Visualisation of Large-Scale Call-Centre Data

Dylan Geraint Rees
Department of Computer Science

Submitted to Swansea University in fulfilment
of the requirements for the degree of Doctor of Philosophy

2020

Abstract

The contact centre industry employs 4% of the entire United Kingdom and United States' working population and generates gigabytes of operational data that require analysis, to provide insight and to improve efficiency. This thesis is the result of a collaboration with QPC Limited who provide data collection and analysis products for call centres. They provided a large data-set featuring almost 5 million calls to be analysed. This thesis utilises novel visualisation techniques to create tools for the exploration of the large, complex call centre data-set and to facilitate unique observations into the data.

A survey of information visualisation books is presented, providing a thorough background of the field. Following this, a feature-rich application that visualises large call centre data sets using scatterplots that support millions of points is presented. The application utilises both the CPU and GPU acceleration for processing and filtering and is exhibited with millions of call events.

This is expanded upon with the use of glyphs to depict agent behaviour in a call centre. A technique is developed to cluster overlapping glyphs into a single parent glyph dependant on zoom level and a customizable distance metric. This hierarchical glyph represents the mean value of all child agent glyphs, removing overlap and reducing visual clutter. A novel technique for visualising individually tailored glyphs using a Graphics Processing Unit is also presented, and demonstrated rendering over 100,000 glyphs at interactive frame rates. An open-source code example is provided for reproducibility.

Finally, a novel interaction and layout method is introduced for improving the scalability of chord diagrams to visualise call transfers. An exploration of sketch-based methods for showing multiple links

and direction is made, and a sketch-based brushing technique for filtering is proposed. Feedback from domain experts in the call centre industry is reported for all applications developed.

Declaration

This work has not been previously accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signature: ████████

Date: 27th of March 2020

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signature: .████████

Date: 27th of March 2020

I hereby give my consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signature: ████████

Date: 27th of March 2020

Contents

Acknowledgements

First and foremost I would like to thank my supervisor Dr Robert S. Laramee for his support and guidance throughout my PhD and my Masters projects before that. We had many interesting discussions over this period covering a large array of topics. I would also like to thank my collaborators at QPC Ltd; Gary Smith, Tony D'Cruze, and Paul Brookes, for their advice, guidance, and for conceiving the project. Additionally I thank QPC Ltd. for providing funding towards the project.

I gratefully acknowledge funding from KESS. Knowledge Economy Skills Scholarships (KESS) is a pan-Wales higher level skills initiative led by Bangor University on behalf of the HE sector in Wales. It is part-funded by the Welsh Government's European Social Fund (ESF) convergence programme for West Wales and the Valleys. Additionally I would like to thank the KESS 2 personnel for their support throughout the duration of my research.

I thank my mother for her support and for proof-reading my work, diolch mam. Thanks also goes to my father for his support. Finally I would like to thank the numerous friends and family for their support, assistance, and needed distractions.

UNDEB EWROPEAIDD
EUROPEAN UNION

Llywodraeth Cymru
Welsh Government

**Cronfa Gymdeithasol Ewrop
European Social Fund**

Ysgoloriaethau Sgiliau Economi Gwybodaeth
Knowledge Economy Skills Scholarships

Contributions

This thesis is based on the following published papers:

1. REES D., LARAMEE R. S.: A survey of information visualization books. *Computer Graphics Forum* 38, 1 (2019), 610–646 [RL19]
   Online Resource: `http://visbooks.swansea.ac.uk/`

2. REES D., ROBERTS R. C., LARAMEE R. S., BROOKES P., D'CRUZE T., SMITH G. A.: GPU-assisted scatterplots for millions of call events. In *Computer Graphics and Visual Computing (CGVC)* (2018), , Tam G. K. L.,Vidal F., (Eds.), The Eurographics Association. [RRL*18a]
   Supplementary Video: `https://vimeo.com/270333276`

3. REES D., ROBERTS R. C., LARAMEE R. S., BROOKES P., D'CRUZE T., SMITH G. A.: Feature-rich, GPU-assisted scatterplots for millions of call events. *Computers* 8, 1 (2019), 1–21 [RRL*19]
   Supplementary Video: `https://vimeo.com/305933032, https://www.youtube.com/watch?v=`
   `AlQOWoOCLwc`

4. REES D., LARAMEE R. S., BROOKES P., D'CRUZE T., SMITH G. A.: AgentVis: Visual Analysis of Agent Behavior with Hierarchical Glyphs *IEEE Transactions on Visualization and Computer Graphics*, (2020), [RLB*20]
   Supplementary Video: `https://vimeo.com/325757055, https://www.youtube.com/watch?v=`
   `hsD21tHybts`

5. REES D., LARAMEE R. S., BROOKES P., D'CRUZE T.: Interaction Techniques for Chord Diagrams. Accepted at *The 24th International Conference on Information Visualization (IV)*, (2020) [RLBD20]
   Supplementary Video: `https://vimeo.com/445249684`

6. REES D., LARAMEE R. S.: An Open-Source, Hardware Accelerated Industry Prototype for Multi-dimensional Glyphs. *The Computer Graphics, Visualization, Computer Vision and Image Processing (CGVCVIP)*, (2020); [RL20]
   Open-Source Code Repository: `https://github.com/dylan-rees/glyphRenderer`

7. REES D., WANG Q., LARAMEE R. S.: The Industry Engagement Ladder. *Journal of Industry - University Collaboration*, (2020) [RWL20]

Chapter 1.

Introduction

"Vision is the art of seeing what is invisible to others."
    — Jonathan Swift – Essayist, Satirist, Poet, Cleric (1667–1745)

1.1. Data Visualisation

Data visualisation is the representation of data in a graphical manner. This entails creating images out of data enabling insights to be made by leveraging the cognitive ability of the vision system. This facilitates the transfer from information to knowledge [CEH*08].

Data visualisation encompasses a number of disciplines as indicated by Figure 1.1.

Modern data visualisation is a relatively new field, however, humans have been making graphical representations for millennia with early cave paintings dating back to the Paleolithic eras. More recently, early pioneers of data visualisation include William Playfair who developed many of the chart designs that are still in use today such as the line chart. See Figure 1.2. Playfair used these charts to communicate statistical economic data. Other notable early pioneers are John Snow who used visualisation to trace the source of a cholera outbreak (Figure 1.3), Florence Nightingale who visualised the causes of deaths in the Army (Figure 1.4), and Charles Minard who visualised Napoleon's attempted invasion of Russia (Figure 1.5). Numerous
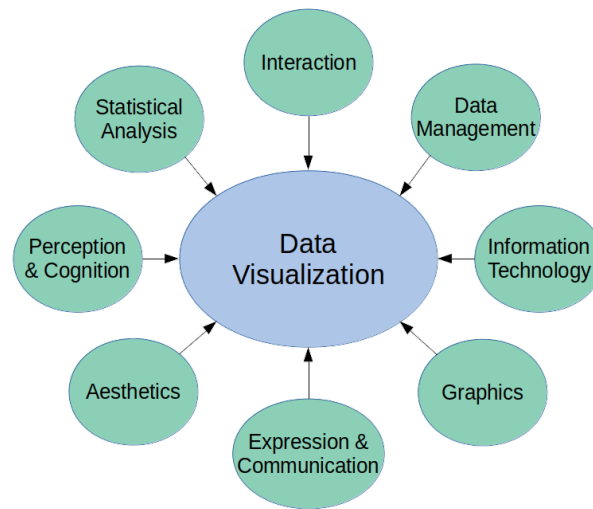
Figure 1.1.: The disciplines of data visualisation. Figure based on Chou [Cho19].

books feature a detailed evaluation of the history of data visualisation, as indicated in Chapter 2 and by Rees and Laramee [RL19].
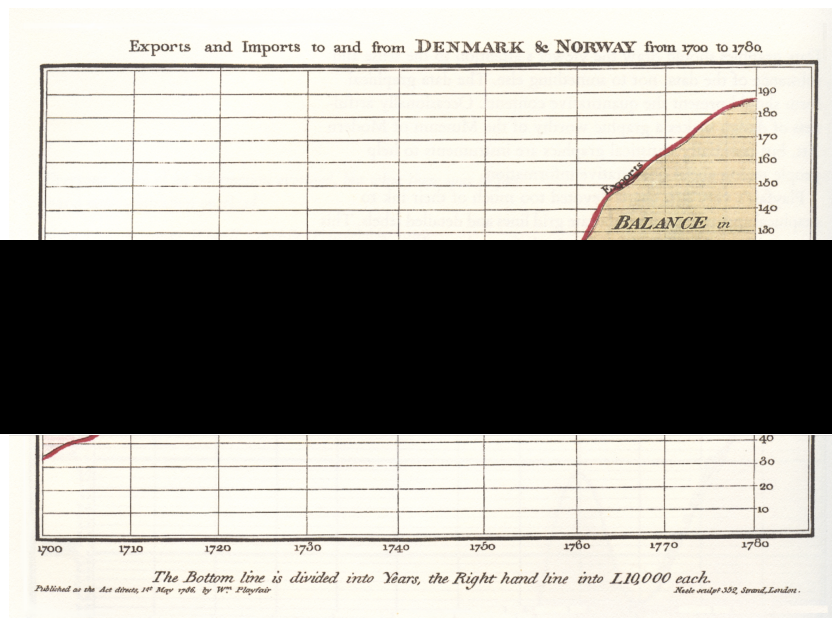
Figure 1.2.: William Playfair's line chart from 1786. [Pla86].

In the last few decades, data visualisation has been revolutionised by the use of computers. The volumes of data that can be processed dramatically increase with the aid of computers and easily rendered. With real-time rendering, interaction with data is possible greatly increasing the possibilities of data exploration.
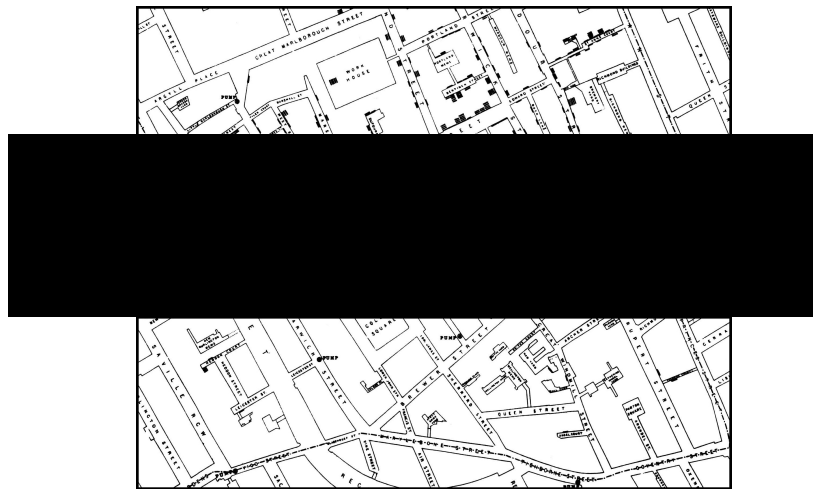
Figure 1.3.: The locations and number of cholera deaths during an outbreak in 1854 as presented by Snow [Sno55].

the centre as the common vertex.
The blue wedges measured from the centre of the circle represent area
for area the deaths from Preventible or Mitigable Zymotic diseases, the
red wedges measured from the centre the deaths from wounds, & the
black wedges measured from the centre the deaths from all other causes.
The black line across the red triangle in Nov.ʳ 1854 marks the boundary
of the deaths from all other causes during the month.
In October 1854, & April 1855, the black area coincides with the red,
in January & February 1856, the blue coincides with the black.
The entire areas may be compared by following the blue, the red & the
black lines enclosing them.

Figure 1.4.: A polar area diagram by Florence Nightingale depicting the causes of mortality in the Army [Nig58].

There are four sub-branches to data visualisation, scientific visualisation, geographical visualisation, information visualisation, and visual analytics.

Scientific Visualisation   Scientific visualisation is the visualisation of scientific phenomena. This data usually has an inherent geometry [FD01]. Examples of the use of scientific visualisation include medical imaging and flow visualisation. Medical imaging usually entails the visualisation of biological data such as MRI scans, allowing densities to mapped to a digital representation of a body, both explicit and implicit
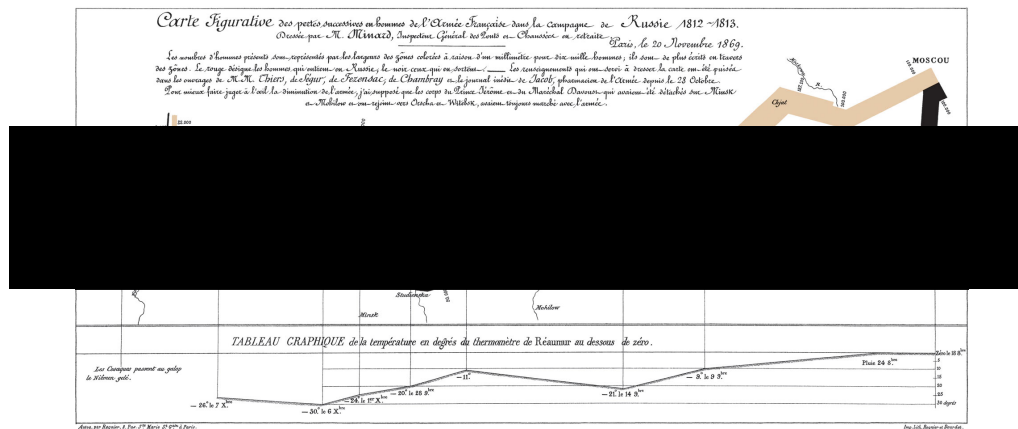
Figure 1.5.: A chart depicting the number of soldiers and movements of Napoleon's army in 1812 and the corresponding temperature by Minard [Min69].

geometries. Similarly the use of Computational Fluid Dynamics (CFD) for simulating flow in engineering has become ubiquitous. The visualisation of the resulting data also has explicit geometry as can be seen in Figure 1.6.
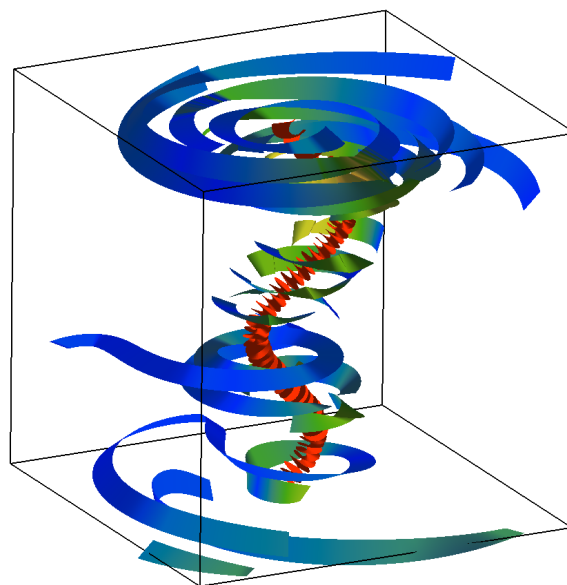
Figure 1.6.: Automatically seeded stream ribbons in a tornado flow simulation, an example scientific visualisation [RLN*17].

Geospatial Visualisation   Geospatial visualisation is the presentation of geographical data [DMK05]. This branch of data visualisation sits between the scientific and information as data usually has an explicit geographical mapping, however the data

for each geographical region does not usually have explicit geometry. Examples of geospatial data are the works of Tong *et al.* [TRL*17, TML*17, TML18].

Information Visualisation   Information visualisation is the display of abstract data that has no inherent geometry. The data can be numerical such as statistical or financial data, or non-numerical such as text data. Munzer describes the difference between scientific and information visualisation: *"It's infovis [information visualisation] when the spatial representation is chosen, and it's scivis [scientific visualisation] when the spatial representation is given"* [Mun08]. Common techniques in information visualisation are the use of scatterplots, bar charts, line charts, parallel coordinates or treemaps.

Visual Analytics   The final branch of data visualisation is visual analytics. This is the use of visualisation and interaction for analytical reasoning. Visual analytics is described by Thomas and Cook as: *"the science of analytical reasoning facilitated by interactive visual interfaces"* [CT05]. This also entails the use of the other branches of data visualisation. Interaction and interface design are key to allowing analysts to visually explore data sets to obtain insight and derive knowledge.

1.1.1. Call Centre Data & Operation

This project is preformed in collaboration with QPC Limited. QPC Limited provide software and analysis solutions for call centres, collecting and storing vast amounts of call data in the process. Data collected is valuable to both QPC Limited and their customers as it provides a means to analyse call-centre operation and customer behaviour. However, a number of metrics from each call is collected creating vasts amount of data that becomes difficult to analyse. As such, this collaboration was formed to help tackle this problem with the project aim of utilising visualisation to derive insight into call-centre behaviour by creating novel visual analytics techniques and tools.

1.1.2. Call Centre Data Characteristics

The data provided for analysis in this project is provided by our partner company QPC Limited. It consists of all calls to one of their client's call centres during February

2015. The data set is split into four separate CSV files, each file consisting of different attributes linked by a common *'Connection Identifier'* to link the individual calls. All calls have been anonymised. In total there are 4,940,292 calls collected from 43 different sites across Egypt, India, Romania, South Africa, and the UK. These calls are handled by over 6,500 unique call-centre agents.

These agents work in 10 call centre business segments as follows:

- Tech Support
- Sales
- Billing
- Company
- Customer Services

- Credit
- Retention
- Collections
- Unknown
- Other

Each call has over 70 attributes, some are recorded directly such as the call duration, whilst others are derived, such as the cost of each call to the call centre. Other attributes are used to identify the customer, the agent(s) involved to and the site where the agent is based.

Each call is initially received by an interactive voice response system (IVR). This is an automated menu system that plays a prerecorded message and directs the call guided by the telepad or voice input from the caller. Following this, calls are usually placed in a queue before being served by an agent, although not all calls are handled by an agent. From here the next state for each call can be another hold period, a transfer to another agent, or call termination.

Some typical call metrics are the duration of each call state, and the number of holds. Hold events have been separated into the reason for the hold. Of interest, in particular, is the cold and warm transfer holds. Cold transfer holds indicate when customers are placed on hold for a transfer to another agent, and information about the call is not transferred between agents. In contrast, warm transfer holds are for customers put on hold for agent transfers where information about the call is transferred.

Of particular interest to the experts is the NPS or Net Promoter Score. This is a feedback score provided by customers who contact the call centres and complete a survey. Customers are asked to rate their experience of their call centre contact on a scale from 0 to 10, with 0 being negative and 10 being positive. Few calls receive an

NPS score, approximately 1% of calls, however it represents a ground truth measure of customer satisfaction (actually reported by customers in surveys) and can be used as an indicator for operations improvement. Those which do not provide a feedback score are assigned a -1 score.

Customer Effort Score (CES) is another attribute of interest. This variable provides a derived value quantifying the effort made by the customer in contacting the call centre. For example, if a customer is kept waiting for a long time and passed through multiple agents, this would incur a large CES, whereas a customer on a short call, speaking to a single agent would acquire a low CES. The CES is a multifaceted metric derived from a number of variables.

1.2. Challenges

The collaboration with QPC Limited was initiated as a result of them recognising a gap in their knowledge and understanding of the latest developments within data visualisation and a need to analyse a vast amount of data. This project is a continuation of a collaboration with QPC Limited which has previously resulted in an Innovate UK project [Inn15] and a PhD research project [Rob19]. The challenges associated with this project are as follows:

Literature explosion: There has been an explosion in the amount of literature pertaining to information visualisation with more and more books being published on the subject with each passing year. As such it is difficult to appreciate what new or relevant information is presented in these publications. In academia, survey papers provide a means of an overview of the state-of-the-art in the subject, however, other sources of knowledge i.e. books do not have any such mechanism. This may lead to a loss of applicable knowledge being buried under the volume of literature.

Visualisation of large amounts of data (scale): The data-set provided by QPC Limited for analysis consists of a single months worth of data which contains almost five million telephone calls. To analyse a data-set of this magnitude with traditional data visualisation tools such as Microsoft Excel or Tableau is not possible due to a cap on the number of data instances that can be imported. Therefore to be able to explore the dataset requires other approaches.

Visualisation of multidimensional data at scale: Not only are there a large number of telephone calls to analyse, each call has a number of attributes to explore. Therefore to allow exploration of multiple attributes at a single time, with a large volume of data is a significant challenge.

Visualisation of large numbers of connections: The visualisation of a large number of connections between a large number of nodes is another issue of scale. Calls recorded within the data-set provided are often not solely dealt with by a single call centre agent. Calls are often transferred between agents in differing departments. With a large number of calls and numerous call centre agents, visualising a large number of connections, with the ability to focus in on individual transfers is another challenge.

The academic–industry gap: The research presented in this thesis is made in collaboration with an industry partner, QPC Limited. Academic–industry research has shown to have a higher impact [Shn18], however challenges exist in aligning possibly diverging needs and expected outcomes [RLB*18b].

1.3. Methodology

To address the challenges outlined, first a survey is presented of books published on the topics of information visualisation and visual analytics. This requires collecting a number of books on the subject, assessing their suitability, and gathering the knowledge from each source. This provides a thorough overview of the topic from a source that can be overlooked in academia, as well as provide knowledge of the fundamentals, and contribute methodologies for addressing other challenges.

Once these have been established, the scalability challenge can be addressed with initial visualisations. Important aspects too address are the ability to explore and filter the large volumes of data. Scatter plots provide a suitable means for exploring these aspects as it is a relative simple method that is well understood. A prototype application will be developed to enable the visualisation of the data-set. Following on from this, an expansion of the scalable solution will be made with methods for the simultaneous visualisation of multivariate data.

The challenge of visualising multiple connections between call centre agents are addressed with an application for the exploration of multi-variate connections. Finally,

reflections on the partnership between academic and industrial parties are made, describing a case study on academic-industry collaboration.

Three domain experts from QPC Ltd. were consulted for this project, each focusing on different aspects of the call centre industry (programmer, researcher, analyst). The first expert is a lead software developer who has 30 years of expertise in the call centre industry. Expert two is a consultant with experience in a variety of roles within the industry, totalling over 20 years. The third expert has 15 years of call centre experience and is a director of product and marketing. He is also an Associate of the Operational Research Society. These three experts are employed at QPC Ltd. Further details of the collaborators are available in Appendix C.

Applications created for this thesis were developed by first analysing the requirements form the domain experts. This was done in a series of formal meetings, recorded for later analysis. Through these discussions requirements of the analysts were established and validated in subsequent meetings. During discussions, visual designs were constructed using a whiteboard and the advantages and disadvantages of each design were discussed before a design was agreed to be implemented. Input was gathered from all three domain experts. Once a design principle had been established, the design was implemented. Periodic meetings were held to make sure that the implementations matched the expectations of all three domain experts.

Once the implementation had been complete, feedback was gathered over a number of guided interview sessions. Interviews were semi-structured according to guidance provided by Hogan et al [HHH16]. Additional feedback was also gathered from a front-line call-centre agent for the work presented in Chapter 4.

1.4. Contributions

The contribution of this thesis are as follows:

Literature survey: This thesis presents the first overview of the books on the topic of information visualisation and visual analytics in Chapter 2 [RL19]. A novel two-level categorisation of both books and the topics covered within each book is provided, enabling for a reader to quickly find topics of interest within an array of books. Meta-data from each book is presented to help the reader compare and navigate this vast

and information-rich literature space. Recommendations for choosing a suitable visualisation book are also provided.

Visualisation of large amounts of data (scale): A novel interaction scatterplot application is developed that visualises 5,000,000 calls with the ability to track customers over multiple calls. This is presented in Chapter 3 based upon a publication [RRL*18a] and an extension [RRL*19]. Advanced interactive and hardware accelerated filtering of call and customer parameters are provided and feedback from partner domain experts in the call centre industry is reported.

Visualisation of multidimensional data at scale: An application is presented that explores and renders thousands of multi-variate glyphs representing real-world commercial call centre data guided by a series of feedback sessions with industry. This is presented in Chapter 4 and a journal publication [RLB*20]. Included is a novel hierarchical glyph aggregation and placement algorithm that alleviates occlusion and complexity for thousands of multi-variate glyphs. A novel implementation for calculating and rendering large numbers of multidimensional data-glyphs is provided that supports easy and fast selection of different glyph designs. A self-contained open-source code implementation to demonstrate the technique is provided (Appendix A, [RL20]), and a case study and feedback from domain experts is reported.

Visualisation of large number of connections: Novel interaction techniques for chord diagrams are introduced in Chapter 5 based on a conference paper [RLBD20]. A deformed layout technique to support detailed views of a particular segment is presented, along with support for joining multiple edge connections in a chord diagram. An interactive sketch-based filtering method for tracking connections across multiple chord segments is also introduced.

Academic–industry bridge: A report on the collaboration with industry, comparing the experience with previously reported pitfalls [SMM12] and reporting new pitfalls experienced, is given in Chapter 4, based on a journal publication [RLB*20].Collaboration opportunities are described and ranked by risk in Appendix B, based on a journal publication [RWL20]. The benefits of each collaboration mechanism are presented, and examples provided.

1.5. Structure

The remainder of this thesis is structured as follows. Chapter 2 provides a survey of information visualisation books. The chapter outlines the challenges associated with the survey, provides a novel classification enabling the reader to quickly find topics of interest, and provides a brief summary of each book. Recommendations for suitable books are also made.

Chapter 3 presents an application for the visualisation of almost five million calls and features methods of hardware accelerated filtering. Chapter 4 builds on the scatter plot application with the addition of multidimensional glyphs for visualising agent based data. This is augmented with a method of clustering glyphs according to screen-space to address the challenge of overplotting.

The visualisation of call connections between agents in different departments is presented in Chapter 5. This chapter features novel interaction methods featuring an overview of all connections as well as connections made by individual agents. Conclusions are drawn in Chapter 6 and future work presented.

Chapter 2.

A Survey of Information Visualisation Books

"The book you don't read won't help."
    — Jim Rohn – Entrepreneur, Author (1930–2009)

"You cannot open a book without learning something."
    — Confucius - Teacher, Politician, Philosopher (551–479 BC)

This chapter is the result of the survey, a version of which is published in *Computer Graphics Forum* [RL19]. The contents of this chapter should be considered as a standalone research chapter rather than providing a background to the research topics discussed within this thesis. The following individual chapters each provide their own background research. This chapter does, however, provide a strong foundation of information visualisation knowledge with many topics encountered beyond a traditional, focused research strategy. Since publication, praise has been received for the survey from the community; "Thanks for the great STAR on textbooks – it is a fantastic resource for the community!" – Miriah Meyer - University of Utah, June 24th 2019.

When choosing a survey topic for this project, one special challenge was encountered with many suitable subjects being covered by recent surveys. The most obvious choice for a survey theme would be business visualisation, however this topic has already been published by Roberts and Laramee [RL18]. A similar candidate topic is

finance data visualisation, again already published [KCA*16]. Another potential survey subject would be glyph based visualisations to complement the work presented in Chapter 4, however, in this case, three surveys already exist [War02, BKC*13, FIBK17]. The existence of a survey of surveys (SoS) by McNabb and Laramee exemplifies the challenge [ML17].

A void was identified in the field for a survey on data visualisation books. A recent explosion in the number of books published dictated that the survey scope should be restricted to only books within the information visualisation and visual analytics domains. Because of availability of many books in my co-author's library and due to the funding provided by KESS 2 and the ESF, books to be reviewed were available or could be afforded, enabling the survey to be made. This survey provides a strong foundation of knowledge in the information visualisation and visual analytics fields, with a broad array of topics addressed, including visualisation fundamentals [Tuf83], perception [War12], and business visualisation [Sim14].

2.1. Introduction and Motivation

Books provide a very valuable resource for education, research and to further understanding in any field. Books cater to many different levels of information visualisers, from beginners to specialists, however, knowing what books are relevant for which topics and competency levels can be very challenging due to the rapidly expanding landscape of information visualisation books.

One of the first books to discuss the use of information visualisation is Jacques Bertin's Sémiologie Graphique [BBS67] published in 1967. Few books were published in the following 40 years, however in the last decade there has been a dramatic increase of published books, as evident in Figure 2.1. In fact there are so many, we cannot survey all of them. They are being published so rapidly that it's very challenging to keep abreast of the volume (similar to research papers). We focus on what we believe is the most important core subset. See section 2.1.3 for more details on the scope. Keeping up with the book volume is a difficult task both in time and money, with over 23,000 pages represented in this survey.

Books are predominantly overlooked in traditional literature surveys, leading to missed opportunities to pass on valuable knowledge which, as scientists, is one

Visualisation books published each year
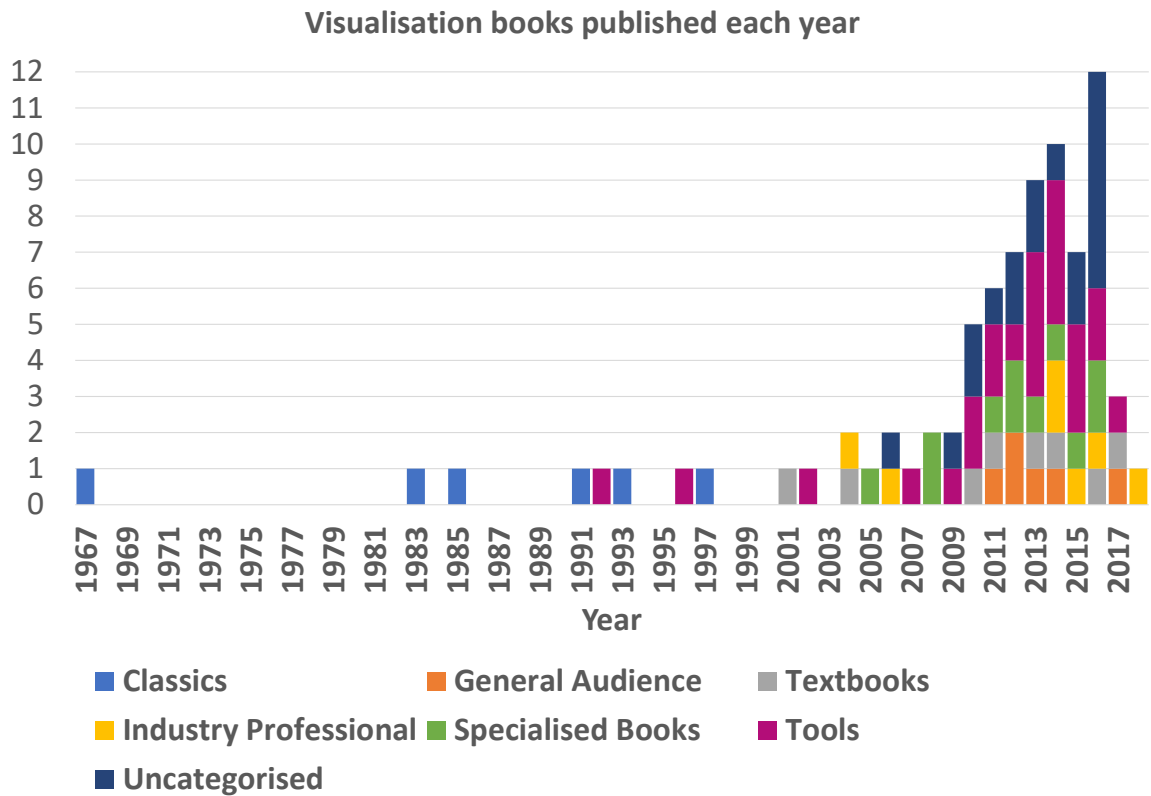
Figure 2.1.: Number of identified information visualisation and visual analytics book publications by year.

of our primary goals. For example, Eduard Imhof published colour use guidelines for geo-spatial visualisation which are rarely (if ever) used by data visualisation researchers [Imh82]. This is unfortunate because the authors of these books invest several years in their writing efforts. As an example, Inselberg's book on Parallel Coordinates [Ins09], required five years to write (based on personal communication) and consists of 554 pages and over 230 colour figures. Inselberg's book contains much mathematical formulation on parallel coordinates not featured elsewhere in the information visualisation conferences and journal papers. Another example is Samet's book on Foundations of Multidimensional and Metric Data Structures [Sam06] which required over ten years of dedication to complete (based on personal communication). Ten years is not unusual when we consider multiple editions of a book.

Books offer some advantages over research papers in that they feature a longer format, therefore enabling greater detail to be provided. Greater use of figures, along with the use of more descriptive case studies, can also be used to aid understanding of a
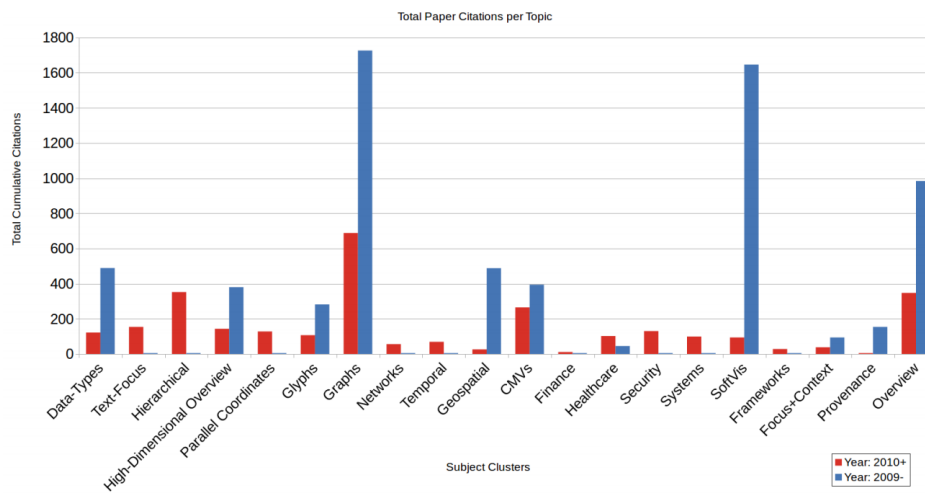
Figure 2.2.: Citations in survey papers, ordered by topic, before and after 2010, courtesy of McNabb and Laramee [ML17], highlighting that the majority of research paper citations predate 2010.

subject. Books also offer advantages to beginners and non-specialists, the large volume of published literature, with the looser length restrictions, enabling simplifications and background to facilitate understanding. Research papers, in contrast, often assume advanced understanding. Books also tend to be more accessible to people outside the subject who are looking for a starting point or overview of visualisation.

Books, however, are at a disadvantage from research papers in that they do not necessarily represent the state of the art within the subject due to the publication time required. However, even this is mitigated with the advent of e-books and short books such as the SpringerBriefs series in computer science. Analysis performed by McNabb and Laramee [ML17] shows that the majority of citations in survey papers are a number of years old already. See Figure 2.2. Books also do not undergo a rigorous peer review process, however, they are scrutinised by customer reviews.

This non-traditional literature survey aims to provide a novel overview of the valuable landscape of the most influential information visualisation and visual analytics books and to categorise the common topics within the book chapters. The contributions of this unique survey are to provide:

- the first overview of the books on the topic of information visualisation and visual analytics,

- a novel two-level categorisation of both books and the topics covered within each book,

- valuable meta-data from each book to help the reader compare and navigate this vast and information-rich literature space,

- recommendations for choosing a suitable visualisation book.

We believe that a literature survey of this type is very useful for readers both inside and outside the visualisation community.

2.1.1. Survey Challenges

There are a number of challenges involved in performing a literary survey of this nature.

Search: Although online book stores, such as Amazon.com [Ama18], enable fast searching of millions of books, sometimes appropriate books can be lost in irrelevant material when performing a generic search. For example, searching for the term *"visualization"* on Amazon.com returns 16 books on the first page, with only five dedicated to data visualisation. Whereas a visualisation research paper in the domain has a limited number of sources, conferences and journals, that can be searched [IHK*17], there are a myriad of publishers capable of publishing appropriate books. Once a book has been identified it can also be difficult to ascertain whether it is appropriate for the reader as summaries are not always available. The only authoritative method of verification is to obtain a full copy.

Cost: Acquiring the literature for a traditional survey is normally an inexpensive task for an individual researcher, with institutional membership often providing access to online digital libraries. This is, however, not the case with books. Although libraries offer access to a range of books, there is a limit to the number of books that they archive. And, in our experience, due to the specialised nature of visualisation books, they are often unavailable. Even if a university library could store all of these books, a library member may only check-out a very limited number simultaneously. Prices for the books contained within this survey range from $20 to $102 with the average book priced at $43, as can be seen in Figure 2.3. The total cost of the literature within this survey comes to a total of approximately $3,600 based on prices from Amazon.com.

Volume: Over 35 focus books are included within this survey, representing a vast amount of reading. To make the survey manageable, we include both focus and non-focus books. Further explanation is provided in section 2.1.3 on scope.
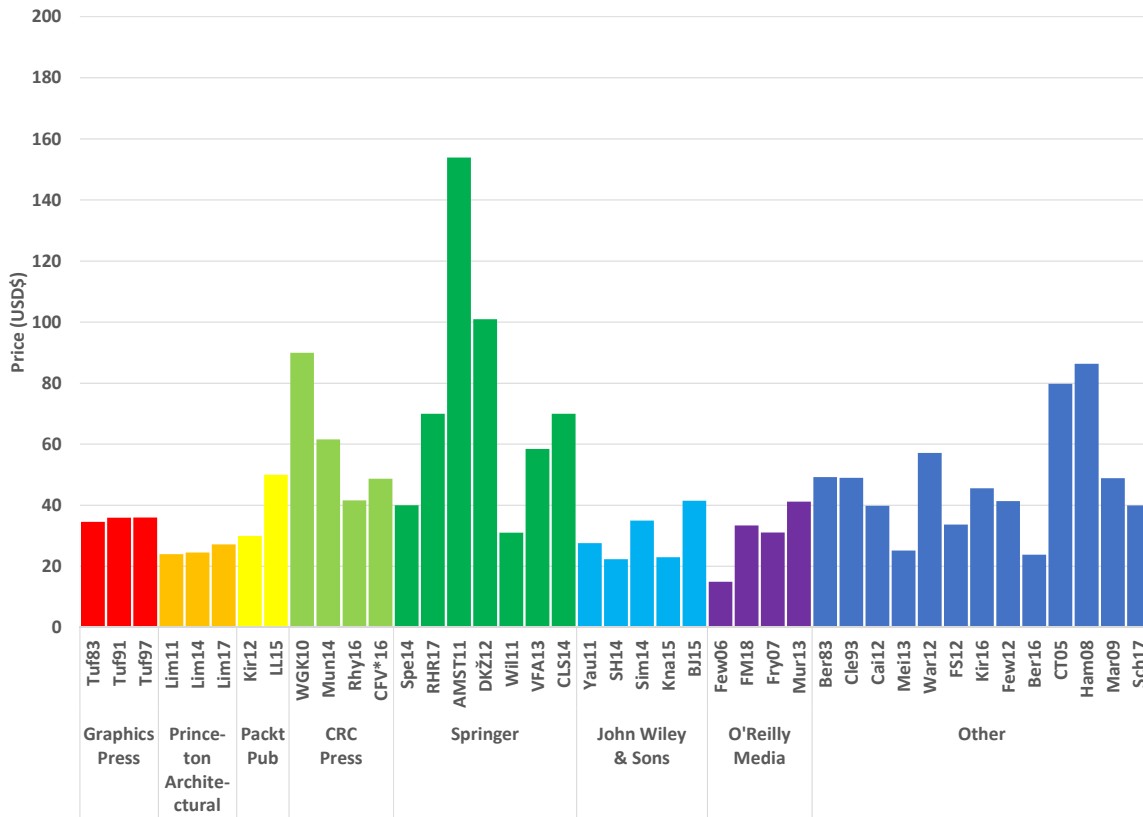
Figure 2.3.: Approximate cost price of each book in USD from Amazon.com. Bars are colour coded according to book publisher.

2.1.2. Book search methodology

This survey is partly inspired by the growing number of books on our own shelves, therefore we began with these. Further references found within books are analysed and relevant books added to the survey. The online book stores Amazon.com and Amazon.co.uk [Ama18] are searched, using both their search function as well as the *"customers who bought this item also bought"* recommendations. Publisher websites are also consulted such as Springer Science+Business Media [Spr18] along with use of their search functions. We searched the following sources for books:

Online book stores

- Amazon.com

- Amazon.co.uk

- John Smith

- Waterstones

- Blackwells

- Wordery.com

Publisher websites

- CRC Press

- Springer Science+Business Media

- John Wiley & Sons

- O'Reilly Media

Other resources

- Our local university Library

  The following search terms were used:

- Visualization

- Data visualization

- Information Visualization

- Visual analytics

- Visual analysis

  We also scanned the references of each book for further related literature.

2.1.3. Book survey scope

Due to the large volume of published data visualisation books, restrictions are applied to constrict the scope of this survey for manageability. The first condition concentrates the survey on information visualisation and visual analytics books. This excludes books that primarily focus on scientific visualisation. We define scientific visualisation as visual design based on data with an intrinsic spatial position such as volume and flow data. Searching for the term *'scientific visualisation'* in Amazon.com's search returns over 300 books, although some books are repeated due to multiple editions and

some results are not suitable. A notable book that falls into the scientific visualisation category is Telea's Data Visualization: Practice and Principles [Tel14], and is therefore out of scope.

Another restriction to the scope is infographics, which are excluded due to the large number of books. Infographics are a manually generated graphic representation of information that presents data quickly and clearly. These are primarily used in journalistic articles and instruction cards. An Amazon.com search for *'infographics'* returns over 600 book results. Example infographic books are *Cool Infographics: Effective Communication with Data Visualization and Design* by Krum [Kru13], *The Wall Street Journal Guide to Information Graphics: The Dos and Don'ts of Presenting Data, Facts, and Figures* by Wong [Won10] and books by McCandless [McC09, McC14]. A particularly excellent example is *Data Flow: Visualising Information in Graphic Design* by Klanten [Kla08] which presents over 200 pages of glossy designs.

Edited volumes are also excluded due to the large volume within this category, with the publisher Springer offering over 50 edited volumes on the subject. Edited volumes are predominantly comprised of a collection of scientific papers which are covered in traditional surveys. This excludes books such as Beautiful Visualization: Looking at Data Through the Eyes of Experts compiled by Steel and Iliinsky [SI10] and Mastering the Information Age: Solving Problems with Visual Analytics by Keim *et al.* [KKEM10].

Books specialising in geographic information systems (GIS) are also excluded due to the extensive volume of books. Searching for *'Geographic Information Systems'* on Amazon.com returns over 4,000 book results, although not all books are relevant and there are some duplicates. Many of these relate to GIS tools such as ArcGIS [esr] and QGIS [QGI]. Cartographic books are considered as GIS and are therefore out of scope. An example cartographic book is the Springer Handbook of Geographic Information by Kresse and Danko [KD12]. This particular book has a cost of $241, highlighting the financial challenge associated with a survey of this kind.

Books dedicated to visualisation tools are partially included. There are a large number of books dedicated to some individual tools, for example D3.js, a JavaScript visualisation library, has over 20 books dedicated to it. As a compromise a single representative book on each individual tool is reviewed with references to the most recent related books. There are many tools that can be used to create data visualisations, therefore a scope boundary is established between propriety tool and open-source

tools, with propriety tools excluded and open-source tools included. A book dedicated to Microsoft's business intelligence tools by Aspin [Asp14] is out of scope because it describes proprietary technology, while a book by Murray on D3.js [Mur13] is included, because it is open source and free.

2.1.4. Two-level classification

We propose a novel two-level classification for the books reviewed. The first level is based on the target audience of each book while the second level classifies the topics discussed within each book chapter.

Primary book classification

We classify books into categories based on the target reader and purpose of each book. We examine both the target reading audience described in the foreword of each book and the book's content in order to categorise each. Using this methodology we define six categories for the classification:

- **Classic:** Books that were originally published over 20 years ago and that are considered as outstanding in the field. For Example Bertin's pioneering Sémiology of Graphics [BBS67].

- **General Audience:** Books with a broader audience that concentrate more on visual impact rather than the visualisation process and casual reading books. We cite Lima's The Book of Trees as an example [Lim14].

- **Academic and Textbooks:** These are books that have a more educational stance written for students of data visualisation, such as Interactive Data Visualization: Foundations, Techniques, and Applications [WGK15]. These are normally targeted at university level students.

- **Industry Professional:** Where the intended reading audience of the book is a professional, from any industry, seeking to learn broad visualisation tips. These books differ from academic books as they provide less background detail. An example in our survey is The Visual Organization: Data Visualization, Big Data, and the Quest for Better Decisions [Sim14].

- **Specialised Books:** We consider the books within this category to be for a very specific niche purpose and audience, such as Aigner *et al.* visualising only time oriented-data [AMST11].

- **Tools:** Books that are dedicated to providing instructions on a particular visualisation tool or programming language for developing visualisations. For example Interactive Data Visualization for the Web [Mur13].

Books are classified chronologically by the year of the first edition within each category.

Second-level book chapter classification

Topics discussed in each book chapter are used as our second-level classification axis. These chapters are then classified into broader topics. These broad topics, based on the information visualisation pipeline [CMS99], consist of introductory topics, data analysis, data mapping, visualisation techniques, rendering, and human factors and future. Table 2.1 shows the overall classification of the books. The values indicate an approximate number of pages allocated to each of the corresponding topics. The introductory topics are subdivided into two categories, visualisation history and the visualisation pipeline. Books often introduce the topic of data visualisation using a historic aspect, citing the work of John Snow [Sno55] and Minard [Min69] amongst others. The visual pipeline by Card *et al.* [CMS99], or modifications of it, are also often used to introduce the subject and to convey its processes. The next broad topic of data analysis includes books that discuss techniques that manipulate the data in some aspect. This is subdivided into six topics: data pre-processing and enhancement, dimension reduction techniques, dealing with uncertainty and error margins within the data, visual analytics, different forms of data, and information theory.

Mapping the data to visual information channels is the third broad topic. This includes the visual design, the use of colour and colour theory, and broader discussion within the topic. Visualisation techniques is the fourth broad topic which is subdivided into the techniques used for various data types and visual design styles. Those subtopics are:

- Spatial data
- Geospatial data

- Time-oriented data
- Multivariate data

- Glyph-based visualisations
- Hierarchical data
- Graphs and networks

- Text and document visualisation
- Scientific visualisation
- Applications

Topics associated with rendering are the next categories listed. These include discussions on presentation, animation, multiple views and the evaluation and comparison of visualisation techniques.

The final subject includes content based around human factors and future research. The full list of subtopics are: interaction, human perception and cognition, aesthetics, profiles of prominent people within the subject, and research directions. Our chapter topic classification mirrors the information visualisation pipeline from Card's classic book [CMS99] and the topic classification used in a Survey of Surveys [ML17].

2.1.5. Online resource

We collect books referenced in this survey in an online resource using SurVis [BKW16] for ease of referencing and book curation. This can be found at the following URL: `http://visbooks.swansea.ac.uk/` The books in this collection can be curated and expanded over time.

2.1.6. Background

To our knowledge, this is the first survey to review and classify books within data visualisation and visual analytics. There are, however, a number of critical reviews of data visualisation books available as articles and blog posts on various websites such as Bremer on her visual cinnamon website [Brea] and a blog post by Cotgreave on the Tableau website [Cot13]. These kinds of reviews have a tendency to be subjective, do not classify the topics covered within each book and rarely review more than ten books. Reviews for each individual book can also be found on online book stores such as Amazon.com [Ama18], however these are scattered, often short, subjective and do not classify topics.

McNabb and Laramee present a Survey of Surveys (SoS) [ML17] which classifies over 80 survey papers within the information visualisation and visual analytics

domains. Alharbi *et al.* [AAM*17] also present a survey of surveys of molecular visualisation within the computational biology field. These meta-surveys present a broad overview of data visualisation landscape, however these surveys only review journal and conference papers and do not consider books.

Group	Book	Visualisation History	Visualisation Foundations/Pipelines	Data Pre-processing & Enhancement	Dimension Reduction	Visual Analytics	Data Types	Information Theory	Visualisation Design	Colour	Data to Visual Primitive Mapping	Spatial Data	Geospatial Data	Time-Oriented Data	Multivariate Data	Glyph-based Visualisation	Hierarchical Data	Graphs and Networks	Text and Document Visualisation	Sci-Vis	Applications	Presentation	Animation	Multiple views	Evaluation	Tools	Interaction	Human Perception and Cognition	Aesthetics	People in Visualisation	Research Directions
Classics	[Ber83]			20					190	8	20		128		76			6	10												
	[Tuf83]	20							110	2			10		20	5									79						
	[Tuf90]	30							126	14			21											14							
	[Cle93]					323	7								253						41			15			14				
	[Tuf97]	50							150	2																					
General Audience	[Lim11]	47																30	172										24		12
	[Cai12]								92						1										50				56	30	139
	[Kir12]		10	5					200				14												2		5				
	[Mei13]	23					2				2		80	57				29	38	20											
	[Lim14]	80															200														
	[Lim17]	60							255				25																		
Textbooks	[Spe14]	7	13						10			27		24	3	2	10	15	1						53		52	14			
	[War12]									26						5											29	393			
	[WGK15]	16	44	14	3		2		26	4		38	32	32	34	6	10	12	24						44		42	58			14
	[Yau11]			27			5		16	10			56	58	106	13															
	[FS12]	20	63	16	1		10		88	6											9						16	2			
	[Mun14]			21			20		51	17	25	20			31	2								32	26		34	14			
	[Kir16]		40	6			34		179	30	7																24				
	[RHR17]								152																			23			
Industry Pro.	[Few12]	4	13				24		216	5	49				15									15			26				
	[Few06]	4					5		110	5					180									180	38		20				
	[Sim14]	1	26	10					5												24						3				8
	[SH14]			30					10						60																
	[Kna15]								160	3	10				50									9			32	4			
	[Ber16]	15							155	8			5												12		20			10	
	[FM18]		45		1		10		31																18		2				
Specialised Books	[CT05]	14		30	2	186	16														26	12					72				100
	[Ham08]								12								176										5				1
	[Mar09]		49	21			45		45	5	8	1	22					8					3		206		8	5			
	[AMST11]	30	15			20	24		33	8			13	250											108		20				10
	[Wil11]	20	2	6			10		44	1	45		2	256	50				4					15	40		27				
	[DKŽ12]					200																									
	[VFA13]					91	74												17												
	[CLS14]												69																		3
	[BJ15]	12		32					61	11	31			23				21	156		29						36	30			
	[Rhy16]	10								170															40					19	
	[CFV*16]		2					194													86										
	[Sch17]	3				38																					32				3
Tools	[Fry07]			67					18	5			60	40	51			38	44												
	[Mur13]			18					9		10		15					7								142	20				
	[LL15]										94									135											

Table 2.1.: Book classification. Books are classified into rows according to their audience and date of first publication. Columns identify topics discussed in each book with an indication of how many pages are dedicated to each topic indicated by numbers and the colour intensity.

2.2. Survey

The surveyed books are organised according to the classification outlined in section 2.1.4, with books placed into one of six categories according to the intended reading audience. Within these categories books are ordered chronologically. The topics discussed within each book and the number of pages dedicated to each topic are indicated in Table 2.1. Topics are outlined in section 2.1.4.

The following section provides summaries of the books within this survey. Each book is presented in its classified category. We attempt to describe each chapter in one to two sentences. We also try to group every set of two chapters into a paragraph. Chapters one and two are in the first paragraph, chapters three and four in the second paragraph etc. This way readers know where in the book they are whilst reading the summaries. It also enables skimming through the context because new paragraphs correspond to odd numbered chapters.

At the end of each classification category, we provide a comparison and recommendation section which compares each book and highlights strengths within the category and provides our book recommendations.

In addition to the meta-data presented in Section 2.1, graphical meta-data for this survey is presented and discussed throughout the survey to improve readability with intermissions from the book reviews and to facilitate comparison of the titles (see Table 2.2).

2.2.1. Classics

The classics section contains books that were initially published over 20 years ago and are noteworthy within the data visualisation community. Six books are found in this category, the pioneering Sémiology of Graphics by Bertin [BBS67], three books by Edward Tufte [Tuf83, Tuf90, Tuf97], and two by Cleveland [Cle85, Cle93], see Figure 2.4.

Description	Section	Description Page Nº	Figure Page Nº
Comparison 1 uses the citation count and Amazon.com sales rank to compare book influence	2.2.1	32	33
Comparison 2 compares the number of references, pages, and figures in each book	2.2.2	41	42
Supplementary material is described in Comparison 3	2.2.3	55	54
Amazon.com reviews are discussed in Comparison 4	2.2.4	64	64
Comparison 5 compares the topics discussed in each of the categories	2.2.5	77	75

Table 2.2.: Tables and figures that include meta-data are distributed throughout this survey.
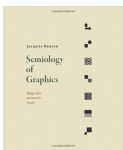
Sémiology of Graphics is an English translation of the 1973 book Sémiology graphique written by Jacques Bertin and translated by William J. Berg [Ber83].

Bertin begins by explaining the difference between the invariant and components of an information graphic, how to identify them, and details the importance of declaring these in titles and legends. Identifying the number of components within a graphic is examined along with the number of categories within the components. The second chapter discusses the properties of graphic marks or glyphs. The three classes of marks are discussed; the point, the line, and the area whilst four groups of glyphs are identified; diagrams, networks, maps, and symbols. A detailed analysis of possible graphical variables is then compiled which are able to encode data, these being variations in two-dimensional location, size, value, texture, colour, orientation, and shape. These graphical variables, first outlined here, form the basis of graphical data visualisation and are discussed in many other texts.

The next chapter begins with a dataset describing the amount and percentage of people working in each of three sectors for each region within France in 1954. One hundred different visualisations are presented of this dataset to demonstrate the effectiveness and variability of visual examples. The efficiency of figures is considered in terms of how easily a user is able to extract information and three functions of graphic representations are identified: to record, to communicate, and to process information. Rules for the construction of different groups of glyphs are then given for their best

Figure 2.4.: Book covers for classic classified books [BBS67, Tuf83, Tuf90, Cle93, Tuf97].

utilisation. Diagrams are the subject of the fourth chapter and are defined as visualisations with one common variable between all data. Multiple examples of diagrams are given for different scenarios such as the number of variables, and different types of variables such as continuous or discrete data.

The fifth chapter's subject is networks - visualisations that have multiple correspondence between variables such as trees and flow charts. Different network arrangements are examined such as rectilinear or irregular, and their construction considered. The final chapter of the book is dedicated to geographic maps. Identifying the scale and the location represented is examined, along with representing the globe as a flat representation and representing geographic features.
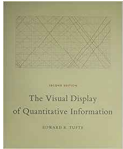
The Visual Display of Quantitative Information Second Edition by Tufte [Tuf83] begins with a chapter that provides guidelines on what makes an excellent visualisation, the ability to simply and clearly show the data and gain insight, along with multiple examples of high quality visual designs. The visual designs also provide a historical perspective of data visualisation, often being the first example of that type of visualisation. The second chapter discusses visual designs that represent data in a poor manner, sometimes deliberately, so that the data can be misinterpreted. Many examples of visual designs are given and explanations of the problems within them, such as miss-aligned axis scales.

The third chapter queries the reasons for poor graphical integrity, concluding that illustrators lack the necessary skill to communicate data effectively. Chapter four introduces the Data-Ink ratio, a ratio of the amount of ink used to represent data divided by the total amount of ink used in the visualisation. Tufte concludes that by maximising the data-ink ratio a better visualisation would be produced, although in some cases, non data-ink is useful for interpretation.

Chartjunk is the focus of the fifth chapter. Three types of chartjunk are identified: moiré patterns, grids and graphical decoration, all of which should be removed from a visual design. A redesign of box plots, bar charts and scatterplots is presented in the next chapter that maximise data-ink.

Chapter seven gives example visual designs that use multifunctional graphics, such as markers that also give data values and encoded glyphs on a scatterplot. In chapter eight, data density is defined as the size of a visualization in relation to the amount of data displayed. Tufte notes that the data density in many published information visualisations is low and that most could easily be shrunk to half their size without the loss of legibility or information. Small multiples are also introduced – shrunk down visualisations that are repeated with one variable changing, highlighting the change in the given variable.

The final chapter provides guidelines for creating aesthetically pleasing information visualisations. These guidelines define when to use tables, the use of words, line weighting, and the shape of the graphic.
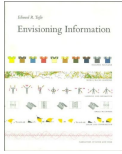
In the first chapter of Envisioning Information [Tuf90], Tufte outlines design strategies that map many dimensions of data onto the 2D medium of paper or a screen and that maximise the amount of data for a given area. This is achieved with the use of multiple examples that include a timetable for a Java railroad from 1937, and methods for representing sunspots over time. The second chapter concerns the representation of large volumes of data in a concise way. Examples of a Tokyo railway timetable and a choropleth of the population distribution around Tokyo are used to demonstrate the principles discussed.

Using layers to differentiate information is discussed in chapter three. This is primarily achieved with the use of colour to differentiate different levels of data, for example, annotations on a diagram. Using lighter colours for the use of guiding marks within figures, such as grid lines, and the consideration of negative space are also examined. Small multiples is the focus of the fourth chapter, the use of multiple small charts to create a single visual design that allows for the comparison of multiple attributes. The use of small multiples demonstrates depicting sequences and comparisons of river lengths.

Chapter five discusses the use of colour and describes the rules set out by Eduard Imhof, a Swiss cartographer, for the use of colour [Imh82]. The final chapter studies the depiction of motion over time using examples that include representations for the movement of the moons of Jupiter, various transport timetables and dance instructions.
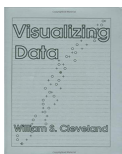
Cleveland begins Visualizing Data [Cle93] by outlining different data types by which the book is organised: uni-variate, bi-variate, tri-variate, hyper-variate, and multiway. For each data type an example chart is provided. The role of visualisation is also outlined; to provide a framework for studying assumptions. Uni-variate data is studied in the second chapter where the visualisation of data using quantile-quantile (q-q) plots is explained and exemplified, along with box plots. Using logarithmic and factor scales for dealing with non-normal data distributions is also demonstrated and validated with the use of visualisation.

Chapter three is dedicated to bi-variate data. Validating line fitting using loess (local regression) to scatter plot data is demonstrated by visualising error residuals. Interaction by brushing data for label information is also briefly discussed. Using techniques presented in the previous chapters, chapter four demonstrates the discovery of variable dependence for tri-variate data. Interacting with scatter plot matrices using brushing in multiple views is also discussed along with contour plots and 3D surface

plots.

Chapter five extends the techniques from the previous chapters to further variables. The addition of data cropping is used to find dependencies. The scatter plot matrix is used to find outliers and dependencies. The final chapter discusses multiway data, where there is only one quantitative variable and multiple categorical variables. Techniques are used to highlight discrepancies in the data. The book concludes with an example of a previously analysed data-set where previous statistical analysis failed to indicate an error, easily identifiable using visualisation.
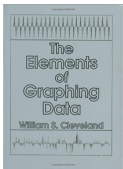
Cleveland has also authored another book titled The Elements of Graphing Data [Cle85]. This book focuses on the basics of visualisation designs such as the proper use of legends, labels, and scales whereas Visualizing Data focuses on visual analytics.

Visual Explanations: Images and Quantities, Evidence and Narrative by Tufte [Tuf97] begins with an analysis of visual techniques used to depict quantities using historic images. The complications of projecting a 3D visualisation onto a 2D display medium, the use of scales to aid perception, and misrepresentation of scales is also discussed. In the second chapter, Tufte explores the statistical and graphical analysis of two events as a case study of the use of visualisation. The first event is the cholera epidemic of London in 1854 where John Snow's analysis proved that cholera was a waterborne disease. The second event was the Challenger space shuttle disaster, where engineers' concerns about the launch temperature failed to prevent the launch.

Chapter three discusses the conveying of instructions demonstrated by pictorial descriptions of magic tricks, as well as the distraction techniques used by the magicians themselves to deceive the audience. The fourth chapter illustrates how minor changes to a visualisation can improve its effectiveness. An example is given of an annotated pictorial diagram of an ear, which is then improved by thinning the annotation lines and another example of a topological map with a suitable and unsuitable colour-map.

The fifth chapter discusses the use of repetition within visualisations. Examples explored include the calendar used on the Russian Salyut 6 space station during the longest space mission of the time, and the repetition of letters on the Trajan Inscription. Chapter six discusses multiples, repetitions of a figure to show different states. Examples given include a depiction of the different views of Saturn as seen from Earth from 1659 and a representation of medical records.

Confectioned figures are the topic of the final chapter. Composite imagined scenes and compartmentalised figures are exemplified and explored.

Comparison of Classics and Recommendations

All of the classic books make excellent use of diagrams to demonstrate principles discussed in the text. The diagrams and figures used in the Tufte texts tend to be of a higher quality than the diagrams in *Sémiology of Graphics* [Ber83] due to the incorporation of colour. The Tufte books also make use of more colour-mapped diagrams in comparison. Computing technology is an underrepresented topic within all books classified as classics due to the immaturity of computer hardware at the time of publication.

Both Bertin and Tufte provide recommendations for producing effective and aesthetically pleasing visualisations, and to avoid perception issues such as moiré effects. Tufte recommends increasing the data ink ratio, where as Bertin describes the optimal use of ink as 5% to 10% for best *"retinal legibility"*.

Sémiology of Graphics [Ber83] provides a comprehensive description of the principles and fundamental rules for the use of graphics to represent data. Bertin incorporates many diagrams to outline principles discussed in the text and presents many visualisation designs, with cartographic designs being particularly well represented. Rules and guidelines for the construction of a graphical diagram and for legibility of a diagram are presented and are particularly useful. The book does however suffer with minimal use of colour and, due to the age of the literature, digitally created images are not discussed. Bertin's *Sémiology of Graphics* is suitable for beginners in data visualisation to understand good graphics principles, or as a reference book for best practice. We recommend the Bertin book for readers that are interested in seeing a historical view of the field, especially on how the field has evolved since 1967. It holds a very special place since it was first, and much of the content is still relevant today.

The Visual Display of Quantitative Information [Tuf83] also makes good use of figures to demonstrate principles outlined in the text. Particularly are figures showing best practice and poor design choices. A good demonstration and description of misleading graphics is provided. Rules for legibility of a diagram are a valuable resource to anyone looking to produce good visual designs. We recommend Tufte's book on *The Visual Display of Quantitative Information* for readers coming from any background. Tufte's

book offers a very engaging and informative exposition accessible to a very wide audience. It is both educational and entertaining.

Tufte's second book *Envisioning information* [Tuf90] continues in a similar vein to his previous book, with many historical figures used to demonstrate themes from the text. Themes discussed in this book include representing multiple dimensions, using layers in visual designs, and the use of colour.

Visualizing Data [Cle93] is more technical than the other books within this category and expects a knowledge of basic statistics. This book provides a statistician's perspective of visualisation. It demonstrates the step-by-step process for the use of visualisation to analyse and validate data. The book is well supported by numerous figures to demonstrate the techniques discussed. We recommend this book for those who are interested in visual analytics as it provides detailed instructions on the subject.

Visual Explanations [Tuf97], as the title suggests, focuses on graphics and diagrams that explain, e.g demonstrates the use of graphics as evidence for decisions. This is the only book surveyed to discuss the use of pictorial descriptions. The chapter titled *"The Smallest Effective Difference"* also offers pictorial examples of how small changes in visual designs can make significantly influence the impact of a diagram, something not offered in other books. The book does however forego information on how to map data to graphics that other books in this category discuss. *Visual Explanations*, like the other Tufte books, makes use of a large number of high quality images and is written in an entertaining style. The book complements Tufte's other books and is suitable to anyone interested in data visualisation. We recommend Tufte's other books to those readers who usually enjoy his classic best seller and are interested in more and similar material including more depth on a range of topics.

Comparison 1.   Comparison of Literature Influence Books within the *'Classics'* category are all highly cited, this can be seen in Figure 2.5 which charts the influence of each focus book described in the survey. The number of citations are on the vertical axis and the Amazon.com sales rank [Ama18] on the horizontal axis (using a logarithmic scale). Tufte's book The Visual Display of Quantitative Information [Tuf83] is notable as being the book with the most citations according to Google Scholar [Goo18], with over 10,500 citations, almost double the citations of the next book, Tufte's Envisioning Information [Tuf90]. Tufte's other book Visual Explanations [Tuf97] is the seventh most cited book out of the books reviewed in this survey, highlighting the influence of Tufte on the data visualisation field. The other book classified as classic, Bertin's
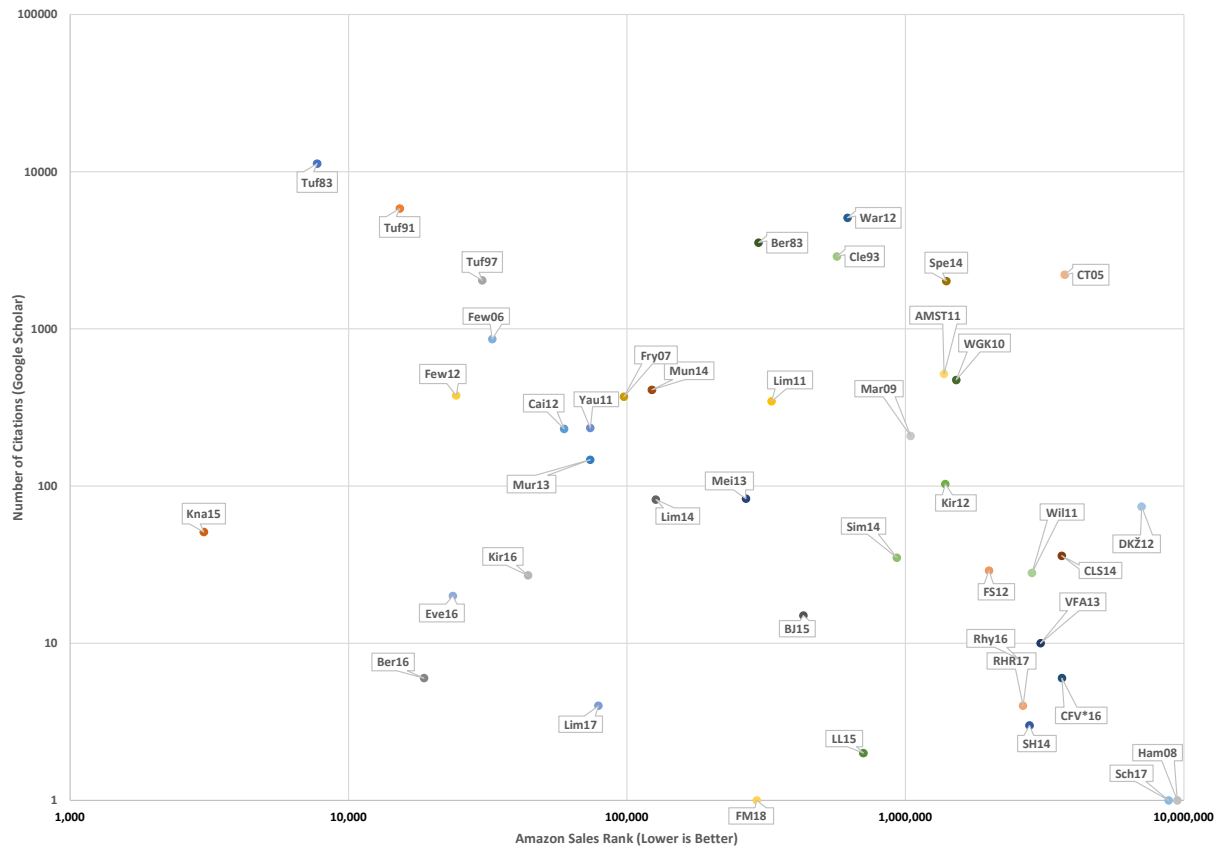
Figure 2.5.: Scatter plot of influence. A visualisation of the number of citations (according to Google Scholar) vs Amazon sales rank. Both axes utilise a logarithmic scale. Analysis of this Figure is discussed in Comparison 1.

Sémiology of Graphics, has also amassed over 3,000 citations according to Google Scholar. It should be noted however, that the books in this category have had a longer time to collect citations.

Knaflic's Storytelling with Data [Kna15] is ranked as the highest book in Amazon's sales rank, followed by two of Tufte's books.

2.2.2. General Audience Books

Books in this section are written for a wider audience than just the data visualisation community. We include six books in this category, the covers for each can be seen in Figure 2.6.

Figure 2.6.: Book covers for general audience information visualisation books [Lim11, Cai12, Kir12, Mei13, Lim14, Lim17] described in section 2.2.2.
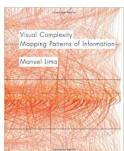
 Visual Complexity: Mapping Patterns of Information by Lima [Lim11] begins by investigating historic tree models, citing them as early precursors to network diagrams. Lima notes that early tree imagery is often due to the cultural importance of trees, and over time these developed into descriptions of scientific knowledge. Dürer's 1509 engraving *"The fall of Man"* [Dur04] and the table of contents from Chambers' Cyclopaedia [Cha38] are examples of the images used in the chapter. The chapter concludes with a note that tree diagrams are still widely used today, although their designs today are non figurative. The second chapter discusses the need for visualising structures more complex than trees. The need for network visualisation is identified in complex problems, such as currency stabilisation, where multiple factors can influence an outcome. Other network problems identified are in urban development, neurological research, the internet, information classification and natural ordering.

The idea of network science is introduced in the third chapter, with the first documented network analysis, credited to Euler [Eul41], with further developments by Moreno and Northway in the field of psychology [NRM55]. The similarity between cartography and network visualisations is highlighted and principles for creating a network visualisation are outlined, such as defining the purpose of the visualisation and encoding the data to visual primitives. The fourth chapter gives multiple examples of network designs organised by the type of data which they convey. Examples of the figures shown are a chord diagram showing email communication by Baker [Bak07], and an arc diagram showing the differences and similarities between the holy books of different religions by Steinweber and Koller [SK07].

Chapter five features 90 images and their descriptions, organised into design type. Figures include a flow chart visualising Last.fm listeners by country made by Adjei and Holland-Cunz [AHC08], and an elliptical implosion depicting word frequency and associations in Carroll's "*Alice in Wonderland*" by Paley [Pal09].The aesthetics of network visualisations is explored in the sixth chapter. Comparisons are made between modern abstract works and network visualisation.

The final chapter features four essays by separate invited authors, which discuss trends in information visualisation and future possibilities such as visualising complex ecological relationships.
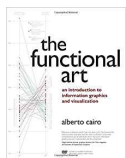
The Functional Art: An Introduction to Information Graphics and Visualization by Cairo [Cai12] begins with an example of fertility rates by country and uses data from the UN to create a visualisation to support a hypothesis. The explosion of data is mentioned and the need to turn the data into knowledge and wisdom. The second chapter uses examples of infographics published in the media analysing them critically and suggesting alternative visualisations that better represent the underlying phenomena, highlighting the need of having a variety of visualisation techniques for the data.

The aesthetic of information visualisations is the subject of the third chapter, with discussion of finding balance between complex and deep visuals, and intelligible and shallower visual designs. The complexity and volume of information displayed is discussed in the context of the audience of the visualisation. Minimalist visualisation versus highly decorated visualisations are also examined. The themes from chapter three continue into chapter four, where further published information visualisations are critically analysed for the information they show and their effectiveness.

The fifth chapter describes the physiological structure of the human eye and how

we build a perception of our surroundings by eye movement. The difference between what we see and what we perceive is also highlighted with some examples of optical illusions. Applying what is known of human perception to visualisations, is the subject of chapter six. The use of shade is shown to be easier to differentiate compared to the use of shape, and space between objects help us create natural groupings. Different methods of encoding data, as outlined by Bertin [BBS67], are also ranked by their ability to allow for more accurate judgements.

Chapter seven describes the process of how humans identify objects, such as faces, by comparing them to our long-term memory. It explains that by only having outlines of objects enables us to recognise objects faster than when they have complex detail. The eighth chapter gives examples of published information visualisations along with some preliminary sketches of those visualisations to show how they developed.

In chapter nine four considerations are highlighted when designing interactive graphics: Visibility - ensure that interactive objects such as buttons are visible, feedback - to offer a conformation of the user input, constraints - the limitations of the interaction are clear, and consistency - fonts, colours, positioning and interaction techniques should be consistent throughout a visualisation. Examples of interactive visual designs are provided. The final chapter presents a profile of ten different information graphic designers, including an interview with each and examples of their work.
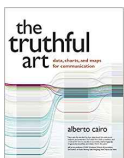
Cairo published a follow up book in 2016, *"The Truthful Art"*, [Cai16] which expands on the principles from The Functional Art [Cai12].

Data Visualization: A Successful Design Process by Kirk [Kir12] highlights the importance of data visualisation in the modern era to help make sense of data, and describes some design objectives that enable assessments of visual designs to be made. The importance of assessing the goal of a visualisation and the management of its creation is the focus of the second chapter. Defining the audience and purpose of the visualisation whether to explain, explore or exhibit are considered to ascertain the design.

Chapter three concentrates on the editorial focus of a visualisation and the process of data acquisition, sorting and analysis. Analysis of data to find stories is discussed and demonstrated with an example exploring the number of medals won by countries over a number of Olympic events. The design options are discussed in the fourth

chapter. Choosing the visualisation method and the accuracy of different encoding channels are considered along with the use of colour and annotations. Interactivity, focus and context and layout are also studied, with all design choices dependant on data properties, data relationships, and audience.

The fifth chapter provides a taxonomy of many different visual designs such as flow charts, bar charts, and different forms of cartographic charts. The final chapter introduces tools that can be used to analyse and represent data such as Tableau, and programming environments such as D3.js. Finally guidance is given on evaluating a finished project and utilising feedback for future work.

Design for Information: An Introduction to the Histories, Theories, and Best Practices Behind Effective Information Visualizations by Meirelles [Mei13] begins by describing the philosophy of the book, to analyse existing visual imagery for design and information content.

The first chapter addresses hierarchies. The importance of spatial layout of a visualisation from a cognitive standpoint and the way humans detect patterns is discussed. Example images of hierarchical visualisations are given including a graphic by Ramon Lull from 1515 and computerised visualisations from the 1990s. A case study of TreeMaps is made, in particular the 'Map of the Market' visualisation by SmartMoney [Wat98], to describe in detail the intricacies of the design. In chapter two the concept of a network as a set of nodes with links is explained, along with extensions such as link direction, weighted links and network path traversal. Node-link diagram examples are given to highlight potential problems in their creation such as occlusion and label placement and to show the many different layout examples such as circular, force directed geographical, radial, and Sankey diagrams.

Visualising time-based data is the subject of the third chapter and is explored through a number of visual designs such as Jacques Barbeu-Dubourg's history since creation from 1753 [BD53] and *"Fifty Years of Space Exploration"* by McNaughton and Velasco in 2010 [MV10]. A brief history of maps is given in chapter four, accompanied by an explanation of map projections, a discussion on map scale and encoding data onto a map. Multiple examples of maps with different coding of data attributes are given, for example, choropleth and isometric maps.

The fifth chapter examines visual designs that have variables in both time and space. Examples are used such as Minard's 1869 *"Napoleon March to and from Russia"* [Min69] and mobility patterns showing traffic in Milan. Visual layouts that represent text data

are the subject of the sixth chapter. Wordle [Fei14] and IBM ManyEyes [VWVH[*]07] are used as case studies to show examples of text visualisation.
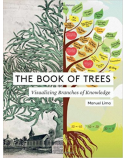
The Book of Trees by Lima [Lim14] starts with a historical look at the use of tree representations throughout history, from tree impressions on Bronze Age, Mesopotamian cylinder seals to treemap representations of folder structures on a PC's hard drive. Significant characters who used depictions of trees throughout history are also discussed, from Aristotle, through René Descartes to Shneiderman.

The first chapter, shows multiple tree images depicted as realistic trees are presented in a chronological order. Chapter two focuses on vertical hierarchical structures without dendrological depictions. Sixteen figures are given of various vertical trees such as *"X-Men Family Tree"* by Joe Stone in 2011 [Sto11].

The third chapter discusses horizontal trees. Multiple figures are given of seminal trees including the table of contents from 1728 Cyclopaedia by Ephraim Chambers [Cha38]. Chapter four discusses multidirectional trees, trees with a hierarchical structure not fixed to a horizontal or vertical axis.

Trees with a radial layout is the subject of the fifth chapter. Examples are given such as a complete phylogenetic tree for all mammal species by Bininda-Emonds et al [BECJ[*]07]. The sixth chapter explores hyperbolic trees, a variation of radial trees which allows the focus of the visualisation to be placed on a single node using layout algorithms. This enables the depiction of large datasets with focus and context interaction.

Chapter seven examines rectangular treemaps, a hierarchical visualisation using nested rectangles. Examples of rectangular treemaps given are a 1874 chart showing the population of states of the USA by Walker [Wal74], Wattenberg's *"Map of the Market"* [Wat98] and a treemap of a PC folder structure by Johnson and Shneiderman [JS91]. Voronoi treemaps, where the space is filled with mathematically derived space rather than rectangles, is the subject of the eighth chapter.

The next chapter discusses another variation of treemaps, the circular treemap. Lima notes that these aren't as popular as other treemaps due to their ineffective use of space. Example images include a circular treemap of the market value of 2,000 public companies featured in the previous chapter. Sunbursts visualisations or radial treemaps are the subject of the penultimate chapter. These are similar to treemaps, but use a radial layout with the root of the hierarchy at the centre of the visualisation.

The final chapter discusses icicle trees, these are a non-nested method of showing a hierarchy that can be used in either a horizontal or vertical arrangement.
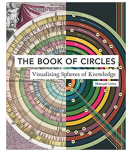
The Book of Circles by Lima [Lim17] commences with an exploration of circular objects in nature and the use of circles in human history. Circular cities and artefacts, such as Stonehenge in the United Kingdom, circular symbols, the use of circles as metaphors and human preference for curvilinear lines are examined. The second chapter classifies different forms of circular designs into seven families upon which each of the next chapters will be based. The families consist of rings and spirals, wheels and pies, grids and graticules, ebbs and flows, shapes and boundaries, maps and blueprints, and nodes and links.

Over 30 images of ring and spiral representations are presented in the third chapter such as a NASA compiled photo of Jupiter focused on its North Pole. Wheels and pies are presented in the next chapter. These are circular diagrams that have radiating lines from the centre of the circle. Some examples presented in the book are multiple pie charts depicting the religious makeup of each US state and Florence Nightingale's diagram depicting causes of mortality during the Crimean war.

The fifth chapter features images combining circular depictions with grid structures. Circular depictions which ebb and flow are the subject of the sixth chapter. Figures such as radar charts, circular bar charts, radial area charts and radial line charts fall into this category. Visual designs that captivate images of circles within a circle, circular treemaps, and Voronoi patterns are presented in the following chapter. An example of an image depicted is a Voronoi diagram representing the various sizes of countries' economies.

The penultimate chapter looks at circular depictions of geospatial information. Example figures include architectural blueprints, historic maps and other geographical depictions. The final chapter explores different circular node link diagrams such as a depiction of co-authorship between physicians publishing on hepatitis C.

Comparison and Recommendations of General Audience Books

These general audience books by their nature feature an accessible reading style and have a high number of large images.

Visual Complexity by Lima [Lim11] features almost three hundred full-coloured images with brief captions on the data they represent. A detailed description of the

visual history of trees and networks is provided. Other than historical aspects, the book does not provide explicit educational function in order to study visualisation techniques. However multiple designs are presented and due to the large number of examples, the book may be a good source of inspiration. We recommend this book to a wide audience, however the book is not recommended to those looking for pedagogical guidance in data visualisation. This book is very image-oriented. Lima does an outstanding job of collecting a wide range of a visual designs and examples inspired by trees and networks. He pays special attention to aesthetics.

The Functional Art by Cairo [Cai12] covers a wide range of data visualisation topics, however the focus of the book is towards infographics. A thorough description of visual perception and cognition is provided, as well as tips for creating interactive graphics. Profiles of infographic designers are also given along with examples of their work, a unique feature within all books surveyed. Images are used to support topics discussed in the text, however *The Functional Art* doesn't have as many visualisations as are presented in the Lima books. We recommend this book to beginners in information visualisation, particularly those with an interest in infographics.

The process of creating a visualisation is discussed by Kirk in *Data Visualization: A Successful Design Process* [Kir12]. Different perspectives on data visualisation are discussed dependant on the purpose of the visualisation and the intended audience. A useful taxonomy of visualisations is presented, showing a number of visualisation designs, along with a useful range of digital tools to aid visualisation. The book is written in an informal style, however it does not feature as many interesting images as the other books within this category. We recommend this book to beginners in data visualisation with the book providing a basic guide and a useful list of digital resources. This book is more pedagogical than the others in this category. It's intended for those that are really interested in applying the guidelines and principles presented. However it's not in the textbook category because it's for a general audience (as opposed to a special target student audience).

Meirelles' *Design for Information* [Mei13] is divided into six chapters, each of which discusses the visualisation of particular data type; hierarchical, relational, temporal, spatial, spatio-temporal, and textual. For each of these data types, a history of visualisation is presented along with a number of visualisation designs to demonstrate different methods to encode the data. The book provides a number of high quality images to compliment the text which provides an entertaining read. This book, along with others in this category do not present information with regard to computational

technology and creating digital representations. It's a concise book which can be read quickly, so we recommend this book to readers interested in a quick and casual introduction to information visualisation. We also recommend this book to those with interest in visualisation designs, such as design students. The author's style follows a principle-by-example style.

Lima's second book within this category, *The Book of Trees* [Lim14], is the successor to visual complexity with a stronger focus on tree visualisations. Almost two hundred examples are presented in colour, along with a detailed history of tree visualisations. The book is arranged by visualisation design such as different forms of treemaps and sunbursts. Similar to Lima's *Visual Complexity*, educational content is limited to a historical aspect, but many visualisation designs are presented. As with Lima's other books we recommend this book to a wide audience or for those interested in hierarchical visualisation and who also liked his first book. Again Lima's style focuses on images and examples. He has gone to great length to collect interesting examples of tree-like visual layouts that are very engaging. This is definitely a strength of each of his books.

The Book of Circles [Lim17] is Lima's third book in this survey and follows a similar recipe to the others. Imagery with of circles is the focus of this book with the examples themselves taking centre stage. A historical aspect is presented along with three hundred graphic examples. Visualisation fundamentals and other educational aspects are not presented. We recommend this book to a wide audience interested in nice visuals or for those interested in circular designs and readers who enjoyed his two previous books. We strongly recommend Lima's books for those readers interested in an example-image-based approach to the topic. His collection of examples are impressive and fascinating.

Comparison 2. Comparison of Book Quantities Figure 2.7 shows the number of references, pages and figures contained in each book described in the survey. The most notable book in the chart is Bertin's Sémiology of Graphics [BBS67] which has the greatest number of figures and references no other work. Books from the Textbooks and Academic category tend to have a higher number of pages and cite more references in comparison to other books within the survey. Ware's Information Visualization: Perception for Design [War12] contains the most references of the surveyed books and Interactive Data Visualization by *Ward et al.* [WGK15] has the highest number of pages.
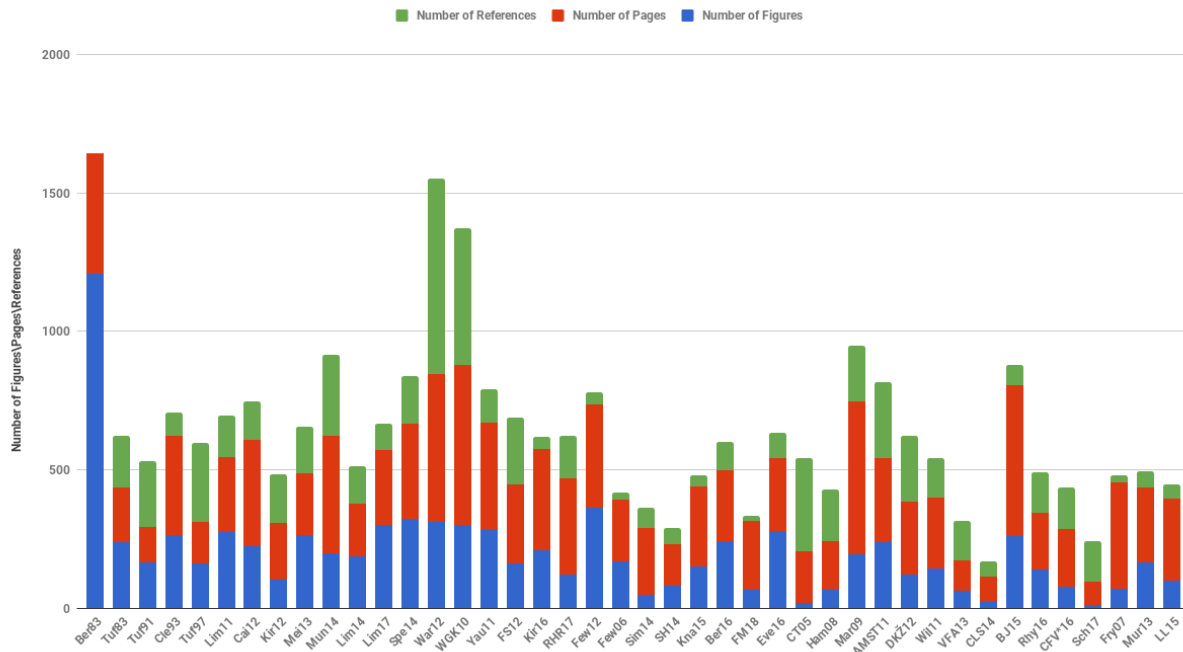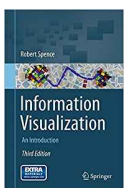
Figure 2.7.: Stacked bar chart comparing the number of references, pages and figures in each book, discussed in Comparison 2.

2.2.3. Textbooks and Academic

This section contains eight books that are targeted predominantly at data visualisation students at the university level. Book covers can be seen in Figure 2.8.

Information Visualization: An Introduction, Third Edition by Spence [Spe14] begins by showing different examples of visual designs from history and explains why information visualisation is important – to gain insight. In chapter two an example dataset is used to construct a visual design and to show the benefits of using the appropriate visualisation method, highlighting challenges along the way.

The third chapter introduces visual layout techniques such as star plots and parallel coordinates. These are then critically analysed with human perception factors in mind. Interaction techniques are briefly discussed and data relationship visualisations are assessed. Display media other than paper or a PC screen are also discussed. Chapter four explores the use of techniques such as focus+context and zooming as methods to make the most of display space. A discussion of using temporal space is also presented and the considerations of the use of both spaces to enable effective visualisation.

Figure 2.8.: Sleeve covers for books classified as textbooks [Spe14, War12, WGK15, Yau11, FS12, Mun14, Kir16, RHR17] described in section 2.2.3.

Norman's action cycle is introduced as a model to develop interaction techniques and an example of its use is given in the fifth chapter. The considerations of intuitive design and expected results from interactions are discussed. The author identifies eight explicit steps for creating an interactive information visualisation in chapter six. These steps range from receiving the commission, through idea generation and sketching, to presenting the final product.

Case studies of designing visualisations for various problems are given and critically analysed in the final chapter. A description of videos to compliment topics raised in the book are given in the appendix.

Information Visualization: Perception for Design, Third Edition by Ware [War12] considers the importance of human perception when designing visual layouts along with the importance of reducing cognitive load. Perceiving real-world environmental factors such as surface textures and the behaviour of light on computer monitors are considered in the second chapter. Other digital visualisation techniques such as augmented reality, virtual reality and display walls are also discussed with a focus on how much the human visual system can comprehend.

Chapter three examines contrast, luminance, and brightness and the way that they are perceived. A detailed description of how colour is perceived and accurately reproduced is given in chapter four along with guidance on the best use of colour.

The fifth chapter explores visual primitives such as colour, size, shape and orientation. To create distinctive glyphs to stand out among others, at least one of those visual primitives must be vastly different. The perception of textures, two dimensional shapes, proximity and connectedness are evaluated in chapter six. The visualisation of multi-variate data is explored, overlapping data, parallel coordinates, treemaps, and dynamic visual designs are all examined with a view to find patterns in the data.

Chapter seven explores the visualisation of data in 3D space. Depth cues such as shading and textures in standard and stereoscopic displays are considered. The perception of surfaces and the positions of data points in space are also examined. Theories of object recognition and pictorial representations are discussed in chapter eight.

The use of images, text, verbal dialogue and animations in data visualisation is scrutinised in the ninth chapter, with the aim of a narrative in mind. The importance of the use of gestures, such as the use of arrows to highlight, is also discussed. The tenth chapter provides a detailed explanation of how data interaction techniques should behave, e.g. providing visual feedback, and the theory of why is discussed.

The thinking process of the user is considered in chapter 11 with emphasis on cognitive load and memory. Example algorithms are given that describe the user interaction with differing visual designs, with the aim of reducing cognitive load. The appendix gives information on CIE colour standards, goals that would evaluate a visu-

alisation, and summarises all of the guidance points highlighted throughout the book.
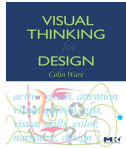
Ware has published an additional book [War10] in which the perception concepts discussed in Information Visualization: Perception for Design are expanded on to provide advice for designers. In the interests of scope, managing volume and overlapping subject material this book is not given a full review within this survey.

Interactive Data Visualization: Foundations, Techniques, and Applications, Second Edition by Ward *et al.* [WGK15] starts with an introduction to information visualisation and why it is important, for information understanding, along with a brief history of information visualisation. The visualisation pipeline is introduced and human perception discussed. Data characteristics are described in the second chapter along with some data preprocessing techniques such as normalisation and sub-sampling.

Chapter three deals with human perception of graphics and images specifically in information visualisations. A physiological description of the eye is given along with many figures to highlight perception oddities. Card *et al.'s* visualisation pipeline [CMS99] is discussed in detail in chapter four, along with the use of visual primitives. Data visualisation taxonomies, such as Shneiderman's [Shn96], are also discussed.

Chapter five presents different techniques used for scientific data, 1D, 2D and 3D spatial coordinates along with dynamic flow. The sixth chapter provides a basic introduction to geospatial data with specific real-world coordinates. Map projections, representing the globe as flat are followed by techniques used to visualise area, line and point data on maps.

Chapter seven focuses on different categorisations of time oriented-data and gives an example of each. The categorisation is based on abstract or spatial data, univariate or multivariable data, linear or cyclic time, instantaneous or interval data, static or dynamic visual designs, and 2D or 3D visualisations. Chapter eight discusses different techniques used to show multivariate data categorised into point, line and region based, including combinations of techniques.

Visualisation techniques for trees, graphs and networks for data interrelationships are presented in the ninth chapter, such as treemaps, sunburst displays and node-link diagrams for both hierarchical and arbitrary data. Chapter ten introduces text and document visualisation techniques and some example approaches such as word clouds and arc diagrams are given. Document collection visualisations are also presented.

Chapter 11 introduces different classes of interaction techniques: navigation, se-

lection, filtering, reconfiguring, encoding, connecting, abstracting, as well as combinations of those classes. Interaction spaces are also identified as screen, data value, data structure, attribute, object or visualisation structure. Chapter 12 discusses and exemplifies algorithms used for each of the interaction spaces identified in the previous chapter. Algorithms for animating the interaction and interaction control principles are also discussed.

In chapter 13 guidance is given for designing visualisations such as data mapping, selecting views, information density, labelling, use of colour, and the importance of aesthetics. Potential problems that may occur when producing visual layouts are also identified. The importance of identifying the purpose and the audience of the visualisation are highlighted in chapter 14, along with data characteristics and image characteristics in order to benchmark different visualisations.

The penultimate chapter presents software tools for producing different forms of data visualisations such as GGobi [SLBC03], Graphviz [EGK*01] and Tableau [Tab]. Active research fields in data visualisation are highlighted in the final chapter. A history of computer graphics and visualisation is given in the appendix along with some sample datasets and some example program code.
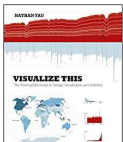
Visualize This: The FlowingData Guide to Design, Visualization, and Statistics by Yau [Yau11] begins by explaining that visualisation is useful to present numerical data in an engaging manner, and to show patterns and relationships. Basic principles are outlined for creating visual designs such as labelling axes and data correctly, citing data sources, and to consider the target audience. The second chapter discusses the data used to create the visual layout, where to find data sources and how to format the data. Example Python code is given to extract data from an online resource and some applications are discussed that will reformat data.

Example tools used to visualise data are presented in the next chapter. The applications are split into 'Out of the box' tools such as Microsoft Excel and Tableau, 'programming' tools such as Python and R, 'illustration' tools such as Adobe Illustrator and Inkscape, and geographical 'mapping' tools such as ArcGIS and Polymaps. Chapter four concerns the visualisation time-dependant data. Bar graphs, scatter plots and line graphs are used to display the time data along with detailed description of their creation using R and refined with Adobe Illustrator.

The fifth chapter concerns graphics that use proportional representations, such as pie charts, stacked bar graphs, treemaps and stacked area graphs. Some R code exam-

ples are provided for their creation. Visual designs for finding data relationship are discussed in chapter six. R code examples are again given for scatterplots, scatterplot matrices, bubble charts and histograms.

Chapter seven is concerned with visualising high dimensional data. Heatmaps, Chernoff faces, star charts and parallel coordinates are used as examples along with R code for their creation. Dimensionality reduction is also briefly addressed. The penultimate chapter is based on geographic data. Python and R visualisation script examples are given that link to various APIs such as Google Maps to create images. Interactive flash based animations are also used along with coding details.
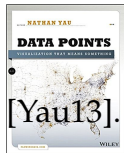
Yau also has a related book called *"Data Points: Visualization That Means Something"* which focuses more on the graphical side of data visualisation. [Yau13].
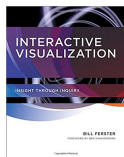
Interactive Visualization: Insight Through Inquiry by Bill Ferster [FS12] commences with different genres of the topic and describes the historic advancements with the introduction of computers and the internet. Some theoretical models are discussed such as the cognitive load theory and, in particular, the ASSERT model. The ASSERT model (Ask, Search, Structure, Envision, Represent, Tell) provides a framework for creating interactive visual designs and also provides the structure for the next chapters. Chapter two is based on the first step of the ASSERT model and is titled *"Ask a Question"*. The chapter discusses how to create a solid question to drive the development of a visual design. Considerations are made to the intended audience and their expertise, and for creating a focused question. Techniques for generating a good question are also discussed.

The third chapter addresses the *"Search for Information"* section of the ASSERT model, which considers sources of data. The difference between primary, secondary and tertiary sources of data are explained, data quality is discussed and example sources of data are provided. Structuring the information is the next step of the ASSERT model. Quantitative versus qualitative information is explained and data structures and comparison with other datasets are discussed. Linked Data and Semantic Web projects that aim to structure data on the internet are also mentioned.

Chapter five concerns the Envision section of the ASSERT model which examines strategies for analysing and representing the data to answer the question. Analysis techniques are discussed for quantitative data such as relationships, and qualitative

data such as frequency analysis. Creating a visual design is the topic of the sixth chapter which is the Represent section of the ASSERT model. Many aspects to be considered when creating imagery are discussed such as perception and cognition, usability, aesthetics, use of colour, and spatial arrangement. Different display strategies are discussed and a number of tools that can be used to create visual depictions are presented such as Wordle [Fei14].

The final part of the ASSERT model is tell a story. The use of a storytelling component in a visualisation is delineated and examples of storytelling visualisations such as Minard's map of Napoleon's march to Moscow are given before a discussion of misleading representations. Chapter eight provides a case study of depicting the books available at the University of Virginia's first library when it opened, to show the application of the ASSERT model.

The ninth chapter discusses the technology behind the internet and how it is accessed including the use of Adobe's Flash and vector graphics. Statistical techniques for data processing are presented in the tenth chapter, for example standard deviation and regression.

Chapter 11 provides guidance on using spreadsheet programs such as Microsoft Excel and Google Docs. Databases are briefly explained in chapter 12 along with XML.

The 13th chapter discusses methods to improve the accessibility of visual designs created for example taking into consideration the colourblind and providing clear alternative text. The final chapter discusses VisualEyes [Uni], a Flash-based authoring tool to create dynamic visual designs.

Visualization Analysis and Design by Munzner [Mun14] begins with a discussion of why computer graphics for data is needed, for the discovery of knowledge in data, and gives a broad overview of considerations, such as resources available, for creating visual designs. Data types, their format and their sources are categorised and explained in the second chapter. Example data types discussed are hierarchical, geometric and time dependant data.

The third chapter explores the importance of considering the reasons for creating visual representations, who the user is and what goals the imagery is trying to achieve. This is to ensure that the appropriate design choices are made when creating a visual design. Examples are given of visual designs created using derived data. Chapter four describes four distinct levels for creating a visual design; identifying the requirements of the visual design, abstracting the data, designing the visual layout and interaction, and encoding the visual design computationally. The importance of validating each of

these stages is highlighted.

Methods for displaying data are discussed in chapter five, including different ways of encoding data into graphical primitives or 'channels' such as colour and size. A detailed analysis is also made of the effectiveness of the different attributes with a consideration of human perception. Chapter six presents eight rules of thumb for consideration when designing visual layouts along with detailed justification of each rule. Example rules of thumb include the use of Shneiderman's information seeking mantra [Shn96].

Visualisation techniques for data found in tabular form i.e. information data, are discussed in the seventh chapter, along with examples such as scatter plots, stream graphs, bar charts and parallel coordinates. Spatially ordinated data visualisations are considered with examples given in chapter eight. These include geographical data as well as scalar and vector fields. The techniques used to visualise them are also discussed here.

The ninth chapter examines network and tree diagrams, various node-link diagram examples are given and their benefits weighed against matrix views. Hierarchical visual designs examples are also provided including treemaps. The use of colour to map data attributes is studied in detail in chapter ten, in particular the perception of luminance, hue, and saturation used in different scenarios. The use of other attributes such as shape, size and texture are also mentioned.

Interaction techniques such as highlighting elements and changing the viewpoint by zooming and scaling are described in chapter 11 along with cutting and slicing of 3D scenes to produce 2D visualisations. Considerations of dissecting data to create multiple views are discussed in chapter 12. Adjacent views and layered views are presented with multiple programs using the techniques described.

Chapter 13 presents options for reducing data by filtering. Three broad methods are described; reducing the number of items, reducing the number of attributes, and grouping data. Examples of applications that use these techniques are given. Focus and context techniques are split into three categories in chapter 14; superimposing layers, geometry distortion, filtration and aggregation. Example applications for each of the categories are given.

The final chapter provides case studies of six different applications, namely Graph-Theoretic Scagnostics [WAG05a], VisDB [KK94], the Hierarchical Clustering Explorer [SS02], PivotGraph [Wat06], InterRing [YWR02], and Constellation [MGR99].
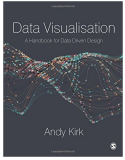
Data Visualisation: A Handbook for Data Driven Design by Kirk [Kir16] begins by defining data visualisation as *"The representation and presentation of data to facilitate understanding"*, and presents three principles for good visualisation design, to be trustworthy, accessible and elegant. The second chapter outlines a workflow and describes a mindset for creating an information visualisation design. The workflow forms a basis for the layout of the rest of the book with the first step of formulating the purpose of the design, followed by data handling, the focus of the visual design, and designing the visualisation.

Chapter three discusses the initial stage of creating a design, formulating the design brief. Considerations of the audience, constraints such as time, available technology and required deliverables are discussed. A discussion is made of different data types, textual, nominal, ordinal, interval and ratio in chapter four. Sources of data and preprocessing the data into a usable format are also discussed.

Chapter five presents the design approach to visualisation, the editorial angle, what data is included and what the focus of the visualisation is. Examples of visual designs are studied to demonstrate the editorial aspect such as a series of charts showing National Football League touchdown passes. The sixth chapter discusses visual encoding methods and describes 49 different types of charts and their properties, ordered by the type of data that they convey.

Interaction options are split into two features, data adjustment and presentation adjustment in chapter seven. Examples of data adjustment include filtering, animating, and data exploration, whilst examples of presentation adjustment are visual emphasis, annotations and orientation. The use of annotations is discussed in chapter eight. The use of headings, introductions and footnotes, as well as chart marks, labels, and legends is examined.

An overview of colour theory is given in chapter nine before a discussion of the most appropriate use of colour for data legibility and emphasis. The penultimate chapter discusses the composition and layout of a visualisation. Details such as the labelling and ordering of scales, and chart size and orientation are examined.

In the final chapter a critical analysis of a visual design is provided, highlighting the techniques made throughout the book. The multidisciplinary nature of being a data visualiser is also explored.

Five Design-Sheets: Creative Design and Sketching for Computing and Visualisation by Roberts *et al.* [RHR17] begins by detailing the book's purpose, as a method to help create a software tool, in particular a data visualisation tool. The layout of the book consists of three parts matching the process of developing a tool: think, prepare, and sketch. The second chapter provides an overview of the Five Design-Sheets method for planning and designing software interfaces by sketching. As the name suggests, the method involves five large sides of paper, the first for sketching ideas, the second, third and fourth for different design solutions and the fifth for a sketch of the final design. The structure of the sheets is described and the use of the Five Design Sheets method in different scenarios is detailed.

Chapter three discusses different forms of problems, characterising them and various ways of thinking to address a variety of problems. Social and ethical considerations are studied in the fourth chapter. Issues such as evaluating the need for a particular software tool, whether an existing tool exists, if the resources are available to create the tool and if the authority exists to create the tool are examined along with ethics of software development.

The fifth chapter looks at sketching skills and techniques. The use of sketching in planning is highlighted and uses of colour and line thickness to enhance sketches is explained. Tips to improve sketching skills are also provided with a recommendation to create a sketching kit. Chapter six explores the Gestalt principles and how humans interpret graphical marks on a page. Bertin's '*Representational Styles'* presented in Semiology of Graphics [BBS67], which classify whole graphical images, are also explored.

Methods to help creative thinking and generate ideas are presented in the seventh chapter such as taking inspiration from nature and being well rested. The intricate details of the Five Design-Sheets method are laid out in the next three chapters. Chapter eight details the first sheet which is to frame the problem, gather ideas, refine them and to define three potential ideas to move onto the next stage.

Sheets 2, 3 and 4 of the method are presented in chapter nine, which describes each of the three design solutions. Each design should be carefully considered and detailed with pros and cons for each design as well as how the user will interact with the design. The final sheet, detailed in chapter ten, features one of the previous three designs taken forward as a final solution. The sheet should be detailed enough that the product can be created from it.

The final chapter of the book provides two examples and runs through the use

of the Five Design-Sheets method with each. The first is a tool for capturing and displaying prehistoric monuments and the second is a tool for visualising computer algorithms.

Comparison and Recommendations of Academic Books

Spence's book *Information Visualization: An Introduction* is aimed at university students from any discipline who need to visualise data. The book focuses on creating a visual design whilst presenting data visualisation principles. Exercises are also given at the end of each chapter. Some unique topics discussed in this book are eye tracking and gaze heat maps as well as alternative canvases. Spence's book is the first academic text book on information visualisation and thus features early research work in this field. We recommend this book for readers that would like a concise academic introduction to the field as it is not as comprehensive as some of the other books.

Information Visualization: Perception for Design by Ware provides a comprehensive analysis of human perception and vision with an emphasis on data visualisation. Although other books within this survey discuss perception, none are as detailed as Ware's work, and often reference Ware's work themselves. Other topics within information visualisation are not discussed in much detail within this book due to its strong focus on visualisation perception. We recommend this book as the principal book for vision and perception topics because the author has a deep knowledge and extensive background in the field of vision psychology.

Interactive Data Visualization: Foundations, Techniques, and Applications by Ward *et al.* is more formal than other books in this survey with references to scientific papers and discussions of algorithms. Like some other books in this category, exercises are presented for university students to complete. A topic unique to this book in this category is research directions, where future research topics are explored. We recommend for those with some prior knowledge of computer science. This book is more technical than other books in this section, with the exception of the focus of Ware's book. This is actually the textbook we recommend as a starting point for university (computer science) students in data visualisation. This is because the book is up-to-date and has the most comprehensive view of the field. It is a good starting point.

Yau's book *Visualize This: The FlowingData Guide to Design, Visualization, and Statistics* provides step-by-step instructions on how to create visual layouts, providing coding examples that a reader can follow along with for multiple tools. The book is written in a casual and informal style. We recommend this book for beginners in visualisation who are looking to quickly create imagery, with the book focusing on producing visual designs over educational content. We also recommend this for university students that are not necessarily computer science students. It provides more step-by-step guidelines for readers than many of the other books in the category.

Interactive Visualization: Insight Through Inquiry by Ferster provides a framework for creating a visualisation design, based on the data visualisation principles. A list of digital visualisation resources is provided for creating a visual design.

Visualization Analysis and Design by Munzner provides an easy to read and educational introduction to data visualisation, with many predominant topics of the subject covered. See Table 2.1. Of particular use are eight rules of thumb that Munzner presents for the creation of a visualisation. In contrast to Ward *et al.'s* book, that works starting with algorithms as its base and moving upwards, Munzner's book starts from the conceptual level and works its way down and stops one level above algorithms. We recommend this book for university students more interested in the conceptual level of visual design.

Kirk's *Data Visualisation: A Handbook for Data Driven Design* provides another perspective for creating a visualisation design from concept to realisation, similar to Spencer's, Ferster's and Yau's books. An easy and informal style is used. A taxonomy of 49 different visual designs is a useful resource available in this book, however fewer images are used overall. Exercises are available with the book, but only from an online resource. We recommend Kirk's book to students that are less interested in the research aspects of data visualisation and are more interested in the contemporary aspects and culture. This is because it draws on many contemporary and popular sources for examples and inspirations such as many web pages, news web sites, and YouTube.

Five Design-Sheets: Creative Design and Sketching for Computing and Visualisation by Roberts *et al.* presents a method for creating ideas, designing, and developing a software tool, with a focus on an information visualisation tool. This book differs from the others in this category by not focusing on visualisation design principles. We recommend this book to readers who are interested in deriving and brainstorming visual designs at the conceptual level. This book goes into the most depth on how to

come up with a range of visual designs centred around addressing a specific challenge. It is very practical in nature and provides detailed guidance on how to brainstorm and even recommends specific materials to use.

Book title	Supplementary material available	Supplementary material description
Visual Explanations [Tuf97]	www.tufte.com	Author website
Visual Complexity [Lim11]	visualcomplexity.com	Showcase of visualisation projects
The Functional Art [Cai12]	DVD	Three video lessons
Information Visualization: An Introduction [Spe14]	http://extras.springer.com/2014/978-3-319-07340-8	Supplementary video clips
Interactive Data Visualization [WGK15]	www.idvbook.com	Data sets, images, videos and software
Visualize This [Yau11]	http://www.wiley.com/WileyCDA/WileyTitle/productCd-0470944889.html http://book.flowingdata.com/	Supplementary code
Visualization Analysis and Design [Mun14]	https://www.cs.ubc.ca/~tmm/vadbook/	Supplementary slides, figures, references and author interviews
Data Visualisation: A Handbook for Data Driven Design [Kir16]	book.visualisingdata.com	Exercises, additional reading and case studies on website
The Visual Organization [Sim14]	www.philsimon.com	Author website
Storytelling with Data [Kna15]	http://www.storytellingwithdata.com/	Author podcast, blog and videos
Making Data Visual [FM18]	http://resources.oreilly.com/examples/0636920041320	Supplementary code and video demonstrations
Illuminating the Path [CT05]	http://nvac.pnl.gov	Site expired
Applied Security Visualization [Mar09]	CD	Live CD containing tools
Visualization of Time-Oriented Data [AMST11]	http://www.timeviz.net/	Showcase of visualisation projects
Multidimensional Data Visualization [DKŽ12]	Datasets	Dataset details are provided in the appendix
Graph-Based Clustering and Data Visualization Algorithms [VFA13]	www.abonyilab.com	Research group website
Visualizing the Data City [CLS14]	Over 60 URLs provided in appendix	Links to projects and data sources
Graph Analysis and Visualization [BJ15]	www.wiley.com/go/GraphAnalysisVisualization	Data files, sample code
Visualizing Data [Fry07]	http://benfry.com/writing/	Author website
Interactive Data Visualization for the Web [Mur13]	http://alignedleft.com/tutorials/d3	Link to code repository
OpenGL Data Visualization Cookbook [LL15]	https://www.packtpub.com/books/content/support/19795	Code available at website

Table 2.3.: The supplementary material available from reviewed books, where available. Books are ordered in classification order, as can be seen in Table 2.1.

Comparison 3.   Comparison of Supplementary Material A number of books within this survey complement the printed volume with additional supplementary material, this is especially true for textbooks where five of the eight books provide additional material. Table 2.3 highlights supplementary material with each publication within this survey. The supplementary material, where available, varies from details of datasets referenced in the books to supporting videos. Supplementary material is usually provided on a dedicated website allowing for corrections to errata as they become apparent and the addition of new material. However websites can sometimes be removed with the loss of the supplementary material as with the web address referenced in *"Illuminating the Path"* [CT05].

2.2.4. Industry Professional

Seven books are identified in this category, books aimed at professionals from any industry, their covers can be seen in Figure 2.9. These differ from textbooks as they do not provide as much background important in a pedagogical context.

Show Me the Numbers: Designing Tables and Graphs to Enlighten by Few [Few12] commences with the purpose of the book which is to enable the reader to produce effective means of displaying quantitative information. The second chapter examines different types of data. The difference between quantitative and categorical data is explored along with different types of relationships, for example nominal, hierarchical and correlation. Statistical methods such as mean, median and standard deviation are also explained. Guidelines for the use of currency data are also given, numbers should be adjusted for inflation when comparing over time and differing currencies should be converted to be uniform.

Chapter three discusses the use of tables and graphs for displaying data and when to use each. Tables are used when precise, individual values are required or when more than one unit of measure is used while graphs are used to display the shape of the data or to reveal relationships. A brief history of the use of graphs is also given with references to the work of Descartes [Des37] and Playfair [Pla86]. The use of tables is also discussed in the fourth chapter. Table layouts and designs are explained for different purposes and data.

The fifth chapter discusses human perception of shapes and marks. A description is given of how the eye and brain processes images and how this influences the per-

Figure 2.9.: Book covers for industry professional books [Few12, Few06, Sim14, SH14, Kna15, Ber16, FM18] described in section 2.2.4.

ception of length, width, size, shape, colour and position as a means of displaying quantitative information. The Gestalt principles of visual perception are also discussed. Different methods of encoding data such as points, lines and colour are introduced along with different graph relationships such as time series, distributions, and geospatial in chapter six. Graph designs for each type of relationship are presented. An additional section appears at the end of the chapter with six scenarios of different types of data, where the reader is tasked with identifying the most suitable visualisation method.

Chapter seven outlines design features of a visualisation such as the layout of a figure, the use of titles, legends and labels, and the use of highlighting and emphasis. The design features specific to tables are discussed in the eighth chapter. The use of grids, white space, and fill colour to delineate columns and rows, the arrangement of the data, formatting of text and the use of summarising values are examined. Exercises asking the reader to identify limitations of tables presented are given at the end of the chapter.

Chapter nine meanwhile features general design features of graphs. The importance of quantitative scales that start at zero and of consistent scale spacing is highlighted, along with the interpretation challenges with using 3D layouts. The tenth chapter continues with a description of the format of more specific design features of graphs such as graph marks, points, lines and bars. The format and use of indicator components such as tick marks, annotations and reference lines is also explored.

Displaying multiple variables in one figure is the topic of chapter 11. This is achieved with the use of small multiples. The technicalities of the layout, sequencing, and general use of the small multiples are studied. Chapter 12 highlights graphs that should be avoided due to their inability to communicate data effectively. These include doughnut charts, radar charts and circle charts along with the reasoning behind their dismissal. The chapter is appended with exercises asking the reader to identify shortcomings with some depicted charts.

Storytelling is the focus of chapter 13. Few presents characteristics that effective stories have in common, for example, they are simple, informative and contextualised. The final chapter concludes that the reader should construct their own design standards that take inspiration from this book.

The appendix contains summaries of when to use which type of graph, recommended reading, answers to exercises from other chapters, how to calculate adjusting for inflation and instructions on how to create a table lens display and box plots in Excel.

Information Dashboard Design: The Effective Visual Communication of Data by Few [Few06] begins with a brief history of Business Intelligence (BI) data visualisation and defines an information dashboard: *"A dashboard is a visual display of the most important information needed to achieve one or more objectives; consolidated and arranged on a single screen so the information can be monitored at a glance."* Few explains that most information dashboards used in business fall short of their potential and provides a number of examples of business dashboards. In chapter two Few identifies three roles for dashboards: Strategic - that provide a quick overview of a business and focus on high-level data; Analytical - providing more detailed data that allow interaction; Operational - for monitoring activities and events that provide instant notifications. Different types of information that are typically used in dashboards are also explored.

Common mistakes in dashboard design is the focus of the third chapter. Examples from vendor websites are used to highlight design mistakes such as, choosing inappro-

priate graphics, visual designs cluttering displays with decoration, and fragmenting data into separate screens. Human perception is the topic of the next chapter. Visual memory is discussed along with data encoding and the Gestalt principles of visual perception.

The fifth chapter discusses characteristics of well designed dashboards and details techniques to producing effective visual designs, such as reducing graphics that do not contribute to the information to be displayed. Using the best type of visual layout for the information to be displayed is the focus of the sixth chapter. A number of visual layout types are given such as bullet graphs, bar charts, scatter plots, and treemaps. The use of icons, spatial maps, tables, and small multiples are also exemplified.

Chapter seven discusses designing dashboards for usability. Effective organisation of the information is considered along with making the dashboard aesthetically pleasing and maintaining consistency of colours and interaction. User testing of the design is recommended. The final chapter provides examples of dashboards in different scenarios. Examples dashboards are given for telesales, marketing, sales, and for use by a chief information officer. Eight examples of sales dashboards are also given for critique.

The Visual Organization: Data Visualization, Big Data, and the Quest for Better Decisions by Simon [Sim14] introduces the benefits of data visualisation in the modern context of big data and emphasises that most businesses do not utilise visualisation.

The first chapter discusses the changing landscape of IT. Changes in the way people interact with IT and how technology companies are adapting by creating new tools and collection data. Data visualisation tools for businesses are explored in the second chapter. Dedicated visualisation tools such as Tableau, Microsoft Excel, and visualisation tools incorporated within business intelligence software are examined. The ability of visualisation tools to integrate with databases and statistical software and their ease of use are discussed. Open-source tools such as D3 and R are also discussed.

The third chapter studies the TV and film streaming site Netflix and its use of data visualisation. The extent to which Netflix utilises the data it collects to provide insights into its customers and to improve the service it offers is examined. The online polling company Wedgies is the subject of the fourth chapter. The company's use of Google Analytics, D3, and other open-source software is highlighted.

Chapter five meanwhile features the University of Texas System and how they use SAS to visualise their data. The sixth chapter introduces four levels of organisations

categorised by the data that they visualise. The first level organisation visualises static, small scale data, the second level utilises interactive visualisations on small datasets. The third and fourth levels utilise big data sets using static and interactive techniques respectively. Progression of organisations up this scale is discussed.

Chapter seven provides an example of human resource visual design at Autodesk. A visualisation shows the personnel changes over time, produced using Java, Processing and Graph Viz. The broad outline to utilise visualisations within business is presented in the eighth chapter. Obtaining the data, processing it and utilising available tools such as Hadoop or Amazon Web Services is the first step. Considering what kind of visual layout is required for individual scenarios is an important factor along with visualisation design, and user experience. Creating a culture of data analysis and visualisation within an organisation is also important.

The penultimate chapter looks at common pitfalls for organisations that may try to utilise visualisation. Common visualisation mistakes, the importance of continuous development and tailoring to the visualisation audience are explored.

Presenting Data: How to communicate your message effectively by Swires-Hennessy [SH14] outlines some principles for using numbers when presenting numeric information, such as consistently using the same symbol for decimal separators, right justifying numbers in columns, and using a typeface that has a constant width for all digits. Details are also given on how to effectively round numbers. Presenting numeric data in tables is the subject of the second chapter. A series of principles are described for the effective use of tables such as putting totals at the bottom of columns and to the right of rows, and putting data for comparison into columns.

Chapter three discusses the use of charts. Principles are outlined for the correct use of bar charts, histograms, pie charts, and graphs. Multiple examples of charts are given along with how they could be improved, for example by ensuring axes start at zero. Numbers in text is the subject of the fourth chapter, such as in headlines or statements. Principles are outlined for their use such as ensuring the correctness of the data, using plain language, and ordering messages in terms of importance.

The final chapter discusses using the internet as a form of sharing data. Some history of national statistics data on the internet is given along with some examples of current designs such as InfoBase Cymru [Dat].
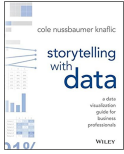
Storytelling with Data: A Data Visualization Guide for Business Professionals by Nussbaumer Knaflic [Kna15] begins with examples of inadequate visual designs. The first chapter, discusses the importance of understanding the purpose of a visual design and to consider the audience before its creation. Chapter two describes different forms of graphic visualisations such as bar graphs, scatterplots and line graphs. Examples of charts to avoid are also given such as pie charts and 3D graphs.

The third chapter discusses the issue of clutter, defined as visual elements that don't contribute to an increased understanding of data. Gestalt principles of visual perception are explored and used in an example where clutter is removed from a scatter graph. Chapter four begins by discussing human perception and memory and how these can be used to draw the attention of an audience to a particular element within a visual design.

The fifth chapter discusses the use of design principles for creating a visual layout. Key considerations are reducing clutter, highlighting important elements, avoiding overcomplexity, textual annotations, and aesthetics. Five examples of model visualisations are given in chapter six that are analysed for the design choices and that highlight the principles outlined so far in the book.

Chapter seven discusses how to build an effective story, using a beginning, a middle, and an ending. The order of the narrative of the story is also discussed. The eighth chapter emphasises principles from the previous chapters and gives an example of a series of line graphs that tell a story of how prices have changed over time.

Chapter nine addresses five situations that have not been addressed so far in the book, visualisations with a dark background, producing a static figure of an animated visualisation, ordering in categorical data, avoiding over complex line graphs, and alternatives to pie charts. The final chapter summarises the principles from the book and provides some extra tips such as learning the visualisation tool, producing draft iterations of visual designs and use of feedback, and to look at other examples of layouts.

Good Charts by Berinato [Ber16] begins with an introduction to data visualisation and how it is becoming an increasingly important skill to provide insightful and relevant designs. The first chapter details some of the history of data visualisations, from works published by Playfair [Pla86], through the work of Bertin [BBS67] and Tufte [Tuf83, Tuf90, Tuf97] to the latest research by Rensink and Baldridge [RB10] and Harrison *et al.* [HYFC14]. Berinato discusses what features tend to stand out in a visualisation, such as a peak on a line graph, and how too much information makes it difficult to find individual data points. The second chapter also discusses the importance of using common conventions, such as red is hot, blue is cold and north is up.

Four types of visualisations are identified in the third chapter: conceptual, declarative visualisations for idea illustration; conceptual, exploratory visualisations for idea generation; data-driven, exploratory visualisations for visual discovery; and data-driven declarative for presentation. The skills and tools required to produce the visual design can be determined by first identifying the type of visual design. The fourth chapter describes an hour long process for developing a visual design; the first five minutes are for preparation, the next 15 are to discuss - to establish the purpose of a visual design and what to highlight, 20 minutes are for sketching ideas, and the final 20 minutes for creating a prototype.

Refining visual layouts to make them visually appealing is the topic of the fifth chapter. Using a fixed structure, aligning elements, removing unnecessary elements, and correct use of colours are all discussed. The next chapter discusses how to emphasise particular data within a visual design to make a statement. Emphasising data using colour, labels, and markers to draw attention, removing elements that distract, adding elements that support and shifting reference points are examined.

Chapter seven features visualisation techniques that create misleading visual designs. These are the use of a truncated axes, using multiple axis on the same plane, and representing quantitative data on geographical maps. The eighth chapter examines verbal presentation of visualisations and storytelling. Guidance for presenting are given such as not to read the picture, using reference charts, use of simple charts, use of storytelling, and to avoid talking when first presenting a chart.

The final chapter encourages the reader to find published visual designs and preform a critical review of them. Three examples of critiques are provided.
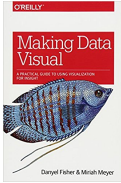
Making Data Visual by Fisher and Meyer [FM18], a new book, commence with an example data set to outline why visualisation is useful, to facilitate discovery. The second chapter focuses on developing a concrete visualisation task from a broad question about a data set, exemplified with the question "Who are the best movie directors". The question is broken down using some visually derived assumptions to actionable tasks.

Chapter three details how to interview and communicate with domain experts and stakeholders to establish what questions and assumptions can be made to create insightful visualisations of a data set. Data types are the focus of the fourth chapter with three principle types identified: continuous data, ordinal data, and categorical data. Three additional specialised data types are also identified: temporal data, geographical data, and relational data. Techniques for processing the data, such as transforming between data types and dimension reduction, are also briefly discussed.

The fifth chapter explores different chart types and explains what charts are suitable for what task and what data type. These charts include scatterplots, histograms, bar plots, node-link views, treemaps, and word clouds. Multiple linked views is the subject of the sixth chapter, which explores visualisation techniques such as small multiples, scatterplot matrices, and dashboards. Some considerations for multiple views are outlined such as consistency of scales, and ensuring interaction, such as filtering, is mirrored across views.

A case study is presented in chapter seven using a real world example that one of the authors personally worked on. The author recounts the iterative process of creating a business intelligence dashboard based on software telemetry. A second case study is outlined in the penultimate chapter focusing on visualisation of fruit fly genetics. The case study is based on another author's experience, demonstrating the approach outlined within the book and making particular use of multiple linked views.

Comparison and Recommendations of Industry Professional Books

Show Me the Numbers by Few [Few12] provides detailed descriptions of the principles of data visualisation for a beginner to get started. Unlike books classified as textbooks, the book is targeted towards professionals required to present data in industry settings. Because of the professional focus of the book, it includes presenting data using tables as well as more advanced visual designs. We recommend this book for business

professionals seeking best practice for presenting data in an industry setting. The hardcover printing of this book is quite big compared to the others. The visual designs featured are not generally interactive and are more at the beginner level.

Few's second book in this survey *Information Dashboard Design* [Few06] focuses on the design on visual dashboards. This is a unique topic within this survey, however some common subjects are also included such as perception. We would recommend this book for people developing visual interfaces who already have basic knowledge of data visualisation. Again the visual layouts are not generally very interactive and the visual designs are at the beginner's level. The book also contains a lot of white space.

The Visual Organization by Simon [Sim14] provides a discussion of the use of data visualisation in industry and how proper implementation can lead to success. The book is written in an entertaining and engaging manner. This book is suitable for a business manager or leader as a demonstration of how effective the use of data visualisation can be. This book does not contain as many images as the other books and also features some infographics themes. The examples used are at the beginner's level.

Presenting Data by Swires-Hennessy [SH14] provides formal instructions for best practice in the presentation of data. A strong focus is placed on the use of tables and spreadsheets however a section is dedicated to charts. Rules are provided for producing simple, truthful, charts. We recommend this book for business professionals who would like a very compact and concise introduction to visualisation. The visual designs are basic and generally not interactive.

Storytelling with Data by Knaflic [Kna15] provides an informal guide for creating engaging visualisations. The book is more concise than Few's *Show Me the Numbers* and provides a more creative viewpoint. Presenting visualisation aspects incrementally to give a narrative is also discussed. We recommend this book for professionals who are required to produce persuasive visualisations in their work. The example visual designs are basic and not interactive. Presentation is the focus. It's easy to read and is targeted for all industry professional backgrounds.

Good Charts by Berinato [Ber16] focuses on developing and improving visualisations skills. This book features the most images, examples, and figures of those in this category. Figures are used to demonstrate good practice and guidance is given on how to go about creating visualisation ideas. The book provides an easy to read style

and uses casual language, but also has a very awkward landscape orientation. We recommend this book to professionals seeking a wide variety of sample visual designs in a concise browsing fashion.

Making Data Visual by Fisher and Meyer [FM18] outlines a method for establishing how to develop visual designs for the visualisation of a particular data set. The book details of how to interrogate stakeholders to establish how the visual design will be used and what insights are required. The iterative process of developing a visualisation design to solve a particular problem is demonstrated with two case studies. The book is written in an easy to read, informal style and only covers topics discussed in a brief fashion without providing any complex topics.
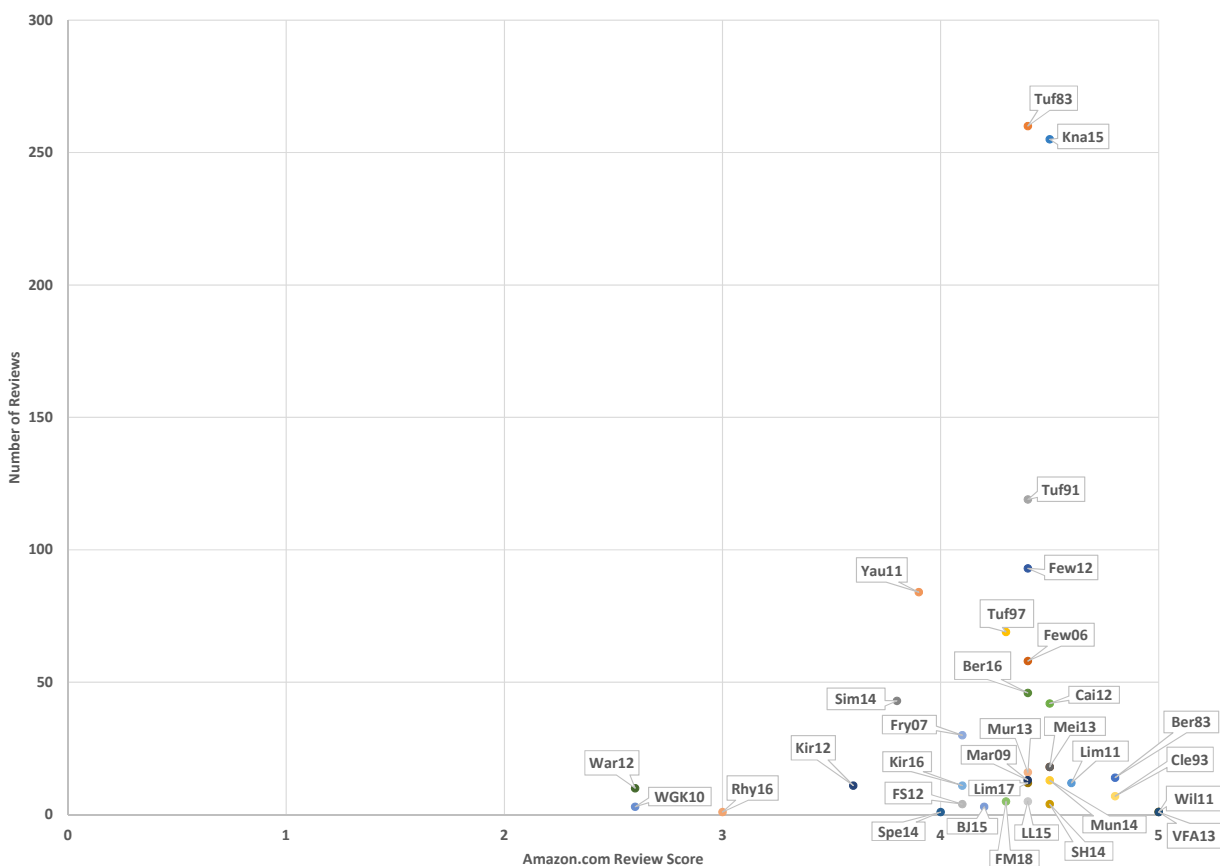
Figure 2.10.: Average review score against the number of reviews on Amazon.com, for all summarised books within the survey with a review, discussed in Comparison 4.

Comparison 4.   Comparison of Online Review Scores Figure 2.10 presents the Amazon.com review score of the summarised books within this survey. Only books

with at least one review are included in the figure, this excludes nine books from the visualisation. Notable from the image is that no books have a review score of less than three. One book, by Vathy-Fogarassy and Abonyi [VFA13], receives the maximum score of five, although it should be noted that this is from only one review. Tufte's The Visual Display of Quantitative Information [Tuf83] has the most reviews, with 270, with an average score of 4.4. Another popular book is Storytelling with Data by Knaflic [Kna15] with 226 reviews and an average score of 4.5. This is despite having a relatively recent publishing date in 2015.

2.2.5. Special Focus

Twelve books have been identified for this category which features books focused on more specific directions within data visualisation. See Figure 2.11. Due to the specialised focus of the books in this category, books generally have a lower Amazon.com sales rank, although the precise way in which the rank is calculated is unknown. The eight books with the lowest sales rank are in this category. The books within this category also tend to be the most expensive.
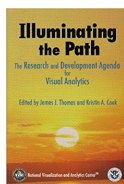
Illuminating the Path: The Research and Development Agenda for Visual Analytics by Thomas and Cook [CT05] is published as a research agenda for visual analytics as a response for the need for better data analysis tools to help counter terrorism. The book is sponsored by the US department of homeland security.

The first chapter explores potential terrorism threats and the challenges of analysing vast amounts of data to counter the threat, while still allowing for personal freedom. The need for advancements in visual analytics to address these challenges is highlighted. Chapter two discusses the use of visual analytics as a tool to support analytical reasoning. The process of analytical reasoning is explored, and the inclusion of visual analytics to facilitate perception and understanding of data and to enable rapid decision making.

Visual representation of data and interaction is discussed in the third chapter. Scientific principles for data visualisation are explored along with interaction methods. The use of the appropriate visual design for specific datasets and queries are considered along with their scalability with increasing data. The fourth chapter discusses data and data processing from raw format to representations that may provide more insight.

Figure 2.11.: Book covers for books classified as special focus [CT05, Mar09, Ham08, AMST11, Wil11, DKŽ12, VFA13, CLS14, BJ15, Rhy16, CFV*16] described in section 2.2.5.

Different forms of data are discussed such as video, image, text, and geospatial data along with their characteristics.

Chapter five highlights the importance of accurate communication of visual analytic results. Existing tools are reviewed and the automation of effective visualisation generation is discussed. A framework for moving visual analytics research into practi-

cal use is presented in the sixth chapter. Four areas are identified as requiring attention: evaluation – to address if a new tool or technique is effective; privacy and security – to support approaches such as anonymisation to protect privacy; system interoperability – to avoid incompatibility with other tools; and technology insertion – the widespread adoption of the tool or technique.

The final chapter discusses the requirements and need for meeting the agenda set out in the book.

Interactive Visualization of Large Networks: Technical challenges and practical applications by Van Ham [Ham08] begins with a chapter outlining the objective of the literature, how computer supported visualisations can be used to help users gain insight into the content and structure of large networks. The second chapter defines a simple graph, a series of nodes connected by edges, and different forms of graphs. Considerations are made from the aim of a user when visualising a graph to what constraints may exist to the creation of a visualisation.

Previous work to design graphs is outlined in the third chapter. This is subdivided into three topics, graph drawing, clustering and visual techniques such as treemaps and interaction. Chapter four acts as an introduction to the next four chapters, outlining four different graph problems for which visual designs are created.

The fifth chapter introduces Beamtrees [VHVW03], a visualisation based on treemaps that uses perceptual cues to clarify its structure. Details of how to construct a Beamtree and how to calculate the size of each node are presented, along with a user evaluation of the method. A method for visualising state transition graphs is presented in the sixth chapter. The technique relies on clustering states to enable the visualisation of large numbers of nodes that provides a global view of the structure of the graph.

A visual layout that enables interactive visualisation of a large graph structure with hierarchy is presented in the seventh chapter. An example of the exploration of a software engineering project is used as an example throughout the chapter. The eighth chapter presents a method for conveying small world graphs. The layout utilises clustering to minimise clutter on a high abstraction level, while focusing on a particular cluster allows for detail to be seen.

The penultimate chapter reflects on each of the four projects outlined in the previous chapters, comparing their flexibility and scalability. Common aspects between the four visualisations are discussed and insights gained from the projects are presented.
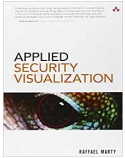
Applied Security Visualization by Marty [Mar09] introduces data visualisation and its benefits in particular within the computer security sector. Perception is briefly discussed along with some graph design principles such as annotating data and showing comparisons. The second chapter discusses security data sources, predominantly log files, the parsing of the data and potential issues such as missing log entries. Specific security data sources are discussed and how to retrieve them such as network traffic, firewall logs and operating system logs.

Methods of visually representing the data are discussed in the third chapter including the use of colour, shape and size. A number of visual designs are described, for example bar charts, line charts, treemaps, node link diagrams and parallel coordinates, and the kind of data each can display. The use of 3D charts is also discussed along with pitfalls such as occlusion and the use of interaction and animation to overcome these. Chapter four introduces the visualisation pipeline as six steps. The six steps are: define the problem, assess the data, process the data, visual transformation, view transformation and refine and iterate. An example of web traffic analysis is used to demonstrate the process. A brief description of data processing tools is also presented.

The fifth chapter applies the principles from the previous chapters to analyse security data. Three broad categories of analysis are identified, reporting, historical analysis and real-time monitoring. Time-based layouts, interaction and forensic analysis are discussed along with examples meanwhile discussion about real-time monitoring is predominantly centred around data dashboards. Examples analysing threats to network perimeters are given in chapter six. Examples include traffic flow analysis using parallel coordinates and vulnerability visual designs using treemaps.

The seventh chapter discusses the use of visualisation for compliance such as with auditing, compliance monitoring and visualising risk. The penultimate chapter discusses the use of visualisation for detecting threats from inside a network, in particular for showing user activity.

The final chapter presents a range of visualisation tools.
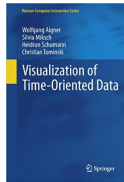
Visualization of Time-Oriented Data by Aigner *et al.* [AMST11] begins with an introduction to time-oriented data along with the special associated challenges such as recognising periodic cycles. A formal introduction to visualisation is also given with academic references describing the process of creating an interactive visualisation. An application example is given of VisuExplore, an interactive medical history visualisation tool [RMA*10]. Historical depictions of time-oriented data are explored in the second chapter such as William Playfair's work [Pla86]. Time in storytelling is also discussed such as the use of comic strips and movies as well as music and paintings.

Chapter three characterises time by its scale, scope and arrangement and also looks at the types of data that can be represented. The considerations of creating a visualisation from data are discussed in the fourth chapter by asking what, why and how data is presented. The visual representation of time is explored along with representation of data, both spatial and abstract, and how to encode both.

Different forms of interaction are first identified such as selecting and filtering before some interaction principles are outlined in chapter five. Methods of interacting are discussed such as direct manipulation and brushing, as well as automatic methods such as event-based visualisation. Analytical techniques for data mining are discussed in the sixth chapter. Analysis tasks for time-oriented data are for classification, clustering, search & retrieval, pattern discovery and prediction. Data abstraction is discussed along with principle component analysis.

Chapter seven presents a survey of over 100 visualisation techniques, categorised by data characteristics. Techniques range from simple point plots to visualisation techniques developed for a specialist field. The final chapter presents future research challenges for time oriented data.

Visualizing Time: Designing Graphical Representations for Statistical Data by Wills [Wil11] begins by emphasising the importance of time and of creating visual designs for it, citing Stonehenge as an example. William Playfair's work [Pla86] and Minard's figure of Napoleon's march on Russia [Min69] are other historical works of time-oriented layouts that are studied, along with comic book style depictions. The second chapter introduces visualisation concepts, briefly describing visual primitives, data processing steps, aesthetic considerations, mapping data, and interaction concepts.

The process taken to design a visualisation is the subject of the third chapter. Wills introduces the Goal Question Metric [VSBCR02] approach to visualisation design; first the user goal needs to be established, questions are then formulated, the answers to which will achieve the goals, finally the data is mapped to the visual primitives. Chapter four discusses different formats of date and time, time intervals and different scales of measurement.

Mapping time to an axis is the topic of the fifth chapter. As time is an independent variable it is usually placed on the horizontal axis. Techniques for plotting events and sequenced data are discussed. The sixth chapter discusses using non-Cartesian coordinate systems, labelling time to an axis, and using small multiples.

Mapping time to another attribute such as colour or size is the subject of the seventh chapter. An example dataset of medieval soldier activity is used to demonstrate the different techniques. A further example is used to demonstrate ordering within text-based data. Chapter eight discusses the distortion of the time axis. This enables for better representation of some time-based data enabling emphasis of a particular element. Techniques discussed include changing from a linear time scale to an ordered time scale and using frequency analysis.

Ways of interacting are discussed in chapter nine. Modifying different parameters of such as axes, mapping primitives, coordinates, and scales are explored along with brushing techniques. Chapter ten exemplifies problems encountered with specific datasets. Methods for dealing with large datasets such as aggregation are examined along with techniques for visualising timelines and linked events.

The final chapter discusses the complexity of visualisations and a formula is introduced to derive and quantify complexity based on the number of variants and primitives mapped. All figures used in the book are then ranked according to the formula.
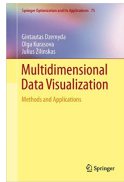
Multidimensional Data Visualization: Methods and Applications by Dzemyda *et al.* [DKŽ12] defines what data is considered multidimensional, i.e. a dataset with multiple features. Two method groups of multidimensional data visualisation are identified, direct visualisation methods – the plotting of each feature; and projection – where dimensional reduction techniques are used. Chapter two describes in further detail the visualisation methods mentioned in the previous chapter. Direct visualisation methods include geometric visualisations such as scatter-plots, multiline graphs, parallel coordinates and RadViz; glyph display methods such as Chernoff faces and star plots; and hierarchical displays methods of dimensional stacking and trellis display. A number of dimension reduction methods are also presented including principle component analysis, linear discriminant analysis, multidimensional scaling, manifold based visualisation, and local linear embedding.

The third chapter explores multidimensional scaling in further detail. Multidimensional scaling describes a range of techniques that reduce the dimensionality of a dataset with the aim of retaining the dissimilarity between any two objects. Different methods of optimising the dissimilarity measure are investigated. The concept of artificial neural networks is explained and the use of a self organising map, neural gas and SAMANN networks with multidimensional scaling is discussed in chapter four.

The final chapter provides examples of the application of the techniques discussed in the book across many different fields.
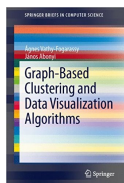
Graph-Based Clustering and Data Visualization Algorithms by Vathy-Fogarassy and Abonyi [VFA13] commences with an examination of vector quantisation algorithms that can be used to convert complex data into connected graphs. Techniques explained include growing neural gas vector quantisation, dynamic topology representing network and the weighted incremental neural network. The second chapter discusses two graph-based clustering algorithms and introduces novel techniques to improve their performance. The first clustering algorithm discussed is the minimal spanning tree method. Vathy-Fogarassy and Abonyi introduce a modification by removing inconsistent edges and utilise the Gath-Geva algorithm [GG89] to cluster the results. The resultant algorithm is titled the Hybrid minimal spanning tree–Gath-Geva algorithm and utilises a fuzzy hyper volume validity measure.

The final chapter provides an overview of algorithms that unfold and represent the topology of data, in particular the Isomap, Isotop and Curvilinear Distance Analysis

methods. An algorithm that combines the topology representing methods and multidimensional scaling is studied and analysed. The appendix provides details of the various algorithms described throughout the book.
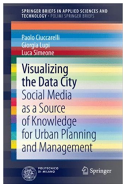
Visualizing the Data City: Social Media as a Source of Knowledge for Urban Planning and Management by Ciuccarelli *et al.* [CLS14] researches and analyses geo-located social media data mining from an urban development standpoint. The second chapter emphasises the possibilities of using geo-located social media to characterise urban uses and issues. Some existing projects are reviewed, and an aim of producing a platform that can visualise the data is proposed.

Chapter three presents a taxonomy of state-of-the-art geo-located data visualisation techniques that deal with large datasets and with real-time interaction. Some 56 projects are analysed and categorised according to the data topic and by project goals. The fourth chapter describes how the authors created a framework for investigating urban interest topics using geo-located social media data. A discussion of the evaluation of the work is also given.

Telltale and Urban Sensing, two projects that visualise geo-located social media data, are presented in the penultimate chapter. Examples of visual designs and data analysis performed using the two projects are presented, such as using twitter to track individual movements, and mapping cultural distribution through the language used in social media. The final chapter presents a discussion on the potential of using geo-located social media analysis and its shortcomings.
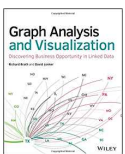
Graph Analysis and Visualization: Discovering Business Opportunity in Linked Data by Brath and Jonker [BJ15] first describes the importance of data visualisation with particular emphasis of the use of graph-based visual designs. Multiple historic examples are given of the use of graph visualisations and the benefits these visual designs provided, such as managing network supply chains and mapping social hierarchies. The second chapter identifies different forms of graphs and their uses in representing relationships, hierarchies, communities, flows and spatial networks.

Sources of data to be visualised are identified, such as flight stats or Tweets, and methods of obtaining the data are discussed in chapter three. Processing the data so that only relevant information is included and file storage methods and formats are discussed such that information can be easily imported into graphing software.

Chapter four discusses meta-data of graphs as the number of nodes and edges, and centrality as an indication of the types of relationship in the graph. Based on this understanding and the objective, different graph layouts are also outlined.

The fifth chapter discusses mapping different visual attributes to a graph. The use of colour, labels, font, line weight and size is discussed along with the merits of the use of each. Interaction techniques such as zooming and filtering are explained and the use of legends, annotations and sequences to help communicate insights are studied in chapter six.

Chapter seven provides an overview of the capabilities of five different graph creating tools, namely Excel, NodeXL, Gephi, Cytoscape and yEd. Meanwhile chapter eight describes how graph layouts can be constructed using the Python environment and the JavaScript environment using the D3 libraries. See section 2.2.6.

The ninth chapter discusses the relationship connections within a graph, when they should be kept separate and their aggregation. Applications of changing the connections are provided such as actor relations across films. Hierarchies are discussed in the tenth chapter, represented as trees and treemaps.

Techniques for identifying and analysing clusters within graphs are discussed in chapter 11. An example of a Facebook discussion about the Toronto Raptors NBA team is used to demonstrate some of the techniques such as identifying cliques. Chapter 12 discusses flow based diagrams such as Sankey diagrams and chord diagrams and gives detailed instructions on how to create an example of each using JavaScript and the D3 library.

The 13th chapter explores spatial networks, where where physical characteristics need to be preserved. The London Underground network map is given as an example. Rose link diagrams and grouping of small worlds are also explored. Chapter 14 discusses issues encountered when dealing with big data. The use of database and query languages for exploration of structures within the data is discussed and detailed examples of subtracting data for visualisation in Gephi is given. The use of tiled approaches for visualising scalable large datasets is also discussed.

Visualising graphs that change with time is discussed in chapter 15. Different techniques for achieving this are discussed including the use of animation, small multiples, focus+context and an additional spatial dimension. The final chapter studies design choices for a graphs such as the shape and size of nodes and links, and the use of iconography. The use of an appropriate colour palette is also examined.
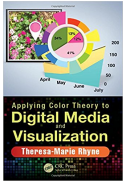
Applying Color Theory to Digital Media and Visualization by Rhyne [Rhy16] introduces the three colour models, the RGB model of light and display, the CMYK model for printing, and the RYB model for paint. A history of the development of colour theory and an example of colour theory application is also given. The second chapter details how human vision perceives colour and luminosity, and discusses various colour vision deficiencies and how to apply colour theory to compensate for this. Problems with using a rainbow colour map are also highlighted.

The colour gamut and the colour spaces that are produced from different displays and media are discussed in the third chapter. Various colour spaces are explained such as the sRGB, and the HSV colour spaces. colour harmony, defined as the colours that work well together in the composition of an image, is the topic of the fourth chapter. Colour wheels for the colour models are reviewed and colour harmonies between hues on the wheels are examined.

Chapter five reviews eight different online and mobile tools for producing colour schemes. These tools include ColorBrewer 2.0 [HB03], Adobe Color CC [Ado18], and PANTONE Studio [Pan18]. The sixth chapter provides three case studies on the use of colour theory to produce colour schemes for visual designs. The case studies consist of a biological dataset, a geospatial dataset, and a hurricane animation.
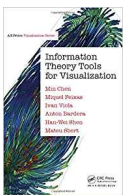
Information Theory Tools for Visualization by Chen *et al.* [CFV*16] begins by describing basic concepts of information theory, such as entropy, conditional entropy, and mutual information. The second chapter details how information theory can be applied to data visualisation, as the communication of information from data, through visual representation, to perception.

The application of information theory in determining the best viewpoints for scientific visualisations is demonstrated in the third chapter. Further examples of the use of information theory within volume visualisations are given in chapter four. Example problems include transfer function design in direct volume rendering and isosurface extraction.

Chapter five provides examples of the use of information theory in flow visualisation. The concept of an entropy field is introduced and applications of its use for streamline analysis and seeding. The final chapter discusses the use of information theory in information visualisation. Among the applications of information theory
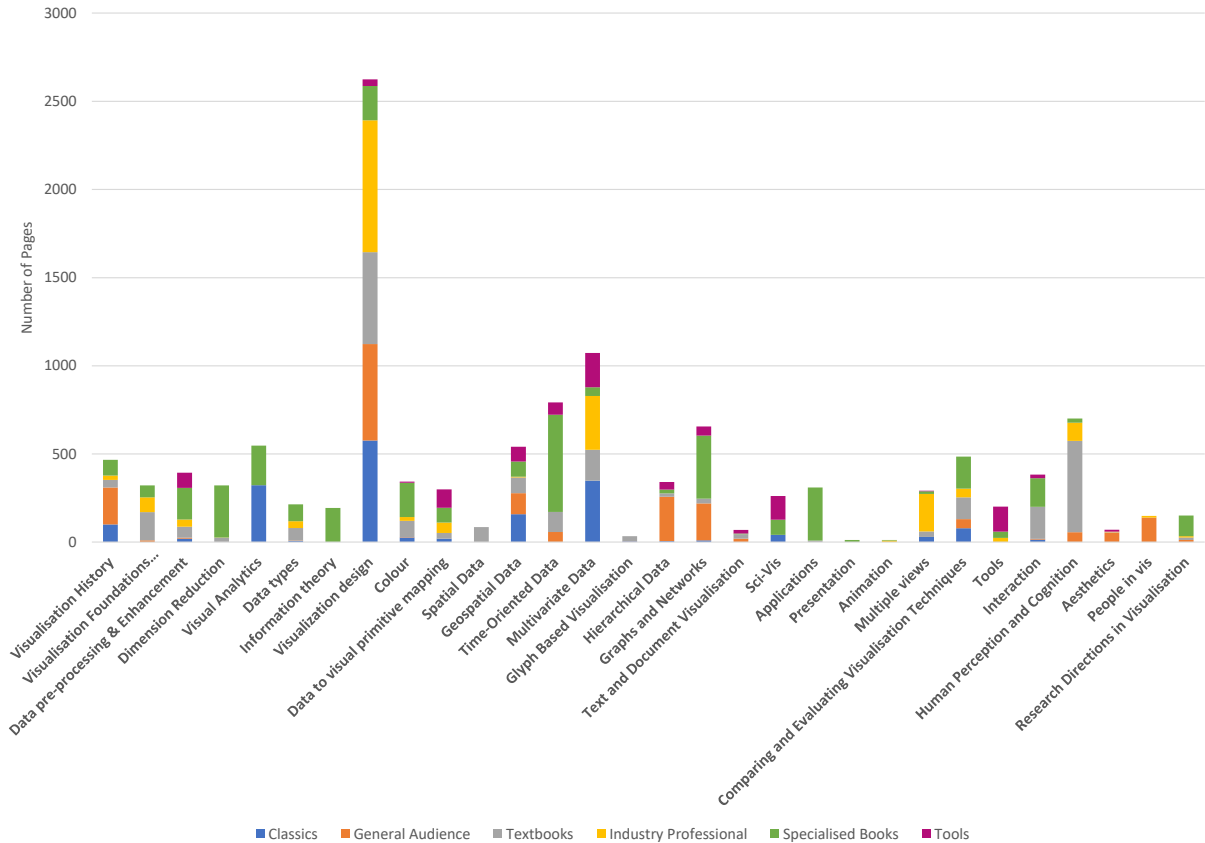
Figure 2.12.: Total number of pages, from surveyed books, dedicated to each topic, coloured by book classification. Analysis of this Figure can be found in Comparison 5.

are the definition of a quality metric of visualisations and measures to analyse the relationships of variables in multivariate datasets.
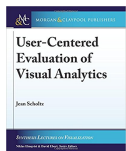
User-Centered Evaluation of Visual Analytics by Jean Scholtz [Sch17] commences with an outline of the objectives of the book, to provide assistance to researchers in visual analytics on conducting formal user-centred evaluations. The second chapter provides a brief history of intelligence analysis in the United States of America as well as a discussion on critical thinking.

Analytical methods are discussed in chapter three, in particular the activities involved with sensemaking. Chapter four describes why visual analytics is needed, citing Thomas and Cook's Illuminating the Path [CT05]. Some visual analytics tools such as Tableau are also discussed.

The evaluation of the ease of use of software is discussed in the fifth chapter, and

how it can be used with visual analytics tools. Different Human-Computer Interaction (HCI) evaluation methods are also presented such as the use of guidelines, usability studies and A/B studies. The sixth chapter discusses the HCI evaluation needs specific to the visual analytics field. Modifications to the HCI methods presented in the previous chapter are presented, that make them more applicable to visual analytics. The use of metrics to measure effectiveness of tools is also discussed.

Chapter seven examines the current use of evaluation in visual analytics systems, and cites work by Isenberg *et al.* [IIC*13] to determine the use of human participants in evaluations of published conference proceedings. Proposals by Sedlmair *et al.* [SMM12] of a methodology of developing visualisation software along with evaluation of its use, are also cited. The penultimate chapter identifies trends in the visual analytics field in particular in collaborative work and in the use of streaming data.

This book is the most recent book in the Synthesis Lectures on Visualization series, another seven books are available in this series which we only reference in the interest of content management [FGKR16, End16, SP16, Hur15, Tom15, Mac11, LM10].

Comparison and Recommendations of Special Focus Books

Due to the varying subject matter of the books within this category, comparisons and recommendations are not suitable across very different topics. However three books categorised in this section discuss network visualisation, and another two books discuss visualisation of time therefore it is logical to compare these.

Van Ham's *Interactive Visualization of Large Networks* [Ham08] is a published PhD thesis therefore is recommended for experienced data visualisationists and researchers. Similarly Vathy-Fogarassy and Abonyi's *Graph Based Clustering and Data Visualization Algorithms* [VFA13] features published research work and is therefore recommended for researchers. This book is part of the Springer Briefs in computer science series and is thus printed in black and white. Both books discuss different subject matters within network visualisation, with Van Ham's work concentrating on visualising network structure and Vathy-Fogarassy and Abonyi's publication focusing on clustering algorithms.

Graph Analysis and Visualization by Brath provides a beginners guide to using graphs to analyse relationships between data. The book is aimed at analysts and data scientists, with many examples taken from a business environment. Network

visualisation tools are explored including some coding examples using Python and D3. We would recommend this book to beginners in data visualisation with linked data to visualise and also those that would like a comprehensive introduction to graph analysis and visualisation. It is also more applied in nature than the other two graph books.

Both Wills' *Visualizing Time* and Aigner *et al.'s Visualization of Time-Oriented Data* examine the visualisation of time with many overlapping topics between the two books such as interaction. Wills' book however has greater emphasis on creating a visual design with data mapping being a strong subject. Meanwhile Aigner *et al.* provide greater prominence to the use of visualisation for data mining and analysis. Aigner *et al.* also present over 100 examples of time visualisations categorised by data characteristics, useful for inspiration. We recommend Wills' book for beginners requiring to visualise time. For people requiring visualisations for data analytics purposes, we recommend the book by Aigner *et al*, and for individuals creating visualisations for presentation and storytelling we recommend the book by Wills. Aigner *et al.'s* book is more appropriate for researchers undertaking time-oriented visualisation and provides a survey of the research landscape.

Comparison 5. Comparison of Topics Across Books Figure 2.12 indicates the number of pages dedicated to each topic from all of the focus books within this survey. Notable is the number of pages dedicated to the visual design of a visualisation, which is spread over most classifications of books apart from tools. The topics with the fewest dedicated pages are glyph based visualisation, presentation and animation. Books classified for general audiences and classics predominantly discuss the same topics, where as textbooks and special focus books focus on specialist topics.

2.2.6. Information Visualisation Tools

Books in this category are instructional books for specific tools. Due to the large number of instructional books for software tools, only one recent book for each tool is reviewed whilst other related books are referenced. The books are summarised for this category cover Processing, D3, and OpenGL, see Figure 2.13.

Figure 2.13.: Book covers for books classified as tools [Fry07, Mur13, LL15] and described in section 2.2.6.

Processing

The Processing language is the first visualisation tool that we survey. In order to keep the survey succinct, we only survey Fry's Visualizing Data: Exploring and Explaining Data with the Processing Environment [Fry07]. Many other books that present information on Processing environment are available, we list five recent ones in Table 2.4 along with meta-data.
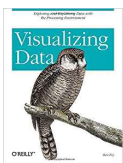
 Visualizing Data: Exploring and Explaining Data with the Processing Environment by Fry [Fry07] identifies seven stages for visualising data; these are Acquire, Parse, Filter, Mine, Represent, Refine and Interact. An example dataset of zip code data is used to demonstrate the seven stages. The second chapter introduces Processing, a programming language for creating visual designs. Some basic commands for using Processing are given such as drawing primitives, loading data and mouse interactions.

Mapping data to visual space is the topic of chapter three. Detailed Processing code of an example of placing data on a geographical map is given with data attributes mapped to size and colour. The next chapter focuses on visualising data that changes over time. An example data set is used to demonstrate importing data into Processing and creating a labelled scatter graph, line graph and bar chart.

Chapter five uses baseball data from MLB.com to create a visual design. Detailed Processing code is given on how to read and process the data and to create a connec-

tion to the data. A Perl script is also given to enable for the image to be shown in a web browser. An example of showing US zip codes on a geographical map is used in chapter six. The seven stages identified in chapter one are followed and Processing code is given for the various stages to create interactive visual layouts.

The seventh chapter discusses using recursion programming in Processing to create hierarchical visualisations such as treemaps. An example is given with Processing code that creates a Treemap of the file structure of the PC. The eighth chapter provides Processing code to create graphs. The integration of Processing with the Eclipse IDE is also demonstrated to create an interactive graph diagram of website traffic.

Chapter nine presents various techniques of acquiring and preprocessing data, including storing data in databases. Parsing Data is the subject of the tenth and penultimate chapter. Different formats of input files are discussed along with techniques of extracting the most relevant data. The final chapter serves as a Process reference for integration with the Java programming language.

Ref	Number of pages	Amazon Sales Rank	Google Scholar Citations	Price (USD)
[RF07]	672	98,445	502	59.66
[Shi09]	564	51,476	93	41.06
[BGLL12]	472	181,654	44	50.63
[RF15]	238	39,897	104	16.62
[NN17]	576	1,813,622	0	59.24

Table 2.4.: Meta-data for five recent Processing books

D3

D3 [BOH11] is a JavaScript library for producing web based data visualisations. We found 23 books that introduce and describe how to use the D3 library, five recent examples can be found in Table 2.5. We survey one recent example for this chapter [Mur13].
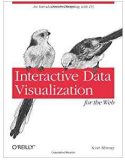
Interactive Data Visualization for the Web by Scott Murray [Mur13] begins with a compilation of tutorials for beginners in D3. In the second chapter Murray introduces the JavaScript library D3, its purpose, its benefits, and its limitations. Alternative tools are also described.

Chapter three briefly describes the technologies behind web hosting and offers a beginners guide to HTML, CSS, JavaScript and SVG. The fourth chapter describes how to setup D3 on a PC or Mac in order to begin coding.

The process of importing data into D3 is described in chapter five. Details are also given on how to check that the data has been properly read, and how to load the data into functions. Chapter six provides detailed code and explains how to draw basic bar graphs, bubble plots, and scatterplots using D3 and SVG. The inclusion of labels is also described.

The details of how to scale data is described in the seventh chapter. The use of the d3.min() and d3.max() functions are described in order to establish the range of the data and therefore establishing the scale required. Alternative methods of scaling, such as logarithmic are briefly described. Chapter eight describes how to add axes to a scatter plot.

The ninth chapter describes how to update visual designs already created with updated data. Details are also given on how to effect the transition of the visualisation from one dataset to another, and how to add additional data and remove data from a visual layout. Interactivity, the tenth chapter guides the reader on how to highlight elements when hovering over with a mouse and how to sort elements. The application of tooltips is also described.

Chapter 11 examines different layouts in D3. Pie, stack, and force layouts are described along with their application. The 12th chapter looks at creating geospatial visualisations using D3. Importing GeoJSON files and mapping data to geographical elements are explained.

The final chapter explores the options for exporting results. Three methods are described, using print screen, printing to a .pdf file, and saving code as a .svg file.

OpenGL

OpenGL is a cross-platform API for rendering computer graphics, first released in 1992, and is used extensively across multiple disciplines within computer science. Due

Ref	Number of pages	Amazon Sales Rank	Google Scholar Citations	Price (USD)
[Zhu13]	338	1,029,905	27	44.99
[Nel14]	316	2,812,251	1	25.69
[Kin14]	288	888,699	5	27.64
[Mee15]	352	169,709	5	31.13
[RT16]	272	1,843,691	0	39.97

Table 2.5.: Meta-data for five recent, related D3 books

to the maturity and the broad adoption of the technology, many books exist on the tool. An Amazon.com search for *'OpenGL'* returns over 500 results. We summarise a recent book by Lo and Lo [LL15] for this survey, which has a focus on data visualisation. We cite another five examples of OpenGL books in Table 2.6 with accompanying meta-data.
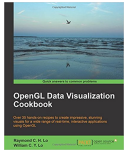
 The OpenGL Data Visualization Cookbook by Lo and Lo [LL15] opens with an introduction to OpenGL and provides details on how to set up an OpenGL programming environment on Windows, OS X and Linux. The coding and compiling of a basic program is also detailed. The second chapter demonstrates drawing simple primitives of points, lines and triangles. An example of drawing an electrocardiogram trace is used and a 3D Gaussian function example with height matched to colour is also demonstrated.

Rendering 3D images is the focus of the third chapter. The camera perspective is discussed and how to set it up, and an example of visualising a 3D Gaussian is given. Details on how to interact with the rendering are also given, with code on how to move the perspective on the model. An example of code for volume rendering is given to finish the chapter. Visualising images and videos are introduced in the fourth chapter. To do this, programming using shaders is introduced, along with textures and the OpenCV library [Bra00] for videos.

Chapter five builds on what has been introduced so far to render 3D depth fields captured using the Microsoft Kinect [Mic13]. The sixth chapter discusses rendering stereoscopic 3D models and using the OpenAsset Import Library (Assimp) [SGK*12] to import 3D models.

Using OpenGL for Embedded Systems (OpenGL ES) to render on Android devices

is explained in the seventh chapter. Detailed instructions on how to setup the framework required, to create the code and to run the code on an Android phone are given. Chapter eight continues from the previous chapter with the addition of real-time rendering of information recorded using the phone sensors to create an interactive demo.

The final chapter details how to create an augmented reality based visual design for mobile and wearable platforms. This builds on the platform developed in the previous two chapters with the addition of the OpenCV library to enable live video capture, and the use of textures to render the video.

Ref	Number of pages	Amazon Sales Rank	Google Scholar Citations	Price (USD)
[Mov13]	326	894,921	9	49.99
[KSS16]	976	362,336	1070	47.29
[Wol11]	394	722,147	43	49.59
[WJHSL15]	880	330,601	482	45.40
[AS11]	768	210,517	3	55.99

Table 2.6.: Meta-data of five recent, related OpenGL books

Other Tools Microsoft Excel is propitiatory software therefore is not included in the survey. However its ubiquitousness merits some inclusion, therefore an Excel book review is included in the supplementary material.

We also note that there are a large number of books on R [Wic16], a useful visualisation tool. It is not covered here due to space limitations. We are generally interested in interactive tools.

We present the tools mentioned in each book in Table 2.7. Tools where explicit instruction guides or code is given are marked in bold, enabling the quick discovery of guidance on a particular software technology.

2.3. Discussion

By analysing Table 2.1 and Figure 2.12 we can begin to see some patterns in the contents of the books with the classification we have applied. The books we have classified as

'classics' have limited pages dedicated to data analysis topics, and to human factor topics. Books classified as 'general audience', 'industry professional' and 'tools' also have few pages dedicated to data analysis topics. 'Industry professional' classified books also do not, in general, discuss topics classified as visualisation techniques.

The most popular topic from all of the books surveyed is the 'visualisation design' category with most books having pages related to the topic. Other topics such as 'information theory' only have one book, which is wholly dedicated to the topic. Other popular topics are multivariate data, time-oriented data, and human perception and cognition.

Topics such as presentation, animation and sci-vis have a low number of pages dedicated to them, however these are covered in books from outside the data visualisation field with presentation covered in infographics books, animation covered in graphics and sci-vis in its own topic.

It is worth noting that there is no scientific method of analysing the topics discussed in each book and that the categorisation used is a subjective matter.

2.3.1. Survey Contribution

Although this chapter does not provide background information on the themes of this thesis, it does provide a broader benefit for the work presented throughout the thesis.

Methods for determining user requirements and for developing visual designs are presented by a number of books. The *Five Design-Sheets* book by *Roberts et al.* [RHR17] provides a method for developing software visualisation tools by prototyping sketch based versions. Spence's book *Information Visualization* [Spe14] provides descriptions of interaction methods and introduces Norman's action cycle as a method for developing techniques for interaction. Spence also outlines eight explicit steps for developing interactive visualisations from commission to the final product. Another book that provides a method for developing visualisation tools is *Visualization Analysis and Design* by Munzner [Mun14] and yet another again by Kirk [Kir16]. The explicit techniques from any one of these books were not used in developing the visualisations and tools presented in this thesis, however the combined knowledge garnered from these books was used to interact with QPC Limited and to develop the visualisations. These books

also outline the appropriate visual design for different forms of information to be displayed.

Other important factors learned from the compilation of this survey include the correct labelling and the correct use of axis scales so not to mislead the reader. These factors are introduced in a number of books within this survey [Tuf83, Few12, SH14, Kna15, Ber16].

A particular book that influenced the colourmap choices of the tools was *Applying color theory to digital media and visualization* by Rhyne [Rhy16]. Of distinct benefit were the tools highlighted for selecting appropriate colour schemes.

As tools were developed using OpenGL, the methods demonstrated by Lo and Lo were useful as a reference [LL15]. A specific book that was continuously used was the *OpenGL SuperBible* by Wright *et al.* which provided many technical solutions throughout the development of the tools.

By compiling this survey, a substantial repository of knowledge was gained that benefited the execution of this thesis.

Title	Microsoft Excel	Tableau	Google Tools	D3	Processing	Adobe Illustrator	Other Tools
Visual Complexity [Lim11]	X			X			Apple Numbers
The Functional Art [Cai12]	X	X					
Data Visualization [Kir12]	X	X	X	X	X	X	Adobe After Effects, Adobe Flash, Arc GIS, CartoDB, closr.it, DataWrapper, Geocommons, Gephi, Google Chart Tools, Google Fusion Tables, Google Visualization API, Grapheur, InDesign, Indiemapper, Inkscape, Instant Atlas, Kartograph, KendoUI, KeyLines, Keynoe, Leaflet, Maya 3D, Microsoft Excel, Microsoft PowerPoint, Miso, Nodebox, Mondrain, OpenStreetMap, Panopticon, Paper.js, Polychart, Polymaps, QlikView, Quadrigram, R, Raphael, Tableau, TIBCO Spotfire, TileMil, WebGL, Wolfram Mathematica, Wordle, zoom.it
Design for Information [Mei13]							ColorBrewer, GLEAMviz.org, ManyEyes, Wordle
Information Visualization: An Introduction [Spe14]							Analyst's Notebook, IBM i2, MIND, Spotfire, WireVis
Information Visualization: Perception for Design [War12]							ColorBrewer
Interactive Data Visualization [WGK15]		X		X	12		ArcGIS, AVS, Circos, ERADS IMAGINE, GGobi, GraphViz, GRASS, InfoScope, Jigsaw, OpenDX, Prefuse, QGIS, SAS, SCIRun, Spotfire, SPSS, Tom Sawyer Software, VTK, Weave, XmdvTool
Visualize This [Yau11]	X	X	X		X	17	**Python (24), R (77), Protovis (11), Adobe Flash Builder (23)** ArcGIS, ColorBrewer, Inkspace, Many Eyes, Modest Maps, Mr. Data Converter, PHP, Polymaps, R
Interactive Visualization [FS12]	19	X	X		X	X	Adobe Flash, Adobe Flex, Apple iWork, ArcGIS, DataWrangler, DBpedia, FOAF, Gephi, Java, JavaScript, Juxta, ManyEyes, Microsoft Word, NodeXL, OpenOffice, OWL, PHP, Prefuse, Protovis, Python, Ruby, SPARQL, VIDI, VisualEyes, WinPure, Wordle
Visualization Analysis and Design [Mun14]		X					ColorBrewer, Constellation, Graph-Theoretic Scagnostics, InterRing, PivotGraph, the Hierarchical Clustering Explorer, VisDB
Data Visualisation [Kir16]	X	X	X	X		X	Microsoft Word
Show Me the Numbers [Few12]	7						
The Visual Organization [Sim14]	X	X	X	X	X		Amazon Redshift, BrioQuery, Business Objects, Cognos, Crystal Reports, DataHero, Gephi, Graph Viz, Hadoop, Hive, Impromptu, Jaspersoft, Lawson Business Intelligence, ManyEyes, MapReduce, Microsoft SQL Reporting Services, MicroStrategy, Pentaho, PIG, PowerPlay, QlikView, R, Reporter-Smith, SAP, SAS, SPSS, Teradata, Visually
Presenting Data [SH14]	X						
Storytelling with Data [Kna15]	X	X	X	X	X	X	Microsoft PowerPoint, Python, R
Good Charts [Ber16]	X	X	X	X		X	Apple Numbers, Chartbuilder, Datawrapper, Highcharts, Inogr.am, Plot.ly, Qlik Sense, Quadrigram, Raw, Silk, Vizable
Making Data Visual [FM18]	X	X	X	X	X	X	**Python(S), Vega(S), VegaLite(S)** ggplot, Microsoft Virtual Earth, MizBee, PowerBI, R
Illuminating the Path [CT05]							AVS software, Comand Post of the Future (CPOF), GeoTime, ILOG library, IN-SPIRE, InfoVisToolkit, Microsoft PowerPoint, Oculus Info Sandbox, Office, OpenDx, POLESTAR, Rivet, SeeSoft, Spotfire, ThemeView, TreeJuxtaposer, Visual Insights ADVIZOR, Word, XmdvTool
Interactive Visualization of Large Networks [Ham08]							Dot, GEM, MGV, Narcissus, Neato, NicheWorks, SemNet
Applied Security Visualization [Mar09]	X	X	X				**Argus(4), awk(1), NetFlow(1), NfDump(1), Perl(2), sed(1)** Advizor, AfterGlow, C#, C++, Crystal Reports, Cyptoscape, Dotty and Ineato, EtherApe, ggobi, glTail, gnuplot, GraphViz, grep, GUESS, InetVis, Java, Large Graph Layout, Many Eyes, Mathamatica, MatLab, Mondrian, MRTG/RRD, Notepad, NVisionIP, Oculus, OpenOffice, Parvis, Ploticus, R, Real Time 3D Graph Visualizer, Rumint, Shoki, SpotFire, StarLight, Swivel, TimeSearcher, TNV, Treemap, Tulip, Walrus
Visualization of Time-Oriented Data [AMST11]							Microsoft Word, Polaris, Visage, VisuExplore, Xmdv-Tool
Multidimensional Data Visualization [DKŽ12]							Matlab, Xmdv
Visualizing Time [Wil11]	X						Adobe Photoshop
Graph Analysis and Visualization [BJ15]	5			29		X	**Cytoscape(10), Gephi(7), Influent(6), NodeXL(8), Python(18), yEd(4)** Aperature Tiles, Geotime, Hadoop, Microsoft PowerPoint, Mondrain, SPARQL, Titan
Applying Color Theory to Digital Media and Visualization [Rhy16]		X					Adobe Color CC, Adobe Photoshop, Coblis, Color Companion, Color Gamut Mask, Color picker tool, Colorbrewer, COLOURlover's Color Palette Software, Paletton, PANTONE Studio app, Vischeck
User-Centered Evaluation of Visual Analytics [Sch17]		X					Gotham, Metropolis, Tableau, TRIST
Visualizing Data [Fry07]					326		**MySQL(4), OpenOffice(2), Peal(4)**
Interactive Data Visualization for the Web [Mur13]		X	X	183	X		**HTML(43)** Arbor.js, Crossfilter, Cubism, Dashku, DataWrapper, dc.js, Flot, ggplot2, gRaphael, Highcharts JS, JavaScript InfoVis Toolkit, jqPlot, jQuery Sparklines, Kartograph, Leaflet, Modest Maps, NVD3, Paper.js, Peity, PhiloGL, Polychart.js, Polymaps, Raphael, Rickshaw, Sigma.js, Three.js, Timeline.js, Tributary, YUI Charts
OpenGL Data Visualization Cookbook [LL15]							**Android SDK(4), Java(11), OpenCV + C++(11), OpenGL + C++(168), OpenGL ES + C++(51)**

Table 2.7.: Table showing software tools mentioned in books. The numbers indicate the number of dedicated pages to explicit operating instructions or code and the colour intensity an indication of the percentage of the book dedicated to the tool. The 'X' mark and non bold text indicates a tool that is mentioned without explicit instruction. **Bold** text indicates tools where explicit operating instructions or code is given along with the number of pages where the tool is described. '**S**' indicates code given in supplementary material.

Chapter 3.

Feature-Rich, GPU-Assisted Scatterplots for Millions of Call Events

"Most of 'big data' is a fraud because it is really 'dumb data'."
— Peter Thiel – Entrepreneur (1967–)

This chapter aims to address the challenge of visualising large amounts of data. The data-set this thesis is based upon consists of almost five million calls, therefore the aim is to produce a visual analysis tool to enable exploration of this data-set. From the knowledge acquired performing the survey presented in Chapter 2, and the requirements established from the domain users, scatter plots were identified as an easily accessible and well understood visualisation design to address the challenge.

The application was developed with feedback from QPC Limited guiding the evolution. Challenges encountered with this application include speed of filtering, with almost five million calls to be assessed with each filtering event, and the responsiveness of the application. This chapter is based on a paper which appeared at the *Computer Graphics & Visual Computing* (CGVC) conference in 2018 [RRL*18a], and extended into a journal article appearing in the MDPI journal *Computers* [RRL*19].

3.1. Introduction and Motivation

In the United States, there are 2.6 million contact centre agent positions in 40,750 contact centre locations. This represents 4% of the adult working population [Con18b].

A similar proportion of the adult working population in the United Kingdom is also employed by the contact centre industry representing 770,00 agent positions across 6,200 sites [Con18a]. This is set to increase with a recent survey revealing that 67.8% of contact centre operators forecast an uplift in the number of overall interactions [Dim16]. This highlights the impact of the call centre industry on the global economy.

The primary way to contact any large customer facing company is through a contact centre, usually by telephone. Therefore it is paramount to provide a satisfactory customer experience. Four out of five organisations recognise customer experience as a key differentiator between them and their competitors and over three-quarters of companies rank customer experience as the most strategic performance measure [Dim16]. Better customer experience also has financial benefits with 77% of organisations able to report cost savings from its improvement [Dim16].

Call-centres are a variant of contact centres that focus solely on telephone communications not other contact methods such as web-chat and email. Call centre metrics have traditionally focused on customer service times, queue wait times, call abandonment rate and other similar metrics [AAM07]. However, customer experience is a multifaceted phenomenon with many influences that span multiple interactions between the organisation and the customer. Customer relationship management systems are used to capture and store information related to customer interaction with a given company. The use of these systems can decrease overall call volume [MG06]. To further improve call centre performance, it is important to collect and analyse detailed call records. Data collection is often performed by call centre operations systems. However, with several attributes for each call recorded and a high call volume, the amount of data becomes difficult to analyse.

Data visualisation and visual analytics provide an effective means of analysing data and facilitate insight into behaviour. In this chapter, we present techniques and an application for visualising a large multi-call centre data set. We demonstrate our application with a data set comprising of almost 5,000,000 calls collected over a month, with each call described by over 70 attributes including over 32 million events. Our application design is based on Shneiderman's visual information-seeking mantra of overview first, zooming and filtering, and details on demand [Shn96]. We present visual designs that enable the linking of calls associated with individual customers to track each customer journey. We also demonstrate the use of CPU vs GPU-based computation for enabling fast filtering and rendering of large data sets filtered by multiple attributes, and asses the performance. Our contributions are:

- A novel feature-rich interactive scatterplot application that visualises 5,000,000 calls

- The ability to track customers over multiple calls

- Advanced interactive and hardware accelerated filtering of call and customer parameters with evaluation of performance

- Multiple methods of exploring call variables including animation features

- The reaction and feedback from partner domain experts in the call centre industry

An example of an image created by the software can be seen in Figure 3.1 which shows call time against call date and time, with point size mapped to call duration.
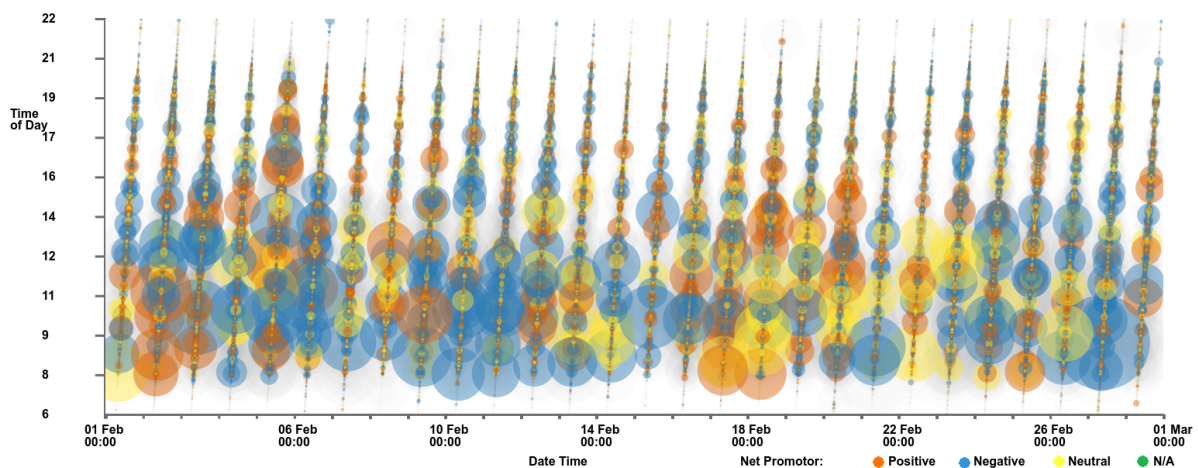
Figure 3.1.: Scatterplot with call time on the y-axis and call date and time on the x-axis. Point size is mapped to call duration and colour to customer feedback score. Data points that represent calls that do not have a feedback score are filtered out. A clear trend is observable: the majority of the longer calls with the largest points are at times before 14:00, with shorter calls in the afternoons and evenings.

The remaining sections of this chapter are as follows: Section 3.2 details work related to this topic, including call centre operations management, hardware acceleration, and scatterplot applications. Section 3.5 outlines the rich set of features and implementation of the application. Domain expert feedback is presented in Section 3.6. A conclusion is drawn in Section 3.7.

3.2. Background

For all related research literature, we first consult a survey of surveys in information visualisation by McNabb and Laramee [ML17] and a survey of information visualisation books in Chapter 2 and [RL19]. Friendly and Denis [FD05] show that the scatterplot has a history dating back to the 17th century, however there are limitations to the visual design when plotting large volumes of data. Some modifications, by methods such as subsampling, binning and clustering, have been proposed to overcome the limitations due to large numbers of points. Ellis and Dix survey clutter reduction methods for information visualisation [ED07]. Methods explored include clustering, sampling, filtering, use of opacity, differing point sizes, spatial distortions and temporal solutions. Fekete and Plaisant present techniques to interactively render a million items [FP02]. The methods presented utilise hardware acceleration and animation techniques are also demonstrated for visual continuity.

Sarikaya and Gleicher also survey scatterplot techniques and identify which design options are best suited to different scatterplot tasks [SG18]. The chapter first outlines analysis tasks performed with scatterplots before examining different data characteristics. A taxonomy of scatterplot designs is presented with reference to suitable tasks and data characteristics.

3.2.1. Overplotting Reduction

A binning technique to reduce clutter is introduced by Carr *et al* [CLNL87]. They demonstrate the use of hexagonal bins with the size and colour of each bin proportional to the number of points.

Bachthaler and Weiskopf present a method of visualising spatially continuous data from an arbitrary density input scatterplot field [BW08]. This method provides a scalable solution to plotting large volumes of data, however the technique is intended for continuous data rather than discrete points. The technique is extended to utilise GPU hardware acceleration [BFW09], and to utilise alternative interpolation methods [BW09].

Keim *et al.* propose a space distortion technique to minimise overlap of data points [KHD*10]. The user is able to control the level of overlap and distortion to view

trends in the data. Deng *et al.* introduce a technique for visualising overlapping data by stacking elements in a third dimension [DWA10].

To overcome overplotting, Chen *et al.* use a sampling method to form a cloud that represents multi-class point distributions [CCM*14]. Mayorga and Gleicher use a kernel-density estimation of multi-class data to visualise dense regions as contour bounded areas [MG13]. The technique presented also supports the use of GPU computation to enable interaction with large data sets comprising of up to three million data points.

Micallef *et al.* consider human perception as a cost function to determine design parameters in dense scatterplots, according to the required data analysis task [MPOW17]. This results in scatterplots with enhanced visual quality.

3.2.2. Information Visualisation and Hardware Acceleration

Elmqvist *et al.* [EDG*08] present a GPU implementation of an adjacency matrix where 500,000 French Wikipedia pages are represented by 6,000,000 links. McDonnel and Elmqvist [ME09] present a refinement of the traditional information visualisation pipeline, to incorporate the use of GPU shaders, enabling the use of parallel computing and interactive plotting of large data sets. The technique is shown to be applicable to many visual designs including treemaps and scatterplots. They highlight that few applications utilise GPU techniques and postulate that this is due to a gap between the abstract data types requiring visualisation and the GPU shader languages that would be used. To remedy this, they present a visual programming environment that generates the required shader code.

Mwalongo *et al.* discuss web-based visualisation applications that utilise GPU-based technologies such as WebGL to render large data sets [MKRE16]. Technologies are categorised according to their application domain with categories covering the scientific visualisation, geovisualisation, and information visualisation fields. The survey features three publications that utilise hardware acceleration to process and render scatterplots [LJH13, AW14, SGC*15]. These publications however only feature small data sets or use a pre-processed aggregation to reduce the number of data points, whereas we demonstrate fast filtering and rendering of almost five million data points. These papers rely on web-based technologies such as JavaScript and WebGL, while we concentrate on local GPU computation.

3.2.3. Call Centre Analysis Literature

The operation of a call centre is complex with many intricacies. We recommend that readers consult *"Call Center Operation: Design, Operation, and maintenance"* by Sharp [Sha03] for a comprehensive overview. Another book details optimisations that can be made in call centres by using statistics is presented by Koole [Koo13].

An assessment by Feinberg *et al.* investigate the relationship between customer satisfaction and a number of variables conclude that the only variables that influence satisfaction are the percentage of calls closed on first contact and average abandonment [FKH*00]. They conclude that more research should be made into providing better predictors.

The demands on a call centre can be challenging to predict even with research studying incoming call rate [JK01, WBS07, BGM*05]. This creates a difficult challenge for call centre managers who have to balance costs and the staffing levels required to cope with the call demand. Failure to achieve a correct balance can lead to either high staffing costs or dissatisfied customers with long waiting times trying to reach the call centre. Gurvich *et al.* present an analysis of the staffing problem an a solution for large call centres [GLT10]. Mehrotra *et al.* present a framework for managers to make short notice agent resource adjustment based on updated call forecasts [MOS10].

Due to the complex nature of call centre management, a large body of research addresses the challenges that they face. Askin *et al.* provide a comprehensive survey of the research up to 2007 [AAM07]. The paper is organised into different aspects of call centre management surveying traditional call centre operations, research into call demand modulation, the effect of technological innovation, human resource issues, and the integration between call centre operations and marketing.

A statistical analysis of call centre data is presented by Brown *et al.* [BGM*05]. Three service processes are explored: call arrival, customer patience, and service duration. Shi *et al.* demonstrate the improvement of a telephone response system in a veterans hospital [SEP*15].

A study by Kumar and Telang investigates the use of web-based self-service portals and the effect this has on call load [KT12]. The study was conducted at a health insurance firm with varying results. With an exposure to a vast amount of information, some customers became uncertain and sought more clarity, increasing the number

of calls. However, with easily accessible, clear information call volume decreased by 29%.

Queueing

An important area of research on call centre management is queueing. A number of works examine the behaviour of people in hold queues and techniques to improve their patience.

Announcing anticipated wait times has been shown to improve customer satisfaction in multiple works [Tay94, KLL03, HZ96]. However, Jouini *et al.* model the effects of informing customers of their expected wait time and how they may react if they face a longer delay than stated [JAD11]. They conclude that in some settings, particularly smaller contact centres, it may be preferable not to provide wait times. Jouini *et al.* further extend wait modelling by considering customers with different priorities, proposing two delay estimators that consider a cost function [JAK*15].

Thiongane *et al.* investigate waiting times by considering multi-skilled call centres using artificial neural networks to predict delays [TCl15]. Ibrahim further extends the research into queueing systems by considering the randomness in the number of available agents to receive calls [Ibr18]. They investigate the use of announcements to control customer abandonment and to mitigate costs.

3.2.4. Call Centre Visualisation

Roberts *et al.* present an interactive treemap application for displaying call metrics of calls serviced at a call centre over one day [RTL*16]. Roberts *et al.* also use the same data to demonstrate a higher-order brushing technique for parallel co-ordinate plots [RLS*19]. Their data set is limited to one day only while this work can render a complete month's worth of data.

3.3. User Requirements

Due to the large volume of calls within the provided dataset, the work presented in this chapter aims to provide a simple exploratory tool for the data. This requirement was

established over a series of initial meetings with the domain experts at QPC Limited and due to them having no other means of visualising a month's worth of data prior to the commencement of this project. Previously the experts would use of commercial tools such as Microsoft Excel and Tableau to preform exploratory tasks with the data. The dataset also comprises a large number of variables to explore.

The requirements of the tool are set out as follows:

- A tool capable of reading and compiling the proprietary dataset consisting five million calls.

- Provide visualisations at multiple time scales between a month and a minute.

- Allow for filtering of the data.

In addition to these requirements, the software should also provide a responsive interface for the user.

3.4. Early Visualisation Prototypes

At the commencement of this project, an early visualisation tool was developed to provide initial exploration the data. This application is based on a chord diagram design and features various interaction techniques. Figure 3.2 shows an image of the prototype.

The application is capable of loading a month's worth of calls totalling almost five million phone records. Customers are rendered as curved polylines visiting each axis which represented different call attributes. Additional axes are available external to the main chord circle. These axes represent the duration spent in each part of the call, IVR, hold, and interacting with an agent. Customers are colour-mapped according to their feedback score.

Interactions supported by the application include individual axis filtering using an axis brush, filtering by feedback score and number of calls, and line curvature adjustment.

This prototype was abandoned due to overplotting challenges, however, components of the application are utilised in other applications presented within this thesis. In particular, data loading and analysis components were used in all applications, and
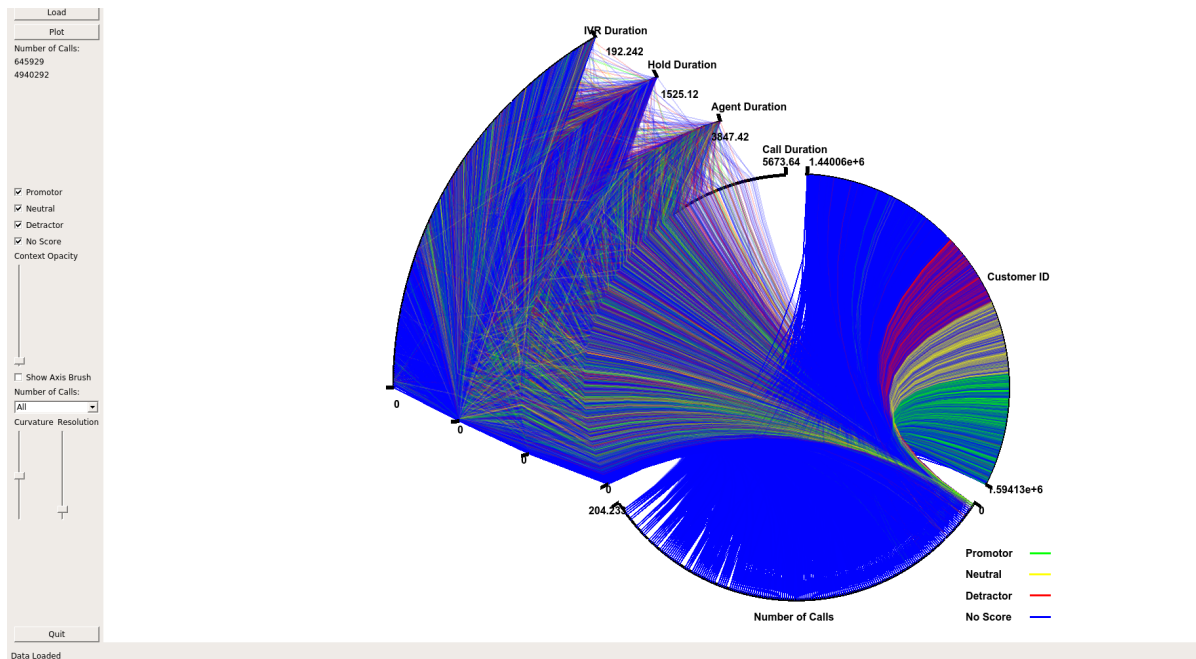
Figure 3.2.: Chord based visualisation prototype developed for the initial exploration of the data.

circular rendering techniques learned were utilised for the work presented in Chapter 5.

3.5. Hardware Accelerated Scatterplots

The software is written in C++ using the Qt framework (version 5.9) [The] and OpenGL (version 4.5) [Khr]. Development was performed on an Ubuntu 18.04 system with an Intel i7-6700k processor, 16GB of RAM and an Nvidia GTX1070 graphics card. The software was also tested on a Windows system with an Intel i7-6700HQ processor with 8GB of RAM and an Nvidia GTX1060 6GB mobile graphics card. The software must first import and process the data before the graphics can be constructed. Processing the data predominantly involves connecting the calls across different files, and linking calls to customers to facilitate look-up. The default view of the application, once data has been pre-processed, can be seen in Figure 3.3, displaying over 4.6 million calls and providing an overview of the data. The daily periodicity of call volume is immediately conveyed. The main window of the application shows the scatterplot chart, with a side panel for various interaction and filtering options, based on Shneiderman's visual information-seeking mantra [Shn96]. These interaction options include:

Variable	Description
Time	The time and date at the start of the call
Normalised Time	The time (duration) since the first contact of that customer
End Time	The time and date at the end of the call
CES	Customer Effort Score - A derived metric for customer investment
Cost	The cost of the call to the operator in pence
Call Duration	The length of the complete call (in seconds)
Agent Duration	The number of seconds of agent interaction in the call
Wait Duration	The length of waiting in a call (in seconds)
IVR Duration	The length of IVR interaction (in seconds)
Hold Duration	The length of hold in a call (in seconds)
Time of Day	The time of day at the start of a call

Table 3.1.: A table showing available axis variables, along with descriptions, within the software.

- Fully interactive zooming on two independent axes

- User-chosen axis variables (see Table 3.1)

- GPU enhanced filtering of multiple call attributes

- Animation of call arrival

- Brushing data points for details on demand

Due to the large volume of data, these interaction options are important to enable exploration of the data. Filtering is provided for a number of call attributes and is split into two categories, customer-centric filters, for customer-oriented attributes such as accumulated CES, and call-centric filters, for call related filters such as call duration (see Table 3.2). To garner more information about a particular data point or collection of points, the user is able to brush the point with the mouse which activates a dialog containing details about the call.

Figure 3.3 shows an overview of the call data set. Notable within the figure is the layered nature of the colours representing the call origin. The calls that do not involve a call centre agent (yellow - conspicuous in zoomed section), are predominantly at the bottom with the lowest CES, calls initiated by a call centre agent (orange) generally have a higher CES with the customer initiated calls (blue) sandwiched in between. The

Customer Filters	Call Filters
Number of Calls	Time of Call
Total CES	CES
Total Cost	Agent Duration
Total Call Duration	Wait Duration
Time of First Call	IVR Duration

Table 3.2.: A table showing available filters, and the category to which they belong. Customer-centric filters filter groups of calls belonging to one particular customer whereas call-centric filters filter individual calls.
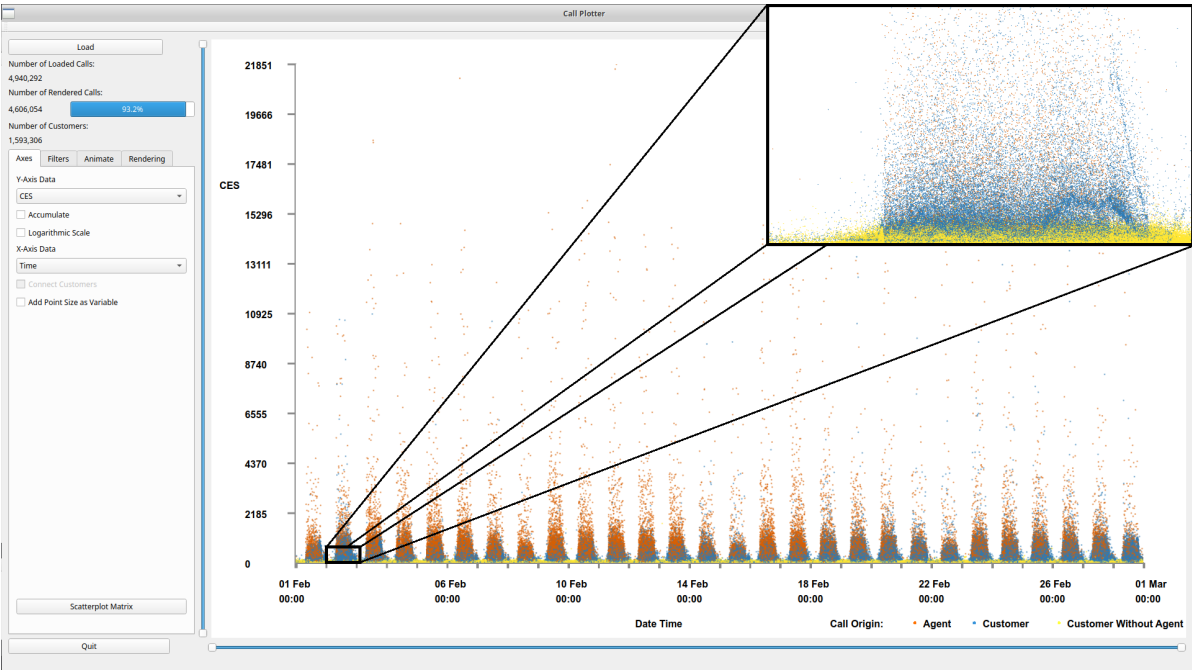
Figure 3.3.: An overview of the application interface with one month of call data loaded. By default, the CES is shown on the y-axis and the time of the call on the x-axis. To increase visibility, we added a zoomed in image from one day. Calls are coloured by their origin, orange indicates an agent initiated the call, blue that the customer initiated the call, and yellow indicates a customer initiated call with no agent interaction. Immediately we can see the periodic pattern of calls spanning a month where peak times are mid-day every day. Within the zoomed framed, zoomed to approximately one day, an interesting wave pattern can be observed in the data.

total number of calls loaded is shown along with the number of calls displayed and a bar displaying the percentage of loaded calls rendered in the top-left corner. Also, the number of customers represented in the scene is given. Within the scatterplot the call volume distribution can be observed, a peak of calls can be seen at each day with

troughs at night times. The majority of the data can be seen in the lower areas of the scatterplot space, with proportionately fewer calls in the upper two thirds.

3.5.1. Scatterplots View

The default view depicts the CES of each call along the left y-axis against the time the call was made along the x-axis, as can be seen in Figure 3.3. Colour is mapped to call origin. Orange indicates an agent initiated the call, blue the customer initiated the call, and yellow indicates a customer initiated call without any agent interaction. An agent interaction might not occur due to the call requirements being served by the IVR or because the customer abandoned the call. The user is able to click on the colour key to choose from a selection of other colour-maps if required. A drop-down menu is available for each axis, to change the axis variables. Options for the y-axis include CES, call cost, call duration, agent duration, wait duration, IVR duration, hold duration and time of day of the call. These call attributes are also available for the x-axis, along with additional attributes of date and time of the start of the call, date and time of the end of the call, and a normalised call date and time. The normalised time is based on the time since the first call of each customer in the data set.

Scatterplot matrix: To provide an overview of all call attribute combinations, the user can choose to view a scatterplot matrix of all available variable choices as in Figure 3.4. This can guide exploration of the data with interesting features, in any particular combination of axis variables, immediately distinguishable. Users are able to click a scatterplot from the matrix to bring the view up in the main window. Once the user selects the scatterplot matrix option, the software takes a snapshot of each scatterplot combination and presents the images in a matrix. All variables available for the x-axis are drawn horizontally and all variables available for the y-axis drawn vertically. Snapshots of each view are saved internally for quicker display of the scatterplot matrix on subsequent uses. Once an individual scatterplot is chosen, the user has the option of keeping the matrix view open, for ease of exploration, or for the matrix view to close, if screen space is limited.

Interaction: The user is able to smoothly zoom in on particular regions of the scatterplot by either using the mouse wheel or sliders at the edge of the plot area. Each axis can be zoomed independently with the mouse wheel zooming in on the x-axis only and the control modifier used in conjunction with the mouse wheel to
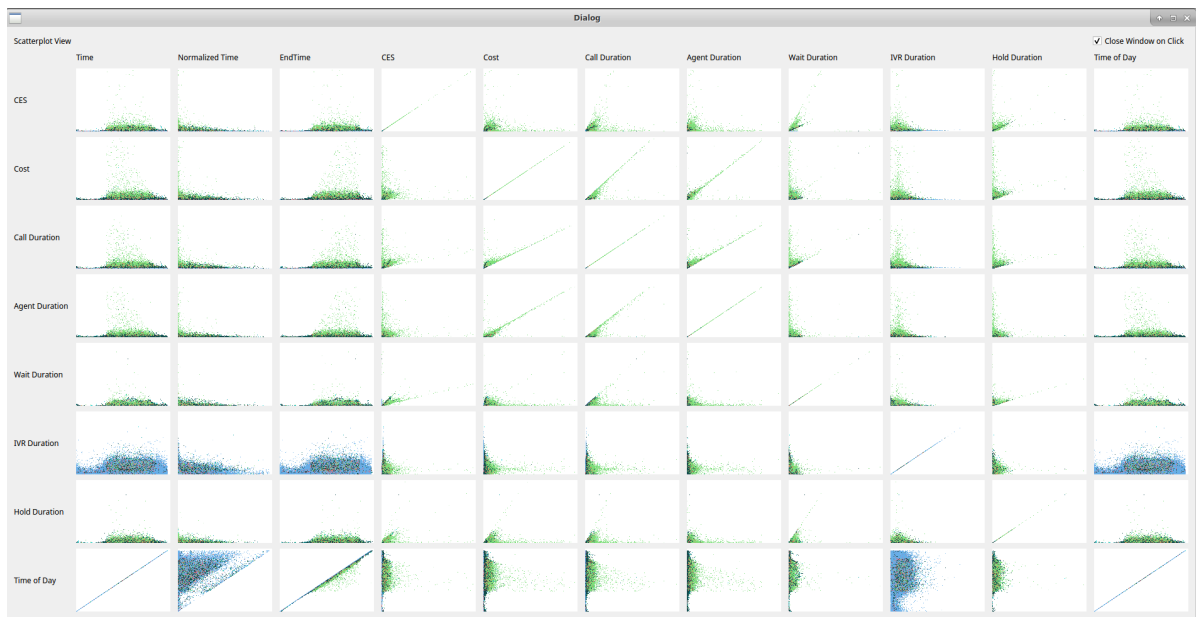
Figure 3.4.: A scatterplot matrix showing all possible axis variables. Users can click any image of interest to bring the view into the main viewing pane.

zoom on the y-axis. Users are able to explore the scatterplot by clicking and dragging the zoomed scene. Figure 3.5 shows a zoomed scene, with zooming on both the x and y-axes. The x-axis has been zoomed from the full month to a obtain a closer look at single day, meanwhile the wait on duration on the y-axis has been zoomed to a maximum of 26 minutes. By applying the zoom a void of calls becomes visible between 17:00 and 18:00, indicating a malfunction with either data recording or call centre operations.

Rendering options: The user also has the option to map the size of the data points to a third call attribute to enable further exploration, as can be seen in Figures 3.1 & 3.6. Figure 3.6 also shows calls connected by a polyline. This polyline is another user option and connects multiple calls that are made from the same customer. To establish a customer's satisfaction with the service they receive, it is important to consider all interactions that the customer makes with the call centre and not treat each call in isolation. To facilitate the exploration of this, we enable the user to accumulate the CES and cost for each customer. This is achieved by ordering all calls from a particular customer chronologically and accumulating the totals for each call.

Focus+Context: Users also have the option to adjust the size and opacity of the data points for easier exploration. In sparsely populated scatterplots, larger data points are easier to distinguish, whilst in over-plotted data smaller points prevent clutter. In
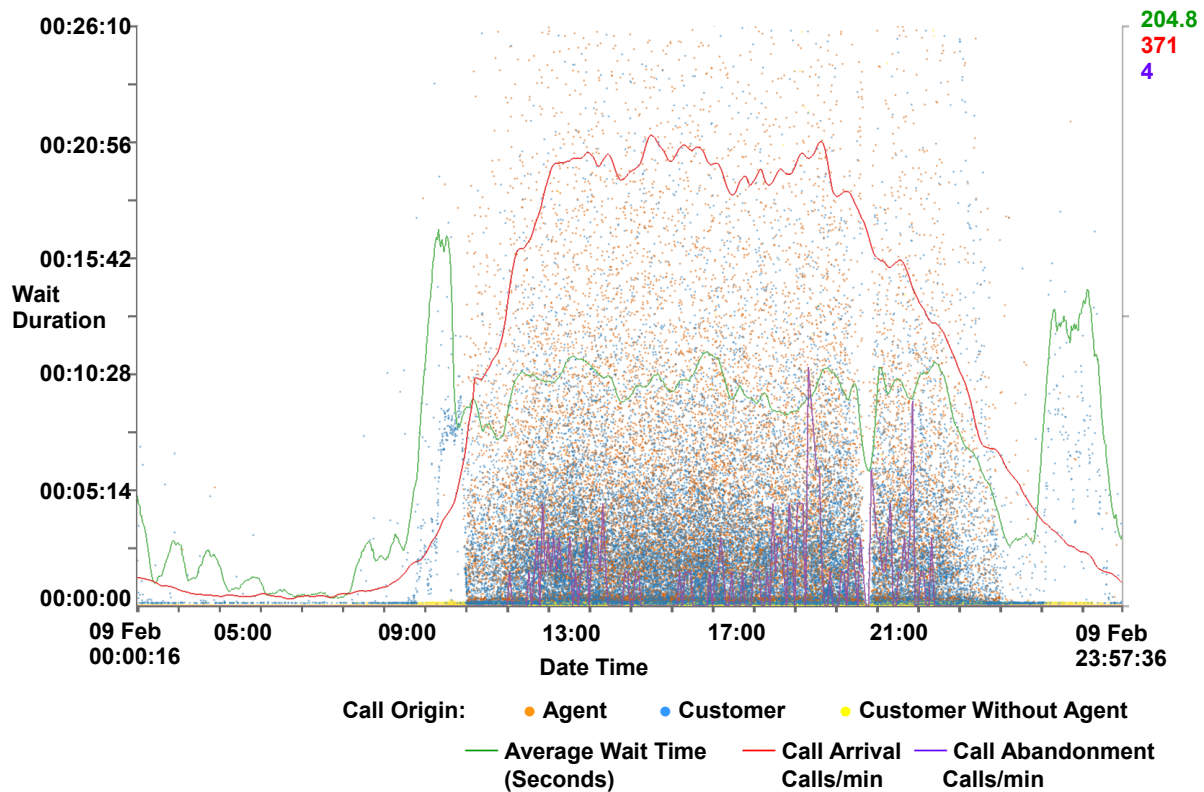
Figure 3.5.: A close-up view of scatterplot with supplementary call metric lines. Wait duration is represented on the y-axis against the date and time on the x-axis. The zoom function is used to obtain a closer look at a single day and to a wait duration of below 30 minutes on the y-axis. Calls are coloured by their origin. Call metrics lines are also drawn. The majority of calls can be observed between 08:00 and 21:00 indicating the times where the main call centres are open. A gap can be seen between 17:00 and 18:00 indicating a malfunction with either data recording or call centre operations. An increase in the waiting times for customers can be observed between 07:00–08:00.

overplotted areas of data, reducing the opacity of the data points enables discovery within dense data regions. This can be seen in Figure 3.7 where the reduced opacity image shows a pattern in the data that was previously hidden.

Users also have the ability to adjust opacity for context calls. Calls that have been filtered out are shown in a faded grey to provide context as described by Card *et al.* [CMS99] For more detail on filtering see Section 3.5.2. Filtered context call data points are also rendered before focus call data points in a two-pass rendering. The first pass renders only context calls while a second pass renders only focus calls. This enables focus calls to be rendered on top of context calls, as in Figure 3.10.
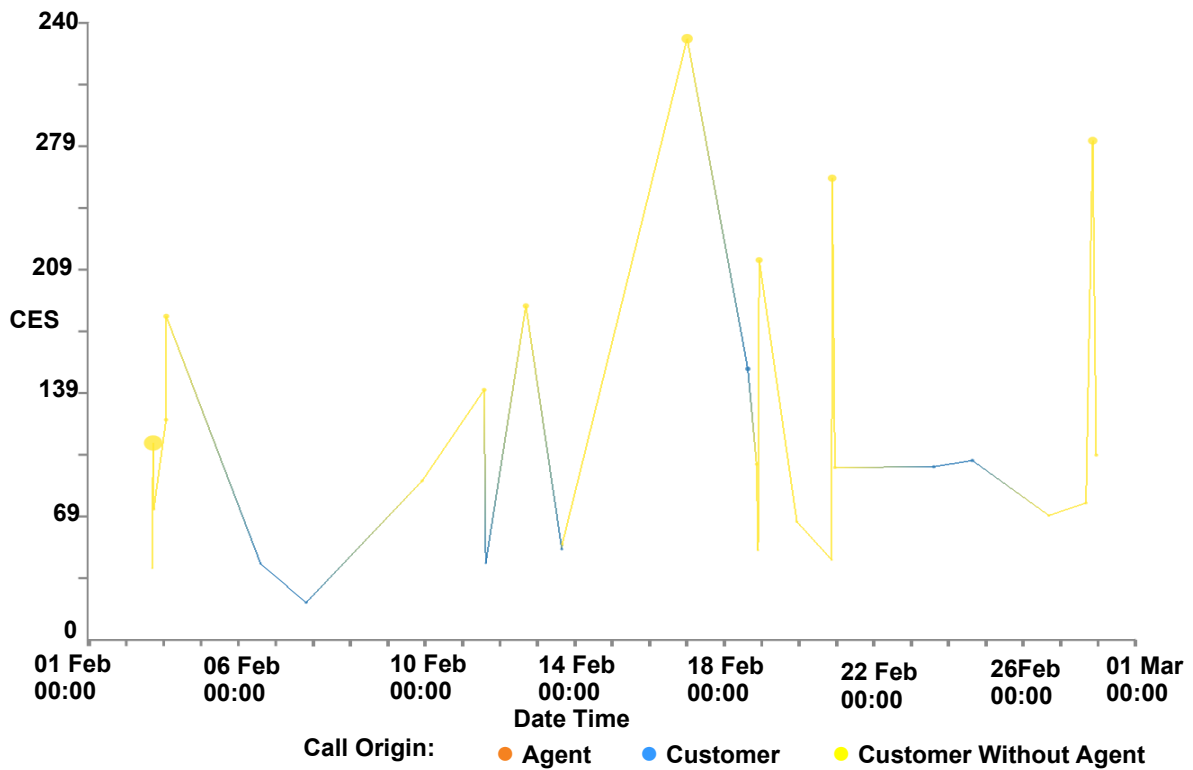
Figure 3.6.: The CES on the y-axis and the time of the call on the x-axis for all of the calls associates with an individual customer over a month. Point size is proportional to individual call duration. Calls are connected with an edge to indicate all calls are made by the same customer. Calls are coloured by their origin. Notable in this figure is that the calls that have the highest CES are the calls without any agent interaction (coloured yellow), while calls initiated by the customer and interact with an agent (in blue) are comparatively short.

It has been found, with initial exploration of the data, that the majority of data points reside in the lower data ranges of CES and cost variables. To enable better exploration of this data we include an option to map to a logarithmic scale, allowing a focus to be put on this data. This is shown in Figure 3.8, where the points are more evenly distributed revealing layers of call origins. Calls without agent interaction have the lowest CES, whereas calls initiated by an agent tend to have the highest CES.

Caller line plots: Call centre metrics are provided to help identify features discovered in the data set as can be seen in Figure 3.5. Metrics provided are call arrival rate, call abandonment rate, and average waiting time. Call arrival rate is calculated by summing the number of calls every minute, and this is then smoothed using a non-parametric regression function on a day-by-day basis, as outlined by Brown [Bro03]. Call abandonment is calculated using the same technique. Average wait time is
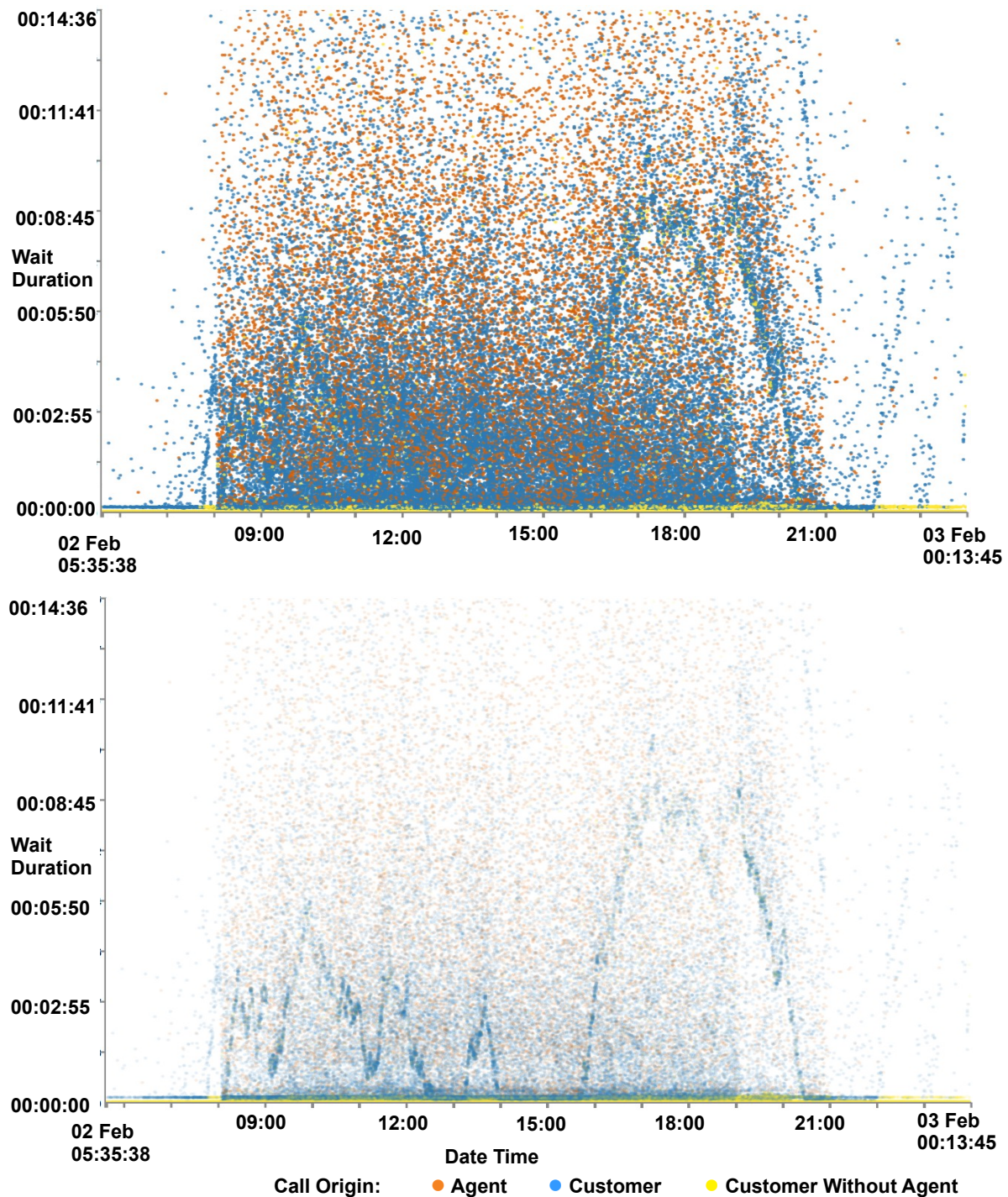
Figure 3.7.: A comparison of plot opacity for the wait time plotted against date and time of calls. By reducing opacity (bottom) different wave patterns become visible within the data set compared to the patterns seen with high opacity (top).

calculated using a tricube function with bandwidths automatically chosen using cross-validation on a day-by-day basis, as described by Brown *et al.* [BGM*05]. To
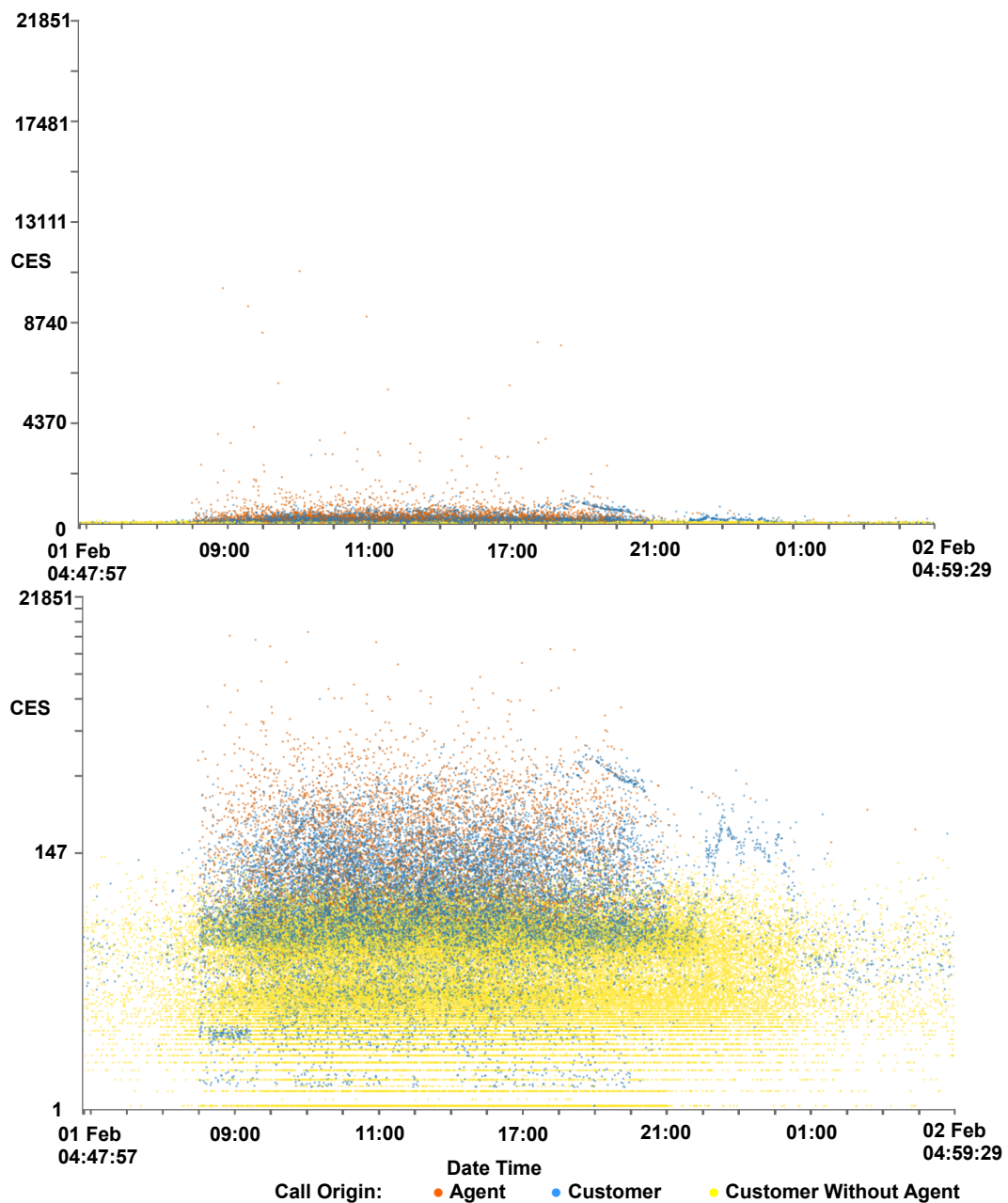
Figure 3.8.: Two scaterplots showing the same data, one with a standard linear y-axis scale (top) and the other using a logarithmic y-axis (bottom). CES is mapped to the y-axis and the time of the call on the x-axis. The x-axis is focused on a single day and calls are coloured by their origin. The three layered trend seen in Figure 3.3 is more visible here, with customers who don't interact with an agent predominantly with lower CES, agent initiated calls with the highest CES and customer initiated calls in between.

supplement this, a typical day line for the wait time, call arrival rate and abandonment can also be shown. The typical day line is constructed by calculating the average day

from a month's worth of data. Because arrival rate is significantly different over the weekend compared to the weekdays, average arrival rate has been separated into weekday values, Saturday values and Sunday values. The typical day metrics can be used as a benchmark and compared to given days to establish if they are above or below average. This feature informs the observation that Mondays are typically busier than other weekdays and that Thursdays are generally quieter. This can be seen in the supplementary video [Ree18]. The metric lines can also be used as benchmarks for comparison across different data sets from different companies.

GPU implementation We utilise OpenGL to provide the graphical element of the software. Encoding data to axis co-ordinates is pre-computed after the data is loaded. This data is loaded into the GPU memory buffer and rendered with the use of OpenGL shaders [KSS16]. Using these techniques and a commodity graphics card, we achieve interactive frame rates with almost 5 million data points. The OpenGL fragment shader code is provided to facilitate reproducibility.

```
layout(location = 0) in float xVert;
layout(location = 1) in float yVert;
layout(location = 2) in float colorAttr;
layout(location = 3) in float filterAttr;
uniform mat4 matrix;
uniform mat4 cMatrix;
uniform float opacityF;
uniform float conOpac;
uniform float pointSz;
uniform float passNo;
out vec4 color;
void main()
{
float grey = 0.7;
vec4 colorAt = vec4(0.0, 0.0, 0.0, 0.8);
gl_PointSize = pointSz;
gl_Position =matrix*vec4(xVert, yVert,0.0,1.0);
//apply color
if (colorAttr == 2.0) color = cMatrix[1];
else if (colorAttr == 1.0)
color = cMatrix[0];
else if (colorAttr == 0.0)
color = cMatrix[2];
else
color = cMatrix[3];
//two pass filter
if (filterAttr == 2.0 && passNo == 0.0)
color = vec4(grey, grey, grey, conOpac);
else if (filterAttr == 0.0
|| passNo == 0.0
|| filterAttr == 2.0)
color = vec4(1.0, 1.0, 1.0, 0.0);
else color *= vec4(1.0, 1.0, 1.0, opacityF);
}
```

3.5.2. GPU Enhanced Filtering

To facilitate user-driven selection and exploration of the call data, we have implemented filters for multiple call attributes. Some filtering can be achieved visually

using the zoom function, however this is limited in functionality. Two groups of filters are used, customer-based filters and call-based filters. Customer-based filters enable filtering of groups of calls belonging to particular customers using variables collated from all calls for each customer. Call filters are used for filtering individual calls. Available customer filters are shown in Table 3.2.

An additional filter is available to distinguish between each of the different origins of the calls. Figure 3.9 shows the user interface to facilitate filtering, with filters split
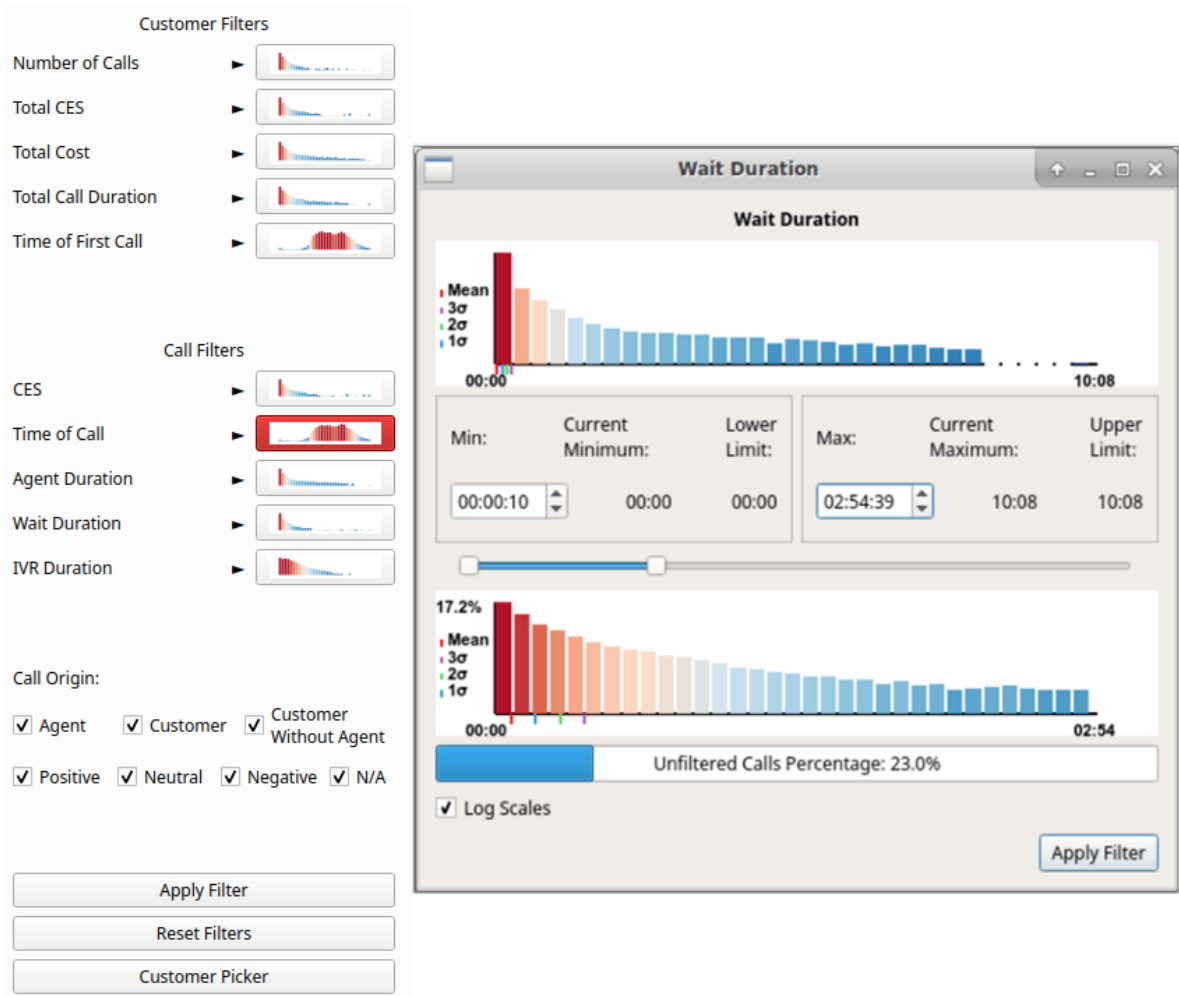
Figure 3.9.: Filter interface including thumbnail previews with call attribute histograms on buttons (left). Buttons are highlighted with red when filters are applied, as can be seen with Time of Call. Filter dialog for wait duration (right). Two distributions are shown, the top shows the total call distribution and the bottom shows the distribution resulting from the user-applied filters.

into customer-based and call-based. The distribution of calls can be seen on the thumbnail previews of histograms, placed on each button, to aid filtering decisions. Filters

that have already been applied are highlighted in red as can be seen with the "Time of Call" filter in the left of Figure 3.9. Clicking a filtering button enables the filtering dialog for that attribute (Figure 3.9 right shows the filter dialog for wait duration). The filter dialog shows two histogram plots of the attribute, the topmost shows the total distribution whilst the lower shows the distribution with user-adjustable lower and upper range limits set in the controls applied. This allows for focus+context style exploration of unevenly distributed data. A selection box at the bottom of the dialog enables a logarithmic function to be applied to the histogram heights, enabling easier exploration of uneven distributions. Filter limits can be set using three control mechanisms, an input box for the lower limit, an input box for the upper limit, and a range slider enabling adjusting of both lower and upper limits. Controls are connected, with changes in one control reflected in the other controls. Indications of the maximum and minimum filtering values, as well as the current applied filter values, are also provided. A bar is shown at the bottom of the call filter dialogs indicating the percentage of total calls that will be displayed after applying the filter, providing an indication of filter effectiveness.

Filters can be applied individually by clicking the apply button in the dialog for the appropriate filter, or all open filters can be applied by clicking the apply button in the main interface. A "reset filters" option is available to set all filters to their maximum and minimum values, and a customer picker is available to choose an individual customer for investigation. Figure 3.10 shows an example of the visualisation with filters applied, along with call metrics. A correlation can be seen with the number of abandoned calls metric line and the call duration of the remaining calls. McDonnel and Elmqvist describe the use of GPU for filtering and visualising using OpenGL shaders [ME09], however this filtering method fails with calls being grouped by customers and requires image processing to ascertain filtering result metrics. In order to remedy this issue, we utilise the parallel processing benefits of a GPU and the Open Computing Language (OpenCL version 2.0) [Mun09a] to quickly filter the number of calls and to return the filtering metrics.

To filter the calls without hardware acceleration requires iterating through each call for each customer, testing if each variable is within filtering limits. With millions of calls, this method can take considerable time to complete. However with the use of parallelism, on the GPU, each call can be tested concurrently. For further guidance and instruction on the use of OpenCL, we recommend the books by Munshi *et al.* and Scarpino [MGMG11, Sca11]. OpenCL functions, known as kernels, are performed on
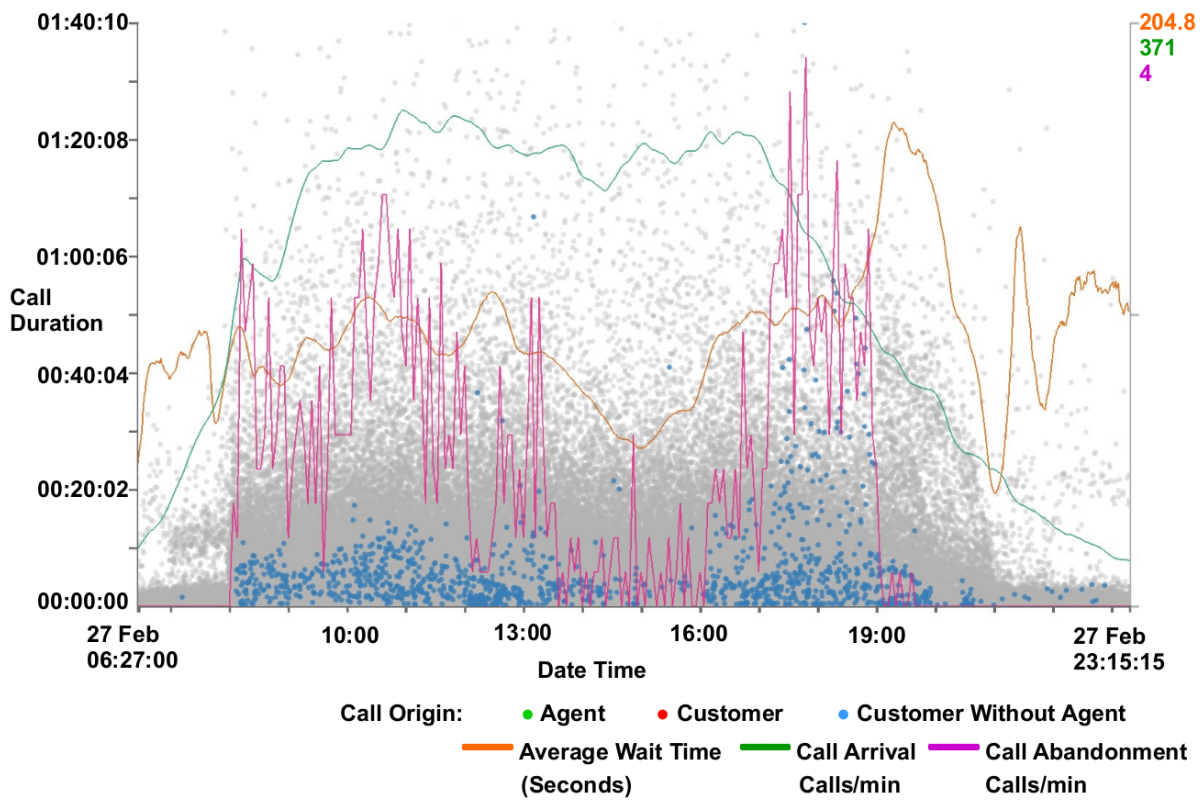
Figure 3.10.: A filtered scatterplot zoomed to a single day and call duration of less than 1hr 40min including supplementary call metric lines. Filters are used to exclude calls with agent interaction and with a waiting time of less than ten seconds. Calls excluded from the filter are rendered gray in the background to provide context. An inverse correlation can be seen with the call duration and the call abandoned rate (red line).

each instance of the data, in this case calls, returning an output. This can be quickly processed to return the number of calls and customers filtered. Our abridged kernel code for filtering follows for reproducibility:

```
int row = get_global_id (0);
if (colIn[row] > 0.0f)
{
if (inputD[row] > maxD || inputD[row] < minD ||
inputE[row] > maxE || inputE[row] < minE ||
inputF[row] > maxF || inputF[row] < minF ||
inputG[row] > maxG || inputG[row] < minG ||
inputH[row] > maxH || inputH[row] < minH ||
inputI[row] > maxI || inputI[row] < minI ||
inputJ[row] > maxJ || inputJ[row] < minJ ||
inputK[row] > maxK || inputK[row] < minK ||
inputL[row] > maxL || inputL[row] < minL ||
inputM[row] > maxM || inputM[row] < minM ||
binaryV == 0 ||
(binaryV%2==0 && binaryL[row]==2) ||
(binaryV/4<1 && binaryL[row]==0) ||
((binaryV==1 ||binaryV==4 ||
binaryV==5) && binaryL[row]==1))
{
sizeOut[row] = 1; //cust filtered out
colOut[row] = 2.0;
}
else
{
sizeOut[row] = 0; //cust stayed in
colOut[row] = 1.0;
}
}
else
sizeOut[row] = 2;
outputA[row] = inputC[row];
```

The kernel code tests if each call variable is between the maximum and minimum ranges specified in the filters and outputs the filtered status. Each call is processed with this code, returning a vector of the filtered status of each call. This vector can then be passed to the OpenGL rendering shader so that the data point for the call can be rendered as focus or context. The vector can easily be processed to calculate filtering statistics quickly.

3.5.3. CPU vs GPU Filtering Performance Comparison

Due to the large number of calls being processed, CPU-based filtering by iterating through an array of calls with multiple attributes was found to take some time. To mitigate this, the calls can be processed in parallel using the parallel compute ability

of a commodity graphics processor using the OpenCL framework. We compare the performance of the OpenCL implementation with a standard C++ implementation for filtering. The OpenCL implementation utilises the GPU for computing the calls and customers to be filtered whereas the standard implementation utilises the CPU. Tests were performed on an MSI GE62VR-6RF laptop with an Intel Core i7-6700HQ processor and an NVIDIA GeForce GTX 1060 (6GB version).

Three tests are performed, one based on a customer-centric filter, time of day, one on a call-centric filter, NPS, and the final test on a combination of filters, call origin, agent duration and number of calls. The time taken to filter the calls and reset the filter was recorded using the OpenCL implementation and without. Each test was repeated three times and an average time taken for each. The data set for one month is used for the performance testing, after erroneous calls are removed, leaving a total of 4,606,054 calls to filter. The first test focuses on the customer filter, in this case, all customers with their first call before 12:00 on any day are removed. This removes 1,552,011 calls and leaves 3,054,043 calls in focus. The second test focuses on the call filter and filters all calls without an NPS score. This option removes 4,425,529 calls leaving 180,525. The final test is a combination filter including call filters and customer filters. Calls without an agent interaction are removed along with calls who spend more than an hour speaking to an agent, and customers with less than ten calls or more than 50 calls. This removes 4,275,003 and leaves 331,051 calls. Results can be seen in Figure 3.11.

For all tests, the OpenCL filtering is shown to be quicker, with test two showing the most significant difference in performance where the OpenCL implementation took an average of 889 milliseconds, and the standard compute took 10,005 milliseconds (10 x longer). Test three has the smallest difference in performance with the OpenCL implementation taking an average of 907 milliseconds and the standard compute taking 3,585 milliseconds. Time to reset the filters is relatively consistent for each test, taking an average of 2,126 milliseconds across all tests for the OpenCL implementation and 11,160 for the standard compute.

From these tests, we can conclude that the OpenCL implementation is between 4 – 11 times faster to filter, depending on the filter applied. Resetting the filter is five times quicker using the OpenCL implementation.
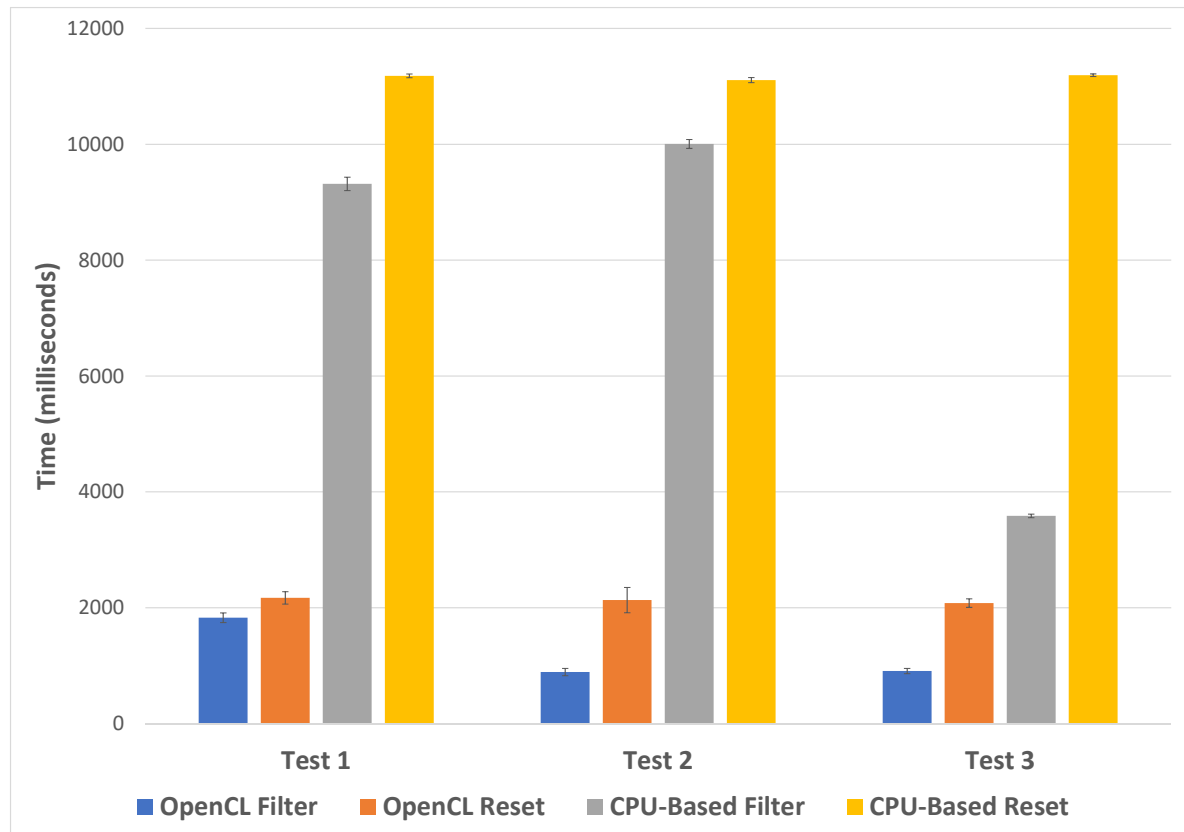
Figure 3.11.: A chart showing average filtering and filter reset performance using OpenCL and without for three separate tests. Error bars indicate 1 standard deviation.

3.5.4. Brushing for Details

Once particular data points of interest have been identified by the user, they are able to brush the desired region on the scatterplot to bring up a dialog featuring all attributes of the brushed calls. This fulfils the final part of Shneiderman's visual information-seeking mantra, [Shn96], details on demand. Figure 3.12 shows an example of the brush dialog. Users are able to copy selected data attributes from the brush dialog for further analysis with other tools. This copy feature was requested by our domain experts to enable further exploration and analysis using different applications.

3.5.5. Animation

To further aid data exploration we have implemented an animation feature that enables the user to view calls arriving as if in real-time or an accelerated simulation of time.

Figure 3.12.: Brush dialog showing all call attributes for brushed calls in full detail.

Users have the option of either discarding calls after they have passed their end time, rendering them in context, or enabling them to be displayed continuously. Figure 3.13 shows the three options for calls passed their end times. The user also has options
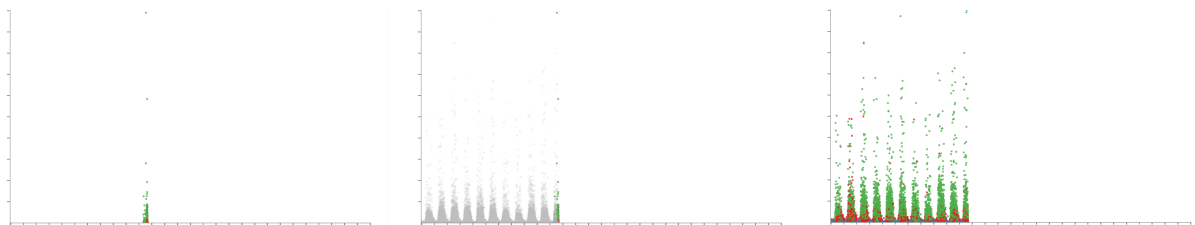
Figure 3.13.: Three screenshots of the software animation feature with the y-axis showing the Customer Effort Score and the x-axis showing the date and time of the calls. The left image shows calls removed after their end time has expired, the centre image shows calls placed into context after the end time, and the right image shows calls continuously once appeared.

to control the speed of the animation by entering a time value for the duration of the animation. The ability to loop the animation such that it restarts after displaying all calls is also available. A progress slider is available to show progression through the animation. Users are able to manually drag the progress slider to advance and rewind the animation and pause the animation. Figure 3.14 demonstrates a feature within the call centre behaviour discovered as a result of the animation feature. The figure shows three images of time of day against agent duration at different time steps of the animation. The left image shows the animation on the fourth day, the middle image during the fifth day, and the right image shows the animation at the end on the

28th day. During the fifth day, a diagonal row of points is seen as highlighted in the middle image of Figure 3.14. Further investigation shows that these calls end within ten minutes of each other, however, why this pattern is not seen on other days is still unknown. Although this observation is visible through trial-and-error using other axis variables, it is particularly apparent using the animation feature.
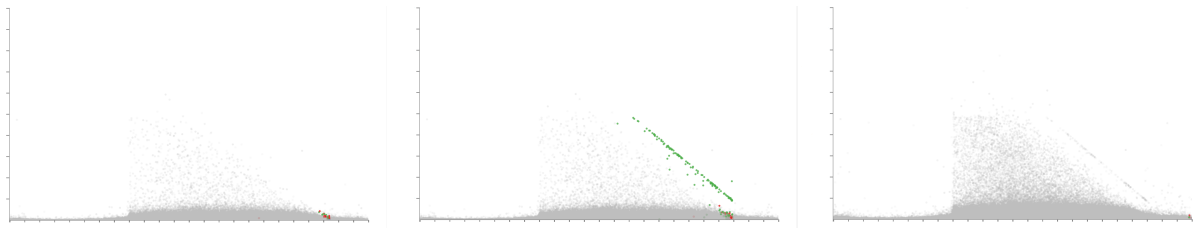
Figure 3.14.: Three images demonstrating the animation feature, showing different stages of the animation. The x-axis is mapped to the time of day at the start of each call while the y-axis shows the agent duration. A month's worth of data is loaded (28 days). The left pane shows the animation during the forth day, the middle image shows the animation towards the end of the fifth day, and the right image shows the animation at the end of the final 28th day. Noticeable is the unexpected appearance of a straight diagonal line on the fifth day seen in the middle panel.

3.5.6. Customer Experience Tipping Point Chart

At the request of our domain experts, we created a chart depicting customer experience against the total journey time, as can be seen in Figure 3.15. The chart depicts two lines showing the percentage of calls, who answered a survey, with positive and negative feedback in green and red respectively. Positive feedback scores are those that provide a NPS feedback score of over eight and are considered customers who would promote the company while customers who provide an NPS score of less than five are considered detractors. Along the x-axis is the customer journey time, this is the total amount of time that the customer has spent interacting with the call centre, across multiple calls. This chart allows for the discovery the call journey tipping point time, where the promoter score first crosses below the detractor line. Customers who exceed this journey time are more likely to be detractors than promoters, providing a maximum journey duration target for call centres to keep customers happy. As can be seen in Figure 3.15 the promoter score first crossed below the detractor line at approximately 2,900 seconds.
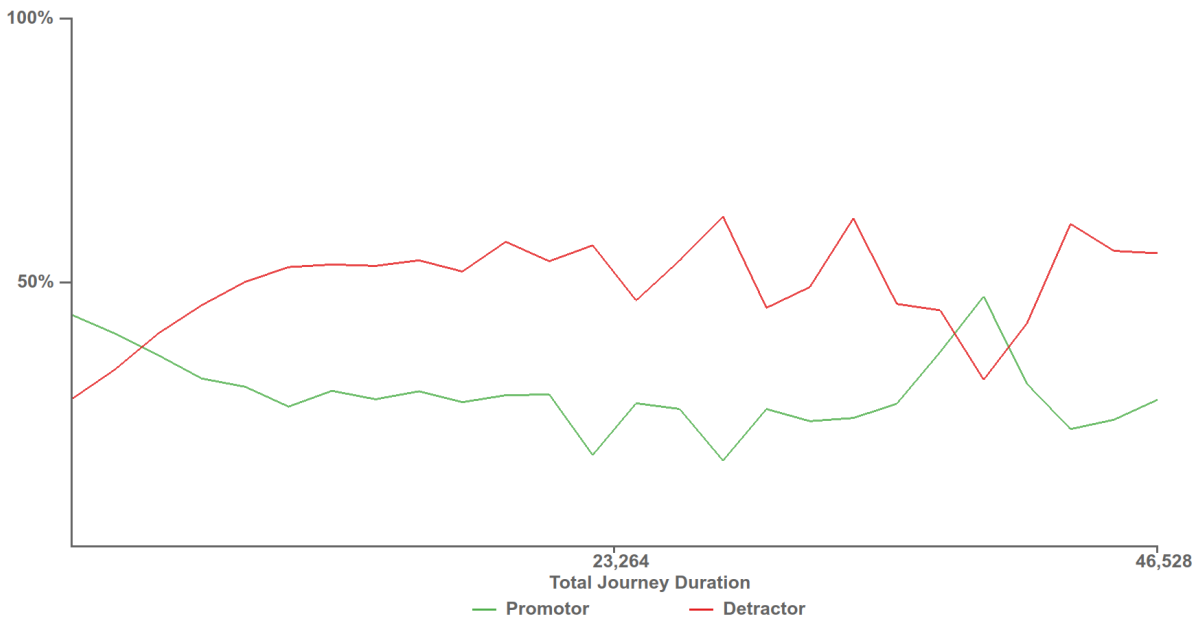
Figure 3.15.: Customer experience tipping point chart depicting when customers who provide negative feedback surpass the number of customers who provide positive feedback. The time at which the detractor score line first surpasses the promoter score line, 2,900 seconds, provides a benchmark for call centres to try to stay below to keep the most customers satisfied.

Video Demonstration: Please visit `https://vimeo.com/305933032` to view an updated demonstration of the application and its features.

3.6. Domain Expert Feedback

The software was developed in collaboration with our industrial partner QPC Limited. The development of this application has been driven by discussions with QPC Ltd. and their requirements. Here we present important feedback garnered from guided interviews [HHH16] with three of their experts.

Feedback was garnered over three recorded interview sessions, the first was an hour meeting in person, the second via a one hour video conference, and the third at an hour and a half meeting with all members present. Interviews were conducted using guidance garnered from Hogan et al. [HHH16].
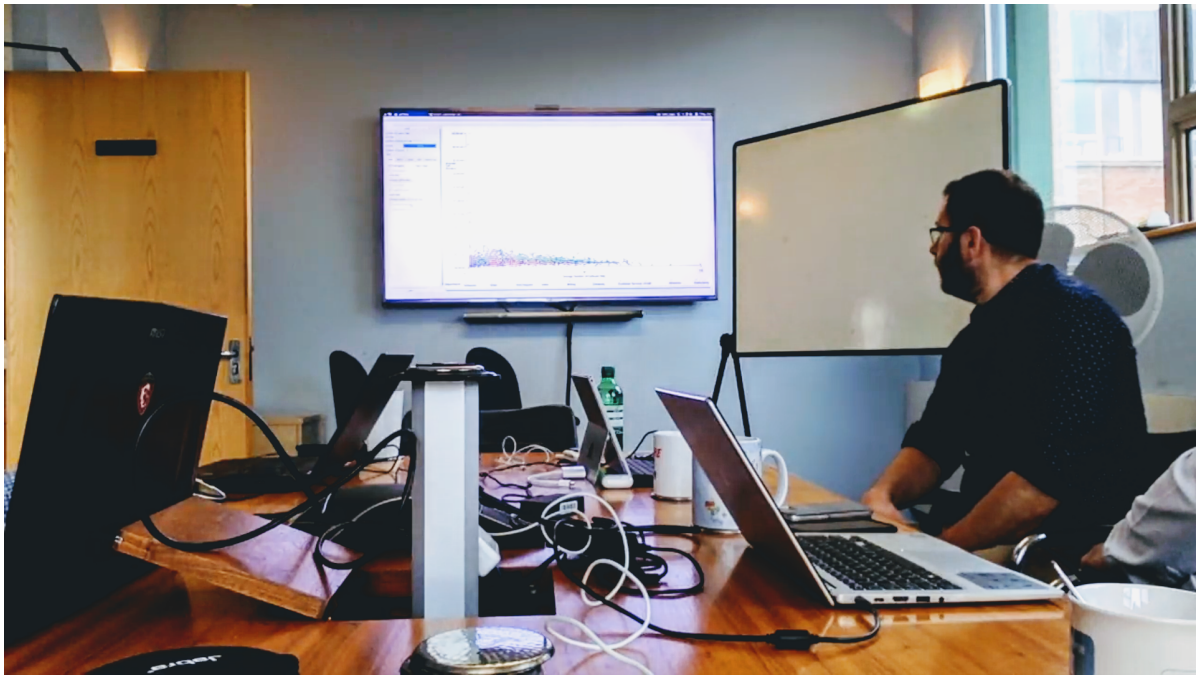
Figure 3.16.: An image from one of our feedback sessions with QPC Ltd.

Initially, when shown the software with a month of data loaded, the experts were impressed with the application's ability to plot a large number of data points. When asked if they had seen a month's worth of data before, an expert one replied:

> "Not at this speed, no. We've had to go down the route of pre-aggregating the data to get the speed."

In fact, this is the first time anyone has seen an entire month's worth of data simultaneously, in their entire company's history. Previous commercial products used to explore the data set have been limited in the size of the input data set. After demonstrating the zooming, panning, and data variable choices, the experts saw the value of the application and the exploration potential it provided, expert two stated:

> "It'll be interesting to put a new data set in that we haven't looked at before, that we haven't got any knowledge of and to instantly then be able to see something."

The filtering ability of the software, in particular, was well received, with the thumbnail previews of histograms exalted for their ability to give an initial summary of the different fields and distributions.

> "I like the look of that, it looks nice first of all, it's giving you a good summary of the different fields and distributions."

The ability to compound the filters and the briskness of the filters were praised by expert two.

> "You've given the ability to filter the contacts in quite a few different ways and to enable you to focus in on particular areas and for the individual contacts you come down to you can look closer, maybe in a different application."

Positive feedback was also received from expert three for the metric and the typical day lines:

> "Yeah, I think it's nice, it lets you look at some standard call centre metrics."

The average plot lines were particularly noted for their ability to benchmark call centre performance. With this feature, our industry partner can, for the first time, compare call centre performance between their customers in addition to different days. The ability to brush for individual call attributes was also welcomed, allowing identification of specifically identified calls.

More general feedback was given with respect to the usefulness of the application to QPC Ltd. and their customers by expert three:

> "I think there are two immediate purposes it serves, one is validation, it'll throw up those outliers we've got... and two, from an insight perspective... we'd probably show this to the customer to demonstrate the insight, to show how flexible the data is."

This was followed up with a statement from expert one which we feel encapsulates the aims of the application:

> "It makes the application that you've created a stepping stone... because you can look at a large set of data and filter down to a smaller number of calls, this application looks useful for that then potentially you can go and look at some more specific detail with another application or even you just literally go to the database and take those call I.D.'s you've listed out there even just go directly to the database."

Recommendations for improvements were received from the feedback sessions in particular requested was the ability to include more caller data dimensions.

3.7. Conclusions

We present an application capable of visualising millions of calls representing a month's worth of real-world data for the very first time. The application enables fast exploration of a large data set including rapid filtering and brushing for further detail, reflecting Shneiderman's visual information-seeking mantra [Shn96]. Details of fast filtering using OpenCL are presented. Insights into the data set are presented, and feedback from our expert industrial partner is also provided.

Evaluation Restrictions   Although the application created for this project was provided to QPC Limited, the industry partner, they were unable to deploy the tool to provide more thorough feedback. This was due to other commercial projects being prioritised at the company. As such the feedback presented is limited to the reaction after a few hours of exposure to the application. The limited exposure to the application, however, was the only time that the experts had seen a whole month's worth of data visualized in a single scene.

Chapter 4.

AgentVis: Visual Analysis of Agent Behaviour with Hierarchical Glyphs

"The greatest value of a picture is when it forces us to notice what we never expected to see."
— John Tukey – Mathematician (1915–2000)

Following the initial exploration of the data-set using scatterplots, Chapter 3, a natural extension to this is to change the points representing the calls into multidimensional glyphs. This allows for multiple attributes to be displayed at the same time. However, an overplotting issue exists with such a design and is a challenge addressed in this chapter.

The focus of the data analysis was also changed to be on the call centre agent, rather than the individual calls. Previous projects have focused on many aspects of the dataset, however this is the first analysis of agent based data metrics. This chapter is based on a paper published in *IEEE Transactions on Visualization and Computer Graphics* [RLB*20].

4.1. Introduction

Data glyphs are a common data representation that provides a means of conveying multiple data-dimensions for each entity. Data glyphs are defined by Fuchs et al. [FIBK17] as:
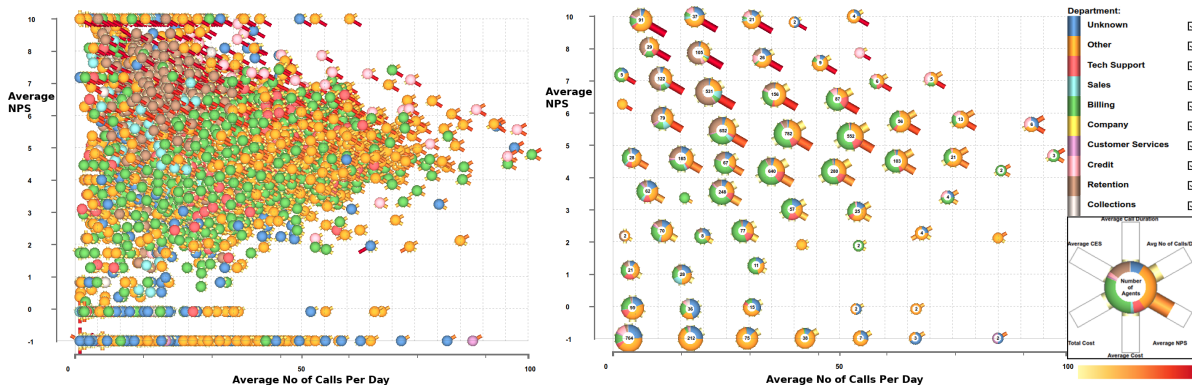
Figure 4.1.: The left figure shows an overplotted scatter plot with more than 6,500 glyphs. The right is the equivalent clustered hierarchical glyph with reduced overplotting and clutter. Each glyph represents a call-centre agent (or a group), showing six attributes and colour-mapped by their department. The x-axis is mapped to the customer feedback Net Promoter Score (NPS), and the y-axis is mapped to the average number of calls the agents take each day. The volume of agents, and the departments they represent, in each region of the plot becomes apparent after clustering has occurred with a number indicating this (see right image). We can observe that agents from the sales, company and collections departments all handle less than 50 calls per day, whereas this information was previously occluded. Outliers are also easily identified by their distance from other agents, therefore they rarely cluster.

"Data-driven visual entities, which make use of different visual channels to encode multiple attribute dimensions. They can be independently spatially arranged and can vary in size."

However, due to the challenges associated with scalability and the complex shapes of multi-variate glyphs, overlapping glyphs can occlude one another and become difficult to interpret, making visual comparison and analysis difficult, see Figure 4.1 (left). Their size constrains the number of glyphs that can be shown on a particular canvas, therefore limiting the number of data samples that can be displayed. This scalability limitation is a general perceptual and technical challenge for the use of glyphs, with large data-sets becoming more of the norm in the age of big data. Some solutions have been proposed to overcome the scalability challenge [WLS98, MRZH14, YWR03, FJWS16], however, these techniques have not been demonstrated with a large real-world data set, multiple overlapping data samples, and a compelling industry application with big data.

The scalability limitation is especially noticeable in the call centre industry, with large call centres featuring thousands of agents collectively handling hundreds of

thousands of calls every day. The call centre industry employs approximately 3.6 million agents in the U.S. alone [Con19], which signifies the importance and size of the industry to the global economy. Call centres represent the main point of contact between most customer-facing businesses and their clients. Therefore large companies require an abundant number of agents to facilitate this. It is not uncommon for large companies to employ thousands of call centre agents. To provide company-wide analysis of agent behaviour and performance, large scale data analysis is required with visualisation of a large number of complex multi-variate data samples.

We present a novel application that incorporates and adapts a number of techniques to develop a system capable of visualising thousands of interactive high-dimensional glyphs by utilising aggregated clusters. We demonstrate the use of dynamic hierarchical glyphs that aggregate according to zoom level and a user-customizable distance metric to manage screen space while reducing glyph overlap. Figure 4.1 shows an example of our clustering on the right with the same data un-clustered on the left. The use of multiple clustering strategies developed in collaboration with a team of domain experts in the call centre industry is also exhibited. We utilise these features to explore, analyse and present complex agent behaviour provided by our industrial partner. Our contributions are:

- An application that explores and renders thousands of multi-variate glyphs representing real-world commercial call centre data, guided by a series of feedback sessions with industry

- A novel hierarchical glyph aggregation and placement algorithm that alleviates occlusion and complexity for thousands of multi-variate glyphs

- An industry-driven usage scenario and feedback from practising domain experts in the call centre industry

The motivation for this study was the need for our domain experts to visually explore agent behaviour, something they were unable to do with currently available tools. This is the very first time our call centre partners have been able to visually explore and analyse an entire month's worth of agent work with almost 5 million calls and 6,500 agents simultaneously. Individual agent behaviour, represented by individual glyphs facilitate a comparison of agents, with trends and unique outlier behaviour easily identifiable. Glyphs provide an intuitive form of conveying behaviour and were favoured by our domain experts for this reason. They found glyphs to be intuitive for data analysts, managers, and for clients. Technical challenges associated

with our development include designing and implementing dynamic glyph clustering criteria for different zoom levels, developing smooth transitions, and interactively rendering large numbers of glyphs. We also consider the influence of clustering behaviour. We report on our prototype's design process utilising guidance provided by Sedlmair et al [SMM12]. On reflection the design process, we realise that the techniques presented in this chapter are generalisable to other applications beyond studying agent behaviour. The techniques and designs presented in this study are transferable to any suitable multivariate data analysis.

In the next section, we present related work on glyph design, glyph placement and call centre operation with a focus on visualisation. We describe some requirements from call centre domain experts and user requirements from an end user. A description of the dataset used and the attributes of interest are provided. Glyph clustering considerations and implementation are then described before a usage scenario and domain expert feedback is reported. We finish with conclusions and a reflection of the design process. Details of the rendering technique used in this chapter are presented in Appendix A.

4.2. Background

Our work is related to a number of topics including glyph design, glyph placement, clutter reduction, hierarchical algorithms, and call centre operation. Here we discuss related work in each of these subtopics.

4.2.1. Overview

McNabb and Laramee's Survey of Surveys [ML17] identifies three surveys related to glyphs in the information visualisation domain. The first of these by Ward [War02] presents a glyph placement taxonomy for multidimensional data. Borgo et al. provide guidelines on glyph design [BKC*13], and more recently Fuchs et al. reviewed a number of user-studies using glyphs to collate insights into glyph design [FIBK17]. Another glyph based survey is available in the scientific visualisation field for spatial medical data [ROP11]. For a comprehensive overview of glyph-based visualisation we recommend the reader consult these surveys and for a more general overview of

visualisation, we recommend a survey of information visualisation books in Chapter 2.

4.2.2. Glyph Design

A large body of work exists for different data glyph design across multiple disciplines from meteorological glyphs [And57], through geographically mapped medical data [TML*17], to sports visualisation [CLP*15, CPG*16]. Borgo et al. survey and collate previous work to develop design guidelines for the use of glyphs. Not all glyph designs are well received, however [Cor18]. Sailem et al. encode up to 15 variables to a glyph for visualising breast cancer cell phenotypes [SSB15]. Osawa introduces 2D and 3D jigsaw-puzzle-like glyphs to show inheritance relationships in object-oriented programming [Osa02, Osa05]. The main focus of our work is not the glyph design itself. We adapt a well-known multi-variate chart, a ringed bar chart, and develop a hierarchical version of it. The same adaption could be performed with other multi-variate glyphs. Table 4.1 summarises the glyphs discussed here, indicating the number of variables encoded by the glyphs and the number of data samples demonstrated. Clustering criteria are not addressed by these papers (marked by 'N/A') nor are hierarchies and clustering.

Recent work by Kammer et al. [KKG*20] utilises a similar technique to ours by using a combination of scatterplots and glyphs for the visualisation of dimensionality-reduced multivariate datasets. Users are able to pan and zoom the scatterplot, with the glyphs becoming visible by zooming, increasing the level of detail, or using a magic lens. A quad-tree aggregation is available to group data points in dense plots. Users are also able to compare different dimension reduction techniques using a side-by-side comparison. This work differs from our own as our design utilises individually zoomable, describable axis, smooth clustering transitions, and preserves glyph visibility at all times. Our work also provides an aggregated glyph to represent cluster details, which is not implemented in the quad-tree aggregation.

Hierarchical Glyphs

Woodruff et al. present techniques to keep the density of data constant in a zoomable interface [WLS98]. Techniques suggested are the aggregation of glyphs to create a single representative glyph, replacing the glyph with one of lower density and the

removal of glyphs. We expand on this work with the use of multi-variate glyphs and incorporate a user-driven distance metric for clustering rather than grid based.

Yang et al. use a single star plot to represent a cluster of data points, where the values at axis intersections represent the mean of the cluster [YWR03]. Maximum and minimum values for each variable within the cluster are also displayed as part of the glyph design. The star plot cluster is demonstrated on the Cars data set [RD] of 400 data samples structured in a grid layout aggregated in 16 glyphs.

Fuchs et al. utilise transparency and aggregation techniques to cluster overlapping glyphs exploiting a leaf-based design [FJWS16]. Transparency is used to enable identification of overlapping glyphs, however, this is limited in the number of discernible overlaps. A grid is used to find clustered glyphs. Two aggregation techniques are demonstrated which they entitle Prototype Generation and Abstraction. Prototype Generation results in the creation of a new glyph of mean values, or by combining visual elements of each merging glyph to create a hybrid parent. Abstraction results in the glyphs being resized to fit around a branch representation of all glyphs in the grid cell. Fuchs et al. demonstrate their technique using Fisher's Iris data [Fis36], and Cortez and Morais' forest fires data [CM07], which contain 150 and 517 data samples respectively. This work differs from ours as we develop the glyph aggregation to incorporate a dynamic zooming function on independent or coupled axes, clustering and separating glyphs according to zoom level rather than a grid. Our work also demonstrates smooth interactive zooming.

Müller et al. present multilevel glyphs for analysis of biomedical data [MRZH14]. The number of data attributes a glyph represents is dependent on the zoom level, and more data attributes are shown as the user interactively zooms in. At the highest level of detail 16 attributes are mapped to the glyphs, and at the lowest level of detail each glyph is mapped to a single colour attribute, however, smooth transitions between zoom levels are highlighted as an open research question [MRZH14]. Glyph spatial positioning is determined by pre-set binned categorical attributes avoiding potential overlap of glyphs. In contrast, we always map the same number of variables to a glyph and instead aggregate the data attributes dynamically while zooming in or out. We also enable the use of continuous space for spatial positioning. Müller et al. exhibit their techniques on 70,000 data samples of cancer cases. Differences between these works and ours are summarised in Table 4.1.

Reference	2 Number of Glyphs	3 Number of Data Variables	4 Placement Strategy	5 Dynamic Level of Detail	6 Hierarchical	7 Aggregation Method	8 Smooth Transitions	9 Independent Axis Zooming
Glyph Design								
Anderson [And57]	4	5	Data Driven	N	N	N/A	N/A	N
Sailem et al. [SSB15]	19	21	Data Driven	N	N	N/A	N	N
Chung et al. [CLP*15]	55	9	Data Driven	N	N	N/A	Y	Y
Chung et al. [CPG*16]	55	9	Data Driven	N	N	N/A	Y	Y
Tong et al. [TML*17]	209	14	Cartographic	N	N	N/A	Y	N
Hierarchical Glyphs								
Woodruff et al. [WLS98]	500	1	Data Driven	Y	Y	Grid Based	N	N
Yang et al. [YWR03]	400	7	Data Driven	Y	Y	Tree Hierarchy	N/A	N
Müller et al. [MRZH14]	70,000	16	Data Driven (Categorical)	Y	N	N/A	N	N
Fuchs et al. [FJWS16]	517	14	Data Driven	N	Y	Grid Based	N/A	N
Glyph Placement								
De Leeuw and Van Wijk [dLvW93]	12	7	Manual	N	N	N/A	N	N/A
Treinish [Tre99]	400	3	Manual + Grid	N	N	N/A	N	N/A
Ward & Lipchak [WL00]	96	10	Spiral Based	N	N	N/A	N	N/A
Peng et al. [PLD*08]	400	6	Grid	N	N	N/A	N	N/A
Kammer et al. [KKG*20]	Unspecified	10	Grid	Y	N	Quad-tree	N	N
AgentVis	6,500	7	Data Driven, Hierarchical	Y	Y	Dynamic Distance Driven	Y	Y

Table 4.1.: A summary comparison of our work with similar related work, focused on the features implemented on our application. Indicated are the maximum number of glyphs demonstrated in the research (column 2), along with the number of glyph variables used (column 3), the glyph placement strategy (column 4), and whether glyphs provide a zoom adjustable level-of-detail (column 5). Research utilising a hierarchical glyph is indicated (column 6), and for those applicable papers, the aggregation method is described (column 7). Work incorporating the use of smooth transitions is also indicated (column 8) as is work featuring independent axis zooming (column 9).

Glyph Aggregation

Glyph aggregation can be achieved in a number of ways. Fuchs et al. [FJWS16] utilised a grid method, where a grid is placed on the scatterplot, and all glyphs that lay in a particular grid cell are aggregated to form a parent glyph. We consider this method limited due to the reliance of the grid placement determining the merging of glyphs. Two glyphs could potentially overlap, but have their respective centres on either side of a grid line in separate cells and therefore not merge. It is also not ideal for the representation of continuous data.

Yang et al. [YWR03] demonstrate the use of a hierarchy tree to cluster glyphs. Clustering is based on a pre-computed hierarchy tree, with the zoom level determining the level in the hierarchy tree and therefore the number of glyphs shown. The hierarchy

tree is constructed using the BIRCH method [ZRL96], however, hierarchy construction is computationally time-consuming [Ber06]. This differs from our work as we utilise a dynamic distance clustering metric that enables interactive zooming on each axis independently, which is difficult with a single hierarchy tree.

Table 4.1 summarises previous related work and compares it to ours. The work from this category most closely matches our work as a hierarchical aggregation of glyphs is utilised along with a dynamic level-of-detail utilised with the use of zooming. The work of Müller et al. [MRZH14] shows more glyphs than any other, however, the glyphs at this level represent a single variable only. Other than this, our work is demonstrated on a significantly larger number of glyphs compared to other related work. Our work is the only one to utilise a dynamically calculated distance metric to cluster agents, to provide smooth transitions between aggregation levels and to provide the ability to zoom independently on each axis.

4.2.3. Glyph Placement

Ward's taxonomy [War02] identifies two predominant glyph placement strategies, data-driven, and structure-driven placement. In data-driven strategies, the glyph location is derived from the data, whereas structure-driven placement relies on spatial connectivity or relationship between glyphs that defines the location, such as a hierarchical relationship. Furthermore, data-driven strategies are divided into raw and derived placement. Raw data placement strategies rely on two or three data dimensions as positional components while derived data position is determined by analytic processes such as Principal Component Analysis (PCA) or Multidimensional Scaling (MDS). Our work relies on a derived data-driven placement strategy using a 2D scatterplot and a customizable clustering distance metric.

Feature-driven placement is another strategy presented by Peng et al. [PLD*08], where glyphs are placed on features such as iso-surfaces. Another placement strategy is demonstrated in medical visualisation [RSMS*07,MSD*08]. Other placement strategies suggested by Borgo et al. are user-driven placement, where glyphs are positioned to query data in a particular region. Examples of this include a probe for visualising small sections of a flow field as presented by De Leeuw and Van Wijk [dLvW93] and interactive visualisation of weather data by Treinish [Tre99]. In contrast to these strategies, our placement relies on a data-driven placement strategy.

Ward also discusses distortion techniques to help avoid clutter and overlapping of glyphs. Jittering of discrete data points and shifting positions are discussed with the work of Woodruff et al. [WLS98]

Categorical data can be binned in horizontal columns as demonstrated by Chung et al [CLP*15]. Our work plots glyphs on a continuous data scale. Glyph location is determined by continuous data values which can be distorted if using binning or jittering. Chung et al. demonstrate a single extendable axis to reduce glyph overlap on a glyph scatter plot [CPG*16]. This mimics zooming on a single axis, whereas we demonstrate zooming on both scatterplot axes and clustering of glyphs. Chung et al. also demonstrate glyph resizing for focus + context and to reduce clutter. Despite plotting a relatively small number of glyphs, occlusion is highlighted as a challenge for future work. In fact, the overlap of glyphs is one of the major inspirations behind this work.

Ward and Lipchak utilise a linear, cyclic, and spiral layout for analysis of cyclic data [WL00]. The cyclic layout enables easy comparison of glyphs which have a common position in a cycle, which are stacked vertically. Similarly, the spiral layout is constructed in a way that aligns common positions in the same section of the spiral. This differs from our 2D scatterplot grid layout, however, it provides an avenue for future work. See Table 4.1 for a summary and comparison of related work. Our system features a unique combination of thousands of glyphs, an interactive, hierarchical placement strategy, dynamic level-of-detail, smooth transitions, and independent axis zooming.

4.3. Agent Call-Centre Data Description

A description of the dataset used for application presented in this Chapter can be found in Chapter 1.1.2.

Each agent has several attributes for analysis, compiled from all calls that an agent is involved with. These attributes (or characteristics) are as follows (List 2):

- Average NPS
- Average Cost
- Total Cost
- Average CES

- Number of Transfers
- Percentage of Transfers
- Average Call Duration
- Number of Calls

- Average Number of Calls per Day
- Number of Short Calls
- Percentage of Short Calls
- Average Hold Duration
- Number of Stand Alone Holds
- Number of Consult Holds

- Number of Transfer Holds
- Number of Cold Transfer Holds
- Number of Warm Transfer Holds
- Number of Conference Holds
- Total Hold Count

Hold events have been separated into the reason for the hold. Of interest, in particular, is the cold and warm transfer holds. Cold transfer holds indicate when customers are placed on hold for a transfer to another agent, and information about the call is not transferred between the agents. In contrast, warm transfer holds are for customers put on hold for agent transfers where information about the call is transferred.

4.4. User Requirements & Design Goals

This application is developed in collaboration with domain experts from the call-centre industry who tasked us with developing a visual tool that enables insight into the data that they collect. In particular, experts are interested in call-centre agent behaviour. We have been working with our domain experts since 2014 on call-centre visual analytics. For this particular project, focusing on call centre agent behaviour, design requirements were drawn up in one of our meetings with three of the domain experts. The following requirements were established to drive the development of the application:

- **R1** An application that enables exploration of multi-variate call-centre agent behaviour.
- **R2** The capacity to handle thousands of agent representatives.
- **R3** The ability to simultaneously compare multiple agent attributes
- **R4** An interactive interface that allows the user to probe for information and read raw data.

Typical tasks were identified throughout the consultation with the domain experts:

- **T1** Enable exploration, investigation, and comparison of agent performance.
- **T2** Enable the identification of agent behaviour outside of the norm – finding outliers.

- **T3** The ability to compare groups of agents from different business operations categories such as department, site, and skill.

The visualisation community also provides us with guidelines for creating visual designs, in particular, Shneiderman's Visual Information-Seeking Mantra [Shn03] of overview first, zoom and filter, then details on demand. Due to the requirement to be able to visualise a large number of agents, clutter is an additional consideration. Ellis and Dix provide a taxonomy of clutter reduction techniques [ED07], and Ward's taxonomy contains considerations for glyph based designs [War02], in particular, they highlight the need for smooth transitions. By taking these into consideration we develop the following design tasks to guide our development:

- **D1** Develop an application for multi-variate agent exploration, analysis, and visualisation.
- **D2** Provide a visual design with an initial overview displaying all agents.
- **D3** Utilize cluttering reduction techniques to provide clutter-free imagery.
- **D4** Enable exploration with flexible zooming and smooth transitions.
- **D5** Supply agent filtering options.
- **D6** Provide both cluster-based and individual agent details on demand.

Currently, analysis of agent based information is performed manually utilising spreadsheet tools such as Microsoft Excel.

The requirements and tasks established are not unique and are common in most multi-dimensional data analysis projects. Therefore the techniques developed here are transferable to many other domains.

4.4.1. Alternative Design Solutions

A number of potential designs were discussed to meet the requirements set out before the solution we implemented was agreed upon. We identify a number of design solutions from consultation of a multi-dimensional visualisation survey by Liu et al. [LMW*16] and other works [KK96, FL03, DBD18]. The survey also discusses multi-dimensional scaling techniques, although we chose not to develop these as the abstraction required was deemed nonintuitive and non-accessible to clients.

Multiple axis solutions such as parallel coordinates [ID90] and an Andrews Plot [EH91] are other potential solutions. A parallel coordinate solution is implemented

for another related project, as part of the same collaboration [RLS*19]. However, challenges with scalability were identified including overplotting and with the comparison of agent attributes across multiple axes. Although there are solutions to the overplotting challenge [HLK*11, MM08, Sii00], simultaneously comparing multiple attributes across multiple axes is difficult (**T1**). A crowded Andrews Plot also suffers the same problem, with tracking a single line across multiple axes difficult, therefore this design solution was also discarded.

A scatterplot matrix [WAG05b] is an alternative method of multivariate visualisation that we use to provide an overview of the dataset (**D2**). However tracking a single agent across multiple plots is again difficult and clutter is again a challenge with scatterplots.

Multiple linked views present another design solution. Although we have not implemented this solution, it provides a complementary solution that can be added to our implementation. Coordinated multiple views are part of the future work for this project.

A design consisting of glyphs in a scatterplot layout was chosen as glyphs convey a number of variables for a particular agent, enabling the comparison of multiple attributes simultaneously. Glyphs on a scatterplot are also very intuitive. This accessibility is a requirement in order for the domain experts to communicate with clients. Unlike other design choices such as parallel coordinates or scatterplot matrices, glyphs provide a clarity of identification of which variables belong to which agent, glyphs group the visual cues of each agent making it easier to see all the attributes associated with the agent together, rather than trying to track polylines across multiple axes, or points across multiple plots. Our use of glyphs on a scatterplot matrix has challenges associated with scalability, although we try to alleviate this problem with a glyph clustering technique.

4.5. Hierarchical, Agent-Based Glyphs

4.5.1. Workflow Overview

The application is designed for the following general workflow: Firstly a scatterplot matrix of all available agent variables is presented once a dataset has been loaded

(Figure 4.2). Each plot provides an indication of variable correlation, see Section 4.5.2. The user is able to select a plot from the matrix to show those agent attributes as the main axes in the focus plot. Here each point represents an agent as a multivariate glyph. For the next step, to reduce clutter, the user may choose to cluster overlapping glyphs, as can be seen in Figure 4.1 and described in Section 4.5.3. Additional clustering options are available by individual teams, departments, agent skill and call centre site.

Filtering is an optional step that allows for further clutter reduction (Section 4.5.5). Exploration is the next step, the user is able to explore the space by zooming and panning the scene, with glyph clusters updating dynamically. This is exemplified in the the supplementary video [Reea, Reeb]. For further exploration, users are able to customise the glyphs using a variety of designs and with agent attributes of their choice (Figures 4.3, 4.4, 4.5).

Finally, for identifying the agent represented by the glyph, hovering over an individual glyph provides details of the agent associated with the glyph as can be seen in Figures 4.7 & 4.9.

To meet the requirements and tasks outlined in Section 4.4 we developed an application following the guidelines of Shneiderman's Visual Information-Seeking Mantra featuring a scatterplot matrix overview (**R1**) for initial exploration of the data and to find correlating variables. To enable simultaneous comparison of multiple agent attributes we developed a multi-variate glyph based visual design (**R1**, **R3**), with each glyph visualising the characteristics of a single call-centre agent. Glyphs are placed in a 2D scatterplot with axes mapping determined by agent properties (**T1**). We chose a scatterplot in order to facilitate comparison of agents (**T1**). Scatterplots are a familiar and popular form of visualisation, Lee et al. preform an extensive study and highlight scatterplots amongst the twelve most popular visual designs [LKK17], while the use of glyphs facilitate the simultaneous comparison of multiple attributes (**R3**). Scatterplots were also chosen as a result of our discussion with QPC Ltd. They are familiar with scatterplots and can easily convey the results of exploration and analysis to their clients and managers. Scatterplots enable the sorting of agents based on any of their characteristics. This capacity enables fast identification of both outlier (**T2**) and average agent behaviour as we shall demonstrate (see Figures 4.1, 4.7, 4.8, 4.9). With thousands of agents to display, clutter is a significant challenge. To counteract this, we cluster glyphs into hierarchical parent glyphs, removing overlap (**T3**, **D3**). Users are able to interact with the glyphs by zooming in independently on each axis (**R4**, **D4**), with clusters updated dynamically to minimise clutter and occlusion. See the

supplementary video for a demonstration of the application [Reea]. We developed the software in Qt, C++ using OpenGL for the implementation.

4.5.2. Overview Scatterplot Matrix

To provide an overview of the data-set (**D2**), we provide a scatterplot matrix of all agent characteristics. This can be seen in Figure 4.2. A correlation coefficient is also calculated for every agent variable combination. This correlation coefficient is overlaid on each of the grid of the scatterplot matrix, facilitating the search for correlating variables. To further enhance the overview, this correlation coefficient is also used to colourmap the background of each scatterplot in the matrix. Scatterplots with a higher correlation coefficient display a dark red background colour, with lower correlation having a lighter red background, as in Figure 4.2. The scatterplot matrix can be accessed using a UI button, bringing up the view (as seen in Figure 4.2) in a new window. Clicking the plot of interest places that plot into the focus view. An option allows for the matrix window to be kept open after clicking, for use in a split screen/dual monitor scenario, otherwise the window will close.

4.5.3. Glyph Design and Hierarchicalisation

The hierarchical glyph can be derived from a number of pre-existing designs, however, a new visual encoding channel is dedicated to showing the number of glyphs represented by the parent glyph as a result of aggregation. Appropriate glyph design will depend on the data to be analysed and the application of the tool. We demonstrate the application with a number of glyph designs developed in conjunction with our industry partner in the following sub-sections. The focus of this work is not to create a single design solution, but rather to present the aggregation method and to outline the process of hierarchicalisation of a glyph to show the aggregation. Borgo et al. and Fuchs et al. both provide guidelines for glyph design [BKC*13,FIBK17] however they do not present guidelines for creating an aggregated, hierarchicalised glyph.

Singular glyphs: Our application features a selection of glyph designs, each of which capable of displaying six agent attributes (**R1**, **D1**), plus their department. These designs include a bar chart, a combination of an ellipse with a bar chart, and a radar glyph design. See Figures 4.3,4.4,4.5 for a comparison of these glyphs. An example

Figure 4.2.: A scatterplot matrix of all agent variables. A value for the correlation coefficient can be seen on each of the individual scatterplots. Background colour is also used to indicate the correlation coefficient. Of the variables shown (scrollbars can be used to view more agent characteristics) the Number of Transfers and Total Cost show the highest correlation, identifiable by the darker red background colour. Plots along the diagonal of the scatterplot matrix show a perfect correlation coefficient of 1.0 as they show the same variable on both axes. These are automatically given a grey background to indicate their irrelevance.

of a single agent, child glyph can be seen in the top of Figure 4.3, a combination of an ellipse with a bar chart with the six bar lengths representing a chosen call centre agent attribute. These bars also feature a colour redundantly mapped to the value of a call centre agent to facilitate visual comparison of agent behaviour (**T1**). The maximum extent of the bars indicate the maximum value of the mapped attribute found in the dataset. The glyph centre of a single agent glyph is colour-mapped to a categorical variable. This increases the analysis potential and aids differentiation between agents. In our call centre agent data set this is mapped to agent call centre department, enabling comparison of relationships across departments to be easily observed (**T3**).

In our initial multi-variate glyph design process, we investigated a range of literature to find some of the most popular ones [ML17]. The bar and radar chart are very popular. In addition we designed a custom glyph (Figure 4.3) in conjunction with the same three domain experts from QPC Ltd. After discussing the full range
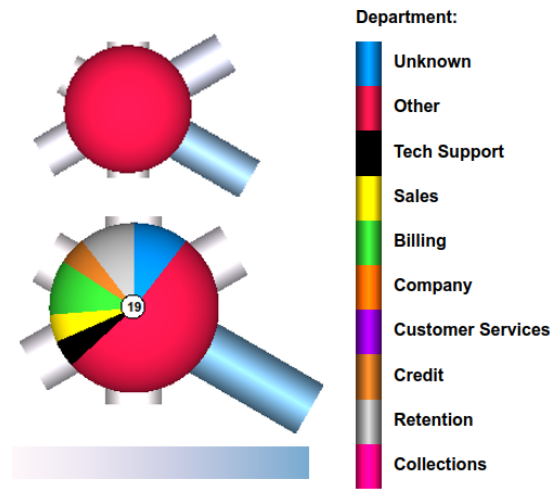
Figure 4.3.: A glyph design featuring a child (top) and parent glyph (bottom). The design features six bars, or spokes, surrounding a central ellipse, each of which represents an agent property. The centre ellipse is colour-mapped to a department, on the top child glyph, representing a single agent, and a single department, whilst on the parent glyph, segment size represents the proportion of each department constituted by the group of agents. Parent glyphs also feature a number at the centre to indicate the number of child agents represented by the glyph. The legend to the right indicates the department colour-mapping of the glyph centres. The bottom colour legend is redundantly mapped (along with spoke length) to any agent attribute other than department, e.g. average hold time.

of possibilities from the literature with QPC, and then providing implementations of the bar, radar, and elliptical glyphs, we focused on the elliptical design due to its symmetry and ease of legibility on a scatterplot. Chung et al. went through a very similar glyph design process for a different application, but also employing a scatterplot layout [CLP*15, CPG*16]. They also arrived at an elliptical glyph design, ours is comparable to theirs. This is one inspiration behind our work.

Hierarchical glyphs: Once two or more glyphs are clustered, a new parent glyph is created as can be seen in the bottom of Figures 4.3,4.4,4.5. Hierarchical glyphs support clutter reduction and minimisation of occlusion (**D3**). They also support comparison of groups of agents (**T3**). The process of glyph hierachicalisation is to start with a child glyph and to add additional encoding channels for the number of child glyphs represented, as well as an indication of the proportion of agents from each department. Parent attribute values shown in the glyph are updated to represent the mean value of all child glyphs. It is important that a parent glyph is distinguishable from a child glyph so that they are not misinterpreted. The parent glyph for the elliptical glyph with bars is shown in Figure 4.3. The parent glyph differs from the child glyph with
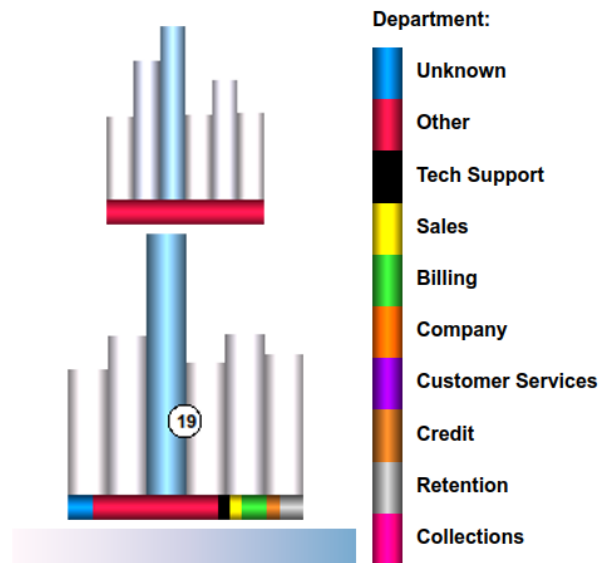
Figure 4.4.: An alternate glyph design featuring a bar chart design with six agent properties, as well as a bar underneath the glyph to indicate the department and department proportions in the child (top) and parent (bottom) glyph respectively.
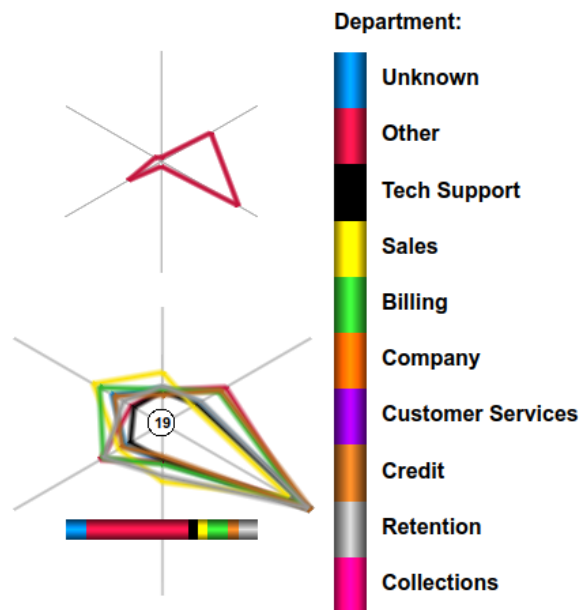
Figure 4.5.: A radar based on glyph design, with six agent attributes. The child glyph (top) features a coloured line to indicate the department while the parent glyph (bottom) features an average line for each of the departments represented within the glyph. A bar underneath the parent glyph indicates the department proportions.

the addition of a centre ellipse and an indicator of the number of child glyphs in the cluster. The coloured segments forming the ring around the glyph centre are also updated to construct a donut-like chart indicating the proportion of each agent's call

centre department represented by the glyph. The protruding bars are also updated to show the mean values of all children for each agent attribute.

In general the process of glyph hierarchicalisation is fairly straightforward. To create a multi-variate parent glyph, a number is added to the glyph centre reflecting the number of children it represents. The size of the glyph itself is also optionally mapped (logarithmically in our case) to the number of children represented. Thirdly, the average value of the child attribute is reflected in the parent, consistent with the child glyph. The parent does not strictly have to represent an average of its children, other group properties could be reflected such as maximum attribute values. This is not part of the requirements in this application however.

Although not implemented in any glyph design within this chapter, other meta-data from the clustered child glyphs can also be encoded within the parent glyph design. As an example, Yang et al. encode maximum and minimum values as bands surrounding the mean values of aggregated child glyphs in a star glyph [YWR03]. Other meta-data that can be encoded include the standard deviations.

Glyph Legend: The glyph legend indicates a property of agent behaviour mapped to each of the bars protruding outwards from the glyph, along with an indication of the maximum size of each bar (Figure 4.6 right). The maximum extent of each bar is calculated from the largest value of the represented attribute from all agents. The colourmap applied to the bars are also visible below the glyph. The colourmap and length of bars is redundant in this design. A separate legend indicates the categorical colourmap for the centre of the glyph. The glyph legend is also used to display the mean values of the entire collection of call centre agents for each mapped attribute (**D2**).

Colourmaps: Based on feedback and discussion with the domain experts, the user has a choice from a selection of colourmaps to customise aesthetics. By clicking either the department colour legend, to the right of the plot, or the glyph bar colour legend, below the glyph legend, a choice of colour maps are shown for the user to select. Colourmaps were chosen from tested and established sources. Categorical colourmaps for glyph centres are collected from Colorgorical [GLS17], Tableau [Tab], and Telea [Tel14]. Colour maps for the data bars are based on sequential and diverging colourmaps from ColorBrewer [Breb]. Various examples of colourmaps can be seen in figures throughout this chapter. The application is designed to support any colour map requested by a user (**D1**). The default colourmaps used based on the Tableau 10

colourmap and the Cool to Warm colourmap developed by Moreland [Mor16]. The Tableau 10 colourmap is the default colourmap used in Tableau and was built around the human perception influenced CIELAB colour space [Sto16], while the Moreland colourmap is the default colourmap used in Paraview [San].

4.5.4. Glyph Legend Interaction

Interaction options are provided for glyphs to facilitate exploration, analysis, and selection of agent attributes (**D5**). By clicking an individual bar on the glyph legend, the selected bar on all glyphs in the scatterplot toggles between focus and context, as demonstrated in Figure 4.10. This option renders the selected bar with less opacity, removing the focus from that bar. User selection enables highlighting individual glyph bars and agent variables. A sub-set selection option is also provided that displays all glyph bars with either above or below average agent values for a given agent attribute (**T1**, **D5**). See Figure 4.6. This option applies to all bars across all glyphs. By utilising this option with the call-centre agent dataset, above or below average agent behaviour is easily observable.

Clicking the centre of the glyph legend returns a dialog from where agent property mapping of each of the glyph bar can be customised allowing the user to tailor the agent properties that can be seen and compared. Any of the attributes in List 2 on page 128 can be individually chosen. By default glyph variables are attributes indicated by the domain experts as most important. These are the variables that can be seen in Figure 4.6.

4.5.5. Filtering and Selection

At the request of the domain experts, we provide a filtering feature based on user-chosen departments, enabling focus to be placed on individual call centre departments (**T3**, **D5**). This focus + context filter removes glyphs according to the categorical variable to which the colour is mapped. With the call-centre dataset, we have mapped the agent's call centre department to this attribute. This is due to the importance of being able to compare agent behaviour based on department. Each agent department has a checkbox next to the colour map on the legend as can be seen in Figure 4.7. Selecting these places each glyph belonging to that category into focus or context.
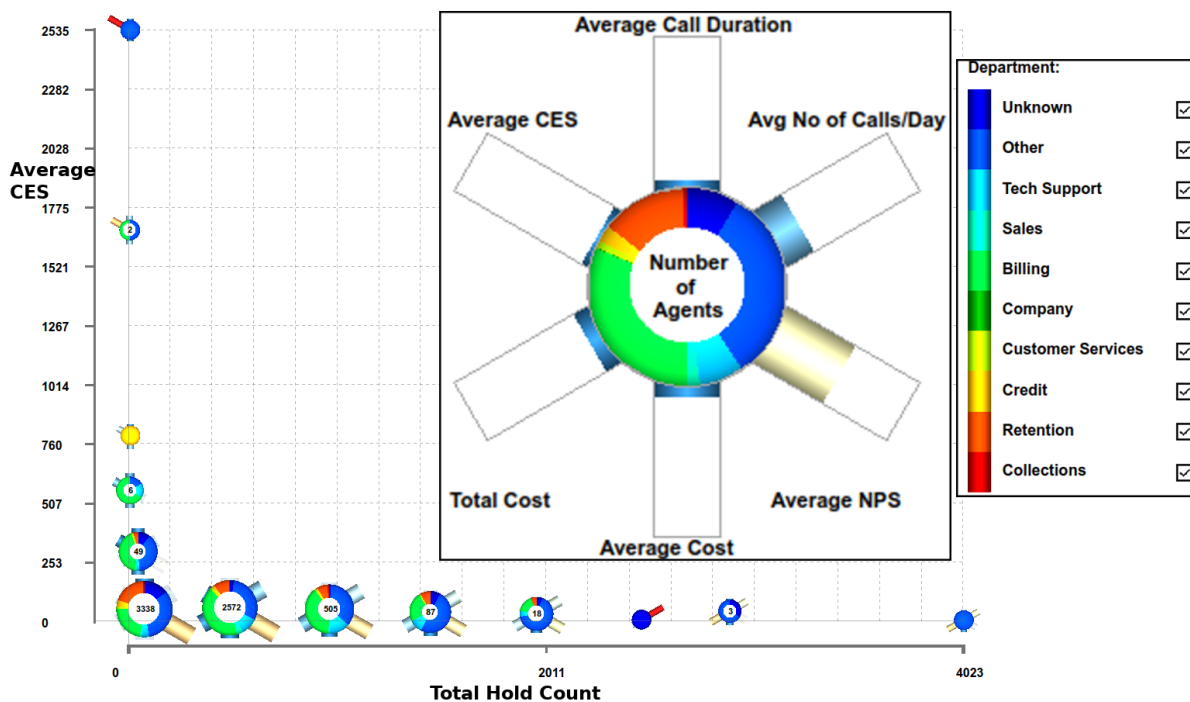
Figure 4.6.: Clusters plotted on a chart with total hold count along the x-axis and agent CES on the y-axis for over 6,500 agents (**R2**). Glyph bars below average have been removed, leaving only bars with above-average values. This highlights some interesting findings, the bar indicating average agent NPS, at the 4 o'clock position, is only visible on glyphs with a low CES and for agents with a low total hold count. Agents with a low total hold count also have a low total cost, as indicated by the absence of the total cost bar at the 8 o'clock position from those glyphs.

When a glyph is rendered as context it is displayed as a semi-transparent grey disk in the background and is not considered for clustering. If a particular category is filtered while already clustered, all parent glyphs are recalculated with the exclusion of the filtered category. An example of this selection and filtering can be seen in Figure 4.7 which plots the NPS against the number of cold transfer calls. An interesting outlier agent who has a low feedback score and a high number of cold transfers is highlighted with the on-mouse-over dialog. The glyph bar at the 8 o'clock position shows that a high number of warm transfers are also attributed to this agent. The on-mouse-over dialog indicates a large number of short calls and a very large number of holds, suggesting that these factors have contributed to the agent receiving a low NPS metric.

Glyph Rendering Options: The user can interactively adjust glyph scale. The user also has the option to map glyph size to the number of clustered agents represented by the given parent glyph (**D4**). Due to the potentially large number of agents clustered

into a single glyph, the size scale increases logarithmically with the number repre-sented. Shading is added to the glyphs, these effects have been used for improved aesthetics and to provide a 3D effect. This is achieved by utilising OpenGL lighting effects and varying normals across the glyphs.

Detail Options: To facilitate data exploration, on-mouse-over information is avail-able for each glyph providing original data values for that particular glyph (**D6, R4**). On clustered glyphs, the number of agents from each department represented in the cluster is provided. Figures 4.7 (bottom) and 4.9 show examples of the dynamic tooltips feature.

4.5.6. An Agent Distance Metric for Clustering

To determine if two glyphs should join to form a cluster, the x and y-axis component distances are tested along with the current zoom level of each axis to determine the Euclidean distance between each glyph in screen space (**D3**). If this is below the clustering distance (Equation 4.1) the glyphs are merged. Our customised clustering distance is as follows:

$$d(a) = s_x \frac{d_x}{d_{max}} + s_y \frac{d_y}{d_{max}} + s_z d_\alpha \qquad (4.1)$$

The components of the distance metric are:

- Euclidean distance (d_x): This constituent measures distance between two agents a_a, a_b in screen space using $d_x = |a_a^x - a_b^x|$, where s_x is a scaling factor in the x-direction controlled by the user.
- Euclidean distance (d_y): This measures distance between two agents a_a, a_b in screen space along y using $d_x = |a_a^y - a_b^y|$, where s_y is a scaling factor in the y-direction controlled by the user.
- Agent attribute (d_α): This constituent compares agent attributes such as de-partment, skill, or team between two agents a_a, a_b where $s_z = \infty$ IFF $a_a^\alpha \neq a_b^\alpha$ i.e. the user turns on clustering by department otherwise $s_z = 0$, $\alpha \in \{department, site, team, skill\}$

The user is able to control the distance, $d(a)$, at which clustering occurs via a slider in the interface. This enables control of the number of visible glyphs in the screen space and the amount of glyph overlap which is dependant on glyphs size, also adjustable
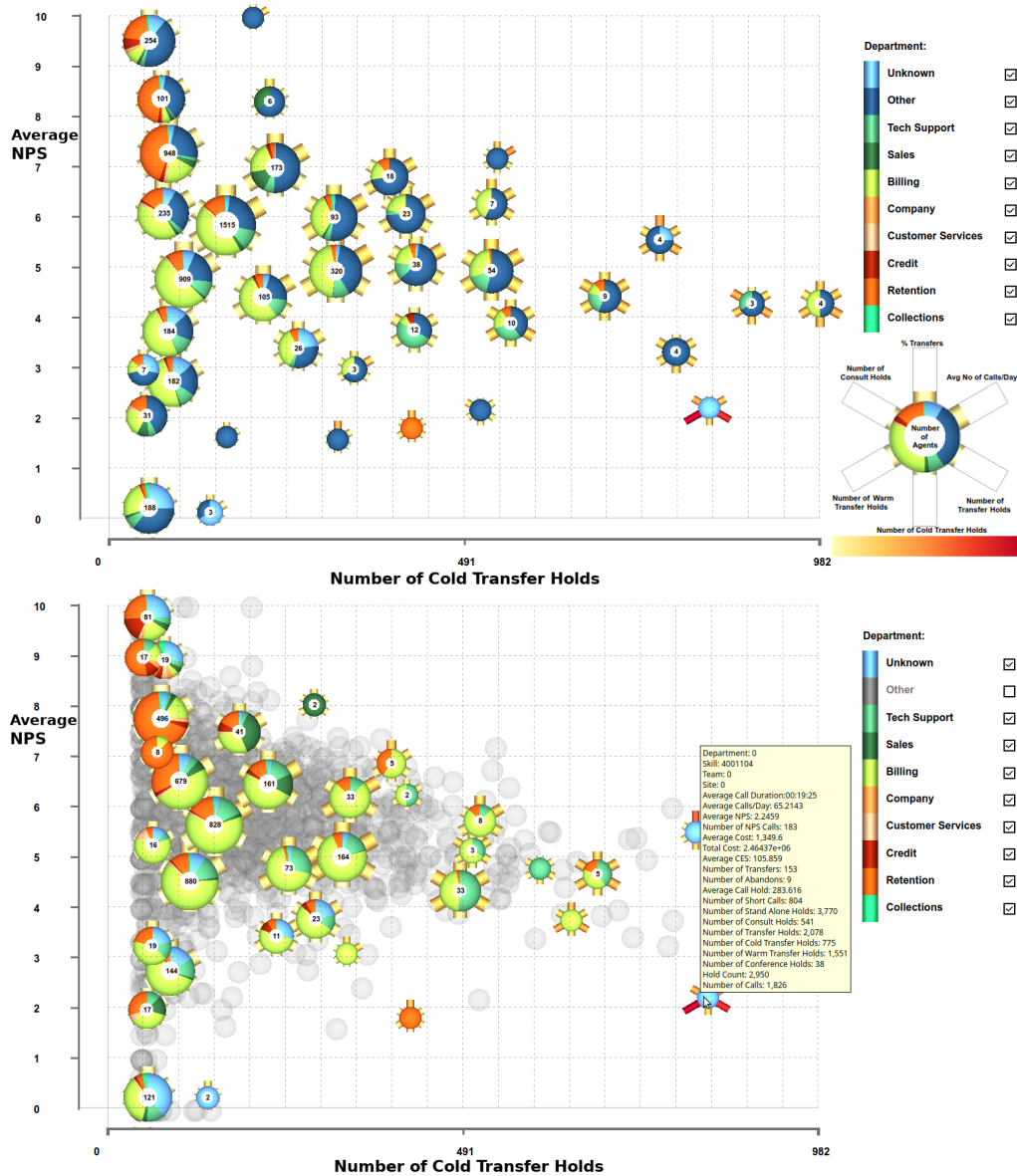
Figure 4.7.: Filtering of agent department. Both figures show the agent NPS on the y-axis
against the number of cold transfer calls on the x-axis. The top image shows all
departments while the bottom image shows the same scene with agents from
other departments filtered and rendered as a grey context in the background, see
Section 4.5.5. Mouse-over interaction is shown in the bottom figure, highlighting
an interesting, outlier agent with a low NPS score and a very high number of
warm transfers indicated by the bar on the 8 o'clock position of the glyph. The
on-mouse-over dialog also indicates a large number of short calls, in this case, calls
less than five seconds, and a very large number of holds, suggesting that these
factors contribute to a low NPS score.

by the user. By default, the clustering distance is set so that a balance is achieved
between the overlap and the glyph size. We also implement other clustering controls

incorporating feedback from the domain experts. Clustering can be customised to only occur if the agents share a common property, α. The user has a selection of categorical properties to choose from for this clustering constraint, in the case of the call centre agent, these are the agent department, site, skill level, and team. This enables the domain experts to concentrate on and compare specific categories of agents (**T3**). Examples of this can be seen in Figure 4.8, where agents are grouped by department, showing the average values for each call-centre department. Observable are the three distinct clusters of departments. The Sales and billing department cluster have almost double the hold duration in comparison to the credit, company, and collections department cluster, suggesting that effort could be made to reduce hold time in the sales and billing departments. Figure 4.9 shows a plot with clustering constrained to those within the same department, highlighting agents with different behaviour in comparison to others from the same department. A customer services agent has been highlighted with a high number of calls per day and a low percentage of short calls, unlike any other from the same department. Short calls are calls that last less than 5 seconds, too short a time to be productive, generally indicating undesirable behaviour from the agent.

4.5.7. Details vs Abstraction

Due to the nature of big-data, overplotting is a challenge, common to most visual designs. The use of clustering in this work is our solution to this challenge, however, as with most solutions, a compromise between detail and data abstraction is made. The more glyphs that are clustered and the data aggregated, the more abstract the glyphs become. The abstraction is linked to the number of child glyphs. As the user zooms in, the fewer glyphs in the screen space and therefore, the less abstract the representation. Overplotting becomes a greater challenge as datasets become larger, therefore, the clustered glyphs also become more abstract with growing datasets.

Glyph overlap is observable in Figure 4.9 (top) due to a constraint of glyphs only being able to cluster with those representing agents from the same department. However, by zooming in to the region indicated by the red box, these overlaps can be eliminated, with the resultant plot shown in Figure 4.9 (bottom). The zooming enables more detail to be seen, and reduces the overlap and abstraction.
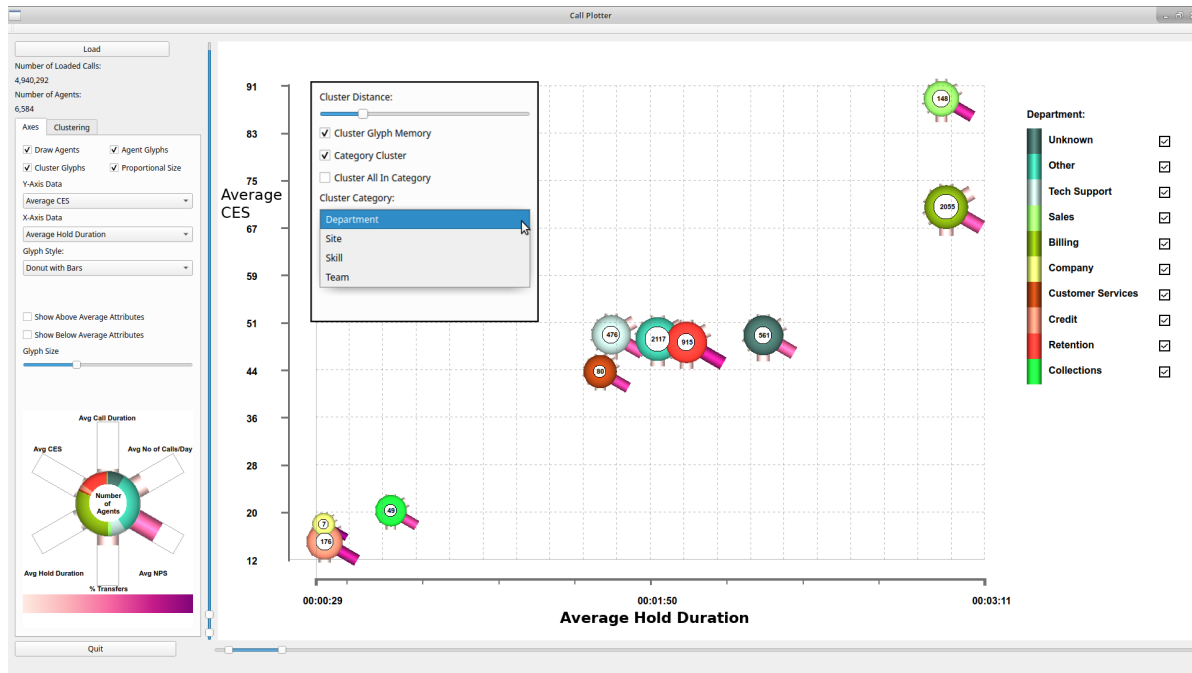
Figure 4.8.: A plot of average CES against average hold duration with agents grouped by departments. Notable are the three very distinct clusters of departments, with the company, credit, and retention forming a cluster with a low hold duration and CES, sales and billing in another high hold duration and CES, and the other departments lie in a cluster in between. See Section 4.5.6. Additionally, this Figure shows the application layout and controls. The insert shows additional clustering controls from the hidden 'Clustering' tab.

4.5.8. Clustering Operations and Placement

Glyph placement is driven by user chosen agent attributes on both the x-axis and y-axis, described by Ward as raw data-driven placement [War02]. Either continuous and non-continuous data can be mapped to both axes, which generally leads to overlapping of glyphs. By enabling interactive zooming of the scene, overlapping can be reduced as axes expand when zooming in. If axes variables are the same for overlapping glyphs, this will have little effect. Another method to redress overlapping is by changing opacity as described by Fuchs et al. [FJWS16], this is however limited by the number of glyphs that can be stacked before perception challenges arise. Alternatively, overlapping can be resolved by clustering glyphs into a single hierarchical glyph representation. This clustering method is equivalent to testing Euclidean distance between glyphs and merging glyphs within a pre-defined distance, Equation 4.1. This distance metric is interactively customised by the user (**R2**, **D3**). Once glyphs are clustered, the positioning of the new parent glyph is determined by the mean of
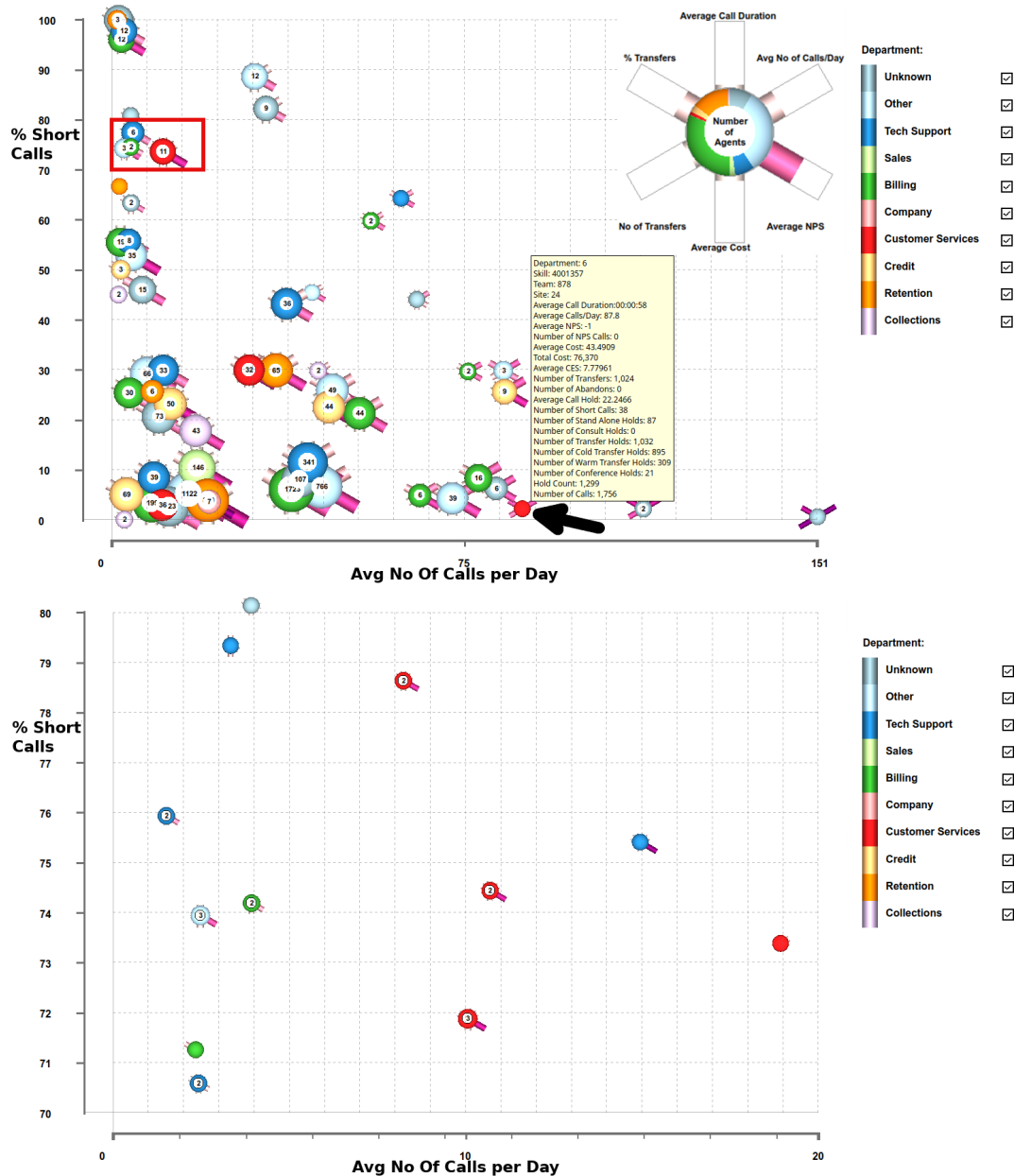
Figure 4.9.: (Top) A plot showing the percentage of short calls against the average number of calls per day. Clustering is specified to within departments only. Although this can result in some overlap, it enables outliers within departments to be easily discovered. A customer services agent is highlighted with an arrow and the on-mouse-over as such an example. No other agent from this department has the same combination of a high number of calls per day and a low percentage of short calls. (Bottom) A zoomed in view of the red highlighted section from the top, showing glyphs dispersed after zooming.

agent-based axis variables of all child glyphs. We call this data-driven hierarchical placement.

The glyph ordering when calculating Euclidean distance influences the result of the clustering. A glyph equidistant between two others could cluster to either of the other two, depending on which one was tested first, giving different clustering results. To ensure a constant clustering order, it is therefore important to ensure that glyph order and priority is consistent. To address this, the user has the option to sort glyph priority by any agent attribute in List 2.

4.5.9. Cluster Aggregation Options

Cluster aggregation can be achieved in a number of ways. Fuchs et al. [FJWS16] previously utilised a grid method, while Yang et al. [YWR03] demonstrate the use of a hierarchy tree to cluster glyphs. We discuss different strategies that dynamically test glyph distances at each zoom level and enable zooming on each axis independently. This is because agents may overlap heavily along one dimension only (and not the second one). This gives the user greater control and enables easier exploration of the data (**R1**, **D1**). This is exemplified in Figure 4.10, where the user has zoomed the x-axis, average hold duration, yet kept the full range of the y-axis, average NPS, to see the full distribution of NPS scores between a particular hold duration. Observable is the distribution of agent departments, with the retention department in sky-blue most prevalent at the high feedback score, while agents from the billing department are predominantly found with a lower score. We can also observe the inverse trend between average NPS score and average hold duration for the billing department. However there is an outlier billing agent with a high NPS score and longer than average hold duration, indicated with an arrow. This agent may deserve recognition.

Dynamic Hierarchical Glyphs   Dynamically hierarchical glyphs are interactively re-calculated at each zoom level or triggered by a user modification to the cluster distance metric. Parent glyphs are discarded at each frame, and new ones recomputed. This entails testing each glyph against every other glyph against the distance metric $d(a)$. The Pseudo-code 4.1 demonstrates this. The convention we use for pseudo-code is:

A lowercase 'g' represents a child glyph, e.g. g_t,

A lowercase, boldface '**g**' represents a list of child glyphs, e.g. $\mathbf{g}_{list}$,

An uppercase 'G' represents a parent glyph, e.g. G_t,

An uppercase, boldface '**G**' represents a list of parent glyphs, e.g. $\mathbf{G}_{list}$.
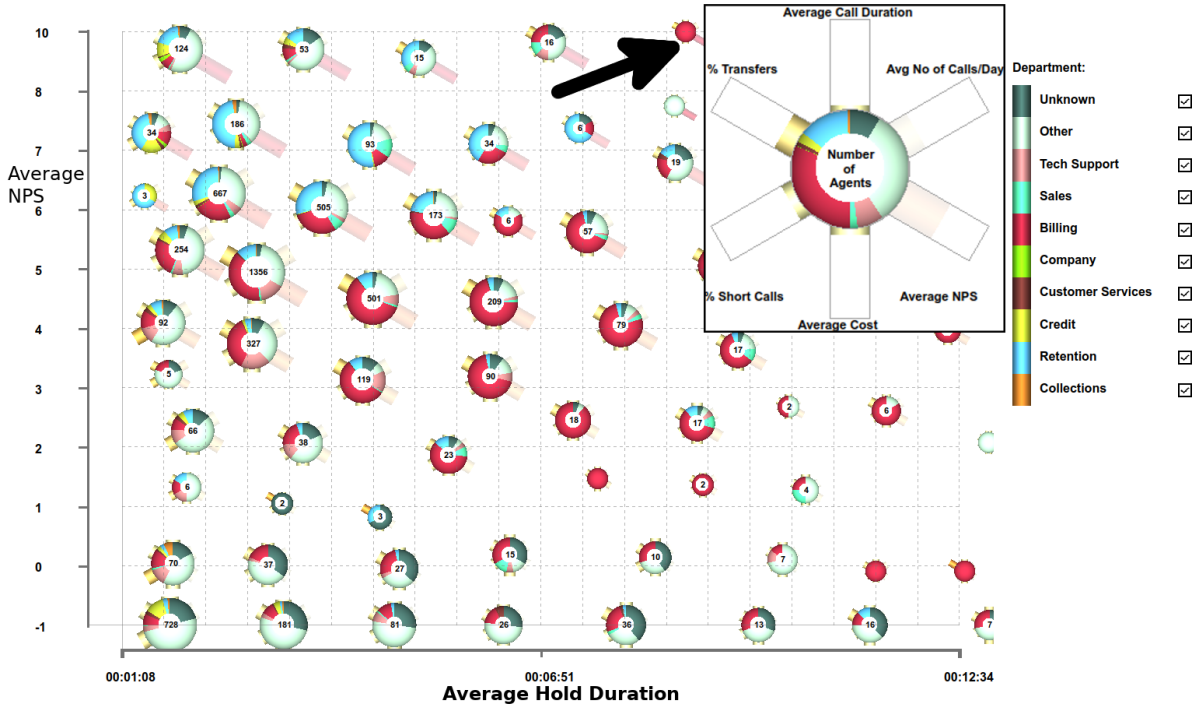
Figure 4.10.: A figure displaying the agent NPS score on the y-axis and the average hold duration on the x-axis. The user has zoomed the x-axis to a hold duration between 1 minute and 12.5 minutes while keeping the y-axis at the maximum extents to explore the full NPS distribution. The user has also filtered the 2 and 5 o'clock bars, average NPS and number of calls per day, from the glyph. We discover that the agents belonging to the billing department, in red, are primarily concentrated below an NPS score of 7, with few agents achieving a higher customer satisfaction score. In contrast, agents representing the retention department, in sky-blue, primarily have an NPS of over 5, with few agents receiving a low customer satisfaction score. An outlier agent is indicated with an arrow. See Section 4.5.9

While using this technique, although it is fast, it was found that it could be confusing to the user as parent glyphs disappear and new parent glyphs reappear when the scene changes. Thus, in order to increase cohesiveness, we developed the next version that preserves parent glyphs between frames.

Dynamic Hierarchical Glyphs with Parent Preservation   To alleviate the confusing experience from the dynamically created glyphs, we modify the algorithm to include the memory of parent glyphs between frames. Initial merges are calculated as previously outlined for dynamically created glyphs to create a list of parent glyphs. On subsequent scene changes, parent glyphs are preserved. Instead, each child glyph within each parent glyph is re-tested based on the current distance, $d(a)$. Glyphs above

```
Input Variables:
g_list       List of all agent glyphs
d            Glyph distance threshold, Eqn 4.1
Local Variables:
g_t          Current glyph being tested
g_c          Candidate glyph for clustering
g_clustered  List of glyphs already clustered
g_tested     List of glyphs already tested
g_parent     List of glyphs clustered to a parent
Output Variables:
G_list  List of parent glyphs

FOR EACH g_t ∈ g_list //for each child glyph
IF g_t ∉ g_clustered //if glyph is un-clustered
FOR EACH glyph g_c ∈ g_list
IF g_c ∉ g_tested AND g_c ∉ g_clustered
IF |g_t − g_c| < d
g_parent·add(g_c)
g_clustered·add(g_c)
END IF
END IF //g_c ∉ g_tested AND g_c ∉ g_clustered
END FOR
END IF //g_t ∉ g_clustered
g_tested·add(g_t)
IF g_parent NOT ∅
g_parent·add(g_t)
G_list·add(g_parent)
END IF
g_parent·clear()
END FOR
```

Pseudo-code 4.1.: Dynamic hierarchical glyph clustering

the distance are individually removed from the parent glyph. If a parent glyph is left with a single child, the parent glyph is replaced by the child.

Once this has been updated, each parent glyph is tested against every un-merged glyph, and every other parent glyph, with glyphs being added to the parent if they fall within the threshold distance $d(a)$. Finally, all un-merged glyphs are tested against each other to potentially create new parent glyphs. Using this algorithm, we are able to support interactive exploration of a dataset containing over 6,500 agents. The process is outlined in Pseudo-code 4.2.

Yang et al. previously utilised a pre-computed binary hierarchy tree [YWR03] to determine what glyphs are shown at each zoom level. However, as we provide independent zooming on each axis, a single hierarchy tree is more difficult. A hierarchy tree would be required for every zoom point on an axis, this would theoretically be an

```
Input Variables:
g_list      List of all agent glyphs
g_clustered List of glyphs already clustered
G_list      List of parent glyphs
G_clustered List of parent glyphs
d           Glyph merging distance, (Eqn 4.1)
Local Variables:
g_t         Current glyph being tested
g_c         Candidate glyph for clustering
G_t         Current parent glyph being tested
G_c         Candidate parent glyph for clustering
g_tested    List of glyphs already tested
G_tested    List of parent glyphs already tested
g_parent    List of glyphs clustered to a parent
Output Variables:
g_list      Updated list of all agent glyphs
g_clustered Updated list of glyphs already clustered
G_list      Updated list of parent glyphs
G_clustered Updated list of parent glyphs

FOR EACH G_t  ∈ G_list // Remove children too far away
FOR EACH glyph g_t ∈ G_t
IF |g_t − G_t| > d
G_t.remove(g_t)
G_clustered.remove(g_t)
END FOR
END EACH
FOR EACH G_t  ∈ G_list // Test for parents and children
// to cluster
IF G_t ∉ G_clustered
FOR EACH G_c  ∈ G_list
IF G_c ∉  G_tested
IF G_t ∉ G_clustered
IF |G_t − G_c| < d
G_t.add(G_c)
G_clustered.add(G_c)
END IF
END IF
END IF
END FOR
FOR EACH glyph g_t ∈ G_list
IF g_t ∉ G_clustered
IF |g_t − G_t| < d
G_t.add(g_t)
G_clustered.add(g_t)
END IF
END IF
END FOR
END IF
G_tested.add(G_t)
END FOR
FOR EACH g_t ∈ G_list //Test for new parents
IF g_t ∉ G_clustered
FOR EACH g_c ∈ G_list
IF g_c ∉ G_tested AND g_c ∉ G_clustered
IF |g_t − g_c| < d
G_parent.add(g_c)
G_clustered.add(g_c)
END IF
END IF
END FOR
END IF
G_tested.add(g_t)
IF G_parent NOT ∅
G_parent.add(g_t)
G_list.add(G_parent)
END IF
G_parent.clear()
END FOR
```

Pseudo-code 4.2.: Dynamic hierarchical glyphs with parent preservation clustering

Glyph	x-axis	y-axis	var 1	var 2	var 3	var 4	var 5	var 6
α	1	1	1	1	3	3	8	8
β	3	3	3	8	8	1	1	3
γ	8	8	8	3	1	8	3	1
$\alpha\,\beta$	2	2	2	4.5	5.5	2	4.5	5.5
$\alpha\,\beta\,\gamma$	4	4	4	4	4	4	4	4

Table 4.2.: Synthetic data and position variables for three glyphs, and two clustered glyphs as depicted in Figure 4.11.

uncountable set of zoom levels, however, a fixed number of resolution zoom points could be set. To pre-compute all possible trees in a large dataset would require a large amount of computing resources and would be impractical, thus we have not implemented such a solution.

Clustering Example   A simple clustering example is demonstrated in Figure 4.11. Three circular glyphs are depicted, each with six continuous variables and an x and y-axis position as indicated by Table 4.2. Three charts are displayed with glyphs at different stages of merging.

Chart A in Figure 4.11 shows the three individual child glyphs as indicated by the first three lines of Table 4.2. Chart B shows the clustering of the two nearest child glyphs (top two rows of Table 4.2) creating a parent glyph with the data indicated on the fourth row of Table 4.2. The variable values for this parent glyph represent the average of the two clustered child glyphs. Chart C shows all glyphs clustered into one parent glyph with the position and all variables showing the average value of all child glyphs as shown on the fifth row of Table 4.2.

Smooth, Interactive Transitions   To enhance the perception of two glyphs joining together and clustering, an animated transition is implemented that lets the user follow the paths of child glyphs to a parent cluster, (**R2**, **D4**). This prevents a visual disconnect that arises if glyphs disappear and parents appear incoherently. An animation is also used for splitting of parent glyphs for the same reasons. The animation is achieved by dynamically calculating the position of a clustered glyph and using this as a target location for the children that are merging at that time. Glyph positions are linearly interpolated and updated to support dynamic motion this target over a short time.
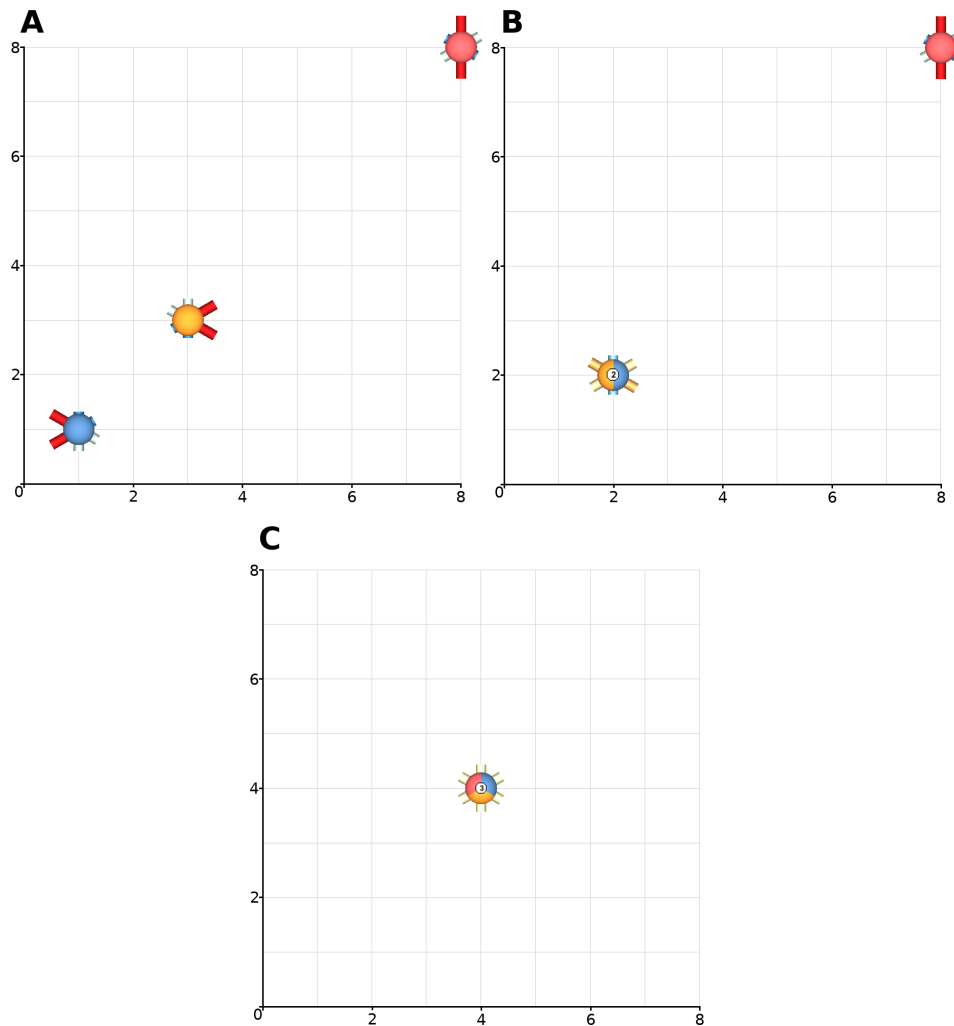
Figure 4.11.: A demonstration of clustering glyphs using synthetic data. In chart **A** three individual child glyphs are displayed. Chart **B** shows the merging of the two nearest glyphs, creating a parent glyph with it's position and geometry updated to an average of the composing children glyphs. Chart **C** shows the clustering of all three child glyphs from chart A with it's position and geometry representing the average of these glyphs.

Figure 4.12 shows an image of transitions in motion. To view the smooth animated transitions, see the accompanying **video** [Reea].

4.6. Evaluation

We evaluate the application by utilising its features to investigate the relationship between CES and call duration. We learn that this varies by department, suggesting
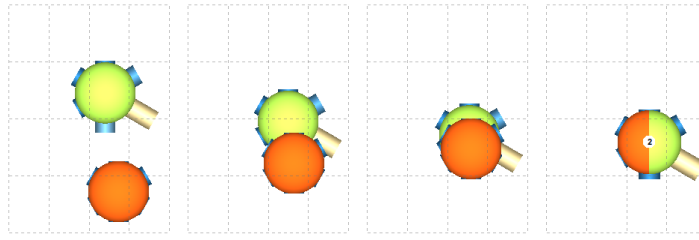
Figure 4.12.: A figure displaying four frames from a smooth animated transition of agent glyphs, representing different departments, following user zooming. See the accompanying video for animation [Reea]

an allowance should be made in the computation of CES for the department. Further observations are available in the accompanying **video** [Reea]. Following the usage scenario, we report feedback from domain experts.

4.6.1. Evaluation Justification

Our evaluation comprises of an industry-inspired usage scenario and reporting of feedback from domain experts. We deem this to be an appropriate evaluation, as this application is built for special-purpose domain experts in the call centre analytics industry, and not more general users. This also reflects guidelines set out by Munzner [Mun09b]. Although an improved evaluation could be achieved with a long term field study, this could be a direction for future work.

4.6.2. Usage Scenario – Customer Effort Score Validation

In addition to examples of new knowledge described previously, an industry-driven usage scenario is provided to demonstrate the utility of the application.

The usage scenario is a demonstration of a workflow that an analyst may apply while using the application. The typical workflow for the application follows Shneiderman's Visual Information-Seeking Mantra of overview first, zoom and filter, then details on demand [Shn03]. With the application this involves first utilising the scatterplot matrix to find correlating variables and to select the focus scatterplot axis variables. Other data variables of interest can then be mapped to the glyphs to facilitate exploration of the data. The clustering, along with the filtering and zooming provides clutter reduction. Finally the mouse-over feature displays individual agent details.

The findings demonstrated in this usage scenario represent real knowledge derived from the use of the AgentVis application whilst exploring of the data.

Identified as of particular interest by our domain experts is Customer Effort Score (CES), a metric derived to asses the amount effort the customer made in engaging with the call centre. As the score is a derived value, our industry partners are interested in assessing the validity of the score, particularly in relation to how the score varies across different departments and how this compared to the customer feedback Net Promoter Score (NPS), considered as a ground truth indication of customer satisfaction from surveys. We demonstrate a typical workflow to investigate how the CES and NPS correlate, as inspired by the domain experts and the tasks that they would typically preform.

To investigate this, a dataset incorporating a month's worth of calls, close to 5 million, is used. We first selected the average agent CES on the y-axis, and place the average call duration on the x-axis (**R1**). This leads to a very cluttered view so we, therefore, utilise the cluster feature for ease of exploration (**D3**). This leads to a single parent glyph based in the lower left of the scatterplot representing the vast number of agents, with a few other clusters and single agents spread across the graph. This indicates that the distribution of the data is in the lower values of each axis, thus we zoom each axis independently to focus on the main body of the data (**R4**).

We select the option that only clusters agents if they share a common department to investigate the differences between departments (**T3**) and adjust the clustering distance metric to find a balance between screen space utilisation and clutter (**R2**, **D4**). To reduce clutter further we filter departments to only show agents from the credit, collections, and customer services departments as these departments are of particular interest to the user.

As we want also to investigate how the CES compares with NPS, we ensure that the NPS is mapped to one of the glyph bars, along with other variables of interest (**R3**). We select the option to filter bars that show below average variables. The resulting plot can be seen in Figure 4.13.

From the image, we learn that average call duration is a driver of CES, as expected, with a general trend seen in the filtered agents rendered in context, and in the focus agents. We observe that in agents from the customer services department, the call duration has a greater effect on the CES compared to agents from the collections department who increase at a shallower incline. Agents from the credit department,
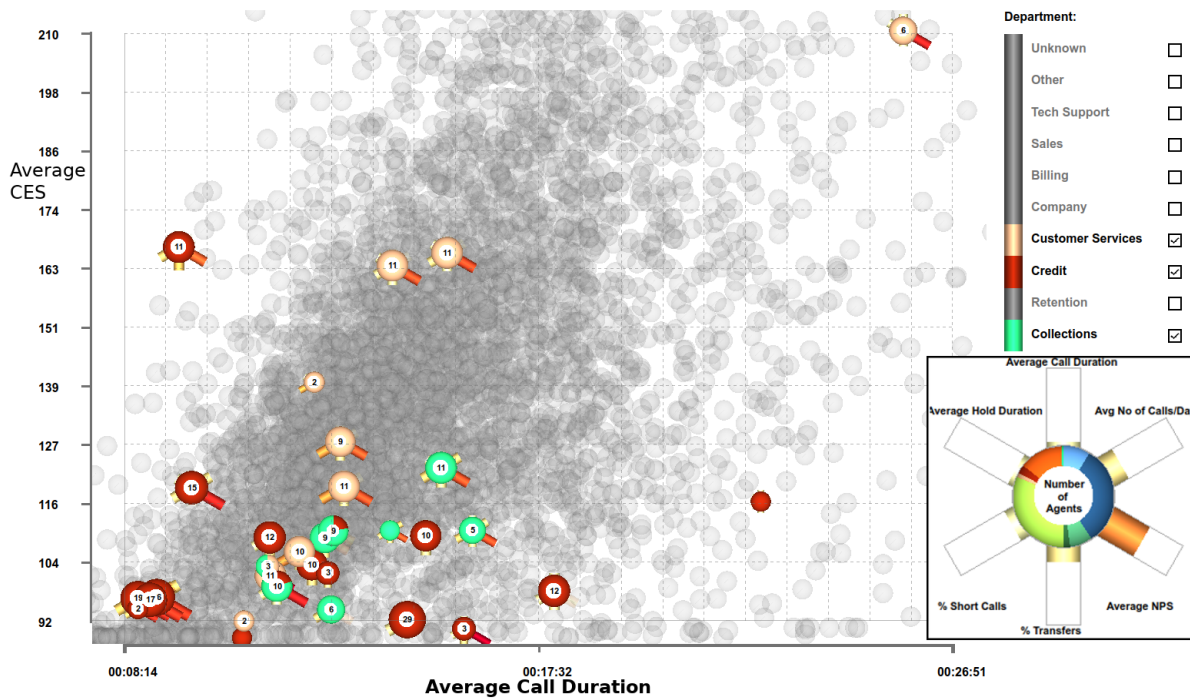
Figure 4.13.: A figure from a usage scenario investigating the effectiveness of CES. Average agent CES is on the y-axis and the average call duration on the x-axis. Agents have been filtered to only display those from the credit, collections, and customer services departments, and have been clustered into their teams. See Section 4.6.2 for more information.

however, do not seem to follow this trend with no obvious correlation between call duration and CES.

When comparing the NPS, mapped to the 5 o'clock position on the glyphs in Figure 4.13, we observe that agents from the credit department generally have higher values. No apparent correlation is evident between either axis and the NPS value. Also evident in the figure is the presence of outlier agents such as the lone credit agent, with longer call duration in relation to other credit agents (**T2**). This agent is also shown to have a low NPS compared to other agents (**T1**).

To investigate the relationship between NPS and CES for each department further, we map the agent NPS on the *y*-axis and the agent CES on the *x*-axis. We hypothesise that call duration, hold duration and the number of transfers have the most significant effect on customer satisfaction. Therefore we ensure that the average call duration, the average hold duration, and the percentage of calls transferred are all mapped to glyph bars.

We filter through different individual departments to search for correlation. To reduce clutter, we cluster agents by their teams. From the resultant sales department plot, we can see a strong negative correlation between the NPS and CES, with an increase in CES returning a decrease in NPS. From comparing glyph variables, we notice that a relationship also exists between the bar mapped to the agent hold duration. It also shows a strong negative correlation to the NPS, however, this is not the case for the percentage of transferred calls and call duration. This tells us that for the sales department, the hold duration is particularly important.

Similarly, a negative correlation is observed between the NPS and the CES for the retention department. Also observable for the retention department is a correlation between the NPS score and the glyph bar representing the percentage of calls transferred, where teams with a large percentage of transfers tend to have a higher CES and a lower NPS (see Figure 4.14). This informs us of the importance of the number of transfers for the retention department. These examples are shown in the accompanying video [Reea].

From this usage scenario, we can conclude that average call duration is a driver of the CES, however special consideration should be given to the department being contacted. It is possible to conclude that different departments have particular properties that influence customer satisfaction NPS.

4.6.3. Domain Expert Feedback

To evaluate the application, we conducted demonstrations and interviews with QPC Ltd., the data suppliers, throughout the development of the software. Their input guided development and provided regular feedback on the application. Here we provide some of their feedback.

A demonstration of the software was first conducted to introduce the clustering concept used, and to exhibit the interaction options available to the agent, highlighting some agent insights found during development.

This was the first time any of the experts have ever seen interactive exploration of 6,500 call centre agents simultaneously. During the demonstration, calls had been clustered into parents with individual agents left scattered, expert three observed that it was easy to find unusual agents (**T2**): *"It's good for finding outliers in this sort of view."*
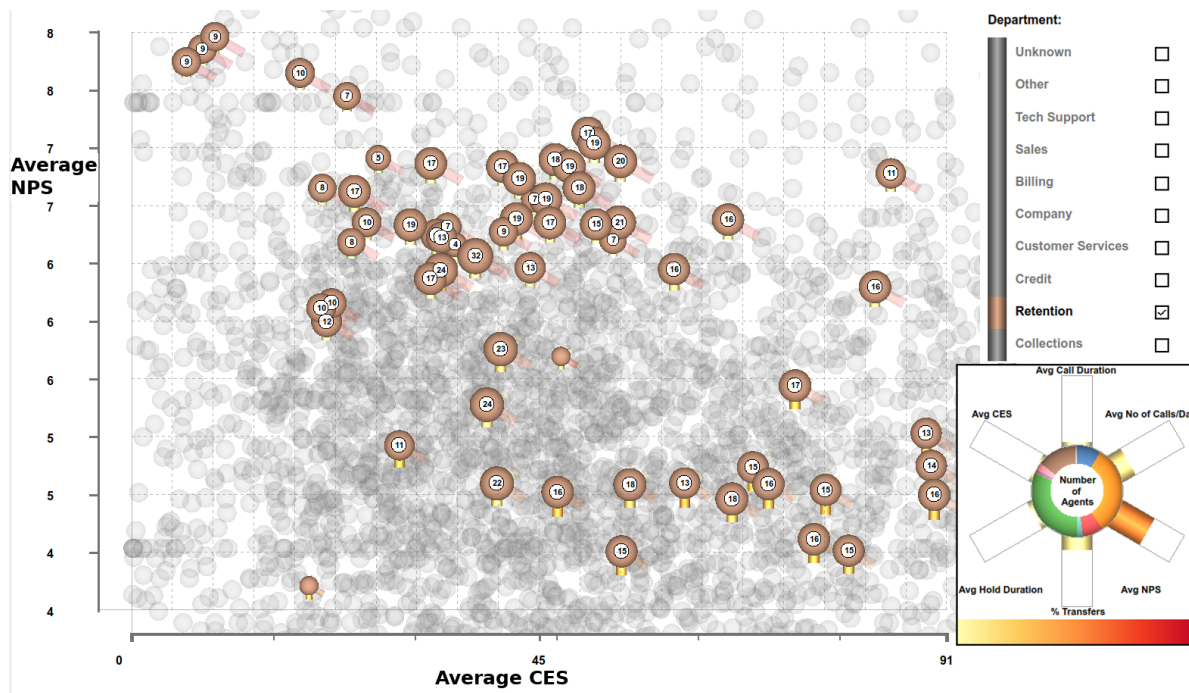
Figure 4.14.: A figure from a usage scenario investigating the effectiveness of CES. Average agent CES is mapped to the x-axis and the average NPS is mapped to the y-axis. The axes have been zoomed to focus on the unfiltered data. Agents have been filtered to display only those from the retention department, and have been clustered into their teams. The 6 o'clock bar on the glyph is selected to show the percentage of calls transferred and the other bars are rendered as context with lower opacity. This allows the user to observe that the teams that have a lower NPS also tend to have a higher percentage of call transfers. See Section 4.6.2 for more information.

In particular, an outlier agent was discovered that was not clustered. The identified agent had particularly poor performance with a low feed back score and where 30 calls out of 31 made were less than 10 seconds in duration, too short to achieve anything useful.

The experts gave positive feedback for the scatterplot-matrix view, and the overview it allowed onto the data (**D2**). Expert three commented: *"I like the high level view that gives"* Expert one, meanwhile, noticed the correlation indication commenting: *"There's a lot of things here that you'd expect to correlate... Yeah, it's good, I think it's a good initial view... The ability to click on it and go view, filter to that dataset is really good."*

The experts were explicitly asked for their opinion of the available glyph designs, to which expert one replied: *"I think, of the three, the donut one is definitely the best for*

me the others I'm not as fussed on, but I think it's just good to have options... I'm sure other people have different preferences."

When exploring individual agents, expert two appreciated the on-mouse-over function (**R4**), allowing him to see all of the call details of a standout agent: *"That agent you hovered over, more than half of their calls are less than 10 seconds... that isn't enough time to establish a need to transfer. That's an interesting one."*

After demonstrating the ability to filter call centre departments, and to cluster agents based solely on their team, skill or site (**T1**, **T3**), expert one commented: *"I like the fact you can select the departments now, that's great, that's really useful because you're going to get a similar pattern of calls for all the agents within that one department. The different departments are going to have different patterns, so to be able to look at one or even compare two or all of them at the same time. That's great."*

When shown the clustering function reducing the clutter on a plot with over 6,500 agents, expert two showed his approval: *"That's a good idea. I like that. It makes it more practical."* After a few minutes of exploration, expert two then declared: *"It's all very interesting, you can definitely see some patterns in there."*

Following the demonstration, structured questions were asked to ascertain the expert's opinions of the application. When asked if they could see a benefit of for the software, expert two replied: *"Yeah, I can. So given that, on the drop downs on the left you can change the axes and change the different values on the glyph on the different clock positions, there's a wide variety of things you could do with this."* This indicates the expert's appreciation of the software's ability to display multiple agent attributes (**R3**). Expert two continued: *"The whole thing looks really good to me, and it seems really flexible. It looks like it does an awful lot. I really like it."* Underlining the expert's enthusiasm for the application and its abilities (**R1**).

Expert one appreciated the application's ability to visualise a dataset comprising of all calls for an entire month, with thousands of agent represented (**R2**), something that they had previously been unable to achieve: *"It really does let you look at a big dataset in a way that we, well personally what I've used in the past, hasn't been able to do. We can definitely find stuff that I didn't know before."* Expert one also provided a spontaneous comment for the aesthetics of the application: *"I think it's very visually appealing as well, it looks unique, it looks different to the usual sorts of visualisations you get, it does grab your attention that it's different."*

Feedback was garnered for subsequent versions of the application, with expert two keen to enable exploration of features not available in the current dataset: *"The dataset that you've got within this is not as rich as some of the datasets that we've got, unfortunately, so you are limited in what you can possibly display because the underlying data isn't in the format you need it."* Ensuring that the richer datasets are compatible with the software and that new properties and variables are utilised is future work for the project.

Additional feedback was gathered from a front-line call-centre agent, to provide a different perspective on the software. This domain expert has over two years experience as a call-centre agent working within a sales and customer services departments at Admiral Group plc. Feedback was again garnered from a guided interview following guidelines by Hogan et al [HHH16]. The domain expert provided feedback in a two hour session, initiated with a demonstration of the software.

On being shown the initial scatterplots the agent noted some agent behaviour reflected in the data. *"The scatterplot is useful for identifying agents that are only taking a very low number of calls. This suggests that they are unavailable."* Availability of agents is very important to operations.

The zooming of the scatterplots was also praised, *"I think it's very good because then you can see what each agent is doing."* and *"I like that because it's so easy to understand."*

While exploring the data, the agent was noticing behaviours that occurs within the call centre, while exploring short calls he noted, *"Calls could be shortened due to end of shift."* This new hypothesis requires follow-up investigation to be confirmed. Another insight provided was relating to customer behaviour when contacting the call centre, *"The sales department line is usually free. However, the customer service line tends to charge a fee. So you'll have customers calling the sales line in order to speak to the customer service department."*

The agent was asked for his overall thoughts about the software and how it would supplement his current workflow, *"To supplement with this would be perfect. It shows you exactly what you are doing, where you can improve, not only that, different departments different sites, for competition purposes because of different call centres, a lot of them use incentivisation of performance, so you do have all that data, you can see that. It's quite visible."* *"The clustering by department is definitely useful. Companies want to see which departments and sites are performing the best."* *"It is useful... it has to be very easy to understand (for managers and analysts) and very quick with minimal cost in time."*

Study Evaluation

This study led to the creation of a tool for the exploration of call-centre agent behaviour. The tool has since been supplied to the collaborating company for testing. However the tool has not been incorporated into their daily workflow due to limited developer resources to commercialise it or perform field evaluations. This challenge is a pitfall of a design study collaboration, highlighted by Sedlmair et al. [SMM12], of insufficient time available from collaborators (PF-5).

A reflection of the experience on this project with the other pitfalls indicated by Sedlmair et al. can be seen in Table 4.3, it reveals that most pitfalls were avoided and some new ones encountered. Challenges were, however, encountered with the task not being regularly preformed preventing a full validation of the application (PF-10).

Other challenges identified from the list identified by Sedlmair et al. where improvements could be made is with the reporting of the domain expert feedback, where the experts involved with the design also provided feedback (PF-26). Although not explicitly objective, feedback was balanced with criticisms as well as praise for the design received. Another potential pitfall suffered by his study is the lack of a long term user study before reporting (PF-32). This is as a result of PF-5 as previously discussed.

A new pitfall this project encountered was the departure of a collaborating domain expert towards the end of the project. This pitfall isn't explicitly highlighted by Sedlmair et al., therefore we recommend adding this as the 33rd pitfall PF-33. Since the departure was near the end of the project, an additional expert was found to only provide feedback. This pitfall would have had a greater impact had it happened earlier, however, disruption was minimised.

Feature creep is another new challenge encountered [McC96], where domain experts would request many additional features once a prototype had been seen, expecting a tool that is able to preform additional tasks We identify this as a new pitfall (PF-34). Some requests were easily implemented, such as UI adjustments. Other requests were discussed and a triage process carried out to determine a priority list of features. An example of a requested feature not implemented, was for a mechanism to show customer calls handled by a user-chosen agent, as halo glyphs encircling the agent glyph.

PF-1	Premature advance: jumping forward over steps	-	Procedure followed
PF-2	Premature start: insufficient knowledge of vis literature	-	Expertise in vis available
PF-3	Premature commitment: collaboration with wrong people	-	Collaboration established for number of years
PF-4	No real data available	-	Data available from outset
PF-5	Insufficient time available from potential collaborators	+	No time/support for full deployment
PF-6	No need for visualisation: problem can be automated	-	Visual analysis required
PF-7	Researcher expertise does not match domain problem	-	Expertise available
PF-8	No need for research: engineering vs. research project	-	Research project
PF-9	No need for change: existing tools are good enough	-	No visual tools available
PF-10	No real/important/recurring task	+	Tasks that application preform are not regularly preformed
PF-11	No rapport with collaborators	-	Suitable rapport built over previous projects
PF-12	Not identifying front line analyst and gatekeeper before star	-	Analyst and gatekeeper identified
PF-13	Assuming every project will have the same role distribution	-	Issue not encountered
PF-14	Mistaking fellow tool builders for real end users	-	Issue not encountered
PF-15	Ignoring practices that currently work well	-	Current practices not adequate
PF-16	Expecting just talking or fly on wall to work	-	Intermediate meetings held to check relevance
PF-17	Experts focusing on visualisation design vs. domain problem	-	Although consulted, designs were introduced by researchers
PF-18	Learning their problems/language: too little / too much	-	A balance was though to have been achieved
PF-19	Abstraction: too little	-	Abstraction deemed suitable
PF-20	Premature design commitment: consideration space too small	-	Range of designs considered before focusing
PF-21	Mistaking technique-driven for problem-driven work	-	Technique driven by tasks
PF-22	Non-rapid prototyping	-	Initial design implemented quickly and honed once approved
PF-23	Usability: too little / too much	-	Careful design and testing of controls used during implementation
PF-24	Premature end: insufficient deploy time built into schedule	-	Time available
PF-25	Usage scenario not case study: non-real task/data/user	+	Some reported discoveries made by developers
PF-26	Liking necessary but not sufficient for validation	+	Domain experts were linked closely during design, however, some criticisms were still reported
PF-27	Failing to improve guidelines: confirm, refine, reject, propose	-	Guidelines for implementing hierarchicalisation of glyphs provided
PF-28	Insufficient writing time built into schedule	-	Sufficient time allowed
PF-29	No technique contribution $\neq$ good design study	-	Recommendations by Sedlmair et al. [SMM12] used
PF-30	Too much domain background in study	-	Restraint used in background description
PF-31	Story told chronologically vs. focus on final results	-	Chronological reporting not used
PF-32	Premature end: win race vs. practice music for debut	+	Study could improve with long term deployment of tool and analysis of use
PF-33	**Departure of collaborator**	+	Collaborator departed near end of project
PF-34	**Feature creep**	+	Collaborators asking for new features
PF-35	**Expectations pitfall**	+	Collaborators asking for new complex features

Table 4.3.: A table of pitfalls identified by Sedlmair et al. [SMM12] and their relevance to this project. '-'indicates a pitfall we did not encounter, '+' indicates pitfalls we did encounter. Pitfalls in **bold** font indicate new pitfalls experienced.

Linked to this is the expectation of features being added without considering the development time required for each feature. We call this an expectations pitfall (PF-35). For the customer halo glyph request, this feature would require substantial development with additional data analysis, further customer metrics, a separate glyph design, and interface mechanisms. To temper these expectations, we recommend discussing the development investment requirements for such features, as was done in this case.

Although not necessarily a pitfall, a minor boundary to the collaboration was the physical distance between the researchers and domain experts. With a 3–4 hour travelling distance, collaboration was not as fluid as it could have potentially been, despite the use of conference calls. Had the collaborators been more physically accessible a closer collaboration could have been conducted.

4.6.4. Conclusion

In this chapter, we utilise multivariate glyphs in a scatterplot layout to visualise call centre agent behaviour. We demonstrate an application with a month's worth of call centre data, representing 6,500 agents handling nearly 5 million calls. To address clutter we cluster glyphs using a customised clustering algorithm based on Euclidean distance, and extensions of this to cluster agents by their department, site, skill, or team. This represents a novel solution to the problem of overplotting for glyph based layouts and we provide guidelines for the hierarchicalisation of glyphs. We report domain expert feedback on the application created to visualise the data, gathered from several interview sessions.

Application Generalisation

The techniques developed and presented in this study can be generalised to other domains and datasets. Many of the requirements established in Section 4.4 are not unique to call-centre agent behaviour domain with large multivariate datasets being ever-more prevalent. The glyph clustering algorithm is applicable to other glyph designs and datasets, as are the interaction techniques. The application specifically developed for this study, however, requires some modifications in order to be used in other domains due to customisation of the given data format and domain specific attributes.

Chapter 5.

Interaction Techniques for Chord Diagrams

*"To know that we know what we know, and to know that we do not know
what we do not know, that is true knowledge."*
— Nicolaus Copernicus – Polymath (1473–1543)

Following the call centre agent based visualisations, an agent attribute not possible to be displayed using the glyph technique presented in Chapter 4 is call transfers between agents. A natural design to visualise transfer data is a chord diagram. This enables for the utilisation of experience gained developing the prototype visualisation presented in Chapter 1. Challenges again exist with the issue of scale and with overplotting again. This chapter is based on a paper accepted for publication at the 24th International Conference Information Visualisation [RLBD20].

5.1. Introduction

Chord diagrams are a popular high-dimensional visualization for representing relations. Their simplicity and the intuitive design results in common use in mainstream media to visualize affiliation [Cli18]. The design and layout of chord diagrams are necessitated to be radial as all nodes can be directly connected to all other nodes, without crossing a third node.
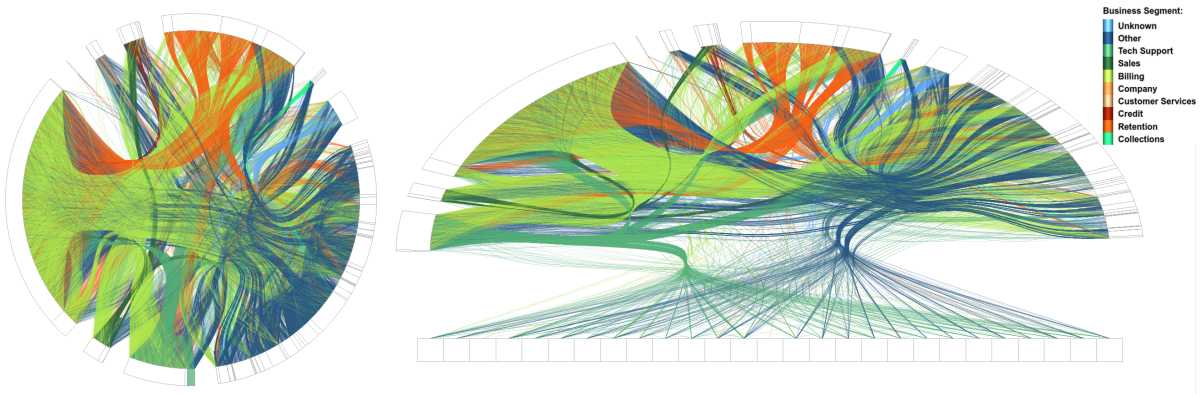
Figure 5.1.: (Left) A chord diagram of calls transferred between agents categorised by department of a call centre. An edge bundling technique is utilised to reduce clutter [Hol06]. A department within the Tech Support business segment is highlighted at the six o'clock position of the chart. (Right) A deformed chord diagram, with the user-chosen Tech Support department from the left image expanded to visualise calls transferred between individual agents.

Chord diagrams are often used to convey hierarchical data, where axis nodes along the circumference are placed into groups. Holten utilises a hierarchical dataset to address another limitation of chord diagrams, namely scalability [Hol06]. This solution offers user adjustable bundling to reduce clutter. This solution does not however, fully address the scalability problem as the addition of more nodes will increase the difficulty of identifying individual nodes.

Other interaction techniques used with chord diagrams include filtering and axis ordering for optimisations and data reasons [GK06, BOH11]. From an interaction point of view, chord diagrams generally lag behind other multi-variate visual designs such as parallel coordinates which offer many advanced interaction techniques [RLS*19].

In this chapter we explore novel interaction techniques for chord diagrams. We introduce a novel interactive deformation that enables focus to be placed on a given segment, offering more detail to be seen as is shown in Figure 5.1 (right). This allows for an increase in the number of nodes that can be displayed, with the interaction allowing for individual node points to be identified. We introduce a novel click-based brushing technique for chord diagrams that enables multi-variate filtering of connections. We also introduce approaches for tracking multi-variate connections across multiple links.

The contributions of this chapter are:

- The introduction of a novel interactive layout technique to support detailed views of a particular segment

- Support for joining multiple edge connections in a chord diagram

- An interactive click-based filtering method for tracking connections across multiple chord segments

- The application of the techniques to a real-world telecommunication dataset [RRL*19]

The rest of this chapter is set out as follows: Section 5.2 outlines other work related to chord diagrams while Section 5.4 outlines a deformation technique. Multiple linked chord diagrams are presented in Section 5.4.3 and we present domain expert feedback in Section 5.5. Limitations are highlighted in Section 5.6 and a conclusion is drawn in Section 5.7.

5.2. Background

A primary resource to provide an overview of the field is supplied by McNabb and Laramee [ML17] which reviews data visualisation surveys. From this report, a survey of radial visualisation methods by Draper *et al.* [DLR09] is identified. From the survey presented in Chapter 2 a book by Lima is identified, which depicts many radial visualisation designs [Lim17]. Lima's book provides a history of radial visualisations and an aesthetically appealing collection, however, these images are static examples only and do not feature interaction.

For a historical look of research featuring chord diagrams we refer readers to the survey by Draper *et al.* Significant work since the publication of the survey are discussed here.

Edge Centred Optimisation Older research not included in the survey by Draper *et al.* is the work of Gansner and Koren [GK06]. They present methods that improve circular graph layouts (chord-diagrams) such as enhancing edge order for minimising maximum edge lengths, bundling of edges, and routing some edges externally to reduce clutter. Many methods to reduce edge crossings in a circular layout have been long established [BB04,ST99,Mäk88]. Other methods seek to increase crossing angles

for increased legibility [NEHH10]. Papp and Kunkli present a web-based circular layout for table visualisation [PK18]. Nodes can be sorted to minimise crossings and thus minimise clutter. Other interactions apart from sort are only aesthetic such as colour changes and arc sizes.

We utilise the external routing method introduced by Gansner and Koren, however we do not implement any edge optimisations. We focus on a deformation technique for a detailed view of individual nodes and multi-variate brushing for filtering.

Data Centred Optimisation   While the previous work cited in this sub-section utilises sorting to reduce clutter or edge crossing angles of the chords, the work of Cava *et al.* utilises sorting based on the data presented. Cava *et al.* present a radial design featuring a number of concentric rings with each ring representing an attribute and instances distributed radially [CFW17]. Categorical values are represented by a coloured circle with the colour indicating the category, while numerical values are represented by rectangles with the height of the rectangle mapped to the value. The centre of the circular design features a chord-diagram indicating connections between instances. Interactions supported include sorting, bundling, and highlighting of nodes.

We also implement a sorting method based on data attributes and a bundling technique, however, this is not the focus of our work. Our work features multi-edge connections, connecting to multiple nodes in a user-specified order, as well as a method for brushing these connections, and an interactive circle warping method for focus on a particular segment.

Edge Bundling   A popular paper discussed in the survey by Draper *et al.* [DLR09] is from Holten, who demonstrates an edge bundling technique for use with a hierarchical chord diagram through the use of b-splines [Hol06]. Users are able to adjust the level of bundling for a balance between clutter reduction and individual chord identification. Interaction for filtering is also demonstrated, chords are filtered with the mouse by drawing a line, with all chords not crossing this line being removed. This bundling technique presented by Holten is used in many other works including our own. A number of papers present edge bundling techniques that are demonstrated on chord-diagrams [TE10, EHP*11, HET12]. Lhuillier *et al.* provide a state of the art report of edge and trail bundling techniques [LHT17].

Crnovrsanin *et al.* [CMF*14] utilise Holten's edge bundling technique [Hol06] in their visualisation of social networks. They present a technique of chords representing intra-group connections being rendered on the outside of the traditional circular layout similar to the technique presented by Gansner and Koren [GK06].

Zeng *et al.* introduce a radial based design for visualising commuter interchanges across a city [ZFAQ13]. Unique features presented include statistics on flow volume at each node, the bundling of bidirectional bundles and placing one segment as a 'flyover ring'.

Although not strictly edge bundling, Nicholas *et al.* demonstrate a multi-chord diagram [NAL14], an extension of chord diagrams enabling for connections between more than a single pair of categories, e.g. three-way connections. Chord thickness to represent a weighted value are utilised by Etemad *et a.* [ECS14]. The chord diagram has thorn shaped chords inspired by Spirograph for visualising ecological networks.

Our work focuses on other novel interaction techniques for chord diagrams although we do implement the hierarchical edge bundling technique introduced by Holten [Hol06] to help reduce clutter.

Segment Centred   Chord diagrams are often used to visualise large amounts of data. As such, segments are often cluttered and require a technique to drill down into individual data points. Borkin *et al.* use a radial-based chord diagram for visualising file system provenance [BYB*13]. Segments of the circle represent a time-based hierarchy which can be interacted with by clicking to expand that segment and its hierarchy.

Wang *et a.*   visualise software calls with the use of a chord-diagram [WGS15]. Users are able to drill-down into the segments to reveal the sub segments and their connections, as well as filter out other segments.

Zhao *et al.* present a radial design for the display of long, time oriented data [ZCB11]. Their application features chord connections at the centre of the design to show connections between time series. The presented work features many interaction techniques although these are predominantly focused on the time series segments, including the zooming of these segments.

Our work also features a method for interacting with a chosen segment for more detail. Our implementation differs from these works as we implement a deformation technique to maximise the screen space to focus on a user chosen segment.

Works	Edge Optimisation	Edge Bundling	Data Centred	Segment Centred	Animation	Filtering	Hierarchical
Holten [Hol06]		✓				✓	✓
Gansner and Koren [GK06]	✓	✓					
Meyer *et al.* [MMP09]		✓		✓		✓	
Krzywinski *et al.* [KSB*09]						✓	
Bae and Lee [BL09]						✓	
Bostock *et al.* [BOH11]		✓	✓		✓	✓	✓
Zhao *et al.* [ZCB11]		✓	✓	✓	✓	✓	✓
Gou and Zhang [GZ11]		✓		✓		✓	✓
Kuhar and Podgorelec [KP12]		✓				✓	✓
Alsallakh *et al.* [AAMG12]						✓	
Borkin *et al.* [BYB*13]			✓	✓		✓	✓
Zeng *et al.* [ZFAQ13]		✓				✓	
Peixoto [Pei14]		✓					✓
Crnovrsanin *et al.* [CMF*14]		✓				✓	✓
Nicholas *et al.* [NAL14]		✓				✓	
Etemad *et a.* [ECS14]		✓				✓	
Wang *et a.* [WGS15]			✓	✓		✓	✓
Cava *et a.* [CFW17]		✓	✓		✓	✓	✓
Papp and Kunkli [PK18]	✓						
Ren *et al.* [RLB18a]							

Table 5.1.: A table summarising feature implementation in the related work. The edge optimisation column indicates work that feature techniques for improving chord diagram layouts by enhancing edge positions, while the edge bundling column shows work featuring clusters of edges. The data centred column mark work that sort edges according to the data being displayed. The column labelled segment centred indicates works that features a method of interacting with segments for displaying more detail. Works featuring animation, filtering, and hierarchical data are indicated in the final columns.

Hierarchical Data   Chord diagrams can be used to represent hierarchical data. A patent application by Burtner *et al.* demonstrates a hierarchical chord diagram including plurality of visual nodes [BBB*13]. Stasko and Zhang provide a flexible method for visualising hierarchical data in a radial format [SZ00]. They utilise a deformation method to zoom in on smaller hierarchies, creating a larger version of the focused section. This differs from our work as the hierarchies do not connect to one another in a chord diagram. Another technique for the same type of data is presented by Yang *et al.* [YWR02].

Gou and Zhang demonstrate a hierarchical chord diagram that offer views of different levels of the hierarchy [GZ11]. Kuhar and Podgorelec utilise a hierarchical chord diagram as part of an application for visualising ontologies featuring edge bundling and mouse-over interaction [KP12]. Peixoto demonstrates a nested generative model for describing the structure of large networks by utilising chord-diagrams with hierar-

chical edge bundling [Pei14]. Our work features a deformation method for zooming on a particular hierarchy, different from these works.

Genomics Visualisation   Krzywinski *et al.* present a tool called Circos for the visualisation of genomics data [KSB*09]. The tool offers clickable figures that can be selected for the display of features or annotations. Meyer *et al.* utilise a chord-diagram for genome visualisation and the connections between chromosomes [MMP09]. Users are able to select an individual chromosome from an overview to see the connections from that specific chromosome.

Our implementation features novel interaction techniques not realised by these works.

Applications   Chord diagrams are used to represent a number of different types of data-sets across many domains such as information technology [SKE15, CDJM14, CZIM18]. Alsallakh *et al.* present Contingency Wheel for the visualisation of contingency tables [AGMS11]. Nodes are represented in multiple category segments arranged radially, with their radial placement within the segments representing the association strength. Chord links between segments indicate the frequency of connections between categories. Alsallakh *et al.* [AAMG12] further expand on the Contingency Wheel to visualise larger datasets and to address limitations of scalability and understanding. Alsallakh *et al.* further build on the Contingency Wheel, using different aggregation and additional visual elements for visualising overlapping sets and their connections [AAMH13]. Bae and Lee use a chord-diagram to visualise tag relations with interactions including selection and focus+context [BL09]. Our implementation utilises call connections through departments of a call centre to demonstrate novel interaction techniques.

Visualisation Tools   The JavaScript D3 visualisation library by Bostock *et al.* [BOH11] utilises the Circos [KSB*09] style for their chord layout. D3 enables users to create their own interaction techniques however these options require programming knowledge. A particular D3 example features hierarchical edge bundling as proposed by Holten [Hol06]. Interaction techniques supported withing the D3 environment include zooming and panning, and mouse-over interaction.

Charticulator is a tool presented by Ren *et al.* for creating bespoke visualisations including chord diagrams [RLB18a]. The tool enables the creation of visualisations without the need for any programming knowledge. However, the visualisations created do not feature interaction mechanisms.

These tools feature means of creating chord diagrams whereas our implementation is designed to showcase some novel interaction techniques.

Radial Layouts   Diehl *et al.* compare the use of radial based visualisations to Cartesian layouts, concluding that Cartesian layouts outperform their radial equivalents [DBB10]. However, they did not analyse layouts for the use of displaying connections between nodes. Burch *et al.* preform an eye tracking study to compare tree diagram layouts, including radial based designs [BKH*11]. They conclude that a traditional tree layout significantly outperforms other designs. However, the radial hierarchical design used did not have a fixed structural layout. A chapter titled 'On the benefits and drawbacks of radial diagrams' in The Handbook of Human Centric Visualization by Burch and Weiskopf further discuses the benefits and drawbacks of radial diagrams [BW14]. Chord diagrams are not discussed in their chapter however, although they do note that curved links are more difficult to follow, that rendering performance of radial diagrams is decreased due to increased computation of polar coordinates, and that radial diagrams can generally be more aesthetic. These works study the use of circular designs and their perceptibility where as our work starts with a traditional circular layout chord diagram, however we deform this to maximise screen space use for focus on a particular segment.

5.3. Data Description and User Requirements

The data used to demonstrate the techniques in this chapter is a subset of the dataset described in Chapter 1.1.2, focusing on call transfers between departments within the call centre. The dataset has a hierarchical structure with each department belonging to a particular business segment and having a number of its own agents. In total the data set consists of 10 business segments, 92 departments, and 3,174 call centre agents. The dataset comprises of one day of calls totalling 225,256 calls, of which the majority feature no agents or are dealt with by a single agent and therefore do not

feature a transfer connection. The remaining 17,914 calls feature 80,361 transfers, with a maximum of 17 transfers on one call.

As well as belonging to departments, agents also belong to specific teams within the departments. The agent's physical site location is also archived along with a categorical skill level. Many call metrics are recorded for each call such as the connection duration with each agent, and a measure of the customer effort made when talking to the agent known as CES (Customer Effort Score). A time stamp is also associated with each call.

User Requirements   The techniques introduced in this chapter are a response to analyse call transfers between different agents in a range of call centre departments. Due to the large number of call centre agents, overplotting becomes a challenge. The methods presented attempt to address the following requirements:

- The ability to visualize transfers at the business, departmental and individual agent scale.

- Study multiple transfers within individual calls

- View call details

The visual design should inform analysis of the dataset and provide insight into agent and customer behaviour.

While techniques such as edge bundling and node position optimization can reduce clutter and line crossings, they do not prevent overplotting of nodes on the circular axis. Consequentially identifying individual nodes amid a large number becomes difficult. Previous work provides drill down methods for exploring segments of a chord diagram (see Section 5.2), however, it does not allow for easy tracking of connections from one individual node to another. The ability to track connections between multiple nodes is generally absent from previous works.

5.4. Interaction Techniques for Chord Diagrams

We introduce an interactive method for expanding a user-chosen segment of a chord diagram, in this case, this represents a particular call centre department. With over 3,000 agents represented, identifying individual agents within the available screen space

can be very difficult. To remedy this problem we develop a deformation technique to expand the size of individual departments. The user is able to pick an individual segment of the radial visualisation to expand it to maximise its size in screen-space to see more detail. An example of this can be seen in Figure 5.1. On the left image, a call centre department within the Tech Support has been chosen and highlighted with colour on the axis segment that represents the chosen department. The colour of the axis segment indicates the business segment to which the department belongs and can be referenced in the legend.

5.4.1. Dynamic Rotate + Expand Layout for Segment Focus

The user is able to select any department by clicking on its representation on the chord diagram segment. On the selection of a department, the segment representing the department will be coloured, and the whole diagram smoothly rotated to position the department at the lowest point at the six o'clock position. This is demonstrated in the first two scenes of Figure 5.3. The use of animation is incorporated to convey the rotation with visual continuity. Once a department has been chosen and highlighted, a second click on that department causes it to expand to form a long edge segment of agents, with other departments rotated to form an arc over this straightened segment as can be seen in Figure 5.1. Animation is again used for the expansion to convey visual continuity, this can be seen in the accompanying video [Ree20a]. Figure 5.3 illustrates the process of a chosen segment rotating to the six o'clock position, then expanding to fill the screen space.

Once expanded, the department is segregated into individual agents, enabling an interactive focus to be placed on individuals rather than whole departments as was only possible previously. As other departments are no longer the focus once the deformation has been made, connections within the same non-focus business segments are not rendered in order to reduce clutter.

The bundling technique first introduced by Holten [Hol06] is demonstrated in both the traditional plot and the deformed plot introduced in this chapter, as can be seen in Figure 5.1.

To support exploration of the focus segment, we implement an on-mouse-over feature that highlights the calls from a particular department, as can be seen in Figure 5.2. By hovering the mouse over the area of a particular department, the calls
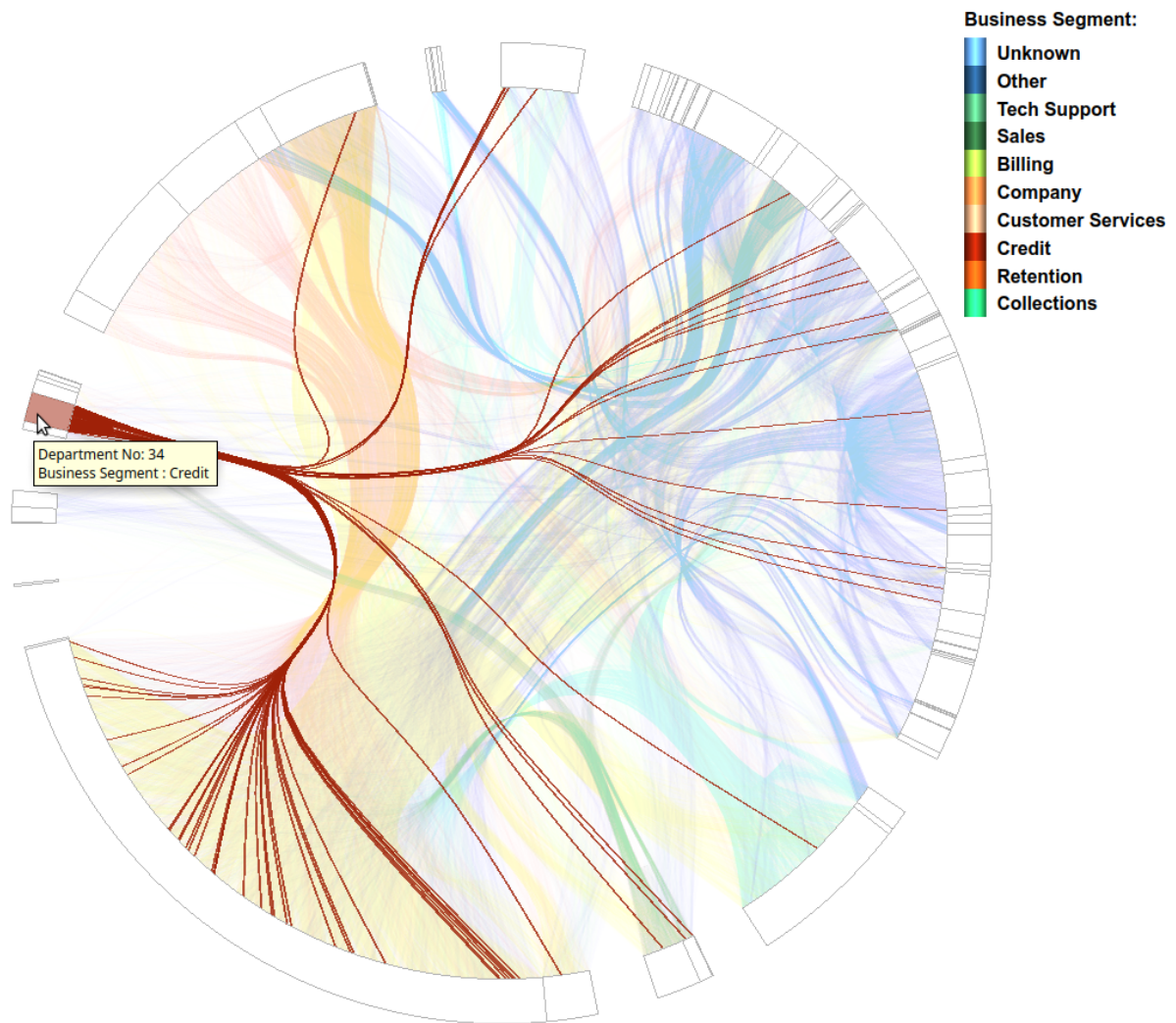
Figure 5.2.: A department belonging to the Credit business segment has been highlighted
using the mouse-over feature, showing all connections from that department.

being transferred from that department are emphasised, while other transfers are
de-emphasised. Emphasis is achieved by decreasing the opacity of the transfers origi-
nating from the chosen department and increasing the opacity of the other transfers.
To further emphasise the highlighted connections, the rendering order is taken into
account with highlighted connections rendered last in order to appear on top of the
other connections. An on-mouse-over information box also appears, providing details
of the department number and to which business segment it belongs.

Once a department has been expanded using the rotate + expand layout outlined
in Section 5.4, the on-mouse-over feature no longer highlights the department but
highlights individual agents. This emphasises all connections originating from a
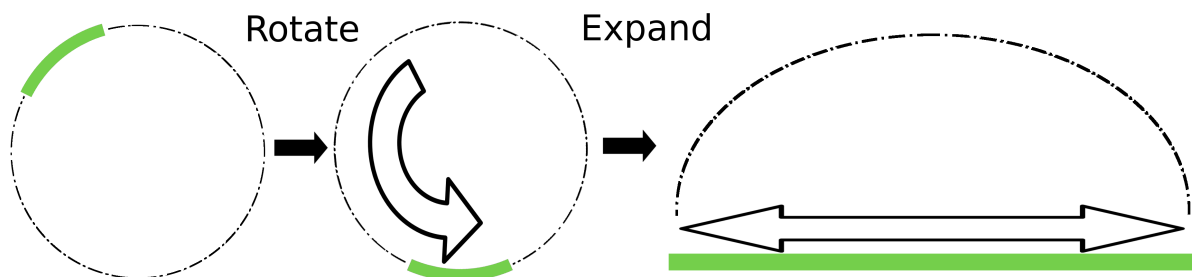
Figure 5.3.: A demonstration of the interaction for expanding a segments. A user chosen segment rotates to the bottom of the diagram to show that it has been selected, then expanding to fill the horizontal screen space.

user-chosen agent, as well as all the connections received by the agent. Agent highlighting is achieved by the same opacity modifications as performed on departmental highlighting. For an individual agent, the on-mouse-over information box provides data about the agent including the agent ID, site location, team and skill levels, the number of transfers in and out, and average call metrics (Figure 5.4 (top)). By clicking an individual agent, additional details of all calls handled by the chosen agent will be returned in a dialog box (bottom). These interactions are inspired by Shneiderman's Visual Information-Seeking Mantra of overview first, zoom and filter, then details-on-demand [Shn03].

5.4.2. Click-Based, Multi-variate Filtering

We introduce an interactive method for filtering chord diagrams based on multiple connections. Users are able to interactively create a filter by clicking on segments representing departments, mimicking a call with connections. These departments are added to a list as the user creates the filter. As departments are being interactively selected, a dynamic edge is rendered between the departments in the list indicating a series of connections for filtering. Calls are then filtered to only those that feature agents in the compiled list. This can be seen in Figure 5.5. The dynamic nature of this click-based interaction is demonstrated in the accompanying video [Ree20a].

Different modes of filtering also apply to the selected list of departments. The first is calls that feature the listed departments, secondly calls that feature the listed departments in the same order specified by the filter, and thirdly only calls that feature direct links between the listed departments in listed order. Additional user options are

Business Segment:
Unknown
Other
Tech Support
Sales
Billing
Company
Customer Services
Credit
Retention
Collections

Transfers In: 9
Transfers Out: 6
Average CES: 8
Averge Call Time: 130
Agent ID: 1014167
Team ID: 1199
Agent Skill: 0
Agent Site: 8
Department No: 34
Business Segment : Credit

Agent 1014167 Call Details

Call Details

	Call ID	Number of Agents	Agent 1	Department 1	Agent 2	Department 2	Agent 3	Department 3	Agent 4	Department 4
1	9220033191348	5	1306826	92	1302581	55	1104118	34	1014167	34
2	9144033200329	1	1014167	34						
3	5021033185320	1	1014167	34						
4	4164033171955	3	1405077	97	1405009	92	1014167	34		
5	4115033180018	1	1014167	34						
6	4063033183634	2	1014167	34	1014007	34				
7	3182033190036	1	1014167	34						
8	3153033200229	1	1014167	34						
9	3096033195041	1	1014167	34						
10	25129033163739	2	1014167	34	1014768	34				

Figure 5.4.: (Top) The on-mouse-over feature for an individual agent. Call connections made to and from the agent are rendered over other connections. The information box provides agent details and metrics. Observable is that no calls are transferred by the agent to other departments, only received from other departments. A zoomed section in the bottom right of the figure shows the mouse-over detail. (Bottom) Agent detail dialog obtained by clicking an individual agent. The dialog lists all calls handled by the user-chosen agent, with details of other departments and agents involved in each call.

to set the first listed department as the originating department of the call, and another option to make the last listed department to be the terminating department.

For user convenience, options are available on a mouse right click. When a user is creating a filter, a right-click shows a menu with options to delete the last drawn
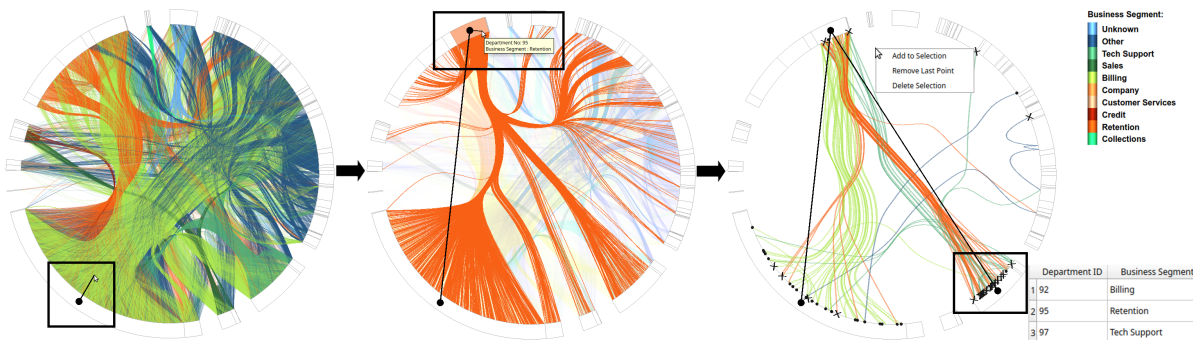
Figure 5.5.: A representation of the interactive click-based filtering of the the chord diagram. In the leftmost image, the user has clicked on department 92 in the Billing segment to begin the click-based filter. A point indicates this selection and a dynamic edge is rendered to the mouse cursor to show that the filter is being constructed. The middle image shows the next selection of department 97 in the Retention segment. Here the mouse-over feature can also be seen showing the connections from that department, providing information informing the construction of the filter. The final image shows a final selected department (97 in the Tech Support segment) with the filter applied. A list pictured in the bottom right provides details of the chosen segments. The filter has been selected to only show calls with these direct connections. The majority of the remaining calls start at department 92 and follow the departments in the filter and terminate at department 97 as indicated by the starting points and ending crosses. However, a minority of the calls start or end at other departments.

point, clear the brush, or to terminate the current selection. This can be seen in the right image of Figure 5.5.

This option allows the user to create the connections of a theoretical call to find calls that match the drawn profile.

5.4.3. Representing Direction and Order

Traditionally chord diagrams are used to show connections between two nodes, however in the application of chord diagrams for call connections, one call may have multivariate or multiple edge connections. To accommodate this, we present methods for tracking a call across multiple connections, in our case, to multiple agents.

To achieve this we first need to establish a direction to the connections. For this we can use arrows as can be seen in Figure 5.6. An alternative method for this is to use colour to code the direction, with start of a connection coloured to an end of a colourmap and the end of the connection coloured to to the opposite end of
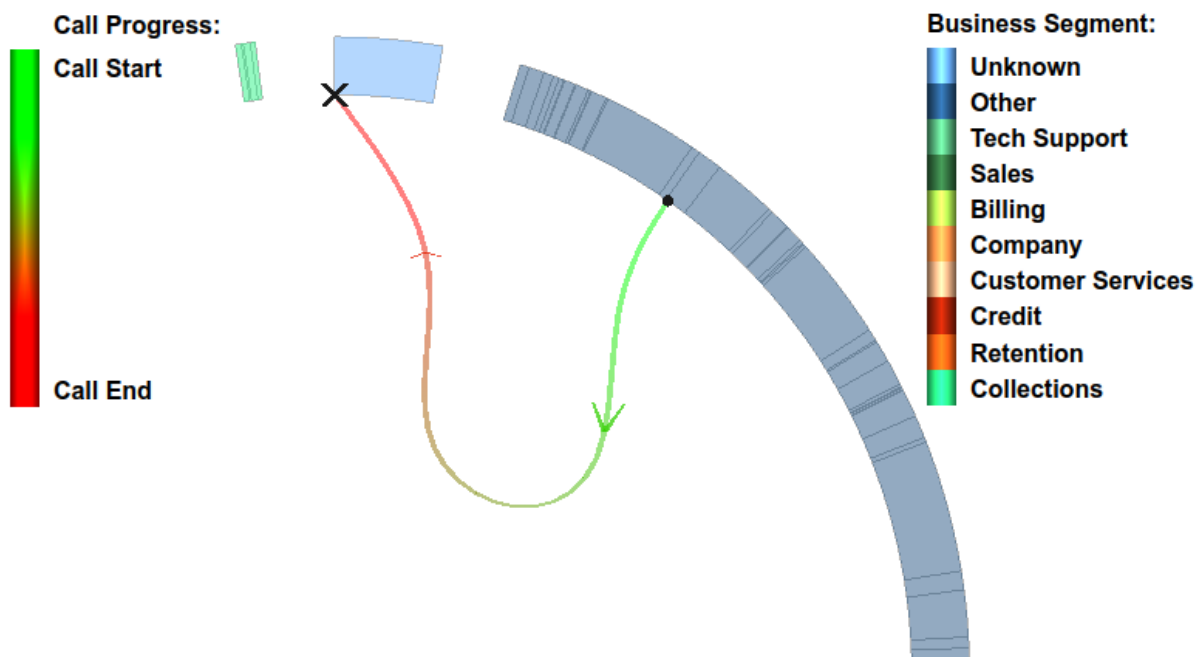
Figure 5.6.: A zoomed chord diagram showing a single connection between two departments. The direction of the connection is indicated by arrows and a green-to-red colourmap, as indicated in the continuous legend. Business segment is indicated by the colour of the axis. A point is also used to indicate the start of a call with a cross to indicate the end.

the colourmap. A continuous colourmap is best for this method. Indicators can also be used on the segments to indicate the start and end of a connection, however, overplotting is a problem with scaling. These techniques are not mutually exclusive and can be combined as in Figure 5.6.

Once one direction has been established, the colour can be used to show call progress across multiple connections. Figure 5.7 shows an example of a single call with 12 transfers. A continuous colourmap with multiple hues has been used to make it easier to distinguish between connections and to determine chronological agent order. Arrows indicate direction of individual connections.

Animated Connection Order and Direction   Another method to show direction and the order of the connections is by using animation. We demonstrate the use of a short black line that travels along the connections in call order. This is demonstrated in the supplementary video [Ree20a].
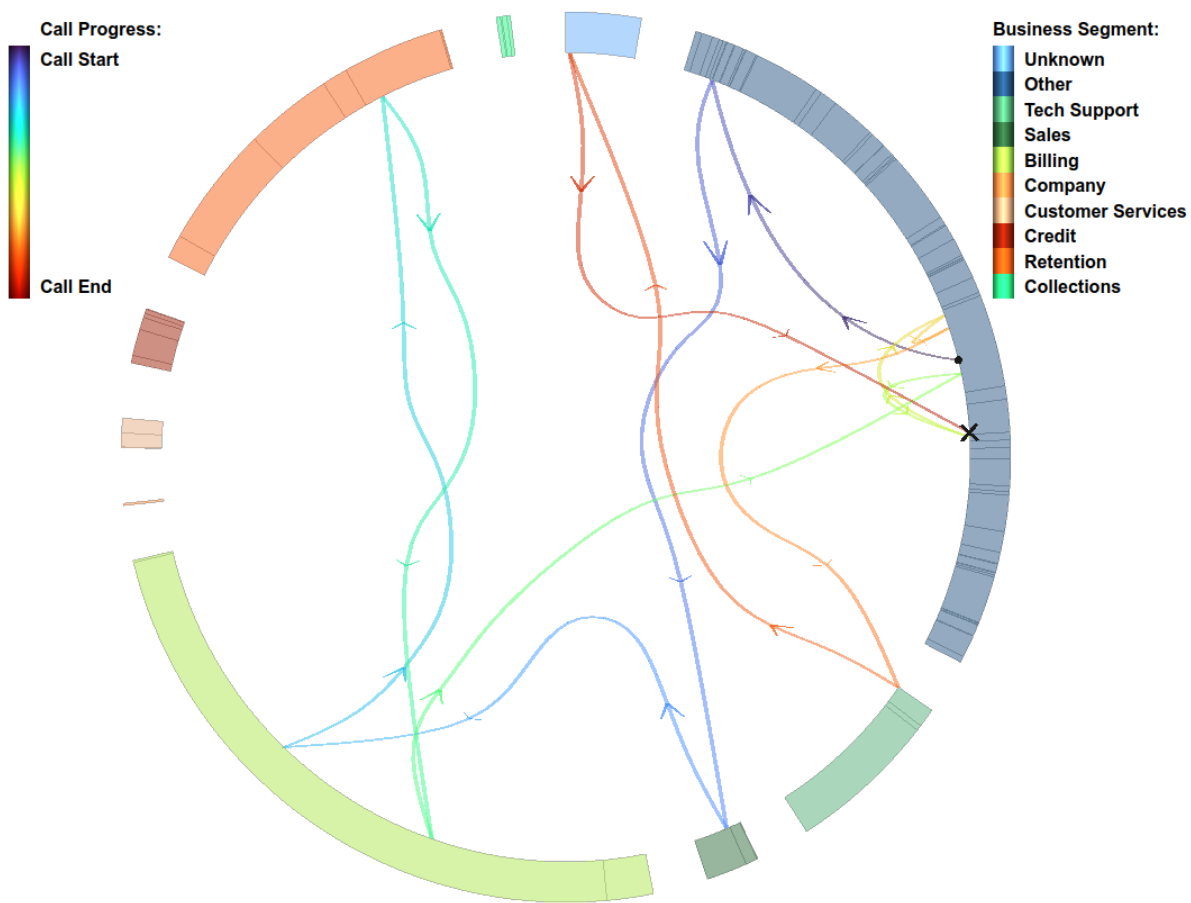
Figure 5.7.: A call featuring 12 transfers between multiple departments. The edge direction representing each connection is indicated by arrows and a modified rainbow colourmap [Mik19], as indicated in the legend. Business segment is indicated by the colour of the segment. A point is also used to indicate the start of a call with a cross to indicate the end.

To give the user more control over the speed of the motion, we provide an interactive version of this using a time-window slider. When this is active, the two inputs of the slider control the extent of how much of a call is rendered black. Having the lower slider input at the very lowest range and the high slider at the highest end of the range, the call edges are entirely black. By moving one of the sliders, the progress of the call can easily and intuitively be followed. This feature is inspired by the work of Laramee [Lra02] and is demonstrated in Figure 5.8 where the slider, seen at the top of the Figure, has been set to highlight the first seven transfers in black.
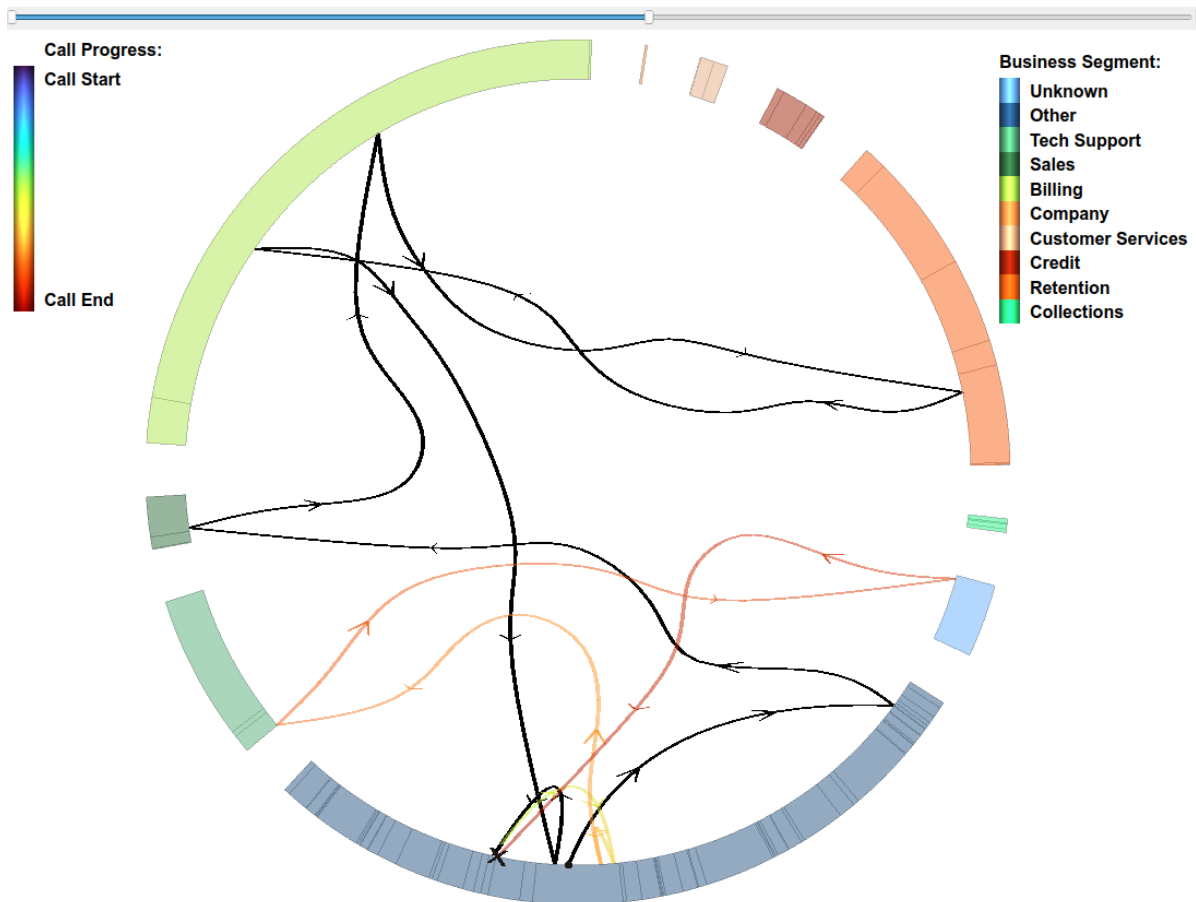
Figure 5.8.: The connections of a call with a black sliding edges interactively set by the user using the pictured range slider. The focus call features 12 transfers between multiple departments. The direction of the connection is indicated by arrows and a modified rainbow colourmap [Mik19], as indicated in the continuous legend. Business department is indicated by the colour of each segment.

5.4.4. Other User Interaction Options

Meta-data Filtering

These filters are based on the meta-data of the call, for example a filter for the number of transfers in a call, or the time of day of the call. These filters can be adjusted by the user based on minimal and maximal values for each attribute, adjustable with user controls such as range sliders. Once the minimum and maximum values have been set, calls are iterated through to find the calls that fit into the indicated criteria.

Agent Sorting

Agent position about the circumference of the plot is determined by the business segment and department of the agent. Within the departments, a user option enables the sorting of the agents by data attributes increasing clockwise. These data attributes include the average call time, average agent effort score, and the number of transfers made. As well as this, agents have a categorical skill level, and a team designation which are also used for sorting. Agents are not necessarily exclusive to one department, therefore an agent will be repeated to be represented in each department that they belong to. An agent may also belong to more than one team or skill level, therefore they are also repeated for each skill or team if those categories are used as sort criteria.

Intra-segment Connections

To aid clutter reduction, we implement a user option that renders connections from the same business segment on the outside of the diagram. This exterior rendering is an imitation of that provided by Gansner and Koren [GK06]. An example of this can be seen in Figure 5.9 (top).

This feature is disabled for non-focus departments when the deformation is used to expand a particular department; this reduces clutter. For the focus, deformed department, internal connections are routed underneath the department as in Figure 5.9 (bottom).

5.4.5. Flight Data

We also demonstrate the techniques introduced in this chapter on an alternative dataset. Figure 5.10 shows completed flights recorded in a single 24 hour period. The edge connections represent flights between airport nodes grouped by their country and continent. The segment representing Germany has been expanded allowing individual airports to be seen. Observable with the expansion is that only one German airport in the dataset (Frankfurt) features flights to other continents, with other airports only featuring flights to other European countries or domestically.

Aircraft that feature more than one flight within the 24 hour time period have their flights connected, which can be explored by utilising the techniques introduced for
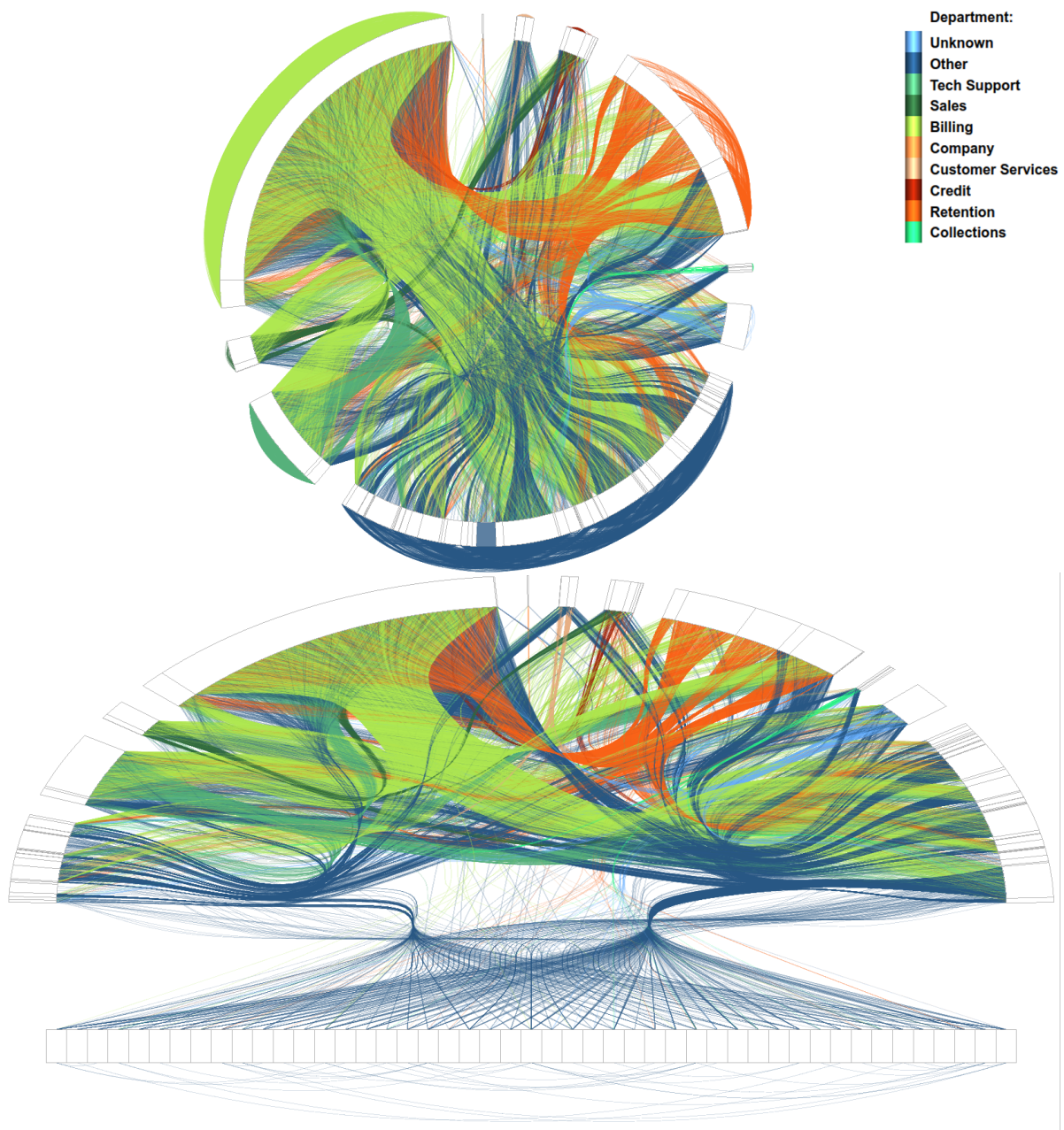
Figure 5.9.: (Top) Connections between agents in the same business segment are routed exter-
nally to reduce clutter. (Bottom) Connections between agents in the same business
departments are not rendered when using the the deformation, however calls
within the focus department are rendered underneath.

direction and order. Different to the call-centre dataset, flights are shown to leave and
return to the same airport whereas an agent is unable to transfer a call to themselves.
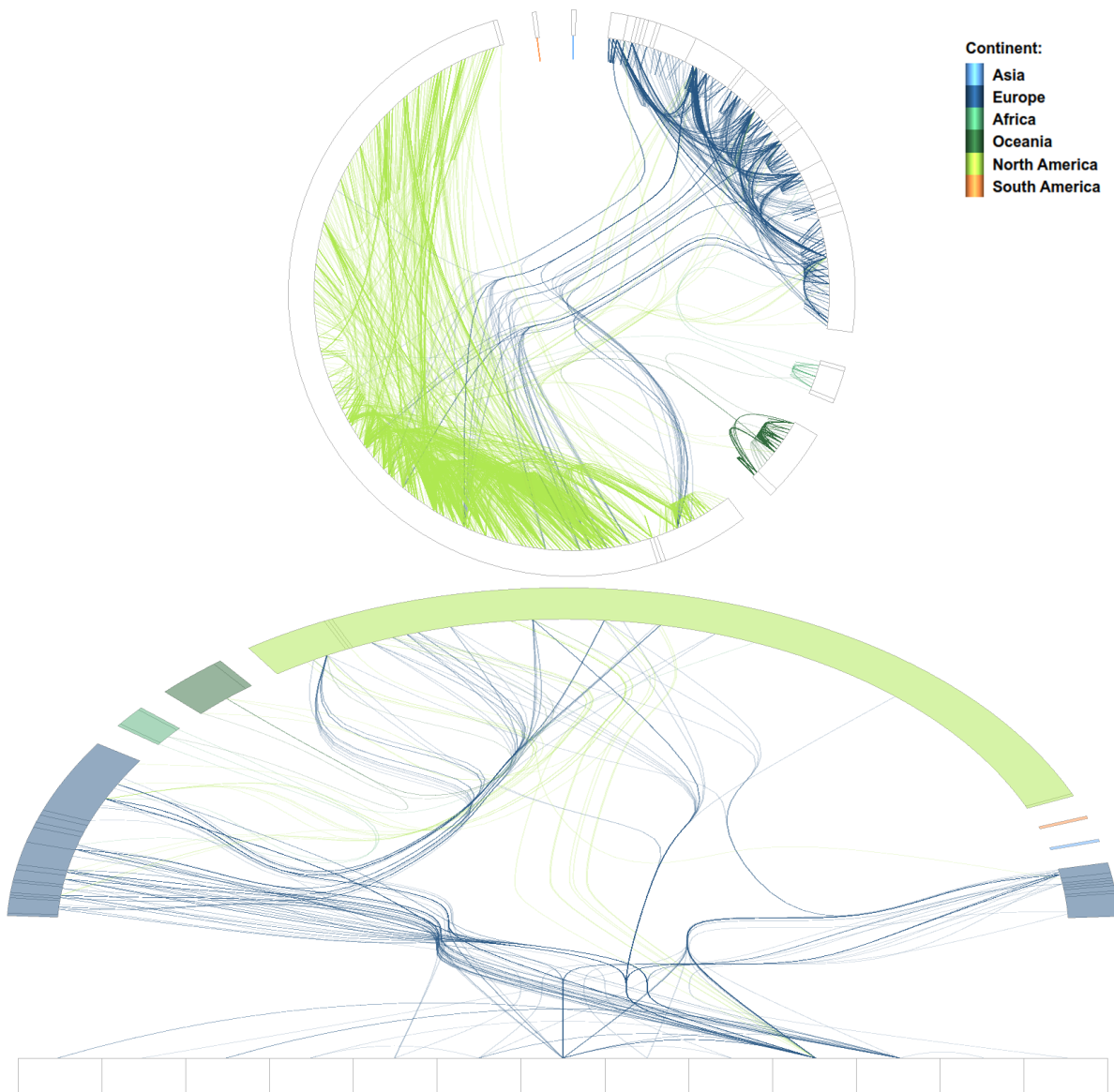
Figure 5.10.: Aircraft flights between airports in different countries and continents. The segment representing Germany has been expanded in the bottom image to focus on individual airports.

5.5. Domain Expert Feedback

Because this application is customised for specialists in the call-centre industry, we sought a domain expert feedback approach for evaluation. Feedback was garnered from two domain experts from the call centre industry, familiar with the call-centre dataset used in this chapter. The two experts have over 50 years of experience between them in the call centre industry. The experts were shown a demonstration of our

implementation of the techniques introduced in the chapter before being asked to provide feedback. We report this feedback here.

A deployment of an application with the techniques described in this chapter, and a more through evaluation, was not possible due to time constraints of the domain experts. For this evaluation, only two domain experts were available for feedback of the techniques presented due to the departure of a collaborating partner from QPC Limited.

When asked how the demonstrated techniques might be useful, Expert 1 replied: *"The entire thing looks really easy to use, that's a big component of making it useful to start off with. The variety of different selection / filter options, and the ability to focus on areas of interest so quickly and dynamically are things that are particularly useful I think."* Meanwhile, Expert 2 was most impressed with the clustering technique used: *"The hierarchical clustering technique and the adjustment of its strength does help reduce clutter."* Expert 1 was asked how the techniques compare with how they were currently explore the data; he was impressed with the ability to focus on an individual agent, and then being able to quickly switch to focus on another. *"What is does do though is make it much easier and quicker to navigate through and to switch perspectives and focus in specific areas."*

Asked if the presented tool could help explore and present the data, Expert 1's answer was: *"Definitely. To me the ease of use and it's general visual appeal / slickness is where the value is."* Expert 2 was also positive about the visual appeal of the implementation: *"The non-standard way of visualising the data and the various animations would help make the visualization engaging for presentations."* Asked if there were any improvements that could be made, Expert 1 replied: *"To have the ability to compare different filtered groups, by having the entire main chord diagram duplicated, once for each group. That way you would use the tool to compare the experiences of say two different customer groups, side-by-side."*

5.6. Limitations

The techniques presented in this article aim to increase the number of nodes that can be used with a chord diagram through the use of interaction and deformation. To evaluate the effectiveness of the presented technique, a comparison of the visual space for each node is made. A number of factors influence the number of nodes that can be displayed including the physical size of the chord diagram, the distance which

the diagram is being read at, and the medium of display. Since these variables are indeterminable and that the deformation is performed on the same canvas as the chord diagram, a common minimal perceptual size can be assumed s_{min}. Given the diameter of the chord diagram d, then the number of visible nodes n_{circ} is given by Equation 5.1.

$$n_{circ} = \pi * d / s_{min} \tag{5.1}$$

The deformation expands a single segment to the width of the available canvas, this can be assumed to be the same size as the diameter of the chord diagram d, although in reality screens are rarely square and have a larger side that can be used to maximize segment expansion. Desktop monitors vary in aspect ratio from 4:3 up to 21:9 therefore the expansion will also increase by this factor. The number of visible nodes for an expanded segment n_{epnd} is given by Equation 5.2.

$$n_{epnd} = d / s_{min} \tag{5.2}$$

By substituting n_{epnd} for d/s_{min} in Equation 5.1, it becomes clear that the number of visible nodes in a chord diagram would be π times greater than an expanded segment. Therefore if more than three segments are available, the technique presented is beneficial.

In a dataset absent of categories to segment nodes, artificial segmentation may be used to enable the expansion deformation technique.

A limitation exists with the maximum number of nodes that can ultimately be visualized. With the assumption that each visible node is an expandable segment the maximum number of nodes achievable with the deform expansion technique n_{circ+} is given by:

$$n_{circ+} = n_{circ} * n_{circ} / \pi \tag{5.3}$$

This shows that the potential increase in the number of nodes that can be displayed is proportional to the square of the number for an ordinary chord diagram.

The expansion demonstrated in Figure 4.1, generated from a real world dataset, provides a 98 times expansion in screen-space, revealing 27 agents which were pre-

viously indistinguishable and cluttered. Figures 5.8 and 5.10 show a 55 times and 23 times expansion in screen-space revealing 47 and 13 nodes respectively.

To further increase the number of possible nodes is future work for the project. Possible solutions to this are the use of lenses [TGK*17] or some further distortion.

Because the techniques presented in this article require interaction, this limits the techniques to an interactive medium, although expanded segments can be visualized statically and are, in our experience, intuitive. Animation techniques are again limited to an interactive medium.

Representing connection order in the case of connections through multiple nodes becomes less effective when overplotted, therefore it can be fairly ineffective without methods to filter connections. In the case we present, we apply meta-data filters to reduce clutter from overplotting, enabling for connections to be clearly seen. Addressing this issue is future work. A potential problem exists with the use of color for tracking calls with multiple connections. Caution must be exercised as color-blind people may not be able to follow as intended. Other avenues for future work are to configure a linked multi-chord diagram for comparisons as recommended by the domain expert. A user study to determine how users comprehend the methods presented in this paper is another future work opportunity.

5.7. Conclusions

We present a novel deformation technique for interactive exploration of connection data in a chord diagram. This method enables a grater number of nodes to be represented in a chord diagram with interaction that facilitates individual nodes to be explored. We demonstrate this technique with a real-world call-centre dataset, visualising call connections to different agents in different departments. This technique maximises the available screen-space to draw focus to a user chosen department. Additionally we introduce a method for tracking calls across multiple connections with multiple call-centre agents. An interactive click-based filtering technique is introduced to filter calls based on the connections through different departments.

We report domain expert feedback on our implementation of the novel techniques presented in this chapter, and discuss limitations of the presented techniques. The

deformation technique is shown to provide an increase in the number of nodes that can be displayed by a factor proportional to the square of a standard chord diagram.

Chapter 6.

Conclusion

"Progress lies not in enhancing what is, but in advancing toward what will be."

— Khalil Gibran – Poet, Visual Artist (1883–1931)

6.1. Outcomes

This thesis explores methods for visualising a real-world call centre data-set. Challenges faced with this task include aspects of scale, with almost five million calls to process. Each of these calls has many attributes therefore this challenge is extended with multidimensional data. Solutions to issues of scale have been developed and implemented allowing for visual analytics to be performed on the provided dataset, leading to new insights. A discussion of how each chapter contributes to the challenges is made here, followed by a more general conclusion.

Chapter 2: A Survey of Information Visualisation Books   A survey of information books is presented, providing insights into which topics are discussed across an ever growing literature explosion. The classification allows for the reader to easily identify topics of interest across the surveyed library. Recommendations are provided for choosing books for each category of book. Meta-data from each book is collated and presented to help the reader compare and contrast the literature.

By surveying the literature space, a vast pool of knowledge was obtained over a number of relevant topics from perception to business visualisation. The knowledge gained provided a foundation for the visualisations developed over the following chapters.

Chapter 3: Feature-Rich, GPU-Assisted Scatterplots for Millions of Call Events
Chapter 3 present an application that visualises large call centre data sets using scatterplots that support millions of points. The application features a scatterplot matrix to provide an overview of the call centre data attributes, animation of call start and end times, and utilises both the CPU and GPU acceleration for processing and filtering. A demonstration of the use of the Open Computing Language (OpenCL) to exploit a commodity graphics card for the fast filtering of fields with multiple attributes is also provided. The application is exhibited with millions of call events from a month's worth of real-world data, the first time a month's worth of data has been visualised. Insights into the data is provided and domain expert feedback is reported from our industry partner.

Chapter 4: AgentVis: Visual Analysis of Agent Behaviour with Hierarchical Glyphs
In Chapter 4, a design study is presented to develop means of visualising call-centre agent behaviour. In the resulting application, multivariate glyphs are used in a scatter-plot layout to visualise a month's worth of call centre data, representing 6,500 agents handling nearly 5 million calls.

To address clutter, glyphs are clustered using a customised clustering algorithm based on Euclidean distance, and extensions of this to cluster agents by their department, site, skill, or team. The resulting hierarchical glyph represents the mean value of all child agent glyphs, removing overlap and reducing visual clutter. Glyph clusters are dynamically controlled according to zoom level and customised distance metrics, to utilise image space with reduced overplotting and cluttering.

This represents a novel solution to the problem of overplotting for glyph based layouts and guidelines for the hierarchicalization of glyphs are proposed. We report domain expert feedback on the application created to visualise the data, gathered from several interview sessions.

The technique used to develop the application presented in Chapter 4 is presented in Appendix A. This technique utilities GPU parallelisation to calculate the individual

geometries of each glyph instance. Source code for a simple version of the technique presented are provided to enable ease of recreation.

Chapter 5: Interaction Techniques for Chord Diagrams A novel deformation technique for interactive exploration of connection data in a chord diagram is presented in Chapter 5. This technique is demonstrated with a real-world call-centre dataset, visualising call connections to different agents in different departments. This technique exploits the available screen-space to draw focus to a user chosen department. Additionally, a method for tracking calls across multiple connections with multiple call-centre agents is presented. An interactive, sketch-based filtering technique is introduced to filter calls based on the connections through different departments. Domain expert feedback on our implementation of the novel techniques presented in this chapter is reported, and limitations of the presented techniques are discussed.

Collaboration This thesis and the collaboration with QPC Limited yielded three customised visual analysis tools for the exploration of the provided call-centre dataset. These tools were provided to QPC Limited for use and development, however, due to other priorities and not enough resource, these have not been deployed. This can be considered a drawback of the project, however, overall the collaboration can be considered a success, with knowledge gained through this process transferred to QPC Limited.

6.2. Future Work

Each project presented in this thesis has expansion potential that is outlined below. Following this, more general future work is provided.

A Survey of Information Visualisation Books Books continue to be published rapidly therefore a natural extension of this work would be to include newly published books as they become available [MP18, Ara18, Gab18, Ren18, Hea18, Ben19]. This survey predominantly covers information visualisation books however many other data visualisation books are available in different areas of the field such as scientific visualisation, infographics, and GIS. In future we would like to extend our survey

to cover these other sub-topics of data visualisation. One area that is experiencing very rapid growth currently is the digital humanities. A survey of literature including books would be very valuable due to some overlap with text analytics and text visualisation. An example of such a book is Introduction to Text Visualization by Cao and Cui [CC16].

We had difficulty in procuring some books that would otherwise be included in this survey, such as Tufte's Beautiful Evidence [Tuf06]. In future we would like to add full reviews of these books into the survey as they are highly cited literature. While conducting the search for suitable books, it was noticeable that there are a number of books dedicated to specific data visualisation tools, for example we found 23 books on D3 [BOH11], a JavaScript data visualisation library, over a dozen books on the Processing language, and nearly 4,000 books on Excel. This indicates that the majority of information visualisation books are related to visualisation tools, and therefore each tool may be suitable for its own book survey. It also indicates that these areas are saturated and that as an author they are areas to best avoid.

Feature-Rich, GPU-Assisted Scatterplots for Millions of Call Events   In future, we would like to further explore improvements with the use of general-purpose compute on the GPU. This includes the use of a shared context between OpenCL and OpenGL memory buffers as demonstrated by Alharabi *et al.* [ACL17] and the use of the Vulkan API [The16]. Following feedback from QPC Ltd., we would also like to extend the software to handle more call variables and to utilise dimension reduction techniques to highlight key caller data dimensions and to find co-relation coefficients.

Further testing of the software would also be beneficial with data sets from other vocations, and larger data sets. The ability to display other visual designs is also a desirable feature. Over-plotting is a significant issue when plotting a large number of data points, in future we would like to provide an auto-detection feature for over-plotting that adjusts the opacity accordingly. Figure 6.1 shows a heat-map density plot of CES against the time of day for a month of data, highlighting the areas of over-plotting.

AgentVis: Visual Analysis of Agent Behaviour with Hierarchical Glyphs   Some limitations of using the dynamically created glyphs with memory have been noted. These are that glyphs may be different when zooming in compared to when zooming
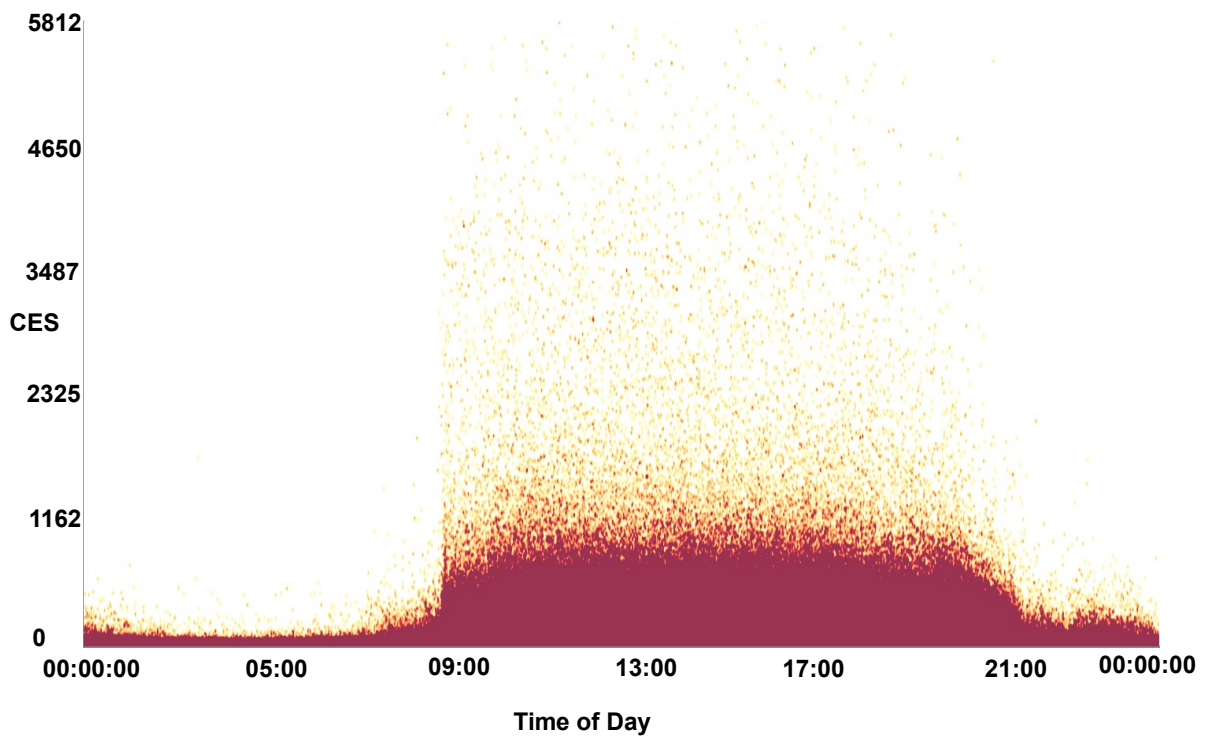
Figure 6.1.: A heat-map density plot of CES versus the time of day for a month of call data. The darker maroon colours indicate areas of over-plotting.

out, potentially leading to confusion. Modifying the algorithm to provide consistent glyphs at different zoom levels is future work.

Developing a workable hierarchical tree solution that allows for zooming on independent axes is another avenue for future work. Further incorporating new variables from richer call datasets is another desirable outcome.

A question also remains in relation to how the hierarchical glyphs are interpreted, especially as the user interacts with the view by zooming, although feedback from our domain experts suggest that it is intuitive. The use of animation of the children glyphs clustering into the parent glyphs is implemented to aid perception, however its effectiveness has not been fully explored. To further explore this, we propose a user study to fully understand how the clustering is interpreted.

A more thorough evaluation in the form of a long term field study, once software has been deployed and used in workflow, would provide a more complete study of the effectiveness of the design.

Interaction Techniques for Chord Diagrams   The techniques presented in this paper have some limitations that represent future work to address. The first limitation is with the segment expansion layout presented. The aim of this design is to improve the scalability of chord diagrams, to enable more nodes to be used and still be individually identifiable. The technique presented does achieve this, however, the solution is not scalable with an increasing number of nodes as the expanded segment can be inundated. To address this problem is future work for the project. Possible solutions to this are the use of lenses or some further distortion.

Another potential problem is with the use of colour for tracking calls with multiple connections. Caution must be exercised as colour-blind users may not be able to follow as intended.

Other avenues for future work are to configure a linked multi-chord diagram for comparisons as recommended by the domain expert. A user study to determine how users comprehend the methods presented in this paper is another future work opportunity.

Other Future Work   Within this thesis, applications have been developed to visualise call-centre aspects including customer calls, Chapter 3, agent behaviour, Chapter 4, and agent transfers in Chapter 5. Other visualisations have previously been developed for visualising customer calls, [RTL*16, RLS*19] and for customer journey analysis [RRL*18b], however there are other avenues for visualisation of the dataset. However, no visualisations are available for agent-customer interaction, therefore, this is an avenue for future work. Some designs were briefly considered for this topic as part of this thesis but were not realised. A designs considered was one based on network graph visualisations [HSS15].

The dataset used within this thesis has comprised of a month's worth of data, almost 5 million calls. Expanding applications to deal with a larger dataset of a year or over is another avenue for future work. This has many additional challenges with data handling and storage.

While previous work on call-centre visualisation has focused on providing visual analytics on a call-centre dataset, real-time visualisation of call-centre data provides a large field for future work. This can be for real-time analysis of call-centre performance, or for customer relationship management.

Appendix A.

An Open-Source, Hardware Accelerated Industry Application for Rendering Multidimensional Glyphs

"I hear and I forget. I see and I remember. I do and I understand."
— Confucius – Teacher, Politician, Philosopher (551–479 BC)

This appendix provides technical details of the rendering technique utilised in Chapter 4. A simplified open-source version is presented to aid reproducibility. Reflections are also made of the industry collaboration of this thesis. This chapter is based on a publication [RL20] presented at the International Conference on Computer Graphics, Visualization, Computer Vision and Image Processing in July 2020.

A.1. Introduction and Motivation

Glyphs provide a popular means for conveying multidimensional data, however their shape and layout can be complex and therefore computationally intensive to calculate their geometry. This is especially true for glyphs where the geometry of the glyph is dependent on the data dimensions to be represented. With an increasing number of glyphs required for ever growing data-sets, this issue is amplified, requiring longer computational time to render a complete image. This becomes a challenge where glyphs are used for interactive exploration of data, with scenes requiring continuous

update. Scenes that are slow to update can be frustrating for users, making adoption of a tool with such a feature difficult.
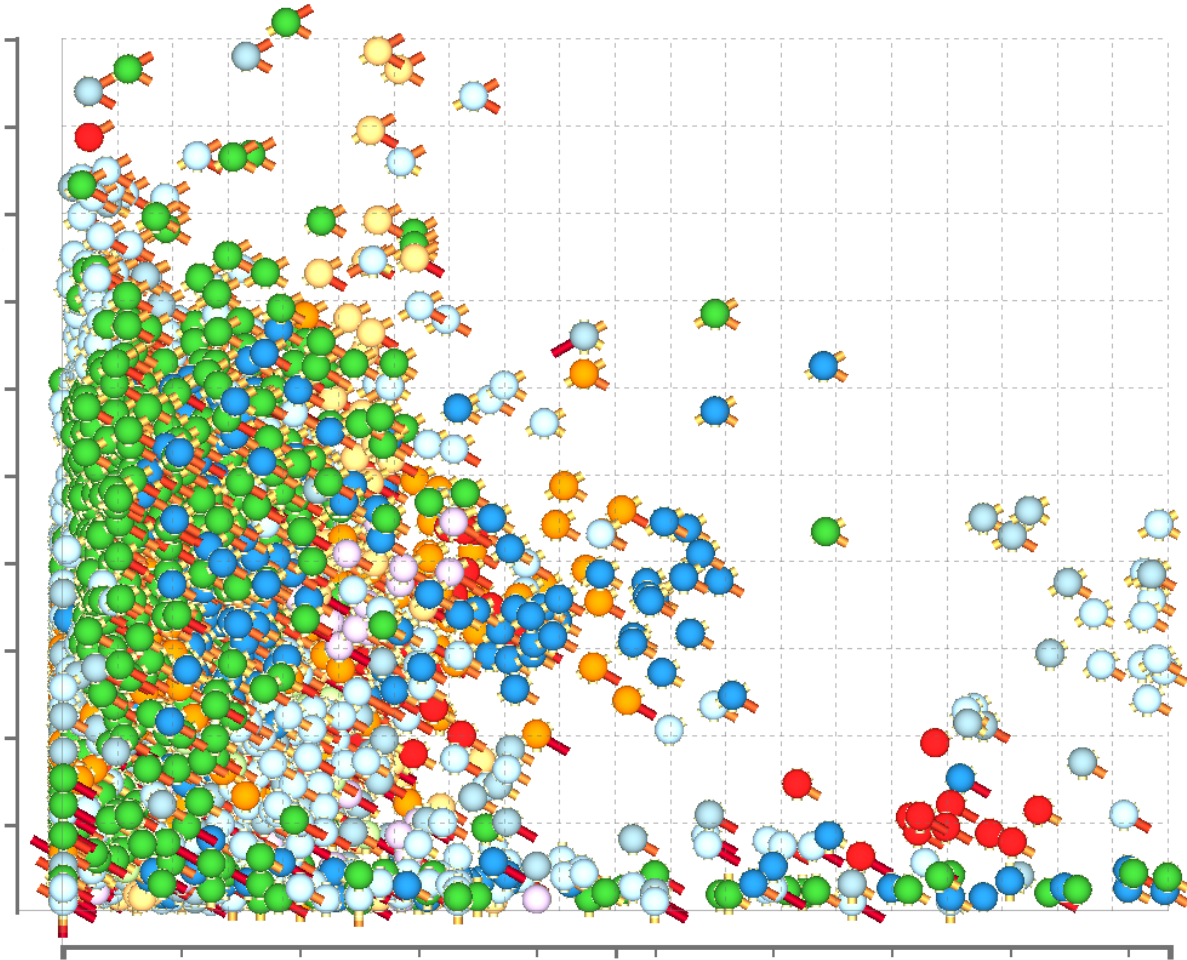
Figure A.1.: Over 6,500 glyphs drawn utilising the technique presented in this appendix. Each glyph features six continuous variables plus a categorical variable mapped to colour. The x and y position of the glyph on the canvas are also mapped to variables.

In this appendix we provide technical details of our software implementation for rendering a large number of multi-dimensional glyphs such as seen in Figure A.1. This is done by leveraging the parallel processing capabilities of a Graphics Processing Unit (GPU). The application was developed specifically for an industry partner.

The reproduction of techniques presented within the visualisation community can often be challenging. Application implementations made to demonstrate techniques are seldom made available. This is especially true for projects where a commercial product may be viable and therefore an implementation may be concealed. To address this challenge and to provide a good example to the community, we present an

open-source example of the technique introduced in this appendix. Open-source implementations help bridge the gap between academia and industry [Lar17a].

The contributions of this appendix are:

- A novel implementation for calculating and rendering large numbers of multidimensional data-glyphs

- A method to support easy and fast selection of glyph design

- The provision of a self-contained open-source code implementation to demonstrate the technique

The rest of the appendix is organised as follows: A background of OpenGL is provided in Section A.2 and related work is examined in Section A.3. The glyph rendering technique is described in Section A.4. A discussion is provided in Section A.5 and a conclusion drawn in Section 2.3. Finally a description of the code is provided in Section A.6.

A.2. Background

Open Graphics Library (OpenGL) is a graphical programming API, enabling hardware accelerated rendering of 2D and 3D graphics. It provides cross-platform support and can be used with many programming languages and is therefore used in a number of commercial applications especially in the gaming industry. Since OpenGL version 2.0, OpenGL has supported the OpenGL Shading Language (GLSL), a high level, C-based shader language. Shaders are short text-based programs that run on a Graphics Processing Unit (GPU), enabling greater rendering control and features such as per pixel shading and shadowing.

The basic operation of GLSL requires a minimum of two shaders, a vertex shader and a fragment shader. A vertex shader operates on every vertex point supplied with data inputs such as the position, the normal, and the colour. Meanwhile a fragment shader operates on each fragment (pixel) of the renderer calculating interpolated values and textures to output fragment colours.

Since the first release of GLSL [The20], additional optional shader objects have been included into the standard, these being the tessellation control, tessellation evaluation,

geometry, and compute shaders, enabling greater control of the OpenGL rendering pipeline. Figure A.2 shows the OpenGL rendering pipeline.
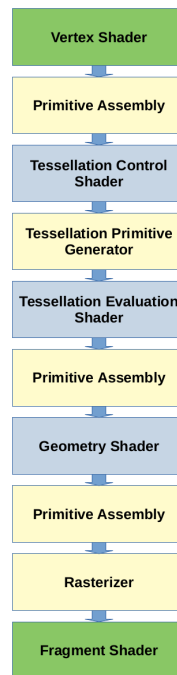
Figure A.2.: The GLSL rendering pipeline [The20]. Boxes shaded green are compulsory programmable shaders, with light blue boxes representing optional programmable shaders. Yellow boxes represent non programmable sections of the pipeline

The tessellation shaders are responsible for sub-dividing primitives (such as a triangle) into multiple smaller primitives. Geometry shaders take the primitives output from the vertex shader, or optionally the fragment shaders, and are able to output zero to many primitives from these. Compute shaders are utilised to leverage the parallel processing power of GPUs for computation and may be used separately from the rendering pipeline.

For a comprehensive description of the capabilities and operation of the OpenGL standard, we refer readers to a book by Kessenich *et al.* [KSS16].

Data is passed into the rendering pipeline through the use of memory buffers on the GPU. This data holds attributes such as the position and colour of each vertex, enabling all vertices to be processed simultaneously, exploiting the parallel processing power of GPUs.

A.3. Related Work

A survey of data visualisation books in Chapter 2 highlight six OpenGL textbooks that introduce and detail the operation of OpenGL [AS11, Wol11, Mov13, WJHSL15, KSS16, LL15].

McNabb and Laramee [ML17] describe four surveys addressing glyph design [War02, BKC*13, FIBK17, ROP11]. The work of Ward presents a taxonomy of glyph placement strategies [War02]. Borgo *et al.* provide design guidelines for glyphs [BKC*13]. The work of Fuchs *et al.* surveys glyph based user-studies [FIBK17] and Ropinski *et al.* survey glyphs for medical visualisation [ROP11]. Consultation of these surveys fails to identify work that discusses rendering implementation techniques for glyphs.

The use of glyph optimisation work is most prominent in scientific visualisation, with glyphs often used to represent flow [PGL*11, PLCZ09] and for molecular visualisation [GKM*14]. Because of the large number of glyphs required to effectively visualise the flow or molecules, optimisations through the use of GPU rendering techniques are widely developed [CM93, RE05, GRE09, PPvA*09, AKCL19].

Falck *et al.* compile and provide detailed explanations of GPU accelerated algorithms for rendering and processing scientific data [FGKR16]. As such, we recommend readers consult this work for a comprehensive overview of GPU accelerated glyph rendering.

Our work differs from others as we utilise GPU shaders for the construction of multi-dimensional data glyphs for information visualisation rather than scientific visualisation. In contrast to previous work, the technique we present leverages the geometry shader to construct each individual data-glyph geometry. We also provide a seldom-seen open-source implementation.

A.3.1. Open Source Provision

For ease reproducibility, Van Wijk provided a simplified code example of the Image Based Flow Visualization technique presented [VW02]. This technique has shown to be be reproducible as it has been extended in a number of other works [LJH03, vW03, LvJH04, TS06]. Source code was made available on a DVD provided with the

paper as well as provided in text form in supplementary material. This highlights the challenges related to the endurance of the code as a DVD is not accessible to the wider community, however, the supplementary material of the work is still accessible. In addition to this an online resource is now available providing the source code to a simple example as well as the implemented application presented in the original paper ensuring its accessibility, 18 years after publication.

To follow the example set by Van Wijk, we provide a simplified code of the technique presented in this chapter in Section A.6, and as a repository on GitHub [Ree20b].

A.4. Glyph Rendering

In this section, we discuss step-by-step the process for replicating the technique that we present in this chapter. The technique treats each glyph as a point primitive and passes the data variables from a vertex buffer through the vertex shader to the geometry shader. Once in the geometry shader, data values are used to determine the complex geometry of each glyph before being passed on in the pipeline to be rendered. This technique enables for complex glyphs to be rendered using a single draw call.

A.4.1. Preprocessing

Before glyphs can be rendered, data must first be processed into a format that can be loaded into vertex buffers for rendering. In our use case, we were supplied with more than 6,500 data instances, each with 20 continuous data variables plus a categorical value. The first step is to find the maximum and minimum value for each of the variables. Once this has been established, the next step is to normalize each of the variables to a range $[0, 1]$.

The data is then ready to be stored into vertex buffers to be transferred to the GPU memory. Before this, the number of variables each glyph is to represent must be established a priori. We demonstrate glyph designs in Figure A.3 featuring six continuous variables plus a categorical value mapped to colour. As well as the glyph variables, we also map the position of the glyphs to a two dimension scatterplot enabling each glyph to represent an additional two variables. For the construction of
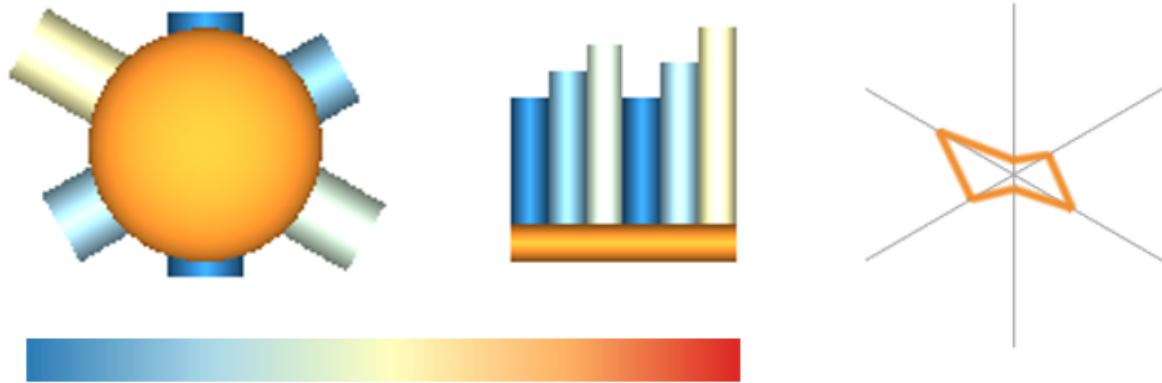
Figure A.3.: Three glyph designs, each of which feature six continuous variables represented by the size of the bars in the left two glyphs and by the radar on the right glyph. A categorical variable is also represented by colour, indicated by the orange in these examples. Colour is also used to redundantly map to the bar size on the right two glyphs, as indicated by the pictured colourmap.

these glyphs therefore requires nine data variables to be copied into vertex buffers and onto the GPU memory to be processed by the rendering pipeline.

The OpenGL rendering is done by calling the `glDrawArrays` command, specifying each glyph as a point primitive using the `GL_POINTS` symbolic constant.

A.4.2. Vertex Shader

The vertex shader determines the positioning of the glyph, and simply passes the other data through to the geometry shader. The location is set using the `gl_Position` function as can be seen in the code in Figure A.4. As the glyph is being placed on a 2D scene, only the x and y portion of the output vector are assigned as variables with the other portions assigned as constants.

A.4.3. Geometry Shader

The geometry shader is used to determine the shape of the glyph according to the data variables passed from the vertex shader. An example of geometry shader code can be seen in Figure A.5, this code is used to draw the centre glyph seen in Figure A.3. The geometry shader for the left glyph can be found in Section A.6 or on the online repository [Ree20b].

```
 1 #version 460
 2
 3 layout(location = 0) in vec2 vertex_pos;
 4 layout(location = 1) in float cat_color;
 5 layout(location = 2) in float barAattr;
 6 layout(location = 3) in float barBattr;
 7 layout(location = 4) in float barCattr;
 8 layout(location = 5) in float barDattr;
 9 layout(location = 6) in float barEattr;
10 layout(location = 7) in float barFattr;
11 out float barA;
12 out float barB;
13 out float barC;
14 out float barD;
15 out float barE;
16 out float barF;
17 out float color;
18
19 void main() {
20     gl_Position = vec4(vertex_pos, 0.0f, 1.0f);
21     barA = barAattr;
22     barB = barBattr;
23     barC = barCattr;
24     barD = barDattr;
25     barE = barEattr;
26     barF = barFattr;
27     color = cat_color;
28 }
```

Figure A.4.: Vertex shader code. Eight data variables are input and seven are directly output to the geometry shader. The remaining variable is used to establish the glyph position using the `gl_Position` function.

The geometry shader takes in the point vertex position from the vertex shader and outputs a series of vertices in a triangle strip to build a bar chart glyph. First the shader calculates the position in the screen-space, then iterates through each data variable to calculate the size of each corresponding bar, emitting five vertices for each bar to the fragment shader using the `EmitVertex()` function. Details being emitted are the positions for each new vertex, a colour attribute to map the bar colour to the data value, a 2D vector for shading effects, and an attribute to distinguish the data bars from the categorical identifying bar. Once the data bars have been emitted, vertices for an additional categorical identifying bar are emitted. In total, 35 vertices are emitted from the geometry shader in Figure A.5.

A variety of different geometry shaders, creating different styles of glyphs, can be used and easily switched by re-running the OpenGL program with the different shader.

```glsl
1  # version 460
2
3  uniform mat4 matrix;
4  layout(points) in;
5  in float barA[];
6  in float barB[];
7  in float barC[];
8  in float barD[];
9  in float barE[];
10 in float barF[];
11 in float color[];
12 layout (triangle_strip, max_vertices = 200) out;
13 out float barColor;
14 out float pass;
15 out vec2 position;
16
17 float glyphSize = 0.06f;
18 float barWidth = glyphSize/3.0;
19 float noOfVariables = 6.0;
20 float halfVar = noOfVariables/2.0;
21
22 void main() {
23     vec4 center = gl_in[0].gl_Position;
24     vec4 pos = matrix*center;
25
26     //draw variable bars
27     pass = 0;
28     for (int i=0; i<noOfVariables;i++) {
29         float barValue = 0.0f;
30         switch (i) {
31         case 0: barValue = barA[0]; break;
32         case 1: barValue = barB[0]; break;
33         case 2: barValue = barC[0]; break;
34         case 3: barValue = barD[0]; break;
35         case 4: barValue = barE[0]; break;
36         case 5: barValue = barF[0]; break;
37         default: barValue = 0.0f;  break;
38         }
39         float barMultiply = barValue*glyphSize + glyphSize;
40         barColor = barValue;
41
42         float xPosAdj = float(i)-halfVar;
43         gl_Position = (pos + vec4(xPosAdj*barWidth, 0.0-glyphSize, 0.0, 0.0));
44         position =vec2(0.f,0.f);
45         EmitVertex();
46         gl_Position =  (pos + vec4((xPosAdj+1)*barWidth, 0.0-glyphSize, 0.0, 0.0));
47         position =vec2(1.f,0.f);
48         EmitVertex();
49         gl_Position =  (pos + vec4(xPosAdj*barWidth, barMultiply-glyphSize, 0.0, 0.0));
50         position =vec2(0.f,1.f);
51         EmitVertex();
52         gl_Position =  (pos + vec4((xPosAdj+1)*barWidth, barMultiply-glyphSize, 0.0, 0.0));
53         position =vec2(1.f,1.f);
54         EmitVertex();
55         gl_Position =  (pos+ vec4((xPosAdj+1)*barWidth, 0.0-glyphSize, 0.0, 0.0));
56         position =vec2(1.f,0.f);
57         EmitVertex();
58     }
59
60     //draw indicator bar
61     pass = 1;
62     barColor =  (color[0]);
63     gl_Position = vec4(pos.x +halfVar*barWidth,pos.y - glyphSize, pos.z,1.0);
64     position =vec2(0.f,0.f);
65     EmitVertex();
66     EmitVertex();
67     gl_Position = vec4(pos.x +halfVar*barWidth,pos.y - glyphSize - barWidth, pos.z,1.0);
68     position =vec2(1.f,0.f);
69     EmitVertex();
70     gl_Position = vec4(pos.x - halfVar*barWidth,pos.y-glyphSize, pos.z,1.0);
71     position =vec2(0.f,1.f);
72     EmitVertex();
73     gl_Position = vec4(pos.x - halfVar*barWidth,pos.y - glyphSize-barWidth, pos.z,1.0);
74     position =vec2(1.f,1.f);
75     EmitVertex();
76 }
```

Figure A.5.: Geometry shader code for a six variable bar chart plus categorical identifier.
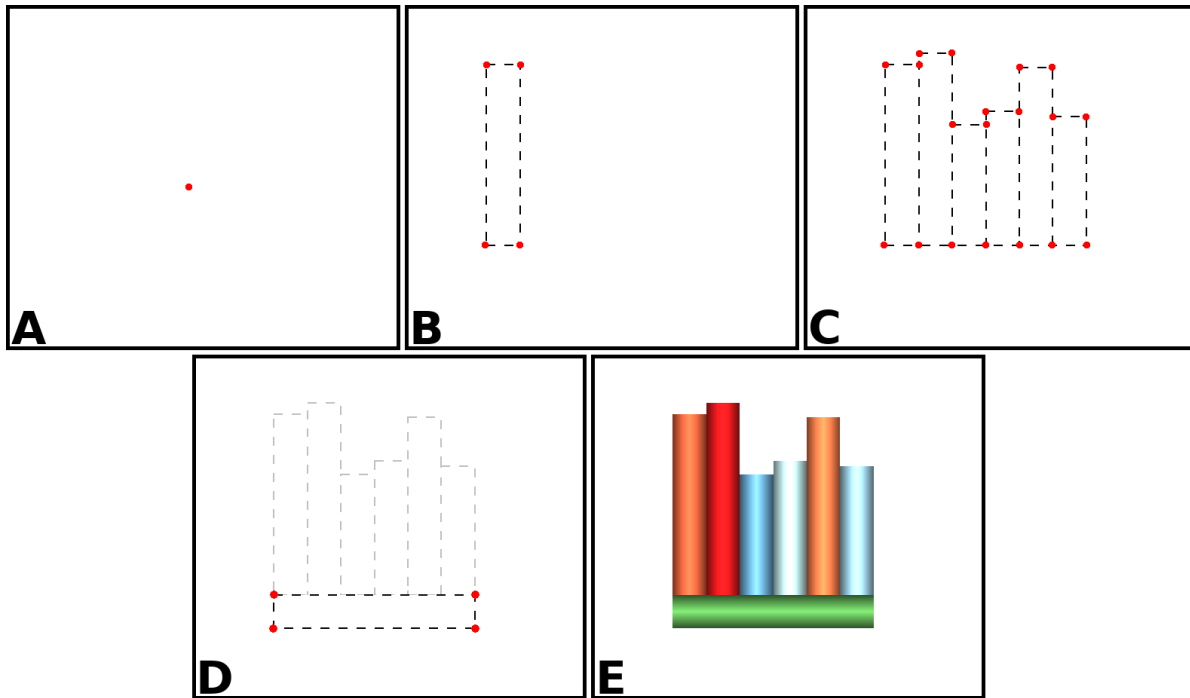
Figure A.6.: OpenGL shader glyph construction. A shows the single vertex position output from the vertex shader. B shows the output of a single bar from lines 42–57 of the geometry shader, while C shows the output of all bars in the encapsulating `for` loop, lines 28–58. The category indicator bar shown in D is output from lines 60–75 from the geometry shader. Finally, colour is added in E by the fragment shader (see Section A.6).

A.4.4. Fragment Shader

Once a vertex has been emitted by the geometry shader, it is processed by the fragment shader. The fragment shader adds shading effects and colour to the vertices emitted from the geometry shader. A texture can also be added to the glyphs, therefore, if a glyph design incorporates an identifying symbol, this can be added as a texture at this stage of the process.

Figure A.6 shows the contribution each part of the shader makes to the construction of the multidimensional bar glyph.

A.4.5. Limitations

A limitation exists in the amount of data that can be passed between shaders. This is dependent on the graphics hardware, for our consumer grade Nvidia GeForce

GTX 1060 6 GB graphics card this is 1024 floating points of data. For example, the
geometry shader example in Figure A.5 has 8 floating point outputs for every vertex
emit, capable of a maximum of 128 vertices to be emitted which may be reached using
a glyph with more complex geometry.

A.4.6. Open Source Code

A self-contained, simplified open source implementation of the code is available in
Section A.6 and on GitHub:

    https://github.com/dylan-rees/glyphRenderer

This open source code is provided to our industry collaborator to increase repro-
ducibility. The license allows free use of the library to use, copy, modify, merge,
publish, distribute, sub-license, and/or sell copies of the software, and to permit
persons to whom the software is furnished to do so (MIT License).

A.5. Discussion

Figure A.1 shows a screenshot of glyphs rendered using the presented technique,
supporting interactive frame rates. The number of glyphs that can be interactively
draw utilising this method is only limited by the hardware capabilities. However with
the games industry driving GPU development, this limitation becomes less relevant
as new GPUs increase capabilities.

Drawing so many glyphs highlights an overplotting issue where the glyphs are
drawn on top of one another become indistinguishable, as can be seen in Figure A.1.
Because the technique introduced in this appendix allows for interactive frame rates, a
user can interactively zoom and explore the glyphs, partially alleviating this issue. A
better solution to this is an avenue for future work.

A.5.1. Performance

To gauge the performance of the presented technique, the number of frames rendered
each second was measured for a varying number of glyphs. Results are shown in Table

Number of Glyphs	Frame Rendering Time (ms)					
	Bar Glyph (With Commodity Graphics Card)	Circular Glyph (With Commodity Graphics Card)	Star Glyph (With Commodity Graphics Card)	Bar Glyph (Without Graphics Card)	Circular Glyph (Without Graphics Card)	Star Glyph (Without Commodity Graphics Card)
1,000	<1	<1	<1	2.0	7.6	1.9
2,500	<1	2.0	<1	3.9	13	3.6
5,000	1.2	4.0	<1	6.9	34	6.4
7,500	1.7	5.6	1.3	9.9	53	9.1
10,000	2.2	7.3	1.7	13	58	12
20,000	4.2	14	3.1	25	113	22
30,000	6.0	21	4.5	44	140	40
40,000	8.0	28	6.0	58	188	52
50,000	10	35	7.4	60	235	54
60,000	12	42	9.0	72	284	65
70,000	14	48	10	84	330	76
80,000	15	55	12	95	377	87
90,000	17	61	13	107	424	97
100,000	20	71	14	119	471	108
110,000	21	77	16	131	517	118
120,000	23	82	17	143	564	130
130,000	25	91	19	155	611	140
140,000	27	98	20	167	685	151
150,000	29	105	21	178	705	162
160,000	30	109	23	190	752	172
180,000	35	129	26	214	845	194
200,000	39	141	29	238	939	216
220,000	42	155	31	262	1030	237

Table A.1.: Frame time rendering performance for drawing differing numbers of glyphs.

A.1. Figure A.7 shows the performance of the open-source implementation presented in this chapter. Performance is measured on two separate personal computers (PC).
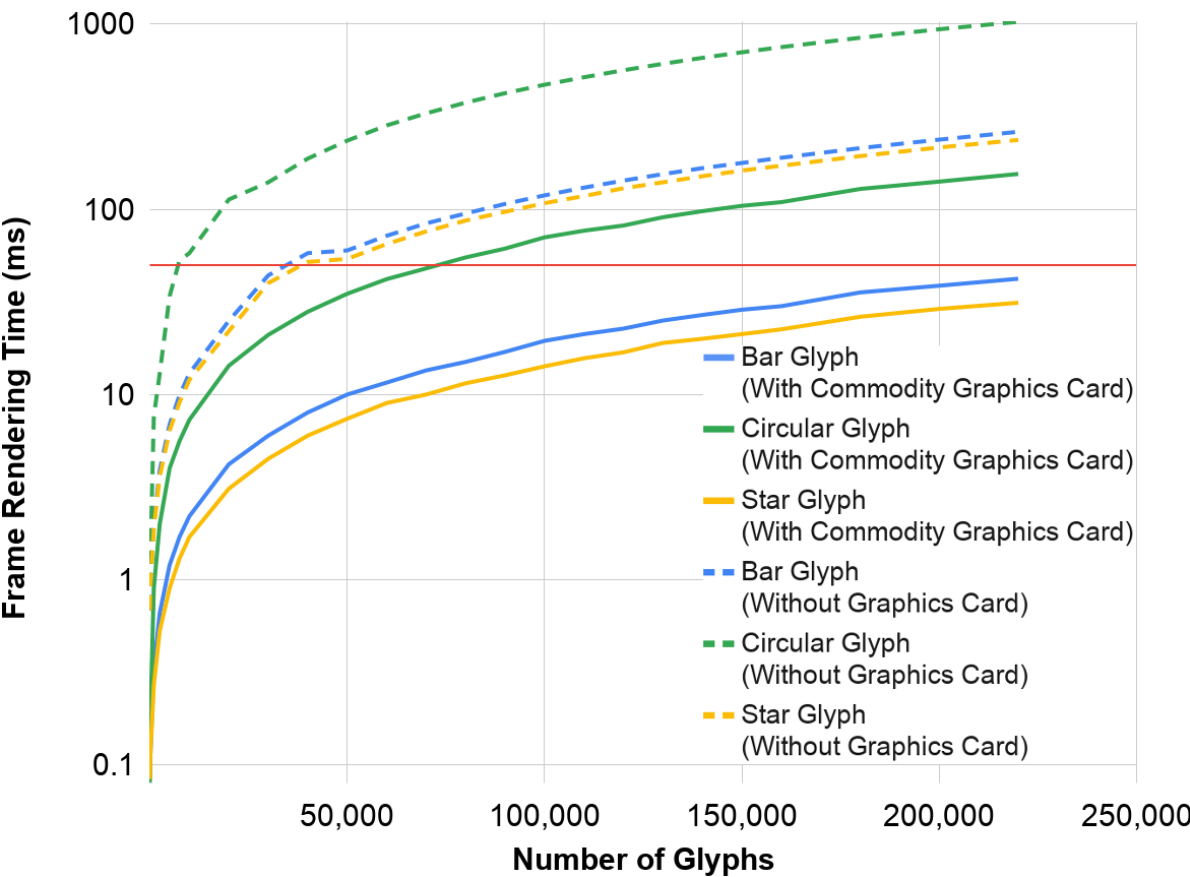
Figure A.7.: Rendering performance against the number of glyphs drawn for three different glyphs with and without a commodity graphics card.

The first PC features an Intel i7-6700HQ processor, 8GB of RAM, and an NVIDIA GeForce GTX 1060 6 GB graphics card. The second PC features an Intel i5-6500 processor, 16GB of RAM, and has no dedicated graphics card, relying on integrated graphics (Intel HD Graphics 530). For compatibility, testing was performed using OpenGL version 4.5.

Performance is also measured for three glyph types, a bar chart glyph, depicted in the center of Figure A.3, a circular glyph, depicted on the left of Figure A.3, and a star chart glyph depicted on the right of Figure A.3.

Due to a limitation on array sizes, the maximum number of glyphs rendered is limited to 220,000. The PC including the graphics card achieved higher frame rates in comparison to the PC without for the same number of glyphs. Due to more calculations being required to compute the geometry, the circular glyph is slower than the bar chart glyph. In all scenarios, a frame rendering time of less than 17 ms was achieved for up to 2,500 glyphs (equivalent to 60 fps) and 50 ms (20 fps) for up to 7,500 glyphs.

Over 50 ms the user will notice severe roughness. The PC without a graphics card takes 35 ms to render a frame with 5,000 circular glyphs, and 58 ms to render 10,000 circular glyphs but remains under the 17 ms threshold for 10,000 bar chart and star glyphs. The 50 ms frame rendering threshold is only crossed for 40,000 bar chart or star glyphs.

The PC with the graphics card remains under a 17 ms frame rendering time for over 20,000 glyphs, with over 70,000 circular glyphs rendered before the frame rendering time increases over the 50 ms threshold. For the bar chart glyphs, the PC with the graphics card remains under 17ms for 90,000 glyphs and under 40 ms for 200,000 glyphs. Over 120,000 star glyphs are rendered before the frame rendering time increases over 17ms and remains under 30ms for 200,000 star glyphs. With every glyph design tested, both with and without a graphics card, overplotting becomes a greater issue before frame rates drop below an interactive rate.

A.5.2. Source Code Provision

Reproducing visualisation methods from previous work can be challenging. In many works, prototype implementations are presented, however the availability of these implementations is scarce. This is especially true for tools developed in partnership with commercial entities, or with commercialisation of a product in mind. This is an example of differences between the rational in industry and academia. A discussion of the dichotomy of interests between industry and academia is provided by Roberts *et al.* [RLB*18b]. Despite this, collaboration between industry and academia is sought after by a number of interested parties as evident by panels discussing the topic in both academic and industrial domains [Lar17a]. Because of this, a compromise should be achievable to enable sample code to be provided, without compromising commercial interests.

We provide open-source code of an implementation of the technique presented ai Section A.6 and as an online repository to ensure reproducibility. This is despite the fact that the technique presented here was conceived in the development of an application with a commercial partner. A screen shot of the open-source program can be seen in Figure A.8.

To mitigate the challenge of reimplementing previously presented methods, we propose that guidelines and best practice should be assembled for the use of prototype
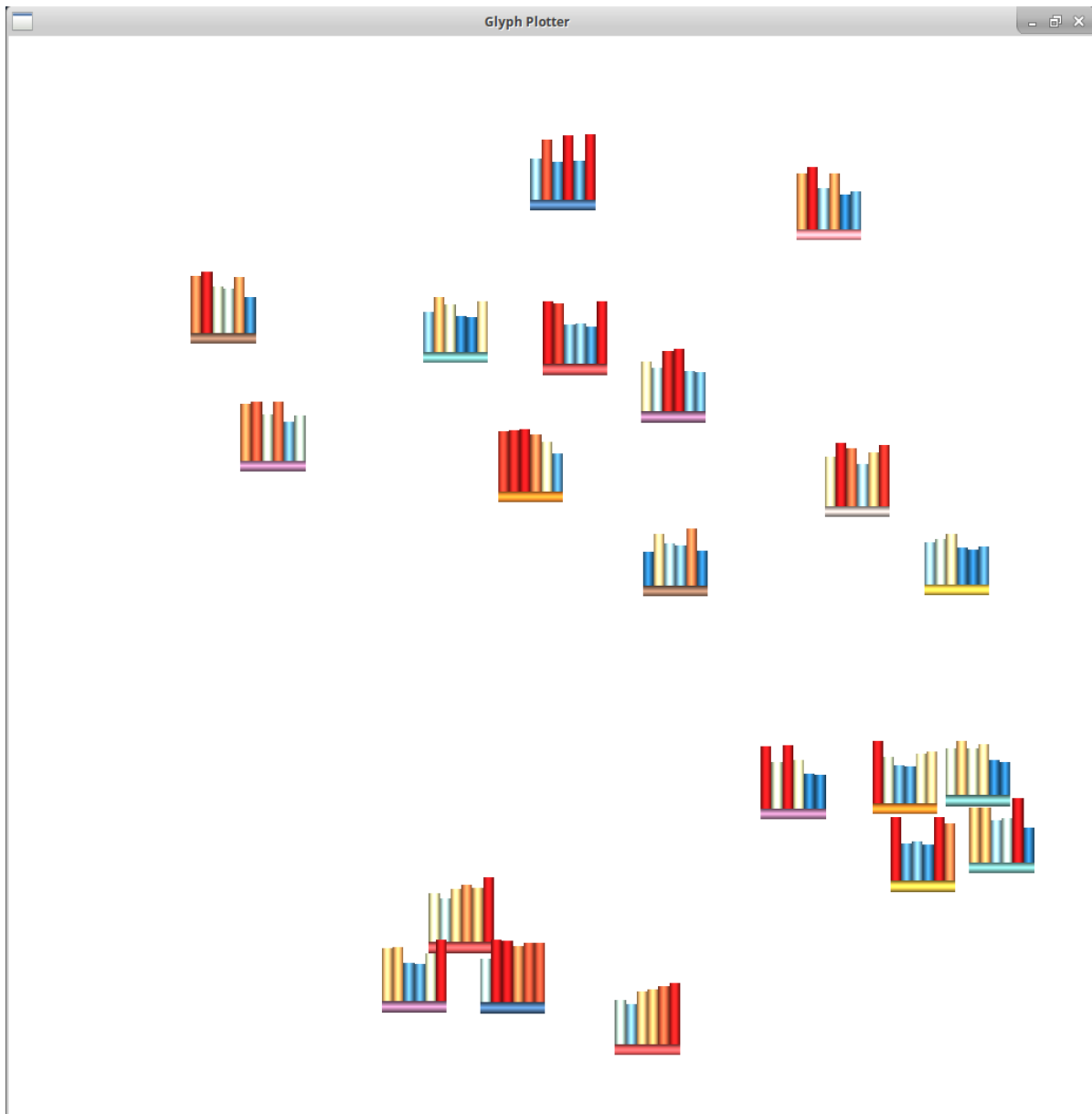
Figure A.8.: A screenshot of the open-source implementation featuring 20 randomly generated bar glyphs.

visualisation tools, advocating for an open source implementation of the methods presented. This may not be possible in scenarios such as where commercially sensitive data is used, however best efforts should be made to provide a simplified tool demonstrating the presented method.

Once guidelines have been established, it is then the responsibility of the visualisation community to ensure that they are adhered to. This is firstly a personal commitment, ensuring that one's own work meets guidelines. Following this, mentors

to new practitioners within the field should establish the use of the guidelines. Finally as reviewers, awareness of the guidelines can be promoted and encouragement for their adoption.

A.5.3. Collaboration with Industry

Here we present a discussion of our experience in collaborating with an industry partner, for whom applications and techniques created to aid their data analysis. We consider this collaboration to have been very successful, with numerous academic publications as an outcome [RTL*16, RLB*18b, RLS*19, RRL*18b, RRL*19] and techniques developed being included into industrial products.

The collaboration initially started as an Innovate UK project [UK 20], followed by two separate KESS2 PhD studentships [KES20]. Both the Innovate UK and KESS2 projects supplied grant funding, reducing the level of investment risk for the company. For the investment made by the company, in return they received exposure to state-of-the-art techniques in information visualisation and visual analytics, and research into specific visualisation designs for their data.

The success of the collaboration, in our opinion was due to a number of aspects. Good communication between the parties involved, good team work and enthusiasm on both sides was key. Having a aligned goals was another important factor and a collection of personalities that fit together well. An important aspect is having relatively easy access to data on the academic side, which is not always the case during collaborations but was a benefit in our collaboration.

We feel that the successful collaboration enabled the technology transfer that bridges the gap between research software and the use of techniques in the intended domain. Sedlmair *et al.* outline pitfalls do be aware of when collaborating [SMM12]. Comparing these potential pitfalls with our own experience we feel that most were avoided, however, we identify additional pitfalls suffered. These additional pitfalls include feature creep, where collaborators ask for additional features, and the departure of a primary collaborator from the company. The departure of the primary collaborator and an additional secondary domain expert near the end of the projects bought about the end of the collaboration after five years.

A.6. Open Source Code Description

Here is detailed how the technique is presented in a simple program. Four libraries are utilised to provide functions for the operation of the program. The *GLEW* library provides access to OpenGL functions, the *GLFW* library provides the user interface, window, and key interaction functions, file reading functions are provided from the *stdio* library, and the *stdlib* library provides a random number generator for data creation. All libraries provide cross platform support. The defined constants are as follows:

`NPIX` Resolution of the OpenGL canvas.

`MAX_SZ` The maximum size of a shader script.

`NO_GLYPHS` The number of glyphs to generate and render.

The global variables are:

`proj_matrix[]` A 4 by 4 matrix establishing the OpenGL viewport projection. The matrix establishes the lower left corner of the canvas as $[-0.1, -0.1]$ and the top right corner $[1.1, 1.1]$.

`cat_colormap[]` The discrete colormap for the category indicator. Colours are provided as RGB unsigned integers between 0–255.

`cat_map_sz` The number of colour variables in `cat_colormap[]`.

`data_colormap[]` The continuous colormap for data mapping. Colours are provided as RGB unsigned integers between 0–255.

`data_map_sz` The number of colour variables in `data_colormap[]`.

`pos_sz` The number of position coordinates for a single instance.

The main program starts by initialising GLFW and then creating the interface window. Following this, eight arrays are initiated for data storage. The first array, `points`, is for storing glyph positions, with two figures for each glyph representing the x and y coordinates. The second array, `cols`, stores categorical values for each glyph. The final six arrays store data variables that determine the glyph geometry. A `for` loop is then used to populate the category glyph with integers between 1–10, and the position array with both x and y values 0–1.

A Vertex Array Object is created and bound ready for the addition of the first Buffer Array Object which is created and populated with the `points` array. A second Buffer Array Object is then created and populated with the `cols` array. Following this, Buffer Array Objects are created and the `randomArray` function is used to generate random data to populate the data variable arrays and to add each to a Buffer Array Object.

The `draw` function is then called. This function is used to compile the OpenGL program and render scenes. Within the `draw` function, the `loadShader` function takes three GLSL shader file names as an input. This function then reads in each shader file in turn and attaches them to compile a shader program, assigns the projection matrix to the program, and creates and attaches textures to the program.

Once a shader program is created, a rendering `while` loop is entered. Within the while loop, the canvas is cleared and set to a white background. The `glDrawArrays` function is then used to draw the scene. Following this, *GLFW* functions are used to check for keyboard inputs before the loop repeats. If a *B* or *C* key is pressed, the `realoadShader` function recompiles the shader with bar and circular glyph respectively. The *ESCAPE* key causes the program to close.

glyphRenderer.cpp

```
1    /*-------------------------------------------------------*/
2    /* glyphRenderer.cpp - Multidimensional Glyph Rendering */
3    /*                                                       */
4    /* Dylan Rees, 2020  -  Swansea University               */
5    /*-------------------------------------------------------*/
6    #include <GL/glew.h>
7    #include <GLFW/glfw3.h>
8    #include <stdio.h>
9    #include <stdlib.h>
10
11   #define NPIX  1025
12   #define MAX_SZ 100000
13   #define NO_GLYPHS 4
14
15   GLfloat proj_matrix[] = {
16   1.667f, 0.0f, 0.0f, 0.0f,
17   0.0f, 1.667f, 0.0f, 0.0f,
18   0.0f, 0.0f, 1.0f, 0.0f,
19   -0.833f, -0.833f, 0.0f, 1.0f
20   };
21   GLubyte cat_colormap[] = {
22   78,121,167, 242,142,44, 225,87,89,
23   118,183,178, 89,161,79, 237,201,73,175,
24   122,161, 255,157,167, 156,117,95,
25   186,176,171
26   };
27   int      cat_map_sz = 10;
28   GLubyte data_colormap[] = {
29   44,123,182, 171,217,233, 255,255,191,
30   253,174,97, 215,25,28
31   };
32   int      data_map_sz = 5;
33   int      pos_sz = 2;
34   /*-------------------------------------------------------*/
35   GLuint makeShader( const char* shader_filename,
36   GLenum shaderType ) {
37   char sdr_str[MAX_SZ];
38   FILE* sdr_file = fopen( shader_filename, "r" );
39   size_t shdr_sz = fread( sdr_str, 1,
40   MAX_SZ - 1, sdr_file );
41   sdr_str[shdr_sz] = '\0';
42   fclose( sdr_file );
43
44   GLuint shader_handle   = glCreateShader( shaderType );
45   const GLchar *shader_ptr = (const GLchar *)sdr_str;
46   glShaderSource( shader_handle, 1, &shader_ptr, NULL );
47   glCompileShader( shader_handle );
48
49   return shader_handle;
50   }
51   /*-------------------------------------------------------*/
52   GLuint loadShader(const char* vert_shad_filen,
53   const char* frag_shad_filen,
54   const char* geom_shad_filen ) {
55
56   GLuint shader_program = glCreateProgram();
57   GLuint vert_shad_handle = makeShader(vert_shad_filen,
58   GL_VERTEX_SHADER);
59   GLuint frag_shad_handle = makeShader(frag_shad_filen,
60   GL_FRAGMENT_SHADER);
61   GLuint geom_shad_handle = makeShader(geom_shad_filen,
62   GL_GEOMETRY_SHADER);
63
64   glAttachShader( shader_program, vert_shad_handle );
65   glAttachShader( shader_program, frag_shad_handle );
66   glAttachShader( shader_program, geom_shad_handle );
67
68   glLinkProgram( shader_program );
69   glDeleteShader( vert_shad_handle );
70   glDeleteShader( frag_shad_handle );
71   glDeleteShader( geom_shad_handle );
72
73   glLinkProgram(shader_program);
74   int matrix_location = glGetUniformLocation(shader_program,
75   "matrix");
76   int cat_tex_location = glGetUniformLocation(shader_program,
77   "cat_texture");
78   int attr_tex_location = glGetUniformLocation(shader_program,
79   "attr_texture");
80   glUseProgram( shader_program );
81   glUniformMatrix4fv( matrix_location, 1, GL_FALSE,
82   proj_matrix );
83   glUniform1i(cat_tex_location, 0);
84   glUniform1i(attr_tex_location,  1);
85
86   GLuint cat_textureID;
87   glGenTextures(1, &cat_textureID);
88   glActiveTexture(GL_TEXTURE0 + 0);
89   glBindTexture(GL_TEXTURE_1D, cat_textureID);
90   glTexImage1D( GL_TEXTURE_1D, 0, GL_RGB, cat_map_sz, 0,
91   GL_RGB, GL_UNSIGNED_BYTE, cat_colormap );
92   glGenerateMipmap( GL_TEXTURE_1D );
93   glTexParameteri( GL_TEXTURE_1D, GL_TEXTURE_WRAP_S,
94   GL_CLAMP_TO_EDGE );
95   glTexParameteri( GL_TEXTURE_1D, GL_TEXTURE_WRAP_T,
96   GL_CLAMP_TO_EDGE );
97   glTexParameteri( GL_TEXTURE_1D, GL_TEXTURE_MAG_FILTER,
98   GL_NEAREST );
99   glTexParameteri( GL_TEXTURE_1D, GL_TEXTURE_MIN_FILTER,
100  GL_NEAREST );
101
102  GLuint attr_textureID;
103  glGenTextures(1, &attr_textureID);
104  glActiveTexture(GL_TEXTURE0 + 1);
105  glBindTexture(GL_TEXTURE_1D, attr_textureID);
106  glTexImage1D( GL_TEXTURE_1D, 0, GL_RGB, data_map_sz, 0,
107  GL_RGB, GL_UNSIGNED_BYTE, data_colormap );
108  glGenerateMipmap( GL_TEXTURE_1D );
109  glTexParameteri( GL_TEXTURE_1D, GL_TEXTURE_WRAP_S,
110  GL_CLAMP_TO_EDGE );
111  glTexParameteri( GL_TEXTURE_1D, GL_TEXTURE_WRAP_T,
112  GL_CLAMP_TO_EDGE );
113  glTexParameteri( GL_TEXTURE_1D, GL_TEXTURE_MAG_FILTER,
114  GL_LINEAR );
115  glTexParameteri( GL_TEXTURE_1D, GL_TEXTURE_MIN_FILTER,
116  GL_LINEAR );
117
118  return shader_program;
119  }
120  /*-------------------------------------------------------*/
121  void reloadShader( GLuint* program,
122  const char* vert_shad_filen,
123  const char* frag_shad_filen,
124  const char* geom_shad_filen ) {
125
126  GLuint reloaded_program = loadShader( vert_shad_filen,
127  frag_shad_filen,
128  geom_shad_filen );
129  if ( reloaded_program ) {
130  glDeleteProgram( *program );
131  *program = reloaded_program;
132  }
133  }
134  /*-------------------------------------------------------*/
135  void randomArray( float* array, GLuint* buffer,
136  int attrib_pos) {
137
138  for (int i=0;i<NO_GLYPHS;i++) {
139  array[i] = float(rand()) / float(RAND_MAX);
140  }
141  glBindBuffer(GL_ARRAY_BUFFER, *buffer);
142  glBufferData(GL_ARRAY_BUFFER, NO_GLYPHS * sizeof(float),
143  array, GL_STATIC_DRAW);
144  glVertexAttribPointer( attrib_pos, 1, GL_FLOAT,
145  GL_FALSE, 0, NULL);
146  glEnableVertexAttribArray(attrib_pos);
```

```
147    }
148    /*-----------------------------------------------------*/
149    void draw(GLFWwindow* window, GLuint vao) {
150
151    GLuint shader_program = loadShader( "vert.glsl",
152    "frag.glsl",
153    "geom.glsl" );
154
155    while(!glfwWindowShouldClose(window)) {
156    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
157    glClearColor( 1.0f, 1.0f, 1.0f, 1.0f);
158    glUseProgram(shader_program);
159    glBindVertexArray(vao);
160    glDrawArrays(GL_POINTS, 0, NO_GLYPHS);
161    glfwPollEvents();
162    glfwSwapBuffers(window);
163    if ( glfwGetKey( window, GLFW_KEY_B ) ) {
164    reloadShader( &shader_program, "vert.glsl",
165    "frag.glsl", "geom.glsl" );
166    }
167    if ( glfwGetKey( window, GLFW_KEY_C ) ) {
168    reloadShader( &shader_program, "vert.glsl",
169    "frag.glsl", "geomAlt.glsl" );
170    }
171    if(glfwGetKey(window, GLFW_KEY_ESCAPE)) {
172    glfwSetWindowShouldClose(window, 1);
173    }
174    }
175    }
176    /*-----------------------------------------------------*/
177    int main() {
178
179    glfwInit();
180    GLFWwindow* window = glfwCreateWindow(NPIX, NPIX,
181    "Glyph Plotter",
182    NULL, NULL);
183    glfwMakeContextCurrent( window );
184    glewExperimental = GL_TRUE;
185    glewInit();
186
187    float points[pos_sz*NO_GLYPHS];
188    float cols[NO_GLYPHS];
189    float varA[NO_GLYPHS];
190    float varB[NO_GLYPHS];
191    float varC[NO_GLYPHS];
192    float varD[NO_GLYPHS];
193    float varE[NO_GLYPHS];
194    float varF[NO_GLYPHS];
195    for (int i=0;i<NO_GLYPHS;i++) {
196    cols[i] = rand() % 10 + 1;
197    points[i*pos_sz] = float(rand())/float(RAND_MAX);
198    points[i*pos_sz+1] = float(rand())/float(RAND_MAX);
199    }
200    GLuint vao = 0;
201    glGenVertexArrays(1, &vao);
202    glBindVertexArray(vao);
203
204    GLuint points_vbo = 0;
205    glGenBuffers(1, &points_vbo);
206    glBindBuffer(GL_ARRAY_BUFFER, points_vbo);
207    glBufferData(GL_ARRAY_BUFFER, pos_sz* NO_GLYPHS * sizeof(float),
208    points, GL_STATIC_DRAW);
209    glVertexAttribPointer(0, pos_sz, GL_FLOAT,
210    GL_FALSE, 0, NULL);
211    glEnableVertexAttribArray(0);
212
213    GLuint colours_vbo = 0;
214    glGenBuffers(1, &colours_vbo);
215    glBindBuffer(GL_ARRAY_BUFFER, colours_vbo);
216    glBufferData(GL_ARRAY_BUFFER,
217    NO_GLYPHS * sizeof(float),
218    cols, GL_STATIC_DRAW);
219    glVertexAttribPointer(1, 1, GL_FLOAT,
220    GL_FALSE, 0, NULL);
221    glEnableVertexAttribArray(1);
222
223    GLuint varA_vbo = 0;
224    glGenBuffers(1, &varA_vbo);
225    randomArray(varA, &varA_vbo,2);
226    GLuint varB_vbo = 0;
227    glGenBuffers(1, &varB_vbo);
228    randomArray(varB, &varB_vbo,3);
229    GLuint varC_vbo = 0;
230    glGenBuffers(1, &varC_vbo);
231    randomArray(varC, &varC_vbo,4);
232    GLuint varD_vbo = 0;
233    glGenBuffers(1, &varD_vbo);
234    randomArray(varD, &varD_vbo,5);
235    GLuint varE_vbo = 0;
236    glGenBuffers(1, &varE_vbo);
237    randomArray(varE, &varE_vbo,6);
238    GLuint varF_vbo = 0;
239    glGenBuffers(1, &varF_vbo);
240    randomArray(varF, &varF_vbo,7);
241
242    draw(window, vao);
243    glfwTerminate();
244    return 0;
245    }
```

vert.glsl

```
1    #version 460
2
3    layout(location = 0) in vec2 vertex_pos;
4    layout(location = 1) in float cat_color;
5    layout(location = 2) in float barAattr;
6    layout(location = 3) in float barBattr;
7    layout(location = 4) in float barCattr;
8    layout(location = 5) in float barDattr;
9    layout(location = 6) in float barEattr;
10   layout(location = 7) in float barFattr;
11   out float barA;
12   out float barB;
13   out float barC;
14   out float barD;
15   out float barE;
16   out float barF;
17   out float color;
18
19   void main() {
20   gl_Position = vec4(vertex_pos, 0.0f, 1.0f);
21   barA = barAattr;
22   barB = barBattr;
23   barC = barCattr;
24   barD = barDattr;
25   barE = barEattr;
26   barF = barFattr;
27   color = cat_color;
28   }
```

geom.glsl

```
1    # version 460
2
3    uniform mat4 matrix;
4    layout(points) in;
5    in float barA[];
6    in float barB[];
7    in float barC[];
8    in float barD[];
9    in float barE[];
10   in float barF[];
11   in float color[];
12   layout (triangle_strip, max_vertices = 200) out;
13   out float barColor;
14   out float pass;
15   out vec2 position;
16
17   float glyphSize = 0.06f;
18   float barWidth = glyphSize/3.0;
19   float noOfVariables = 6.0;
20   float halfVar = noOfVariables/2.0;
21
22   void main() {
23   vec4 center = gl_in[0].gl_Position;
24   vec4 pos = matrix*center;
25
26   //draw variable bars
27   pass = 0;
28   for (int i=0; i<noOfVariables;i++) {
29   float barValue = 0.0f;
30   switch (i) {
31   case 0: barValue = barA[0]; break;
32   case 1: barValue = barB[0]; break;
33   case 2: barValue = barC[0]; break;
34   case 3: barValue = barD[0]; break;
35   case 4: barValue = barE[0]; break;
36   case 5: barValue = barF[0]; break;
37   default: barValue = 0.0f;  break;
38   }
39   float barMultiply = barValue*glyphSize + glyphSize;
40   barColor = barValue;
41
42   float xPosAdj = float(i)-halfVar;
43   gl_Position = ( pos + vec4(xPosAdj*barWidth,
44   0.0-glyphSize, 0.0, 0.0));
45   position =vec2(0.f,0.f);
46   EmitVertex();
47   gl_Position =  ( pos + vec4((xPosAdj+1)*barWidth,
48   0.0-glyphSize, 0.0, 0.0));
49   position =vec2(1.f,0.f);
50   EmitVertex();
51   gl_Position =  ( pos + vec4(xPosAdj*barWidth,
52   barMultiply-glyphSize, 0.0, 0.0));
53   position =vec2(0.f,1.f);
54   EmitVertex();
55   gl_Position =  ( pos + vec4((xPosAdj+1)*barWidth,
56   barMultiply-glyphSize, 0.0, 0.0));
57   position =vec2(1.f,1.f);
58   EmitVertex();
59   gl_Position =  ( pos+ vec4((xPosAdj+1)*barWidth,
60   0.0-glyphSize, 0.0, 0.0));
61   position =vec2(1.f,0.f);
62   EmitVertex();
63   }
64
65   //draw indicator bar
66   pass = 1;
67   barColor =  (color[0]);
68   gl_Position = vec4( pos.x +halfVar*barWidth,
69   pos.y - glyphSize, pos.z,1.0);
70   position =vec2(0.f,0.f);
71   EmitVertex();
72   EmitVertex();
73   gl_Position = vec4( pos.x +halfVar*barWidth,
74   pos.y - glyphSize - barWidth, pos.z,1.0);
75   position =vec2(1.f,0.f);
76   EmitVertex();
77
78   gl_Position = vec4( pos.x - halfVar*barWidth,
79   pos.y-glyphSize, pos.z,1.0);
80   position =vec2(0.f,1.f);
81   EmitVertex();
82   gl_Position = vec4( pos.x - halfVar*barWidth,
83   pos.y - glyphSize-barWidth, pos.z,1.0);
84   position =vec2(1.f,1.f);
85   EmitVertex();
86   }
```

frag.glsl

```
1    #version 460
2
3    in float barColor;
4    in float pass;
5    in vec2 position;
6    uniform sampler1D attr_texture; //data map
7    uniform sampler1D cat_texture; //cat map
8    uniform mat4 matrix;
9    layout (location = 0) out vec4 color;
10
11   float ambient = 0.5;
12   float opacity = 1.0f;
13   float no_cat = 10.0f;
14
15   void main() {
16   vec3 Normal;
17   Normal =vec3(position,0.f);
18   vec4 colors;
19   float barC =float(barColor/(no_cat + 1.0f));
20
21   if (pass < 1.0) {
22   colors = texture( attr_texture, barColor);
23   Normal.z= 1.f-2*(abs(position.x-0.5));
24   } else {
25   colors = texture( cat_texture, barC);
26   Normal.xy = position * 2.0 - vec2(1.0);
27   float mag = dot( Normal.xy, Normal.xy );
28   Normal.z = 1.f-2*(abs(position.x-0.5));
29   if (pass > 1.0) {
30   if (mag > 1.0) {
31   discard;
32   }
33   Normal.z = sqrt(1.0-mag);
34   }
35   }
36   colors.w = opacity;
37
38   vec4 temp = matrix * vec4(0.f,0.f,1.f,0.f);
39   vec3 lightDir = normalize(temp.xyz);
40   float diffuse = clamp(dot(lightDir, Normal),0,1);
41   color = colors*(vec4(vec3(ambient),1.f) +
42   vec4(vec3(diffuse),1.f));
43   }
```

Appendix B.

The Industry Engagement Ladder

This appendix chapter outlines mechanisms for industry to collaborate with academia. This aims to address issues of collaboration between industry and academia and is based on a publication in the Journal of Industry-University Collaboration [RWL20].

Purpose: This chapter is a response to a frequently asked question from prospective industry partners searching for opportunities to collaborate with the Computer Science Department of Swansea University, UK. This chapter seeks to provide structured guidance in the form of what is titled the Industry Engagement Ladder, shown in Figure B.1.

Design/methodology/approach: The Industry Engagement Ladder presents a selection of collaboration opportunities, outlining the possible cooperation mechanisms between an industry partner and the academic institution.

Findings: Opportunities are described and ranked according to the amount of investment required by the industrial partner, and therefore risk, from low to high.

Originality/value: This concise chapter provides a quick reference for perspective industry academia collaboration opportunities, the risks involved with each, the possible funding available to help foster these partnerships, and the benefits to industry. These collaborative activities (and variations thereof) can be incorporated by any university department. The chapter offers an ideal starting point for industry – university collaboration.

Keywords: Engagement, Opportunities, Collaboration, Industry–University

	Activity			Risk	Investment
1	R		Joint Innovate UK project	3	4
2	R		Joint KTP, Industrial CASE projects	3	3
3	R		Joint supervision/sponsorship of PhD student	3	3
4	R		Local conference sponsorship/exhibition	2	2
5	R		Joint publications	2	1
6	T&R		Joint supervision/sponsorship of MRes student	2	2
7	R		Advisory board of funded projects	1	1
8	T		Joint supervision/sponsorship final year/ MSc thesis	1	1
9	T&R		Project fair sponsorship/exhibition	1	1
10	T		Guest lecture	1	1

T = Teaching, R = Research

Figure B.1.: The Industry Engagement Ladder, outlining collaboration opportunities for industry partners looking to collude with academia. The higher up the ladder, the higher the investment and therefore risk. Greater investment usually corresponds to a longer relationship.The level of risk has been graded on a scale from 1 – very low to 5 – very high. The level of investment is also graded on a scale from 1 – 5; 1 – negligible costs, 2 – <£5K, 3 – <£50K, 4 – <£500K, 5 – >£1million

B.1. Introduction and Motivation

There is a growing body of evidence to suggest that industry-academic collaboration provides benefits for both parties. Shneiderman provides evidence that academic papers co-authored with a corporate partner generate a higher impact [Shn18]. Additionally a systematic review by Perkman et al. [PTM*13] shows a positive correlation between academic success and academic collaboration. However, there are known to be gaps between industry and academia [Lar17a]. For industry partners, collaboration with academia provides an avenue for knowledge transfer, raising awareness of the state-of-the-art technology, providing access to specialists in the field and opportunities to potential hires. Collaboration also offers an opportunity to advertise

to talented students and develop a relationship with them for potential employment. Collaboration also mitigates risk of testing new potentially risky, ideas and projects.

The contributions of this chapter are:

- The provision of concise and practical examples of industry-higher education collaborations.

- Explicit recommendations outlining the opportunities for industry collaboration with the Computer Science Department at Swansea University.

- A ranking of collaboration opportunities by investment and risk, see Figure B.1.

These ideas are generalised and can be applied to any cross-disciplinary configuration. The remainder of this chapter begins with related work in Section B.2, the Industry Engagement Ladder is presented in Section B.3. Section B.5 presents a discussion and future work.

B.2. Related work

Industry-university collaboration has long been recognised as a beneficial process and has long been a subject of investigation. As such, many research papers have been published on the subject. We identify six survey papers that address the topic of university-industry collaboration: [PTM*13,AAT15,GPO16,dWdVDvdWG18,MFM18]. Perkmann et al. provide a systematic review of papers published between 1980–2011 concerned with academic engagement, and compare their results with academic commercialisation and how the two differ [PTM*13]. They define academic commercialisation as the creation of intellectual property or the forming of joint enterprises. Specific aspects of collaboration, such as the type and extent of engagement, and the consequences of engagement form the categorisation of the findings and discussion. Their findings suggest that academic success and academic engagement are positively correlated.

Ankrah and Al-Tabbaa [AAT15] provide a systematic review of 109 studies spanning 1990–2014, classified based on five main aspects; types of collaboration, collaboration motivation, collaboration formation and operation, facilitators and inhibitors, and outcomes. They present a list of methods for industry-university collaboration as a part of their survey. The list modifies work previously presented by Bonaccorsi

and Piccaluga [BP94] which presents six categories of collaboration: personal informal relationships, personal formal relationships, third party, formal targeted agreements, formal non-targeted agreements, and focused structures. Ankrah and Al-Tabbaa extend this to include additional forms of collaborations they identify in their survey [AAT15]. The list is organised according to the resource involvement from the university, for example a low involvement interaction is for an industry partner to deliver a lecture, classified as a personal informal lecture, while a high involvement activity would be funding a university post, classed as a formal non-targeted agreement. This chapter bears resemblance to this work and explicitly outlines collaboration opportunities within the UK academic arena, providing a detailed account of each opportunity according to ranking.

Garousi et al. [GPO16] provide a systematic review of industry-academia collaborations in the software engineering sector to identify the challenges and potential pitfalls of such collaborations such that future collaborators may remain aware of them [GPO16]. They also provide an inventory of best practices for collaboration in the sector. The authors provide a detailed description of the paper selection and review process. They provide four classifications of the papers that they review comprising of the knowledge area that they address, the contribution type, the research type, and the scale of the review. Papers are also reviewed for collaboration models, challenges encountered, and best practice advice. A comprehensive list of categorised challenges of industry-academia collaboration are supplied along with a complete list of best practices.

Mascarenhas et al. analyse co-cited references of university-industry collaboration to visualise clusters of topics they form [MFM18]. The analysis is performed using VOSviewer software [VEW09] which constructs bibliometric maps. Using this technique, categorisation of papers is performed automatically into clusters. Four clusters are identified: absorption capacity, knowledge and competitiveness of collaborations, impact of knowledge spillovers on collaborations, strategic alliances for industry innovation, and university-industry cooperation. The top cited papers in each category are highlighted. They are also able to identify areas of university-industry collaboration that require more research, such as the monetary and social effects of these collaborations.

Wit-de Vries et al. review 35 papers, focusing on the topic of knowledge transfer in industry-university partnerships [dWdVDvdWG18]. They identify barriers and facilitators for knowledge transfer between university and industry partners. Three
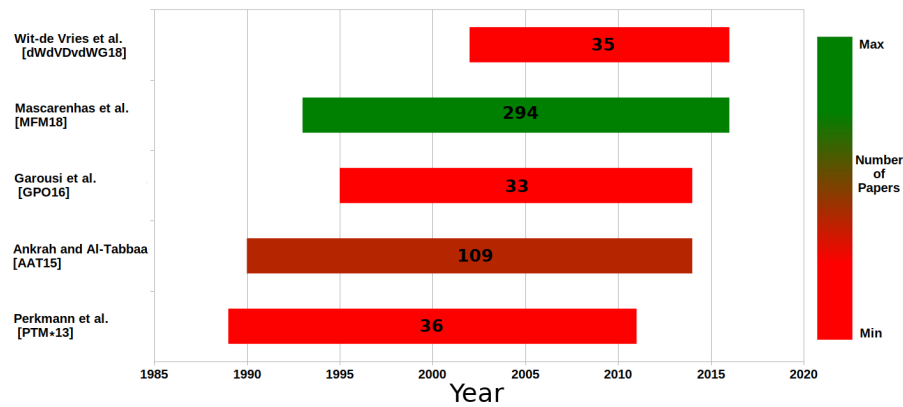
Figure B.2.: A visualisation of the citation distribution for university-industry collaboration surveys. The number of papers each review surveys are indicated on the bars and indicated by the colour-map. See Section B.2 on related work.

themes of knowledge transfer barriers are identified which form the basis of their classification. These themes are cognitive difference, institutional differences, and social capital. Wit-de Vries et al. provide a future research agenda following their work. This was compiled from inconsistencies found during their review, for example the use of the term cultural difference can be ambiguous.

Roberts et al. have explored the motives for collaboration from both the industrial and academic side, and how they do not always align [RLB*18b]. They provide recommendations for both industrial and academic partners to ensure a successful collaboration.

A book by Shneiderman [Shn16] articulates the need for research to be more focused on applied problems and to combine research across science, engineering, and design disciplines. A section on industry-university collaboration highlights its benefits. Firstly, industry have real-world problems that university researchers can address as opposed to theoretical problems. Secondly, industry has data that universities would find difficult to obtain themselves. And finally universities have a number of young researchers, whereas industries have resources for them to use.

Shneiderman does caution against some of the pitfalls of collaboration however. He notes that it can take time for each party to understand the internal processes of one another which could cause barriers. He also notes that often universities and businesses have different policies with regards to confidentiality, with businesses wanting to keep a competitive advantage over their competitors, and universities looking to publish results.

Our work differs from previous work in that it offers concrete examples, based on practice and experience, in the form of an investment and risk ladder. These examples can be used by any department interested in collaboration with industry. We use each of these in the Department of Computer Science at Swansea University.

Engagement Opportunity	Level of Risk	Time Commitment (Duration)	Monetary Investment	Benefits to Industry Partner	Benefits to University Partner	Examples
Guest lecture	Very low	1-2 hours	Travel costs	Recruitment, Networking	Inspiration for students	The Visible Lunch [Lar19b]
Project fair sponsorship/exhibition	Very low	1 day	None	Recruitment, Networking	Employment opportunities for students	Swansea University project fair [Lar19a]
Joint supervision/sponsorship of final year/MSc thesis	Very low	An average of 1 hour per week (6 months)	None	Test high risk project(s), Recruitment	Real world experience for students	HEOR Ltd., QPC Ltd.
Advisory board of funded projects	Very low	~5 hours per quarter (up to 5 years)	None	Exposure to state-of-the-art, Research, Networking, Knowledge transfer		HEOR Ltd., QPC Ltd. Digital Health Labs, We Predict Ltd.
Joint supervision/sponsorship of MRes student	Low	An average of 1 hour per week (1 year)	£4k per year	Recruitment, Knowledge transfer, Additional labour, Test high risk project(s)	Real world experience for students	KESS projects [KES20]
Joint publication	Low	A few hours per month (6-12 months)	None	Exposure to state-of-the-art, Research, Knowledge transfer, Additional labour, Test high risk project(s)	Research publications	Publications: [RTL*16, RLB*18b, RRL*18b, RRL*18a, RRL*19, RLS*19]
Local conference sponsorship/exhibition	Low	1 day	In-kind, £100 - £5k	Recruitment, Networking, Exposure to state-of-the-art	Employment opportunities for students	EuroVis 2014 [Eur14], CGVC 2018 [EGU18]
Joint supervision/sponsorship of PhD student	Medium	(3-4 Years)	£11k per year	Research, Access to expertise, Knowledge transfer, Additional labour, Recruitment	Real world experience for students	Leadin UK Ltd., QPC Ltd.
Joint KTP, Industrial CASE projects	Medium	(12-36 months)	£35k per year	Research, Access to expertise, Knowledge transfer, Additional labour	Real world project, facilitate larger project	
Joint Innovate UK project	Medium	(6-24 months)	~ £10k-100k	Research, Access to expertise, Knowledge transfer, Additional labour	Real world project, facilitate larger project	Traydstream Limited [Inn18], QPC Ltd. [Inn15]

Table B.1.: A table summarising the engagement possibilities, the associated risks and commitments required, and our collaboration examples.

B.3. The Industry Engagement Ladder

There are a number of options for collaboration. They are presented as list starting with the smallest and lowest risk investment and then generally increasing as the ladder ascends. Figure B.1 shows the Industry Engagement Ladder. An indication also outlines the type of activity each rung represents, being a teaching activity or

a research activity. For most, it may make sense to start at the bottom of the ladder and work your way up as the collaborative relationship develops. Here we outline the details of each rung on the Industry Engagement Ladder. Table B.1 provides a summary of the engagement opportunities and the associated investments.

B.3.1. Guest Lecture

Guest lectures are a great way to recruit and advertise to students. For example, anyone may give a guest lecture in the software engineering module in the Computer Science program at Swansea University and we are always looking for industry sponsors for our group prizes. Past industrial sponsors include CA Technologies and Grid-Tools Ltd.

The Computer Science Department has many seminar series covering different topics. An example of which is the Visible Lunch [Lar19b]. The Visible Lunch event aims to bring together researchers at Swansea University (and surroundings) with a common interest in visual computing. This event encourages informal discussion, open exchange, and personal interaction in order to foster both research directions in visual computing and a sense of local community. Anyone may give a guest lecture at our regular Visible Lunch event. This can take place at any time of the year. Past guest lectures have been presented by the Reinshaw Group, City and County of Swansea, and General Dynamics.

We also invite industry collaborators to give invited talks at our Annual Undergraduate Colloquium at Gregynog [Lar14]. Since 1985, students and staff of the Computer Science Department attend an annual 2–3 day Undergraduate Computer Science Colloquium at Gregynog, the conference centre near Newtown, Wales. The aim of the colloquium is to provide:

- An overview of Computer Science, its research and applications, its history and future developments

- An opportunity for every student to give a presentation about their project

- An open discussion between staff and students on education in Computer Science

- A reflection on career prospects for Computer Science students

- Informal conversation between staff and students

Figure B.3.: An image from the annual undergraduate computer science colloquium at Gregynog. See Section B.3.1.

As a part of the colloquium every student presents their project work and invited speakers from both academia and industry talk on various aspects of Computer Science, including research, commercial, and industrial themes. In the past, the colloquium has hosted speakers from a variety of industries including BT, Fujitsu, IBM and DVLA, as well as presentations from other universities and other departments within Swansea University. An image of a previous colloquium can be seen in Figure B.3.

This option represents the lowest risk and investment for collaboration, requiring only a minimal investment of time and perhaps some minimal expenses.

B.3.2. Project Fair Sponsorship, Exhibition

Some universities showcase undergraduate projects to a wider audience of industrial partners and interested parties in an exhibition. At Swansea University, final year computer science students dedicate two semesters to their final year project as part of their degree qualification. Since 2008, these students present the results of their project to the departmental public as well as to local IT industry. The aim of the fair is to provide students with a professional environment where they:

- Present and defend their final year projects

Figure B.4.: An image from the annual undergraduate student project fair. See Section B.3.2.

- See how their own project fits into the context of the entire year's results

- Network professionally with IT companies

During the fair, students present by their exhibition space for discussions with visitors from the entire Swansea Bay area community. This is a great opportunity to come and meet the latest group of graduates as well as to network with visitors from local academia and industry. Industry-sponsored prizes are also awarded on the day. In the past, the fair has been sponsored by companies such as ERS Insurance Group and Health Economics and Outcomes Research (HEOR) Ltd. Figure B.4 shows an image of a previous project fair. Similarly to the guest lecture, this collaboration requires minimal resources and represents an extremely low investment.

B.3.3. Joint Supervision/Sponsorship of Final Year/MSc Thesis

Every final year undergraduate student at Swansea University works on a project for two semesters. They can choose to work with industry on their project. Students choose their project around April-May each year from a project selection brochure we distribute annually. We can include a collaborative project in the brochure. Laramee provides a sample list of data visualisation projects [Lar18]. This can serve as a low risk, low investment starting point for collaboration, enabling pilot-testing of an idea. Examples of companies collaborating with Swansea University in this way are Health Economics and Outcomes Research (HEOR) Ltd. and QPC Ltd. There are some

drawbacks however. Students may not choose an industry project. Also, a student that does choose such a project may not succeed at it.

Every masters student chooses a project for their thesis in the December-January time frame. A collaborative project may be included in their project selection brochure. This option is very similar to the third year project option although the students are more mature. One advantage of this option again is that it is very low risk and low investment and can serve as a starting point for collaboration. However, the same disadvantages as third year projects hold, a student may not choose this project. Also, only the best student projects provide a proof-of-concept to industry.

B.3.4. Advisory Board of a Funded Project

Academics are often applying for funding grants for research projects from funding bodies such as the EPSRC [EPS19b,EPS19a]. These funded projects are often supported by industrial partners to provide industrial expertise and validation of projects. For the industrial partner, this gives an insight into the latest state-of-the-art in research or provides a proof-of-concept. Companies that have collaborated with us in this way in the past include QPC Ltd., Digital Health Labs, HEOR Ltd. and We Predict Ltd.

This is another very low risk collaboration, again requiring only time as an input. However funded projects may run over a number of years so represents a slightly higher risk than collaboration opportunities on lower rungs of the ladder.

B.3.5. Joint Supervision/Sponsorship of MRes Student

Both Master of Research (MRes) and Master of Science by Research (MbyRes) degrees are research-oriented masters degrees. Students dedicate significantly more time and effort to their thesis than in the cases of a third year project or a taught (standard) masters degree. This enables the pilot-testing of a more substantial project. Funding opportunities can be used to attract high quality students. The Knowledge Economy Skills Scholarships (KESS 2) program funds high-quality MRes degrees [KES20]. KESS 2 links companies and organisations with academic expertise to undertake research projects that meet the needs of an active business or its sector. Companies that we have collaborated with on KESS 2 projects include Grid-Tools Ltd. (now CA Technologies), We Predict Ltd., Leadin UK Ltd. (now Gofore UK Ltd.) and QPC Ltd. KESS 2 is a

collaborative project supported by European Social Funds (ESF) through the Welsh Government involving all universities in Wales.

In this case the academic leads the application process. We have had a number of successful KESS-funded MRes and PhD degree students in conjunction with industry [KES20]. Application deadlines are usually between January to June each year. These degrees usually start in October or January each year. The industry contribution to KESS degrees is about £4K GBP per year. The chances of success are much higher in this case than BSc or MSc degree projects. These degrees tend to attract high quality students that dedicate a significant amount of time to their industry projects.

We rate this collaboration as low compared to very low with the opportunities on lower rungs of the Engagement Ladder. The reason for this is that some financial investment is required, although this is relatively small. Most of the investment comes from the academic side.

B.3.6. Joint Publications

Collaboration on joint publications are often welcome. This may be to provide industrial domain expert validation of a particular project. As an example, we cite joint publications between us and QPC Ltd. a call centre infrastructure provider [RTL*16, RLB*18b, RRL*18b, RRL*18a, RRL*19, RLS*19]. These publications present novel, feature-rich, and interactive application to visualise various aspects of up-to 5,000,000 calls. For this collaboration, QPC Ltd. provided a data-set, guidance, and domain expert feedback on the product developed. Another collaborating company is We Predict Ltd. whom also provided a data-set and guidance towards a joint publication [NAL14].

This is another low risk collaboration, with the risk dependant on the specific agreement of the collaboration, which may include a small financial contribution from the industry partner towards the project costs.

B.3.7. Local Conference Sponsorship, Exhibition

Universities often host conferences to showcase their latest research findings. These conferences can vary in size and exposure, ranging from small, to the international

Figure B.5.: An image from the 2014 EuroVis conference hosted at Swansea University. See Section B.3.7.

level. These conferences offer sponsorship opportunities for industry partners. In return, industry partners are able to use these conferences for exposure to a larger audience through an exhibition or by giving a presentation, providing advertising and recruitment opportunities.

The frequency of these conferences can be sporadic and for a variety of different computer science subjects. Examples of previously hosted conferences at Swansea University are EuroVis 2014 [Eur14] sponsored by Winton, Tableau, and NVIDIA amongst others (see Figure B.5), and CGVC 2018 [EGU18] sponsored by Computers Journal.

The investment risk of a conference sponsorship or exhibition is low, requiring only a relatively small financial contribution. This contribution is dependant on a number of factors such as the level of involvement and the nature and profile of the conference or exhibition.

B.3.8. Joint Supervision/Sponsorship of PhD student

PhD candidates may choose to work with industry on their research degrees. There is also the possibility to sponsor a PhD student through a match-funded (50/50%) PhD Studentship in conjunction with the College of Science at Swansea University. This program is normally not advertised publicly. The deadline for match-funding PhD studentship applications is usually once each year. The cost is around £11K GBP / year for 3 years for industry partners. PhD students can start 4 times per year at the beginning of each quarter (1 Jan, 1 Apr, 1 July, 1 Oct). There are many benefits to sponsoring a PhD student with an industry project. Appendix B.7 describes some more details and benefits.

KESS 2 [KES20], described in Section B.3.5, also has part-funded PhD Studentships with an emphasis on industrial collaboration. We have successfully collaborated on a number of KESS PhD studentships. Companies that we have jointly supervised a PhD student with on KESS projects include Grid-Tools Ltd., Leadin UK Ltd. and QPC Ltd.

B.3.9. Joint KTP, Industrial CASE Project

Knowledge Transfer Partnerships (KTP) are very popular for collaboration. The KTP scheme links a UK based business with an academic institution and provides a graduate Research Assistant (RA) [Inn19]. The RA normally resides at the industry partner for the majority of the time, with schemes lasting 12–36 months. This provides the business with academic expertise that they otherwise might not have access to for development of new prototypes and products. The application process is lead by the academic. The risk associated with this collaboration is low with small and medium-sized enterprises contributing around £35,000 per year, about one-third of the project costs.

Industrial Cooperative Awards in Science & Technology (CASE) are specifically for PhD studentships in conjunction with industry [EPS19c]. Students are required to spend a minimum of three months at the partner business, enabling an opportunity to explore novel research. This studentship is funded by the ESPRC [EPS19b], and represents a medium-risk collaboration.

Figure B.6.: An image of our Innovate UK collaboration with QPC Ltd. See Section B.3.10.

B.3.10. Joint Innovate UK Project

Innovate UK is part of a public body funded by the UK government to nurture innovation and create successful products from joint ideas [UK 20]. Innovate UK projects can be used to hire a dedicated Research Associate (RA) for varying lengths of time, typically from 6–24 months. The UK government typically covers about 70% of the company's costs on the project (and 100% of the academic's costs).

Swansea University has positive experiences and successes with Innovate UK programs and have some previous applications we are happy to share to get an idea of what an application is like [Inn15] (See Figure B.6. The application process is lead by the industry partner, however, we can certainly help an industry partner write an application. Examples of successful Innovate UK project collaborators include Traydstream Limited [Inn18], and QPC Ltd. [Inn15].

This collaboration requires the most investment in terms of time and financial resource, and therefore has the highest risk of all the collaborations outlined in this chapter. However an Innovate UK project also can provide the biggest benefit, with the resources available to produce a high quality prototype, product, research, or study.

Activity		Investment/Risk
R	Joint UKRI project	M
R	Joint bid development workshops/sandpits	M
R	Joint supervision/sponsorship of PhD student	M
R	Joint publications	M
T&R	Joint supervision/sponsorship of MRes student	L
R	Advisory board of funded projects	VL
T	Joint supervision/sponsorship final year/ MSc thesis	VL
T&R	Project fair sponsorship/exhibition	VL
T	Guest lecture	VL

T = Teaching, R = Research M = Medium, L = Low, VL = Very Low

Figure B.7.: The Interdisciplinary Academic Engagement Ladder, outlining what collaboration opportunities exist for university departments for interdisciplinary collusion. The higher up the ladder, the higher the investment and therefore risk. Greater investment usually corresponds to a longer relationship. The Interdisciplinary Academic Engagement Ladder is very similar to the Industry Engagement Ladder with minor variations due to available funding.

B.4. Academic Engagement Ladder

The Industry Engagement Ladder can be adapted to create a similar engagement structure for cross-department collaboration within the University or across universities. This is illustrated in Figure B.7.

Many of the engagement rungs are similar to that of the Industry Engagement Ladder, and so the engagements follow a similar protocol but with the academic partner instead of the industry partner.

The primary difference between the academic and industry engagement ladders are the absence of opportunities for conference sponsorship or exhibitions due to differing disciplines. Governmental initiatives for fostering industry-university collaboration are also not available without an additional industrial partner.

B.5. Future Work and Discussion

Collaboration enables exchange of ideas and resources to the benefit of all collaborating partners. However not all collaborations end in success. By providing an engagement ladder, opportunities for lower risk collaborations are indicated, allowing for collaboration where high exposure is not an option. Lower risk collaboration also enables testing a relationship, before climbing the ladder to a higher risk exposure.

The landscape of available grants is not fixed, successive governments may have different policies on research, therefore not all funding opportunities will be available in the future and other opportunities may also be introduced. To keep this document up-to-date with latest developments is future work.

B.6. Conclusion

In this chapter we outline the available opportunities for collaboration between industry and universities, drawn from our own experiences. We present this information in the form of a ladder with a ladder scale of increasing risk and reward. This manuscript offer an ideal starting point for industry – university collaboration.

B.7. Appendix: A Business Case for Sponsoring a PhD Student in Data Visualisation and Visual Analytics

Britain's universities play an essential role in shaping the UK's future and delivering prosperity, from educating future business leaders to developing innovations crucial to the nation's progress. The HE sector contributes more than £73 billion a year to the UK economy. Swansea University has a growing interest in outward facing collaboration including industry in order to maximise relevance and impact.

PhD candidates may choose to work with industry on their research degrees. Swansea University offers the possibility of sponsoring a PhD student through a match-funded (50/50%) PhD Studentship in conjunction with the College of Science. This program is generally not advertised publicly. The deadline for match-funding PhD studentship applications is usually at the beginning of February each year for

PhD candidates with an October start. This cost of sponsoring a PhD student is around 11.5K GPB per year for 3 years for partners. (The total cost of a PhD candidate is around £70K GBP.) PhD students can start 4 times per year at the beginning of each quarter (1 Jan, 1 Apr, 1 July, 1 Oct). Here are some great reasons to sponsor a PhD candidate:

- **Innovation:** One key to staying competitive in any market is innovation. Research is a key element to drive innovation. Academic leaders invest several thousand pounds every year in maintaining their knowledge of the most recent developments in the field on a global level.

- **Expertise:** Laramee, the PhD candidate's supervisor, has been studying and specialising in data visualisation and research since 1999. This kind of expertise is very rare in the UK. There are only a handful of academics specialising in visualisation across the entire UK. As an academic, he is also interested in maintaining a competitive edge in a global research field.

- **Inexpensive Labour:** Hiring a PhD candidate is much less expensive than an employee. This is a great opportunity to test out high-risk proof-of-concept prototype software. It's also a great opportunity to test out a prospective employee.

- **2 for the price of 1:** Although you may choose to sponsor one PhD student, you are really hiring two people, the candidate, and the PhD supervisor. The candidate's PhD supervisor works very closely with the PhD candidate on training, guidance, software development, and innovation.

- **Tax deductions for research:** Research and development (R&D) tax credits are a valuable government tax relief that rewards UK companies for investing in innovation. Companies that spend money developing new products, processes or services; or enhancing existing ones, are eligible for a cash payment and/or Corporation Tax reduction. R&D reliefs support companies that work on innovative projects in science and technology. It can be claimed by companies that incorporate research in order to advance in their market. It can even be claimed on unsuccessful projects. More information on Research and Development tax reliefs can be found on the Government website [GOV19]. Some companies may even generate a net profit from such collaboration.

Appendix C.

Domain Experts

This appendix provides a profile of the domain experts consulted during the compilation of this thesis.

C.1. Profiles

C.1.1. Gary A. Smith

Gary received his MBA from Leicester University in 2008 and a BSc in Mathematics & Statistical Computing from Liverpool John Moores University in 2006. He is also an Associate of the Operational Research society. Gary was the Director of Product & Marketing at QPC Group and was responsible for the strategic direction & innovation of QPCs products. Gary departed the collaboration at the end of August 2019 when moving to a new position at Aria Solutions in Canada.

C.1.2. Tony D'Cruze

Tony is a contact centre expert, with more than 20 years spent in a variety of contact centre roles in the Financial and Outsourcing sectors. Passionate about efficiency and driving improvements in customer service, Tony was a Consultant with QPC supporting clients with leveraging QPC solutions to transform their customer contact operations. Tony's involvement in the collaboration ended in order for him to take up a new position at QStory at the end of January 2020.

C.1.3. Paul Brookes

Paul is currently a lead developer at QPC Limited. Since joining Callscan Limited in 1989 he has continued to work on development of numerous computer telephony integration projects for contact centres providing real time and historic management information.His current interests include new and unique ways visualising of customer journey data for contact centres.

C.1.4. Aslam Miah

Aslam has worked as a call-centre agent for over two years with experience across sales and customer service departments. Aslam's inclusion in the collaboration began in January 2020 when his front line experience was sought.

Appendix D.

A Tribute to The Minutes of Meeting Protocol

D.1. Introduction

This appendix is a tribute to the minutes of meeting protocol which helped guide the PhD process [Lar10c]. The minutes of meeting protocol is a part of the The PhD in Visualization Starter Kit (PVSK), developed by Laramee to guide students through a PhD in data visualisation [Lar10d, Lar10a, Lar10b].

The minutes of meetings protocol was applied during this PhD candidature period, having been inherited from supervisors to PhD candidates whom become supervisors themselves. A presentation of the minutes of meeting protocol can be seen on YouTube [Lar17b].

Simple in nature, the minutes of meeting protocol provides clarity on the direction of the research, outlining the progress made and next research steps. The minutes protocol force all who attend the meeting to think of actionable steps to write in the 'to do' section, rather than having a vague sense of what needs to be done or being lost entirely. Having focused steps written, along with regular meetings, always facilitates progress rather than being stuck on a particular problem. Anecdotal evidence from other PhD candidates suggests that this is a common problem, with dips in progress and loss of passion often felt. This is exemplified by the popular PhD comics [Cha99].

In my personal experience, a problem was encountered during development of the scatterplot application presented in Chapter 3, this led to a loss of passion as progress

became retarded. However, with clear steps outlined in the minute meetings, progress was made and recovered swiftly afterwards.

The minutes of meeting also provide a memory of previous meetings. By consulting the previous meeting's minutes an immediate return to progress can be made following an absence or break. This also avoids repetition of work.

D.2. The Minutes of Meetings Protocol

The minutes of meeting protocol defines four sections to be completed. The first is a logistical section, outlining who was present at the meeting, the data and time of both the meeting start and end, and the date and time of the next meeting.

The second section summarises the topics discussed during the meeting, the third the progress made since the last meeting, and finally an outline of what to do for the next meeting.

D.3. Analysis

Having a record of meetings throughout the PhD candidature period allows for some analysis of the process to be made.

In total, 143 meetings were held as of writing this totalling over 236 hours.

An analysis is made of the text from the '*Topics Discussed*', '*Progress*', and '*To Do*' sections, compiled from all minutes. Stop words, as defined by the NLTK [NLT20], are removed and the most frequent words used are compiled. Figure D.1 presents a word cloud generated from the results using the Word Cloud Generator application [Gup15].

To enhance analysis, an application was built for visualising when in the course of the research occurrences of words in the minutes appeared. Figure D.2 provides a screenshot of the application and shows the instances of words chosen to describe the story of the PhD. The words chosen represent each key project and chapter in turn. The term '*book*' represents the survey provided in Chapter 2, with 119 instances of the word overall. This work was ongoing for the first 17 months of the PhD as reflected

Figure D.1.: A word cloud of the most common words used in the collated minutes collected from supervisory meetings, with stop words removed. The image is created using Word Cloud Generator [Gup15]. As the PhD is a research degree, it is unsurprising that the most prominent words are '*paper*' and '*draft*'. Other than these, strong themes of the thesis can be seen with '*agent*', '*book*' and '*books*', '*survey*', '*data*', '*visualization*', '*glyph*', '*user*', and '*calls*' being eminent.

in the visualisation, with an initial submission to EuroVis before being accepted for publication to Computer Graphics Forum following modification (major revision).
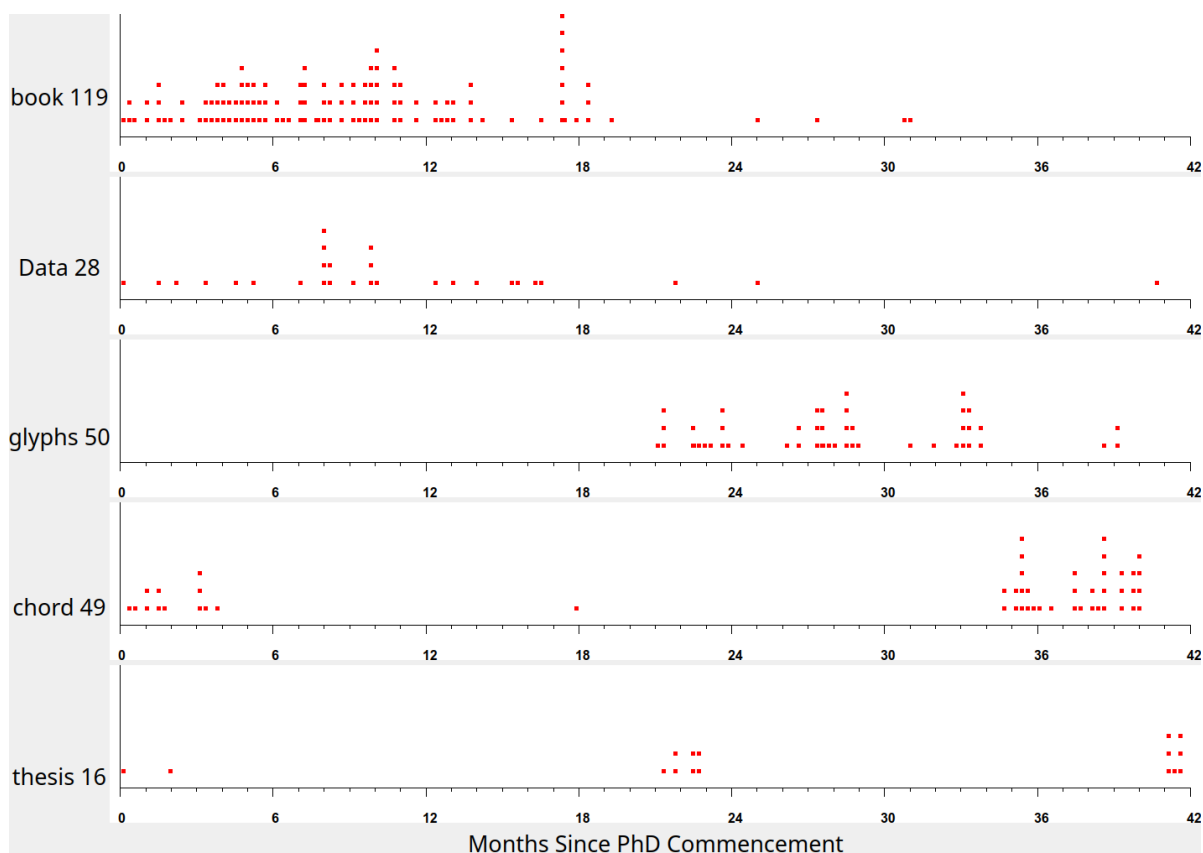
Figure D.2.: Five stacked scatterplots showing when key words were mentioned in the meeting
minutes. Each red dot represents an instance of the term, as indicated to the left
of each plot. Where words are repeated in a single meeting minutes, they are
stacked. The key words are chosen to represent some of the chapters of this thesis,
providing an insight into when the work for each was made.

No Single word could be found for the work presented in Chapter 3, however '*Data*'
was chosen. This project was mostly run in parallel with the book survey and was
published at the CGVC conference in the 24th month of the PhD before being extended
as a journal publication three months later again.

The term that best represents the work presented in Chapter 4 is '*glyphs*'. There
are 50 instances of '*glyphs*' in the minutes, predominantly occurring between months
21–34, the time in which the work in Chapter 4 was performed. Noticeable in the
representative plot for '*glyphs*' in Figure D.2 is a gap for months 25 and 29–31, this is
due to annual leave and the Christmas break periods.

To represent the work of Chapter 5, the term '*chord*' is used. From the plot in Figure
D.2 it is observable that there are two distinct clusters. The first cluster at the beginning
of the PhD represent the trial chord diagram work made, as can be seen in Figure 3.2.

The second cluster is the work made towards Chapter 5. The single instance in the 18th month was a topic of discussion with no reference to this work.

Finally the word '*thesis*' was chosen to represent the compilation of this document. Three clusters can be seen, the first cluster of two instances are a passing reference to other work, while the second cluster represents discussion about the thesis writing of another PhD student who was observing the meetings. The final cluster in the 42nd month indicates the work of compiling this document.

Colophon

This thesis was made in LaTeX 2_ε using the "hepthesis" class [Buc].

List of figures

List of tables

Bibliography

[AAM07]   AKSIN Z., ARMONY M., MEHROTRA V.: The modern call center:
             A multi-disciplinary perspective on operations management
             research. *Production and operations management 16*, 6 (2007), 665–
             688.

[AAM*17]   ALHARBI N., ALHARBI M., MARTINEZ X., KRONE M., ROSE
             A. S., BAADEN M., LARAMEE R. S., CHAVENT M.: Molecular Vi-
             sualization of Computational Biology Data: A Survey of Surveys.
             In *EuroVis 2017 - Short Papers* (2017), Kozlikova B., Schreck T.,
             Wischgoll T., (Eds.), The Eurographics Association, pp. 133–137.

[AAMG12]   ALSALLAKH B., AIGNER W., MIKSCH S., GRÖLLER M. E.: Rein-
             venting the contingency wheel: Scalable visual analytics of large
             categorical data. *IEEE Transactions on Visualization and Computer
             Graphics 18*, 12 (2012), 2849–2858.

[AAMH13]   ALSALLAKH B., AIGNER W., MIKSCH S., HAUSER H.: Radial
             sets: Interactive visual analysis of large overlapping sets. *IEEE
             Transactions on Visualization and Computer Graphics 19*, 12 (2013),
             2496–2505.

[AAT15]   ANKRAH S., AL-TABBAA O.: Universities–industry collabora-
             tion: A systematic review. *Scandinavian Journal of Management 31*,
             3 (2015), 387–408.

[ACL17]   ALHARBI N., CHAVENT M., LARAMEE R. S.: Real-Time Ren-
             dering of Molecular Dynamics Simulation Data: A Tutorial. In
             Computer Graphics and Visual Computing (CGVC) (2017), Wan T. R.,
             Vidal F., (Eds.), The Eurographics Association.

[Ado18]   ADOBE SYSTEMS INCORPORATED: Adobe color cc. `https://`

color.adobe.com/, 2018. URL Accessed: 20 March 2020.

[AGMS11]   ALSALLAKH B., GRÖLLER E., MIKSCH S., SUNTINGER M.: Contingency Wheel: Visual Analysis of Large Contingency Tables. In *EuroVA 2011: International Workshop on Visual Analytics* (2011), Miksch S., Santucci G., (Eds.), The Eurographics Association, pp. 53–56.

[AHC08]   ADJEI C., HOLLAND-CUNZ N.: Monitoring and visualizing last.fm. *Diplom thesis, Fachhochschule Mainz* (2008). URL Accessed: 20 March 2020.

[AKCL19]   ALHARBI N., KRONE M., CHAVENT M., LARAMEE R. S.: Hybrid Visualization of Protein-Lipid and Protein-Protein Interaction. In *Eurographics Workshop on Visual Computing for Biology and Medicine* (2019), Kozlíková B., Linsen L., Vázquez P.-P., Lawonn K., Raidou R. G., (Eds.), The Eurographics Association.

[Ama18]   AMAZON.COM, INC.: Amazon.com. https://www.amazon.com, 2018. URL Accessed: 20 March 2020.

[AMST11]   AIGNER W., MIKSCH S., SCHUMANN H., TOMINSKI C.: *Visualization of Time-Oriented Data*. Springer Science & Business Media, Berlin, Germany, 2011.

[And57]   ANDERSON E.: A semigraphical method for the analysis of complex problems. *Proceedings of the National Academy of Sciences 43*, 10 (1957), 923–927.

[Ara18]   ARAGUES A.: *Visualizing Streaming Data: Interactive Analysis Beyond Static Limits*. O'Reilly Media, Sebastopol, CA, 2018.

[AS11]   ANGLE E., SHREINER D.: *Interactive Computer Graphics: A Topdown Approach with Shader-Based OpenGL*. Addison-Wesley, Boston, MA, 2011.

[Asp14]   ASPIN A.: *High Impact Data Visualization with Power View, Power Map, and Power BI*. Apress, Berkeley, CA, 2014.

[AW14]   ANDREWS K., WRIGHT B.: FluidDiagrams: Web-Based Information Visualisation using JavaScript and WebGL. In *EuroVis*

- Short Papers (2014), Elmqvist N., Hlawitschka M., Kennedy J., (Eds.), The Eurographics Association.

[Bak07] BAKER C. P.: Email map. `http://christopherbaker.net/projects/mymap/`, 2007. URL Accessed: 20 March 2020.

[BB04] BAUR M., BRANDES U.: Crossing reduction in circular layouts. In *International Workshop on Graph-Theoretic Concepts in Computer Science* (2004), Springer, Springer, pp. 332–343.

[BBB*13] BURTNER E. R., BOHN S. J., BLEEKER A. H., GOSINK L. J., GILLEN D. S.: Visual representation of data according to an abstraction hierarchy, February 2013. US Patent App. 13/211,587.

[BBS67] BERTIN J., BARBUT M., SERGE B.: *Sémiologie graphique: les diagrammes, les réseaux, les cartes.* Mouton/Gauthier-Villars, Paris, 1967.

[BD53] BARBEU-DUBOURG J.: Chronographie universelle. `http://dataphys.org/list/barbeu-dubourgs-machine-chronologique/`, 1753. URL Accessed: 20 March 2020.

[BECJ*07] BININDA-EMONDS O. R., CARDILLO M., JONES K. E., MACPHEE R. D., BECK R. M., GRENYER R., PRICE S. A., VOS R. A., GITTLEMAN J. L., PURVIS A.: The delayed rise of present-day mammals. *Nature 446*, 7135 (2007), 507.

[Ben19] BENOIT G.: *Introduction to Information Visualization: Transforming Data into Meaningful Information.* Rowman & Littlefield Publishers, Lanham, 2019.

[Ber83] BERTIN J.: *Semiology of graphics: Diagrams, networks, maps (WJ Berg, Trans.).* ESRI Press, Madison, WI, 1983.

[Ber06] BERKHIN P.: *A Survey of Clustering Data Mining Techniques.* Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 25–71.

[Ber16] BERINATO S.: *Good Charts.* Harvard Business Review Press, Boston, MA, 2016.

[BFW09] BACHTHALER S., FREY S., WEISKOPF D.: Poster: Cuda-accelerated continuous 2d scatterplots. In *IEEE Visualization*

Conference 2009 (2009).

[BGLL12]   BOHNACKER H., GROSS B., LAUB J., LAZZERONI C.: *Generative design: visualize, program, and create with processing*. Princeton Architectural Press, New York, NY, 2012.

[BGM*05]   BROWN L., GANS N., MANDELBAUM A., SAKOV A., SHEN H., ZELTYN S., ZHAO L.: Statistical analysis of a telephone call center: A queueing-science perspective. *Journal of the American statistical association 100*, 469 (2005), 36–50.

[BJ15]   BRATH R., JONKER D.: *Graph analysis and visualization: discovering business opportunity in linked data*. John Wiley & Sons, Indianapolis, IN, 2015.

[BKC*13]   BORGO R., KEHRER J., CHUNG D. H., MAGUIRE E., LARAMEE R. S., HAUSER H., WARD M., CHEN M.: Glyph-based visualization: Foundations, design guidelines, techniques and applications. In *Eurographics (STARs)* (2013), pp. 39–63.

[BKH*11]   BURCH M., KONEVTSOVA N., HEINRICH J., HOEFERLIN M., WEISKOPF D.: Evaluation of traditional, orthogonal, and radial tree diagrams by an eye tracking study. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2440–2448.

[BKW16]   BECK F., KOCH S., WEISKOPF D.: Visual analysis and dissemination of scientific literature collections with survis. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (2016), 180–189.

[BL09]   BAE J., LEE K.: Tagreel: A visualization of tag relations among user interests in the social tagging system. In *2009 Sixth International Conference on Computer Graphics, Imaging and Visualization* (2009), IEEE, IEEE, pp. 437–442.

[BOH11]   BOSTOCK M., OGIEVETSKY V., HEER J.: D^3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2301–2309.

[BP94]   BONACCORSI A., PICCALUGA A.: A theoretical framework for the evaluation of university-industry relationships. *R&D*

Management 24, 3 (1994), 229–247.

[Bra00]   BRADSKI G.: The opencv library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer 25*, 11 (2000), 120–123.

[Brea]   BREMER N.: Reviews of data visualization books - visual cinnamon. `https://www.visualcinnamon.com/resources/learning-data-visualization/books`. URL Accessed: 20 March 2020.

[Breb]   BREWER C. A.: ColorBrewer 2.0. `https://colorbrewer2.org/`. URL Accessed: 20 March 2020.

[Bro03]   BROWN L. D.: *Empirical analysis of call center traffic*. Tech. rep., Wharton School of the University of Pennsylvania, Philadelphia, PA, USA, 2003.

[Buc]   BUCKLEY A.: The hepthesis LaTeX class.

[BW08]   BACHTHALER S., WEISKOPF D.: Continuous scatterplots. *IEEE transactions on visualization and computer graphics 14*, 6 (2008), 1428–1435.

[BW09]   BACHTHALER S., WEISKOPF D.: Efficient and adaptive rendering of 2-d continuous scatterplots. In *Computer Graphics Forum* (2009), vol. 28, Wiley Online Library, pp. 743–750.

[BW14]   BURCH M., WEISKOPF D.: On the benefits and drawbacks of radial diagrams. In *Handbook of human centric visualization*. Springer, 2014, pp. 429–451.

[BYB*13]   BORKIN M. A., YEH C. S., BOYD M., MACKO P., GAJOS K. Z., SELTZER M., PFISTER H.: Evaluation of filesystem provenance visualization tools. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2476–2485.

[Cai12]   CAIRO A.: *The Functional Art: An introduction to information graphics and visualization*. New Riders, Berkeley, 2012.

[Cai16]   CAIRO A.: *The truthful art: Data, charts, and maps for communication*. New Riders, Berkeley, 2016.

[CC16]   CAO N., CUI W.: *Introduction to text visualization.* Springer, London, 2016.

[CCM*14]   CHEN H., CHEN W., MEI H., LIU Z., ZHOU K., CHEN W., GU W., MA K.-L.: Visual abstraction and exploration of multi-class scatterplots. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 1683–1692.

[CDJM14]   CHENG S., DE P., JIANG S. H.-C., MUELLER K.: Torusvis nd: unraveling high-dimensional torus networks for network traffic visualizations. In *Proceedings of the First Workshop on Visual Performance Analysis* (2014), IEEE Press, IEEE Press, pp. 9–16.

[CEH*08]   CHEN M., EBERT D., HAGEN H., LARAMEE R. S., VAN LIERE R., MA K.-L., RIBARSKY W., SCHEUERMANN G., SILVER D.: Data, information, and knowledge in visualization. *IEEE Computer Graphics and Applications 29*, 1 (2008), 12–19.

[CFV*16]   CHEN M., FEIXAS M., VIOLA I., BARDERA A., SHEN H.-W., SBERT M.: *Information Theory Tools for Visualization.* CRC Press, Boca Raton, FL, 2016.

[CFW17]   CAVA R., FREITAS C. M., WINCKLER M.: Clustervis: visualizing nodes attributes in multivariate graphs. In *Proceedings of the Symposium on Applied Computing* (New York, NY, USA, 2017), ACM, Association for Computing Machinery, pp. 174–179.

[Cha38]   CHAMBERS E.: *Cyclopaedia: Or an Universal Dictionary of Arts and Sciences.* Midwinter, London, 1738.

[Cha99]   CHAM J.: Graph - work output. `http://phdcomics.com/comics/archive.php?comicid=124`, 1999. URL Accessed: 20 March 2020.

[Cho19]   CHOU L.: What is data visualization? `https://towardsdatascience.com/9-data-visualization-tools-that-you-cannot-miss-in-2019-3ff23222a927`, 2019. URL Accessed: 20 March 2020.

[Cle85]   CLEVELAND W. S.: *The Elements of Graphing Data.* Wadsworth Advanced Books and Software, Monterey, CA, 1985.

[Cle93]   CLEVELAND W. S.: *Visualizing data*. Hobart Press, Summit New Jersey„ 1993.

[Cli18]   CLIFFORD T.: 5 inspirational chord diagrams: Visualizing data. `https://www.streetlightdata.com/chord-diagrams-visualizing-data/`, 2018. URL Accessed: 20 March 2020.

[CLNL87]   CARR D. B., LITTLEFIELD R. J., NICHOLSON W., LITTLEFIELD J.: Scatterplot matrix techniques for large N. *Journal of the American Statistical Association 82*, 398 (1987), 424–436.

[CLP*15]   CHUNG D. H., LEGG P. A., PARRY M. L., BOWN R., GRIFFITHS I. W., LARAMEE R. S., CHEN M.: Glyph sorting: Interactive visualization for multi-dimensional data. *Information Visualization 14*, 1 (2015), 76–90.

[CLS14]   CIUCCARELLI P., LUPI G., SIMEONE L.: *Visualizing the Data City: Social Media as a Source of Knowledge for Urban Planning and Management*. Springer, Cham, 2014.

[CM93]   CRAWFIS R. A., MAX N.: Texture splats for 3d scalar and vector field visualization. In *Proceedings Visualization'93* (1993), IEEE, pp. 261–266.

[CM07]   CORTEZ P., MORAIS A. D. J. R.: A Data Mining Approach to Predict Forest Fires using Meteorological Data. In *New Trends in Artificial Intelligence, 13th EPIA 2007 - Portuguese Conference on Artificial Intelligence* (Guimarães, Portugal, Dec. 2007), Neves J., Santos M. F., Machado J., (Eds.), APPIA, pp. 512–523.

[CMF*14]   CRNOVRSANIN T., MUELDER C. W., FARIS R., FELMLEE D., MA K.-L.: Visualization techniques for categorical analysis of social networks with multiple edge sets. *Social Networks 37* (2014), 56–64.

[CMS99]   CARD S. K., MACKINLAY J. D., SHNEIDERMAN B.: *Readings in information visualization: using vision to think*. Morgan Kaufmann, San Francisco, CA, 1999.

[Con18a]   CONTACTBABEL: *UK Contact Centres: 2018-2022 The State of the Industry & Technology Penetration*. Tech. rep., ContactBabel, 2018.

[Con18b]   CONTACTBABEL: *US Contact Centres: 2018-2022 The State of the Industry & Technology Penetration.* Tech. rep., ContactBabel, 2018.

[Con19]   CONTACTBABEL: *US Contact Centres: 2019-2023 The State of the Industry & Technology Penetration.* Tech. rep., ContactBabel, 2019.

[Cor18]   CORRELL M.: Ross-chernoff glyphs or: How do we kill bad ideas in visualization? In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2018), CHI EA '18, ACM, pp. alt05:1–alt05:10.

[Cot13]   COTGREAVE A.: 8 great books about data visualization. `https://www.tableau.com/about/blog/2013/7/list-books-about-data-visualisation-24182`, 2013. URL Accessed: 20 March 2020.

[CPG*16]   CHUNG D. H. S., PARRY M. L., GRIFFITHS I. W., LARAMEE R. S., BOWN R., LEGG P. A., CHEN M.: Knowledge-assisted ranking: A visual analytic application for sports event data. *IEEE Computer Graphics and Applications 36*, 3 (May 2016), 72–82.

[CT05]   COOK K. A., THOMAS J. J.: *Illuminating the path: The research and development agenda for visual analytics.* Tech. rep., Pacific Northwest National Laboratory (PNNL), Richland, WA, 2005.

[CZIM18]   CHENG S., ZHONG W., ISAACS K. E., MUELLER K.: Visualizing the topology and data traffic of multi-dimensional torus interconnect networks. *IEEE Access 6* (2018), 57191–57204.

[Dat]   DATA CYMRU: Infobase cymru. `www.infobasecymru.net`. URL Accessed: 20 March 2020.

[DBB10]   DIEHL S., BECK F., BURCH M.: Uncovering strengths and weaknesses of radial visualizations—an empirical approach. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 935–942.

[DBD18]   DIMARA E., BEZERIANOS A., DRAGICEVIC P.: Conceptual and methodological issues in evaluating multidimensional visualizations for decision support. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (Jan 2018), 749–759.

[Des37]   DESCARTES R.: *La géométrie*. Jan Maire, Leiden, 1637.

[Dim16]   DIMENSION DATA: *Global contact centre benchmarking report*. Tech. rep., Dimension Data, 2016.

[DKŽ12]   DZEMYDA G., KURASOVA O., ŽILINSKAS J.: *Multidimensional Data Visualization: Methods and Applications*, vol. 75. Springer Science & Business Media, New York/London, 2012.

[DLR09]   DRAPER G. M., LIVNAT Y., RIESENFELD R. F.: A survey of radial methods for information visualization. *IEEE Transactions on Visualization and Computer Graphics 15*, 5 (2009), 759–776.

[dLvW93]   DE LEEUW W. C., VAN WIJK J. J.: A probe for local flow field visualization. In *Proceedings of the 4th Conference on Visualization '93* (Washington, DC, USA, 1993), VIS '93, IEEE Computer Society, pp. 39–45.

[DMK05]   DYKES J., MACEACHREN A. M., KRAAK M.-J.: *Exploring geovisualization*. Elsevier, 2005.

[Dur04]   DURER A.: The fall of man, 1504.

[DWA10]   DANG T. N., WILKINSON L., ANAND A.: Stacking graphic elements to avoid over-plotting. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1044–1052.

[dWdVDvdWG18]   DE WIT-DE VRIES E., DOLFSMA W. A., VAN DER WINDT H. J., GERKEMA M.: Knowledge transfer in university–industry research partnerships: A review. *The Journal of Technology Transfer* (2018), 1–20.

[ECS14]   ETEMAD K., CARPENDALE S., SAMAVATI F.: Spirograph inspired visualization of ecological networks. In *Proceedings of the Workshop on Computational Aesthetics* (New York, NY, USA, 2014), ACM, Association for Computing Machinery, pp. 81–91.

[ED07]   ELLIS G., DIX A.: A taxonomy of clutter reduction for information visualisation. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (Nov 2007), 1216–1223.

[EDG*08]   ELMQVIST N., DO T.-N., GOODELL H., HENRY N., FEKETE

J.-D.: Zame: Interactive large-scale graph visualization. In *IEEE Pacific Visualization Symposium, 2008. PacificVIS'08.* (2008), IEEE, pp. 215–222.

[EGK*01] ELLSON J., GANSNER E., KOUTSOFIOS L., NORTH S. C., WOODHULL G.: Graphviz–open source graph drawing tools. In *International Symposium on Graph Drawing* (2001), Springer, pp. 483–484.

[EGU18] EGUK: CGVC 2018. `http://www.eguk.org.uk/CGVC2018/`, 2018. URL Accessed: 20 March 2020.

[EH91] EMBRECHTS P., HERZBERG A. M.: Variations of andrews' plots. *International Statistical Review/Revue Internationale de Statistique* (1991), 175–194.

[EHP*11] ERSOY O., HURTER C., PAULOVICH F., CANTAREIRO G., TELEA A.: Skeleton-based edge bundling for graph visualization. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2364–2373.

[End16] ENDERT A.: Semantic interaction for visual analytics: Inferring analytical reasoning for model steering. *Synthesis Lectures on Visualization 4*, 2 (2016), 1–99.

[EPS19a] EPSRC: EPSRC Grant - Bringing Healthcare Data to Life. `https://gtr.ukri.org/projects?ref=EP%2FS010238%2F1`, 2019. URL Accessed: 20 March 2020.

[EPS19b] EPSRC: Esprc. `https://epsrc.ukri.org/`, 2019. URL Accessed: 20 March 2020.

[EPS19c] EPSRC: Industrial case. `https://epsrc.ukri.org/skills/students/coll/icase/intro/`, 2019. URL Accessed: 20 March 2020.

[esr] ESRI: ArcGIS. `https://www.arcgis.com/`. URL Accessed: 20 March 2020.

[Eul41] EULER L.: Solutio problematis ad geometriam situs pertinentis. *Commentarii academiae scientiarum Petropolitanae 8* (1741), 128–140.

[Eur14] EUROGRAPHICS: Eurovis 2014. `http://eurovis.swansea.ac.`

uk/, 2014. URL Accessed: 20 March 2020.

[FD01]   FRIENDLY M., DENIS D. J.: Milestones in the history of thematic cartography, statistical graphics, and data visualization. *URL http://www. datavis. ca/milestones 32* (2001), 13.

[FD05]   FRIENDLY M., DENIS D.: The early origins and development of the scatterplot. *Journal of the History of the Behavioral Sciences 41*, 2 (2005), 103–130.

[Fei14]   FEINBERG J.: Wordle. `http://www.wordle.net/`, 2014. URL Accessed: 20 March 2020.

[Few06]   FEW S.: *Information Dashboard Design*. O'Reilly, Sebastopol, CA, 2006.

[Few12]   FEW S.: *Show Me the Numbers: Designing Tables and Graphs to Enlighten*. Analytics Press, Burlingame, CA, 2012.

[FGKR16]   FALK M., GROTTEL S., KRONE M., REINA G.: Interactive gpu-based visualization of large dynamic particle data. *Synthesis Lectures on Visualization 4*, 3 (2016), 1–121.

[FIBK17]   FUCHS J., ISENBERG P., BEZERIANOS A., KEIM D.: A systematic review of experimental studies on data glyphs. *IEEE Transactions on Visualization and Computer Graphics 23*, 7 (2017), 1863–1879.

[Fis36]   FISHER R. A.: The use of multiple measurements in taxonomic problems. *Annals of eugenics 7*, 2 (1936), 179–188.

[FJWS16]   FUCHS J., JÄCKLE D., WEILER N., SCHRECK T.: Leaf glyphs: Story telling and data analysis using environmental data glyph metaphors. In *Computer Vision, Imaging and Computer Graphics Theory and Applications* (Cham, 2016), Braz J., Pettré J., Richard P., Kerren A., Linsen L., Battiato S., Imai F., (Eds.), Springer International Publishing, pp. 123–143.

[FKH*00]   FEINBERG R. A., KIM I.-S., HOKAMA L., DE RUYTER K., KEEN C.: Operational determinants of caller satisfaction in the call center. *International Journal of Service Industry Management* (2000).

[FL03]   FERREIRA DE OLIVEIRA M. C., LEVKOWITZ H.: From visual data

exploration to visual data mining: a survey. *IEEE Transactions on Visualization and Computer Graphics 9*, 3 (July 2003), 378–394.

[FM18]   FISCHER D., MEYER M.: *Making Data Visual*. O'Reilly, Sebastopol, CA, 2018.

[FP02]   FEKETE J.-D., PLAISANT C.: Interactive information visualization of a million items. In *IEEE Symposium on Information Visualization, 2002. INFOVIS 2002.* (2002), IEEE, pp. 117–124.

[Fry07]   FRY B.: *Visualizing data: Exploring and explaining data with the processing environment*. O'Reilly Media, Inc., Sebastopol, CA, 2007.

[FS12]   FERSTER B., SHNEIDERMAN B.: *Interactive visualization: Insight through inquiry*. MIT Press, Cambridge, MA, 2012.

[Gab18]   GABRIELLE B. R.: *Storytelling with Graphs: A New Approach Beyond Data Visualization*. Insights Publishing, Kirkland, WA, 2018.

[GG89]   GATH I., GEVA A. B.: Unsupervised optimal fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence 11*, 7 (1989), 773–780.

[GK06]   GANSNER E. R., KOREN Y.: Improved circular layouts. In *International Symposium on Graph Drawing* (Berlin, Heidelberg, 2006), Springer, Springer-Verlag, pp. 386–398.

[GKM*14] GROTTEL S., KRONE M., MÜLLER C., REINA G., ERTL T.: Megamol – a prototyping framework for particle-based visualization. *IEEE Transactions on Visualization and Computer Graphics 21*, 2 (2014), 201–214.

[GLS17]   GRAMAZIO C. C., LAIDLAW D. H., SCHLOSS K. B.: Colorgorical: creating discriminable and preferable color palettes for information visualization. *IEEE Transactions on Visualization and Computer Graphics 23*, 1 (2017), 521–530.

[GLT10]   GURVICH I., LUEDTKE J., TEZCAN T.: Staffing call centers with uncertain demand forecasts: A chance-constrained optimization

approach. *Management Science 56*, 7 (2010), 1093–1115.

[Goo18]   GOOGLE LLC: Google scholar. `https://scholar.google.co.uk/`, 2018. URL Accessed: 20 March 2020.

[GOV19]   GOV.UK: Research and development (r&d) tax reliefs. `https://www.gov.uk/guidance/corporation-tax-research-and-development-rd-relief`, 2019. URL Accessed: 20 March 2020.

[GPO16]   GAROUSI V., PETERSEN K., OZKAN B.: Challenges and best practices in industry-academia collaborations in software engineering: A systematic literature review. *Information and Software Technology 79* (2016), 106–127.

[GRE09]   GROTTEL S., REINA G., ERTL T.: Optimized data transfer for time-dependent, gpu-based glyphs. In *2009 IEEE Pacific Visualization Symposium* (2009), IEEE, pp. 65–72.

[Gup15]   GUPTA T.: Word cloud generator. `http://learnin60seconds.com/`, 2015. URL Accessed: 20 March 2020.

[GZ11]   GOU L., ZHANG X. L.: Treenetviz: Revealing patterns of networks over tree structures. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2449–2458.

[Ham08]   HAM F. V.: *Interactive Visualization of Large Networks: Technical Challenges and Practical applications.* VDM Verlag, Saarbrücken, 2008.

[HB03]   HARROWER M., BREWER C. A.: Colorbrewer. org: an online tool for selecting colour schemes for maps. *The Cartographic Journal 40*, 1 (2003), 27–37.

[Hea18]   HEALY K.: *Data Visualization: A Practical Introduction.* Princeton University Press, 2018.

[HET12]   HURTER C., ERSOY O., TELEA A.: Graph bundling by kernel density estimation. *Computer Graphics Forum 31* (2012), 865–874.

[HHH16]   HOGAN T., HINRICHS U., HORNECKER E.: The elicitation interview technique: Capturing people's experiences of data rep-

resentations. *IEEE Transactions on Visualization and Computer Graphics 22*, 12 (Dec 2016), 2579–2593.

[HLK*11]   HEINRICH J., LUO Y., KIRKPATRICK A. E., ZHANG H., WEISKOPF D.: Evaluation of a bundling technique for parallel coordinates. *arXiv preprint arXiv:1109.6073* (2011).

[Hol06]   HOLTEN D.: Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 741–748.

[HSS15]   HADLAK S., SCHUMANN H., SCHULZ H.-J.: A survey of multifaceted graph visualization. In *EuroVis (STARs)* (2015), pp. 1–20.

[Hur15]   HURTER C.: Image-based visualization: Interactive multidimensional data exploration. *Synthesis Lectures on Visualization 3*, 2 (2015), 1–127.

[HYFC14]   HARRISON L., YANG F., FRANCONERI S., CHANG R.: Ranking visualizations of correlation using weber's law. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 1943–1952.

[HZ96]   HUI M. K., ZHOU L.: How does waiting duration information influence customers' reactions to waiting for services? *Journal of Applied Social Psychology 26*, 19 (1996), 1702–1717.

[Ibr18]   IBRAHIM R.: Managing queueing systems where capacity is random and customers are impatient. *Production and Operations Management 27*, 2 (2018), 234–250.

[ID90]   INSELBERG A., DIMSDALE B.: Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proceedings of the First IEEE Conference on Visualization: Visualization90* (1990), IEEE, pp. 361–378.

[IHK*17]   ISENBERG P., HEIMERL F., KOCH S., ISENBERG T., XU P., STOLPER C. D., SEDLMAIR M. M., CHEN J., MÖLLER T., STASKO J.: vispubdata.org: A Metadata Collection about IEEE Visualization (VIS) Publications. *IEEE Transactions on Visualization and Computer Graphics 23* (2017).

[IIC*13] ISENBERG T., ISENBERG P., CHEN J., SEDLMAIR M., MÖLLER T.: A systematic review on the practice of evaluating visualization. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2818–2827.

[Imh82] IMHOF E.: *Cartographic relief presentation*. Walter de Gruyter GmbH & Co KG, Berlin/Boston, 1982.

[Inn15] INNOVATE UK: Innovate UK Grant - Visualising Customer Journeys Through the Call Centre Landscape. `https://gtr.ukri.org/projects?ref=101939`, 2015. URL Accessed: 20 March 2020.

[Inn18] INNOVATE UK: Innovate UK Grant - Trade Finance Fraud Detection Project in Dual Use Goods with Machine Learning and Visual Analytics. `https://gtr.ukri.org/projects?ref=104413`, 2018. URL Accessed: 20 March 2020.

[Inn19] INNOVATE UK: Knowledge transfer partnership. `https://www.gov.uk/guidance/knowledge-transfer-partnerships-what-they-are-and-how-to-apply`, 2019. URL Accessed: 20 March 2020.

[Ins09] INSELBERG A.: *Parallel Coordinates: Visual Multidimensional Geometry and its Applications*. Springer, New York, NY, 2009.

[JAD11] JOUINI O., AKŞIN Z., DALLERY Y.: Call centers with delay information: Models and insights. *Manufacturing & Service Operations Management 13*, 4 (2011), 534–548.

[JAK*15] JOUINI O., AKŞIN O. Z., KARAESMEN F., AGUIR M. S., DALLERY Y.: Call center delay announcement using a newsvendor-like performance criterion. *Production and Operations Management 24*, 4 (2015), 587–604.

[JK01] JONGBLOED G., KOOLE G.: Managing uncertainty in call centres using poisson mixtures. *Applied Stochastic Models in Business and Industry 17*, 4 (2001), 307–318.

[JS91] JOHNSON B., SHNEIDERMAN B.: Tree-maps: A space-filling approach to the visualization of hierarchical information structures. In *Proceedings of the 2nd conference on Visualization'91* (1991), IEEE

Computer Society Press, pp. 284–291.

[KCA*16]   KO S., CHO I., AFZAL S., YAU C., CHAE J., MALIK A., BECK K., JANG Y., RIBARSKY W., EBERT D. S.: A survey on visual analysis approaches for financial data. In *Computer Graphics Forum* (2016), vol. 35, Wiley Online Library, pp. 599–617.

[KD12]   KRESSE W., DANKO D. M.: *Springer Handbook of Geographic Information*. Springer, Dordrecht, 2012.

[KES20]   KESS: Kess 2 knowledge economy skills scholarships. `http://kess2.ac.uk/`, 2020. URL Accessed: 20 March 2020.

[KHD*10]   KEIM D. A., HAO M. C., DAYAL U., JANETZKO H., BAK P.: Generalized scatter plots. *Information Visualization 9*, 4 (2010), 301–311.

[Khr]   KHRONOS GROUP: Opengl. `https://www.opengl.org/about/`. URL Accessed: 20 March 2020.

[Kin14]   KING R. S.: *Visual Storytelling with D3: An Introduction to Data Visualization in JavaScript*. Addison-Wesley Professional, Upper Saddle River, NJ, 2014.

[Kir12]   KIRK A.: *Data Visualization: a successful design process*. Packt Publishing Ltd, Los Angeles, CA, 2012.

[Kir16]   KIRK A.: *Data visualisation: a handbook for data driven design*. Sage, Los Angeles, CA, 2016.

[KK94]   KEIM D. A., KRIEGEL H.-P.: Visdb: Database exploration using multidimensional visualization. *IEEE Computer Graphics and Applications 14*, 5 (1994), 40–49.

[KK96]   KEIM D. A., KRIEGEL H. .: Visualization techniques for mining large databases: a comparison. *IEEE Transactions on Knowledge and Data Engineering 8*, 6 (Dec 1996), 923–938.

[KKEM10]   KEIM D., KOHLHAMMER J., ELLIS G., MANSMANN F.: Mastering the information age-solving problems with visual analytics. *ISBN-13 9783905673777* (2010).

[KKG*20]   KAMMER D., KECK M., GRÜNDER T., MAASCH A., THOM T., KLEINSTEUBER M., GROH R.: Glyphboard: Visual exploration of high-dimensional data combining glyphs with dimensionality reduction. *IEEE Transactions on Visualization and Computer Graphics* (2020).

[Kla08]   KLANTEN R.: *Data Flow: Visualising Information in Graphic Design*. Die Gestalten Verlag, Berlin, 2008.

[KLL03]   KATZ K., LARSON B., LARSON R.: Prescription for the waiting-in-line blues entertain, enlighten, and engage. *Oper Manag Crit Perspect Bus Manag 2* (2003), 160.

[Kna15]   KNAFLIC C. N.: *Storytelling with data: a data visualization guide for business professionals*. John Wiley & Sons, Hoboken, NJ, 2015.

[Koo13]   KOOLE G.: *Call center optimization*. MG books, Amsterdam, 2013.

[KP12]   KUHAR S., PODGORELEC V.: Ontology visualization for domain experts: A new solution. In *2012 16th International Conference on Information Visualisation* (2012), IEEE, IEEE, pp. 363–369.

[Kru13]   KRUM R.: *Cool infographics: Effective communication with data visualization and design*. John Wiley & Sons, Indianapolis, IN, 2013.

[KSB*09]   KRZYWINSKI M., SCHEIN J., BIROL I., CONNORS J., GASCOYNE R., HORSMAN D., JONES S. J., MARRA M. A.: Circos: an information aesthetic for comparative genomics. *Genome research 19*, 9 (2009), 1639–1645.

[KSS16]   KESSENICH J., SELLERS G. M., SHREINER D.: *OpenGL Programming Guide: The Official Guide to Learning OpenGL*. Addison-Wesley Professional, Boston, 2016.

[KT12]   KUMAR A., TELANG R.: Does the web reduce customer service cost? empirical evidence from a call center. *Information Systems Research 23*, 3-part-1 (2012), 721–737.

[Lar10a]   LARAMEE R.: The PhD in Visualization Starter Kit (PVSK). In *IEEE VisWeek 2010* (October 2010). Refereed Conference Tutorial

in Salt Lake City, Utah.

[Lar10b]   LARAMEE R.: The PhD in Visualization Starter Kit (PVSK). In *The International Conference on Information Visualization (IV) 2010* (July 2010). Invited Conference Tutorial at London South Bank University, London, UK (available online).

[Lar10c]   LARAMEE R. S.: *Bob's Minutes of Meeting Protocol: Incentive and a Description*. Tech. rep., Department of Computer Science, Swansea University, Swansea, UK, March 2010.

[Lar10d]   LARAMEE R. S.: The PhD in Visualization Starter Kit (PVSK). `http://cs.swan.ac.uk/~csbob/research/starterKit/`, 2010. URL Accessed: 20 March 2020.

[Lar14]   LARAMEE R. S.: The annual computer science undergraduate colloquium at gregynog. `http://cs.swan.ac.uk/~csbob/teaching/gregynog`, 2014. URL Accessed: 20 March 2020.

[Lar17a]   LARAMEE R. S.: Academic - industry gaps: Panel at the uncertainty quantification & management for industry event; youtube video. `https://www.youtube.com/watch?v=fCYhTX7Imc8`, 2017. URL Accessed: 20 March 2020.

[Lar17b]   LARAMEE, ROBERT S.: Bob's minutes of meeting protocol – youtube. `https://www.youtube.com/watch?v=CbaDy25oVmM`, 2017. URL Accessed: 20 March 2020.

[Lar18]   LARAMEE R. S.: Projects in data visualization. `http://cs.swan.ac.uk/~csbob/teaching/cs354-projectSpec/laramee18projectBrochure.pdf`, 2018. URL Accessed: 20 March 2020.

[Lar19a]   LARAMEE R. S.: The annual computer science project demonstration fair. `http://cs.swan.ac.uk/~csbob/teaching/projectFair/`, 2019. URL Accessed: 20 March 2020.

[Lar19b]   LARAMEE R. S.: The visible lunch. `http://cs.swan.ac.uk/~csbob/visibleLunch/`, 2019. URL Accessed: 20 March 2020.

[LHT17]   LHUILLIER A., HURTER C., TELEA A.: State of the art in edge

and trail bundling techniques. *Computer Graphics Forum 36*, 3 (2017), 619–645.

[Lim11]   LIMA M.: *Visual Complexity. Mapping Patterns of Information.* Princeton: Princeton Architectural Press, Princeton, NJ, 2011.

[Lim14]   LIMA M.: *The Book of Trees: Visualizing Branches of Knowledge.* Princeton Architectural Press, Princeton, NJ, 2014.

[Lim17]   LIMA M.: *The Book of Circles: Visualizing Spheres of Knowledge.* Princeton Architectural Press, Princeton, NJ, 2017.

[LJH03]   LARAMEE R. S., JOBARD B., HAUSER H.: Image space based visualization of unsteady flow on surfaces. In *IEEE Visualization, 2003. VIS 2003.* (2003), IEEE, pp. 131–138.

[LJH13]   LIU Z., JIANG B., HEER J.: immens: Real-time visual querying of big data. *Computer Graphics Forum 32*, 3pt4 (2013), 421–430.

[LKK17]   LEE S., KIM S., KWON B. C.: Vlat: Development of a visualization literacy assessment test. *IEEE Transactions on Visualization and Computer Graphics 23*, 1 (Jan 2017), 551–560.

[LL15]   LO R. C., LO W. C.: *OpenGL Data Visualization Cookbook.* Packt Publishing Ltd, Birmingham, UK, 2015.

[LM10]   LAM H., MUNZNER T.: A guide to visual multi-level interface design from synthesis of empirical study evidence. *Synthesis Lectures on Visualization 1*, 1 (2010), 1–117.

[LMW*16]   LIU S., MALJOVEC D., WANG B., BREMER P.-T., PASCUCCI V.: Visualizing high-dimensional data: Advances in the past decade. *IEEE Transactions on Visualization and Computer Graphics 23*, 3 (2016), 1249–1268.

[Lra02]   LRAMEE R. S.: Interactive 3d flow visualization using a stream-runner. In *CHI'02 Extended Abstracts on Human Factors in Computing Systems* (2002), pp. 804–805.

[LvJH04]   LARAMEE R. S., VAN WIJK J. J., JOBARD B., HAUSER H.: Isa and ibfvs: image space-based visualization of flow on surfaces. *IEEE Transactions on Visualization and Computer Graphics 10*, 6

(Nov 2004), 637–648.

[Mac11]   MACIEJEWSKI R.: Data representations, transformations, and statistics for visual reasoning. *Synthesis Lectures on Visualization 2*, 1 (2011), 1–85.

[Mäk88]   MÄKINEN E.: On circular layouts. *International Journal of Computer Mathematics 24*, 1 (1988), 29–37.

[Mar09]   MARTY R.: *Applied security visualization.* Addison-Wesley, Upper Saddle River, NJ, 2009.

[McC96]   MCCONNELL S.: *Rapid development: taming wild software schedules.* Microsoft Press, Redmond, WA, 1996.

[McC09]   MCCANDLESS D.: *Information is Beautiful.* Collins, London, 2009.

[McC14]   MCCANDLESS D.: *Knowledge is Beautiful.* HarperCollins, London, 2014.

[ME09]   MCDONNEL B., ELMQVIST N.: Towards utilizing GPUs in information visualization: A model and implementation of image-space operations. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1105–1112.

[Mee15]   MEEKS E.: *D3. js in Action.* Manning Publ., Shelter Island, 2015.

[Mei13]   MEIRELLES I.: *Design for Information: An Introduction to the Histories, Theories, and Best Practices behind Effective Information Visualizations.* Rockport Publishers, Beverly, MA, 2013.

[MFM18]   MASCARENHAS C., FERREIRA J. J., MARQUES C.: University–industry cooperation: A systematic literature review and research agenda. *Science and Public Policy 45*, 5 (2018), 708–718.

[MG06]   MEHROTRA V., GROSSMAN T.: *New processes enhance crossfunctional collaboration and reduce call center costs.* Tech. rep., Working Paper, Department of Decision Sciences, San Francisco State University, 2006.

[MG13]   MAYORGA A., GLEICHER M.: Splatterplots: Overcoming overdraw in scatter plots. *IEEE Transactions on Visualization and Com-*

puter Graphics 19, 9 (2013), 1526–1538.

[MGMG11]   MUNSHI A., GASTER B., MATTSON T. G., GINSBURG D.: *OpenCL programming guide.* Pearson Education, 2011.

[MGR99]   MUNZNER T., GUIMBRETIERE F., ROBERTSON G.: Constellation: A visualization tool for linguistic queries from mindnet. In *Information Visualization, 1999.(Info Vis' 99) Proceedings. 1999 IEEE Symposium on* (1999), IEEE, pp. 132–135.

[Mic13]   MICROSOFT CORPORATION: Kinect - Windows app development. `https://developer.microsoft.com/en-us/windows/kinect`, 2013. URL Accessed: 20 March 2020.

[Mik19]   MIKHAILOV A.: Turbo, an improved rainbow colormap for visualization. `https://ai.googleblog.com/2019/08/turbo-improved-rainbow-colormap-for.html`, 2019. URL Accessed: 20 March 2020.

[Min69]   MINARD C. J.: *Carte Figurative des pertes successives en hommes de l'armée qu'Annibal conduisit d'Espagne en Italie en traversant les Gaules (selon Polybe).* Regnier et Dourdet, Paris, 1869.

[MKRE16]   MWALONGO F., KRONE M., REINA G., ERTL T.: State-of-the-art report in web-based visualization. *Computer Graphics Forum 35*, 3 (2016), 553–575.

[ML17]   MCNABB L., LARAMEE R. S.: Survey of surveys (sos) - mapping the landscape of survey papers in information visualization. *Computer Graphics Forum 36*, 3 (2017), 589–617.

[MM08]   MCDONNELL K. T., MUELLER K.: Illustrative parallel coordinates. *Computer Graphics Forum 27*, 3 (2008), 1031–1038.

[MMP09]   MEYER M., MUNZNER T., PFISTER H.: Mizbee: a multiscale synteny browser. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 897–904.

[Mor16]   MORELAND K.: Why we use bad color maps and what you can do about it. *Electronic Imaging 2016*, 16 (2016), 1–6.

[MOS10]   MEHROTRA V., OZLÜK O., SALTZMAN R.: Intelligent proce-

dures for intra-day updating of call center agent schedules. *Production and operations management 19*, 3 (2010), 353–367.

[Mov13]   MOVANIA M. M.: *OpenGL Development Cookbook*. Packt Publishing Ltd, Birmingham, UK, 2013.

[MP18]   MILLER-PRICE J.: *GREAT STORYTELLING IN BUSINESS: How to present your data as a compelling story*. Independently published, 2018.

[MPOW17]   MICALLEF L., PALMAS G., OULASVIRTA A., WEINKAUF T.: Towards perceptual optimization of the visual design of scatterplots. *IEEE transactions on visualization and computer graphics 23*, 6 (2017), 1588–1599.

[MRZH14]   MÜLLER H., REIHS R., ZATLOUKAL K., HOLZINGER A.: Analysis of biomedical data with multilevel glyphs. *BMC Bioinformatics 15*, 6 (May 2014), S5.

[MSD*08]   MEYER-SPRADOW J., STEGGER L., DÃŰRING C., ROPINSKI T., HINRICHS K.: Glyph-based spect visualization for the diagnosis of coronary artery disease. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (Nov 2008), 1499–1506.

[Mun08]   MUNZNER T.: Process and pitfalls in writing information visualization research papers. In *Information visualization*. Springer, 2008, pp. 134–153.

[Mun09a]   MUNSHI A.: The opencl specification. In *Hot Chips 21 Symposium (HCS), 2009 IEEE* (2009), IEEE, pp. 1–314.

[Mun09b]   MUNZNER T.: A nested model for visualization design and validation. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 921–928.

[Mun14]   MUNZNER T.: *Visualization Analysis and Design*. CRC Press, Boca Raton, FL, 2014.

[Mur13]   MURRAY S.: *Interactive Data Visualization for the Web*. O'Reilly Media, Inc., Sebastopol, CA, 2013.

[MV10]   MCNAUGHTON S., VELASCO S.: Fifty years of space exploration.

National Geopraphic (2010).

[NAL14]   NICHOLAS M., ARCHAMBAULT D., LARAMEE R. S.: Interactive Visualisation of Automotive Warranty Data Using Novel Extensions of Chord Diagrams. In *EuroVis - Short Papers* (2014), Elmqvist N., Hlawitschka M., Kennedy J., (Eds.), The Eurographics Association, pp. 115–119.

[NEHH10]   NGUYEN Q., EADES P., HONG S.-H., HUANG W.: Large crossing angles in circular layouts. In *International Symposium on Graph Drawing* (2010), Springer, Springer, pp. 397–399.

[Nel14]   NELLI F.: *Create Web Charts with D3.* Apress, Berkeley, CA, 2014.

[Nig58]   NIGHTINGALE F.: *Notes on Matters Affecting the Health, Efficiency and Hospital Administration of the British Army.* Harrison & Sons, London, 1858.

[NLT20]   NLTK PROJECT: Natural language toolkit. `https://www.nltk.org/`, 2020. URL Accessed: 20 March 2020.

[NN17]   NYHOFF J., NYHOFF L.: *Processing: An Introduction to Programming.* Chapman and Hall, Boca Raton, FL, 2017.

[NRM55]   NORTHWAY M. L., ROOKS M. M., MORENO J.: Creativity and sociometric status in children. *Sociometry 18*, 4 (1955), 194–201.

[Osa02]   OSAWA N.: Visualization of inheritance relationships using glyphs. *IEICE Transactions on Information and Systems 85*, 1 (January 2002), 275–282.

[Osa05]   OSAWA N.: Application of jigsaw-puzzle-like 3d glyph to visualizing grammatical constraints. In *2005 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'05)* (Sep. 2005), pp. 293–295.

[Pal09]   PALEY W.: Textarc: Alice's adventures in wonderland. `http://www.textarc.org/Alice.html`, 2009. URL Accessed: 20 March 2020.

[Pan18]   PANTONE INC.: Pantone studio. `https://www.pantone.com/studio`, 2018. URL Accessed: 20 March 2020.

[Pei14]   PEIXOTO T. P.: Hierarchical block structures and high-resolution model selection in large networks. *Physical Review X 4*, 1 (2014), 011047.

[PGL*11]   PENG Z., GRUNDY E., LARAMEE R. S., CHEN G., CROFT N.: Mesh-driven vector field clustering and visualization: An image-based approach. *IEEE Transactions on Visualization and Computer Graphics 18*, 2 (2011), 283–298.

[PK18]   PAPP G., KUNKLI R.: Tabularvis-a circos-inspired interactive web client based tool for improving the clarity of tabular data visualization. In *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 2: IVAPP,* (2018), INSTICC, SciTePress, pp. 120–131.

[Pla86]   PLAYFAIR W.: *The commercial and political atlas and statistical breviary.* J. Debrett, Piccadilly, 1786.

[PLCZ09]   PENG Z., LARAMEE R. S., CHEN G., ZHANG E.: Glyph and streamline placement algorithms for cfd simulation data. In *NAFEMS World Congress Conference Proceedings (NWC)(June 2009), NAFEMS–The International Association for the Engineering Analysis Community. forthcoming* (2009).

[PLD*08]   PENG Z., LARAMEE R. S., DEUSSEN O., KEIM D., SAUPE D.: Vector glyphs for surfaces: A fast and simple glyph placement algorithm for adaptive resolution meshes. In *Vision, Modeling, and Visualization* (2008), pp. 61–70.

[PPvA*09]   PEETERS T. H., PRCKOVSKA V., VAN ALMSICK M., VILANOVA A., TER HAAR ROMENY B. M.: Fast and sleek glyph rendering for interactive hardi data exploration. In *2009 IEEE Pacific Visualization Symposium* (2009), IEEE, pp. 153–160.

[PTM*13]   PERKMANN M., TARTARI V., MCKELVEY M., AUTIO E., BROSTRÖM A., D'ESTE P., FINI R., GEUNA A., GRIMALDI R., HUGHES A., ET AL.: Academic engagement and commercialisation: A review of the literature on university–industry relations. *Research policy 42*, 2 (2013), 423–442.

[QGI]   QGIS DEVELOPMENT TEAM: QGIS. `https://www.qgis.org/`.
             URL Accessed: 20 March 2020.

[RB10]   RENSINK R. A., BALDRIDGE G.: The perception of correlation
             in scatterplots. In *Computer Graphics Forum* (2010), vol. 29, Wiley
             Online Library, pp. 1203–1210.

[RD]   RAMOS E., DONOHO D.: Cars Dataset. `http://davis.wpi.edu/`
             `xmdv/datasets/cars.html`. URL Accessed: 20 March 2020.

[RE05]   REINA G., ERTL T.: Hardware-accelerated glyphs for mono- and
             dipoles in molecular dynamics visualization. In *Proceedings of the
             Seventh Joint Eurographics / IEEE VGTC Conference on Visualization*
             (Goslar, DEU, 2005), Eurographics Association, pp. 177–182.

[Reea]   REES D.: Vimeo AgentVis Demonstration. `https://vimeo.com/`
             `325757055`. URL Accessed: 20 March 2020.

[Reeb]   REES D.:   YouTube AgentVis Demonstration.   `https://www.`
             `youtube.com/watch?v=hsD21tHybts`. URL Accessed: 20 March
             2020.

[Ree18]   REES D.:   Feature-rich, gpu-accelerated scatterplots for mil-
             lions of call events – demonstration video. `https://vimeo.com/`
             `305933032`, 2018. URL Accessed: 20 March 2020.

[Ree20a]   REES D.: Chord diagram interactions – vimeo. `https://vimeo.`
             `com/445249684`, 2020. URL Accessed: 10 August 2020.

[Ree20b]   REES D.: glyphrenderer – github. `https://github.com/dylan-`
             `rees/glyphRenderer`, 2020. URL Accessed: 20 March 2020.

[Ren18]   RENDGEN S.: *The Minard System: The Complete Statistical Graphics
             of Charles-Joseph Minard*. Princeton Architectural Press, New York,
             NY, 2018.

[RF07]   REAS C., FRY B.: *Processing: a programming handbook for visual
             designers and artists*. Mit Press, Cambridge, MA, 2007.

[RF15]   REAS C., FRY B.: *Getting Started with Processing: A Hands-On
             Introduction to Making Interactive Graphics*. Maker Media, Inc.,
             San Francisco, CA, 2015.

[RHR17]   ROBERTS J. C., HEADLEAND C. J., RITSOS P. D.: *Five Design-Sheets: Creative Design and Sketching for Computing and Visualisation*. Springer, Cham, 2017.

[Rhy16]   RHYNE T.-M.: *Applying color theory to digital media and visualization*. CRC Press, Boca Raton, FL, 2016.

[RL18]   ROBERTS R., LARAMEE R.: Visualising business data: A survey. *Information 9*, 11 (2018), 285.

[RL19]   REES D., LARAMEE R. S.: A survey of information visualization books. *Computer Graphics Forum 38*, 1 (2019), 610–646.

[RL20]   REES D., LARAMEE R. S.: An open-source, hardware accelerated industry prototype for multidimensional glyphs. In *Proceedings of the International Conference on Computer Graphics, Visualization, Computer Vision and Image Processing 2020* (2020), iadias, pp. 217–221.

[RLB18a]   REN D., LEE B., BREHMER M.: Charticulator: Interactive construction of bespoke chart layouts. *IEEE Transactions on Visualization and Computer Graphics 25*, 1 (2018), 789–799.

[RLB*18b]   ROBERTS R. C., LARAMEE R. S., BROOKES P., SMITH G. A., D'CRUZE T., ROACH M. J.: A tale of two visions-exploring the dichotomy of interest between academia and industry in visualisation. In *VISIGRAPP (3: IVAPP)* (2018), pp. 319–326.

[RLB*20]   REES D., LARAMEE R. S., BROOKES P., D'CRUZE T., SMITH G. A., MIAH A.: Agentvis: Visual analysis of agent behavior withhierarchical glyphs. *IEEE Transactions on Visualization and Computer Graphics* (2020), 1–1.

[RLBD20]   REES D., LARAMEE R. S., BROOKES P., D'CRUZE T.: Interaction techniques for chord diagrams. In *The 24th International Conference on Information Visualization (IV)* (2020).

[RLN*17]   REES D., LARAMEE R. S., NGUYEN D., ZHANG L., CHEN G., YEH H., ZHANG E.: A Stream Ribbon Seeding Strategy. In *EuroVis 2017 - Short Papers* (2017), Kozlikova B., Schreck T., Wischgoll T., (Eds.), The Eurographics Association.

[RLS*19]   ROBERTS R. C., LARAMEE R. S., SMITH G. A., BROOKES P.,
                D'CRUZE T.:   Smart brushing for parallel coordinates.   *IEEE
                Transactions on Visualization and Computer Graphics 25*, 3 (2019),
                1575–1590.

[RMA*10]   RIND A., MIKSCH S., AIGNER W., TURIC T., POHL M.: Visu-
                explore: gaining new medical insights from visual exploration.
                In *Proc. Int. Workshop on Interactive Systems in Healthcare (WISH@
                CHI2010)* (2010), pp. 149–152.

[Rob19]   ROBERTS R.: *Visualising the Customer Journey Through the Call
                Centre Landscape.* PhD thesis, Swansea University, 2019.

[ROP11]   ROPINSKI T., OELTZE S., PREIM B.:   Survey of glyph-based
                visualization techniques for spatial multivariate medical data.
                Computers & Graphics 35, 2 (2011), 392–401.

[RRL*18a]   REES D., ROBERTS R. C., LARAMEE R. S., BROOKES P., D'CRUZE
                T., SMITH G. A.: GPU-Assisted Scatterplots for Millions of Call
                Events. In *Computer Graphics and Visual Computing (CGVC)* (2018),
                Tam G. K. L., Vidal F., (Eds.), The Eurographics Association.

[RRL*18b]   ROBERTS R. C., REES D., LARAMEE R. S., BROOKES P., SMITH
                G. A.: RiverState: A Visual Metaphor Representing Millions
                of Time-Oriented State Transitions. In *Computer Graphics and
                Visual Computing (CGVC)* (2018), Tam G. K. L., Vidal F., (Eds.),
                The Eurographics Association, pp. 81–89.

[RRL*19]   REES D., ROBERTS R. C., LARAMEE R. S., BROOKES P., D'CRUZE
                T., SMITH G. A.:   Feature-rich, GPU-assisted scatterplots for
                millions of call events. *Computers 8*, 1 (2019), 1–21.

[RSMS*07]   ROPINSKI T., SPECHT M., MEYER-SPRADOW J., HINRICHS K.,
                PREIM B.: Surface glyphs for visualizing multimodal volume
                data. In *VMV* (01 2007), pp. 3–12.

[RT16]   RININSLAND Æ., TELLER S.: *Learning D3.js Data Visualization.*
                Packt Publishing Ltd, Birmingham, UK, 2016.

[RTL*16]   ROBERTS R., TONG C., LARAMEE R., SMITH G. A., BROOKES
                P., D'CRUZE T.: Interactive analytical treemaps for visualisa-

tion of call centre data. In *Proceedings of the Conference on Smart Tools and Applications in Computer Graphics* (2016), Eurographics Association, pp. 109–117.

[RWL20] REES D., WANG Q., LARAMEE R. S.: The industry engagement ladder. *Journal of Industry – University Collaboration* (2020).

[Sam06] SAMET H.: *Foundations of multidimensional and metric data structures*. Morgan Kaufmann, San Francisco, CA, 2006.

[San] SANDIA NATIONAL LABORATORIES: Paraview. `https://www.paraview.org/`. URL Accessed: 20 March 2020.

[Sca11] SCARPINO M.: *OpenCL in action: how to accelerate graphics and computations*. hgpu. org, 2011.

[Sch17] SCHOLTZ J.: User-centered evaluation of visual analytics. *Synthesis Lectures on Visualization 5*, 1 (2017), i–71.

[SEP*15] SHI J., ERDEM E., PENG Y., WOODBRIDGE P., MASEK C.: Performance analysis and improvement of a typical telephone response system of va hospitals: A discrete event simulation study. *International Journal of Operations & Production Management 35*, 8 (2015), 1098–1124.

[SG18] SARIKAYA A., GLEICHER M.: Scatterplots: Tasks, data, and designs. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (2018), 402–412.

[SGC*15] SARIKAYA A., GLEICHER M., CHANG R., SCHEIDEGGER C., FISHER D., HEER J.: Using webgl as an interactive visualization medium: Our experience developing splatterjs. In *Proceedings of the Data Systems for Interactive Analysis Workshop (Oct. 2015), DSIA* (2015), vol. 15.

[SGK*12] SCHULZE T., GESSLER A., KULLING K., NADLINGER D., KLEIN J., SIBLY M., GUBISCH M.: Open asset import library (assimp). *Computer Software, URL: https://github.com/assimp/assimp* (2012).

[SH14] SWIRES-HENNESSY E.: *Presenting Data: How to Communicate Your Message Effectively*. John Wiley & Sons, Chichester, 2014.

[Sha03]   SHARP D.: *Call center operation: design, operation, and maintenance.* Elsevier, 2003.

[Shi09]   SHIFFMAN D.: *Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction.* Morgan Kaufmann, Amsterdam, 2009.

[Shn96]   SHNEIDERMAN B.: The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on* (1996), IEEE, pp. 336–343.

[Shn03]   SHNEIDERMAN B.: The eyes have it: A task by data type taxonomy for information visualizations. In *The Craft of Information Visualization*, Bederson B. B., Shneiderman B., (Eds.), Interactive Technologies. Morgan Kaufmann, San Francisco, 2003, pp. 364 – 371.

[Shn16]   SHNEIDERMAN B.: *The new ABCs of research: Achieving breakthrough collaborations.* Oxford University Press, 2016.

[Shn18]   SHNEIDERMAN B.: Twin-win model: A human-centered approach to research success. *Proceedings of the National Academy of Sciences 115*, 50 (2018), 12590–12594.

[SI10]   STEELE J., ILIINSKY N.: *Beautiful Visualization: Looking at Data through the Eyes of Experts.* O'Reilly Media, Inc., Beijing, 2010.

[Sii00]   SIIRTOLA H.: Direct manipulation of parallel coordinates. In *2000 IEEE Conference on Information Visualization. An International Conference on Computer Visualization and Graphics* (2000), IEEE, pp. 373–378.

[Sim14]   SIMON P.: *The visual organization: data visualization, Big Data, and the quest for better decisions.* John Wiley & Sons, Hoboken, NJ, 2014.

[SK07]   STEINWEBER P., KOLLER A.: Similar diversity, 2007.

[SKE15]   SARIGIANNIDIS P., KARAPISTOLI E., ECONOMIDES A. A.: Visiot: A threat visualisation tool for iot systems security. In *2015 IEEE International Conference on Communication Workshop (ICCW)*

(2015), IEEE, IEEE, pp. 2633–2638.

[SLBC03]   SWAYNE D. F., LANG D. T., BUJA A., COOK D.: Ggobi: evolving from xgobi into an extensible framework for interactive data visualization. *Computational Statistics & Data Analysis 43*, 4 (2003), 423–444.

[SMM12]   SEDLMAIR M., MEYER M., MUNZNER T.: Design study methodology: Reflections from the trenches and the stacks. *IEEE Transactions on Visualization and Computer Graphics 18*, 12 (2012), 2431–2440.

[Sno55]   SNOW J.: *On the mode of communication of cholera*. John Churchill, London, 1855.

[SP16]   SEDIG K., PARSONS P.: Design of visualizations for human-information interaction: A pattern-based framework. *Synthesis Lectures on Visualization 4*, 1 (2016), 1–185.

[Spe14]   SPENCE R.: *Information visualization: An introduction*. Springer, Cham, 2014.

[Spr18]   SPRINGER, SCIENCE+BUSINESS MEDIA: Springer - international publisher science, technology, medicine. `http://www.springer.com/`, 2018. URL Accessed: 20 March 2020.

[SS02]   SEO J., SHNEIDERMAN B.: Interactively exploring hierarchical clustering results [gene identification]. *Computer 35*, 7 (2002), 80–86.

[SSB15]   SAILEM H. Z., SERO J. E., BAKAL C.: Visualizing cellular imaging data using phenoplot. *Nature Communications 6* (January 2015), 5825.

[ST99]   SIX J. M., TOLLIS I. G.: Circular drawings of biconnected graphs. In *Workshop on Algorithm Engineering and Experimentation* (1999), Springer, Springer, pp. 57–73.

[Sto11]   STONE J.: X-men family tree. `https://www.joe-stone.co.uk/familytrees`, 2011. URL Accessed: 20 March 2020.

[Sto16]   STONE M.: How we designed the new color palettes in

Tableau 10. `https://www.tableau.com/about/blog/2016/7/colors-upgrade-tableau-10-56782`, 2016. URL Accessed: 20 March 2020.

[SZ00] STASKO J., ZHANG E.: Focus+ context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. In *IEEE Symposium on Information Visualization 2000. INFOVIS 2000. Proceedings* (2000), IEEE, IEEE, pp. 57–65.

[Tab] TABLEAU SOFTWARE: Tableau. `https://www.tableau.com/`. URL Accessed: 20 March 2020.

[Tay94] TAYLOR S.: Waiting for service: the relationship between delays and evaluations of service. *Journal of marketing 58*, 2 (1994), 56–69.

[TCl15] THIONGANE M., CHAN W., L'ECUYER P.: Waiting time predictors for multi-skill call centers. In *2015 Winter Simulation Conference (WSC)* (2015), IEEE, pp. 3073–3084.

[TE10] TELEA A., ERSOY O.: Image-based edge bundles: Simplified visualization of large graphs. *Computer Graphics Forum 29*, 3 (2010), 843–852.

[Tel14] TELEA A. C.: *Data visualization: principles and practice*. CRC Press, Boca Raton, FL, 2014.

[TGK*17] TOMINSKI C., GLADISCH S., KISTER U., DACHSELT R., SCHUMANN H.: Interactive lenses for visualization: An extended survey. *Computer Graphics Forum 36*, 6 (2017), 173–200.

[The] THE QT COMPANY: Qt application framework. `http://www.qt.io/`. URL Accessed: 20 March 2020.

[The16] THE KHRONOS GROUP INC.: Vulkan overview. `https://www.khronos.org/vulkan/`, 2016. URL Accessed: 20 March 2020.

[The20] THE KHRONOS GROUP INC: 5 inspirational chord diagrams: Visualizing data. `https://www.khronos.org/registry/OpenGL/index_gl.php`, 2020. URL Accessed: 20 March 2020.

[TML*17] TONG C., MCNABB L., LARAMEE R. S., LYONS J., WALTERS A., BERRIDGE D., THAYER D.: Time-oriented Cartographic

Treemaps for Visualization of Public Healthcare Data. In *Computer Graphics and Visual Computing (CGVC)* (2017), Wan T. R., Vidal F., (Eds.), The Eurographics Association.

[TML18]   TONG C., MCNABB L., LARAMEE R. S.: Cartograms with topological features. In *Proceedings of the Conference on Computer Graphics & Visual Computing* (2018), Eurographics Association, pp. 127–134.

[Tom15]   TOMINSKI C.: Interaction for visualization. *Synthesis Lectures on Visualization 3*, 1 (2015), 1–107.

[Tre99]   TREINISH L. A.: Task-specific visualization design. *IEEE Computer Graphics and Applications 19*, 5 (Sept. 1999), 72–77.

[TRL*17]   TONG C., ROBERTS R., LARAMEE R. S., BERRIDGE D., THAYER D.: Cartographic treemaps for visualization of public healthcare data. *Computer Graphics and Visual Computing (CGVC)* (2017).

[TS06]   TELEA A. C., STRZODKA R.: Multiscale image based flow visualization. In *Visualization and Data Analysis 2006* (2006), vol. 6060, International Society for Optics and Photonics, p. 606001.

[Tuf83]   TUFTE E. R.: *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, 1983.

[Tuf90]   TUFTE E. R.: *Envisioning Information*. Graphics Press, Cheshire, CT, 1990.

[Tuf97]   TUFTE E. R.: *Visual Explanations*. Graphics Press, Cheshire, CT, 1997.

[Tuf06]   TUFTE E. R.: *Beautiful Evidence*. Graphics Press, Cheshire, CT, 2006.

[UK 20]   UK RESEARCH AND INNOVATION: Innovate UK. `https://www.gov.uk/government/organisations/innovate-uk`, 2020. URL Accessed: 20 March 2020.

[Uni]   UNIVERSITY OF VIRGINIA: Visualeyes. `http://www.viseyes.org/viseyes.htm`. URL Accessed: 20 March 2020.

[VEW09]   VAN ECK N., WALTMAN L.:   Software survey: Vosviewer, a
              computer program for bibliometric mapping. *Scientometrics 84*,
              2 (2009), 523–538.

[VFA13]   VATHY-FOGARASSY Á., ABONYI J.: *Graph-based clustering and
              data visualization algorithms*. Springer, London, 2013.

[VHVW03]   VAN HAM F., VAN WIJK J. J.: Beamtrees: Compact visualization
              of large hierarchies. *Information Visualization 2*, 1 (2003), 31–39.

[VSBCR02]   VAN SOLINGEN R., BASILI V., CALDIERA G., ROMBACH H. D.:
              Goal question metric (GQM) approach. *Encyclopedia of software
              engineering* (2002).

[VW02]   VAN WIJK J. J.: Image based flow visualization. In *Proceedings
              of the 29th annual conference on Computer graphics and interactive
              techniques* (2002), pp. 745–754.

[vW03]   VAN WIJK J. J.:   Image based flow visualization for curved
              surfaces. In *IEEE Visualization, 2003. VIS 2003.* (2003), IEEE,
              pp. 123–130.

[VWVH*07]   VIEGAS F. B., WATTENBERG M., VAN HAM F., KRISS J., MCK-
              EON M.: Manyeyes: a site for visualization at internet scale.
              IEEE Transactions on Visualization and Computer Graphics 13, 6
              (2007).

[WAG05a]   WILKINSON L., ANAND A., GROSSMAN R.: Graph-theoretic
              scagnostics. In *Proceedings of the Proceedings of the 2005 IEEE
              Symposium on Information Visualization* (Washington, DC, USA,
              2005), INFOVIS '05, IEEE Computer Society, pp. 21–.

[WAG05b]   WILKINSON L., ANAND A., GROSSMAN R.: Graph-theoretic
              scagnostics. In *IEEE Symposium on Information Visualization, 2005.
              INFOVIS 2005.* (2005), IEEE, pp. 157–164.

[Wal74]   WALKER F. A.: Chart showing the principal constituent elements
              of the population of each state. In *Statistical atlas of the United
              States based on the results of the ninth census 1870 with contributions
              from many eminent men of science and several departments of the
              government* (New York, 1874), J. Bien.

[War02]   WARD M. O.: A taxonomy of glyph placement strategies for
                 multidimensional data visualization. *Information Visualization 1*,
                 3-4 (2002), 194–210.

[War10]   WARE C.: *Visual Thinking: For design*. Morgan Kaufmann, Ams-
                 terdam, 2010.

[War12]   WARE C.: *Information visualization: perception for design*. Elsevier,
                 Amsterdam, 2012.

[Wat98]   WATTENBERG M.: Map of the market. `http://www.bewitched.`
                 `com/marketmap.html`, 1998. URL Accessed: 20 March 2020.

[Wat06]   WATTENBERG M.: Visual exploration of multivariate graphs. In
                 *Proceedings of the SIGCHI conference on Human Factors in computing
                 systems* (2006), ACM, pp. 811–819.

[WBS07]   WEINBERG J., BROWN L. D., STROUD J. R.: Bayesian forecasting
                 of an inhomogeneous poisson process with applications to call
                 center data. *Journal of the American Statistical Association 102*, 480
                 (2007), 1185–1198.

[WGK15]   WARD M. O., GRINSTEIN G., KEIM D.: *Interactive data visualiza-
                 tion: foundations, techniques, and applications (2nd Edition)*. CRC
                 Press, Boca Raton, FL, 2015.

[WGS15]   WANG L., GONG J., SHI L.: Olap visual analytics on large
                 software call graphs with hierarchical chordmap. In *2015 IEEE
                 International Conference on Data Mining Workshop (ICDMW)* (2015),
                 IEEE, IEEE, pp. 675–679.

[Wic16]   WICKHAM H.:   *ggplot2: Elegant Graphics for Data Analysis*.
                 Springer, New York, NY, 2016.

[Wil11]   WILLS G.: *Visualizing time: Designing graphical representations for
                 statistical data*. Springer Science & Business Media, New York,
                 2011.

[WJHSL15]   WRIGHT JR R. S., HAEMEL N., SELLERS G. M., LIPCHAK B.:
                 OpenGL SuperBible: Comprehensive Tutorial and Reference. Addison-
                 Wesley Professional, Boston, MA, 2015.

[WL00]   WARD M. O., LIPCHAK B. N.: A visualization tool for exploratory analysis of cyclic multivariate data. *Metrika 51*, 1 (Jul 2000), 27–37.

[WLS98]   WOODRUFF A., LANDAY J., STONEBRAKER M.: Constant density visualizations of non-uniform distributions of data. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 1998), UIST '98, ACM, pp. 19–28.

[Wol11]   WOLFF D.: *OpenGL 4.0 Shading Language Cookbook*. Packt Publishing Ltd, Birmingham, 2011.

[Won10]   WONG D. M.: *The Wall Street Journal Guide to Information Graphics: The Dos and Don'ts of Presenting Data, Facts, and Figures*. WW Norton, New York, NY, 2010.

[Yau11]   YAU N.: *Visualize this: the FlowingData guide to design, visualization, and statistics*. Wiley, Indianapolis, IN, 2011.

[Yau13]   YAU N.: *Data points: Visualization That Means Something*. John Wiley & Sons, Indianapolis, IN, 2013.

[YWR02]   YANG J., WARD M. O., RUNDENSTEINER E. A.: Interring: An interactive tool for visually navigating and manipulating hierarchical structures. In *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on* (2002), IEEE, pp. 77–84.

[YWR03]   YANG J., WARD M. O., RUNDENSTEINER E. A.: Interactive hierarchical displays: a general framework for visualization and exploration of large multivariate data sets. *Computers & Graphics 27*, 2 (2003), 265 – 283.

[ZCB11]   ZHAO J., CHEVALIER F., BALAKRISHNAN R.: Kronominer: Using multi-foci navigation for the visual exploration of time-series data. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2011), CHI '11, Association for Computing Machinery, pp. 1737–1746.

[ZFAQ13]   ZENG W., FU C.-W., ARISONA S. M., QU H.: Visualizing interchange patterns in massive movement data. *Computer Graphics*

Forum 32, 3 (2013), 271–280.

[Zhu13]   ZHU N. Q.: *Data visualization with D3. js cookbook.* Packt Publishing Ltd, Birmingham, 2013.

[ZRL96]   ZHANG T., RAMAKRISHNAN R., LIVNY M.: Birch: An efficient data clustering method for very large databases. *SIGMOD Rec. 25*, 2 (June 1996), 103–114.