatterplots was proposed and called Scatterplot Matrix, which is currently a well-known visual technique in statistic science (Carr et al. 1987; Friendly & Dennis 2005). The basic idea of a scatterplot matrix is to visually represent the pairwise comparison of *n* dimensions by a super panel of *n x n* sub plotting panels. Figure 2.8 illustrates an application of a scatterplot matrix with *4* dimensions.

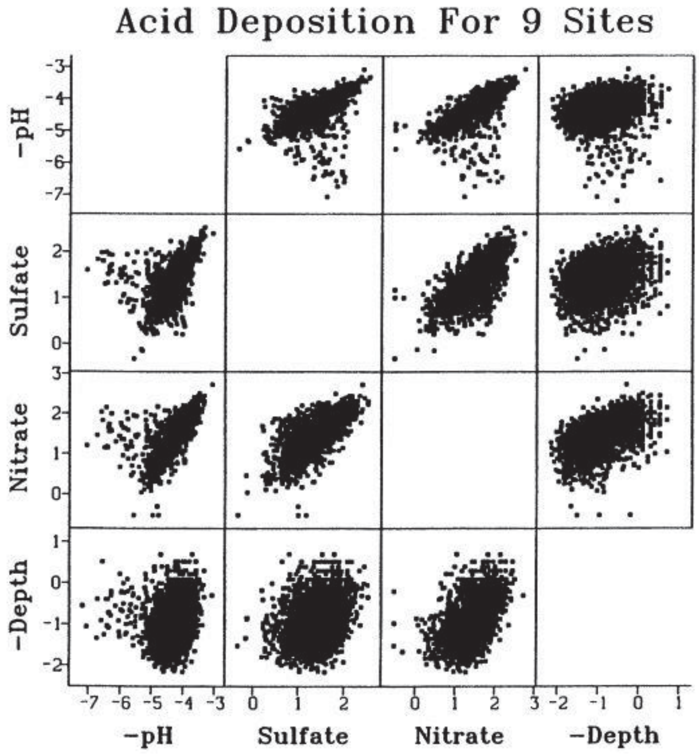


Figure 2.8 The scatterplot matrix with 4 dimensions (Carr et al. 1987).

The display space optimization is a common challenge of this area. Because of the duplicating all dimensions twice, the effectiveness of using the display space is not high. In reality, there is more than a half of plotting panels which are repeated and might be omitted or used for other purposes. The minimal number of the plots to be essential for all pairwise comparisons of *n* dimensions is just *n(n-1)/2*.

2.3.3 Star Coordinates

Star coordinates is another visual technique extended from 2D and 3D scatterplots in order to enhance the number of dimensions encoded in a single view by plotting data points between circling-around-same-origin axes in 2D space (Kandogan 2000) (see Figure 2.9). Although star coordinates can improve the display space optimization, clutter reduction and layout familiarization might be the challenges in the case of the highly increased density of data points.

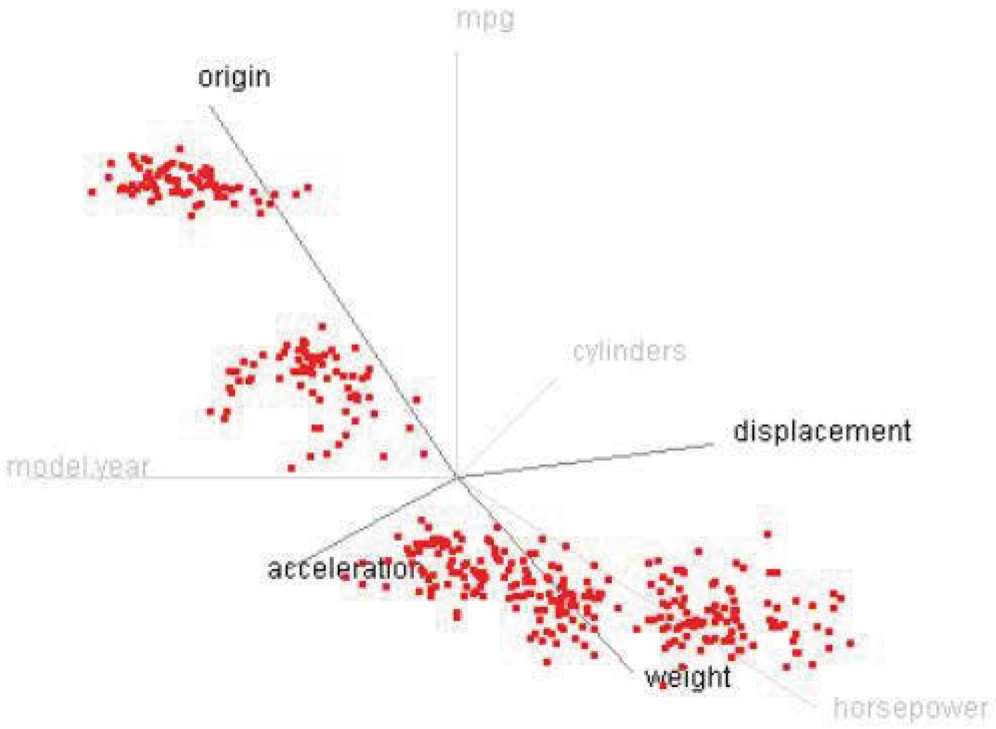
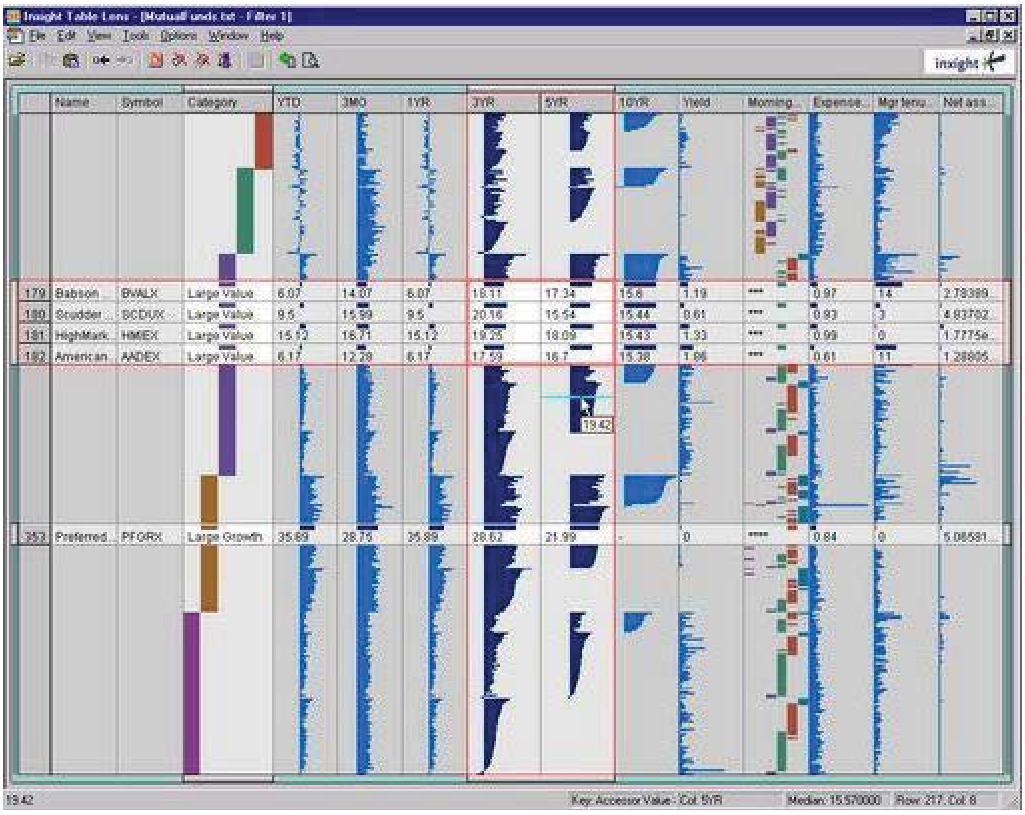


Figure 2.9 The star coordinates with 8 dimensions (Kandogan 2000).

2.3.4 TableLens

While parallel coordinates, scatterplots, and star coordinates primarily and visually encode data with non-interaction support, TableLens is a visualization technique for multi-dimensional data exploration with rich manipulation features. TableLens visualizes a data point by a bar graph placed in a cell of a dynamic table (Rao & Card 1994) (see Figure 2.10). The columns and rows of this table, corresponding to dimensions and instances, are adaptable with focus+context technique, which enables users to expand or collapse analysis zone as demands. By employing flexible and independent scale of each column, these multiple bar charts allow comparing both inside and across dimensions quantitatively.


Insight Table Lens - [MutualFunds.txt - Filter 1]
File
Edit
View
Tools
Options
Window
Help
insight
19.42
Key: Accessor Value - Col: 5YR
Median: 15.570000
Row: 217, Col: 8

	Name	Symbol	Category	YTD	3MO	1YR	3YR	5YR	10YR	Yield	Morning...	Expense...	Mgr tenu...	Net ass...
179	Babson ...	BVALX	Large Value	6.07	14.07	6.07	18.11	17.34	15.6	1.19	***	0.97	14	2.78389...
180	Scudder ...	SCDUX	Large Value	9.5	15.99	9.5	20.16	15.54	15.44	0.61	***	0.93	3	4.83702...
181	HighMark...	HMIEX	Large Value	15.12	16.71	15.12	19.25	18.09	15.43	1.33	***	0.99	0	1.7775e...
182	American...	AADEX	Large Value	6.17	12.28	6.17	17.59	16.7	15.38	1.86	***	0.61	11	1.28805...
352	Preferred...	PFGRX	Large Growth	35.89	28.75	35.89	28.62	21.99	-	0	****	0.84	0	5.08581...

Figure 2.10 A sample layout of the TableLens employment (Rao, http://www.ramanarao.com/articles/2001-12-online-info/cviz.html).

2.3.5 Discussion

Multi-dimensional visualization is more and more important in the age of big data, and there are a great number of advanced visualization techniques proposed for big-data mining power enhancement. Four of the basic approaches have been revisited above, from which the objectives of our studies in this dissertation are to develop novel techniques. The deep insights and related works of considered subjects would be interpolated in further sections.

Chapter 3 A Framework of Visual Data Exploration

Visual analytics play a fundamental role in bringing insights to the audiences who are interested and dedicated to data exploration. In the area of relational data, not a small number of advanced visualization tools and frameworks are proposed in order to deal with such data features. However, a majority of those have not significantly considered the whole process from data-model mining to dimension-based querying, which might enhance the flexibility of continuous and further exploration. This chapter presents a new interactive exploration framework for relational data analysis through the automatic interconnection of data models and data dimensions. The approach is to enable users to make relative queries flexibly on desired contexts at any stage of exploration for in-depth data understanding.

3.1 Introduction

Data mining through visualization is not a new subject, but always motivated by special considerations. In relational data application, besides single visual techniques proposed, many integrative tools were presented for incorporated data exploration such as Tulip (Auber 2012), Orion (Deer & Perer 2011), Polaris (Stolte, Tang & Hanrahan 2002), and Tableau (https://www.tableau.com/ 2017). The theme of the current models is about using various kinds of graphs for visual analysis of given data in different facets, which allows users to reach exploring objectives with many options. For instance, statistical tasks probably work well with bar and pie charts, and object-relation observation likely adapts with network graphs and tree maps. The flexibility of the integrated tools brings users the comprehensive benefits; however, the procedure of interconnection establishment is still a challenge, especially for the novices and general users who have not much technical knowledge and well-trained experience. The existing advanced tools often require programming skills in the configuration of data representation and further investigation, which can cause the increase of users' mental and physical effort during discovery procedures and might interrupt continuous exploration as well.

Data types are mostly compulsory to be formatted in a particular model for usage, and relational data are unexceptional. They are well-known with the model named Entity-Relationship (Chen 1976). Entity-Relationship model enables mapping real-world objects into conceptual diagrams that can serve information system analysis and development. For data-model (DM) interaction, conventional interfaces are basically designed in table and form layouts, mainly associated with text-based queries (Microsoft Access 2017; Oracle 2017). Such tools are often exclusive to database management of experts rather than common data exploration of general users.

Nowadays, along with big-data development, multi-dimensional analysis plays an essential duty in data mining, especially in correlation examination. Parallel coordinates (PCs) and scatterplots (SPs) are powerful techniques in this area. PCs deal with a high number of dimensions simultaneously, while SPs handle pairwise comparisons. Query manipulation on these two visualizations is greatly about brushing on data points and polylines, which can support to discover quantitative and statistical aspects of data (Ho et al. 2011; Siirtola 2000; Huang et al. 2012). By such feature, novices can directly interact and explore data with PCs, where their layout seems to be preferred by senior technicians (Siirtola & Räihä 2006). The dual analytics through PCs and SPs are quite popular and can be found in advanced suites for visual data mining such as Tableau (Tableau 2017). This software offers rich libraries of diverse graphs with various analytical tasks and also requires programing or technical experiences to perform data exploration.

The role of DMs and dimensions in data mining is not trivial; nevertheless, the practical combination of those for visual analytics has being considered separately. The two groups of techniques revisited above are being currently employed in distinct aims, where the former is favorable to database and system design (Microsoft Access 2017; http://www-03.ibm.com/software/products/en/ratirosefami 2017), and the latter is for data analysis (Auber 2004; Heer & Perer 2014; Stotle et al. 2002; Tableau 2017). We argue and believe that the visual cooperation of those two would bring empirical benefits to users, both experts and novices. In this paper, we propose a framework for the coherent interconnecting of DMs and dimensions for continuous and friendly data exploration.

In this chapter, we attempt to design a framework for coherent exploration through multiple visual contexts of graphics with automatic interconnection. By the framework employment, users are able to perform further investigation across the representations by simple interaction, with a minimum of the interruption of the technical setting procedure. Novices and general users are preferred by our motivation which is to create a friendly tool towards a right job for various target users.

3.2 A Multiple-Visual-Context Framework of Data Exploration

The framework input concentrates upon two characteristics of relational data including data models and data dimensions. We propose a coherent process of data exploration based upon an incremental chain of the visual representations of data models, multiple dimensions, and pairwise dimensions. These visual contexts are automatically linked for readiness in use and switchable for flexibility in navigation. Users are able to select and change a desired context by simple manipulation. Figure 3.1 and the following sections illustrate the model in detail.

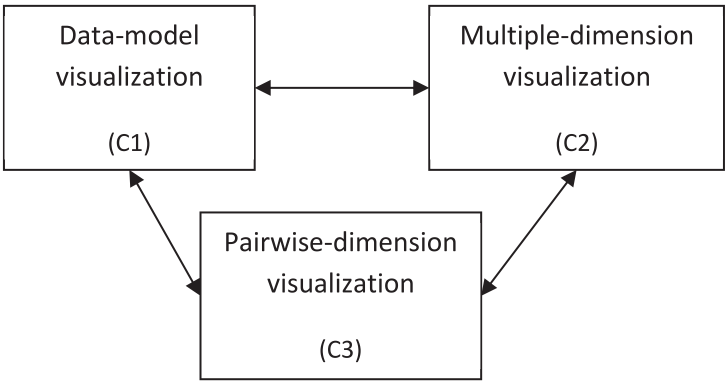

Data-model visualization
(C1)
Multiple-dimension visualization
(C2)
Pairwise-dimension visualization
(C3)

Figure 3.1 The proposed multiple-visual-context framework of relational data exploration.

3.2.1 Data-Model Context

The data-model context (C1, see Figure 3.1) is the initial stage of the process, providing general information about given data schemas or data models, including table names, dimension names (field names), and their relationships, of which users can use all to create exploring queries. We offer a new visual query environment based upon interactive node-link visualization which can aid users in exploring data with minimal technical experience. The detail of our approach will be particularly presented in Chapter 4 A New Interactive Visual Query for Relational Data Models.

3.2.2 Multiple-Dimension Context

The multiple-dimension context (C2, see Figure 3.1) is drilled down from a chosen table or a query result in the data-model context to supply a multi-dimensional background for quantitative queries. We propose an advanced technique based upon a parallel coordinate geometry for the visual analysis of this context. In the context, users can make relative comparisons and observe data summary patterns inside and across dimensions. Chapter 5 of this dissertation will present and discuss our methodology for the query design in dealing with considered challenges.

3.2.3 Pairwise-Dimension Context

The pairwise-dimension context (C3, see Figure 3.1) is drilled down from chosen dimensions of the other contexts to support correlative queries on data values in pairs. We offer interactive query methods based on dynamic scatterplots that enable users to observe the correlation with the detail of data summary in pairs of focus dimensions. The theory and practice applied and developed for the context will be illustrated in Chapter 5 of this dissertation.

3.3 Discussion

This chapter has introduced a new framework for interactive data exploration through multiple visual contexts of data models, multiple dimensions, and pairwise dimensions. The main idea is about the integration of single visual techniques with interactive queries into logical-frame-based exploration. With the framework deployment, users are enabled to interact with and utilize both data models and dimensions synchronously for analyzing different data facets, which can not only reduce the interruption of technical configuration but also maintain flexible navigation. The proposed framework plays the key role in the synthesis of the goals of our visual query designs for data exploration, whose details will be presented in the further chapters, including related works, design methods, implementation, case studies, and evaluations.

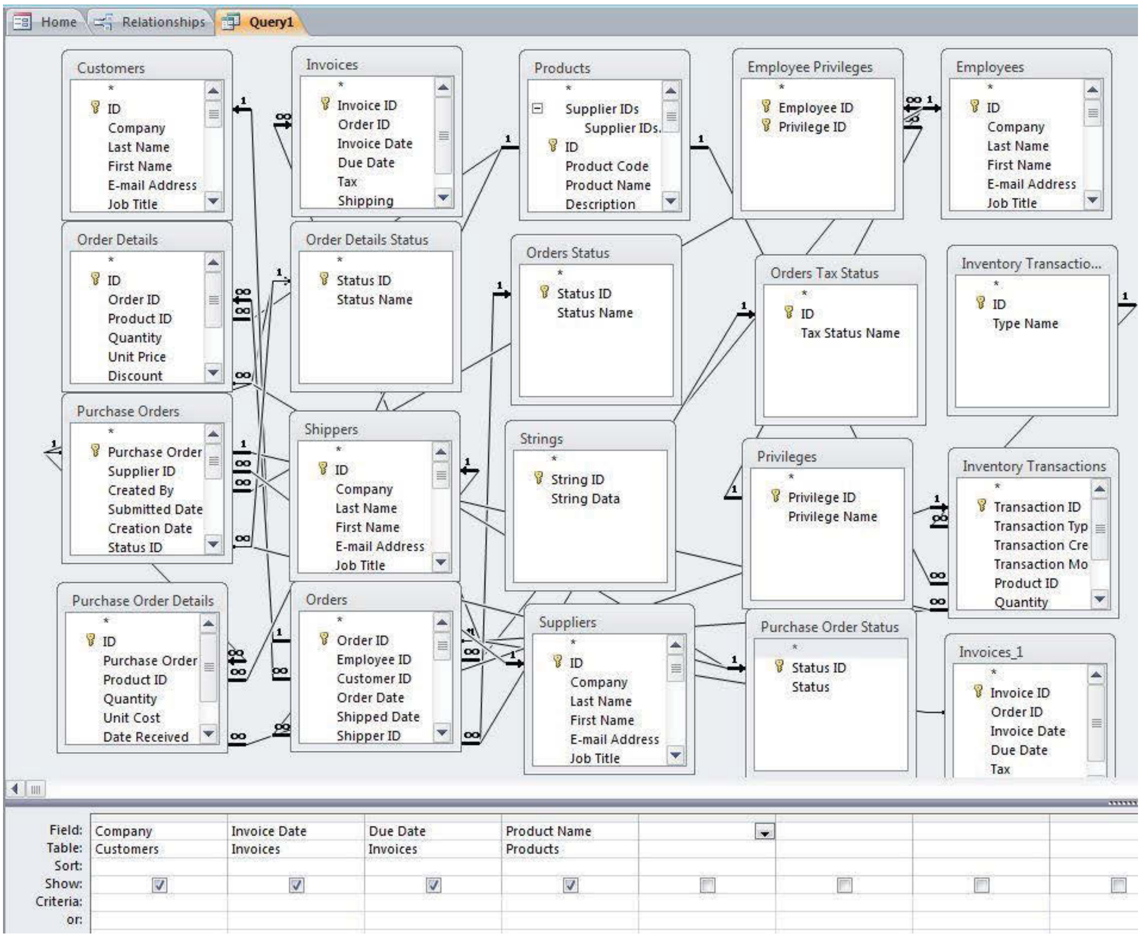
Chapter 4 A New Interactive Visual Query for Relational Data Models

The use of visual queries for relational data search and retrieval is a well-known approach, and various advanced visualization methods have been proposed to improve the quality of such queries. Nevertheless, most current methods pay attention to constructing queries in a single visualization context, which causes the isolation of data models and query results in exploration procedures. Consequently, the cognitive effort of users for query formulation and adjustment is increased, significantly in visual matching the query details with their contexts. How to visually and synchronously represent all data models, query formulation, and query results in a single screen of multi-contexts is a concerned challenge in order to make great convenience for data exploration.

This chapter presents a novel visual method called MCquery for dealing with the above challenge. This method allows query specification in the coordinating visual contexts of data models and query results by interaction on node-link graphs of relational data representation. MCquery enables relational data to be analysed with relative retrieval from the incremental exploration of data models, queries, and query results.

4.1 Revisiting Visual Queries for Relational Data

The traditional table-based and form-based representations are used as simplified query interfaces, which can support the input of query specifications and are suitable for different databases, especially for relational data schemas. The query modules of Microsoft Access and MySQL are the examples of providing such graphical interfaces in database software (Microsoft 2017; Oracle 2017) (see Figure 4.1). These are essential environment for users in observing data model details and collecting information for query making. However, since the view of data models and the view of query formulation are located separately, users could spend extra cognitive effort to trace and match useful details between those views.



Field:	Company	Invoice Date	Due Date	Product Name				
Table:	Customers	Invoices	Invoices	Products				
Sort:								
Show:	☑	☑	☑	☑	□	□	□	□
Criteria:								
or:								

Figure 4.1 The relational data query interface of MS Access 2010. The above view displays tables and relationships, and the under view shows a form for query formulation.

Besides the conventional query interfaces, a number of visualization techniques are recently offered, typically in the field of query language representation. For example, QueryViz was proposed to support SQL learning by SQL structure representations (Danaparamita & Gatterbauer 2011), and QGraph was recommended to process relational data queries with conceptual diagrams (Blau 2001) (see Figure 4.2).

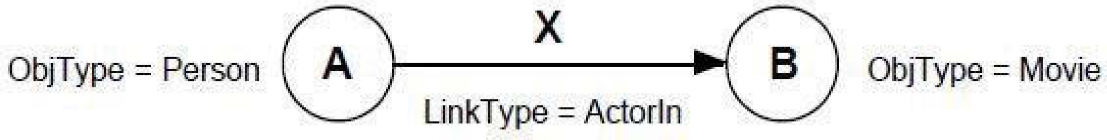


Figure 4.2 The relational data query interface of QGraph. The query is based on entities *Person* and *Movie* with relationship *ActorIn* (Blau 2001).

While the above visual query techniques are designed to control the details of query construction through conceptual data models, they pay less attention to further query result analysis. The other techniques of this category are Nitelight and Dataplay. The former was developed for semantic queries with its syntax views (Russell 2008) whereas the latter enabled users to create a query on nested universal data tree representations with result suggestion for query refinement (Abouzied, Hellerstein & Silberschatz 2012) (see Figure 4.3).

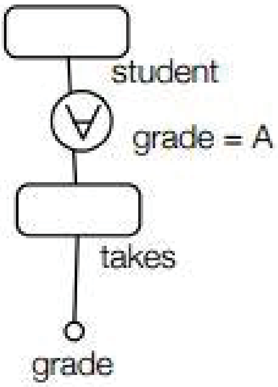


Figure 4.3 The relational data query interface of Dataplay. This method is about the query correction processes that can help users to learn nested data model constraints via the nested tree and table representations. The query is to search who are the students taking grade A (Abouzied, Hellerstein & Silberschatz 2012).

As a type of user-friendly interfaces, Hierarchical Views (HV), which are similar to file system views, can be found in many works, such as Polyviou, Samaras and Evripidou 2005 (see Figure 4.4) and Tanin, Shneiderman and Xie 2007. The tables of data models were visually converted to hierarchical elements which provide the view of entity relation levels and the components of filtering in query formulation.

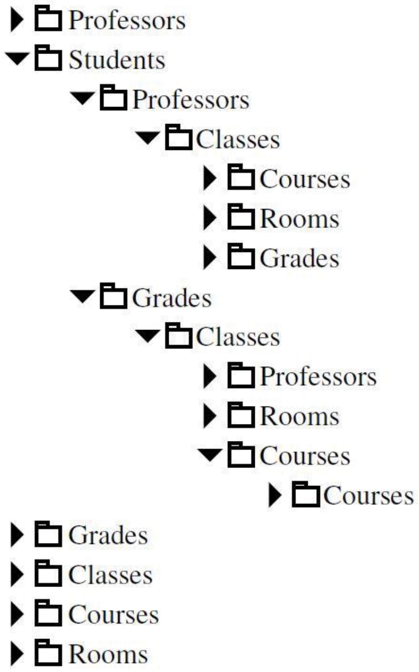


Figure 4.4 The relational data query interface based on HV. The query is drilled down from either *Students, Grades, Classes,* and *Courses* or *Students, Professors,* and *Classes* (Polyviou, Samaras & Evripidou 2005).

In addition to relational databases, graph databases and key-value databases have been being under the consideration of many user interface studies. For the features of such data associating with semantic networks, node-link graphs are preferred in most of the cases, and a high number of query interaction methods were successfully proposed, such as term-network-based query intersection (Liu, Navathe & Stasko 2001; Hoeber, Yang & Yao 2005; Seifert 2001) (see Figure 4.5) and semantic-graph-based expanding on demands (Van-Ham & Peter 2009; Tu & Shen 2013).

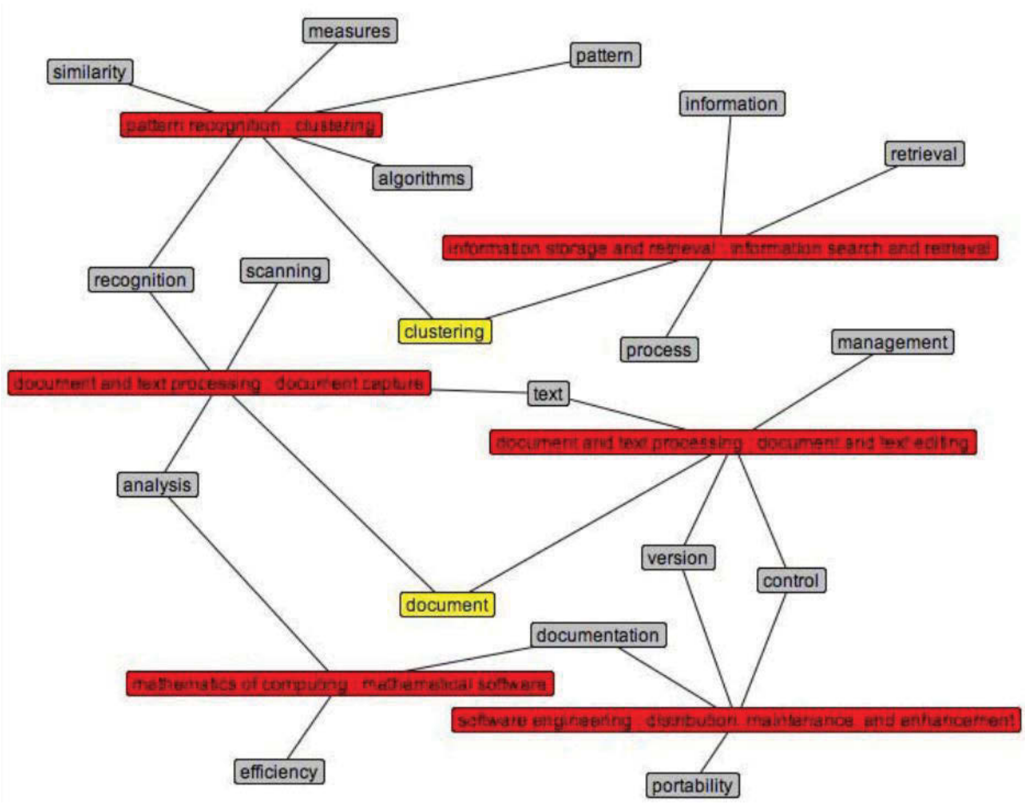

measures
similarity
pattern
pattern recognition : clustering
information
retrieval
algorithms
information storage and retrieval : information search and retrieval
recognition
scanning
clustering
process
management
document and text processing : document capture
text
document and text processing : document and text editing
analysis
version
control
document
documentation
mathematics of computing : mathematical software
software engineering : distribution, maintenance, and enhancement
efficiency
portability

Figure 4.5 The relational data query interface based on a node-link graph. The query is about the searching of two principal terms *clustering* and *document* (yellow nodes). The result pattern suggested six related term clusters (red nodes) (Hoeber, Yang & Yao 2005).

Although there are various approaches in this area, most of the current techniques would rather focus on data exploration in the single context of query formulation than deeply consider that exploration in association with the data models and query results. In these cases, users have to pay much attention to tracing and matching corresponding helpful information together, while still taking efforts to execute data exploration procedures. Our approach, distinct from the current works, is to allow users to perform query interaction on both data models and query results in their coordinating visual contexts. Thus, users are able to flexibly utilize helpful information for query making while they can still directly utilize data features for further investigation. The technique formed by the approach is called MCquery, which will be presented and discussed particularly in the following sections of this chapter.

4.2 Data Model Visualization

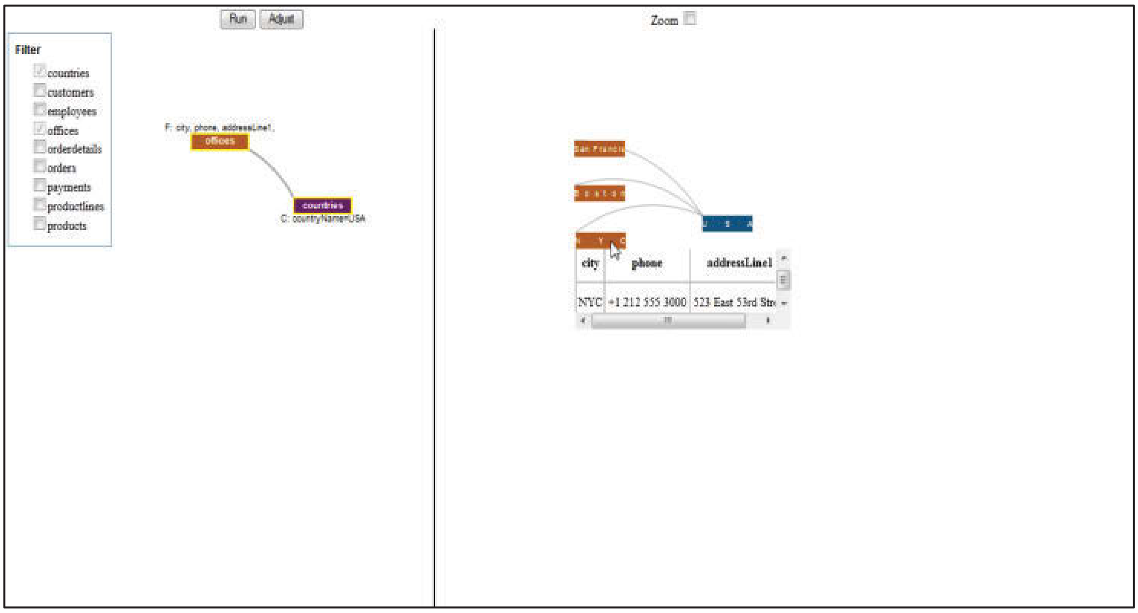
For relational databases, data models provide significant and mandatory information for query formulation; in spite of that, the data models represented by the current visual techniques have not had a firmly close connection to the data exploration processes. The views of data models are often separately located from query making areas, which can cause the increase of users' mental effort to observe and match useful query details together properly. For instance, Microsoft Access and MySQL are the general tools for data management, and their interfaces for query functions in data model views are facing such difficulties (Microsoft 2017; Oracle 2017). Moreover, although the current techniques can visualize table relationship information, they have not greatly supported the further exploration based upon data instances and tuples in query result representation. In reality, the query results are mostly visually formatted in static spread sheets and table-based forms for browsing activities.

The following sections describe a new visualization approach for the visual representation of both data models and query results in dealing with the mentioned challenges. The primary approach of this technique is the constructing synchronously operated parallel views of the data model and query result representation on interactive node-link graphs. In this method, not only the data models but also the query results are simultaneously visualized together in adaptable ways for data continuous exploration and analysis.

4.2.1 Coordinating Context Views

Coordinating context views are a well-known design in graphical user interfaces. Regularly, a couple of vertical or horizontal views enable visual navigation on overall and detailed contexts during data exploration. The type of this design is suitable for hierarchical and incremental browsing such as file management in Windows (Microsoft 2017) and code management in programming tools. We have adapted the interface design to suit our data exploration methods.

The general goal of the MCquery interface is to help users synchronously watch and interact with the visualization of both data models and query results during data discovery activities. The parallel views of MCquery are two main panels including a data model view on the left side and a query result view on the right side (see Figure 4.6). The data model view plays the role of the displaying visual representation and control information of a data model including table names, relations, context-based query formulation details, a filtering component and control buttons. The query result view is to display the visual representation and details of a query result in the associated context of the data model.



Filter

- ☑ countries
- □ customers
- □ employees
- ☑ offices
- □ orderdetails
- □ orders
- □ payments
- □ productlines
- □ products

F: city, phone, addressLine1,

C: countryName=USA

city	phone	addressLine1
NYC	+1 212 555 3000	523 East 53rd Str

Figure 4.6 The main user interface for the coordinating visual contexts of data models and query results by technique MCquery.

The coordinating context views of MCquery are synchronously updated during query formulation processes in order to maintain visual correlation properly between the graphics of data models and query results. The visual correlation plays the key role in supporting users to not only manage query input but also trace or observe the formulation changes of query adjustment in further investigation. The visual correlation is maintained by the following procedure.

Step 1: Once mapped from a relational schema and specified by a filtering component, a node-link graph representing the tables and relationships of the model is displayed with default colour attributes of the corresponding nodes.

Step 2: When the query input is activated and entered on a node, this node is then assigned a distinct colour, and the illustrated tags of an associated query command are attached on the top and under the bottom of the node (see Figure 4.6). In the event of the query input modified in the same relationships (no table added or removed), this step is repeated. On the other hand, Step 1 is repeated if the query input is modified with updated relationships (tables added or removed).

Step 3: After the query execution, besides result data, colour attributes are mapped to the result context view for the visual representation of the associated result. Thus, the nodes of the same table always obtain the same colour.

Step 4: When the query input is updated by the result context, the visual status of the model graph, including the number of tables, the number of colours, and the query tags, is transformed as well.

4.2.2 Node-link Graph Design for Relational Data Models

In the scope of this dissertation, a node-link graph refers to the graph which is drawn by the networks or connections of nodes and links. Misue et al. (1995) indicated that a node-link graph view includes creation and adjustment procedures. The creation refers to the representation of the graph by using its characteristics, and the adjustment is to make the graph adaptable to changing the values of its characteristics. When the graph changes, its layout will be adjusted for a new display. Still, these changes can prevent users from concentrating on the patterns of previous graphs. Misue et al. (1995) also proposed that preserving a mental map should be considered in order to help users more easily understand the graphs when their elements change. This theory was, then, taken into account in many works such as Li, Eades and Nikolov (2005) and Dwyer, Marriott and Stuckey (2006). From those results, the concept of *mental map* was intensely examined in term of maintaining useful information in mind to support interaction and improve navigation tasks (Purchase, Hoggan & Görg 2007; Frishnam & Tal 2008).

In our design for the data-model context, the node-link graph construction is based upon the details of the relational schema of associated data, which is adapted and transformed from a conceptual diagram (Entity-Relationship diagram). Figure 4.7 shows an example of the relational schema of six tables.

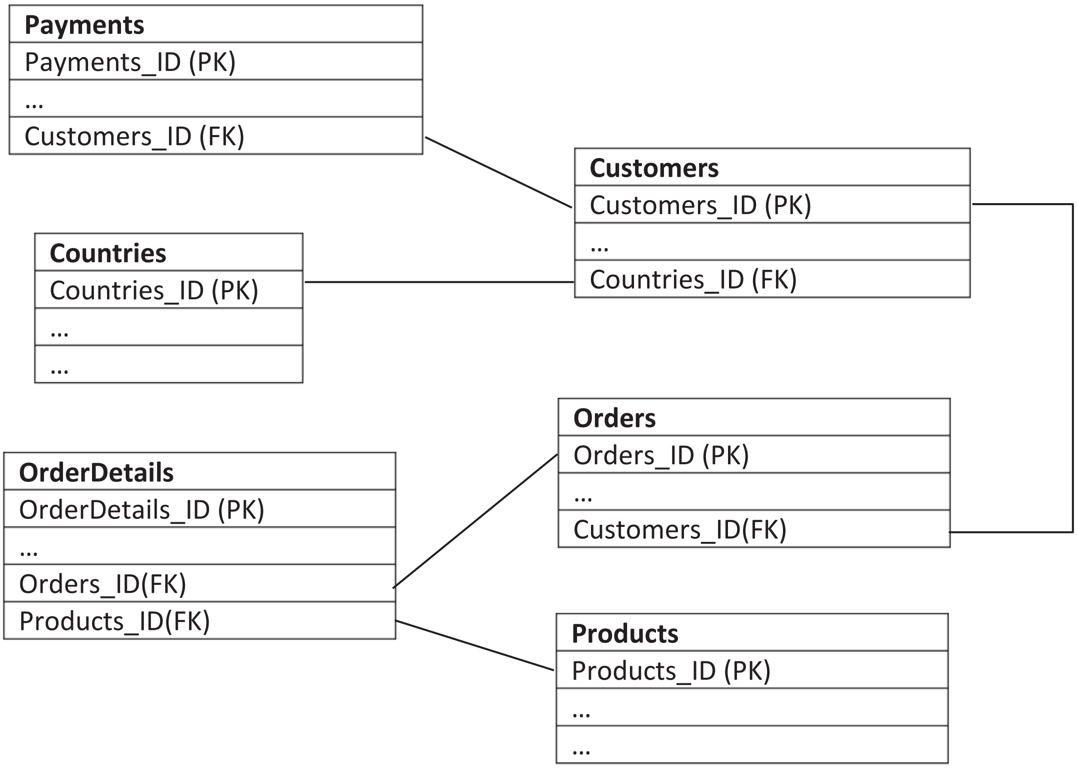


Figure 4.7 The relational schema of six tables *Payments, Customers, Countries, Orders, OrderDetails,* and *Products* used in the samples of this chapter. *PK* and *FK* stand for a primary key and a foreign key. The corresponding visualization of this schema is shown in Figure 4.8.

A graph *G* representing a data schema *(T, R)* is defined as a pair *(V, E),* where *V* is the set of vertices or nodes, *E* is the set of edges or links between nodes, $E = \{(u,v) \mid u, v \in V\}$, *T* is the set of tables, and *R* is the set of relationships (see Figure 4.8). The rectangle nodes visually represent the data tables, and the curve links between the nodes visually represent the relationships between the data tables, including *one-one, one-many*, and *many-many* relationships. Figure 4.8 shows a sample of the relational data model representation for the schema in Figure 4.7.

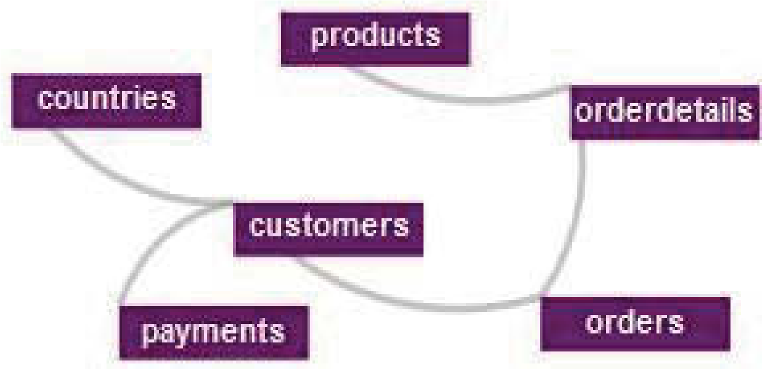


Figure 4.8 A sample of the data model representation for six tables. The name of the tables is placed in the central of the nodes.

In the query result context, MCquery uses node-link graphs with force-directed layouts for query result visualization. We attempt to make the query result representations significantly concentrated on the matched tables of the data model context by applying the architecture of data tuples and their relationships for the result graphs.

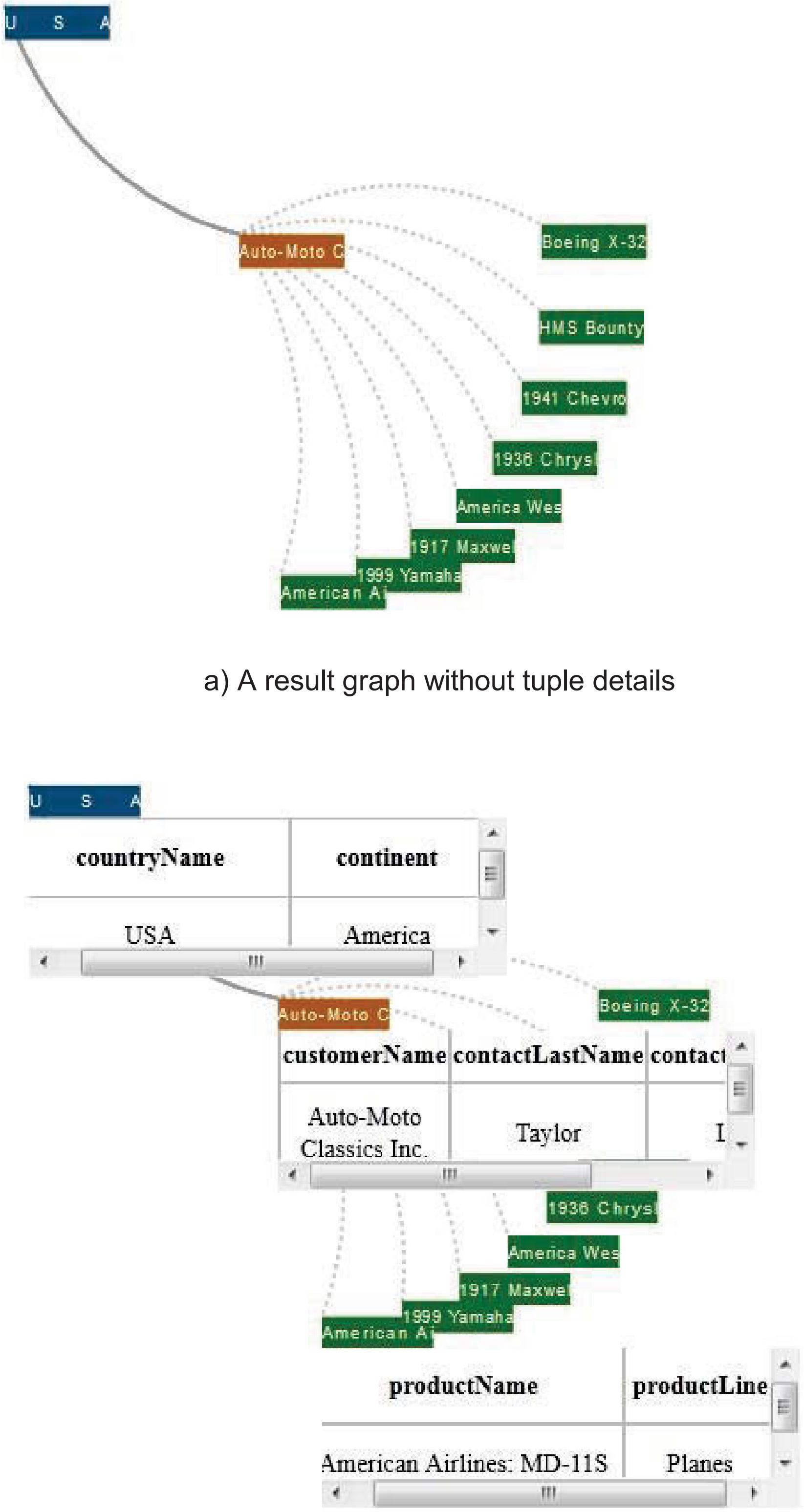
The visualization of a query result is defined as follows.

$$G_i = (V_i,E_i),\ G_i \in G',\ G' = (V',E'),\ V = f(T_1) + f(T_2), \text{ and } Emax_i = |f(T_1)| \text{ x } |f(T_2)|,$$

Equation 4.1

where G_i is the graph representing a given pair of tables T_1 and T_2, V_i is the set of the nodes representing T_1 and T_2, $f(T_1)$ is the set of m-tuples of T_1 involved in Q_i, $f(T_2)$ is the set of n-tuples of T_2 involved in Q_i, $Emax_i$ is the maximum number of the links between the tables of Q_i, and Q_i is i^{th} query.

In this method, a tuple involved in the query result is represented by a dynamic node, which is featured to flexibly display *n* and *m* field values or *n* and *m* dimensions (see Figure 4.9.b). The tuple details are visualized by pop-up panels. A pop-up panel is visible when a mouse pointer goes over the node. Users are enabled to replace or switch the label of the node by a click on a favourite dimension.



a) A result graph without tuple details

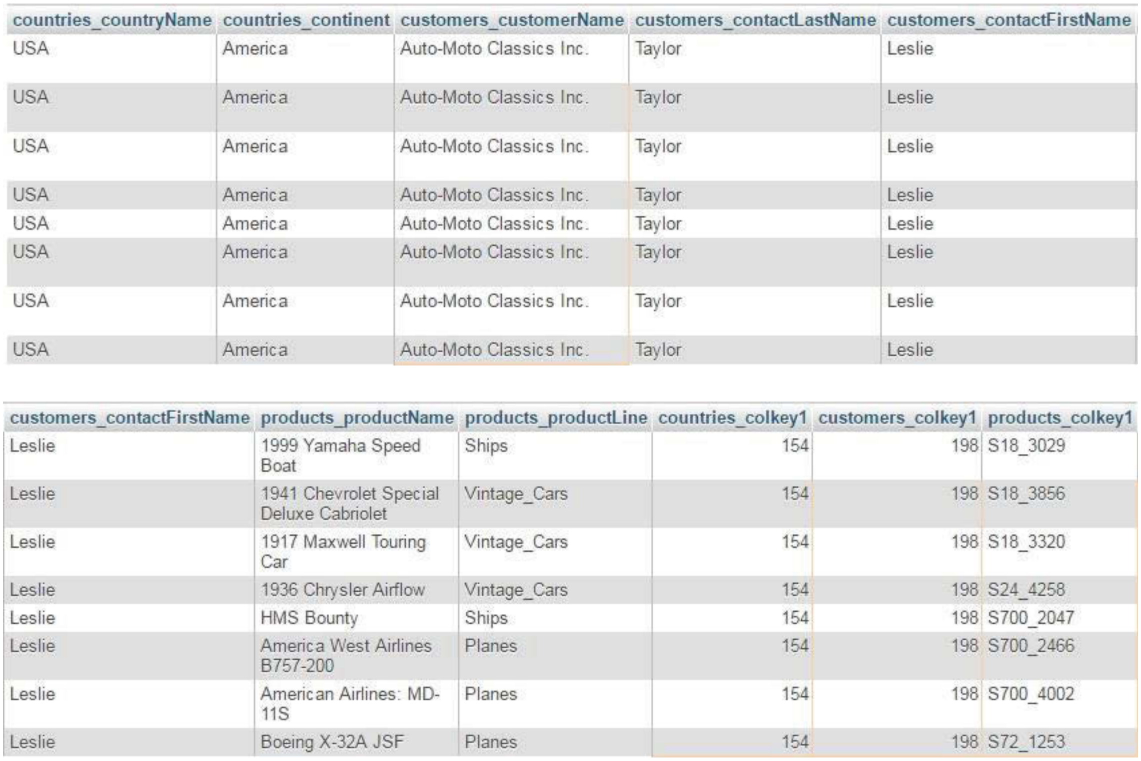
countryName	continent
USA	America

customerName	contactLastName	contact
Auto-Moto Classics Inc.	Taylor	I

productName	productLine
American Airlines: MD-11S	Planes

b) A result graph with the activated tuple details

Figure 4.9 A sample of the result graph corresponding to three tables *countries* (a blue node), *customers* (an orange node), and *products* (the green nodes).



countries_countryName	countries_continent	customers_customerName	customers_contactLastName	customers_contactFirstName	products_productName	products_productLine	countries_colkey1	customers_colkey1	products_colkey1
USA	America	Auto-Moto Classics Inc.	Taylor	Leslie	1999 Yamaha Speed Boat	Ships	154	198	S18_3029
USA	America	Auto-Moto Classics Inc.	Taylor	Leslie	1941 Chevrolet Special Deluxe Cabriolet	Vintage_Cars	154	198	S18_3856
USA	America	Auto-Moto Classics Inc.	Taylor	Leslie	1917 Maxwell Touring Car	Vintage_Cars	154	198	S18_3320
USA	America	Auto-Moto Classics Inc.	Taylor	Leslie	1936 Chrysler Airflow	Vintage_Cars	154	198	S24_4258
USA	America	Auto-Moto Classics Inc.	Taylor	Leslie	HMS Bounty	Ships	154	198	S700_2047
USA	America	Auto-Moto Classics Inc.	Taylor	Leslie	America West Airlines B757-200	Planes	154	198	S700_2466
USA	America	Auto-Moto Classics Inc.	Taylor	Leslie	American Airlines: MD-11S	Planes	154	198	S700_4002
USA	America	Auto-Moto Classics Inc.	Taylor	Leslie	Boeing X-32A JSF	Planes	154	198	S72_1253

Table 4.1 The data of the query result visualized in Figure 4.9. The two tables are only one which is separately allocated for the page width fitting.

In addition, by assigning a distinct colour for each of the tables, the corresponding nodes of one table are filled the same colour during query operations, which enables the query result layout to probably look concise with the data model matching (see Figure 4.9.a).

The colour design employed throughout of visualization techniques in this dissertation is based on the categorical colour scheme of d3js library including the category of ten and twenty colours (d3js 2017). The detail of the colour design is presented in Table 4.2 and Table 4.3.

Colour	Hex Code	RGB Code
●	#1F77B4	RGB(31,119,180)
●	#FF7F0E	RGB(255,127,14)
●	#2CA02C	RGB(44,160,44)
●	#D62728	RGB(214,39,40)
●	#9467BD	RGB(148,103,189)
●	#8C564B	RGB(140,86,75)
●	#E377C2	RGB(227,119,194)
●	#7F7F7F	RGB(127,127,127)
●	#BCBD22	RGB(188,189,34)
●	#17BECF	RGB(23,190,207)

Table 4.2 The 10-categorical colour scheme of d3js library (d3js 2017).

Colour	Hex Code	RGB Code
●	#1F77B4	RGB(31,119,180)
●	#AEC7E8	RGB(174,199,232)
●	#FF7F0E	RGB(255,127,14)
●	#FFBB78	RGB(255,187,120)
●	#2CA02C	RGB(44,160,44)
●	#98DF8A	RGB(152,223,138)
●	#D62728	RGB(214,39,40)
●	#FF9896	RGB(255,152,150)
●	#9467BD	RGB(148,103,189)
●	#C5B0D5	RGB(197,176,213)
●	#8C564B	RGB(140,86,75)
●	#C49C94	RGB(196,156,148)
●	#E377C2	RGB(227,119,194)
●	#F7B6D2	RGB(247,182,210)
●	#7F7F7F	RGB(127,127,127)
●	#C7C7C7	RGB(199,199,199)
●	#BCBD22	RGB(188,189,34)
●	#DBDB8D	RGB(219,219,141)
●	#17BECF	RGB(23,190,207)
●	#9EDAE5	RGB(158,218,229)

Table 4.3 The 20-categorical colour scheme of d3js library (d3js 2017).

4.2.3 Data Mapping for Visualization

In the context view of a data model, graph *G = (E,V)* is constructed by using the pairs of Source-Target associated with the tables and relationships which are extracted from the data schema information. The procedure of searching the relationships and forming the nodes of *G* in the data mapping is presented in Algorithms 4.1, 4.2, and 4.3.

```
procedure SetRelationships (information_schema) /* The schema information of a given database*/
    Links = ø , e = 0 /*The list of links is initialized to be empty*/
    for each Rec in information_schema.Key_Column_Usage /*Browse all the records of Key_Column_Usage*/
        if Rec[Referenced_Table_Name] not Empty then /*if the current table refers to another one, a connection is available */
            Links[e++][Source] = Rec[Table_Name]
            Links[e][Source_Key] = Rec[Column_Name] /* A foreign key of Source*/
            Links[e][Target] = Rec[Referenced_Table_Name]
            Links[e][Target_Key] = Rec[Referenced_Column_Name]/*Either a primary key or a partly composite key*/
        end if
    end for
  return Links
end procedure
```

Algorithm 4.1 The algorithm of setting relationships from the data schema information.

```
procedure IsAForeignKey (table_name, column_name, Links)
    e = 0 /*A link counter is initialized to be zero*/
    while not (e <= Links.length  && Links[e++][Source] == table_name && Links[e][Source_Key] = column_name) /*Browse each link until reach a foreign key of the right table*/
    end while
    if e <= Links.length then
        return true
    else
        return false
end procedure
```

Algorithm 4.2 The algorithm of checking a foreign key from the data schema information.

```
procedure SetNodes (information_schema) /* The schema information of a given database*/
    Nodes = ø, v = 0 /*The list of nodes is initialized to be empty*/
    for each Table in information_schema.tables
        Nodes[v++][Table_Name]= Table[Name]
        c = 0
        for each Column in Table[Name].columns
            Nodes[v][c++][Column_Name]= Column[Name]
            if Column[Key] == "PRIMARY" then
                Nodes[v][c][Primary_Key] = "True"
            else
            if IsAForeignKey(Nodes[v][Table_Name], Nodes[v][Column_Name], Links) then
                Nodes[v][c][Foreign_Key] = "True"
            end if
        end for /*Column*/
    end for /* Table*/
    return Nodes
end procedure
```

Algorithm 4.3 The algorithm of setting nodes from the data schema information.

In the query result graphs, the data of a node include a table name, key values, and the tuple data involved in the query. All the values of a primary key and foreign keys of each the tuple are compulsorily taken into account. The combination of the key values and their table names would ensure the node to be identified by the others. These mentioned data are used further in order to update query formulation and filter query result graphics.

The links of a result graph are mapped from the relationships of the associated data model graph. Once received from query execution, the query result is parsed into the pairs of a table name and its fields. Each of the pairs is matched to the pairs of the source and target data of the links in the data model graph for defining the connections of the corresponding nodes.

Furthermore, technique MCquery supports indirect link representation of the query result graphs for graphical comparing the relationships of the data models. A direct link represents one join, and an indirect link represents many joins of a query result. The joins are represented by indirect links when one of the involved tables in the joins is not activated. Figure 4.9 shows indirect links between the node of table *customers* and the nodes of table *products*.

We assume that an underlying query is executed by the following relational algebra operations.

$$\pi_{c1,\, c2, \ldots cm,\, c'1,\, c'2, \ldots c'n} (F(T,T')) \text{ and}$$

$$F(T,T') = \sigma_{cond} (T \bowtie T_0 \bowtie T_1 \bowtie T_2 \bowtie T_i \bowtie \cdots \bowtie T'),$$

Equation 4.2

where *T, T'* are the two tables involved in the query, and T_i is a table which might join in the query. Then, a graph link is direct when there is no T_i, and a link is indirect when there is a T_i in the join. The procedure of visualization mapping from the data of a query result is given in Algorithm 4.4.

```
procedure VisResultMapping (QueryRelationship, QueryResult)
    SourceTargetList = Ø, v = 0 /* A list of the pairs of Source and Target is initialized to be empty*/
    for each Ri in QueryRelationship/*Browse in each relationship*/
        for each Rec in QueryResult /*Browse in each result data record*/
    v++
            for each Tup in Rec /*Browse in each tuple of the current record*/
                if SourceTargetMatch(Tup, Ri) then /*If the tubple matches Source or Target of relationship Ri*/
                SourceTargetList[v][Source/Target] = Tup/*Assign the tupple into Source or Target List at position v*/
                end if
```

```
                    if Full(SourceTargetList[v]) then /*If position v sufficiently has a pair of Source and Target */
                        break/*Exit the current loop in Rec*/
                    end if
                end for /*Tup in Rec*/
            end for /*Rec in QueryResult*/
        end for /*Ri in QueryRelationship*/
    return SourceTargetList
end procedure
```

Algorithm 4.4 The algorithm of the visualization mapping from the data of a query result.

4.2.4 Discussion

We have presented a new methodology in visualizing relational data models for query formulation interfaces. The core technique is about automatic transforming the information of data schemas from a database to node-link graphs in coordinating context views and visual representing tuples in single and dynamic nodes. The main benefits of the design are to support the convenient observing of visual correlation between query requests and result graphs and to aid the quick performing of further exploration through direct interaction on the visual contexts.

4.3 Query Interaction

Query interaction plays a vital role in cognitive operations for constructing and formulating a precise query. The impacts of the cognition on users' behaviours can lead to the changes of exploration trends and decision making. Therefore, the query quality partly depends upon user interface interaction activities. A query formulation process probably mainly includes four stages *exploration, construction, feedback*, and *presentation* (ter Hofstede, Proper & van der Weide 1996). The idea is illustrated in Figure 4.10.

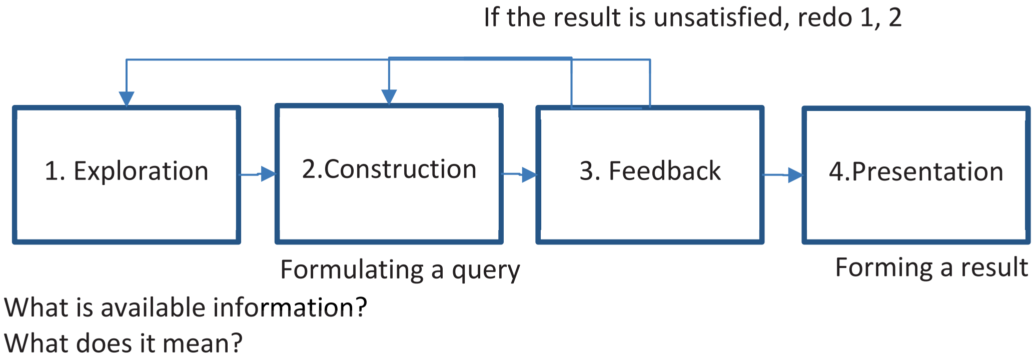


Figure 4.10 A query formulation model (ter Hofstede, Proper & van der Weide 1996).

According to the model, at the first stage, a user needs to learn and gain understanding from available information. Then, the knowledge gained of the first stage is used for query construction and formulation of Stage 2. At Stage 3, once query execution, if the result is not satisfied, he or she would go back and conduct Stage 1 and Stage 2 again. Finally, the satisfying result would be normally formed in a report type at Stage 4.

In term of interactive query formulation, a well-known classification by Wacholer (2011) offered three types of query formulation as follows.

- Contextualized: "*indicate that characteristics of the user, setting, and domain directly affect the development of the information need and indirectly affect the query itself*"
- Iterative: refer to query reformulation and relative queries
- Assisted: refer to query expansion and query supported by relevant information.

We modify the mentioned models and make them adaptable with our interactive visual query technique. For the query formulation model of ter Hofstede, Proper and van der Weide (1996), we apply and integrate Stages 1, 2, and 3 in the visualization of data models and queries through the coordinating context views. For the query types proposed by Wacholder (2011) including contextualized, iterative, and assisted ones, we equip our interaction techniques with all the queries in the expectation of good supporting diverse query interaction for users. The details of our model construction are illustrated in the following sections.

4.3.1 Interaction Model for Coordinating Context Views

In addition to the related works above, we are partly motivated by limitations of the query modules of Microsoft Access and MySQL (Microsoft 2017; Oracle 2017), which have not well encouraged users to directly formulate queries on data model representation and have not genuinely supported further data exploration. Their interaction model might be described in Figure 4.11.

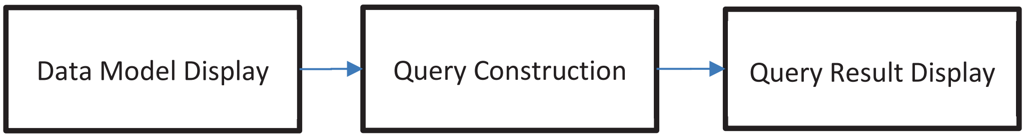


Figure 4.11 The query interaction model proposed for the query module of Microsoft Access and MySQL.

We present a new model that supports data exploration by using the coordinating context interaction of data models and query results (see Figure 4.12). Users are able to adjust relative queries flexibly to seek their favourite information through continuous exploration of visual representations.

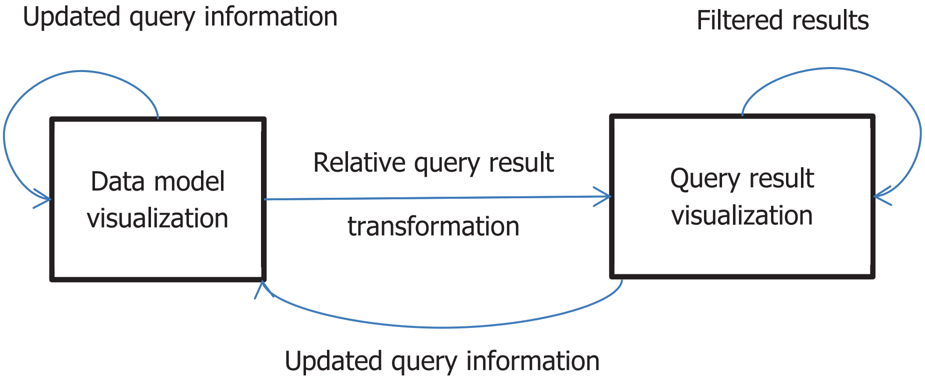

Updated query information
Filtered results
Data model visualization
Relative query result
transformation
Query result visualization
Updated query information

Figure 4.12 The new model for the data exploration by interaction on the coordinating visual contexts of data models and query results.

According to our model, a query is firstly formulated from the data model visualization. Once received from the query execution, the query result is visualized in the query result context, and users could quickly navigate the favourite result pattern with a filtering feature. For further exploration, users are enabled to adjust the query formulation instantly and directly in the visualization of both the data model and the query result. Searching relative results is the top priority of the continuous exploration. All updates of the query information are displayed in the data model visualization synchronously for managing query requests during exploration activities.

4.3.2 Interaction Model for Node-Link-Based Queries

We can informally say "*Find what with what conditions*".

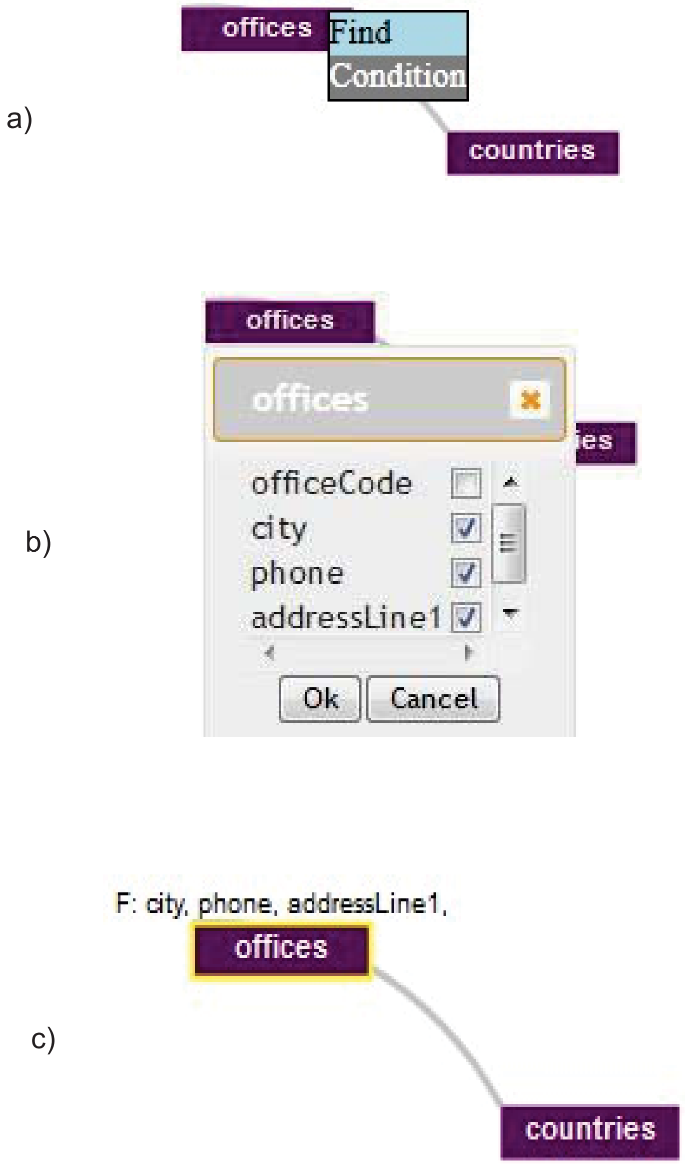
The requests of a query are performed by finding and condition components. The finding function *F* allows users to select dimensions for display. The condition function *C* enables users to enter desired values for seeking. A basic query Q_i is defined as follows:

$$Qi = (F\ (f_1, f_2, f_3,\ldots,f_n), C(c_1, c_2, c_3,\ldots, c_m)), \text{ and } c_m = (f'_m, v_m),$$

Equation 4.3

where f_n is the n^{th} field in the collection of selected fields for *F*, c_m is the m^{th} query condition in the collection of entered conditions for *C*, and f'_m and v_m are the pair of the field and its search value of condition m^{th}. The search values are entered by users or suggested by the system. Referred to relational algebra operations, *F* is associated with projection, whereas *C* is equivalent to selection.

The primary interaction used in query creation is clicks and selection. A series of the core steps for creating the query in the data model context is illustrated in Figure 4.13 and Figure 4.14.



officeCode □
city ☑
phone ☑
addressLine1 ☑

F: city, phone, addressLine1,

Figure 4.13 The instance of a query formulation with the finding component in the data model context. The request is to find *city, phone,* and *addressLine1* of *offices* which are located in *USA*.

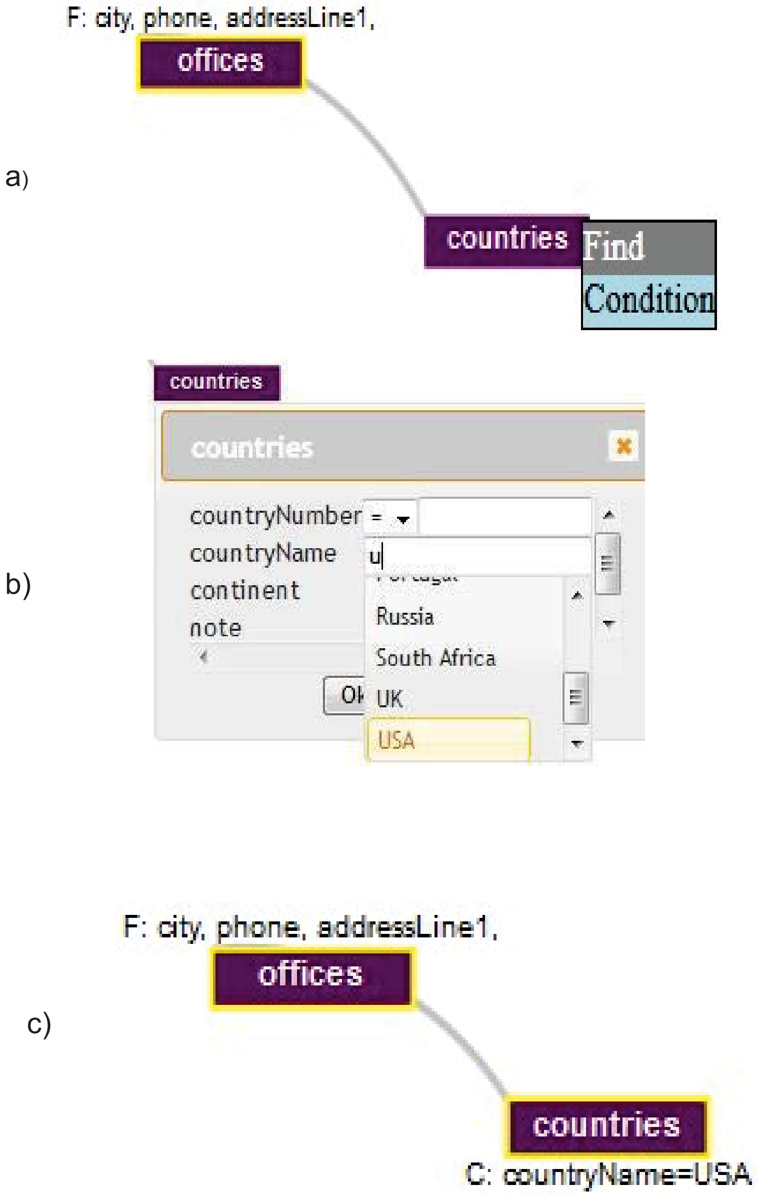


Figure 4.14 The instance of a query formulation with the condition component in the data model context. The request is to find *city, phone,* and *addressLine1* of *offices* which are located in *USA*.

The request in Figures 4.13 and 4.14 is to find *city, phone,* and *addressLine1* of *offices* which are located in *USA*. At first, a user accesses the finding function by a right-click on *offices* and selects *Find* on the popup menu (see Figure 4.13.a). Then, the user selects dimensions *city, phone,* and *addressLine1* via dialog box *offices* for finding information (see Figure 4.13.b). Node *offices* gets in the yellow border with the annotation above (see Figure 4.13.c). Similarly, after accessing the condition function, the user enters word *USA* in field *countryName* on dialog box *countries* for searching criteria (see Figure 4.14). The complete representation of the query is shown in Figure 4.14.c.

For dealing with multiple joins in a query visualization, by recalling the formula from Equation 4.2

$$\pi_{c1,\, c2, \ldots cm,\, c'1,\, c'2, \ldots c'n} (F(T,T')) \text{ and}$$

$$F(T,T') = \sigma_{\text{cond}} (T \bowtie T_0 \bowtie T_1 \bowtie T_2 \bowtie T_i \bowtie \cdots \bowtie T'),$$

we offer a component allowing users to omit the links or relationships which do not take part in the query, and it is about using popup menu *Not involved/Involved*. Users can access the menu by a right-click on a link and select *Not involved/Involved* to ignore or keep the link. The uninvolved link will be marked by a transparent colour with an annotation (see Figure 4.15).

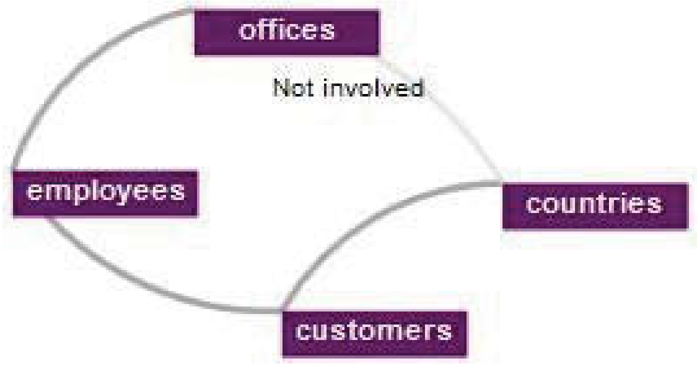
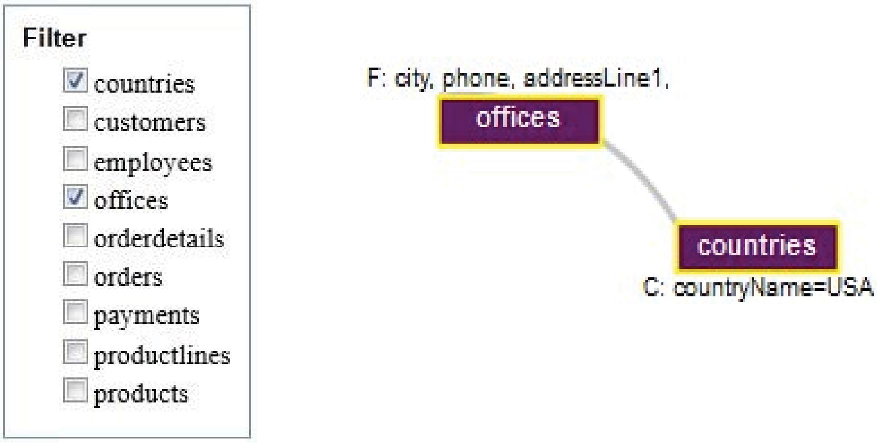


Figure 4.15 The instance of removing a link from a query. This query does not involve the relationship between *offices* and *countries*.

Besides, for the handling of displaying a large number of tables in the data model context, a table filtering panel is featured in order to help users select and focus on the considered tables. Only the displayed tables are allowed to join in queries. Thus, the data model views can be kept clear and concentrated (see Figure 4.16).



Filter

☑ countries
□ customers
□ employees
☑ offices
□ orderdetails
□ orders
□ payments
□ productlines
□ products

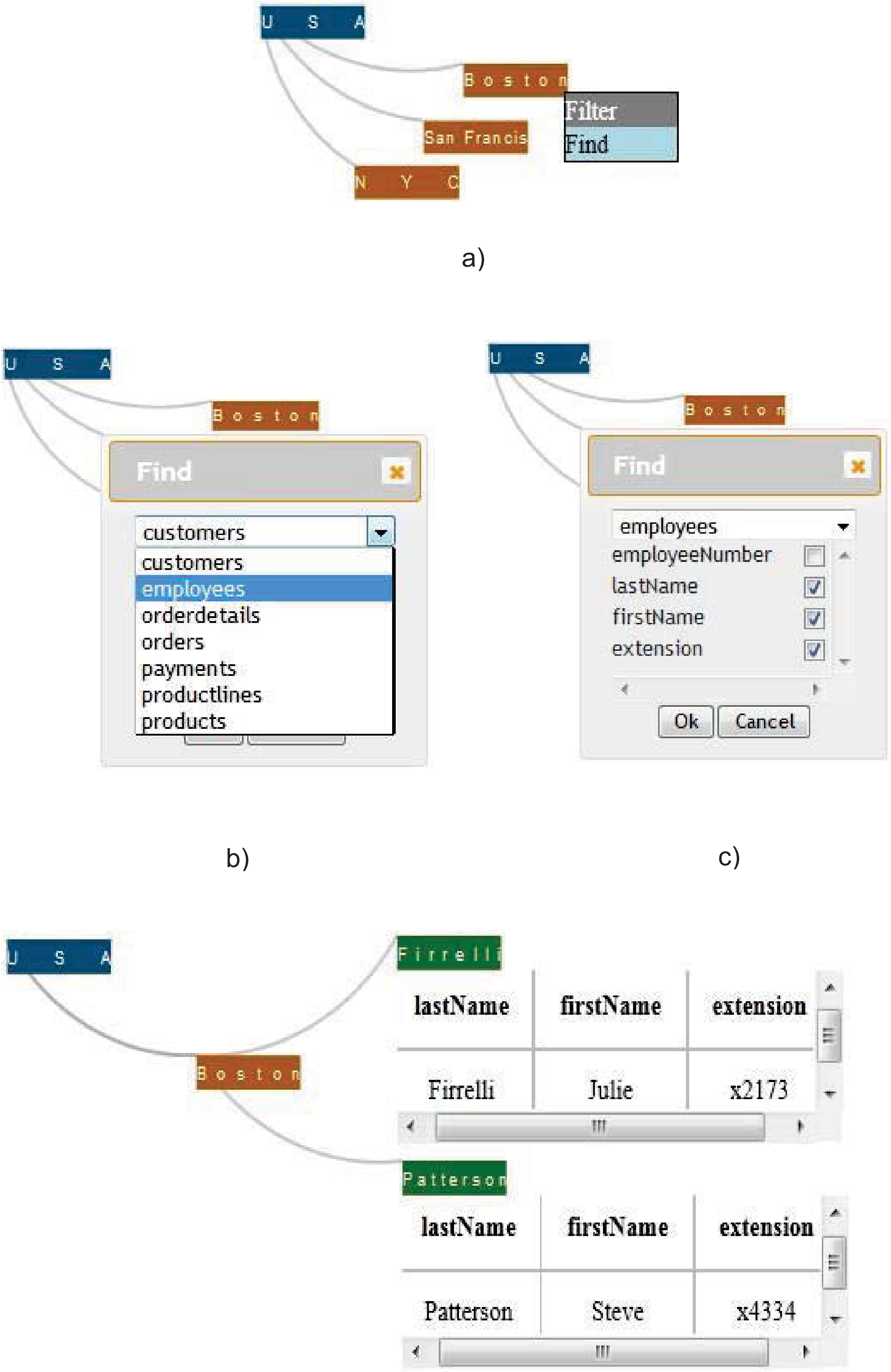
Figure 4.16 The filtering feature in the data model context. *Offices* and *countries* are the two tables selected to join in the query.

In the context of query results, MCquery enables users to use the query results directly for other related requests. The task can be done by selecting finding dimensions with the condition values of an activated node. Users are able to explore data continuously and quickly for further analysis without the interruption of query reformulation in the data model context. We recall Equation 4.3 above

$$Qi = (F\ (f_1, f_2, f_3,\ldots,f_n), C(c_1, c_2, c_3,\ldots, c_m)), \text{ and } c_m = (f'_m, v_m).$$

Here, f_n, f'_m, and v_m are directly selected on the visual representation of the query results for further discovery. This approach is our novel interaction method proposed for relational query formulation, which has not been previously developed and deployed in the area, according to our best knowledge.

For instance, a query result shows that there are three offices *Boston, San Francisco,* and *NYC* located in *USA* (see Figure 4.17.a). Now, a user would like to learn who are employees working in *Boston*. To find the answer, the user does a right-click on *Boston* and clicks on popup menu *Find* (see Figure 4.17.a). Then, dialog box *Find* is displayed for the user to select table *employees* (see Figure 4.17.b), dimensions *lastname, firstname,* and *extension* (see Figure 4.17.c). The result pattern shows that there are two employees working in *Boston* (see Figure. 4.17.d).



employees

employeeNumber □
lastName ☑
firstName ☑
extension ☑

Firrelli

lastName	firstName	extension
Firrelli	Julie	x2173

Patterson

lastName	firstName	extension
Patterson	Steve	x4334

Figure 4.17 The instance of query interaction in the query result context.

4.3.3 Incremental Data Exploration

Incremental exploration refers to processes which display small parts of a whole graph and allow altering these detailed views to other remaining parts (Herman, Melancon & Marshall 2000). This technique is suitable for representation of the large graph whose details could not be shown at the same time in the same view. Each view displayed at a time is named a logical frame, and it is greatly considered that how logical frames can be created and allocated appropriately (Huang 1998). Figure 4.18 indicates the concept of *logical frames* which will be used to guide the design in our method about constant changes of the context views.

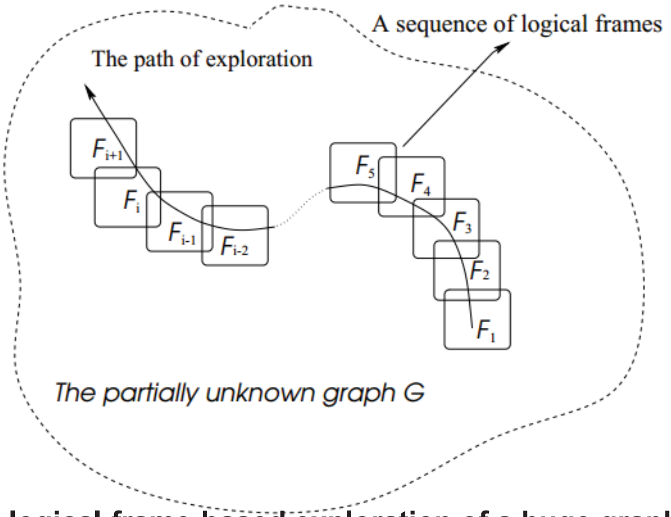


Figure 4.18 The logical-frame-based exploration of a huge graph (Huang et al. 1998).

The MCquery interaction process is constructed in the incremental exploration for the purpose of enabling query making in relative and continuous steps. Consequently, the formulation of the current query is based upon the previous one, and so their results are relative certainly. The level of relation between the queries depends on the amount of information reused by and of others. Therefore, we define our incremental graphs for the relative queries as follows.

$$G'_1 \text{ -> } G'_2 \text{ -> } G'_3 \text{ ->} \ldots\ G'_n, \textit{ and}$$

$$\{N_1,L_1\} \text{ -> } \{N_2,L_2\} \text{ -> } \{N_3,L_3\} \text{ ->} \ldots\ \{N_n,L_n\},$$

where G_n is the graph of query n^{th}, and N_n and L_n are the set of nodes and the set of links of G_n respectively.

Then, relative query $Q_n = (F_n, C_n)$ is illustrated by:

$$\begin{cases} F_n = F_n \cup F_{n-1} \text{ or } F_n = F_n \cap F_{n-1} \\ C_n = C_n \cup C_{n-1} \text{ or } C_n = C_n \cap C_{n-1}, \end{cases}$$

Equation 4.4

where Q_n is n^{th} query, F_n is the set of the finding dimensions of n^{th} query, and C_n is the set of condition values of n^{th} query.

4.3.4 Discussion

We have introduced a new technique named MCquery, which features novel visual representation and flexible interaction for relative query making through coordinating context views and node-link based exploration. According to our best knowledge, there is, currently, no technique proposing the same method as ours in the area. The data query tool of Microsoft Access (Microsoft 2017) and Ploceus (Liu, Navathe & Stasko 2011) are known as the most similar software. The difference of these facilities might be influenced by the usage purpose and the target user of the tools. While Access and Ploceus would focus on data management and user-defined queries for experts, MCquery tool prefers to deep exploration with flexible query making for novices and ordinary users. Relational data in MCquery are visually represented by tuple-based-node-link graphics, not only for data model interaction but also for relative and further discovery. The approach effectiveness is evaluated by case studies in Chapter 6 and technical assessment in Chapter 7.

4.4 Visual Navigation Methods

4.4.1 Focus + Context

Technique *focus + context* can display details while keeping the general view of them. This method does not change technique *zooming*; yet, it refines the zooming. Use of fish-eye views is a primary way to apply the focus + context technique in node-link graphs. The fish-eye algorithm specifies focal nodes and calculates necessary space size for the zooming area. The remaining nodes far from the foci could be maintained in the rest space (Noik 1993). Technique fish-eye mostly only supports to keep the general or contextual views in the same level of the detailed view, while navigation tasks are often required to change different context views. A number of algorithms and methods have been examined to deal with multiple context views. EncCon of Nguyen and Huang (2005) used a zooming + layering technique integrated with semi-transparent layers to enhance effectiveness of the multi-context navigation. Huang and Nguyen (2008) also proposed an innovative system combining fish-eye views and Chain-Context views for increasing the context flexibility in navigation paths. Tu (2010) proposed view switching for context altering support; however, differently from the mentioned other works, this technique did not substantially focus on maintaining general views.

In our design, the general view is assigned to the data model context since the context contains and displays all the query description including table names, field names, search values, conditions, and relationships. By observing the general view, users can locate what the query is being used and where the level of relationships impacts on, which supports the management of almost whole the query. Additionally, users can drill down or roll up the relation levels to trace exploration paths.

On the other hand, the query result context plays the role of the focus view. While the data model view represents the general relationships, the focus view shows the details of those with tuple representation. This view accurately indicates what the tuple of a table directly or indirectly connects with the tuples of others, which could not be observed and analysed by the general view. In simple words, the focus view enables to locate what a node is representative of the table and the tuples. These navigation activities are not trivial for completing the query formulation and adjustment.

4.4.2 Zooming and Filtering

Zooming is a fundamental method to observe graphs in detail levels. The small parts of a graph are enlarged to be displayed in a limited space. Technique zooming can be classified into two groups *geometric zooming* and *semantic zooming* (Herman, Melancon & Marshall 2000). While geometric zooming is not different from the original technique, semantic zooming requires the context information of selected area to be displayed. Such techniques are appropriate to the graphs encoding not a big amount of information. For huge graphs or the graphs with a lot of details, zooming in a local area could cause the missing of contextual or global information. For improving the simple zooming, a filtering method is often embedded for ignoring unrelated or unused details. As a result, the display space optimization is improved as well. A successful deployment of zooming + filtering can be found in Bederson, Grosjean and Meyer (2004).

For handling the display of a vast number of nodes in the query result context, MCquery employs the zooming and filtering based upon the node-link graphs. The zooming might affect the readable details whereas the filtering could be better for quick switching the general and targeted views without readability influences. When the filtering is activated, a focused node and its related neighbours would be kept visible, and the others would be hidden. The focus zone could be easily changed for result analysis and further exploration. Since our node-link graphics are drawn by the tuple representation, the filtering rule is defined as follows.

It is supposed that

$R=\{r_1, r_2, r_3,\ldots,r_x\}$, $r_x=\{tx_1,tx_2,tx_3,\ldots,tx_y\}$, and $n_x=\{nx_1,nx_2,nx_3,\ldots,nx_y\}$ **Equation 4.5**

where R is the set of query result records, r_x is x^{th} record of R, tx_y is y^{th} tuple of r_x, n_x is a set of associated nodes of r_x, and nx_y is y^{th} node of n_x and represents for tuple tx_y.

Therefore, n_x would be maintained if associated r_x contains a satisfying tuple.

In addition, in order to save the display space, the division of the two context visualizations could be customized with flexible layout demands. MCquery also features sticky layouts which enable users to arrange the graph elements by their favourite. Moreover, for serving multi-dimensional displays, MCquery provides vertical and horizontal scroll bars in the tuple panels at the activated nodes. This component allows users to watch, find, and replace a focus dimension for visual analysis.

4.5 Query Implementation

4.5.1 A System Framework for Visual Queries with Coordinating Contexts

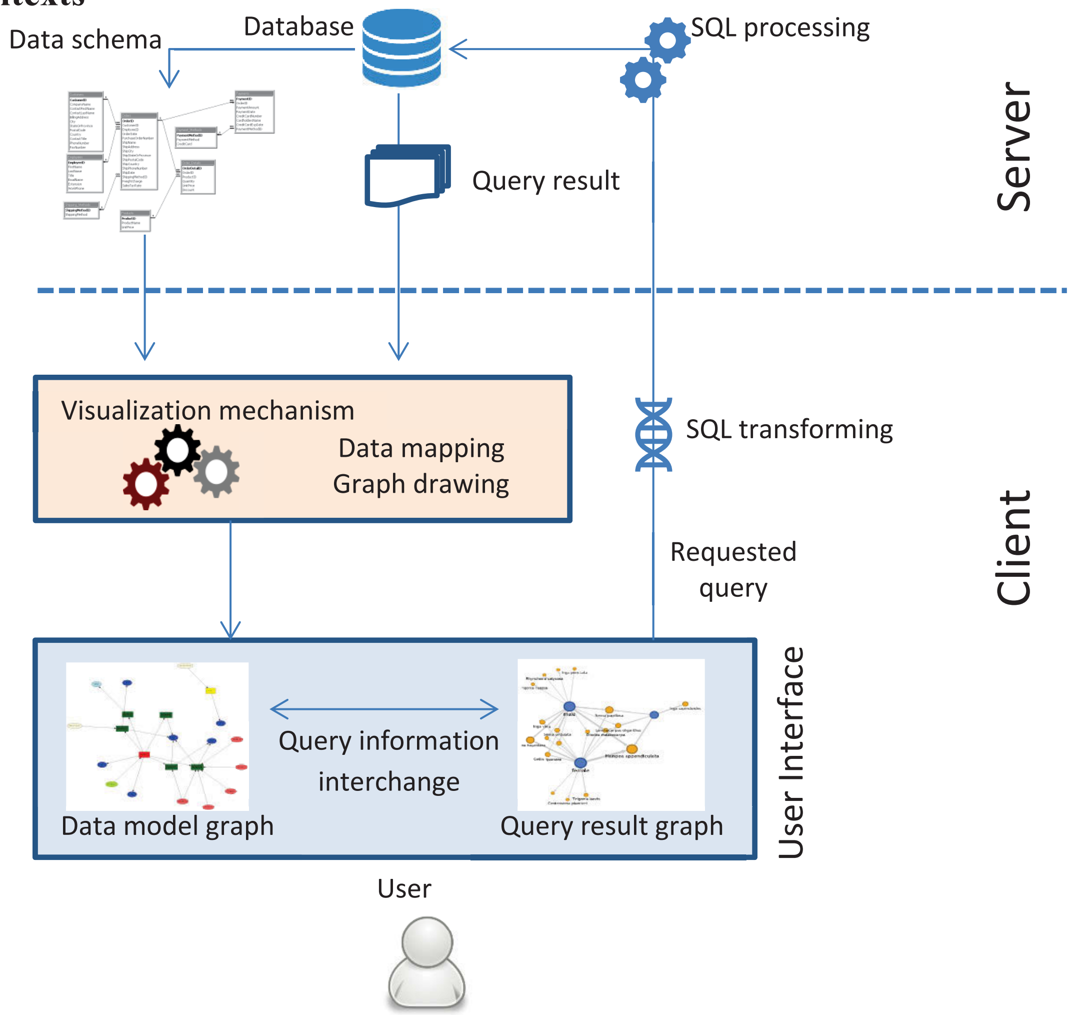


Figure 4.19 The proposed system framework for visual queries with the coordinating context views.

We propose a framework for the system implementation of visual exploration based on coordinating contexts. The system is a web-based application running on a server side with a database processor and a client side with query visualization mechanism. The server-side responsibilities are to extract the schema information, process SQL, and execute the queries while the client side is for SQL transforming, data mapping, visualization processing, and user-interface interaction performing.

The query information received from the user interface is transformed to SQL and submitted to the server for execution. The finding feature provides the dimension detail for *SELECT* statements whereas the condition feature supplies filtering elements for *WHERE* clauses. The operators implemented are *equal, greater than,* and *less than* ones. Besides, the connection elements corresponding to the data schema relationships are parsed into *WHERE* clauses, and involved tables are embedded into *FROM* clauses. The query can be executed in either concrete or relative ways. The relative queries reuse the previous results while the concrete ones run independently.

A standard SQL transformation is defined below with recalled Equation 4.3.

$$Qi = (F\ (f_1, f_2, f_3,\ldots,f_n), C(c_1, c_2, c_3,\ldots, c_m)), \text{ and } c_m = (f'_m, v_m),$$

Thus,

$$
\begin{array}{ll}
\textit{SELECT} & f_1, f_2, f_3, \ldots f_n \\
\textit{FROM} & t_1, t_2, t_3, \ldots t_n \\
\textit{WHERE} & \textit{con1 AND con2 AND con3 AND}\ldots \\
 & \qquad\qquad c_1 \textit{ AND } c_2 \textit{ AND } c_3 \textit{ AND } \ldots
\end{array}
$$

where $t_n \in T$.

The instances throughout this chapter use the Classic model database, a traditional and popular database of MySQL. The data include the enterprise data of classic-car retailers such as offices, employees, customers, products, etc.

All the images of the instances about MCquery are taken from the developed prototype of the technique. The visual demonstration is based upon D3.js, a JavaScript library with a wide range of visualization components for DOM (Document Object Model) manipulation (d3js 2017).

4.6 Summary

This chapter has introduced a new interactive visual technique for relational data exploration. The primary method is based upon the model for query interaction in the coordinating visual contexts of data models and query results. Our interaction and visualization are designed to support exploring relational data by visual queries in the step-by-step progress of incremental operations. In addition to using the data model information, users are enabled to directly employ the query results to make relative queries and adjust the exploration direction continuously without the query reformulation. The node-link graphs representing the query results with a zooming + filtering feature allow users to recognize related clusters and tuple relationships quickly. These patterns could be helpful for further query adjustment.

In summary, the contributions of this chapter are about the graphical user interface design in term of the direct manipulation of visual data elements in search-retrieval-based discovery. The core of the conception is the integration of relative-query interaction into node-link graph visualization. Our approach is distinct from the current works since they have not significantly considered the parallel usages of data models and query results for continuous discovery.

Chapter 5 New Interactive Visual Queries for Multi-Dimensional Data

Parallel coordinates and scatterplots are the visual techniques widely used for representing multivariate and multi-dimensional data. While parallel coordinates is well suited to provide the general display of a vast number of attribute values given by a large number of dimensions, scatterplots is a right choice in the detailed comparison of pairs of dimensions. One of the most considered issues in this area is how to quantitatively analyze the data density caused by polyline growth in the limited space of the parallel coordinate axes and resulted by dot increasing in the plotting space of scatterplots. The existing visualization techniques can successfully support the comparison of data summary among single dimensions and independent focus zones. Nevertheless, these might face challenges in complex analytics, which require the flexibility and multiplicity of such comparison.

This chapter introduces two new visual query techniques named SumUp, for the statistical analysis of the multiple attributes of dimensions with the multiple ranges of the polylines of parallel coordinates, and FigAxis, for the quantitative analysis of the focus zones of scatterplots. The methodology of SumUp is primarily based on developing dynamic queries by using brushing operation to deliver summary stacked bars on the parallel coordinate axes. Users can quickly retrieve quantitative information from the data patterns and flexibly make the multiple-attribute comparison with minimum effects of the polyline density. SumUp is able to enhance direct manipulation on parallel coordinates in term of statistical discovery and scalability. In FigAxis, zooming operation is adapted for data plot density measurement with dynamic stacked bars on the scatterplot axes. Users are enabled to quantitatively explore plotted data instances in the same space and at the same time of correlation observing.

5.1 Revisiting Manipulation on Multi-Dimensional Data

5.1.1 Parallel Coordinate Interaction

In most of cases, the interaction in parallel coordinates is operated for three common aims including information retrieval, view selection, and decision making.

The information retrieval refers to the activities of searching and expressing desired data through a series of related tasks. Conventionally, tabular-formed data and tabular-based widgets are often used as indirect components in data specification and the parallel coordinate input. Although the layouts are simple for implementation, they can cause the increase of users' mental effort in mapping raw data to the visual representation.

In order to reduce such interruption to user cognition, many latest techniques take direct interaction into account. The simplest one might be 2D-plane-based selection, which allows navigating polylines or instance patterns by direct drawing a rectangle zone on selected polylines (Siirtola & Räihä 2006) (see Figure 5.1).

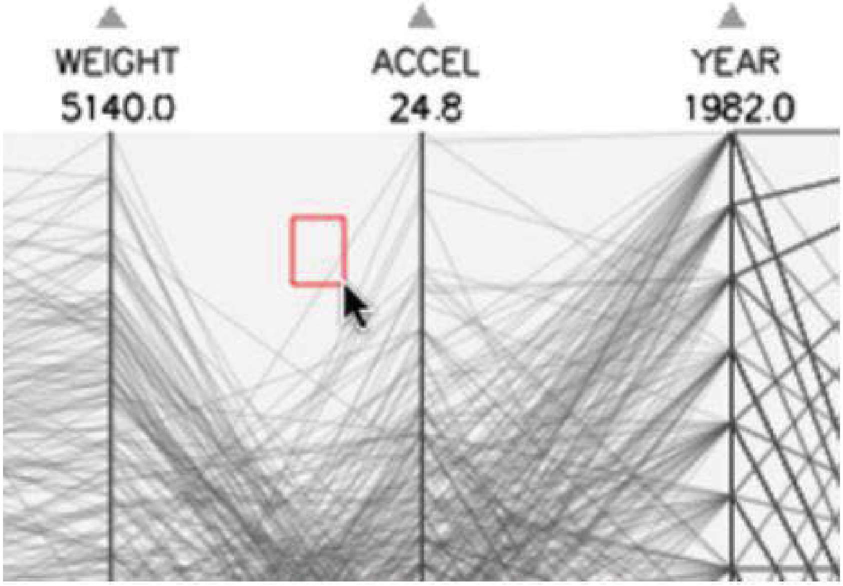


Figure 5.1 The direct manipulation with rectangle drawing (Siirtola & Räihä 2006).

In visual analytics, the interaction also refers to the process of interacting with data besides graphical user interfaces. A well-known method based upon data interaction for enhancing information display was proposed by Artero, de Oliveira and Levkowitz (2004) (see Figure 5.2). They used the characteristics of the frequency and density information extracted from data to compute the noise capability of high-density visualization and then attempted to minimize it. Thanks to this method, the polyline density can be reduced, and so it brings tidy visualization.

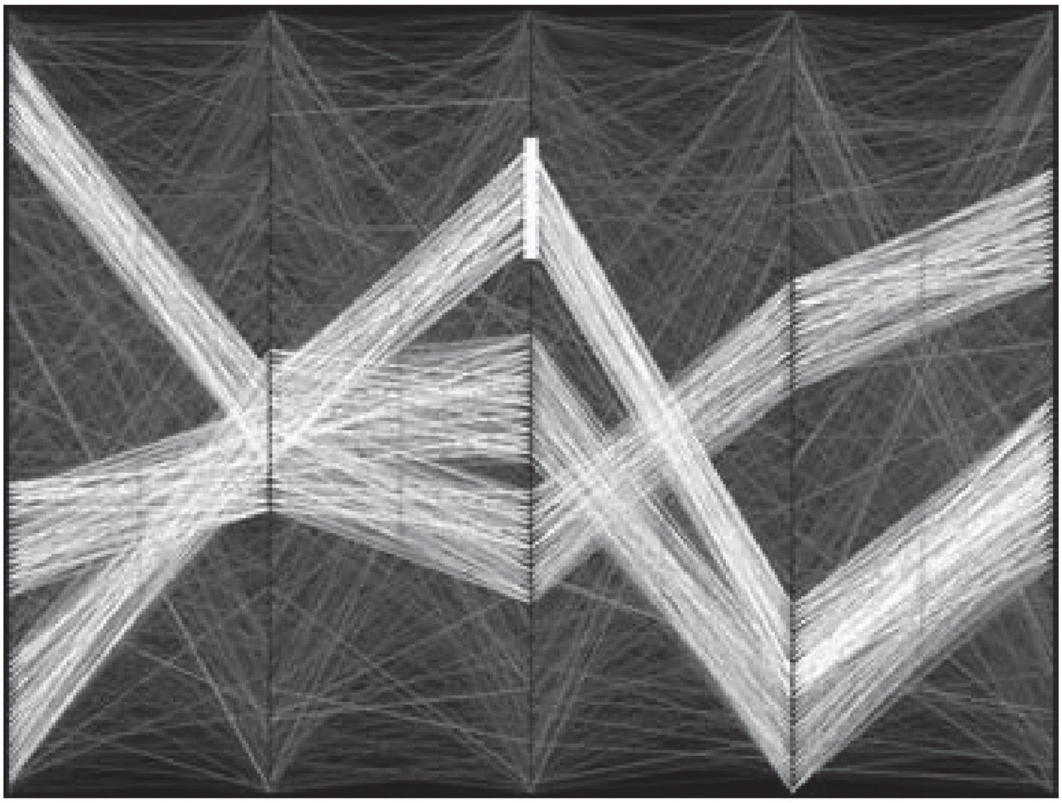


Figure 5.2 The density based filtering in parallel coordinates (Artero, de Oliveira & Levkowitz 2004).

For the interaction by user behaviours, focus+context is a favourite technique which can be found in various visualization, including parallel coordinates. Novotny and Hauser (2006) presented a model that can enable outlier detection at the same view of trend and detail patterns (see Figure 5.3). The fundamental idea is to scale dimensions flexibly and independently with respect to the current task.

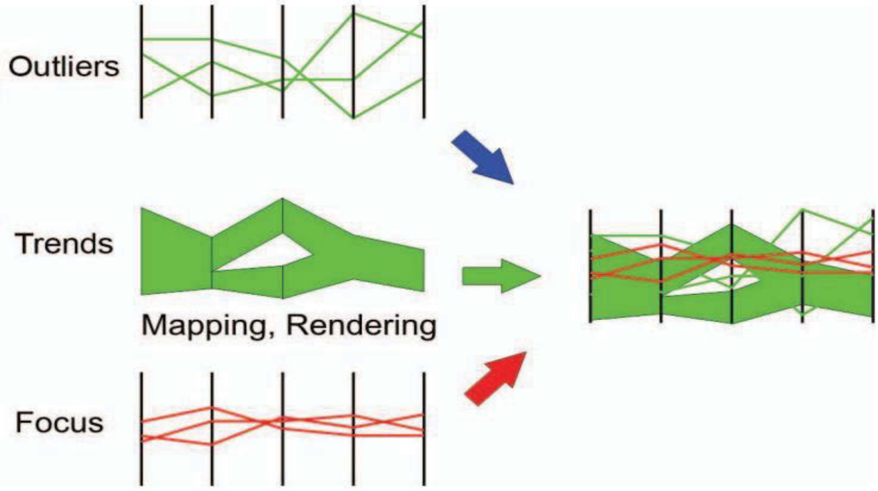


Figure 5.3 A focus+context visualization model in parallel coordinates (Novotny & Hauser 2006).

One of the greatest challenges in the parallel coordinate display is the representing a significant number of dimensions in a limited space. Dimensionality reduction is, currently, highly considerable topics with the attempt of using a minimum number of the original `dimensions to illustrate a maximum amount of the concerned information. Visual hierarchical dimensional reduction is one of the typical methods proposed for dealing with the challenge (Yang, Ward & Rundensteiner 2002) (see Figure 5.4). The basic idea is to hierarchically organize data by a dimensional tree which users can interact to customize the data input for usage.

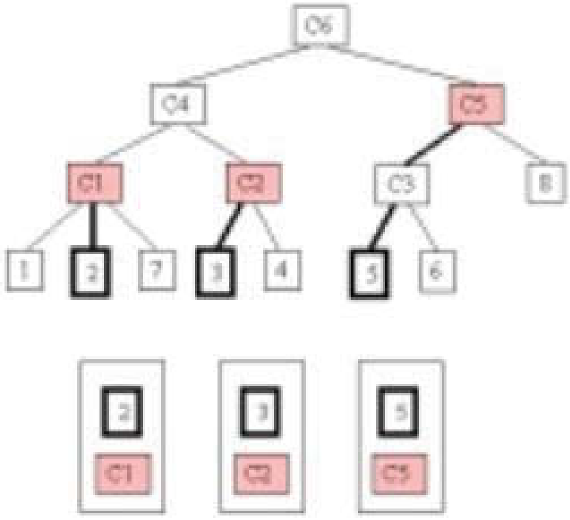
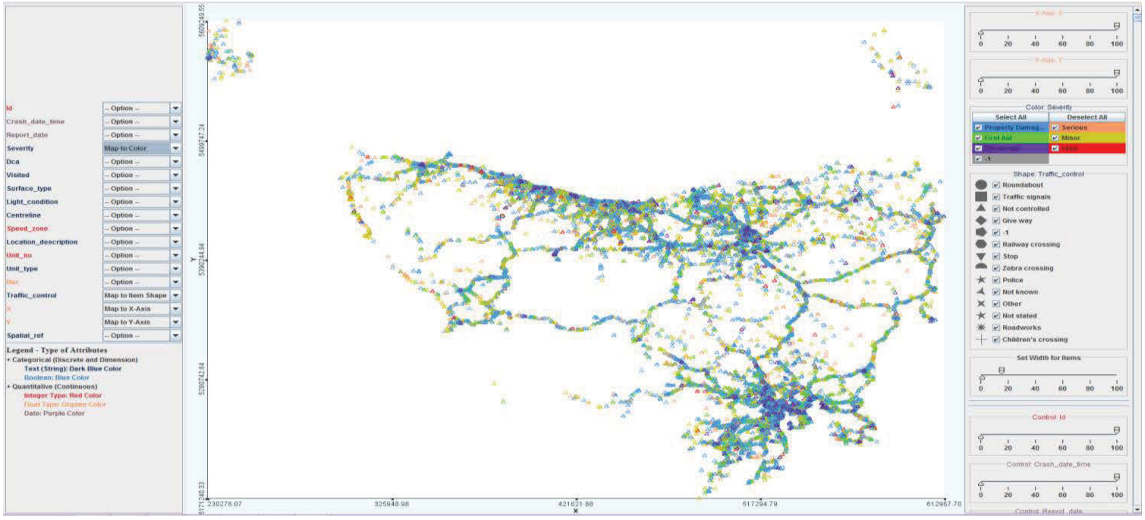


Figure 5.4 A dimensional tree of visual hierarchical dimensional reduction (Yang, Ward & Rundensteiner 2002).

5.1.2 Scatterplot Interaction

Scatterplots is usually visualized in either 2D or 3D display spaces. While the application of 3D models is complicated due to dimension specification, density increase, and unfamiliar navigation (Sedlmair, Munzner & Tory 2013), the use of 2D layouts is quite regular with friendly interfaces and conventional interaction, especially for non-spatial data (Tory et al. 2007; Tory, Swindells & Dreezer 2009). Since our design goal is for various target users including novices and nontechnical users, 2D models are chosen for examination. The following are the representative works concerning the interaction on 2D scatterplots.

According to Shneiderman (1996), the useful interaction which should be considered for a scatterplot interface might be filtering and zooming. These two techniques are very well-known for detail-on-demand exploration in a limited display space such as scatterplots. The uses of dynamic queries and brushing control are the standard approaches of scatterplot filtering. The former method concentrates on the use of widgets and control panels to create and adjust queries (Nguyen, Simoff & Qian 2016) (see Figure 5.5), whereas the latter focuses on algorithms for optimizing data point selection from clicks and brushes (Huang, Huang & Zhang 2012) (see Figure 5.6).



Id	-- Option --
Crash_date_time	-- Option --
Report_date	-- Option --
Severity	Map to Color
Dca	-- Option --
Visited	-- Option --
Surface_type	-- Option --
Light_condition	-- Option --
Centreline	-- Option --
Speed_zone	-- Option --
Location_description	-- Option --
Unit_no	-- Option --
Unit_type	-- Option --
	-- Option --
Traffic_control	Map to Item Shape
X	Map to X-Axis
Y	Map to Y-Axis
Spatial_ref	-- Option --

Legend - Type of Attributes
+ Categorical (Discrete and Dimension)
Text (String): Dark Blue Color
Boolean: Blue Color
+ Quantitative (Continuous)
Integer Type: Red Color
Float Type: Orgame Color
Date: Purple Color

Color: Severity

Select All	Deselect All
☑ Property Damag...	☑ Serious
☑ First Aid	☑ Minor
☑	☑ Fatal
☑ -1	

Shape: Traffic_control
- ☑ Roundabout
- ☑ Traffic signals
- ☑ Not controlled
- ☑ Give way
- ☑ -1
- ☑ Railway crossing
- ☑ Stop
- ☑ Zebra crossing
- ☑ Police
- ☑ Not known
- ☑ Other
- ☑ Not stated
- ☑ Roadworks
- ☑ Children's crossing

Figure 5.5 The scatterplots with dynamic queries (Nguyen, Simoff & Qian 2016).

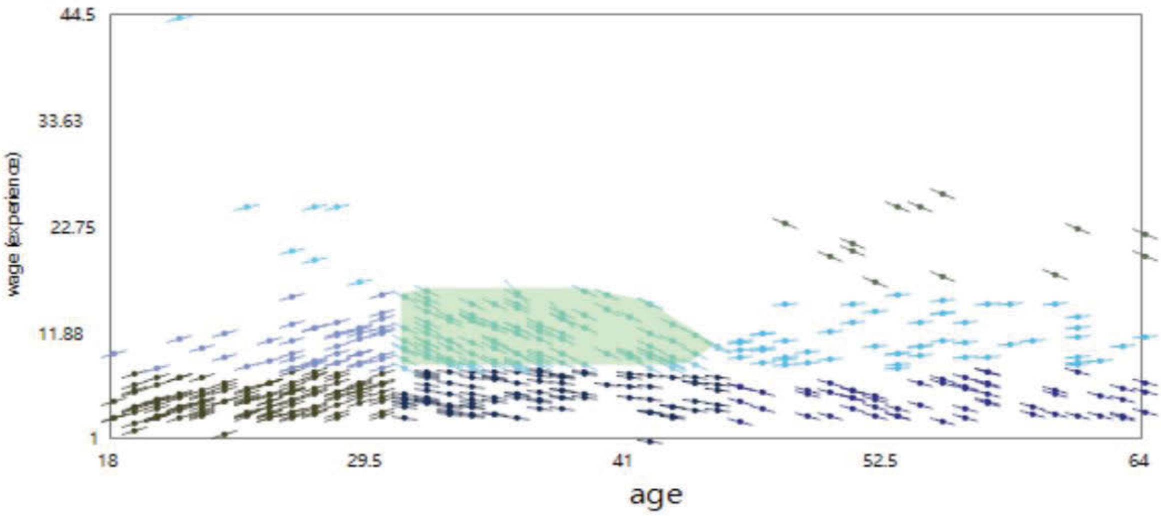


Figure 5.6 The scatterplots with data point selection optimization (Huang, Huang & Zhang 2012).

To deal with the point density growth in scatterplots, zooming by fish-eye based on points and axes is a common choice; yet, this kind of distortion is not proper for the overlapping caused by the growth of same values of one data point (Keim et al. 2010). Method *rolling the dice* is offered to reduce such overlapping by display space transformation and query interaction enhancement (Elmqvist, Dragicevic & Fekete 2008) (see Figure 5.7).

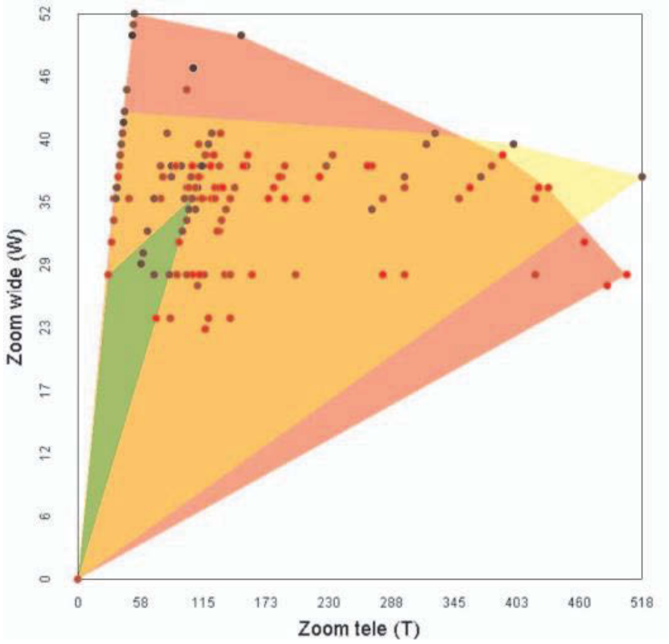


Figure 5.7 The scatterplots with display space transformation (Elmqvist, Dragicevic & Fekete 2008).

Furthermore, in order to improve the representation of data points in 2D space, a flow-based approach is proposed with using virtual Z dimension and local variation (Chan, Correa & Ma 2010; Huang, Huang & Zhang 2012).

All the revisited techniques allowed scatterplots to encode and display a large volume of information, which brings much help to users in analysis activities. As the scatterplot characteristics strongly support the relation discovery of data dimensions in pairs, it is widely available in many visual analytics tools such as R-Statistics (Muehlenhaus 2012), Matlab (Majumdar 2012), Tableau (http://www.tableau.com, 2017), and Polaris (Stolte, Tang & Hanrahan 2002).

5.2 Quantitative Visualization with SumUp

Quantitative visualization is traditional and representative in statistical visual analytics which specially handle quantitative data. In spite of the fact that there are diverse approaches in this area such as form-based (spreadsheet), point-based (scatterplots), line-based (slope graphs) approaches, etc., we revisit the highlights of the length-based approach, which is the focus of this dissertation.

The most well-known visualization of the length-based approach for quantitative data is bar charts. A bar chart refers to the chart using the bars located in the vertical or horizontal direction to encode quantitative data. The bars can be placed in either the same basic line or different basic lines and arranged in either the same direction or diverging directions with individual or grouped positions (see Figure 5.8). While the use of the same basic line is appropriate for the comparison of single targets, the other one is widely used for the targets in pairs.

The method based on different basic lines or multiple basic lines is preferred for multiple attribute comparison since it enables multiple analyses across items. In other words, the method is the flexible combination of multiple same-basic-line charts (see Figure 5.8).

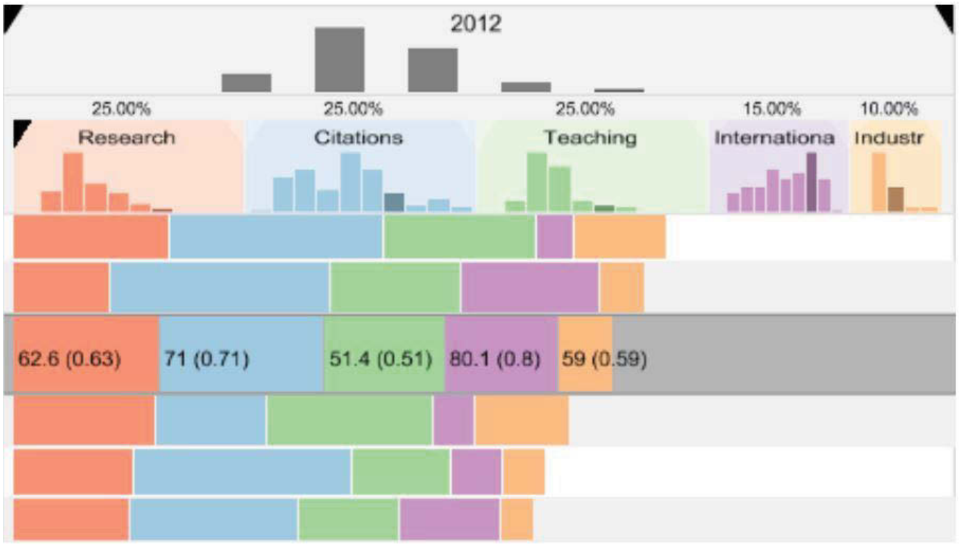


Figure 5.8 The bar and stacked bar layout with various arrangements for visual rankings (Gratzl et al. 2013).

In the case of a multiple-basic-line graph simplified by removing basic lines and putting the bars up together, a stacked bar chart is established. TableLens, a typical technique in this field, is to construct an adjustable interface by multiple-basic-line bar charts in which users can summarize and analyse the attribute values of both single and grouped comparison (Rao & Card 1994) (see Figure 5.9). This method enables statistical analysis inside and across dimensions. Additionally, it can be effectively combined with other charts such as slope charts for optimizing the usage of visual ranking details (Gratzl et al. 2013).

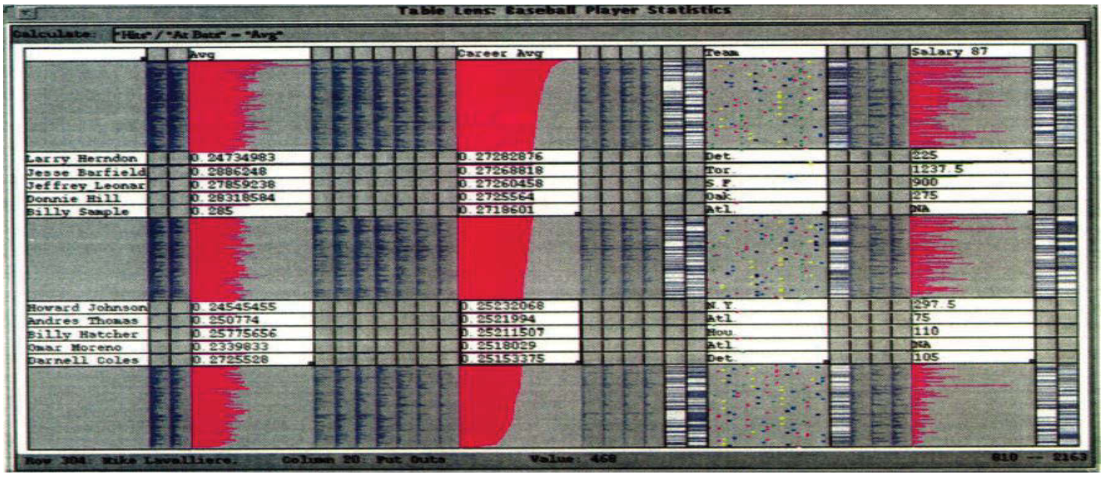


Table Lens: Baseball Player Statistics

Calculate: "Hits" / "At Bats" = "Avg"

	Avg	Career Avg	Team	Salary 87
Larry Herndon	0.24734983	0.27282876	Det.	225
Jesse Barfield	0.2886248	0.27268818	Tor.	1237.5
Jeffrey Leonar	0.27859238	0.27260458	S.F.	900
Donnie Hill	0.28318584	0.2725564	Oak.	275
Billy Sample	0.285	0.2718601	Atl.	NA
Howard Johnson	0.24545455	0.25232068	N.Y.	297.5
Andres Thomas	0.250774	0.2521994	Atl.	75
Billy Hatcher	0.25775656	0.25211507	Hou.	110
Omar Moreno	0.2339833	0.2518029	Atl.	NA
Darnell Coles	0.2725528	0.25153375	Det.	105

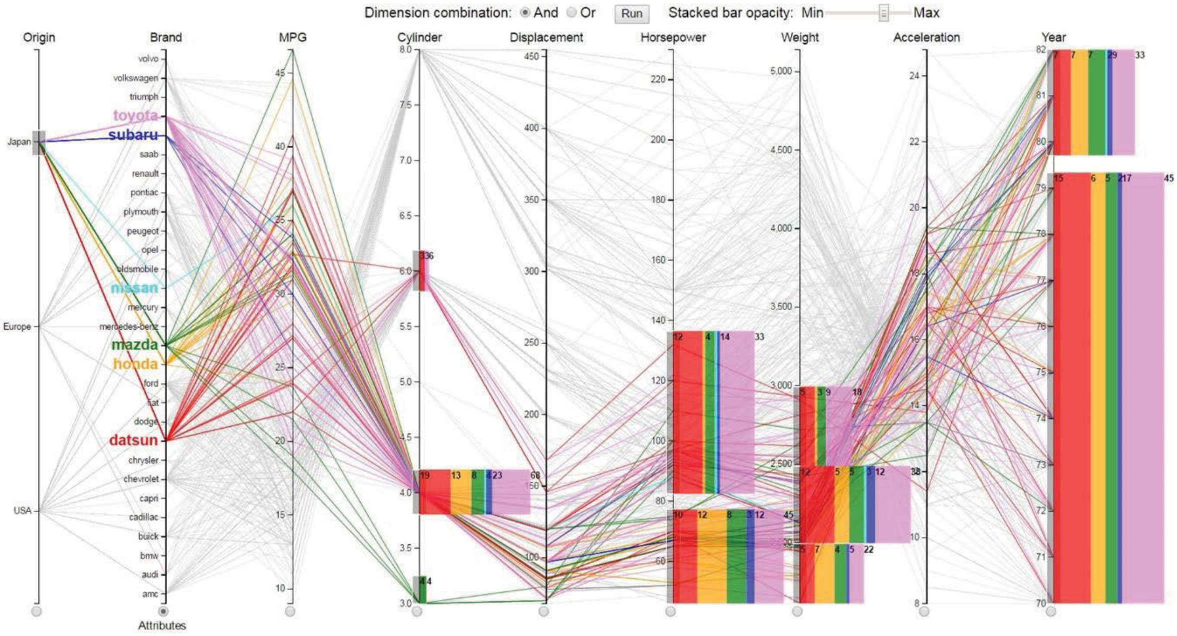
Row 304: Mike Lavalliere. Column 20: Put Outs Value: 468 810 -- 2163

Figure 5.9 The TableLens layout (Rao & Card 1994).

In our approach, we employ the stacked-bar ideas from the literature as a key guide in the design of visual and functional components for our stacked bars to be adapted for the parallel coordinate visualization. The remaining methods are not proposed to integrate with parallel coordinates because their features might be not proper for saving space and reducing density in the considered circumstances. The further sections will illustrate how the stacked bars are visually combined with the original parallel coordinates for visual exploration and analysis.

5.2.1 Double Layer Views

SumUp is an interactive visual query technique developed for the statistical comparison of multi-dimensional data on parallel coordinates. The technique is based on the interactive stacked bars embedded on querying axes to encode the quantitative data. By using the interface of the design, users are able to perform the statistical comparison inside and across the dimensions with multiple attributes and flexible ranges of traversing polylines.



Dimension combination: ● And ○ Or Run Stacked bar opacity: Min Max

Figure 5.10 The main layout design of the SumUp user interface.

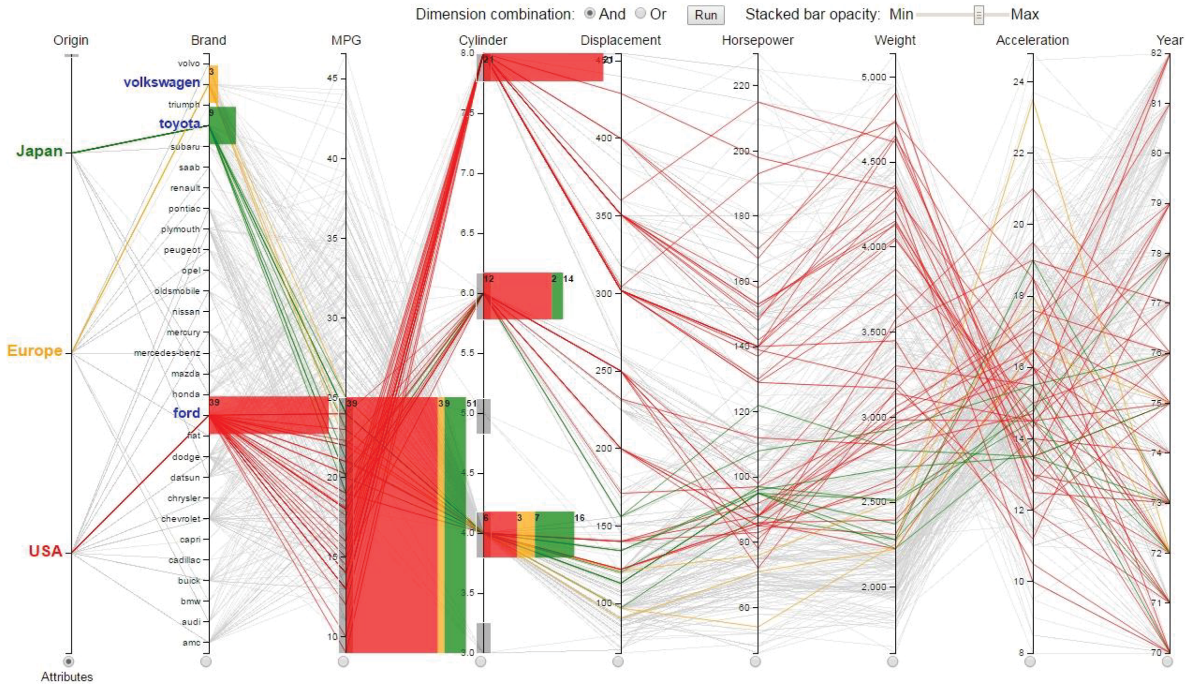
The primary design of SumUp includes the two overlayed layers of the visualization of parallel coordinates and stacked bars, where the parallel axes are the basic lines of the stacked bars and the components for the attribute and range selection (see Figure 5.10). Whereas the parallel coordinates plays the role of a main browser in the ground layer, in the front layer the horizontal stacked bars encode the statistical results associated with the selected ranges on each axis. The strength of this method is to utilize the vertical space between the parallel axes and to allow users to trace the statistical results directly in the same view of the data browsing.

The length of a total bar represents the total number of polylines based on all the target attributes, and that of each stacked bar encodes the number of polylines given by a corresponding attribute. The height of the stacked bars encodes the range size of the traversing polylines selected in an axis as the input of queries. The size of a bar is proportionally represented to the associated statistical values. As a consequence, these stacked bars can show the statistical results towards the multiple attributes of the dimensions, which enriches the amount of information encoded in the double layer display of the parallel coordinates.

5.2.2 Parallel Coordinate and Stacked Bar Integration

In the character of the casual employment of stacked bars, the size, colour, and data label representing the attribute value summary are the visual comparison principles of SumUp. The height of the stacked bars is equivalent to the height of the associated traversing-polyline ranges, and the length of the total stacked bars encodes the total number of the associated traversing polylines to be satisfied the involved target attributes.

The colour variety with the defaulted scheme of ten and twenty categorical colours of d3js (2017) assigned to the stacked bars depends upon the number of the target attributes being used. Coloured labels and coloured brushes are corresponding to the graphic legend of the attribute values and the traversing-polyline ranges.



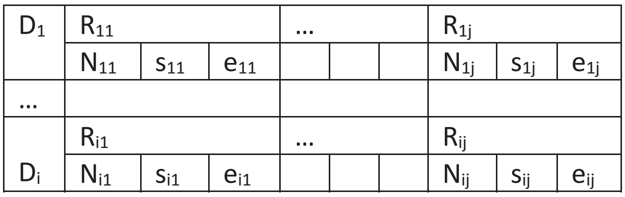
Dimension combination: ● And ○ Or Run Stacked bar opacity: Min Max

Figure 5.11 An instance of the SumUp query comparing the car models of three representatives *Toyota of Japan (green), Volkswagen of Europe (orange),* and *Ford of USA (red)*.

Figure 5.11 shows a query example that compares the car models of three representatives from *Japan, Europe*, and *USA* including *Volkswagen, Toyota,* and *Ford* regarding 25-and-lower *MPG* (Miles per gallon) and all cylinder numbers. To perform the query, we select all the origins as target attributes, select the three brands, then select the associated ranges of traversing polylines as the requirement on axes *MPG* and *Cylinder*. The result pattern indicates that there was a significant difference of the total number of delivered models about the origins. The majority of the models were owned by *Ford*, with 39 ones whereas the minority of that belonged to *Toyota* and *Volkswagen*, with 9 and 3 ones. While all of the brands employed 4-cylinder engines, only *Ford* employed 8-cylinder engines, and only *Volkswagen* did not install 6-cylinder engines for their cars with the fuel economy for *MPG* 25 and lower.

5.2.3 Data Mapping for Visualization

SumUp extracts the geometric and graphical data from a database storing the information of parallel coordinates and stacked bars. Once the interaction on the graph is activated by user activities, the related visualization details are recorded and organized in a data matrix exclusively designed for the stacked-bar representation in integrating with the parallel coordinates. One of its dimensions contains the axis indices, and the other contains the indices of polyline ranges and the number of polylines with their associated attributes (see Figure 5.12). The polyline range indices are mapped to the clustered colours and used to serve navigation procedures.

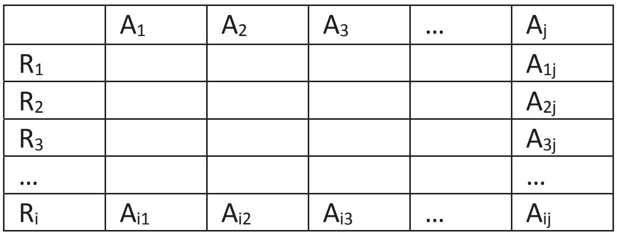


D_1	R_{11}			…			R_{1j}		
	N_{11}	s_{11}	e_{11}				N_{1j}	s_{1j}	e_{1j}
…									
	R_{i1}			…			R_{ij}		
D_i	N_{i1}	s_{i1}	e_{i1}				N_{ij}	s_{ij}	e_{ij}

Figure 5.12 The data structure for considered polyline ranges.

In the structure shown in Figure 5.12, *D* and *R* are the sets of the dimensions and ranges of selected polylines, N_{ij} is the number of the selected polylines in associated range j^{th} of dimension i^{th}, and s_{ij} and e_{ij} are the starting and ending points of selected range j^{th} of dimension i^{th}. Thus, $|s_{ij} - e_{ij}|$ is the size of the selected range for numerical dimensions. For categorical dimensions, s_{ij} and e_{ij} are one value.

Once a query is executed, its result is transferred to a structure formed for stacked bar representation. The output of the display depends upon the number of target attributes with the statistical summary. Figure 5.13 illustrates the data structure used for handling the stacked bar visualization on the parallel axes.



	A_1	A_2	A_3	…	A_j
R_1					A_{1j}
R_2					A_{2j}
R_3					A_{3j}
…					…
R_i	A_{i1}	A_{i2}	A_{i3}	…	A_{ij}

Figure 5.13 The data matrix for stacked bar representation.

In Figure 5.13, $A_j \in A$ is the j^{th} set of the target attributes, and a_{ij} is the statistical result corresponding to range R_i with A_j.

In addition to the automatic mapping function, the customized data selection component is featured for users to choose desired data flexibly for visualization and analysis, and it can help users to specify what attributes are concentrated on the selected dimensions as well.

The data sets used throughout this chapter are a Car data set with nine dimensions and 398 instances and a set of the Census income data with fifteen dimensions and 9998 instances (Lic 2013).

5.2.4 Discussion

We have proposed a quantitative visualization method based upon double layer views, the integration of stacked bars and parallel coordinates, and the mapping for geometric and graphical data. This method enables the parallel coordinates to encode the instance summary in the same view and at the same time, which can optimize the display space between the parallel axes and enrich manipulation on such a user interface. This graphical user interface plays the role of being the major base to a data browser on which further functional components would be deployed.

5.3 Query Interaction on Parallel Coordinates with Quantitative Approach

Parallel coordinates visualizes multi-dimensional data by a view containing parallel axes and traversing polylines, which might cause the challenges of understanding and interpreting the pattern meaning to users (Siirtola & Räihä 2006). Applying the appropriate interaction on parallel coordinate browsers is able to improve the effectiveness of visual data mining and analysis. Thus, a number of interaction techniques have proposed. Most favourite manipulations are about brushing, ordering, and scaling (Heinrich & Weiskopf 2013). Brushing enables users to select and highlight considered instances by drawing and dragging the selected zones on polylines and the focus ranges on axes (Martin & Ward 1995; Fua, Ward & Rundensteiner 2000; Hauser, Ledermann & Doleisch 2002). This method allows the parallel-coordinate browser to be more adaptable with dynamic views for effective navigation. While the brushing targets to the data instance highlight, the ordering and scaling help reduce the overlapping and adjust the density opacity of polylines (Andrienko & Andrienko 2001; Lu, Huang & Huang 2012). In parallel coordinates, the correlative comparison of the dimensions is much considered; however, one axis is not always placed next to the others for observing such comparison, which causes the increase of users' effort to trace the correlative patterns and match the details together. Method ordering, one of the simplest ways to deal with the problem, is to reorder the axes automatically or manually for flexible views of analytics. Another challenge in the area is about the increase of the number of the traversing polylines through the axes. Actually, a large number of intersection points displayed in a limited range can make plenty of clutter in analysis views. Dimension zooming is a regular way for polyline counting support, but not suitable in the case of overlapped polylines (Fua, Ward & Rundensteiner 1999). The visual scalability of such clutter can be improved by hierarchical clustering with similarity-based computation on dimensions and data instances (Fua, Ward & Rundensteiner 1999; Andrews, Osmić & Schagerl 2015).

Although the mentioned methods accomplished handling a large volume of data, they have not substantially concentrated on quantitative comparisons. Siirtola (2000) and Ho et al. (2011) proposed the models applying box plots and bar charts for quantitative mining of the polyline density measurement (see Figure 5.14). Nonetheless, the techniques were designed for the analysis based on single dimension attributes and single ranges of polylines, which might not fulfil the sophisticated analytics requiring the multiple-attribute and flexible-range comparison.

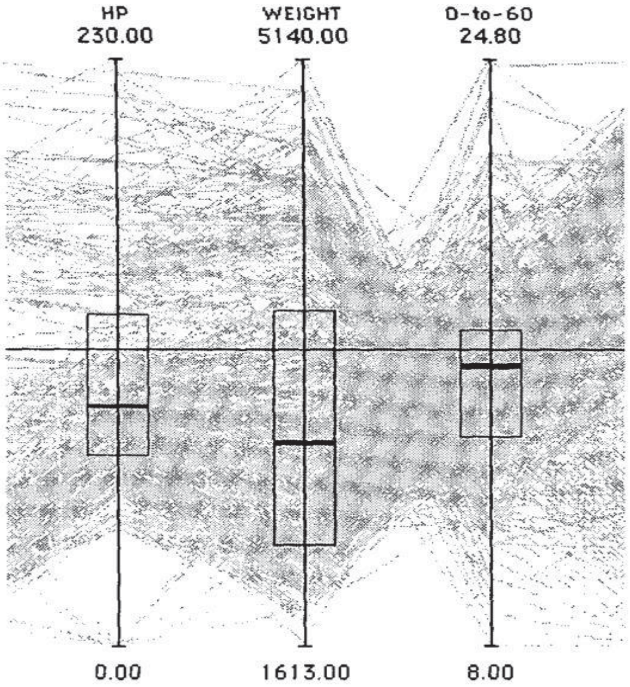


Figure 5.14 The parallel coordinates with box-plot embedded for data instance summary (Siirtola 2000).

The following sections present a visual query technique for the statistical analysis on parallel coordinates that enables users to summarize the number of data instances by dynamic queries and flexible density measurement. The methodology is to design brushing-based queries embedded on the parallel axes and to encode the statistical results by the interactive stacked bars. By employing this method, users can not only make the statistical comparison quickly but also navigate the desired clusters of the instances easily.

5.3.1 Quantitative Visual Query Making by Brushing

SumUp query components are designed to be embedded on the parallel-coordinate axes for data discovery manipulation. A query is created by the conventional brushing operation. The brushing operation is adaptive with multiple brushes for selecting the multiple ranges of the traversing polylines or intersection points to be involved in the query. The function of the query is to compute the number of the traversing polylines through all the axes grouped by a set of target attributes.

The formal definition of SumUp query function is described as follows.

• R_k is the axis of the target attributes; $D' \subset D$ and $D' \not\ni R_k$, where D and D' are the sets of all dimensions and activated dimensions respectively.

• $A = \{a_i|i=1,2,\ldots n\}$ is the set of the attributes selected from axis k, and a_i is either a single value (one intersection point) or a range of values (a set of intersection points).

• $R_q = \{r_{qj}|j=1,2\ldots m\}$ is the set of the traversing-polyline ranges on axis q $(Rq \in D')$, and r_{qj} is either a single value or a range of values.

Functions $S_q(A, R_q)$ and $S'_q(A, D')$ are to compute the statistical results by operators *Or* and *And*. The computation by operator *Or* considers the polyline ranges in R_q satisfying A, whereas that by operator *And* takes both A and D' into account as the satisfactory condition for R_q.

Thus, the length of the total bar at range r_{qj} is computed by

$$l\,(r_{qj}) = \sum_{i=1}^{n} S(a_i, r_{qj}) \text{ and } l'(r_{qj}) = \sum_{i=1}^{n} S'(a_i, D')$$

Equation 5.1

where l and l’ are the lengths resulted with operators *Or* and *And*.

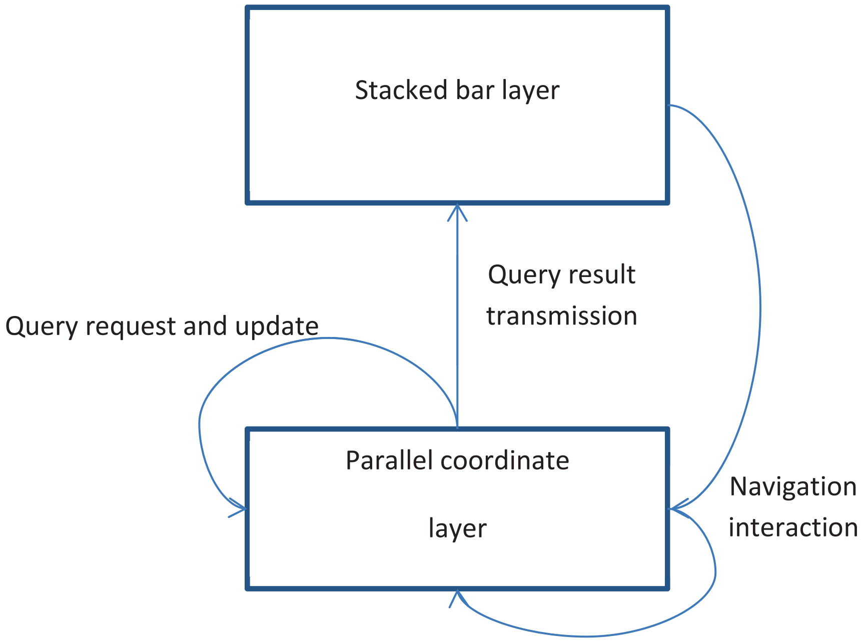
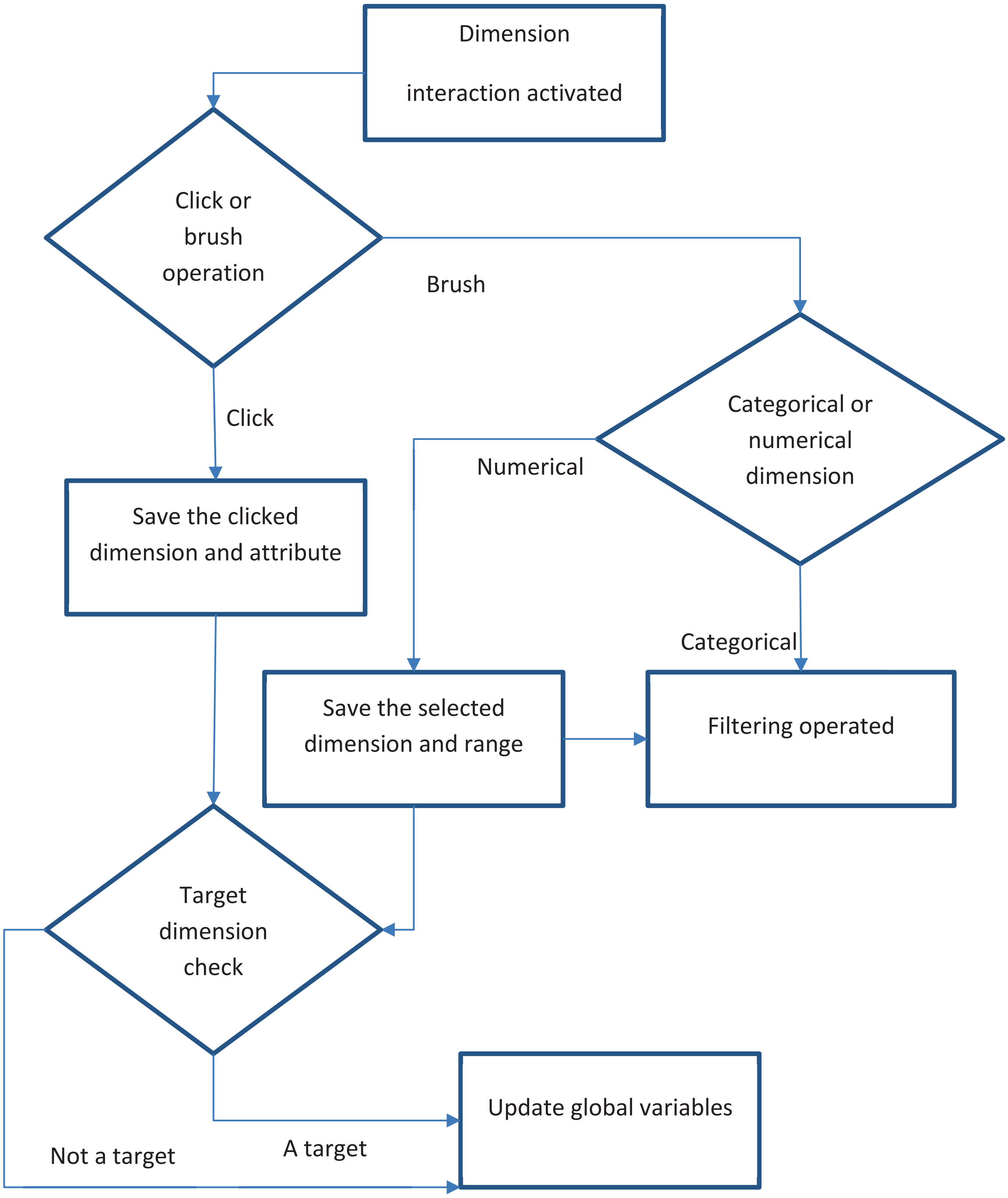


Figure 5.15 The interaction model of query interaction on the double layer views. The parallel coordinate layer is the background for query requests and query updates while the stacked bar layer is for statistical result representation. Users can navigate the polylines from both the layers.

Our query interaction process is probably generalized by the model in Figure 5.15. In order to create and adjust a query, users can directly click and brush on the representation of the attribute values. Numerical values can be chosen by both clicking on a single value and brushing on a range of values, whereas categorical values can be selected by only the clicks on ordinary values (see Algorithm 5.1). Once a target dimension and its values are established, the query function posts the request for the query execution. Then, the query result is represented in the stacked bar layer (see Figure 5.15). Users can navigate desired instances (polylines) and considered stacked bars by the interaction on both the stacked bars and the parallel coordinates. The procedure of recording the associated interaction on the dimensions and their attributes in the query creation and update is described in Algorithm 5.1.



Algorithm 5.1 The procedure of recording the interaction on the parallel coordinate dimensions.

The procedure of displaying the query results for the stacked bar interaction is illustrated in Algorithm 5.2.

```
procedure representingStackedBars(global dimensions)
    Targets = dimensions.targets
    for each Dim in dimensions
        if activated(Dim) && Dim != Targets then
            for each Range in Dim.Ranges
                for each Tag in Tagets
                    if statisticOf(Range, Tag) > 0 then
                        Draw(Range, Tag)
                    end if
                end for
            end for
        end if
    end for
end procedure
```

Algorithm 5.2 The procedure of displaying the query results for the stacked bar interaction.

5.3.2 Discussion

Our interactive visual query technique is well suited for the statistical analysis on parallel coordinates in term of enhancing direct manipulation and scalability. We have found the two techniques supporting the statistical functions on parallel coordinates including Siirtola (2000) and Ho et al. (2011); however, they have not greatly focused on the comparison of the multiple attributes and the flexible ranges of traversing polylines. Therefore, the statistical queries conducted by our method in this dissertation would be very hard to be resolved by those two.

We have introduced the SumUp query technique which is developed for the statistical discovery associated with parallel-coordinate geometries. The technique enables users to perform statistical queries by dynamic visual representation. Our approach is to build interactive stacked bars embedded in the parallel axes to encode and represent the statistical results of multiple attributes and to equip dynamic queries with the standard brushing operation. Thanks to the approach, users are able to analyse the summary of the polylines arbitrarily with multiple dimension attributes inside and across the axes.

5.4 Query Interaction on Scatterplots by FigAxis

5.4.1 Quantitative Visual Queries by Zooming

Almost the current scatterplot applications enhance the navigation function by increasing the number of representation views (see Figure 5.16). The focus zone on the main plot is synchronously shown in other views in different facets or different scales, which allows users to explore data details in various aspects.

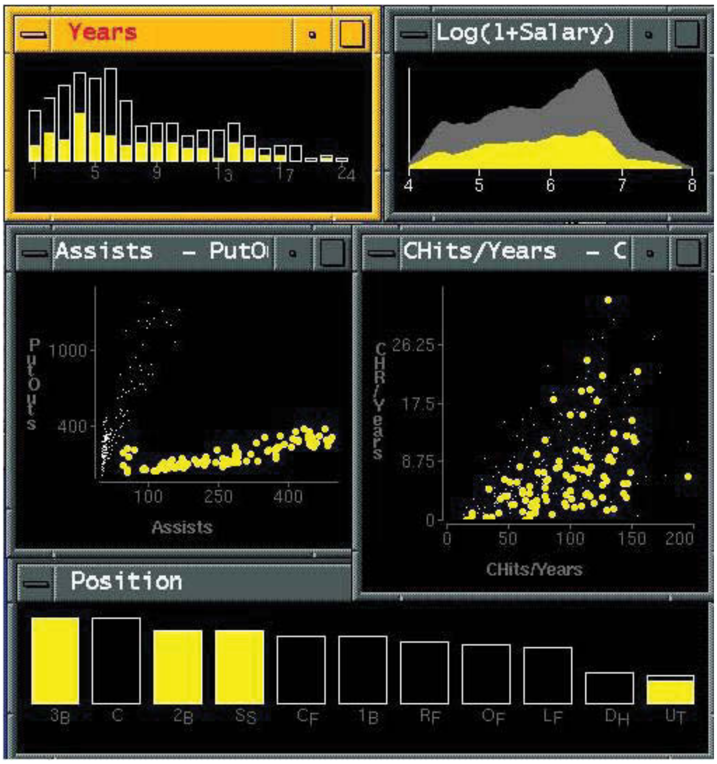


Figure 5.16 The linked views of scatterplots with various graphs (Heer, Shneiderman & Park 2012).

Nevertheless, using the multiple views might face the limitation of matching the corresponding information of diverse views together. Figure 5.17 illustrates the procedure of the seeking information in the multiple views.

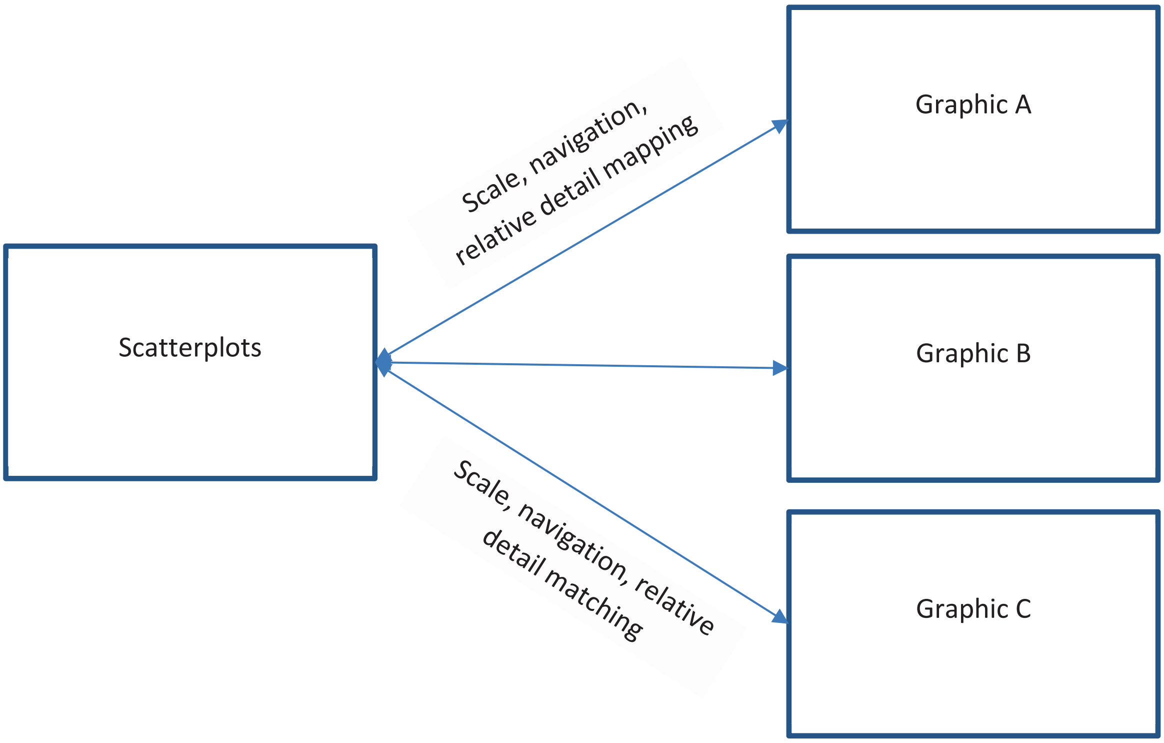

Scatterplots
Graphic A
Graphic B
Graphic C
Scale, navigation, relative detail mapping
Scale, navigation, relative detail matching

Figure 5.17 The procedure of multiple-view matching between scatterplots and other graphics.

For the multiple view interaction (see Figure 5.17), users are required to repeat the procedure of mapping and matching different scales, navigation, and other relative details many times, which causes the increase of users' cognitive effort in visual exploration. For overcoming the challenge, we offer an interactive visual method, named FigAxis, which displays statistical charts in the same space of the scatterplot representation. The approach enables the quantitative queries to be performed by zooming operation on the dimension axes. The overview of FigAxis design is illustrated in Figure 5.18.

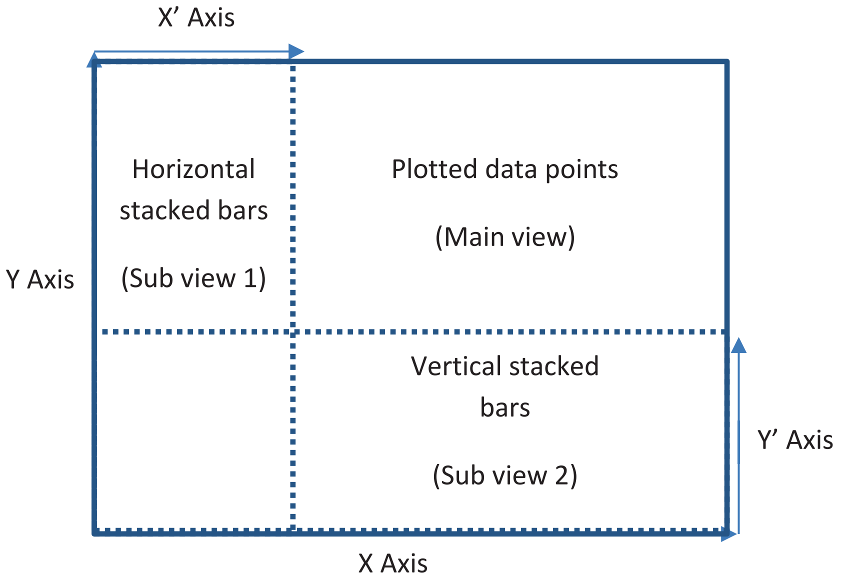


Figure 5.18 The overview of the FigAxis layout design.

According to the design (see Figure 5.18), the core of this visual technique is the reuse and share of X and Y axes of the scatterplots in order to play the role of the baselines of the statistical stacked bars. The design method would help reduce the users' cognitive effort in the mapping and matching corresponding information between different graphics during the exploration. The horizontal and vertical stacked bars encode the data plot density of an activated zone. The focus zone of plotted data is activated by the zooming and panning components on both the main view and the two sub views.

The length of the horizontal bars and the height of the vertical bars represent the number of the data instances plotted in the associated zone towards target attributes. The height of the horizontal bars and the width of the vertical bars are equivalent to the activated zone which plots the data. Figure 5.19 illustrates the main layout of FigAxis.

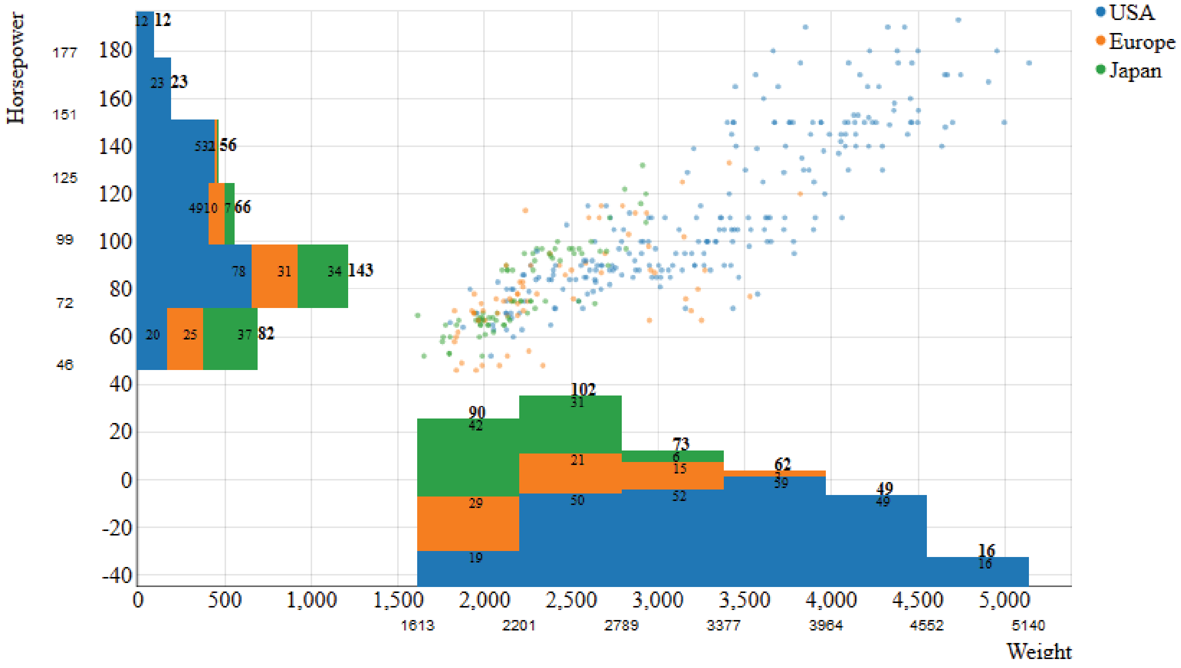


Figure 5.19 The layout of a FigAxis application for the correlative comparison of new car model delivery in term of *Horsepower* and *Weight* with the targets of original brands from *USA, Europe,* and *Japan.*

The query by zooming in Figure 5.19 is adjusted at 6 value ranges. It is clear from the figure that most of the models were from *USA*, with all of the ranges of *Weight* and *Horsepower*, whereas the remaining models were from *Europe* and *Japan*, with centring at the average and low *Weight* and *Horsepower*. In overall, this was highly possible for a positive correlation, and the majority of the models were focused on the average and low segments of *Weight* and *Horsepower*. There was no significant difference of the number of the delivered models towards the weights lower *2200* and the horsepower lower *72* (Hp). For *Weight* ranges, while *Japanese* brands occupied nearly half of *90*, with *42* models, *Europe* and *USA* contributed *29* and *19* models respectively. On the other hand, there was a great difference of the number of car models in the *Weight* ranges higher *2200*. *USA* cars appeared the most, exclusively for the ranges over *4000*. Similarly, for *Horsepower* perspective, *USA* cars occupied the most in the ranges over *72*, with the highest at the range *72-99*, exclusively for the ranges over *150* (Hp).

For the plotted data point measurement, we offer a statistical query based upon the zooming levels. In other words, the more the zooming in gets, the more the detailed measurement is retrieved, and the more the zooming out gets, the more the general measurement is retrieved. For numerical and linear data, the query *Q* is formally defined as follows (see Figure 5.20 & Equation 5.2).

The zooming levels are calculated by the change of the tick range size compared to the data display.

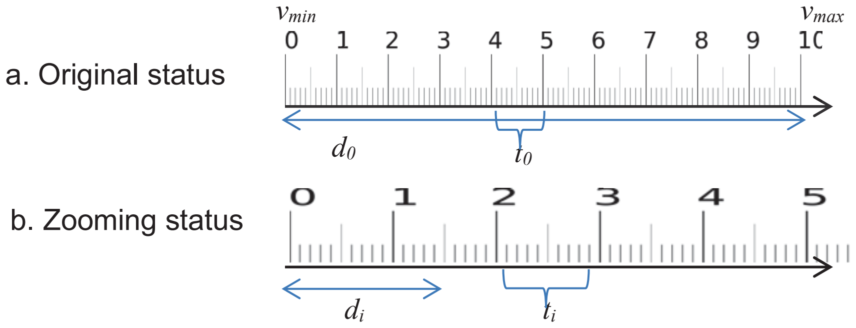


Figure 5.20 The zooming level measurement background.

$$Q_i = S(A, R),\ R = \{r_1, r_2, r_3, \ldots r_n\},\ A = \{a_1, a_2, a_3, \ldots a_p\},$$

$$d_i = \frac{v_{max} - v_{min}}{l_i}$$

Equation 5.2

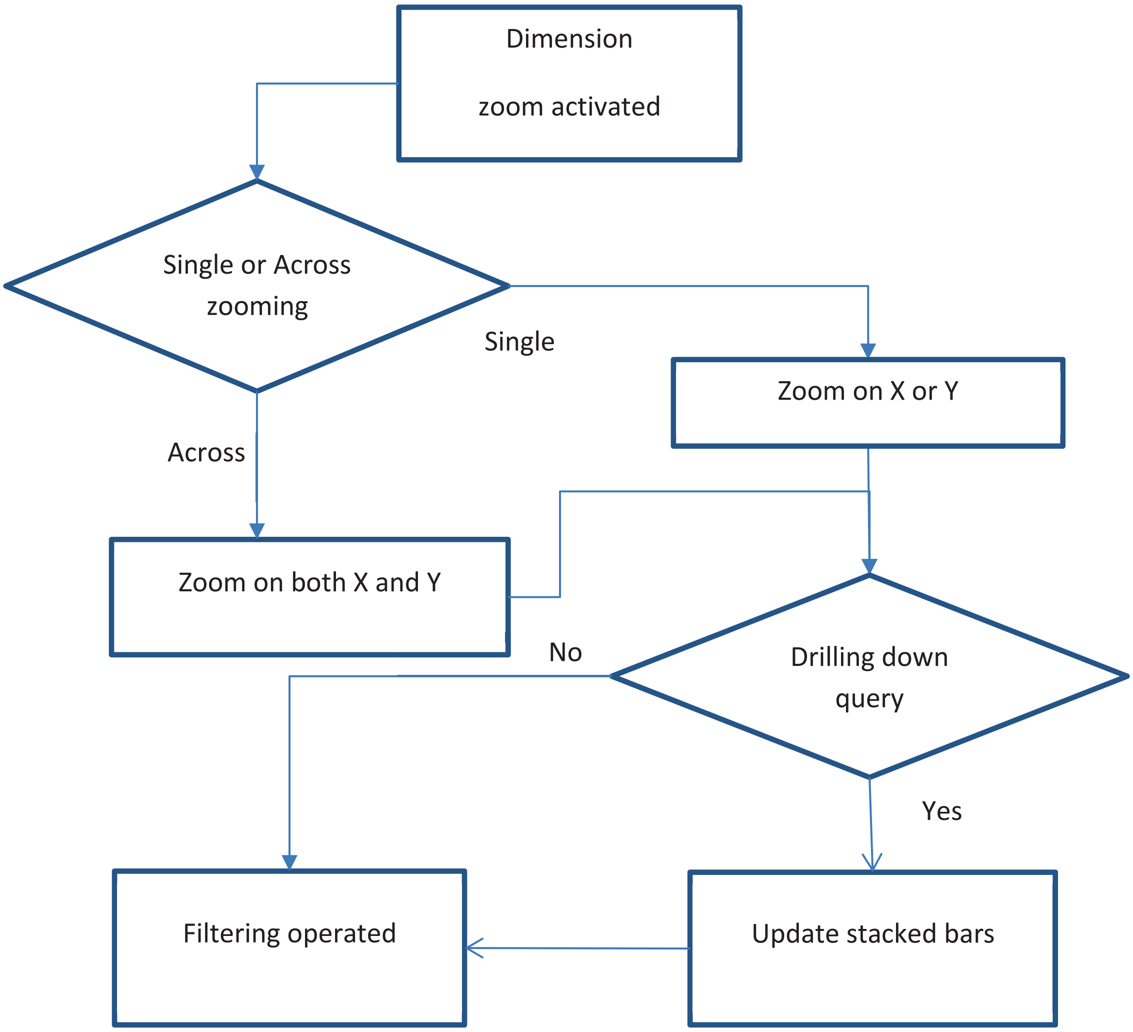
$$l_i = m\frac{t_i}{t_0}$$

Equation 5.3

In Equations 5.2 and 5.3, *A* is the set of target attributes, a_p is either a categorical value or a discrete value, *R* is the set of even ranges d_i of axis-plotted data, *S* is the statistical function to compute the data summary of R satisfying A, v_{max}, v_{min} are the maximum and minimum data values plotted by the axis, and l_i is the current zooming level computed by customized parameter m multiplied by the ratio between current tick range size t_i and default one t_0.

For categorical data, $Q = S(A, R)$, where R is the set of all categorical values, and Q is not impacted by l.

The procedure of the interaction on FigAxis with the zooming-based query is illustrated in Algorithm 5.3 below.



Algorithm 5.3 The procedure of recording the interaction on FigAxis.

5.4.2 Data Mapping

Our interactive scatterplots is designed to support up to five dimension encoding. Thus, in the data mapping procedure, users are able to configure the considered dimensions for favourite tasks with the data types supported as follows (see Table 5.1).

Data Encoding	Numerical	Categorical	Target
X, Y	✓	✓	
Shape		✓	
Colour		✓	✓
Size	✓		

Table 5.1 The FigAxis data support summary.

Since the data summary for the target attributes is represented in the stacked bar visualization, only the categorical dimension is applied as the query target.

5.5 Visual Enhancement Methods

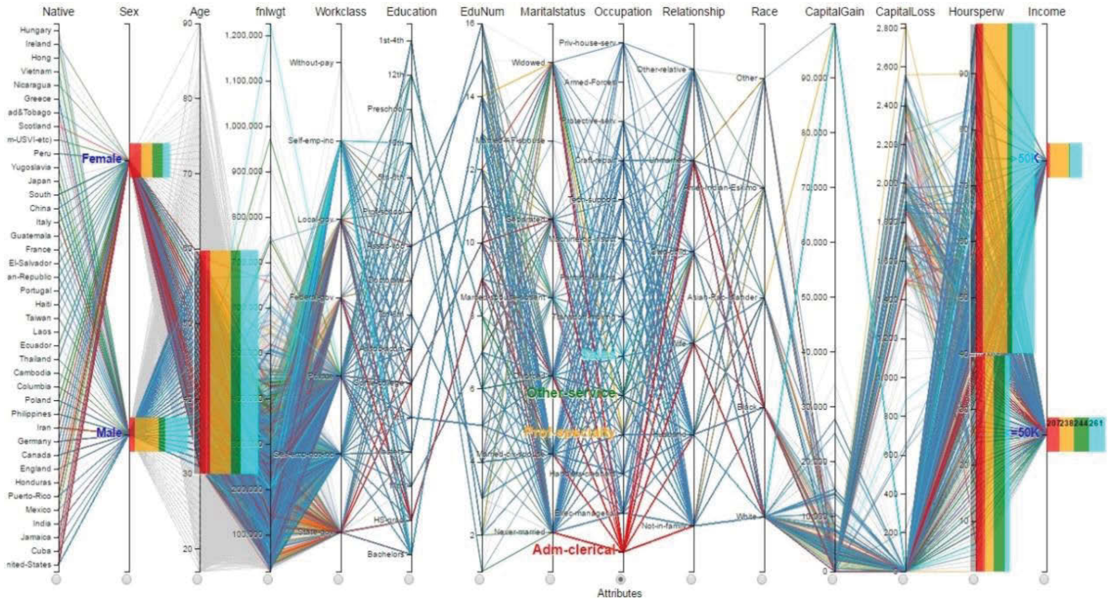
5.5.1 Parallel Coordinate Scalability

In this section, we discuss the visual scalability of the parallel coordinates of SumUp in term of the display space optimization, increasing-polyline issues, and categorical-data queries.

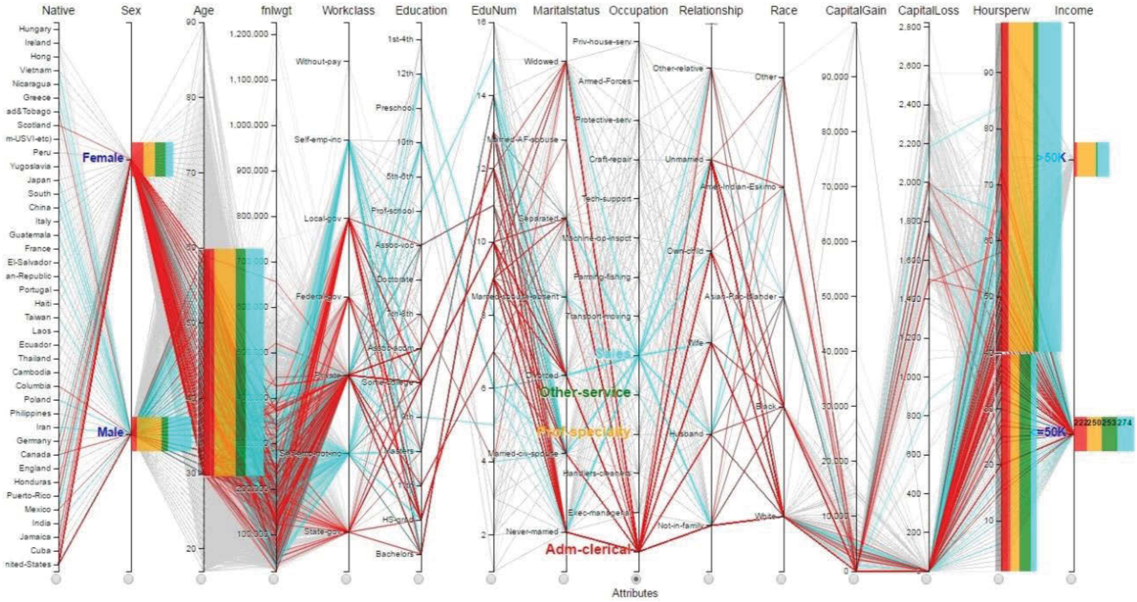
That the layout of the SumUp browser is the stacked bars overlayed on the parallel coordinates might cause the loss of the visual pattern of polylines due to their overlapped position. For dealing with this, the transparency of the stacked-bar layer can be freely customized, which can keep both the layers clearly visible and switchable in the same display space without the loss of the visual patterns.

Besides, SumUp applies the regular and flexible scale of the bar lengths to support the statistical comparison inside and across axes. While the regular scale of the bar lengths is computed with the maximum length of bars on all the axes, the flexible one is given by the maximum length of the bar inside an axis only. Consequently, the spaces between axes are efficiently utilized for the stacked bar representation. Additionally, in the case of the small-length stacked bars, their labels would be showed when the display space is sufficient, or the mouse-over interaction is applied.

A common visual challenge of the parallel coordinates is representing a high number of the polylines in a limited 2D space, which might cause the increase of the pattern clutter and affect the navigation tasks. We develop an adaptive filtering feature using the stacked-bar interaction to handle the challenge. Users can emphasize and trace the polylines by selecting the clustered colours on the associated stacked bars. As a result, the considered polyline clusters can be kept distinctive by the colour-based groups as the user needs. The filtering could make the clutter decreased and appropriate for comparison and analysis, so the navigation tasks can be performed easily (see Figure 5.21).



a) The visual comparison with the general view of all the polylines and stacked bars.



b) The focus comparison with the filtered view of the instances of *Income 50K and lower*, *Occupation* with *Sales* (cyan polylines), and *Adm-clerical* (red ones).

Figure 5.21 The filtering feature of SumUp applied in the visual analysis of Census income data concerning *Income, Hourperweek, Age, and Sex* towards *Occupation*.

Figure 5.21 indicates that Image (a), resulted from a Census income data set, shows the query comparing four attributes *Sales, Other-service, Pro-specialty,* and *Adm-clerical* of *Occupation* by *Sex, 30-60 Age, under-40 or over-40 Hoursperw,* and *Income*. Image (b) is the filtered pattern of *Sales* (cyan polylines) and *Adm-clerical* (red ones) cases in *50K-and-lower Income*. Although the distribution measurement is not applied to all the dimensions, the filtered view shows plenty of analysis useful information, especially about *Workclass, Education,* and *CapitalGain*. For *Workclass*, the focused population almost occupied their own groups excluding *Private* class and including *self-emp-inc* and *self-emp-not-inc* for *Sales* people, and *Local-gov, State-gov,* and *Federal-gov* for *Adm-clerical* ones. For *Education*, there was no the considered population at *1st-4th* level. Levels *5th-6th, 10th, 12th*, *Preschool,* and *Prof-school* were given by *Sales* population, and all the remaining levels were given by both of them. Furthermore, for *CapitalGain*, all of the *Sales* and *Adm-clerical* population had the profits lower *10-thousand* level.

In spite of that some of the existing parallel-coordinate browsers support statistical functions, they would rather focus on the numerical data than the categorical ones. We, partly motivated by this reason, delivers a statistical query feature that can work with both of the data types. The categorical data can be mapped as either the input for creating the target attributes or the value of the selected traversing polylines. Besides, as with other standard browsers of parallel coordinates, SumUp is equipped with the reordering function to support flexible organizing the axis positions for reasonable analysis.

5.5.2 Scatterplot Navigation

Due to the sharing display space of the stacked bars and the plotting zone of FigAxis, the zooming a focus zone might result in the loss of the context view of the close proximity. Thus, we provide the panning component to deal with the issue. By using the method, users can swing the main view in different directions to navigate the desired proximity (see Figure 5.22).

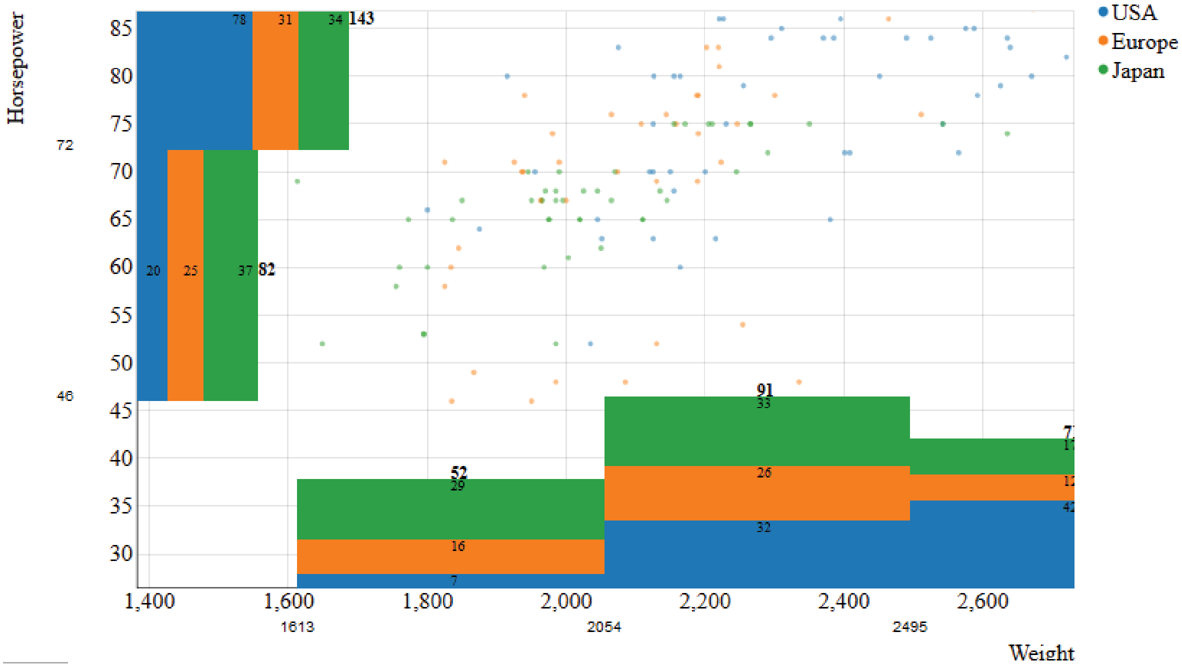


Figure 5.22 The zooming and panning feature of FigAxis in the proximity navigation.

The model of FigAxis is able to support up to five-dimension encoding including two-axis-based encoding and colour, size, and shape based representation. We offer the friendly filtering component of colour distribution for interaction enhancement. Figure 5.23 is a sample of the colour-based filtering.

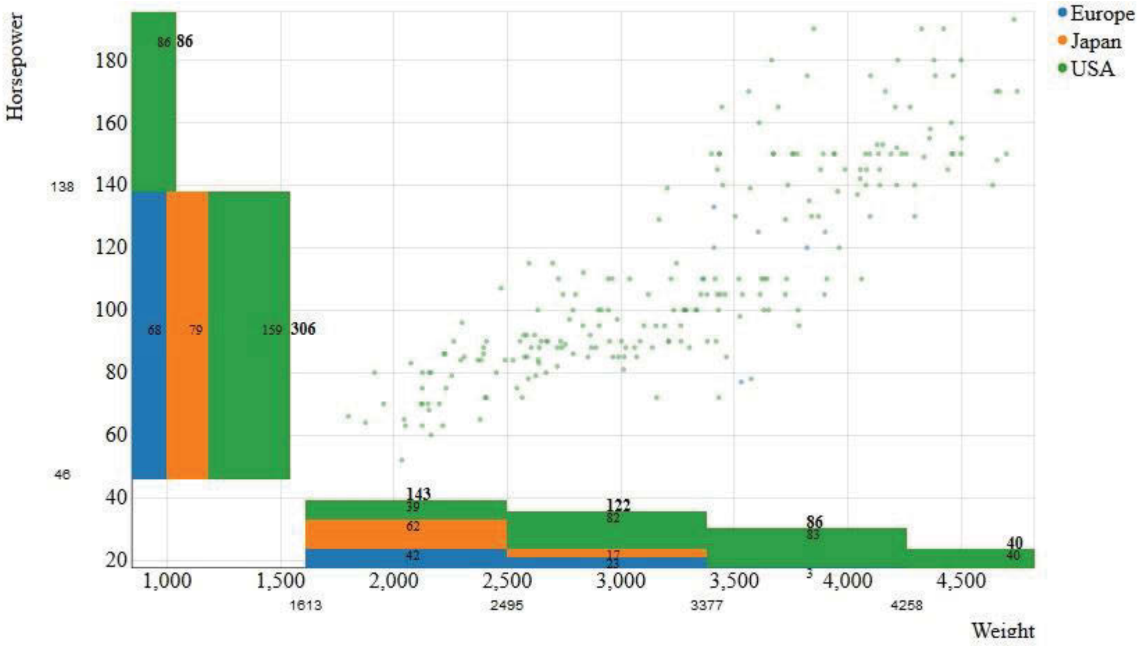


Figure 5.23 The colour-based highlight of the car models delivered by *USA* (the green plotted points and the green stacked bars). The filtering can be activated by a click on the considered stack bar or a data point.

For the scatterplot layouts, one of the scalability challenges is to deal with the clutter of the data points increased by plotting a large number of data instances in the limited plotting space. In order to avoid the issue, we propose a method, extended from FigAxis, for correlation data summary based upon plotting data cells. The method employs the stacked bars to be embedded in the data cells instead of on the axes, which creates a scatterplot browser with clear and coherent views for statistical analysis.

Extended FigAxis is the matrix of data summary for representing the correlation of a couple of dimensions (see Figure 5.24). The horizontal and vertical dimensions plot the data cells, which contains horizontal stacked bars to encode the numbers of instances towards the target attributes. Moreover, the scatterplots is designed to be adaptive with fish-eye distortion, which can be helpful for the navigation in the case of the high increase of data cells.

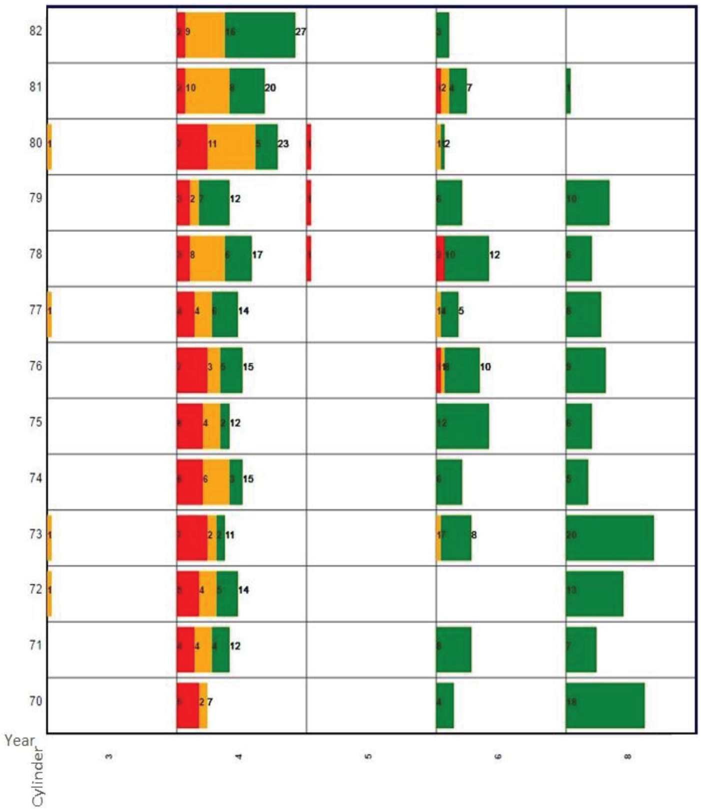


Figure 5.24 The extended FigAxis layout of the visual comparison of the car model delivery in term of *Year* and *Cylinder*.

According to Figure 5.24, the pattern is resulted by the comparison of *Year* and *Cylinder* of a Car data set. *Japan* exclusively delivered *3*-cylinder models at *1972*, *1973*, *1977,* and *1980*, with one model of each, whereas only *Europe* delivered *5*-cylinder models from *1978* to *1980*, with one of each also. The *4*-cylinder segment was contributed by all manufacture origins *Japan, Europe*, and *USA*. The highest number of the delivered models was at *1982*, with *27* models including *2* of *Europe*, *9* of *Japan* and *16* of *USA*, while the smallest number of that was at *1970*, with *7* models including *5* of *Europe*, *2* of *Japan* and no one of *USA*. For the *6*-and-*8*-cylinder market, *USA* largely contributed and exclusively delivered *8*-cylinder models. The most contribution of *USA* for *6*-cylinder engines was at *1975*, with *12* models, and that for *8*-cylinder engines was at *1973*, with *20* models. However, there was neither *6*-cylinder models delivered at *1972* nor *8*-cylinder models introduced at *1980* and *1982*.

The height of the stacked bars of extended FigAxis is designed to be eighty-percent high to the cells, while their length calculation is based on the statistical results. We define extended FigAxis with the additional requirement of further investigation as follows.

It is assumed that

- X_t and Y_t are the two sets of the attribute values of the horizontal dimension and vertical one, selected at step *t*, and *t>1,*
- $C(a_i, x_t, y_t)$ is the set of the data instances satisfying target attribute a_i by the values x_t and y_t, $x_t \in X_t$ and $y_t \in Y_t$.

Therefore,

$$S_t(A,x_t,y_t) = \sum_{i=1}^{n} \left(C(a_i, x_t, y_t) \cap C(a_i, x_{t-1}, y_{t-1})\right)$$

Equation 5.4

where $S_t(A,x_t,y_t)$ is the function to compute the length of a stacked bar as the summary of the data instances satisfying all the target attributes in *A* by x_t and y_t.

5.6 Query Implementation

5.6.1 A System Framework for Visual Queries with Quantitative Approach

We propose the following system framework for deploying our visual query techniques concerning SumUp (see Figure 5.25). The framework is mostly similar to the one for deploying MCquery. In fact, the significant difference of those is about the graphical user interface design.

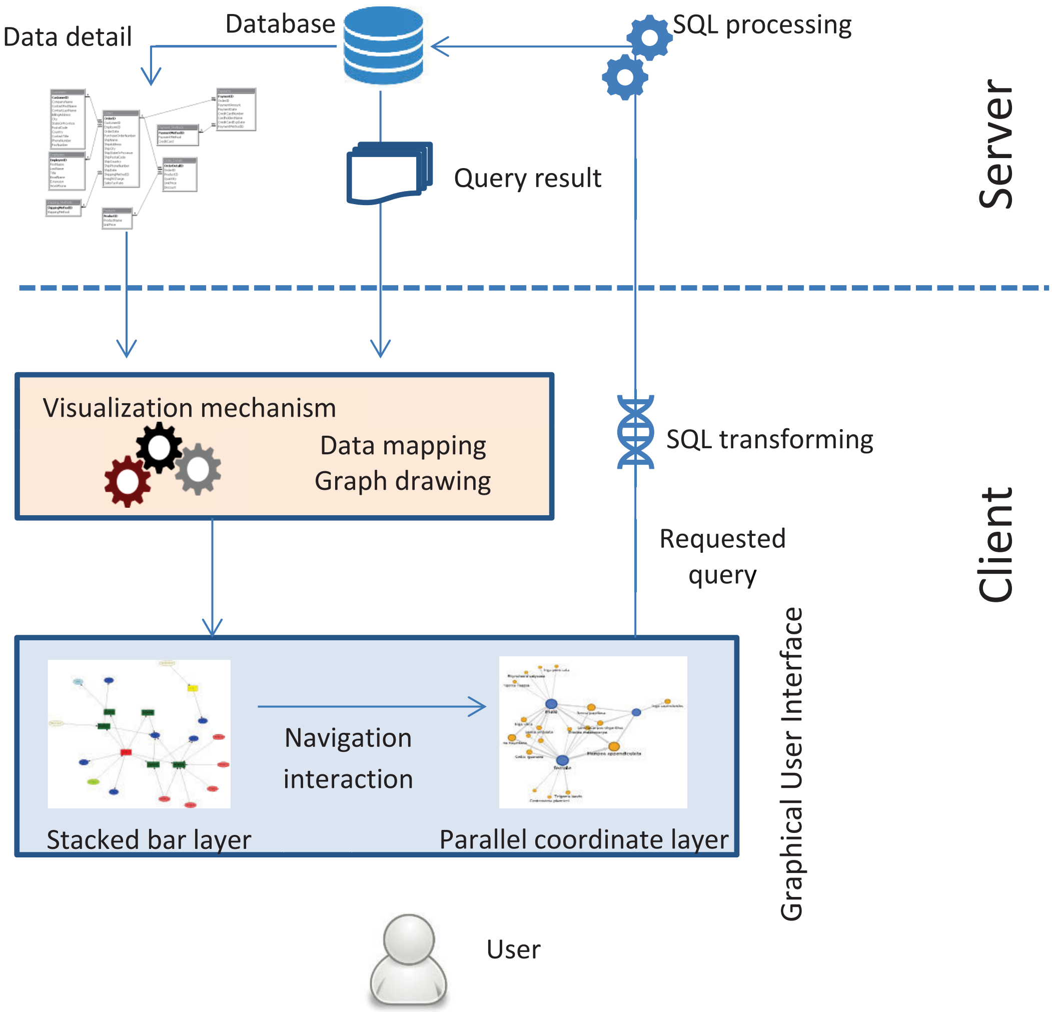


Figure 5.25 The system framework proposed for the visual query deployment of SumUp.

A standard SQL transformation is defined below with recalled functions *S_q (A, R_q)* and *S'_q (A, D')*

SELECT *DISTINCT (Target Dimension),*

COUNT (instances)

FROM *database.tables*

```
WHERE       instances IN
                  (range $R_q$) //in the case of operator OR
                  (range $R_q$ satisfying A, D’) // in the case of operator AND

AND         Target Dimension
                  = $A_j$ // if the target attribute is a single value
                  IN (range of $A_j$)// if the target attribute is a range of values
GROUP BY    Target Dimension
```

5.7 Summary

This chapter has presented two new visual query techniques for improving the statistical functions of the visualization of parallel coordinates and scatterplots. The approaches are mainly about the integration of interactive stacked bars with the brushing operation on the parallel axes and the combination of sharing dynamic axes with the zooming control on the plotting space. The former, named SumUp, serves the multiple-attribute queries on the flexible ranges of traversing polylines on parallel coordinates, and the latter, called FigAxis, helps to trace and match the focus plotting zones quickly and easily with the instance density measurement. These techniques enable users to perform the statistical tasks with plenty of useful information and rich manipulation, which contributes to reducing the users’ attempts in the processes of visual data exploration. The case studies and technical evaluation of these approaches will be presented in Chapter 6 and Chapter 7.

Chapter 6 Case Studies

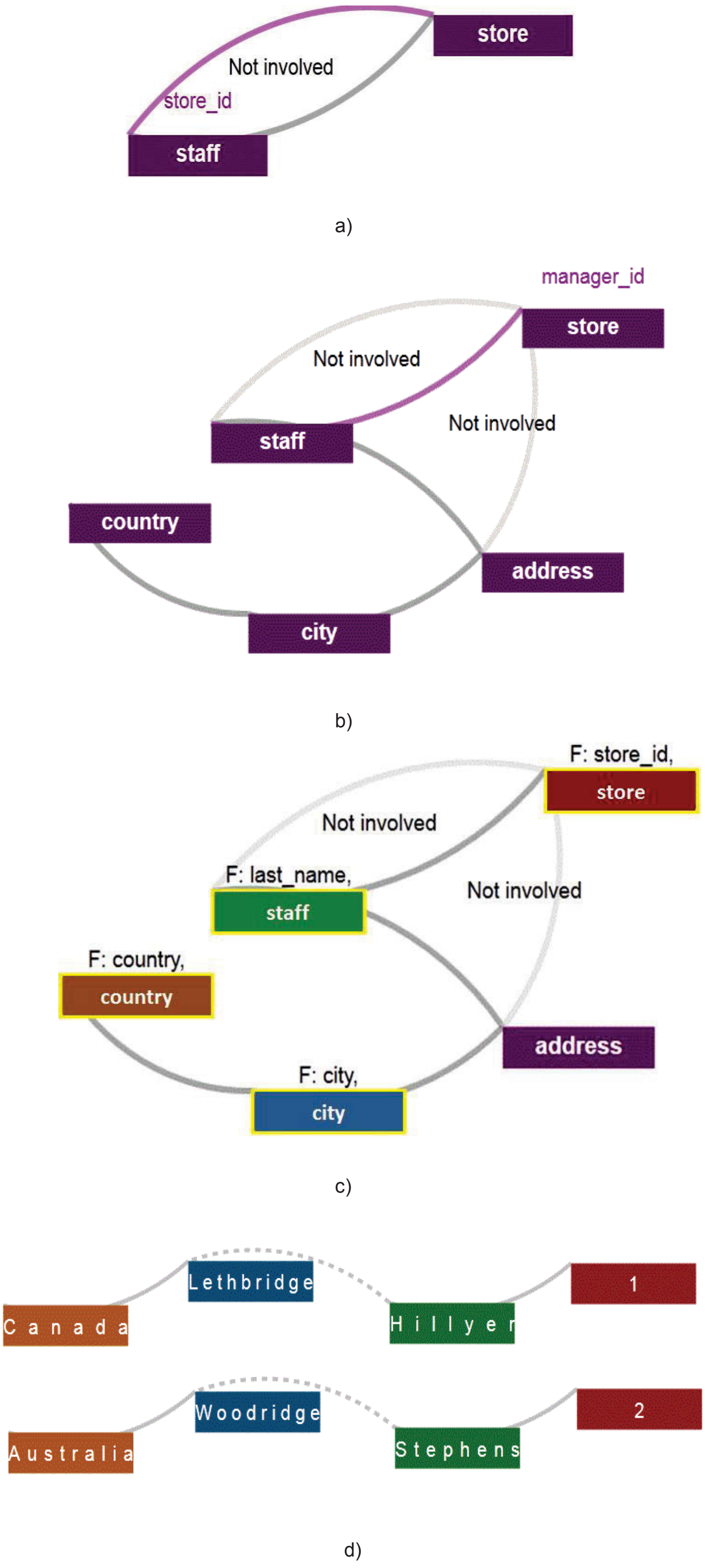
This chapter presents three case studies for demonstrating the implementation, usefulness, and effectiveness of our interactive visual queries introduced in Chapter 4 and Chapter 5. Case Study 1 is for MCquery concerning the visual data exploration with relational models, Case Study 2 is about SumUp and FigAxis towards the visual analysis of multi-dimensional data, and Case Study 3 is about the multiple-visual-context exploration.

6.1 Case Study 1: Visual Data Exploration with Relational Models

This section illustrates a case study with four typical scenarios for MCquery demonstration. The case study uses the DVD rental store data in Sakila database of MySQL (Oracle 2017). These data are about films, categories, actors, customers, countries, and other rental information and totally include 16 tables, which are stored under InnoDB, a default storage engine of traditional MySQL servers.

6.1.1 Scenario 1

This scenario considers the interaction on relationship representations. It is supposed that we would like to know who the managers of stores are and where they live. Because a staff of a store is a casual one and might be a management staff, we need to specify which relationship is used in the query (see Figure 6.1). The relationship via *store_id* plays the role of a casual staff, so this relationship is removed (see Figure 6.1.a). The other relationship via *manager_id* referring to a management staff is preserved for the query (see Figure 6.1.b). Besides, an address might belong to a store or a staff, so we must remove the un-used relationship as well (see Figure 6.1.b). Then, we add the finding dimensions *store_id, last_name, city,* and *country* on nodes *store, staff, city,* and *country* for the query (see Figure 6.1.c). The result pattern in Figure 6.1.d shows that the manager of store 1 is Hillyer, living in Lethbridge, Canada, and the one of store 2 is Stephens, living in Woodridge, Australia.



a)

b)

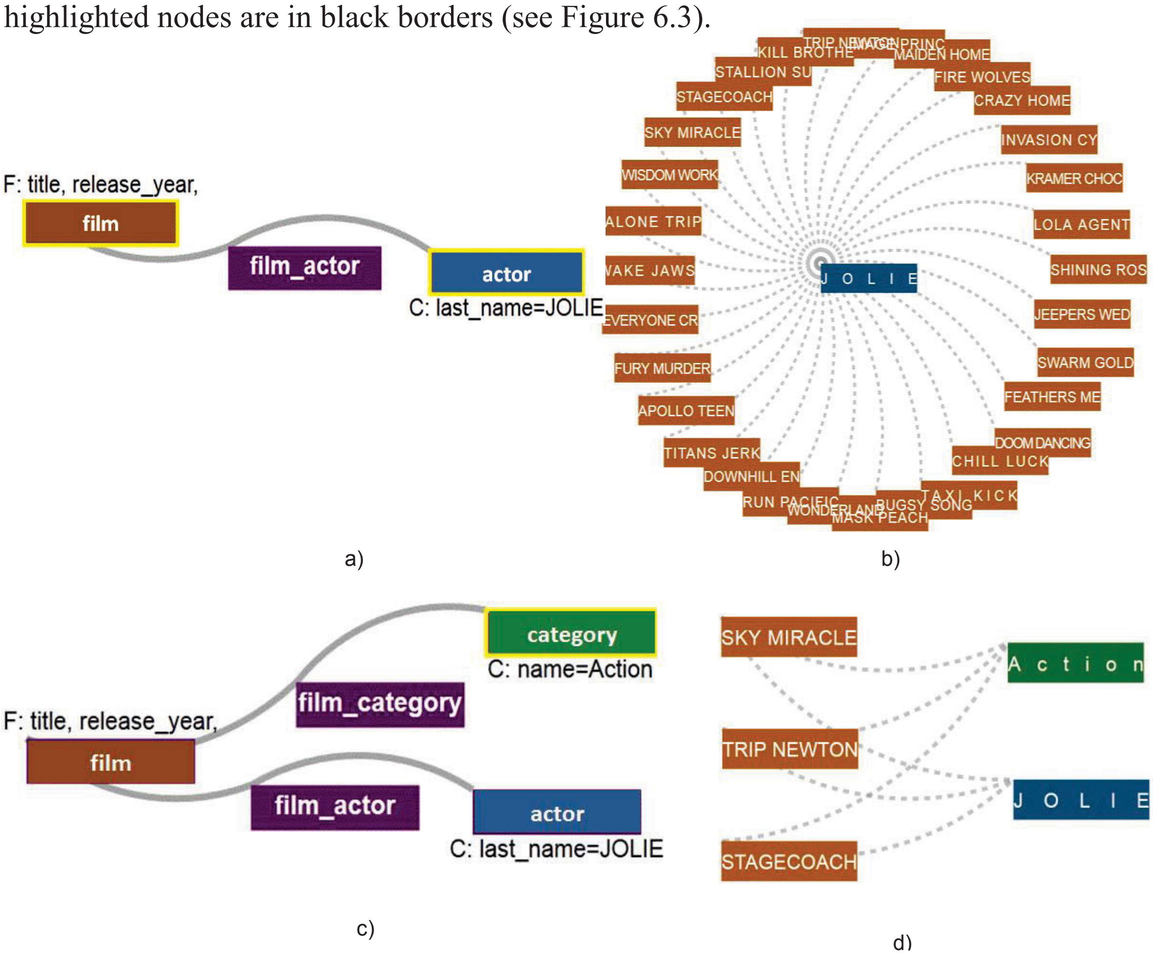
c)

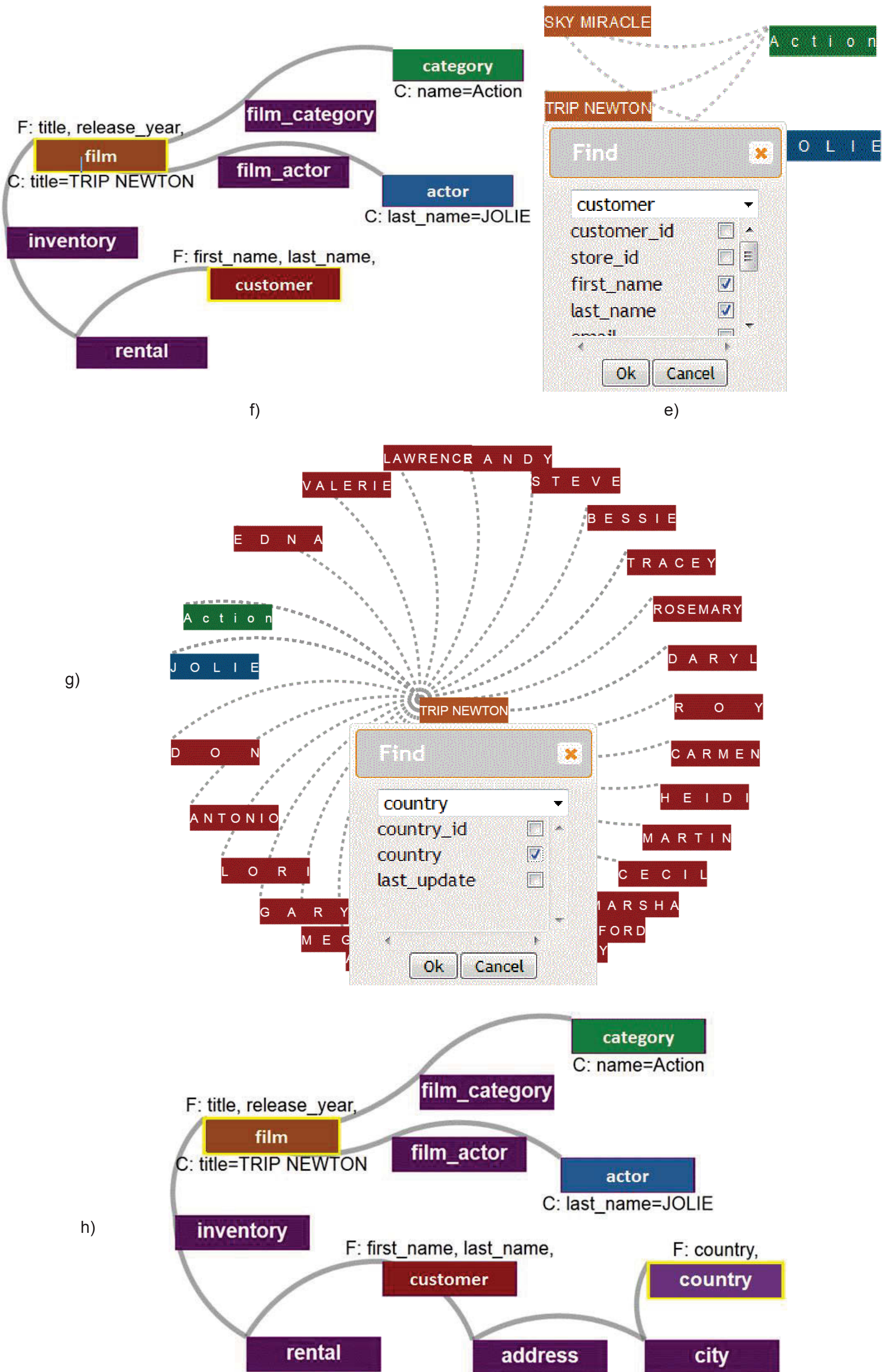
d)

Figure 6.1 The query interaction on the relationship representations of *store, staff, city,* and *country*.

6.1.2 Scenario 2

This scenario proves the flexible interaction on the coordinating visual contexts of data models and query results, which enables users to quickly adjust further queries for continuous exploration without the query reformulation. Firstly, it is assumed that we would like to know all the film titles and released years of the actor with last name *Jolie*. After formulating the query and observing the result (see Figure 6.2.a, 6.2.b), we want to focus on the films categorized in type name *Action*. We make a relative query (see Figure 6.2.c) and receive an adjusted result (see Figure 6.2.d). Now, film *Trip Newton* sounds interesting, and we are curious to know who watched it. To meet this, we just simply update the request on the query result graph (see Figure 6.2.e). This request is synchronized in the data model context as well (see Figure 6.2.f). Additionally, for detail information about the customers, we would like to know where they are from, and again we add the request on the query result graph (see Figure 6.2.g) and see the update in the data model context (see Figure 6.2.h). Finally, with the filtering feature, we could concentrate on each group of the results by countries such as *China* and *India*, and the highlighted nodes are in black borders (see Figure 6.3).





customer
customer_id ☐
store_id ☐
first_name ☑
last_name ☑

country
country_id ☐
country ☑
last_update ☐

Figure 6.2 The flexible interaction on the coordinating visual contexts of the data model and query results in Scenario 2.

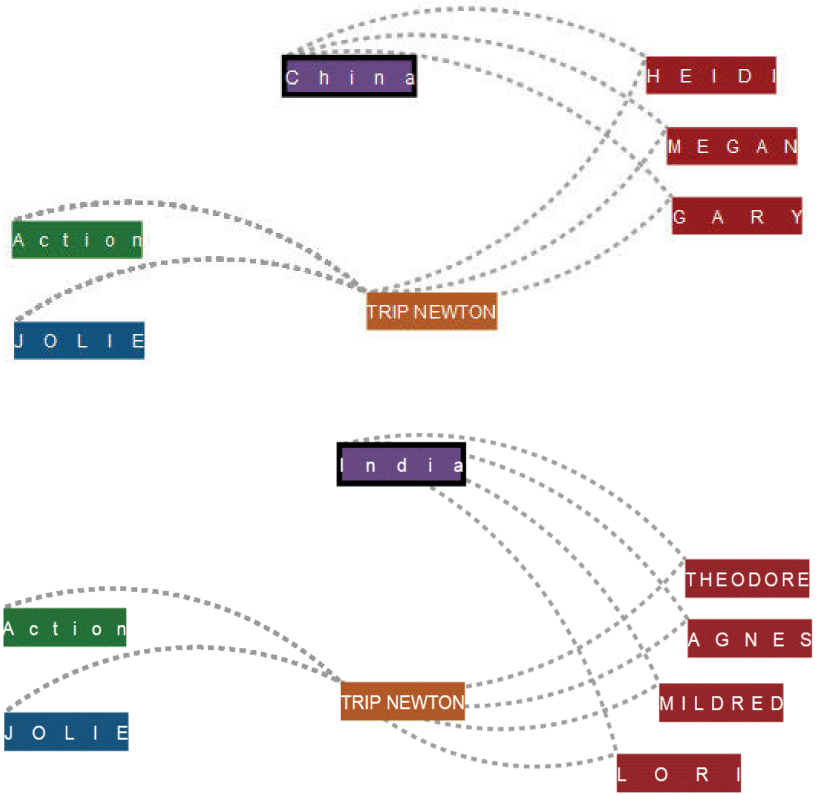
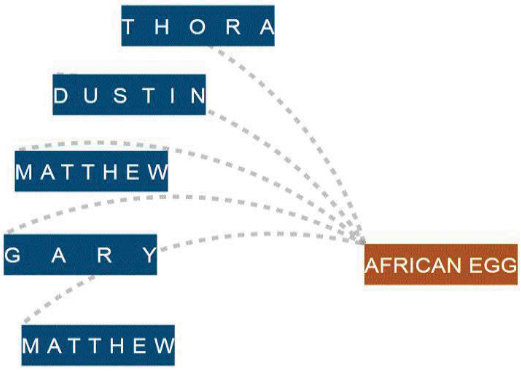


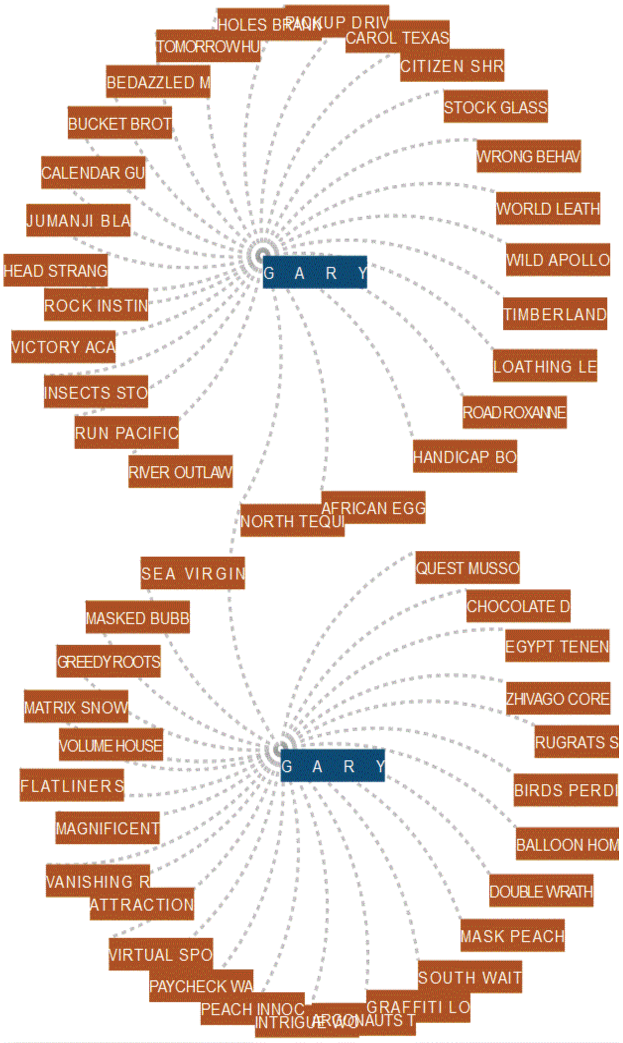
Figure 6.3 The filtering feature applied for countries *China* and *India*.

6.1.3 Scenario 3

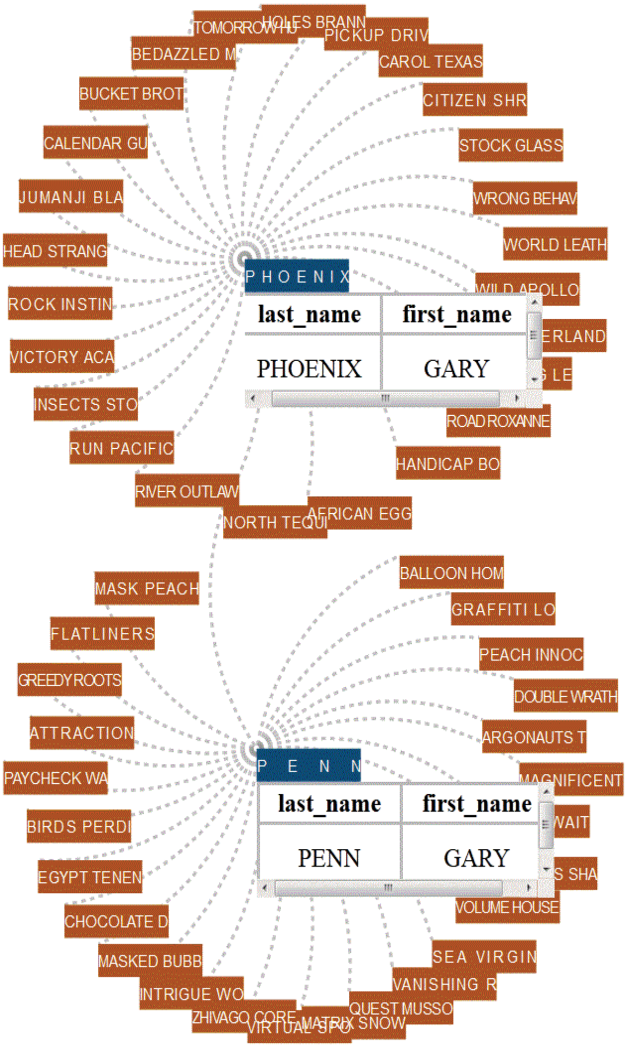
This part shows the query adjustment with the highlighted-dimension support. In the beginning, we would like to find the actors with their first names in film *African Egg*. After watching the result (see Figure 6.4.a), we change the finding to all the films of actor *GARY*. Surprisingly, there are two actors with first name *GARY* displayed (see Figure 6.4.b). To identify these two actors, we add dimension *last_name* to the query and make the node highlighted on *PHOENIX* (see Figure 6.4.c). Now, we could use last name *PHOENIX* for a further query such as looking for *PHOENIX's* favourite categories (see Figure 6.4.d). The result is shown in Figure 6.4.e.



a)



b)



last_name	first_name
PHOENIX	GARY

last_name	first_name
PENN	GARY

c)

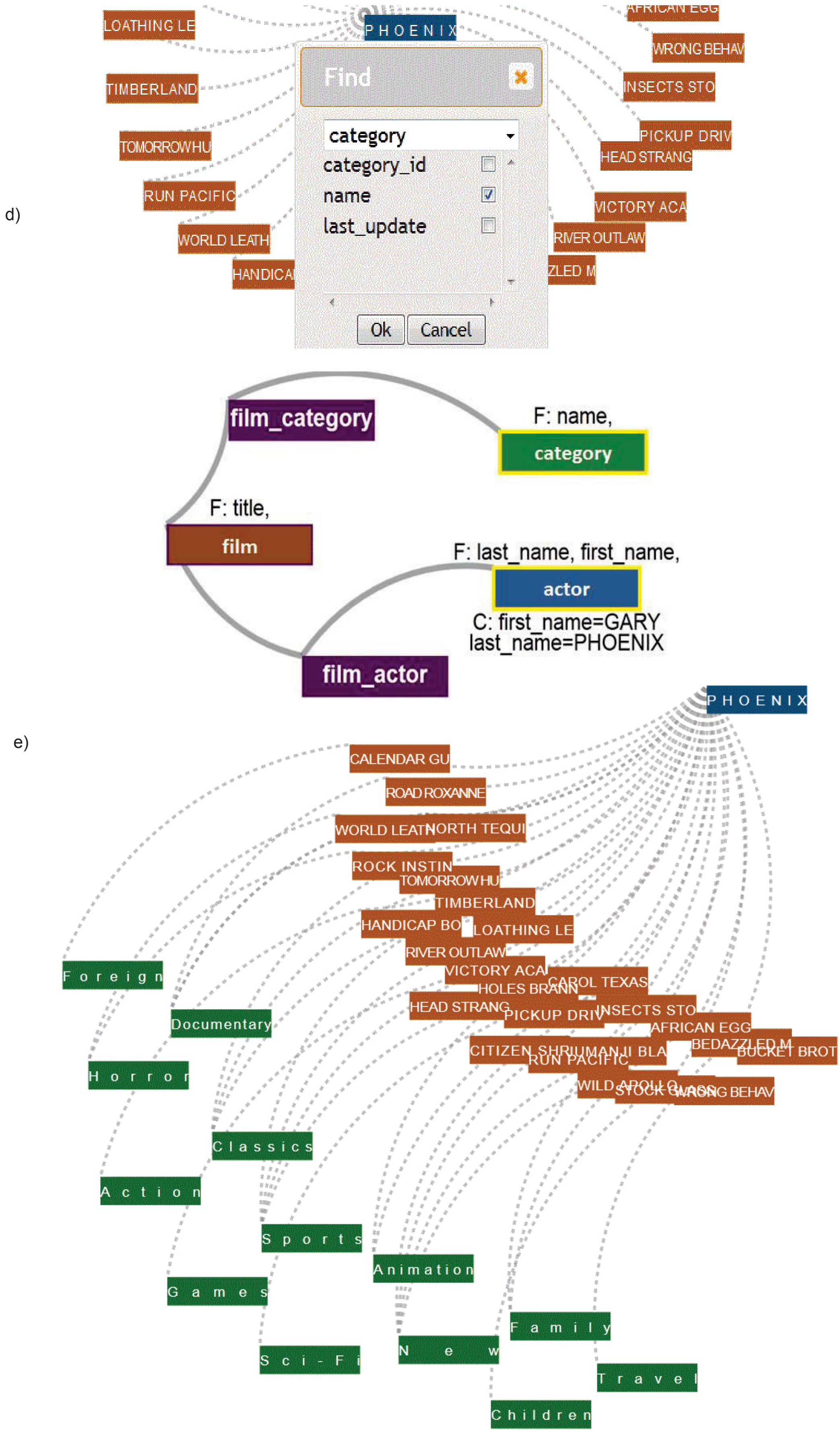


Figure 6.4 The query interaction with the highlighted dimensions of *film, category, and actor*.

6.1.4 Scenario 4

This scenario is about the ease of relationship recognition of the tuple representations. At first, we would like to search all the films in category *Children*. Then, we want to see which ones of those are in category *Comedy*. In the result graph (see Figure 6.5), we could quickly recognize the five films of category *Children* which are also in category *Comedy*.

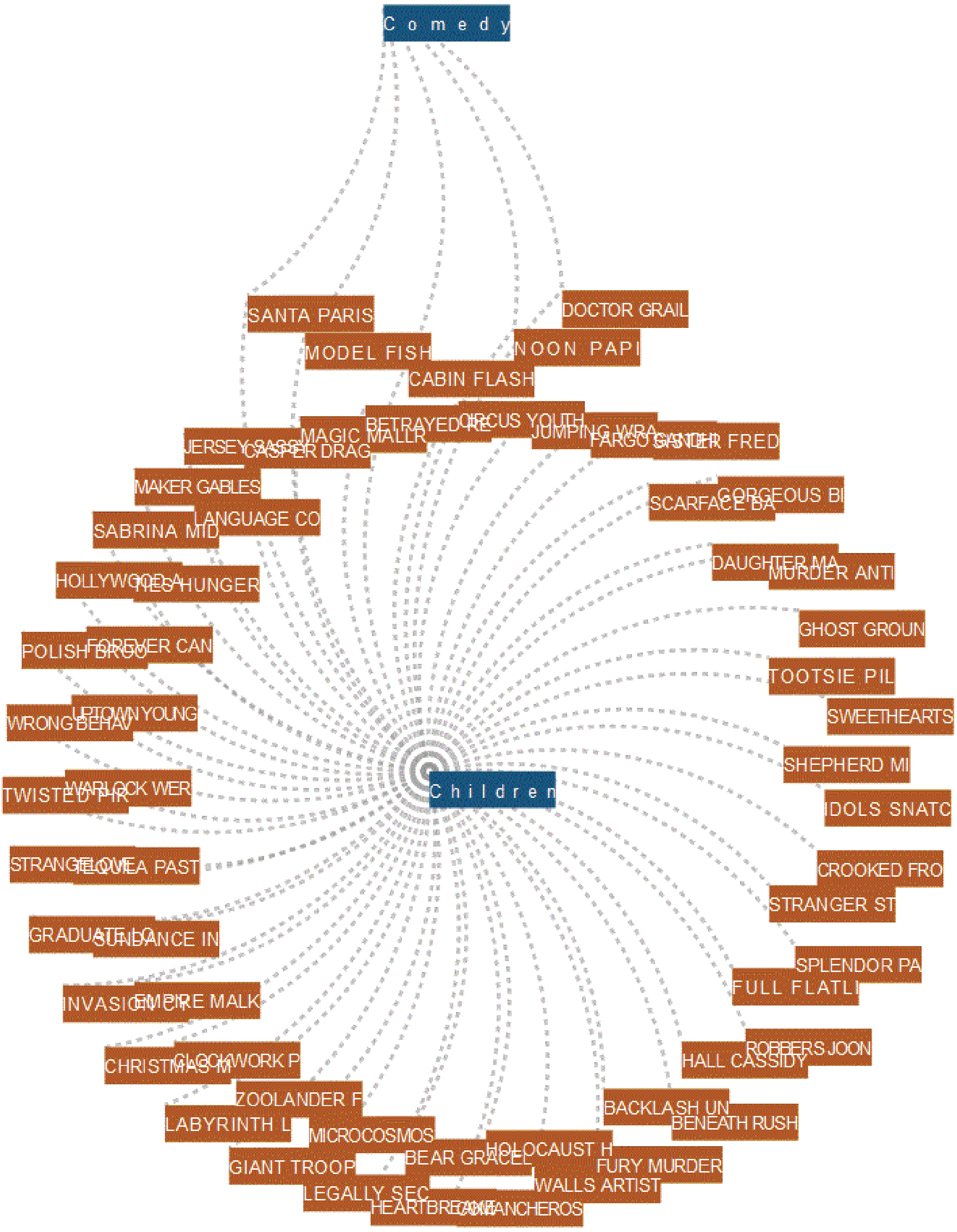
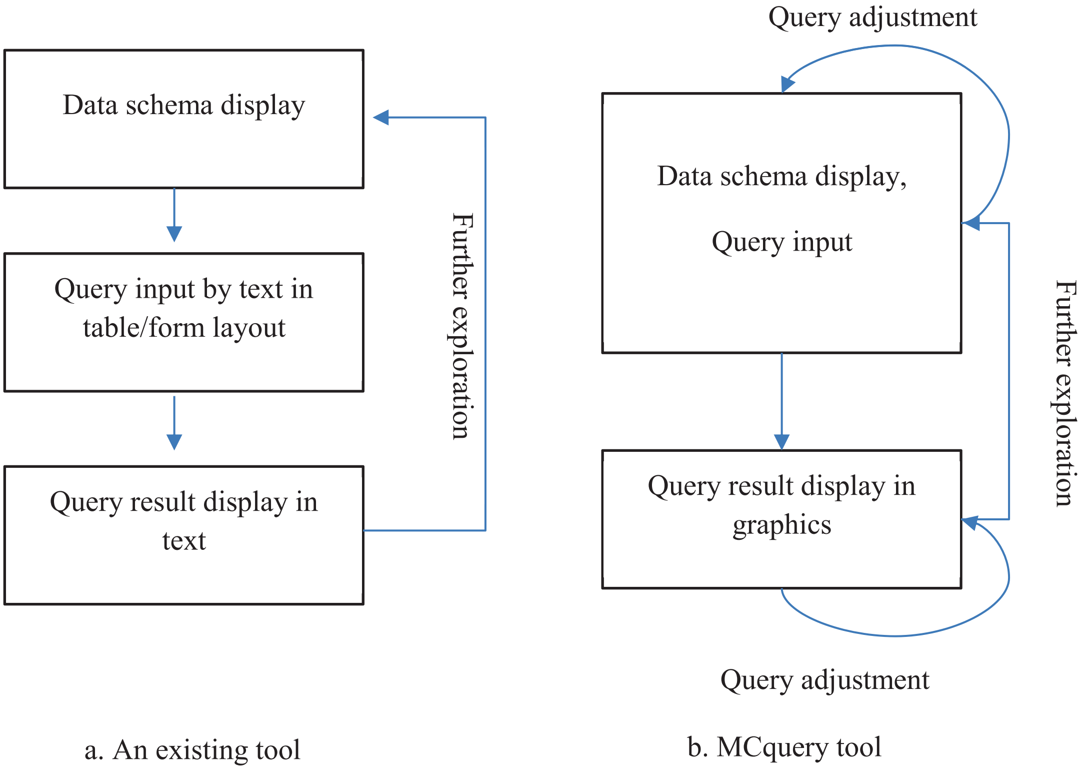


Figure 6.5 The relationship recognition in the query result of categories *Children* and *Comedy*.

6.1.5 Discussion

The above scenarios are completed in convenient steps compared with the existing tools such as the query tools of Access (MS Access 2017) and MySQL (Oracle 2017). The general procedures of those can be summarised in Algorithm 6.1. It is clear that while the existing tool procedure would focus on single exploration steps with text-based controls, MCquery supports continuing discovery with interactive-visualization manipulation. Therefore, users would have more options and flexibility in query creation and adjustment during the addressed scenarios. The detail of feature comparison of these tools is carefully described in Section 7.2.1 of Chapter 7.



Algorithm 6.1 Scenario completion procedures of MCquery and a data query tool such as MS Access or MySQL.

6.2 Case Study 2: Visual Analysis of Multiple-Dimensional Data

This section illustrates the distinctive features of SumUp regarding quantitative analysis on parallel coordinates. The three use cases below are about the main analysis tasks of SumUp including multiple-attribute comparison, correlative analysis, and flexible data support. The data during this case study are the Car data set with nine dimensions of 398 instances and a set of the Census income data with fifteen dimensions of 9998 instances (Lic 2013).

6.2.1 Multiple-Attribute Comparison

This use case demonstrates the capability of SumUp in the statistical visual comparison by multiple attributes with the flexible ranges of traversing polylines. This feature has not been greatly supported by the existing parallel-coordinate browsers. The overall purpose is to compare the numbers of the delivered models of six *Japanese* brands based on *Cylinder, Horsepower, Weight,* and *Year*. Firstly, for the query formulation (see Figure 6.6), we make a brush to filter *Japanese* brands, then we select the mentioned ones as the query attributes including *Toyota, Subaru, Nissan, Mazda, Honda,* and *Datsun*. From the highlighted details of the browser, we configure the query ranges as follows. For *Cylinder*, we select three filtered *Cylinder* numbers *3, 4,* and *6*. We divide *Weight* by three ranges including the weights from *2500* to under *3000*, the weights from *2000* to under *2500*, and the weights under *2000*. For *Horsepower*, we take two ranges including the powers greater *80* and lower *140*, and the powers lower *80*. We consider two periods of the model production including before *1980* and *1980* and after.

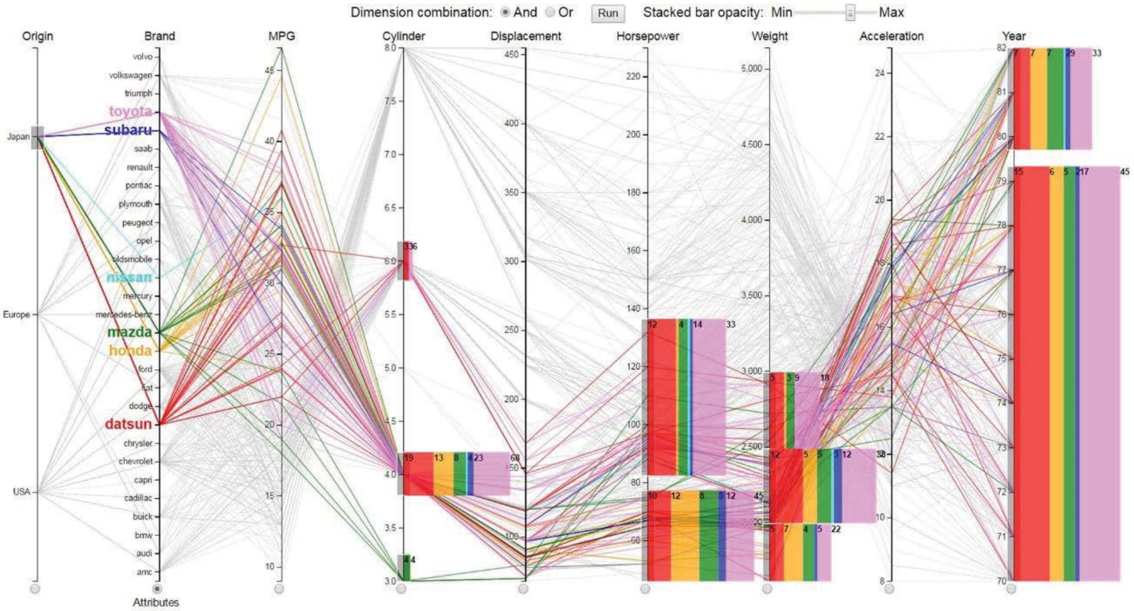


Figure 6.6 The multiple-attribute comparison for the number of models of six *Japanese* brands based on *Cylinder, Horsepower, Weight*, and *Year*.

Once the query is performed, the browser shows a meaningful pattern (see Figure 6.6). Generally, the models of *Toyota, Mazda, Honda,* and *Datsun* were more than that of *Subaru* and *Nissan*. There was a significant difference in the number of the models regarding the *4-cylinder* level. Whereas all of the brands used *4-cylinder* engines, with 68 totally, only *Mazda* featured *3-cylinder* engines, with 4, and just *Toyota* and *Datsun* featured *6-cylinder* engines, with 6 totally and 3 of each. The greatest number of the models equipped with *4-cylinder* engines was for *Toyota*, with 23 models, compared to just 1 model of *Nissan*. The number of the models for *Horsepower* under *80*, with 45, was slightly more than that from *80* to *140*, with 33 models. For the weight aspect, the highest figure was at the range from *2000* to under *2500*, with 38, and about double times the remaining ranges, with 18 and 22 models of *2500-3000* weights and under-*2000* ones respectively. Before *1980* the *Toyota* and *Datsun* models were produced, with over 14 models of each, more than the others, with under 7 of each; however, from 1980 to 1982 the model delivery of *Toyota, Datsun, Mazda,* and *Honda* became more balanced, with about 8 of each. While *Subaru* delivered 2 in each period of the time, *Nissan* only introduced 1 model in the latter one.

6.2.2 Correlative Analysis

Parallel coordinates can show the correlative relationship between two dimensions by the visual pattern of traversing polylines through the axes. The order of the polylines in a pair of the axes increasing or decreasing together indicates a positive correlation while that of the axes increasing or decreasing inversely points out a negative correlation. However, since the considered axes are not always placed together, users need much effort to trace and match the details correctly.

This use case explores how SumUp helps to improve the correlative comparison of the parallel coordinates. SumUp enables the users to examine the correlative coefficients of multiple dimensions without the impact of the axis positions. It is supposed that we would like to analyse the correlation of *Weight* and *MPG (Miles per Gallon)*, *Weight* and *Horsepower*, and *Weight* and *Displacement*. Once the query targeting *Weight* attributes is performed with the ranges of *MPG*, *Horsepower,* and *Displacement*, we analyse the results as the pattern of Figure 6.7.

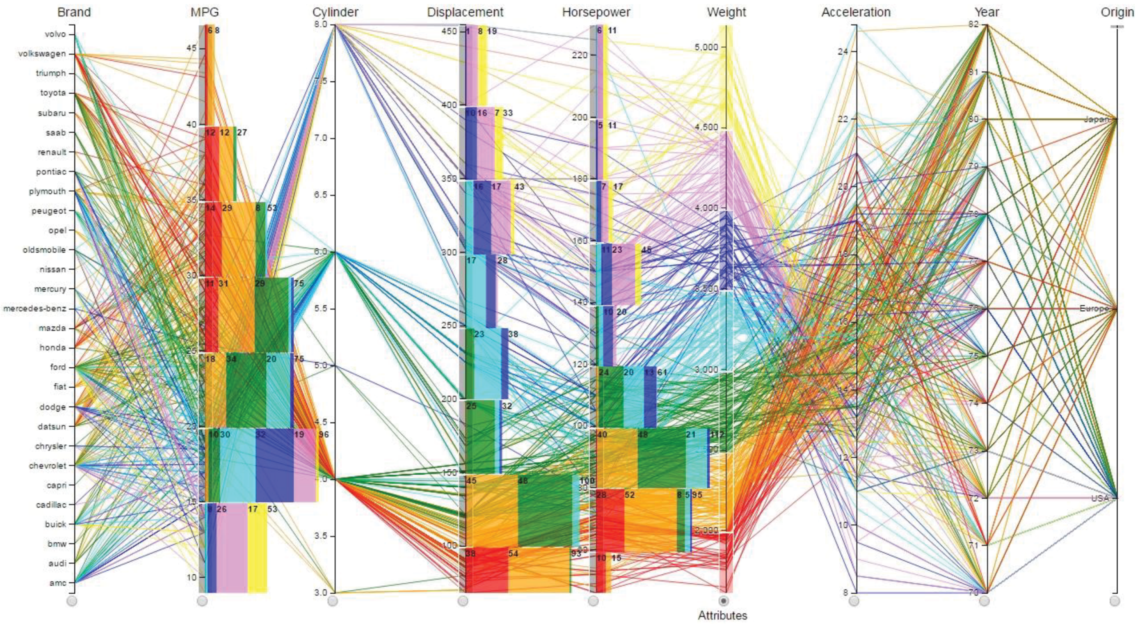


Figure 6.7 The correlative analyses of *Weight* and *MPG, Weight* and *Horsepower*, and *Weight* and *Displacement*.

Clearly in Figure 6.7, for the *MPG* summary towards *Weight*, the number of low-weight models (the red, orange and green stacked bars) occupied in most of the ranges of high *MPG* (25 and over), and conversely the number of high-weight ones (the blue, violet, and yellow stacked bars) appeared in most of the ranges of low *MPG* (lower 25), which indicates their relationship was reaching a negative correlation with high probability. For *Displacement*, the number of low-weight models filled in the majority of the ranges of low *Displacement* (lower 200), and the number of height-weight ones appeared in the majority of the ranges of high *Displacement* (200 and over), which points out their relationship was reaching a positive correlation with high probability. Similarly, the relationship between *Weight* and *Horsepower* was a positive correlation as well.

6.2.3 Flexible Data Support

This use case illustrates that SumUp can well support for both numerical and categorical data. We use the Census income data set, and the purpose is to explore *Income* and *Workclass* towards the ages of the population in *United States*. We consider numerical dimension *Age* as the query attributes by four ranges including the ages under *30*, from *30* to under *50*, from *50* to under *70*, and *70* and above. All the categorical names of *Workclass* and *Income* are selected in addition to value *United States* of *Native*, and the result is displayed in Figure 6.8.

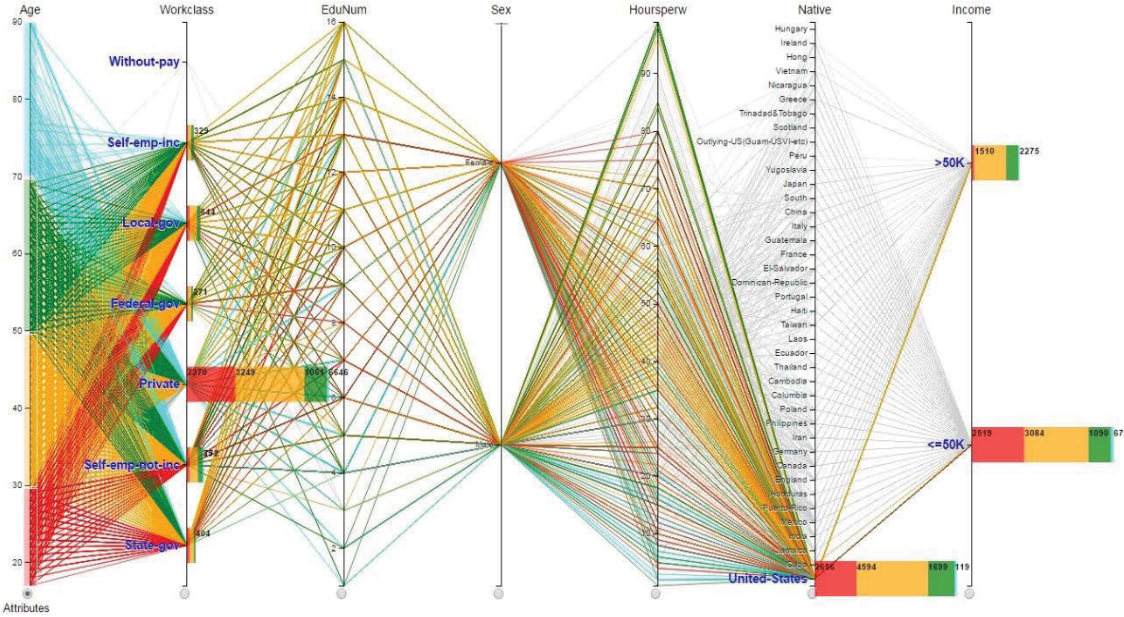


Figure 6.8 The data summary with flexible data support to explore *Income* and *Workclass* towards the ages of the population in *United States*.

Generally, in Figure 6.8, there was a large difference in the number of people at the *Workclass* summary; this at *Private* was the highest, with *6646*, and multiple times greater than that at the others, with just a few hundred. The population aged *30* to *49* years was the most, with *3249*, followed by that aged under *30* years and aged *50* to *69* years, with *2270* and *1061* respectively, whereas the smallest was of the population aged *70* and above, with *66* people. The number of *50-69*-year-old people at the level of *50K*-and-under incomes, with *1090*, was mostly two times greater than that of above-*50K* incomes, with *609*. While the population aged *30-49* years, with *3084*, was about *500* more than that aged under *30* years, with *2519*, at the level of *50K*-and-under incomes, at the level of above-*50K* incomes, the difference was more significant, with *1510* of *30-49* ages and *137* of under-*30* ages. Overall, the population at the level of *50K*-and-under incomes, with *6793*, was about three times greater than that at the level of above-*50K* incomes, with *2275* people.

6.2.4 Discussion

The scenarios easily completed by SumUp tool in this section would face difficulties when given to the existing tools such as Siirtola H. 2000 and Ho Q. et al. 2011. The challenge is that these major tasks are about multiple target comparison and correlation across dimensions, while the two mentioned tools mainly support histogram determination with single foci. Thus, in order to complete the above scenarios, much time and effort would be needed. The detail of feature comparison of these tools is carefully illustrated in Section 7.2.2 of Chapter 7.

6.3 Case Study 3: Interactive Data Exploration of Multiple Visual Contexts

This section describes how the visual data exploration framework proposed in Chapter 3 is to be deployed with our developed visual techniques. The following use cases are mainly about the coherent linking of the visualizations and the relative analysis of the visual query contexts. The Census income data is used in the first use case, and the DVD rental store data is for the second use case.

6.3.1 In-depth Exploration on Multiple Dimensions and Pairwise Comparison

This use case illustrates how our parallel coordinates and scatterplots support further investigation of multi-dimensional data. It is assumed that we would like to explore the Census income data of *United States* in term of aspect *Income* including over *50K* and *50K*-and-lower classifications.

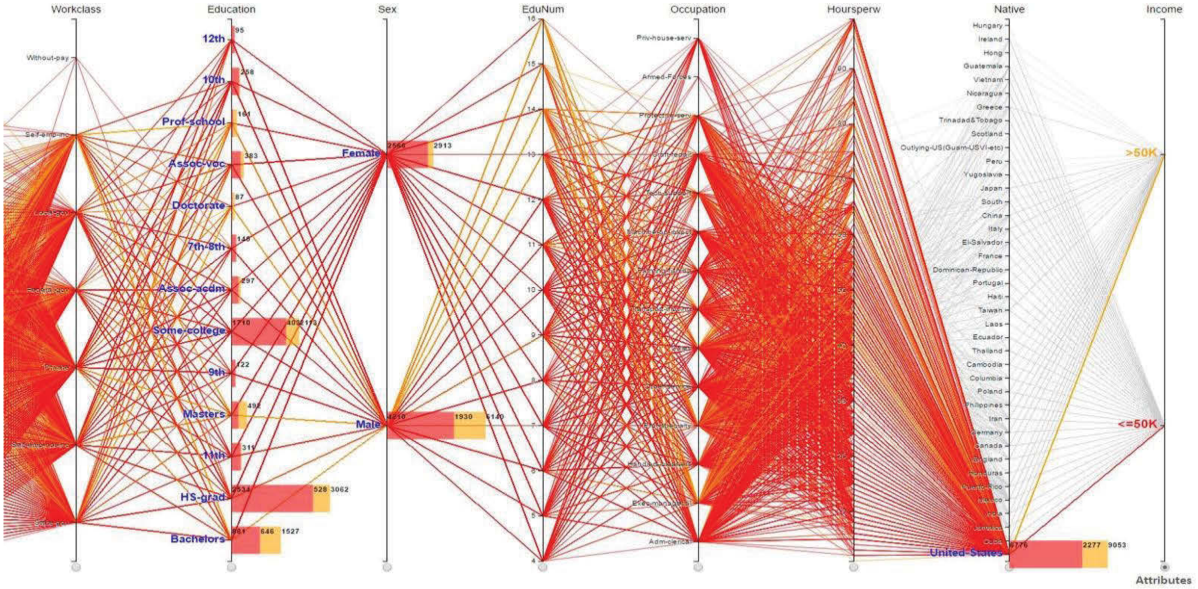


Figure 6.9 The statistical parallel coordinates of the *United States* Census income data towards *Sex* and *Education*.

Firstly, we want to know the general income distribution towards *Education* and *Sex*. According to the parallel-coordinate pattern of SumUp (see Figure 6.9), there was a significant difference of the number of the people having incomes at the two considered levels. The population having *50K*-and-lower incomes, with *6776*, was about triple times greater than that having over-*50K* incomes, with *2277*. The number of the people studying in *HS-grad, Bachelors,* and *Some-college* were highly more than that at the remaining levels of *Education*. The largest population, of those, having *50K*-and-lower incomes was at *HS-grad*, with *2534*, while the least one of that was at *Prof-school* and *Doctorate* of *Education*, with a few cases. The greatest number of the people having over-*50K* incomes was at Bachelor, with *646*, whereas the smallest numbers of that were at the education levels from *7th* to *12th*, with a few hundred cases. In this data set of *United States*, the number of the recorded *Male* population, with *6140*, was double times greater than that of *Female*, with *2913*.

Afterward, we would like to have more understanding of the income distribution between *Education* and *Workclass* through the *Male* population. According to the scatterplots drilled down on the *Male* population by *Education* and *Workclass*, the highest population was at *Private* of *Workclass* and *HS-grad* of *Education*; therefore, we are eager for finding more details of that data cell in term of other dimensions, particularly *Occupation, Age,* and *HoursPerWeek*.

Now, we would like to use FigAxis to obverse the data distribution between *Age* and *HoursPerWeek*, with the focus on the range *40* hours and over (see Figure 6.10).

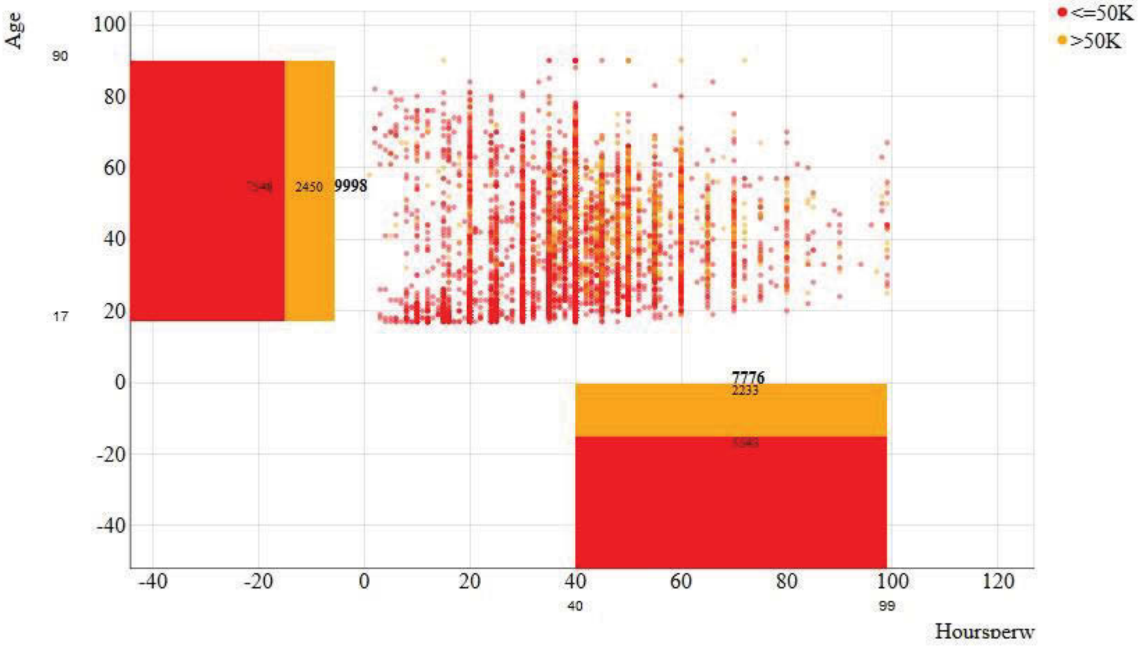


Figure 6.10 The level-1 summary of *Age* and *40-and-over Hoursperweek*.

.

According to Figure 6.10, it is clear that of *9998* recorded cases of all ages, *7776* cases working from and over *40* hours per week. *5543* of them having incomes *50K* and lower are slightly more than double times of the remaining ones having incomes over *50K*, with *2233* cases.

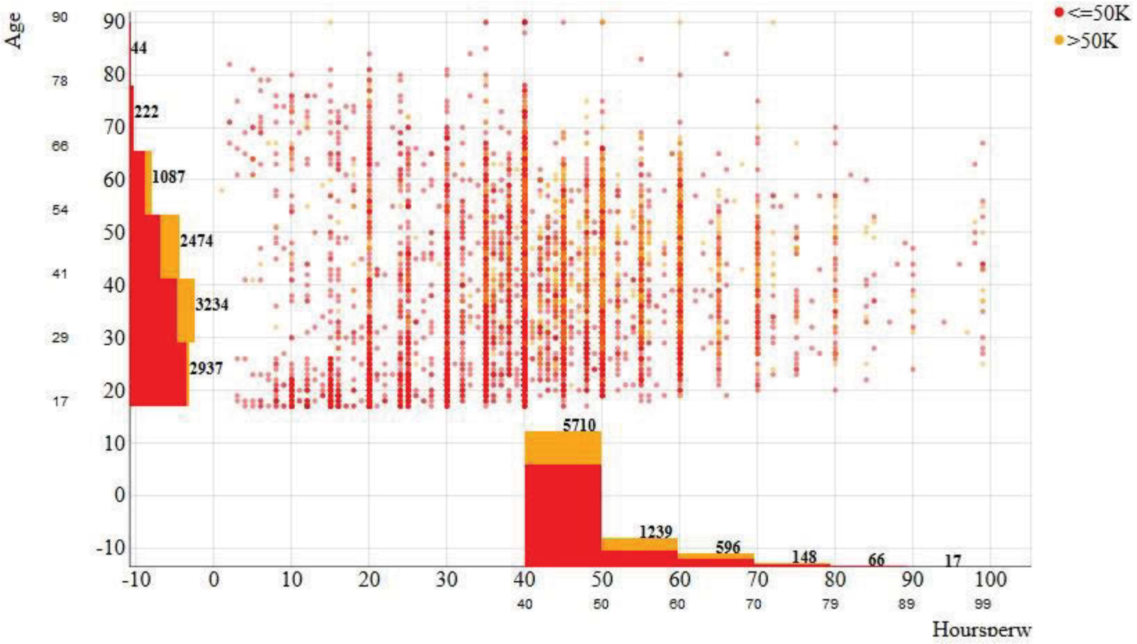


Figure 6.11 The level-6 summary of *Age* and *40-and-over Hoursperweek*.

For more details of this context, at the *level-6* summary (see Figure 6.11), it indicates that there was a significant difference of the number of the considered cases in the value ranges of *HoursPerWeek*. The greatest number of the cases was condensed at the range of *40* to *50* hours, with *5710*, whereas the remaining ranges of over *50* hours had just around *1200* and fewer cases. Meanwhile, these numbers seemed more even in term of the working ages. There were not many differences of the number of the workers in the ages from *17* to *54*, with the highest at *3234* cases and the lowest at *2474* cases. On the other hand, the most different number of the workers was allocated at the ages of *54* to *66* and *78* to *99*. While the former had *1087* cases, the latter had just *44*.

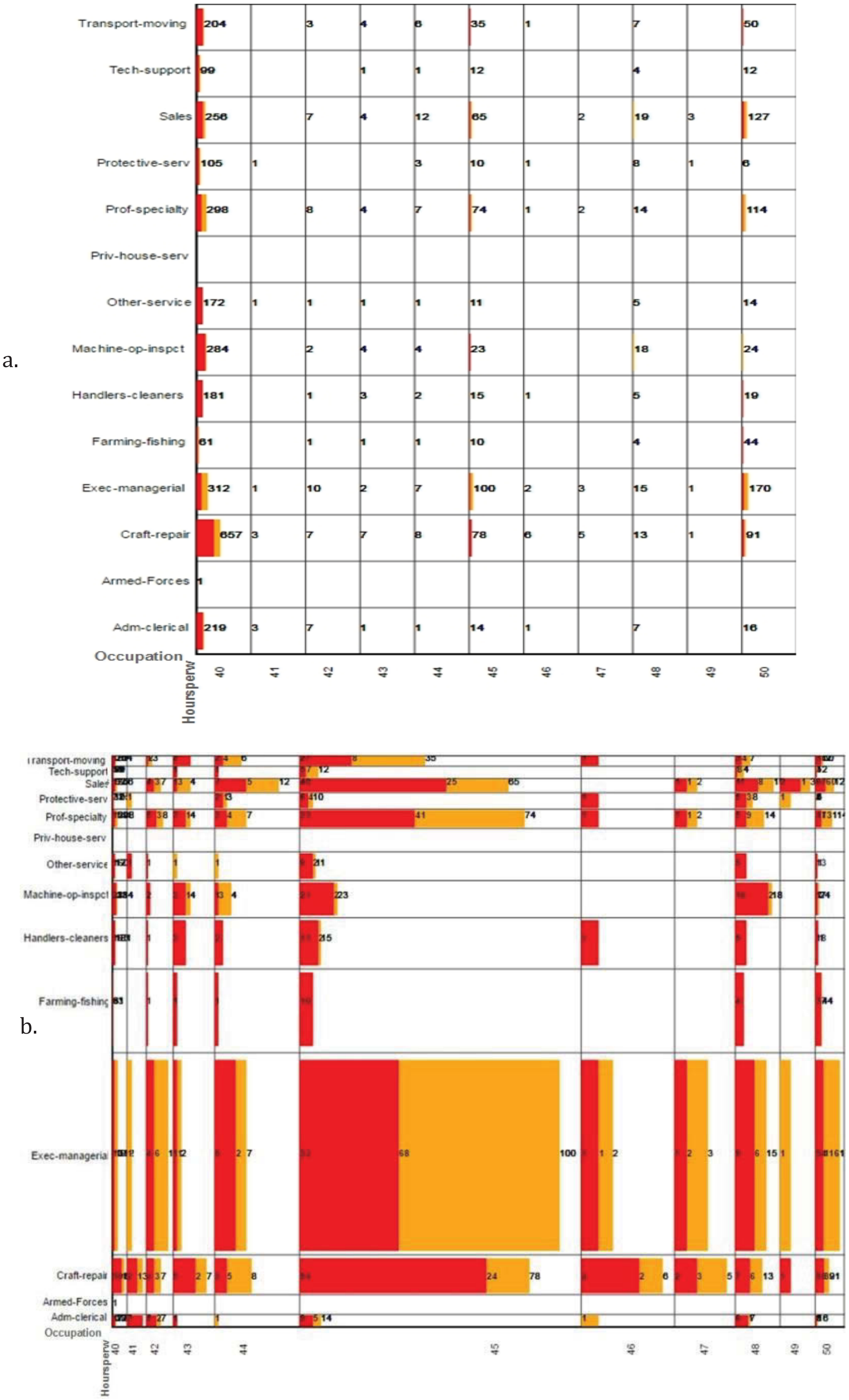


Figure 6.12 The quantitative plotting comparison between *Occupation* and *40-50 HoursPerWeek*.

Continuously for in-depth exploration, we drill down the context of *HoursPerWeek* with the concentration on the greatest number of working time in the ranges from *40* to *50*. We concern all of the *Occupation* and *40-50 HoursPerWeek*. By observing the final pattern (see Figure 6.12.a), it is easy to recognize that most of the people worked *40, 45,* and *50* hours per week. We are curious to see the details of those who worked *45* hours per week. By employing the visual enhancement with the fish-eye distortion (see Figure 6.12.b), it is clearly indicated that the largest population earning *50K*-and-lower incomes was at *Craft-repair*, with *54* cases, whereas that earning over-*50K* incomes was at *Exec-managerial*, with *68* ones. The smallest number of the population having *50K*-and-lower incomes was at *Tech-support*, with *5* cases, while that having over-*50K* incomes was at *Handlers-cleaners, Machine-op-inspct,* and *Other-service*, with 2 of each. There were no data recorded of both *Priv-house-serv* and *Armed-Forces* in this context.

6.3.2 Multiple-Context Queries of Data Models and Multi-Dimensional Data

This use case shows the interactive data exploration through the automatic interconnection between logical frames and coherent operation of MCquery, SumUp, and FigAxis.

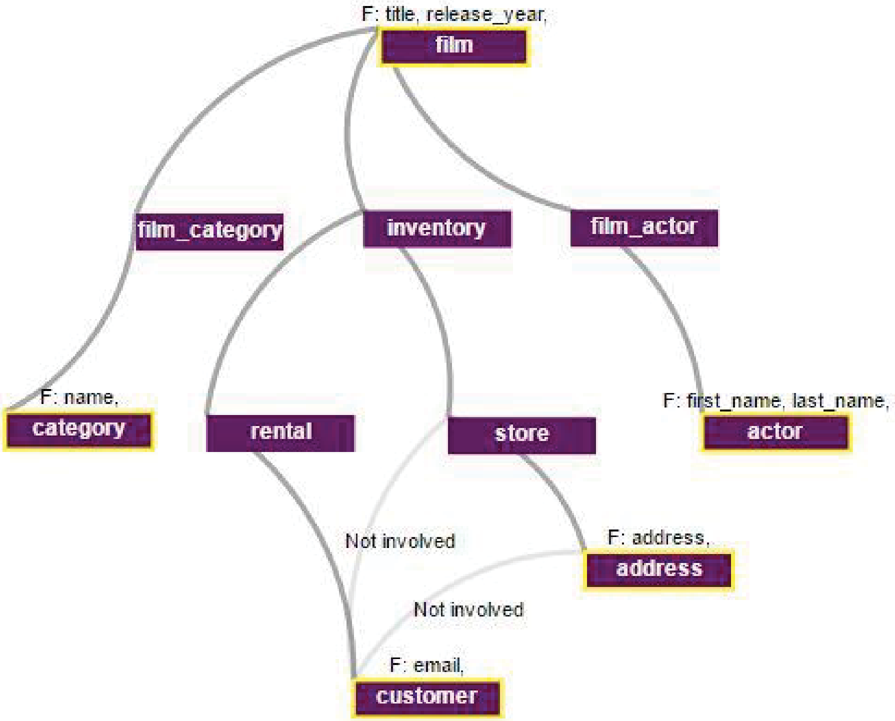


Figure 6.13 The data-model context of *customer, film, category, actor,* and *store*.

The assumed goal is to explore different facets of the DVD rental store data. From the data-model context of MCquery, we browse the data, and then we are curious to look for the information mainly about *customers, films, categories, actors,* and *stores* (see Figure 6.13). We go through a number of queries such as *who are the customers watching given films*, *what is the category the films belong to*, *where is the store the films are available*, *who are the actors acting in the given films*, etc.

Once discovering those, we want to narrow the consideration by drilling down to a few of data tables. They are *store, film,* and *category* with their representatives including *storeaddress, filmreleaseyear,* and *categoryname*, and we would like to compare the total number of films at the stores in *filmreleaseyear 2006*. After *storeaddress* is selected for the target comparison, and the transmission to SumUp is executed, the data pattern is clearly displayed (see Figure 6.14).

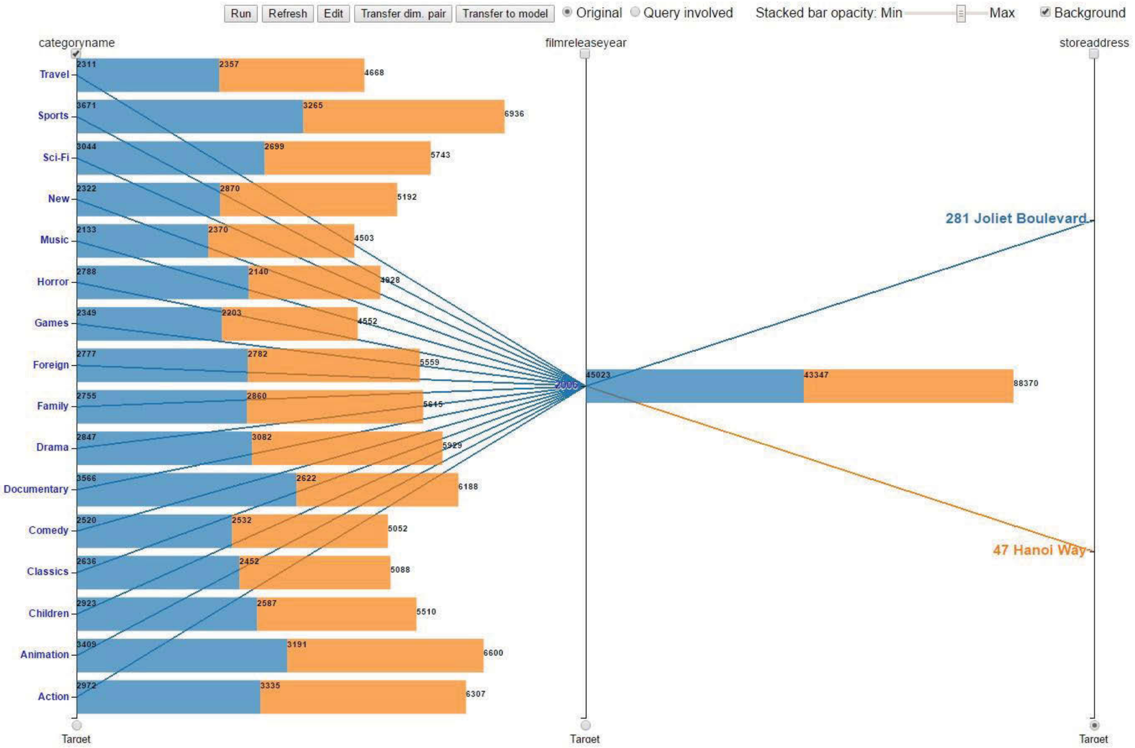
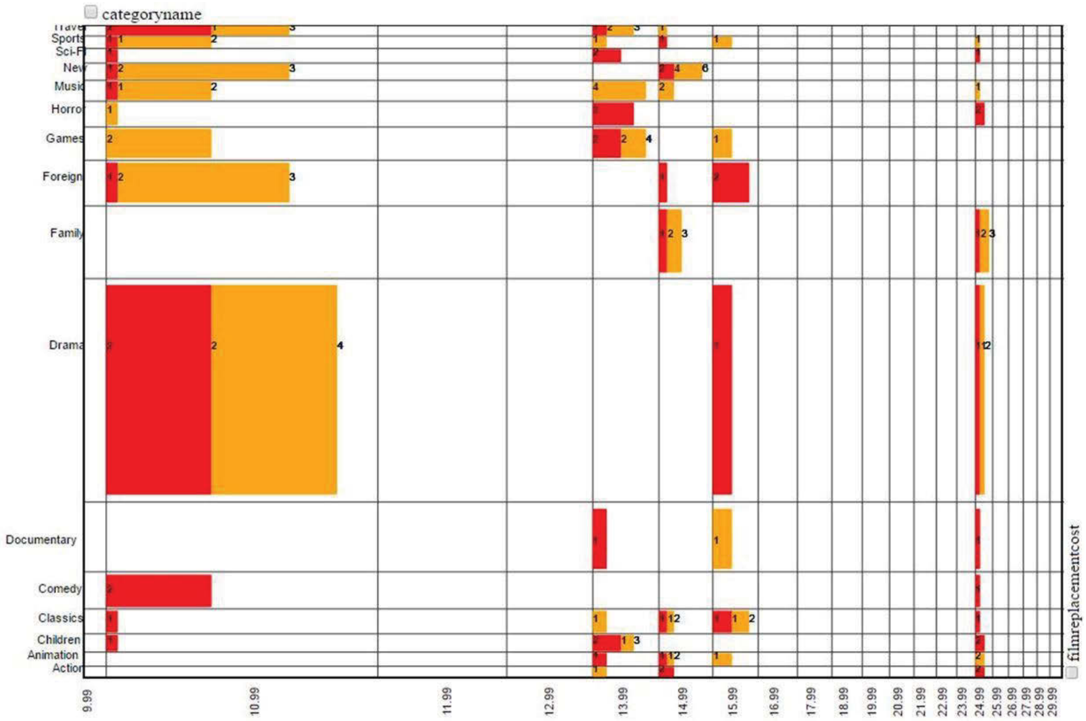


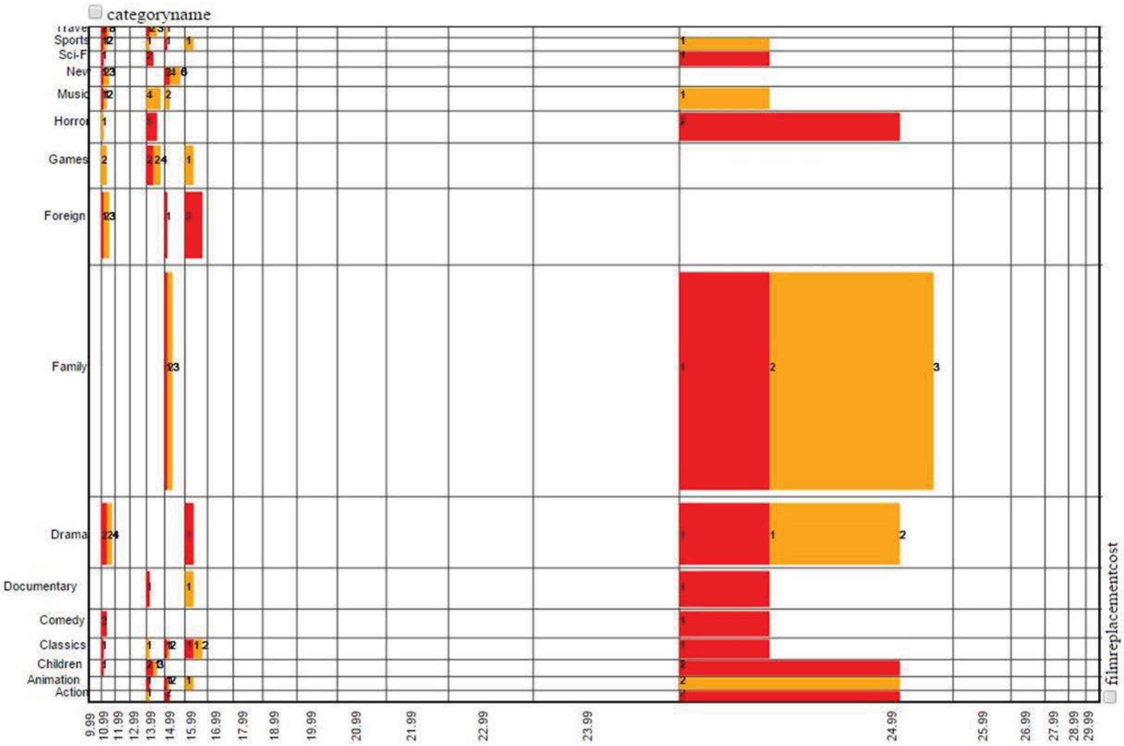
Figure 6.14 the multi-dimension context of *categoryname, filmreleaseyear, and storeaddress.*

The significant difference in Figure 6.14 was at the smallest number of the films of *Travel, Music,* and *Games*, with around *4500* to *4600*, compared to the largest number of films of *Sports, Action,* and *Animation*, with *6300* to nearly *7000*. In term of the total number of the films available at the stores, there was no very great difference in these, with *43347* for the store at *47 Hanoi Way* and *45023* for the one at *281 Joliet Boulevard*, which was about *2000* films unequally.

Later, we wonder what the other details of the films are available and valuable for the exploration, so we return the data-model context. Here, we are curious about *Replacement cost*, the cost customers have to pay in case of films damaged or lost, and *Rating*, the audience suitability classification. Thus, we would like to learn the correlation between *Category* and *Replacement cost* towards *Rating*. There are five rating levels including *G, PG, PG-13, R,* and *NC-17*, equivalent to *General Audiences, Parental Guidance Suggested, Parent Strongly Cautioned, Restricted,* and *No one 17 and under admitted*. We are eager for comparing *PG-13* and *NC-17*, so these are configured for the target in the next stage.



a)

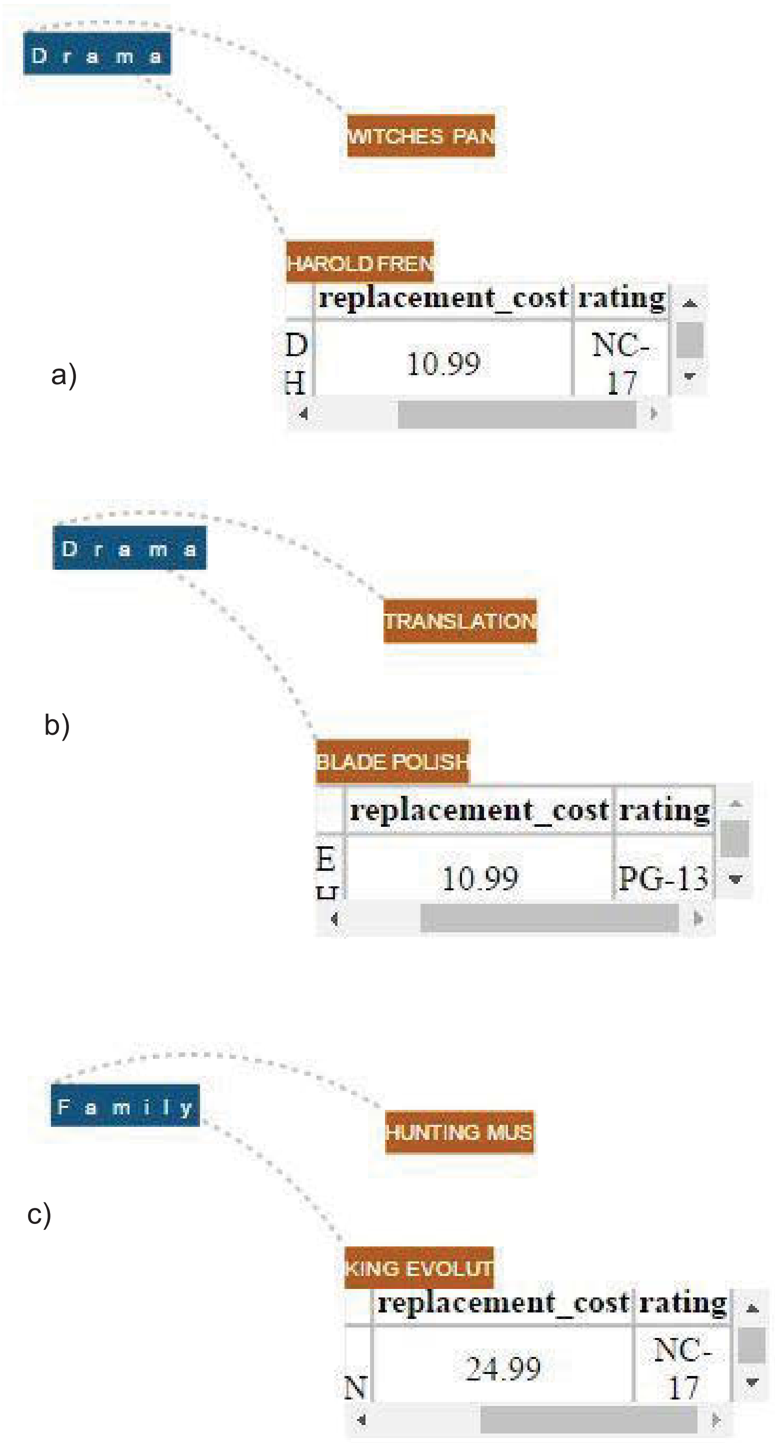


b)

Figure 6.15 The pairwise comparison of *categoryname* and *filmreplacementcost*. Figure *a* focuses on cost *10.99* whereas Figure *b* concentrates on cost *24.99*.

After the transmission of those to the pairwise comparison context of extended FigAxis, the result pattern is shown in Figure 6.15. The vertical axis is *categoryname*, and the horizontal one is *filmreplacementcost*. The red color encodes *PG-13*, and the yellow one encodes *NC-17*. There were only *5* of *21* cost levels affected by the comparison including *10.99, 13.99, 14.99, 15.99,* and *24.99*. There was no significant difference of the number of the impacted films, with the range from *1* to *4* films. For the lowest cost *10.99*, the biggest number of the influenced films was of *Drama*, with *4* totally, *2* of each rating, while there was no film of *Family, Documentary, Animation,* and *Action* influenced. For the highest cost *24.99*, the most impacted films were of *Family*, with *3* totally, *1* of *PG-13* and *2* of *NC-17*, while there was not any impact on the films of *News, Games,* and *Foreign*.

Further, we would like to know exactly the name of the impacted films of *Drama* and *Family* in the comparison of costs *10.99* and *24.99*. By transferring the considered values to the data-model context of MCquery, we receive the answer clearly shown in Figure 6.16, and then we click on the considered nodes to see the film details.



a)

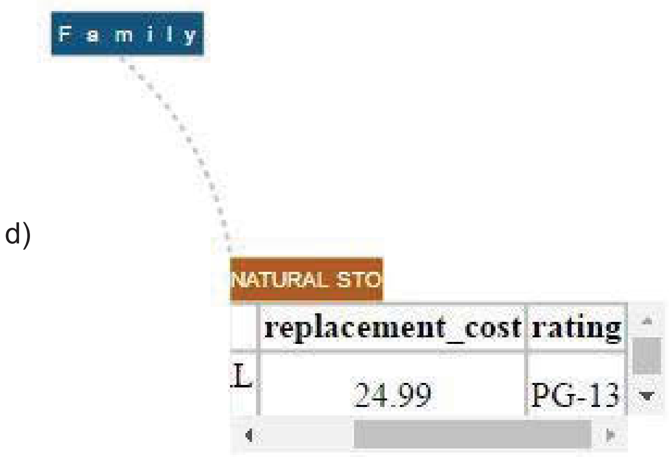
	replacement_cost	rating
D H	10.99	NC- 17

b)

	replacement_cost	rating
E	10.99	PG-13

c)

	replacement_cost	rating
N	24.99	NC- 17



	replacement_cost	rating
L	24.99	PG-13

Figure 6.16 The name of the impacted films of *Drama* and *Family* in the comparison of costs *10.99* (Figures a, b) and *24.99* (Figures c, d) visualized in the data-model context of MCquery.

6.3.3 Discussion

Data mining through visualization is not a new subject, but always in active consideration. In relational data application, besides of a large number of single visual techniques proposed, many integrative tools are presented for incorporated data exploration such as Tulip, Orion, Polaris and Tableau. The current models focus on using various kinds of graphs for visual analysis of given data in different facets. Thus, users have been currently accessing the diverse methods that most probably meet their exploring objectives. For instance, statistical tasks probably work well with bar and pie charts, and object-relation observation likely adapts with network graphs and tree maps. The flexibility of the integrated tools brings users many benefits; however, the procedure of interconnection establishment is still a challenge, especially for the novices who have not much technical knowledge. The advanced tools often require programming skills in configuration of data representation and further investigation, which can cause increase of users' metal and physical efforts to reach discovery goals and might interrupt continuous exploration.

In the above scenarios, the core of the procedures is for coherent exploration by multiple visual contexts of graphics with automatic interconnection. Thus, users are able to perform further investigation across the representations by simple interaction without the interruption of technical setting procedure. Novices and general audiences are preferred by our motivation which is to create a friendly tool for a right job with various targeted users.

6.4 Summary

This chapter has presented the typical case studies of our developed techniques in term of functional usefulness. These visual query techniques can either work independently or interconnect cooperatively, which is able to support both single analysis activities and further relative investigation with the utilization of data models and multi-dimensional data. The case study results indicate that our proposed techniques probably support and deal with the analysis tasks addressed by the dissertation's objectives such as information search-retrieval, correlative observation, and in-depth exploration.

Chapter 7 Evaluations

This chapter presents the procedures of evaluation for our techniques in term of space-efficient visualization, distinctive features, and the friendliness of the techniques, which are the empirical objectives of this research. Space efficiency is a critical key in graphic user interface design since it motivates the attempts of increasing the amount of information represented in a limited display space. Many of the related works revisited in this dissertation such as in Chapter 2, Chapter 4, and Chapter 5 greatly considered and discussed this criterion. Thus, for the evaluation section, this criterion could not be excluded. In addition, one of the primary objectives of this dissertation is to create expert tools for ordinary users, and all of the proposed techniques were designed and implemented in the effort of bringing convenience to the users. Therefore, the friendliness criterion chosen for the evaluation is definitely necessary. For creativity measurement of the proposed techniques, the distinctive feature comparison would be conducted between existing tools or software and the proposed tools in this section as well.

The procedures include the performance analysis, the feature comparison, and the usability study that have been conducted with three developed techniques MCquery, SumUp, and FigAxis. The goal of these evaluations is to observe how the developed techniques meet the research objectives, including both application and theory aspects, particularly for the new data exploration framework proposed in Chapter 3.

7.1 Space-Efficient Visualization

7.1.1 Dynamic Representation

We measured the space-efficient visualization of MCquery by the comparison of the dynamic tuple representation applied in MCquery and the key-value representation method widely applied in node-link graph visualization. The considered parameters were the reduction number of displayed nodes and links of the two methods. To simplify the assessment, it was supposed that the activated case was the visualization of a relational query result *R,* including two tables $T_1(m,n)$ and $T_2(m',n')$, where *m* and *m'* were the number of instances involved in the query, and *n* and *n'* were the number of dimensions involved in the query.

Then, the number of the nodes and links with the key-value representation were defined as follows.

$$F_{kv}(N) = mn + m'n'$$

$$F_{kv}(L) = nm'n', m'>= m$$

Equation 7.1

On the other hand, those numbers with the dynamic tuple representation were equivalent to:

$$F_{dt}(N) = m + m'$$

$$F_{dt}(L) = nm', m'>= m$$

Therefore,

$$r(N) = 1 - \frac{m + m'}{mn + m'n'}$$
$$r(L) = 1 - \frac{1}{n'}$$

Equation 7.2

where *r(N)* and *r(L)* were the reducing rates of the displayed nodes and links, which were equivalent to the saving rates of the node-link graph display of MCquery. Figure 7.1 and Figure 7.2 illustrate the correlation between the size of the two tables and the space-saving rates of MCquery compared to the key-value method.

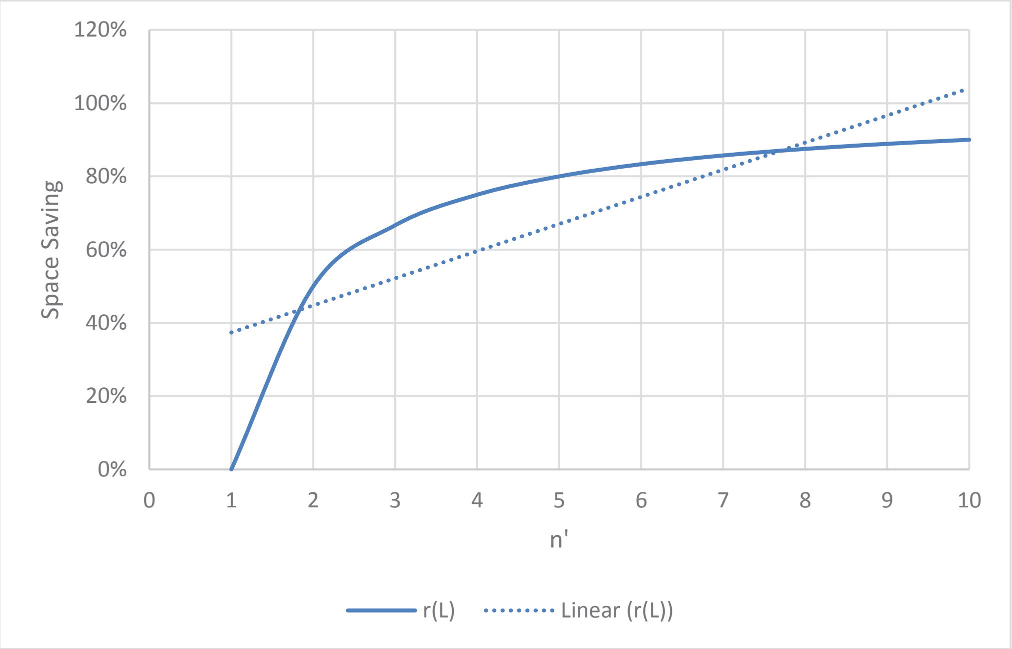


Figure 7.1 The space-saving rates in term of the link display of MCquery.

The Figure 7.1 indicates that the space-saving rates in term of the link display were always higher *40* percent when there was more than one dimension in T_2 evolved in the query. However, MCquery method had no impact on the saving display space in the case of only one dimension of the two tables involved in the query.

To simplify the space-saving assessment for the node display, it was assumed that the parameters in the following table were chosen for T_1 and T_2.

Variables	m	n	m’	n’
Values	100	1	1-100	10

Table 7.1 The parameter values chosen for the space-saving assessment in term of the node display of MCquery.

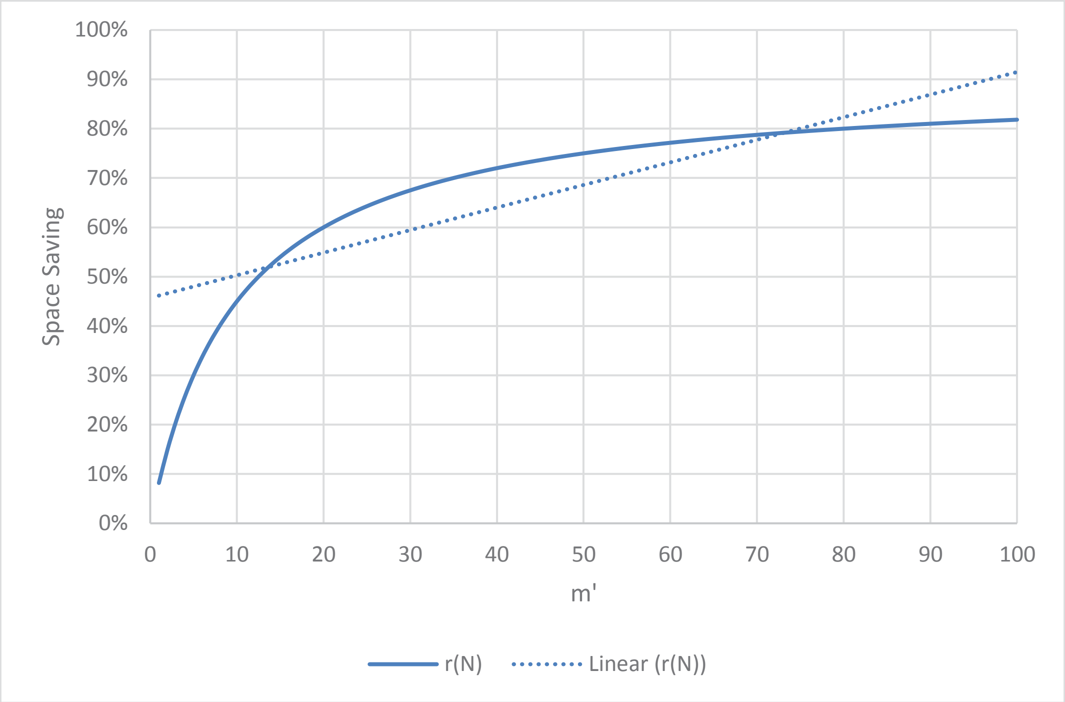


Figure 7.2 The space-saving rates in term of the node display of MCquery.

The Figure 7.2 points out that the space-saving rates in term of the node display were always higher 40 percent when there were more than ten instances in T_2 involved in the query. Nonetheless, the rates were lower 40 percent in the case of only one dimension of the two tables involved in the query.

7.1.2 Layering Display + Sharing Axes

We evaluated the space utilization of SumUp and FigAxis by the capability of the number of graphs displayed in the space of a basic graph or the background of a graph. The considered basic graphs were parallel coordinates and scatterplots, which are the two typical characteristics of the visualization methods applied in SumUp and FigAxis. The following paragraphs are the discussion about the features of the display layering and the axis sharing for the space utilization of the two methods.

The display layering is a well-known technique in the enhancement of showing multiple graphs within a limited display space. In theory, a layer can represent more than one graph type, yet in practice, that number is often kept at one due to the clear-view maintenance and high-overlapping prevention. In our designs, the employment of double layers is to increase the number of displayed graphic types at least twice in a single view.

The axis-sharing method is a simple idea, but a compelling solution in the space optimization, especially in parallel coordinates and scatterplots. This approach can support the enhancement of the number of displayed graphs or charts and meanwhile reduce the axes associated with those.

For SumUp, the increase of the number of the displayed stacked bar charts $N_{stackedgraphs}$ is a positive correlation related to the number of the activated dimensions $N_{dimensions}$ of the parallel-coordinate display space, which is equivalent to

$$\mathrm{N}_{\mathrm{stackedgraphs}} = \mathrm{N}_{\mathrm{dimensions}} \text{ and}$$

$$\mathrm{N}_{\mathrm{displayedgraphs}} = \mathrm{N}_{\mathrm{dimensions}} + 1$$

Equation 7.3

For the scatterplot visualization of FigAxis, those numbers are

$$\mathrm{N}_{\mathrm{stackedgraphs}} = 2 \text{ and } \mathrm{N}_{\mathrm{displayedgraphs}} = 3.$$

The axis-saving ratio of FigAxis is fifty percent (50%) in the case of a standard scatterplot visualization embedded with two stacked charts. It is

$$\mathrm{N}_{\mathrm{usedaxes}} / \mathrm{N}_{\mathrm{thoeryaxes}} = 2/4$$

where $N_{usedaxes}$ and $N_{thoeryaxes}$ are the number of the empirically used axes and the number of required axes in theory respectively.

That rate of SumUp is also 50 percent (50%) in the case of a parallel coordinate graphic ($N_{dimensions}$) embedded with $N_{stackedgraphs}$. With recalled Equation 7.3, we have

$$N_{usedaxes} / N_{thoeryaxes} = N_{dimensions} /(N_{dimensions} + N_{stackedgraphs})$$

$$\Rightarrow \quad N_{usedaxes} / N_{thoeryaxes} = 1/2.$$

7.2 Distinctive Features

We assessed the distinction of our interactive visual queries by comparing their features to the current similar tools.

7.2.1 Relational Query Making through Node-Link Graphics

We conducted the feature comparison of MCquery and the current similar tools including the visual query tool of Microsoft Access (Microsoft 2017) and Ploceus (Liu, Navathe & Stasko 2011). The result is illustrated in Table 7.2 as follows.

Features **Tools**	**Visual queries**	**Node-link manipulation**	**Direct manipulation**	**Further investigation**
MCquery	Graph-based	Model-based	Yes	Model-based and Query-result based
Visual query tool of Microsoft Access (Microsoft 2017)	Form-based	Model-based	None	None
Ploceus (Liu, Navathe & Stasko 2011)	Graph-based	User-defined	None	Query-result based

Table 7.2 The feature comparison of MCquery, the visual query tool of Microsoft Access, and Ploceus.

According to Table 7.2, MCquery featured the direct manipulation and supported the further investigation on both the model representation and the query result representation, while the Microsoft Access tool and Ploceus did not feature the direct manipulation, and one of them only supported the further exploration by query result visualization. The difference of these facilities might be influenced by the usage purpose and the target user of the tools. MCquery provides the common data exploration features for non-technical and novice users, whereas the Microsoft tool is mostly for the expert database management, and Ploceus focuses on the practicing and experimenting user-defined queries for experienced users.

7.2.2 Quantitative Query Making through Parallel Coordinates

As discussed in Chapter 5, the two tools of Siirtola (2002) and Ho et al. (2011) are currently considered to be the closest to our tool SumUp. Thus, we created the feature comparison of SumUp, the tool of Siirtola (2002), and the tool of Ho et al. (2011), which is shown in the following table.

Features / **Tools**	**Mean/Median determination**	**Flexible data support queries**	**Multiple targets**	**Correlation across dimensions**	**Query operators**	**Histogram views**
SumUp		✓	✓	✓	✓	✓
Siirtola H. 2000	✓				✓	
Ho Q. et al. 2011	✓					✓

Table 7.3 The feature comparison of SumUp, the tool of Siirtola (2002), and Ho et al. (2011).

Table 7.3 indicates that only SumUp supported the flexible-data queries (the queries for both numerical and categorical data), the multiple target comparison, and the correlation tracing across dimensions. Whereas Siirtola (2000) and Ho et al. (2011) concentrated on the mean-median determination with single targets, SumUp focused on the data summary with multiple targets across dimensions.

7.2.3 Quantitative Query Making through Scatterplots

With our best knowledge, we could find neither technique nor tool which is similar to FigAxis and extended FigAxis. FigAxis is a technique enabling to quantitative queries by the direct zooming on a plotted-data zone with the sharing scatterplot axes. Certainly, FigAxis is a basic method which is offered and experimented independently, so it would be essential to be under further tests and refinement for widespread deployment.

7.3 Friendliness of Techniques

We evaluated the friendliness of our techniques by qualitative user studies in term of the ease of use, usefulness, interesting visualization, and the user preference. As mentioned in the objectives of this dissertation, non-technical and novice users are preferred by our approaches.

7.3.1 Query Making with Node-Link Graphics

MCquery is developed for the querying data which is converted from ER model. This model is a favorite study subject of diverse courses such as information technology (IT) and business administration (BA) in universities and colleges. Therefore, our idea is to evaluate the friendliness of the technique by a user study on the undergraduate students of the two above majors.

We recruited 24 participants, with 10 from IT and 14 from BA, including 9 men and 15 women. These IT students studied ER model, database theory, database management (Access and MySQL), and query languages (SQL – Structured Query Language), whereas the BA students had the knowledge only about ER model application for information system management.

The experiment lab featured workstation desktops, HP Compaq PCs, and LCDs, with Intel Core i5 and Windows 7. An MCquery system was hosted on a local server and accessed via the web browsers of Google Chrome v.54.0.

Firstly, the experiment goal, the functions, and the how-to-use procedure of MCquery were introduced to the participants. Then, they had 15 minutes to practice and play with MCquery on sample data to become familiar with the tool. When ready, every participant was asked to perform the four tasks of the use cases of MCquery in Chapter 6. Their time for completing all the tasks was recorded, and a questionnaire was delivered to receive their feedbacks. The questionnaire was mainly about the qualitative measurement in term of the ease of use (A/A’), usefulness (B/B’), interesting visualization (C/C’), and open questions about pros and cons of the using MCquery, where A, B, C and A’, B’, C’ refer to the criterion associated with the IT group and the BA group respectively. The used grading system was based on Likert with the domain from 1 to 7 for *completely-disagree* to *completely-agree* assessment.

Overall, the results were optimistic with 6.2 (Mean) for the ease of use, 6.1 for usefulness, and 5.8 for interesting visualization. The detailed comparison between the IT and BA groups is illustrated in Figure 7.3.

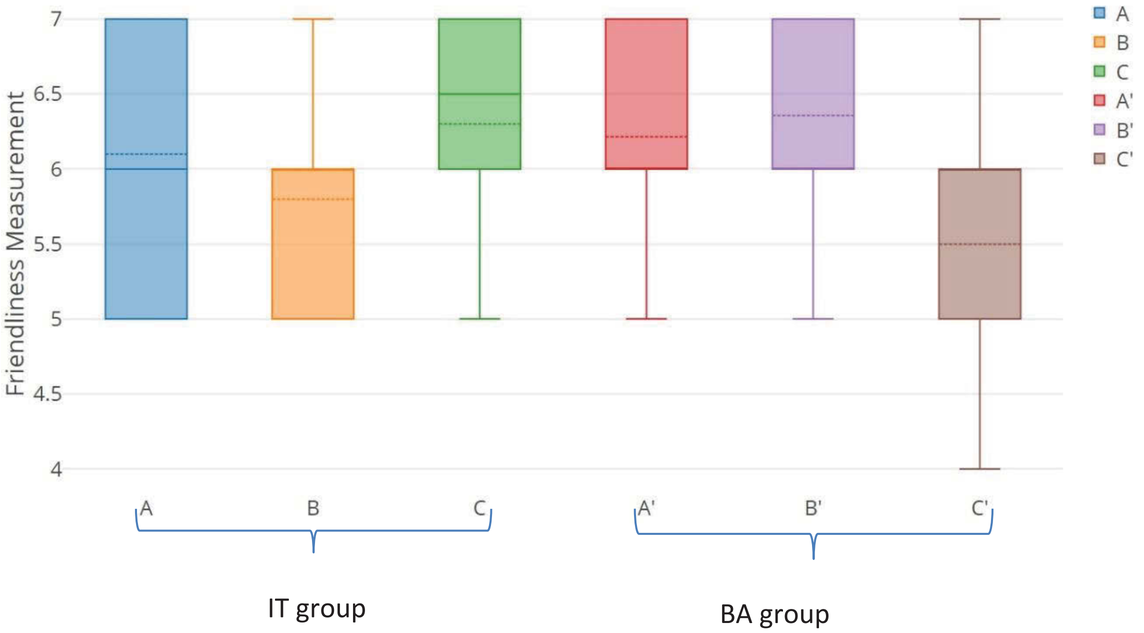


Figure 7.3 The detailed comparison between the IT and BA groups on the MCquery feedbacks.

According to Figure 7.3, the mean grades (the dash lines) of the IT group were 6.1, 5.8, and 6.3, while those of the BA group were 6.2, 6.4, and 5.5, respectively to A-B-C and A’-B’-C’. In general, there was no significant difference of the mean grades of the two groups. The ease-of-use feedbacks were almost same, with 6.1 and 6.2 of IT and BA. There were slight differences for the remaining feedbacks. For the usefulness, the IT group marked 5.8 while the BA marked 6.4. Inversely, for the interesting visualization, the IT group assessed 6.3, whereas the BA one assessed 5.5.

The open comments of the BA group were mainly that people could easily get the finding information and desired data from the management diagram (ER diagram) with a minimum of IT background, and many of them felt interested in the interaction by the direct clicks on the images. For the improvement feedbacks, this group suggested that there should be more statistical functions to be more helpful in BA tasks and needs, and a few of the participants did not like the layout of the query results in the case of the rush exploration with a big network graph. On the other hand, the comments of the IT group were mostly that the queries by the click action was simpler and faster than the queries by text-based commands such as SQL, and they could explore ER-model based data instantly without the database configuration like the usage procedures of Access and MySQL. For the constructive comments, this group said that in the case of the increased node-link clutter, besides the currently featured navigation, there should be more other options to deal with it, and the dynamic animation was not smooth sometimes.

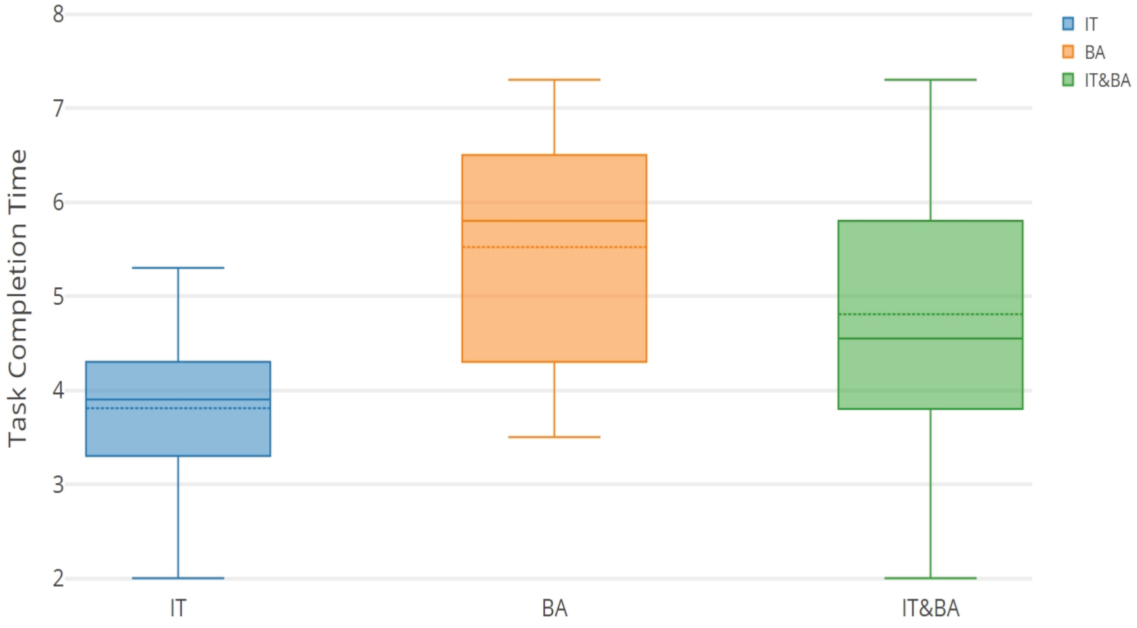


Figure 7.4 The task completion time of the MCquery usage for the IT and BA groups.

Overall, the positive feedbacks were resulted by the simple and direct manipulation of MCquery on ER model based visualization, which does not require the technical knowledge or deep experience in usage. The average time for the task completion of the IT and BA participants was 3.8 and 5.5 minutes respectively and was 4.8 for all the participants (see Figure 7.4). These results were positive. Although the time of the BA group was nearly half greater than the IT group, it is still considered as a short time to deal with all the tasks. The time difference might be because the IT students were familiar with the graphic interaction more than the BA ones, especially for the network layout. This situation might be a reason to support the fact that the fastest completion belonged to the IT ones, with around 2 minutes, and the longest work was from the BA group, with around 7.3 minutes.

The suggestion of the statistical functions was quite reasonable, and it seems favorable for the BA-driven professional tools. Thus, such an idea is valuable for the further development since MCquery is, currently, a general exploration tool with the common tasks concentrated on the information search and retrieval. The scalability of the MCquery layout was mentioned by both the groups. In fact, the visual scalability is an easily found challenge in information visualization in the case of the displaying a large number of visual objects in a limited space. The core method of MCquery helped to save the display space in some aspects which were evaluated in Section 7.1.1, and the visual enhancement was carefully considered as well. The filtering and zooming feature of MCquery would probably help in most of the cases only when they are applied to each query instead of after a set of queries. In other words, the *explore-filter-explore* procedure would be better for the display space than the *explore–explore–filter* way. For the dynamic animation that can impact on the interesting visualization criterion, the commented issue would be significantly improved when the system is hosted on a real server and accessed via updated web browsers.

7.3.2 Query Making with Parallel Coordinates

As discussed in the previous sections, the characteristic of SumUp is distinctive from the existing parallel coordinate browsers. At this stage of the research, we evaluated the friendliness of SumUp in term of the ease of use, usefulness, and interesting visualization by a qualitative user study.

We recruited ten participants including four women and six men in ages from 23 to 40. All of them were from engineering and computer science and had experience in the statistical analysis by many kinds of graphs including stacked bars and bar charts. While two of them had the practical knowledge of parallel coordinates, the other eight participants were the novices in it.

The experiment lab featured workstation desktops, HP Compaq PCs, and LCDs, with Intel Core i5 and Windows 7. A SumUp prototype was hosted on a local server and ran on the web browsers of Google Chrome v.54.0.

After given the experiment goal, the introduction of parallel coordinates, the features of the SumUp browser, and the usage instruction of it, the participants arbitrarily used the SumUp browsers to play and explore the Car and Census income data sets in 15 minutes. Once getting well familiar with the tool, the participants were required to complete the three tasks of the use cases in Chapter 6 including the tasks for multiple-attribute comparison, flexible data support, and correlative analysis.

Then, after the task completion, their feedbacks were recorded via a questionnaire for measuring the ease of use, usefulness, and interesting visualization of SumUp. The used grading system was based on Likert with the domain from 1 to 5 for *strongly-disagree* to *strongly-agree* assessments.

The results were quite positive with 4.5 (Mean) for the ease of use, 4.6 for the usefulness, and 4.3 for the interesting visualization. The open feedbacks indicated that it was easy to make the statistical queries by the click and brush action and make the comparison by the stacked bars, and they could quickly compare many attributes at the same time on the multiple axes. The constructive comments were that there should be a feature like a browsing history to trace and compare the correlation and changes of the visual patterns, the filtering by multi-criteria-based classification should be considered on the axis ticks, and the statistical proportion should be added for more flexible data-label display.

In general, the high satisfaction from the participants was due to the simplicity and friendliness of the SumUp usage, though a majority of them had little experience in it.

The conventional click and brush action on the parallel axes made the query creation and adjustment quite simple and fast, and the observation of the statistical comparison was supported well by the developed stacked bars, which are a quite regular chart type of this area.

The constructive comments recorded from the user study would be significantly valuable for the SumUp improvement. During the data discovery process, the users might be either forget previous steps or pay much effort to connect the concrete query results; therefore, a browsing history might be useful to help the tracing and matching the relative results. Another limitation is that the displaying a large amount of data in an axis might cause the overlapping of the scaling tick marks and labels. Although SumUp featured the colour-based filtering, the users might face the difficulty in the case of the high density of the scaling ticks. The multi-criteria-based classification could be a good option to make the query details more concentrated, especially in the query formulation and the adjustment steps. Another finding is that most of the non-technical users preferred watching the bar charts to observing the polyline patterns.

7.3.3 Query Making with Scatterplots

One of the key objectives of the query design with the scatterplots is to reduce the users' effort in observing and matching the details between the plotted-data pattern of the scatterplots and the pattern of the query results. The proposed technique FigAxis is to overcome the limitation of the multi-view layout in term of the addressed aspect. Therefore, we conducted a comparison survey for the friendliness measurement between the single advanced layout of FigAxis and the layout of multiple views. In the scope of this study, a multi-view layout refers to the layout in where the plotted-data pattern of the scatterplots and the pattern of query results are located in single views linked together.

The questionnaire serving for the study is mainly about four criteria below:

- (A/A') How easy to navigate the focus zones of data plots?
- (B/B') How easy to recognize the difference of various dimensions?
- (C/C') How easy to track and match the data density measurement between the dimensions and the focus zones?

- (D) What is your preference?

A, B, C and A’, B’, C’ are for the concerning of the FigAxis layout and the multi-view layout respectively. We recruited 9 participants, including 4 women and 5 men, with the ages of 26 to 34 years; all were graduate and well knew the visualization of scatterplots and bar charts. The survey procedure was deployed as follows.

The participants were given an introduction to the meaning, goal, and method of the study. Then, all of them carefully watched the pairs of photos via the vertically dual screens, where the left one was a use case of the FigAxis layout, and the other right one was a use case of a multi-view layout. The used multi-view layout consisted of a single-standard-scatterplot view horizontally placed between two vertical-stacked-bar views. The participants were allowed to watch the photos as much time as they want, and when being ready, they used the questionnaire to express their feedbacks. The used grading system was based on Likert with the domain from *1* to *7* for the *completely-hard* to *completely-easy* assessments.

In general, all the participants preferred to the FigAxis layout, which was not out of our prediction. The detailed result is shown in Figure 7.5.

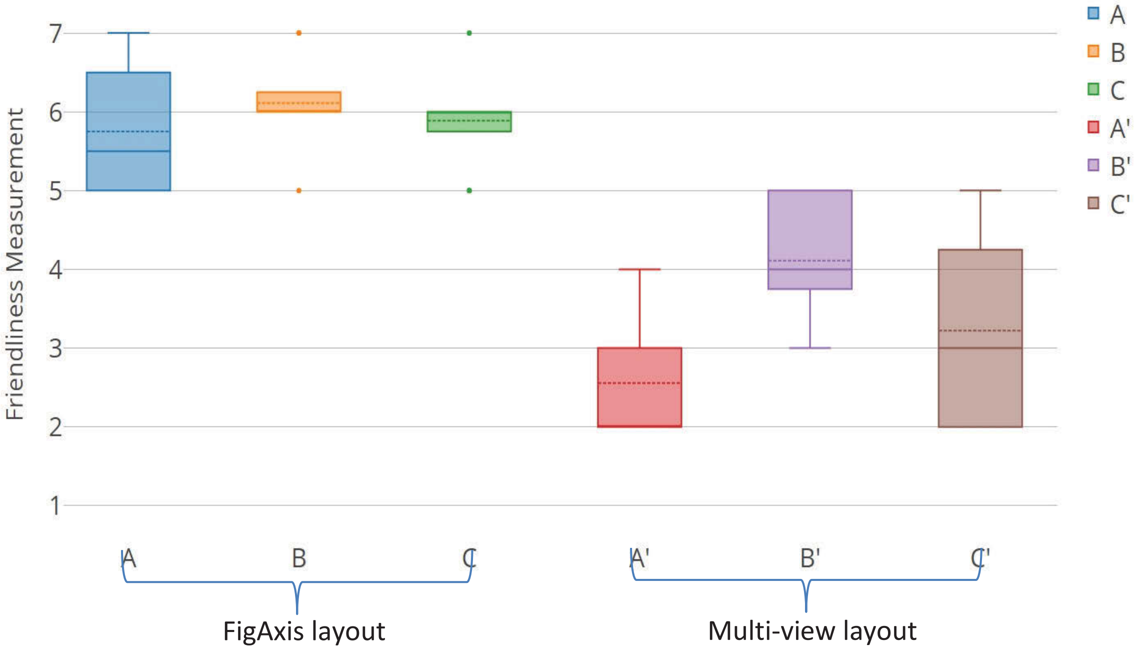


Figure 7.5 The feedback result of the friendliness comparison between the FigAxis layout and the multi-view layout.

It is clearly shown in Figure 7.5 that there was a great difference in the mean grades (the dash lines) for the focus zone navigation. The FigAxis layout received 5.8 (A), while the multi-view layout had 2.6 (A’). The scores might reflect that FigAxis well supported the focus zone by the zooming+paning navigation, whereas the multi-view layout had no such a feature. The focus-context flexibility is the strength of FigAxis, while this seems to be the hardest task by the other. There was a small difference in the mean grades for the various dimension identification. The FigAxis layout had 5.4 (B), whereas the layout of multiple views had 4.1 (B’). This criterion was the best score of the multi-view layout, yet still not good as FigAxis. As with the first criterion, there was an important difference in the mean grades for the data summary tracking and matching. FigAxis got 5.9 (C), the highest score, while the multi-view layout got 3.2 (C’).

The open comments confirmed that for the multi-view layout people need to move their eyes and to change the views frequently to find and match the considered information among the views, whereas with FigAxis there was no much such effort due to the close and flexible positions of the stacked bars and the plotted data points. For the constructive feedbacks, they were almost about the difficulty in specifying the zooming levels due to the unstable control of the scrolling wheel of mouse devices. Specifically, the mouse wheel was scrolled, yet the zooming did not show the effect. For such an issue, we tested and classified it as the objective consideration since either the mouse wheel was not maintained carefully or the users made the manipulation mistakes.

7.4 Discussion

This chapter has presented the evaluations of our developed new techniques in term of the space-efficient visualization, the distinctive features, and the friendliness of the techniques. The overall results were quite optimistic towards the goals of this dissertation, and the limitations were carefully addressed as well. Of course, what has been done of this research up to the present time is just in the beginning stage of long-term studies, and the achievement still needs the further advanced works for the widespread deployment.

Chapter 8 Extended Work

This chapter presents a dynamic and interactive query system applying double layer display, zooming, and brushing features. Notwithstanding that this additional work is not a primary contribution to the goal of this dissertation, its theory and practice are closely related to the visual queries developed for the data analytics presented in the previous chapters. We illustrate the query method via topic *multiple-tag visual rankings*.

The multi-tag-based search is quite popular on collaborative websites and sharing-online-content systems. For this kind of search results, the challenge is how to compare the grouped-tag values of tag collections on heterogeneous alternatives. This chapter introduces a new visualization approach named Qstack for dealing with the challenge. Qstack's purpose is to help users to rank multi-tags visually based on the grouped-score combination within and across the categorized alternatives. The methodology applying interactive stacked bars, dynamic queries, and an adaptive focus+context technique enables users to create and adjust the grouped-tag rankings of a vast number of the heterogeneous alternatives. A case study on the Flickr photo award allocation will be presented for the Qstack demonstration. We conducted a qualitative study for evaluating the Qstack effectiveness, and the result indicated that our approach is probably useful for the multi-tag rankings.

8.1 Introduction

Along with the development of social networks, the multi-tag-based queries are becoming popular for meta-data construction and information seeking such as the photo search on Flickr and the book search on LibraryThing (https://www.flickr.com/, 2016; https://www.librarything.com/, 2016). The characteristics of a multi-tag-based search might lead to a heterogeneous result which includes both the totally satisfying items and partially satisfying ones. Normally, the total satisfaction is concerned; however, in the sophisticated analytics, the remaining ones might be valuable as well. For instance, we are looking for books by terms *hacker, internet, security,* and *secret* and receive a number of diverse books. Besides the four-tag-satisfying books, we might be still interested in the three-or-two-tag-satisfying ones. Furthermore, the multi-tag values could be used to support decision-making procedures. For instance, in a photo association, using the scores of tag values towards topics to rank members is essential for judges to make decisions on annual award allocation. The topics could be formed by single tags and the groups of tags, which would lead to the heterogeneous alternatives. Our motivation is to deal with the challenge of analyzing the heterogeneous items given by flexible sets of considered tags.

For the generalization of the topic, the multi-tag comparison is referred to the multi-attribute comparison due to the similarity of score computation algorithms. We have studied the multi-attribute visual ranking techniques and believe they would be useful to display given items in visual patterns for analysis. The existing visual ranking tools can well support the combination of single or individual attributes such as Gratzl et al. 2013; nonetheless, they have not aided the rankings of the grouped-attribute combinations.

We propose technique Qstack, which enables to rank and analyze the grouped-tag combination within and across the categories of given items. Our methodology is to develop the visual rankings of multi-tags based on a chain of multiple stacked bars and dynamic queries. By employing an adaptive focus+context feature, users are able to navigate combined tags and clustered items easily during exploration.

8.2 Review of Tag Visualization

A tag is a word or a group of words assigned to online items or extracted from a text collection. While the multi-tag layout is commonly represented in lines, circulars, and clusters, the tag frequencies are visualized by text font sizes and colors (Clark 2008; Lohmann, Ziegler & Tetzlaff 2009). Tag cloud visualization is a well-known approach for analyzing text collections at a single time point and over time series. The combination of trend charts and tag clouds is a typical method proposed for the overtime tag comparisons (Cui et al. 2010; Lee et al. 2010). The charts show the change in the tag usage, and the clouds display the representative key words of the tags via force layouts. In addition to the trend charts, parallel coordinating visualization was offered for faceted text collection analysis (Collins, Viegas & Wattenberg 2009). The tag clouds of this visualization were represented in multiple columns, the associated items were visualized in stacked bars, and the tag relations were displayed in colorful lines, which can support tag-relation observation across collections. Besides, tag relationships could be analyzed by hierarchical networks with a semantic model (Di Caro, Candan & Sapino 2011). For the tag-based search, the clustering algorithms were usually used for improving such search results (Begelman, Keller & Smadja 2006; Hassan-Montero & Herrero-Solana 2006; Zubiaga 2009). For dealing with the variety of online information, the tag clouds could be combined with multi-graph interaction to improve navigation processes (Dörk 2008).

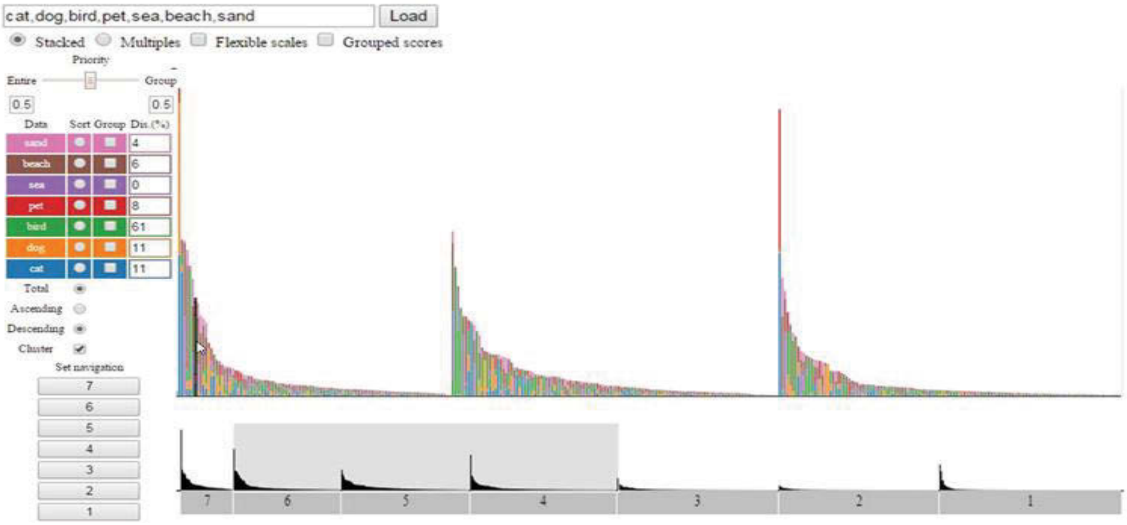
The target of the current approaches almost focuses on the visual comparison of tag values for the query refinement and the text collection mining; however, the challenge of how to use the tag values to support choosing a right item in a heterogeneous collection has not been addressed deeply. With the motivation, our study aims to deliver a visualization technique which enables to analyze, compare, and rank the items towards the groups of the tag values. These operations could help to choose a suitable alternative in a heterogeneous collection.

The current works would rather support the single-attribute or single-score combination rankings than consider the rankings based on grouped attributes or grouped scores. The grouped-score rankings are essential when the analytic criteria concentrate on sets of attributes instead of single ones. This chapter addresses such a challenge via the case of multi-tag rankings with the grouped-tag comparison and proposes an interactive visualization method for the multi-attribute ranking of the grouped-score combination.

8.3 Ranking Visualization Technique

8.3.1 Basic Design and Interaction

Qstack is an interactive visualization technique developed for the multi-tag rankings by grouped-score comparison. The technique constructs the chains of stacked bar charts with focus+context features which are synchronously displayed in a single screen for visual analysis. By employing the design, users are able to analyze both inside and across the clusters of given items concerning the groups of tags.



cat,dog,bird,pet,sea,beach,sand [Load]

● Stacked ○ Multiples □ Flexible scales □ Grouped scores

Priority: Entire 0.5 — Group 0.5

Data	Sort	Group	Dis.(%)
sand	○	□	4
beach	○	□	6
sea	○	□	0
pet	○	□	8
bird	○	□	61
dog	○	□	11
cat	○	□	11

Total ●

Ascending ○

Descending ●

Cluster ☑

Figure 8.1 The basic layout design of Qstack.

The general design of Qstack is a serial stacked bar representation controlled by a query panel (see Figure 8.1). The visual representation includes a context side at the bottom as the sub view and a focus side at the center as the main view. In the context side, the height of bar charts encodes the total score of all grouped tags as the rank of all the given items, whereas the stacked bar charts in the focus view visualize the original distribution of the tag values for single tags and grouped tags. The total height of the stacked bars encodes the grouped scores of the tags (see Figure 8.2).

For the ranking control, Qstack features a dynamic query panel and interactive bars. The query panel supports creating and adjusting the visual comparison. The operations are listed below.

- Multi-tag Load: to load the relevant items towards all considered tags,
- Stacked, Multiples, Flexible scales: to change the layouts between the stacked bars and multi-baselines with flexible scales,
- Entire, Group Priority: to assign priorities as the weights on entire tags and grouped tags,
- Grouped-scores: to switch between the grouped-score comparison and the single-score comparison,
- Data column: to list all tags and its encoding colors,
- Sort column: to sort the results in order by the single or total scores of tags,

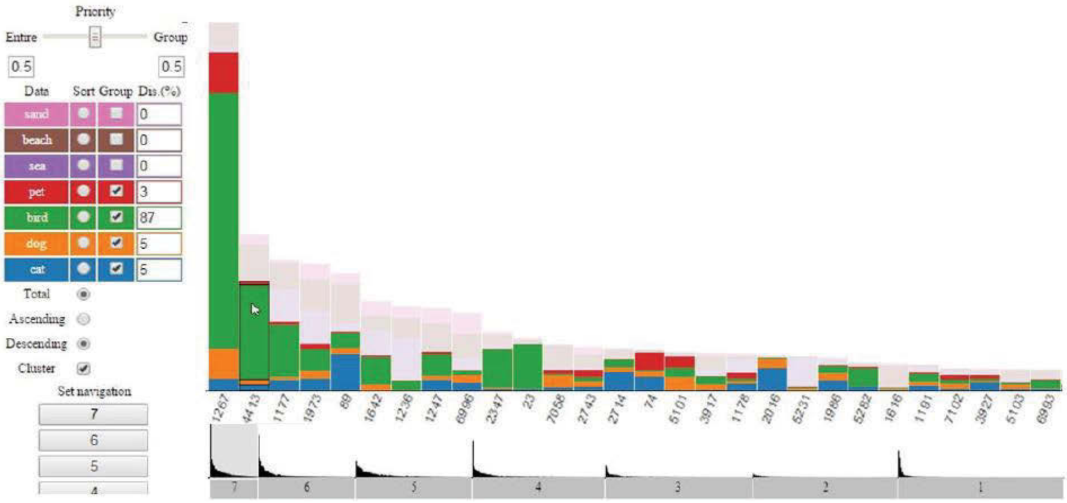
- Group column: to combine the tags for the grouped scores,
- Dis.(%): to show the distribution of the tag values on each item,
- Cluster: to cluster the results towards all groups of the tags.

Users can observe the detail of the tag values by moving a mouse pointer over the corresponding bars. The activated bars will be then highlighted in both the focus and context views for the item tracing. The identified information of the items is displayed only when the horizontal space is sufficient.

8.3.2 Grouped-Score Rankings

The strength of Qstack is the ability to compare and rank the grouped scores of tags flexibly. Users are able to combine the tags visually to a group, assign the group weights, and analyze the ranking patterns of the given items (see Figure 8.2). The ranking patterns in the focus view are the double-layer stacked bars consisting of the original scores of entire tags displayed in the bright transparent background and the grouped scores of combined tags displayed in front. The design enables users not only to keep the tracking on total rankings but also to observe the causes from the sets of tags.

Additionally, the ranking component allows users to cluster and navigate the given items quickly towards the groups of tags. All the clusters are aligned in a serial order by user demands. Therefore, the comparison of the multi-tags could be analyzed both inside and across the clusters.



Priority
Entire 0.5 Group 0.5

Data	Sort	Group	Dis.(%)
sand	●	□	0
beach	●	□	0
sea	●	□	0
pet	●	☑	3
bird	●	☑	87
dog	●	☑	5
cat	●	☑	5
Total	◉		

Ascending ○
Descending ◉
Cluster ☑

Figure 8.2 An instance of the grouped-score visual ranking.

The formal definition of the rankings is based on the grouped scores of tag sets *A* and *A'* on each item with *W* and *W'*, where $A=\{A_i|i=1,2...n\}$ is the entire considered tag values, $A'=\{A_j|j=1,2...n'\}$ is the sub set of *A* grouped by the users, and *W* and *W'* are the given weights on *A* and *A'*. It is supposed that *C* and *D* are the sets of the given items where *C* satisfies all the tag values, and *D* partially satisfies the tag values. Thus, we compute the total scores $G(C_x)$, $G'(D_y)$ for $C_x \in C$ and $D_y \in D$ by

$$\mathrm{G(Cx)} = \mathrm{W} * \sum_{\mathrm{i=1}}^{\mathrm{n}} \mathrm{S(Cx, Ai)} + \mathrm{W'} * \sum_{\mathrm{j=1}}^{\mathrm{n'}} \mathrm{S(Cx, Aj)},$$

$$\mathrm{G'(Dy)} = \mathrm{W} * \sum_{\mathrm{i=1}}^{\mathrm{m}} \mathrm{S(Dy, Ai)} + \mathrm{W'} * \sum_{\mathrm{j=1}}^{\mathrm{m'}} \mathrm{S(Dy, Aj)}$$

Equation 8.1

where *S* is the single score of a tag value on one item, $m < n$ and $m' <= n'$.

8.4 Visual Enhancement

For handling a large number of items displayed in the limited horizontal direction, Qstack applies zooming with brushing, a powerful focus+context navigation technique. There are two brushing components of Qstack including the navigation menu and the customized brushing (see Figure 8.2). The users can either click on the navigation menu to jump quickly to a zoomed cluster or directly customize the width of the brushing rectangle and drag it to a focus area. While the former method is appropriate for individual cluster access, the latter one is useful for the flexible focus on the details across the clusters.

The entire tags and grouped tags are represented in the double-layer stacked bars with the same baseline, which supports to save the vertical space of the focus view (see Figure 8.2). For dealing with a wide range number of the tag values represented in the stacked bars, Qstack provides a multi-tag filtering function. The highlighted tag bars are kept visible whereas the remaining bars are converted to a transparent layer (Figure 8.3).

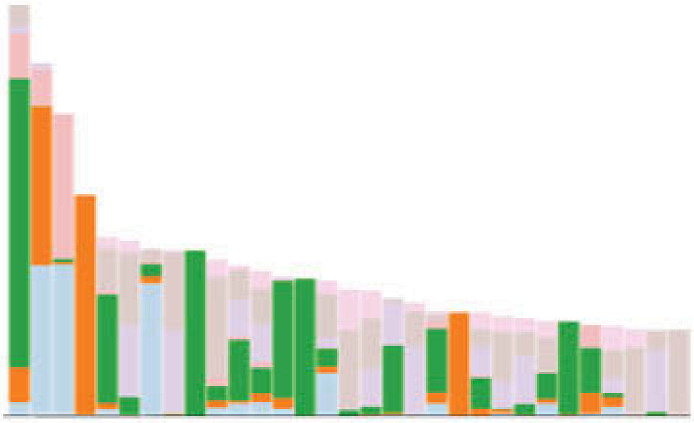


Figure 8.3 A stacked bar chart with multi-tag filtering (The green bars and orange bars are highlighted whereas the others are getting transparent).

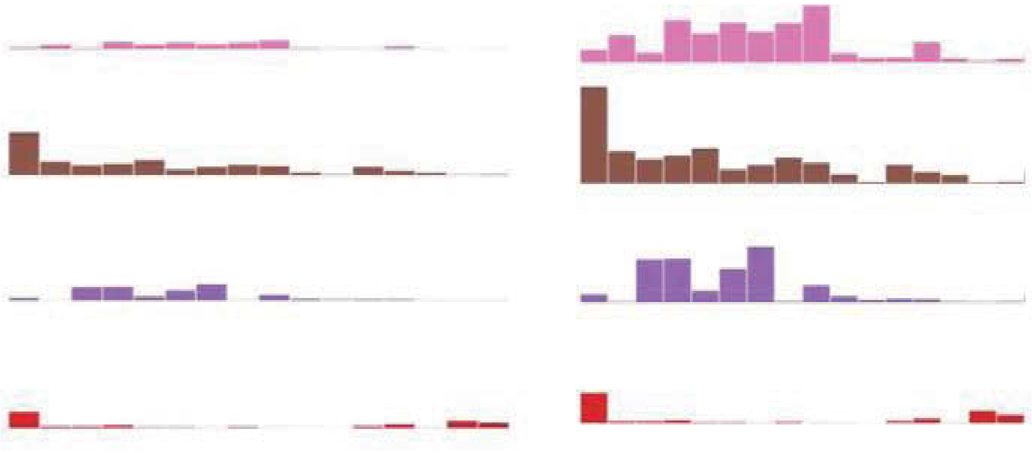


Figure 8.4.a A normal scale view. **Figure 8.4.b** A flexible scale view.

Figure 8.4 The scaling components of Qstack.

Furthermore, in order to support the multi-tag comparison across the items, the stacked bars can be transformed to a multi-baseline chart with flexible scales (see Figure 8.4). The flexible-scale function improves the vertical space usage by adjusting the height of each bar towards the highest one of the same baseline. Besides, this component is helpful for the users to recognize the missed or unapplied tag values in an item or of a cluster.

8.5 Data Mapping

Qstack extracts the used data from a database where the tag information is stored with the details of user identification (ID), tag names, and the number of tag frequencies. Once normalized from the tag frequencies, the tag values are organized in a two-dimension matrix appropriately designed for the stacked bar visualization. One of the two dimensions contains the user ID, and the other contains the single tag values, the total tag values, and the tag set indices. The tag set indices are used to categorize the given items and serve the navigation feature. The number of the tag set indices is the number of the sub sets of *n* tags defined by $p = \sum_{k=1}^{n} C(n, k)$. In order to reduce the item clutter, the navigation function is implemented according to *k* instead of p.

The data used for the instances throughout this chapter are a set of Flickr data (Plangprasopchok, Lerman & Getoor 2010). Only user IDs are shown in the Qstack prototype instead of other real information due to the anonymous characteristic of the records.

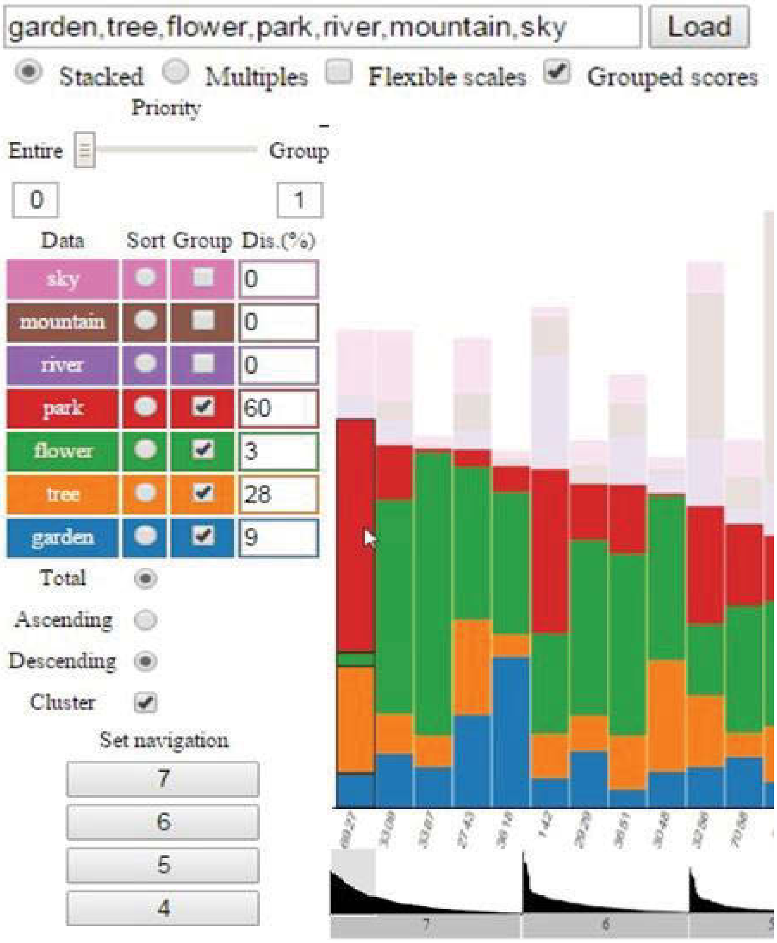
8.6 Case Study

This section demonstrates the ability of Qstack via the two use cases of Flickr photo award allocation. We assume that Emma, a president of a photo association on Flickr, would perform the annual award allocation to the members. The allocation criteria are mainly about the tag values which a member tagged in his or her collection in term of overall or specific contributions.

8.6.1 Overall Contribution Rankings

This use case demonstrates the ability of Qstack for the interactive rankings of multi-tags based on the grouped-score combination. For the overall contribution award, there are two types of prices including *Best grouped-topic contribution - A* and *Best overall contribution - B*. Whereas price *A* considers the grouped-topics with the highest priority, price *B* takes both the grouped-topics and the overall contribution into account. The eligible candidates are the members who had the contribution to all of considered tags.

Firstly, Emma enters seven considered tags of the year into the Qstack system for the data load including *garden, tree, flower, park, river, mountain,* and *sky*. There are two grouped topics: *garden, tree, flower, park (T1)* and *river, mountain, sky (T2)*. For *T1* of price *A*, once reviewing the tag value distribution, she uses the query panel to combine *garden, tree, flower,* and *park* and ranks the members towards this group with the highest priority. She enables the grouped-score calculation, clustering, and descending order to make the view concentrated on the cluster where the members satisfy all the seven tags. Emma customizes a brushing area to zoom in on the cluster and takes note three highest-ranked members *6927, 3309,* and *3387* for *T1* (see Figure 8.5). Then, she goes through the similar tasks with *river, mountain,* and *sky* for the *T2* allocation.



garden,tree,flower,park,river,mountain,sky

● Stacked ○ Multiples □ Flexible scales ☑ Grouped scores

Priority: Entire 0 – Group 1

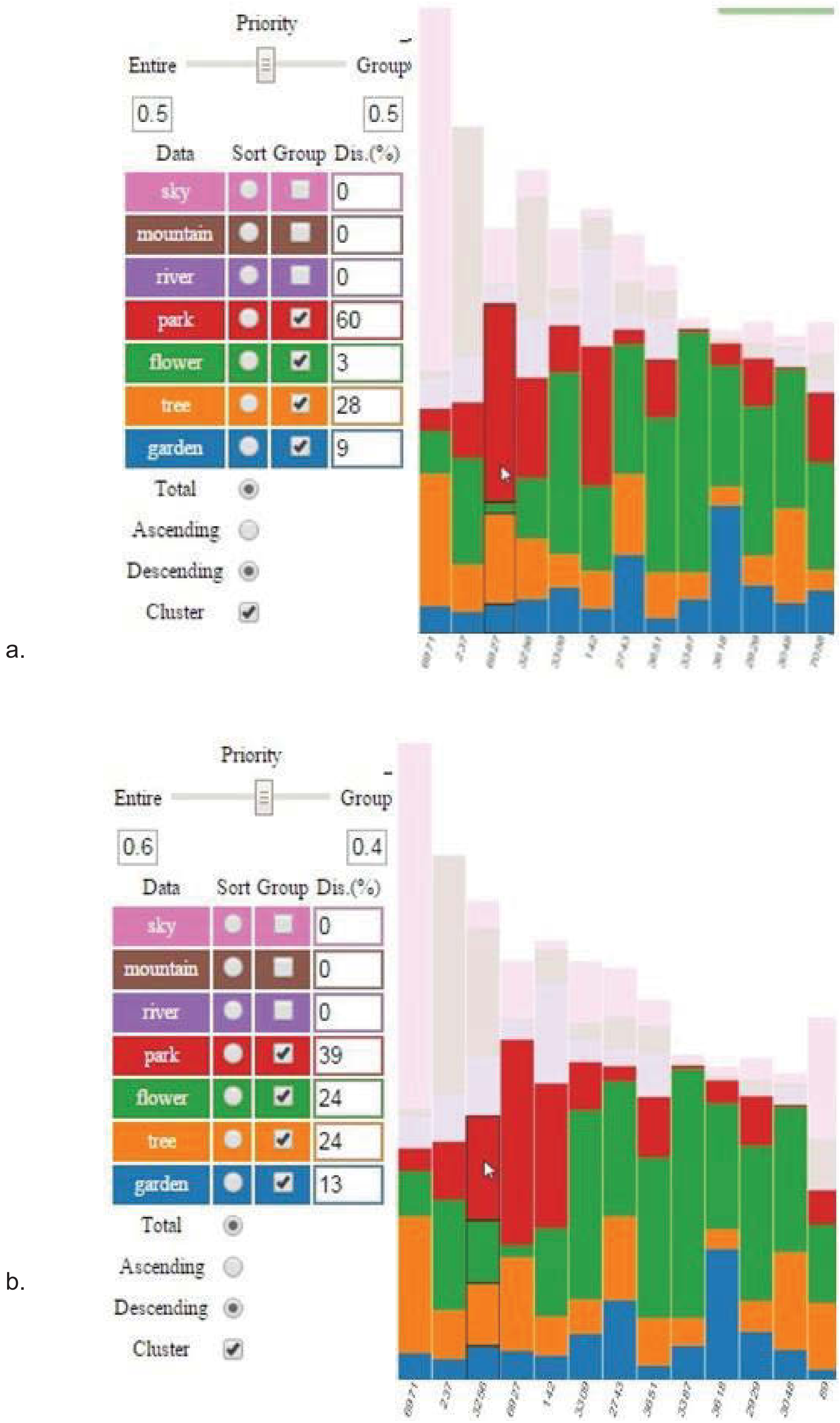
Data	Sort	Group	Dis.(%)
sky	○	□	0
mountain	○	□	0
river	○	□	0
park	○	☑	60
flower	○	☑	3
tree	○	☑	28
garden	○	☑	9
Total	●		

○ Ascending

● Descending

☑ Cluster

Figure 8.5 The overall contribution ranking for price *A*.



a.

Priority

Entire 0.5 — Group 0.5

Data	Sort	Group	Dis.(%)
sky	○	□	0
mountain	○	□	0
river	○	□	0
park	○	☑	60
flower	○	☑	3
tree	○	☑	28
garden	○	☑	9

Total ●

Ascending ○

Descending ●

Cluster ☑

b.

Priority

Entire 0.6 — Group 0.4

Data	Sort	Group	Dis.(%)
sky	○	□	0
mountain	○	□	0
river	○	□	0
park	○	☑	39
flower	○	☑	24
tree	○	☑	24
garden	○	☑	13

Total ●

Ascending ○

Descending ●

Cluster ☑

Figure 8.6 The overall contribution ranking for price *B*.

About price *B*, the final score of each member is the combination of the score on the entire contribution and a grouped topic. In the beginning, Emma makes it simple by giving the same priority (*0.5*) on the whole tags and the grouped tags for *T1* (see Figure 8.6.a). The update shows that the three highest-ranked members involve *6927* who has just received a price *A*. Emma carefully watches the visual pattern to analyze the scores, and she recognizes the score of *6927* on the entire contribution is lower than a few others. To encourage more members to access the prices, Emma adjusts a little higher priority (*0.6*) on the entire contribution than the grouped topic (*0.4*) (see Figure 8.6.b). The result, then, includes member *3256* instead of *6927*. She takes note the result and does a similar procedure for the *T2* allocation.

8.6.2 Topic Contribution Rankings

This use case demonstrates the ability of Qstack for the flexible rankings of multi-tags across categorized alternatives. The topic contribution awards are evaluated for the members who at least took four of the seven considered tags (over fifty percent) of the year, are not eligible for the overall contribution prices, and had the highest score of the tag values for single topics. In order to filter the eligible members for the price, Emma makes a brushing area cover all categories *4, 5,* and *6* (see Figure 8.8). Once the members are navigated, she uses the multi-baseline and flexible scale mode to make the view suitable for the separate topic comparison. Emma adjusts the query options to sort the result by each topic tag in descending order and selects the highest ones for the awards (see Figure 8.7).

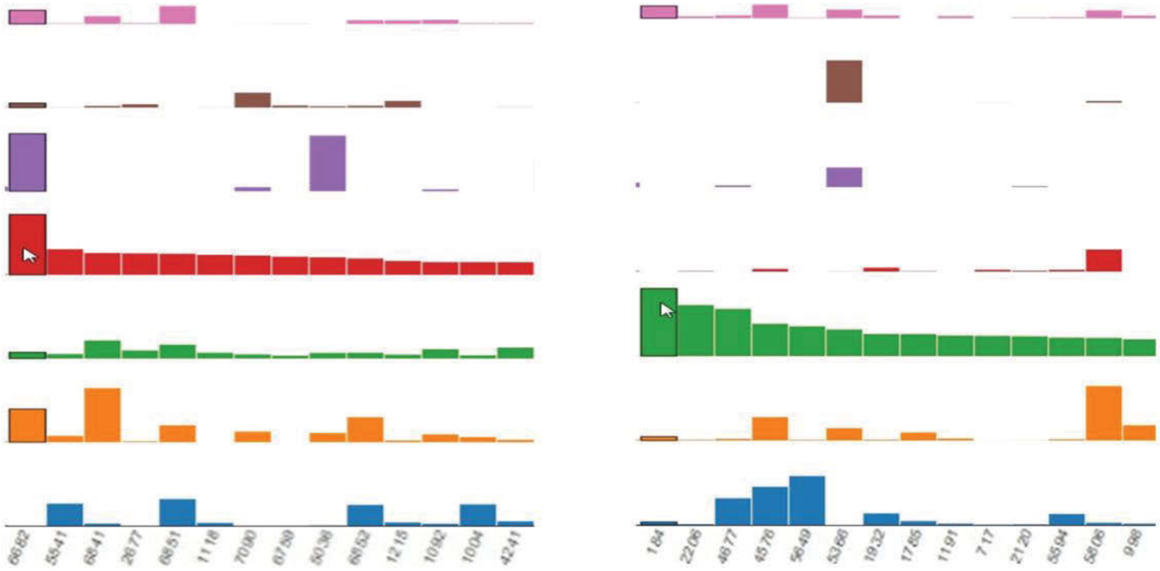


Figure 8.7 The sorting view by tag *park* (red bars) and tag *flower* (green bars) with multi-baselines and flexible scales.

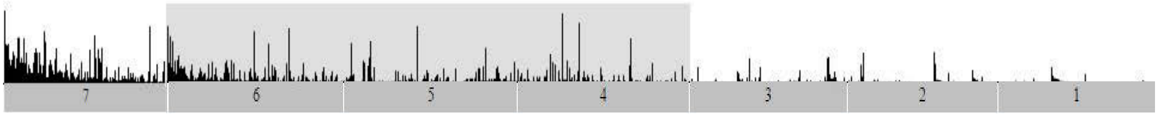


Figure 8.8 The brushing across categories *4, 5,* and *6*.

8.7 Evaluation

Our goal is to measure the effectiveness of Qstack towards the tasks of comparing and ranking the multi-tags based on the grouped-score combination within and across categorized alternatives. According to our best knowledge, currently, there is no single visualization tool with similar purposes of Qstack for multi-tag rankings; therefore, an existing-tool comparison study is impossible to be conducted. In term of generalization of Qstack approach, multi-attribute visualization tools including Excel (Microsoft 2017), Tableau (http://www.tableau.com/ 2017) and LineUp (Gratzl 2013) are taken into account. Although Excel and Tableau are powerful tools for regular data visualization on diverse purposes, to access the solution for the addressed problem, much time, the programming knowledge, and the tool usage experience are required. Those technical activities make such tools inappropriate for casual users or novices to solve such problems. LineUp, one of the current studies, is a task-driven visual technique which is the closest to the generalized purpose of Qstack. As with the target of LineUp, Qstack is designed for the novices to deal with the multi-attribute comparison by simple tasks without many technical requirements. Although LineUp can well support the attribute combination, their approach has not considered the rankings using the grouped-score combination for categorized alternatives. Due to all the reasons above, we conducted a qualitative study on Qstack usage in order to assess its usefulness instead of comparing to other techniques.

8.7.1 Study Setup and Procedure

We recruited six participants. Four of them were in computer science, and two were from engineering. Through an informal survey before the experiment, it was known they all had the Excel experience on basic tasks, but none of them had worked with Tableau before. The participants were familiar with stacked bar charts, but they were novices in the dynamic queries and the visual interaction on graphic charts. About the Qstack usage, we designed nine tasks including all primary features of the tool as three types of tasks below:

- Type 1: for data loading and the standard comparison and ranking,
- Type 2: for the multi-tag rankings using the grouped-scored combination,
- Type 3: for the multi-tag rankings across categorized alternatives.

For receiving the feedbacks from the participants, a questionnaire consisting of twelve questions was constructed and based on Likert scale system with the range of *strongly-disagree* and *strongly-agree* in the domain from *1* to *5*. Besides, there was an open question for the participants to comment benefits and drawbacks for the tool improvement. After given a Qstack introduction and tutorial, the participants freely asked and discussed to ensure they all understood the detail of the task requirements. Once completing all the tasks, they used the questionnaire to express their feedbacks.

8.7.2 Result

Generally, most of the participants successfully and easily completed all the requests, and two participants took a short interruption. The reason of the interruption was in type-3 tasks since they customized a brushing rectangle to the wrong focus areas of categorized alternatives.

The qualitative feedbacks indicated that all the participants felt confident in using the tool to solve the addressed problems. The result was quite positive with 4.7(Mean) for the ease of use, 4.5 for usefulness, and 4.2 for the interesting visualization. The open comments showed a constructive opinion for the drawback regarding the zooming. In addition to the featured horizontal zooming, the vertical zooming should be added to the stacked display mode, which might reduce the users' effort to recognize the patterns in the case of the minimal-tag-value display.

8.8 Discussion

This chapter has introduced Qstack, an interactive visualization approach for multi-tag rankings based on the grouped-score combination. The approach is to organize a series of stacked bars in categories, display multi-tags in a double layer, and use a brushing feature for the navigation on focus alternatives. This method enables users to analyze the single tag values, the grouped scores, and the total rankings efficiently within or across categories. Additionally, by performing the interaction on the dynamic query panel, users are able to drive the rankings with a minimum of technical knowledge. A case study on a photo award allocation has been presented for the Qstack demonstration, and according to the qualitative study result, Qstack is probably a useful tool for the multi-tag visual rankings.

Although the initial purpose of Qstack is for the tag data application, we believe our visualization approach might be generalized to the multi-attribute or multi-criteria rankings for general analytic data. In future, we will improve the data mapping module in order to handle various input data and provide other navigation components for enhancing Qstack benefits. Further experiments would be conducted to test the suitability of Qstack for different kinds of data.

Chapter 9 Conclusion

9.1 Summary

In summary, this dissertation has presented the models and techniques of interactive visual queries for the relational data discovery, particularly in data models and multi-dimensional data. The contributions of this dissertation are briefly condensed as follows.

Chapter 3 introduced a new relational data exploration framework with multiple visual contexts, which could be used to design visual analytic tools and help users to explore data with a minimum of effort on technical configuration. The framework focuses upon the automatic interconnection between three relative contexts including data models, multi-dimensional visualization, and pairwise analysis. We recommend equipping such interconnection with the context transmission components, which would be embedded into interactive techniques on the visual representations of data models and data dimensions. This framework is a general guide throughout this dissertation for designing and developing the techniques corresponding to the characteristics of relational-data user interfaces.

Chapter 4 presented a new interactive visual query technique, named MCquery, for data exploration based upon data models. This technique could help users to search, retrieve information, and do further investigation continuously and directly on the visualization of data models. The core of this technique is about modelling and building dynamic tuple representation and relative visual queries on node-link graphics and Entity-Relationship model. Thanks to the approach, the visualization of data models and query results are synchronized in coordinating views that could support the manipulation of the data discovery processes.

Chapter 5 presented two new interactive visual query techniques, named SumUp and FigAxis, developed for multi-dimensional analytics, particularly for the quantitative data exploration based upon parallel coordinates and scatterplots. These techniques could help users to analyze the statistical facets and examine the correlation of data based upon multiple dimensions and pairwise dimensions. The primary methods are the design of visual queries by using the brushing and zooming manipulation, which are adaptable and directly embedded on the visual interfaces of parallel coordinates and scatterplots. These approaches enable query making with flexible data support of multiple targets within and across dimensions.

Chapter 6 introduced the evaluation procedures of the proposed models through practical case studies. The case studies were mainly about how the proposed visual query techniques, including MCquery, SumUp, and FigAxis, are separately used to support and enhance individual data exploration tasks or functions and are either synchronously combined or integrated together to provide benefits for non-stop discovery and further analysis processes. The case-study outcome indicated that while independent application of the proposed visual techniques had positive effects on objective issues of visual-query direct manipulation, their integration based on the proposed framework was highly possible in improving meaningful connection between data analysis visual contexts.

Chapter 7 presented the evaluation procedures of the proposed visual query techniques through empirical objectives including space-efficient visualization, distinctive features, and the friendliness of the techniques. The major methodologies used were about performance analysis, feature comparison, and usability study. While the two formers were for measuring space-saving rates with dynamic presentation, layering display, and sharing axes on node link graphics, parallel coordinates, and scatter plots, the latter was conducted for user-feedback-based assessment of all of the techniques. The overall outcome indicated that the implementation and practical deployment of the proposed techniques in this dissertation are useful and satisfy the goals of the research.

Chapter 8 introduced an extended work from previous proposed technique FigAxis. This work was about a dynamic multiple-tag query tool. The tool was designed with double layer display and sharing axes which could help improve multiple-tag visual rankings activities.

9.2 Final Conclusion

In conclusion, this research has studied and examined the considered challenges with the fulfillment of the defined goals. The developed interactive visual query techniques and the data exploration framework can successfully support reducing the users' cognitive effort in data discovery activities. These benefits are brought to the visual data analytics and could be freely extended to be useful in any related fields of human-computer interaction, especially in the area of intelligent-user-interface design.

In future, we are taking into account the study of intellectual query development that can improve the exploration methods in big data mining in the collaborative environment. The achievement of Artificial intelligence and Internet of things would play a significant role in the study and development of the intellectual queries. Such new queries would be able to perform self-learning and automatic-adapting on data exploration activities from both systems and users. They could be available on cloud services and digital devices which are featured advanced graphic user interfaces for various using purposes, especially for query making and optimizing. Therefore, users could benefit from the intellectual query interface and smart interaction methods and do not need to worry about the technology complication of big data and network connection.

All of the techniques proposed in this dissertation would be further upgraded and extended for the other possible application areas.

List of Publications

- Pham, P.G. and Huang, M.L., 2016, February. MCquery: interactive visual query of relational data with coordinating context displays. In Proceedings of the Australasian Computer Science Week Multiconference (p. 47). ACM.

- Pham, P.G., Huang, M.L. and Nguyen, Q.V., 2016, October. SumUp: statistical visual query of multivariate data with parallel-coordinate geometry. In International Conference on Cooperative Design, Visualization and Engineering (pp. 386-393). Springer International Publishing.

- Pham, P.G., Huang, M.L. and Nguyen, Q.V., 2017. Interactive Data Exploration through Multiple Visual Contexts with Different Data Models and Dimensions. In Proceeding of 21st International Conference Information Visualisation. IEEE.

- Pham, P.G. and Huang, M.L., 2016. Qstack: Multi-tag Visual Rankings. Journal of Software, 11(7), pp.695-703.

- Pham, P.G. and Huang, M.L., 2018. Quantitative Approach on Parallel Coordinates and Scatter Plots for Multidimensional-Data Visual Analytics. Journal of Computers.

Bibliography

Abouzied, A., Hellerstein, J. and Silberschatz, A., 2012, October. DataPlay: interactive tweaking and example-driven correction of graphical database queries. In Proceedings of the 25th annual ACM symposium on User interface software and technology (pp. 207-218). ACM.

Andrews, K., Osmić, M. and Schagerl, G., 2015, October. Aggregated parallel coordinates: integrating hierarchical dimensions into parallel coordinates visualisations. In Proceedings of the 15th International Conference on Knowledge Technologies and Data-driven Business (p. 37). ACM.

Andrienko, G. and Andrienko, N., 2001, March. Constructing parallel coordinates plot for problem solving. In 1st International Symposium on Smart Graphics (pp. 9-14).

Angelaccio, M., Catarci, T. and Santucci, G., 1990. Query by diagram: A fully visual query system. Journal of Visual Languages & Computing, 1(3), pp.255-273.

Artero, A.O., de Oliveira, M.C.F. and Levkowitz, H., 2004, October. Uncovering clusters in crowded parallel coordinates visualizations. In Information Visualization, 2004. INFOVIS 2004. IEEE Symposium On (pp. 81-88). IEEE.

Auber, D., Archambault, D., Bourqui, R., Lambert, A., Mathiaut, M., Mary, P., Delest, M., Dubois, J., and Mélançon, G. 2012. The tulip 3 framework: A scalable software library for information visualization applications based on relational data. Research Report -7860, 31.

Bederson, B.B., Grosjean, J. & Meyer, J. 2004, 'Toolkit design for interactive structured graphics', IEEE Transactions on Software Engineering, vol. 30, no. 8, pp. 535-46.

Begelman, G., Keller, P. and Smadja, F., 2006, May. Automated tag clustering: Improving search and exploration in the tag space. In Collaborative Web Tagging Workshop at WWW2006, Edinburgh, Scotland (pp. 15-33).

Blau, H., 2001. A visual query language for relational knowledge discovery. Computer Science Department Faculty Publication Series, p.105.

Carpendale, S., 2008. Evaluating information visualizations. In Information Visualization (pp. 19-45). Springer Berlin Heidelberg.

Carr, D.B., Littlefield, R.J., Nicholson, W.L. and Littlefield, J.S., 1987. Scatterplot matrix techniques for large N. Journal of the American Statistical Association, 82(398), pp.424-436.

Caschera, M.C., D'Ulizia, A. and Tininini, L., 2009. Visual Query Languages, Representation Techniques, and Data Models. In Selected Readings on Database Technologies and Applications (pp. 206-233). IGI Global.

Chan, Y.H., Correa, C.D. and Ma, K.L., 2010, October. Flow-based scatterplots for sensitivity analysis. In Visual Analytics Science and Technology (VAST), 2010 IEEE Symposium on (pp. 43-50). IEEE.

Chen, P.P.S., 1976. The entity-relationship model—toward a unified view of data. ACM Transactions on Database Systems (TODS), 1(1), pp.9-36.

Clark, J. (2008). Clustered Word Clouds. http://www.neoformix.com/2008/ClusteredWordClouds.html, retrieved on 2009-01-30.

Collins, C., Viegas, F.B. and Wattenberg, M., 2009, October. Parallel tag clouds to explore and analyze faceted text corpora. In Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on (pp. 91-98). IEEE.

Cui, W., Wu, Y., Liu, S., Wei, F., Zhou, M.X. and Qu, H., 2010, March. Context preserving dynamic word cloud visualization. In Visualization Symposium (PacificVis), 2010 IEEE Pacific (pp. 121-128). IEEE.

D3.js. http://d3js.org/ 2017.

Danaparamita, J. and Gatterbauer, W., 2011, March. QueryViz: helping users understand SQL queries and their patterns. In Proceedings of the 14th International Conference on Extending Database Technology (pp. 558-561). ACM.

Dastani, M. 2002. The role of visual perception in data visualization. Journal of Visual Languages & Computing, 13, 6, 601 – 622.

Deer, J., and Perer, A. 2011. Orion: A system for modeling, transformation and visualization of multidimensional heterogeneous networks. In IEEE Conference on Visual Analytics Science and Technology (VAST), 51–60.

Di Caro, L., Candan, K.S. and Sapino, M.L., 2011. Navigating within news collections using tag-flakes. Journal of Visual Languages & Computing,22(2), pp.120-139.

d'Ocagne, M., 1885. Coordonnées parallèles & axiales: méthode de transformation géométrique et procédé nouveau de calcul graphique déduits de la considération des coordonnées parallèles. Gauthier-Villars.

Dörk, M., Carpendale, S., Collins, C. and Williamson, C., 2008. Visgets: Coordinated visualizations for web-based information exploration and discovery. IEEE Transactions on Visualization and Computer Graphics,14(6).

Dwyer, T., Marriott, K. & Stuckey, P.J. 2006, 'Fast node overlap removal', Graph Drawing, Springer, pp. 153-64.

Elmqvist, N., Dragicevic, P. and Fekete, J.D., 2008. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. IEEE transactions on Visualization and Computer Graphics, 14(6), pp.1539-1148.

Engel, K., Hadwiger, M., Kniss, J.M., Lefohn, A.E., Salama, C.R. and Weiskopf, D., 2004, August. Real-time volume graphics. In ACM Siggraph 2004 Course Notes (p. 29). ACM.

Flickr. https://www.flickr.com/ 2017.

Forsell, C., Seipel, S. and Lind, M., 2005, October. Simple 3d glyphs for spatial multivariate data. In Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on (pp. 119-124). IEEE.

Friendly, M. and Denis, D., 2005. The early origins and development of the scatterplot. Journal of the History of the Behavioral Sciences, 41(2), pp.103-130.

Frishman, Y. & Tal, A. 2008, 'Online dynamic graph drawing', IEEE Transactions on Visualization and Computer Graphics, vol. 14, no. 4, pp. 727-40.

Fua, Y.H., Ward, M.O. and Rundensteiner, E.A., 1999, October. Hierarchical parallel coordinates for exploration of large datasets. In Proceedings of the conference on Visualization'99: celebrating ten years (pp. 43-50). IEEE Computer Society Press.

Fua, Y.H., Ward, M.O. and Rundensteiner, E.A., 2000. Structure-based brushes: A mechanism for navigating hierarchically organized data and information spaces. IEEE Transactions on visualization and computer graphics, 6(2), pp.150-159.

Gratzl, S., Lex, A., Gehlenborg, N., Pfister, H. and Streit, M., 2013. Lineup: Visual analysis of multi-attribute rankings. IEEE transactions on visualization and computer graphics, 19(12), pp.2277-2286.

Hassan-Montero, Y. and Herrero-Solana, V., 2006, October. Improving tag-clouds as visual information retrieval interfaces. In International conference on multidisciplinary information sciences and technologies (pp. 25-28).

Hauser, H., Ledermann, F. and Doleisch, H., 2002. Angular brushing of extended parallel coordinates. In Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on (pp. 127-130). IEEE.

Heer, J., Card, S. K., and Landay, J. A. 2005. Prefuse: a toolkit for interactive information visualization. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 421– 430. ACM.

Heer, J., Shneiderman, B. and Park, C., 2012. A taxonomy of tools that support the fluent and flexible use of visualizations. Interactive Dynamics for Visual Analysis, 10(2), pp.1-26.

Heinrich, J. and Weiskopf, D., 2013. State of the Art of Parallel Coordinates. In Eurographics (STARs) (pp. 95-116).

Herman, I., Melancon, G. & Marshall, M.S. 2000, 'Graph visualization and navigation in information visualization: A survey', IEEE Transactions on Visualization and Computer Graphics, vol. 6, no. 1, pp. 24-43.

Ho, Q., Lundblad, P., Åström, T. and Jern, M., 2011, January. A web-enabled visualization toolkit for geovisual analytics. In IS&T/SPIE Electronic Imaging (pp. 78680R-78680R). International Society for Optics and Photonics.

Hoeber, O., Yang, X.D. and Yao, Y., 2005, September. Visualization support for interactive query refinement. In Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference on (pp. 657-665). IEEE.

Huang, M.L. & Nguyen, Q.V. 2008, 'Space-Filling interaction with Chain-Context view', IEEE Symposium on Information Visualization 2008 (InfoVis 2008)-Poster Section.

Huang, M.L., 2001, December. Information visualization of attributed relational data. In Proceedings of the 2001 Asia-Pacific symposium on Information visualization-Volume 9 (pp. 143-149). Australian Computer Society, Inc.

Huang, M.L., Eades, P. & Wang, J. 1998, 'On-line animated visualization of huge graphs using a modified spring algorithm', Journal of Visual Languages & Computing, vol. 9, no. 6, pp. 623-45.

Huang, T.H., Huang, M.L. and Zhang, K., 2012, December. An interactive scatter plot metrics visualization for decision trend analysis. In Machine Learning and Applications (ICMLA), 2012 11th International Conference on (Vol. 2, pp. 258-264). IEEE.

IBM. http://www-03.ibm.com/software/products/en/ratirosefami 2016.

Inselberg, A. and Dimsdale, B., 1991. Parallel coordinates: A tool for visualizing multivariate relations. Human-Machine Interactive Systems, pp.199-233.

Inselberg, A., 1985. The plane with parallel coordinates. The visual computer, 1(2), pp.69-91.

Kandogan, E., 2000. Star coordinates: A multi-dimensional visualization technique with uniform treatment of dimensions. In Proceedings of the IEEE Information Visualization Symposium (Vol. 650, p. 22).

Kang, H., Getoor, L., Shneiderman, B., Bilgic, M., and Licamele, L. 2008. Interactive entity resolution in relational data: A visual analytic tool and its evaluation. IEEE Transactions on Visualization and Computer Graphics, 14, 5, 999-1014.

Keim, D.A. and Kriegel, H.P., 1996. Visualization techniques for mining large databases: A comparison. IEEE Transactions on knowledge and data engineering, 8(6), pp.923-938.

Keim, D.A., 2002. Information visualization and visual data mining. IEEE transactions on Visualization and Computer Graphics, 8(1), pp.1-8.

Keim, D.A., Hao, M.C., Dayal, U., Janetzko, H. and Bak, P., 2010. Generalized scatter plots. Information Visualization, 9(4), pp.301-311.

Keim, D.A., Mansmann, F., Schneidewind, J. and Ziegler, H., 2006, July. Challenges in visual data analysis. In Information Visualization, 2006. IV 2006. Tenth International Conference on (pp. 9-16). IEEE.

Kruger, J., Schneider, J. and Westermann, R., 2006. Clearview: An interactive context preserving hotspot visualization technique. IEEE Transactions on Visualization and Computer Graphics, 12(5).

Lee, B., Riche, N.H., Karlson, A.K. and Carpendale, S., 2010. Sparkclouds: Visualizing trends in tag clouds. IEEE transactions on visualization and computer graphics, 16(6), pp.1182-1189.

Li, W., Eades, P. & Nikolov, N. 2005, 'Using spring algorithms to remove node overlapping', Proceedings of the 2005 Asia-Pacific symposium on Information visualisation, vol. 45, Australian Computer Society, Inc., pp. 131-40.

LibraryThing. https://www.librarything.com/ 2016.

Lima, D. M. d., Rodrigues, J. F., and Traina, A. J. M. 2013. Graph-based relational data visualization. In 17th International Conference on Information Visualisation (IV), 201–219. IEEE.

Little et. al. 1972 (Ware, C., 2012. Information visualization: perception for design. Cha. 1-p 2. Elsevier.)

Liu, Z., Navathe, S. B., and Stasko, J. T. 2001. Network-based visual analysis of tabular data. In IEEE Conference on Visual Analytics Science and Technology (VAST), 41–50.

Lohmann, S., Ziegler, J. and Tetzlaff, L., 2009, August. Comparison of tag cloud layouts: Task-related performance and visual exploration. In IFIP Conference on Human-Computer Interaction (pp. 392-404). Springer Berlin Heidelberg.

Lu, L.F., Huang, M.L. and Huang, T.H., 2012, December. A new axes re-ordering method in parallel coordinates visualization. In Machine Learning and Applications (ICMLA), 2012 11th International Conference On (Vol. 2, pp. 252-257). IEEE.

Majumdar, N., 2012. MATLAB Graphics and Data Visualization Cookbook. Packt Publishing Ltd.

Martin, A.R. and Ward, M.O., 1995, October. High dimensional brushing for interactive exploration of multivariate data. In Proceedings of the 6th Conference on Visualization'95 (p. 271). IEEE Computer Society.

Merzkirch, W., 1987. Techniques of flow visualization (no. agard-ag-302). advisory group for aerospace research and development neuilly-sur-seine (france).

Microsoft. http://office.microsoft.com/en-001/access/ 2017.

Misue, K., Eades, P., Lai, W. & Sugiyama, K. 1995, 'Layout adjustment and the mental map', Journal of Visual Languages and Computing, vol. 6, no. 2, pp. 183-210.

Muehlenhaus, I., 2012. Visualize This: The FlowingData Guide to Design, Visualization, and Statistics. Cartography and Geographic Information Science, 39(3), pp.170-172.

Nguyen, Q.V. & Huang, M.L. 2005, 'EncCon: An approach to constructing interactive visualization of large hierarchical data', Information Visualization, vol. 4, no. 1, pp. 1-21.

Nguyen, Q.V., Simoff, S. and Qian, Y., 2016, September. Deep Exploration of Multidimensional Data with Linkable Scatterplots. In Proceedings of the 9th International Symposium on Visual Information Communication and Interaction (pp. 43-50). ACM.

Noik, E.G. 1993, 'Layout-independent fisheye views of nested graphs', Proceedings of 1993 IEEE Symposium on Visual Languages, IEEE, pp. 336-41.

Novotny, M. and Hauser, H., 2006. Outlier-preserving focus+ context visualization in parallel coordinates. IEEE Transactions on Visualization and Computer Graphics, 12(5), pp.893-900.

Oracle. http://www.mysql.com/ 2017.

Plangprasopchok, A., Lerman, K. and Getoor, L., 2010, July. Growing a tree in the forest: Constructing folksonomies by integrating structured metadata. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 949-958). ACM.

Polyviou, S., Samaras, G. and Evripidou, P., 2005, April. A relationally complete visual query language for heterogeneous data sources and pervasive querying. In Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on (pp. 471-482). IEEE.

Purchase, H.C., Hoggan, E. & Görg, C. 2007, 'How important is the “mental map”?–An empirical investigation of a dynamic graph layout algorithm', Graph drawing, Springer, pp. 184-95.

Rao, R. and Card, S.K., 1994, April. The table lens: merging graphical and symbolic representations in an interactive focus+ context visualization for tabular information. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 318-322). ACM.

Russell, A., Smart, P.R., Braines, D. and Shadbolt, N.R., 2008. Nitelight: A graphical tool for semantic query construction.

Sedlmair, M., Munzner, T. and Tory, M., 2013. Empirical guidance on scatterplot and dimension reduction technique choices. IEEE Transactions on Visualization and Computer Graphics, 19(12), pp.2634-2643.

Seifert, I., 2011. A pool of queries: Interactive multidimensional query visualization for information seeking in digital libraries. Information Visualization, 10(2), pp.97-106.

Shneiderman, B., 1992. Tree visualization with tree-maps: 2-d space-filling approach. ACM Transactions on graphics (TOG), 11(1), pp.92-99.

Shneiderman, B., 1996, September. The eyes have it: A task by data type taxonomy for information visualizations. In Visual Languages, 1996. Proceedings., IEEE Symposium on (pp. 336-343). IEEE.

Siirtola, H. and Räihä, K.J., 2006. Interacting with parallel coordinates. Interacting with Computers, 18(6), pp.1278-1309.

Siirtola, H., 2000. Direct manipulation of parallel coordinates. In Information Visualization, 2000. Proceedings. IEEE International Conference on (pp. 373-378). IEEE.

Stolte, C., Tang, D. and Hanrahan, P., 2002. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. IEEE Transactions on Visualization and Computer Graphics, 8(1), pp.52-65.

Stolte, C., Tang, D., and Hanrahan, P. 2003. Multiscale visualization using data cubes. In IEEE Transactions on Visualization and Computer Graphics, 9, 2, 176–187

Swayne, D.F., Buja, A., Lang, D.T. and Cook, D., 1990. GGobi.

Sybase. http://www.sybase.com/products/modelingdevelopment/powerdesigner 2016.

Tabeau. http://www.tableau.com/ 2017.

Tanin, E., Shneiderman, B. and Xie, H., 2007. Browsing large online data tables using generalized query previews. Information Systems, 32(3), pp.402-423.

ter HOFSTEDE, A.H., Proper, H.A. and van der Weide, T.P., 1996. Query formulation as an information retrieval problem. The Computer Journal, 39(4), pp.255-274.

Tory, M., Sprague, D., Wu, F., So, W.Y. and Munzner, T., 2007. Spatialization design: Comparing points and landscapes. IEEE Transactions on Visualization and Computer Graphics, 13(6), pp.1262-1269.

Tory, M., Swindells, C. and Dreezer, R., 2009. Comparing dot and landscape spatializations for visual memory differences. IEEE transactions on visualization and computer graphics, 15(6).

Tsuda, K., Yoshitaka, A., Hirakawa, M., Tanaka, M. and Ichikawa, T., 1990. IconicBrowser: An iconic retrieval system for object-oriented databases. Journal of Visual Languages & Computing, 1(1), pp.59-76. Tulip.

Tu, Y. 2010, 'Multi-con: Exploring graphs by fast switching among multiple contexts', Proceedings of the International Conference on Advanced Visual Interfaces, ACM, pp. 259-66.

Tu, Y. and Shen, H.W., 2013, February. GraphCharter: Combining browsing with query to explore large semantic graphs. In Visualization Symposium (PacificVis), 2013 IEEE Pacific (pp. 49-56). IEEE.

Van Ham, F. and Perer, A., 2009. “Search, show context, expand on demand”: supporting large graph exploration with degree-of-interest. IEEE Transactions on Visualization and Computer Graphics, 15(6).

Wacholder, N., 2011. Interactive query formulation. Annual review of information science and technology, 45(1), pp.157-196.

Ware, C., 2005. Visual queries: The foundation of visual thinking. In Knowledge and information visualization (pp. 27-35). Springer Berlin Heidelberg.

Ware, C., 2012. Information visualization: perception for design. Elsevier.

Yang, J., Ward, M.O. and Rundensteiner, E.A., 2002. Visual hierarchical dimension reduction for exploration of high dimensional datasets.

Zubiaga, A., García-Plaza, A.P., Fresno, V. and Martínez, R., 2009, July. Content-based clustering for tag cloud visualization. In Social Network Analysis and Mining, 2009. ASONAM'09. International Conference on Advances in (pp. 316-319). IEEE.