Dynamic Geospatial Visualization

Liam McNabb
of Swansea University

Swansea University
Prifysgol Abertawe

A dissertation submitted to the University of Swansea
for the degree of Doctor of Philosophy

Abstract

The thesis concerns the topic of dynamic geospatial maps and how the user can leverage interaction techniques to reduce the complexity of maps while avoiding the removal of detail completely. Our first chapter introduces the greater context of the research and provides a breakdown of the following chapters.

Information visualization as a field is growing rapidly in popularity since the first information visualization conference in 1995. However, as a consequence of its growth, it is increasingly difficult to follow the growing body of literature in the field. Survey papers and literature reviews are valuable tools for managing the high volume of previously published research papers, and the quantity of survey papers in visualization has reached a critical mass. To this end, chapter 2 takes a quantum step forward by surveying and classifying literature survey papers in order to help researchers understand the current landscape of Information Visualization. It is, to our knowledge, the first survey of surveys (SoS) in Information Visualization. The second chapter classifies survey papers into natural topic clusters which enable readers to find relevant literature and develops the first classification of classifications. The chapter also enables researchers to identify both mature and less developed research directions as well as identify future directions. It is a valuable resource for both newcomers and experienced researchers in and outside the field of Information Visualization and Visual Analytics.

Choropleths are a common and useful way of depicting area-coupled data on a geo-spatial map. One advantage they provide is combining area-based data accurately with geo-space. However perceptual problems arise when areas are too small, i.e. when they only cover a few pixels or less. This is a widespread occurrence when zooming or in densely populated areas like capital cities. In Chapter 3, we present a novel algorithm that ensures the user can observe area-based data coupled to geo-space based on their interactive level of zoom without distorting the original geo-spatial

map. This is resolved by building a hierarchical data structure in which each area and its data is merged with one of its smallest neighbors recursively until only one polygon covers each contiguous region. The benefits are that the viewer can always view area-based data contained in the map regardless of how small any individual area becomes during interactive zooming. We break down each step of the algorithm and provide pseudo-code to enable reproducibility. We also discuss unique test cases that challenge the robustness of the algorithm with 30,000 polygons and 4,652,800 vertices as well as the performance.

Choropleth maps are an invaluable visualization type for mapping geospatial data. One advantage to a choropleth map over other geospatial visualizations such as cartograms is the familiarity of a non-distorted landmass. However, this causes challenges when an area becomes too small in order to perceive the underlying color accurately. When does size matter in a choropleth map? In Chapter 4, we conduct an experiment to verify the relationship between choropleth maps, their underlying color map, and a user's perceivability. We do this by testing a user's perception of color relative to an administrative area's size within a choropleth map, as well as user-preference of fixed-locale maps with enforced minimum areas. Based on this initial experiment we can make the first recommendations concerning a unit area's minimum size in order to be perceivably useful.

Maps are one of the most conventional types of visualization used when conveying information to both inexperienced users and advanced analysts. However, the multivariate representation of data on maps is still considered an unsolved problem. In Chapter 5, we present a multivariate map that uses geo-space to guide the position of multivariate glyphs and enable users to interact with the map and glyphs, conveying meaningful data at different levels of detail. We develop an algorithm pipeline for this process and demonstrate how the user can adjust the level-of-detail of the resulting imagery. We present a selection of user options to facilitate the exploration process and provide case studies to support how the application can be used. We also compare our placement algorithm with previous geo-spatial glyph placement algorithms. The result is a novel glyph placement solution to support multi-variate maps.

In Chapter 6, we discuss some of the software design and development challenges we encountered throughout the PhD candidature. The biggest challenge came with the development of our area merging process. When

algorithms are needed for large or complex geo-spatial data, it is essential that the user understands the progression, and identifies errors in their code. We present the techniques we used to debug the algorithm discussed in the main body of the thesis, and some of the ways they helped facilitate the construction of the algorithm. Chapter 6 also contains a section dedicated to intersection testing. Performance was not always crucial, being a pre-processing step, however common intersection testing could be run millions of timea every time a new file is loaded. We examine common intersection tests that could be necessary and compile them into a manifest header file for use with C++, for basic 2D primitives.

Our final chapter provides closure to the thesis. We discuss the state of scale, and it's perceived reception within the field, we give our thoughts on the use of the algorithm itself, and review future work that could be reviewed in a follow-up PhD project.

We also provide an Appendix section presenting an educational paper on how to prepare and design a survey paper. We discuss how to search for papers, how to identify a topic, as well as how to create a classification and present findings.

Declaration

This dissertation is the result of my own work, except where explicit reference is made to the work of others, and has not been submitted for another qualification to this or any other university. I herby give 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: .....

Acknowledgements

I am thankful for all the help I have been given along my PhD candidature including my Parents, brother and sister, who made jokes and who supported me when necessary. I'd like to thank my Granny Rose and Nana Mairaed who also reassured me when stress got the better of me. I dedicate my thesis to my Grandad Sam and Papa Jonjo, who passed away during 2018.

I also want to thank my friends who kept me going and those who made sure my work was clear and professional. Finally, I want to thank my supervisor who let me voice my opinions clearly and did not budge on his solutions, allowing us to build some great software and papers.

Contributions

This thesis is based on the following papers:

1. [**?**]
   Supplementary Video: n/a

2. [**?**]
   Supplementary Video: https://bit.ly/2wYX0Ok

3. [**?**]
   Supplementary Video: https://bit.ly/2M9wIvY

4. [**?**]
   Supplementary Video: https://vimeo.com/314225790 [password:mcnabbthesis]

5. [**?**]
   Supplementary Video: n/a

Contents

“It’s a rare artistic choice to have the bar fill up but not actually be done loading.”

— Ryan Letourneau

Chapter 1

Introduction

“ A graphic is no longer “drawn” once and for all; it is “constructed” and reconstructed (manipulated) until all the relationships which lie within it have been perceived.”

— Jacques Bertin, 1981

Contents

1.1 Data Visualization

Murray describes Data Visualization as "*a process of mapping information to visuals*" [**?**], ideally something that improves over the raw data and proves a more comprehensible concept to a reader. Data can come from a potentially infinite number of sources. Designing robust techniques that handle data, as well as present meaningful interpretations that harbor large amounts of research and exploration into design are some goal of Data Visualization.

Evidence of Data Visualization is found as early as the 17th Century. In 1679, John Adams used Maps to depict the distance between different cities which could be used by travellers [**?**] (Figure **??**). These are comparable to modern network maps. They gathered distance metrics with visual representation in geospace. Around the dawn of the 19th Century is considered the birth of modern data graphics [**?**]. William Playfair played an essential role in the history of visualization, known as the inventor of many common visual designs such as the bar chart, line chart, area chart, and pie chart [**?**, **?**, **?**]. See Figure **??**.

Data visualization has only become more intrinsic to data analysis, with an increasing number of books being published on the topic [**?**]. In order to focus the discussion, we only discuss the related sub-fields that provide a natural transition between Data Visualization and our thesis topic. Refer to **??** for a visual breakdown.

Figure 1.1: John Adams' Map of England, courtesy of Heawood [**?**].

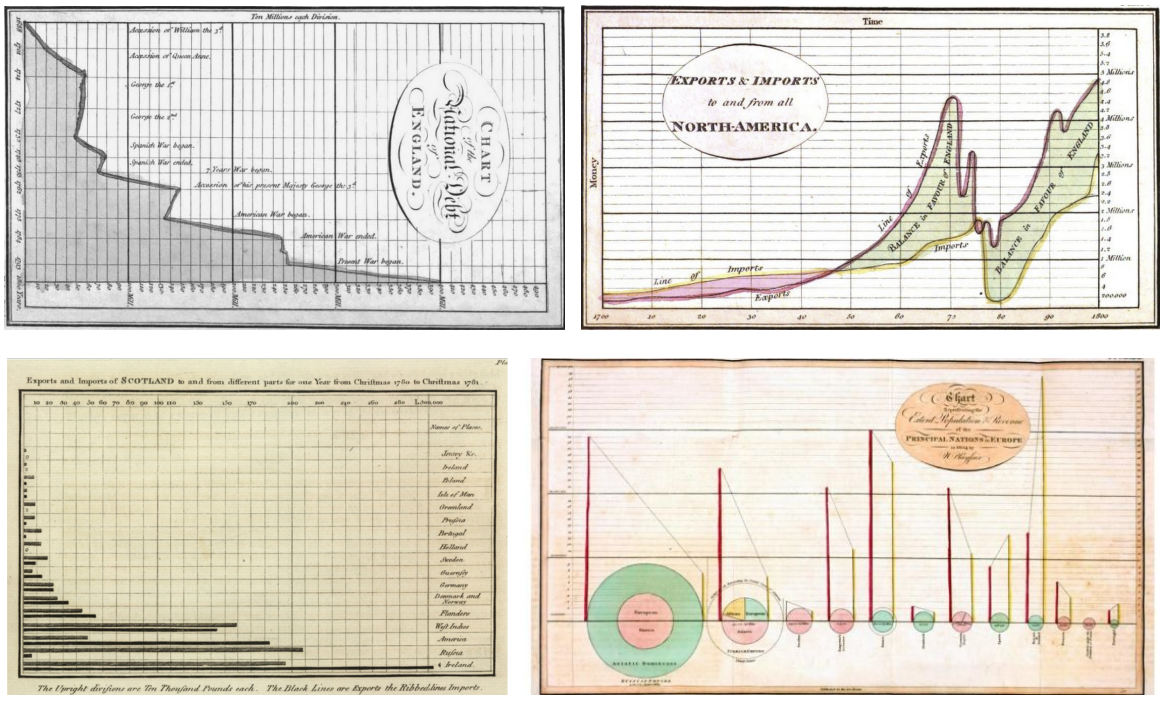


Figure 1.2: Some of Playfair's original works. In order, Area chart and Line chart [**?**], Bar chart [**?**], and Pie chart. Image courtesy of Playfair [**?**]

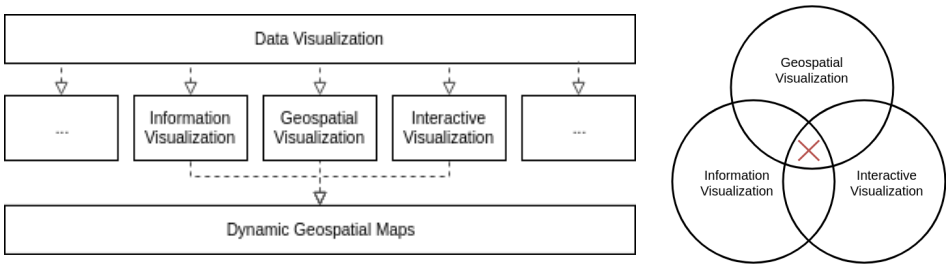
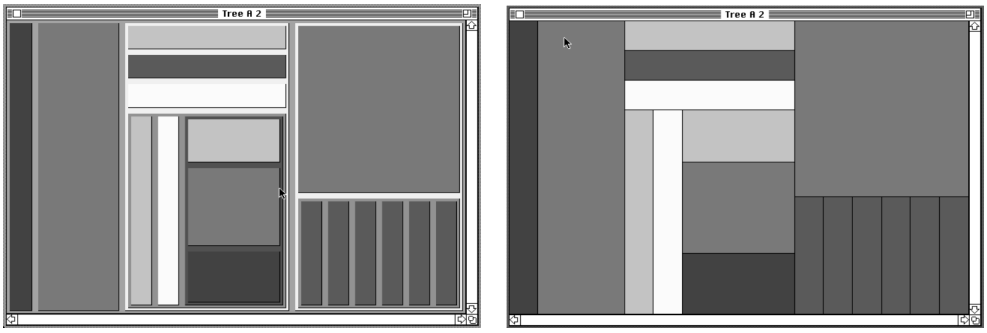


Figure 1.3: A breakdown of the relevant fields discussed in the thesis. In the form of (a) a relationship diagram and (b) a venn diagram. Field selections are guided by Keim et al. [**?**]

1.1.1 Information Visualization

Information visualization (InfoVis) is a sub-field of Data Visualization that focuses on abstract representations of abstract data where abstract is defined as: "*existing in thought or as an idea but not having a physical or concrete existence*" by the Oxford Dictionary [**?**]. InfoVis uses data and calculates or assigns a new abstract visual design to represent information to a user. Because of this, InfoVis provides a large range of unique techniques that can seem almost unrelated. We provide a few examples.

Treemap: A treemap is a representation of hierarchical data. By assigning the area of a treemap to the value an area holds, and applying this per hierarchy level, Johnson and Schneiderman created a space-filling algorithm [**?**]. See Figure **??**. The treemap has become popular amongst visual analysts, with the main algorithm tweaked for many visual designs such as squarified [**?**], strip [**?**], and slice and dice [**?**], and is applied to many hierarchical datasets such as time [**?**].



(a) Nested Treemap **(b)** Non-Nested Treemap

Figure 1.4: The early treemap designs. Image courtesy of Johnson and Schneiderman [**?**]

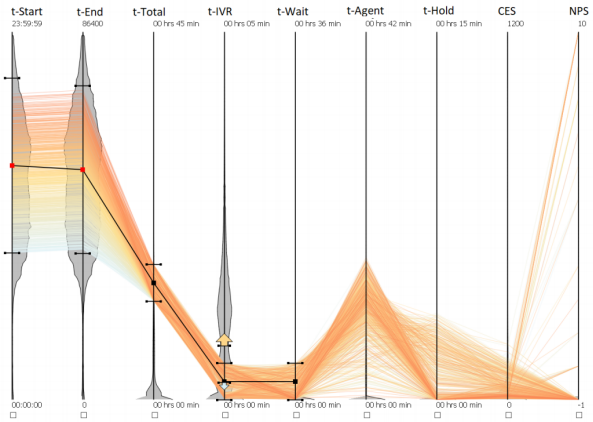


Figure 1.5: An example of a parallel coordinate plot, courtesy of Roberts et al. [**?**].

Parallel Coordinates: Parallel Coordinates are high dimensional representations of multivariate datasets used to search for relationships between attributes of data. For a standard parallel coordinate plot, each dimension is represented by a vertical axis, where the line represents the data range for the dimension. Each data record is assigned a polyline which connects each dimension in the range that represents each attribute. The parallel coordinate was created by Inselberg [**?**]. See Figure **??**.

Graphs: Graphs are a large subset of InfoVis due to their utility. Graphs are used to represent connections or interfacing between objects. They can be used to visualize clusters or networks and flow in abstract space. Figure **??** presents a collaboration diagram created to convey how a class (created for later techniques) interacts with other classes.

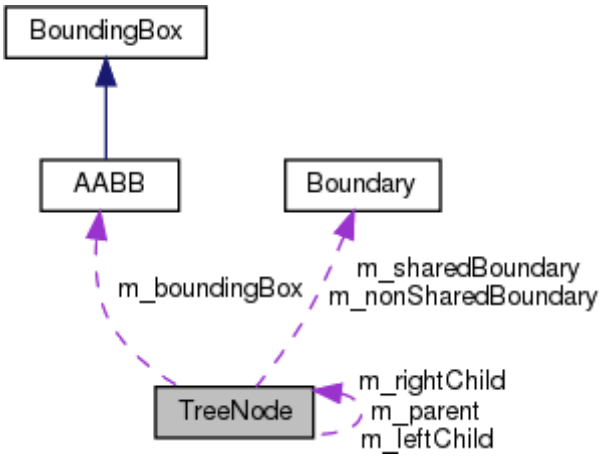


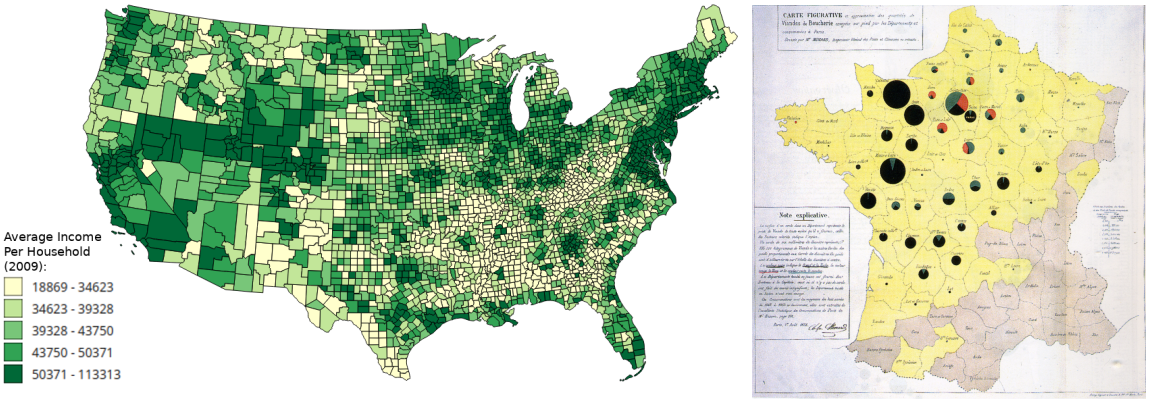
Figure 1.6: A collaberation diagram to present how a class interacts with other classes. This graph shows TreeNode contains three TreeNode objects, two Boundary objects, and an AABB, of type BoundingBox. Refer to Chapter **??**.

1.1.2 Geospatial visualization

Geospatial visualization differs from InfoVis by incorporating spatial context as a geographical representation [**?**]. The benefits of Geospatial visualization are with recognizability to the general public. However, their static nature can pose challenges. We present some examples of geospatial visualization.

Choropleth Map: Choropleth maps are one of the oldest geographic visualization types [**?**]. A choropleth map uses a standard geographical map split into administrative areas. Each administrative area uses a visual mapping technique (such as color) to present associated values. The choropleth map is an integral part of Chapter **??** and is discussed more in detail there. See Figure **??**(left).

Symbol Maps: A symbol map differs from the choropleth map by using a symbol to visualiz the value of specific locations rather than mapping straight to the area itself [**?**]. However, both are often used in unison. Many symbols maps use size to represent the value of the underlying area. See Figure **??**(right).



(a) Choropleth **(b)** Minard's Symbol Map

Figure 1.7: (left) A choropleth map depicting the average income of a household per US County [**?**]. Generated using QGIS [**?**] . (right) Minard's famous example of a proportional symbol map, representing the cow consumption of France per area. Image courtesy of Palsky [**?**].

1.1.3 Interactive Visualization

Dynamic visualization varies from our previous two topics by acting as an enhancement to existing tools. We consider interactive visualization to represent critical areas of engagement with a user including animation, user-input and adaptive views. All of these give the user control over how they want to manipulate or modify a visualization, whether that is through direct or indirect input. Interactive visualization is a prominent theme throughout the thesis. Tominski describe interactive techniques in visualization as "the basic building blocks to complement the visualization pipeline with respect to interaction" [**?**].

1.1.4 Dynamic Geospatial Maps

We review elements of information, geospatial, and dynamic visualization in order to develop dynamic geospatial maps. We input raw geographic data, and present the data in geospatial visualization form, however, we incorporate steps to enhance the spatial context of the geographic data, using information visualization techniques. Finally, we enable dynamic visualization of our data through both direct and indirect manipulation of the view by the user. We provide a visual representation in Figure **??**.

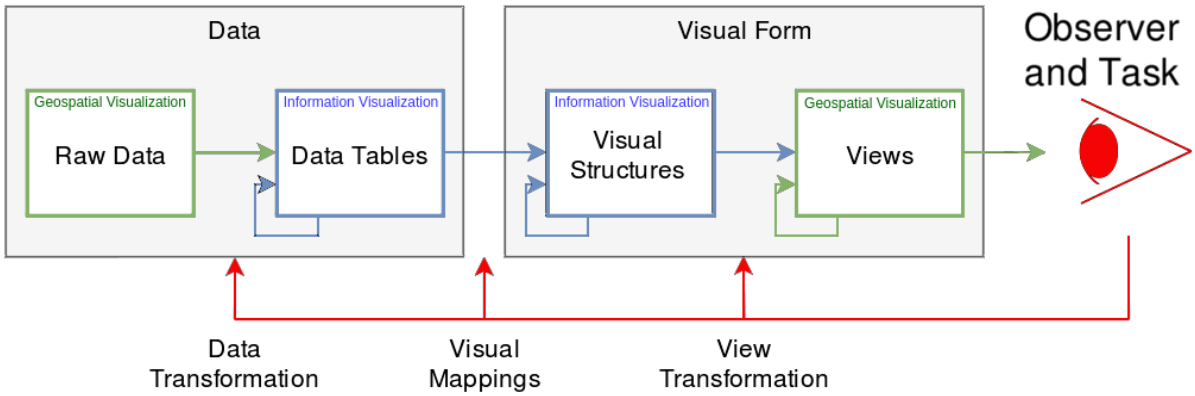


Figure 1.8: The information visualization pipeline introduced by Card et al [**?**]. We highlight different sections to show how our three topics combine, green represents geospatial visualization, blue represents information visualization, and red represents dynamic visualization.

1.2 Challenges

We consider three main challenges within this thesis and our goals:

1. **Volume of Literature:** There is a large volume of literature in data visualization. In order to understand the needs of a user, we must clearly understand the existing in literature to inform the current research topics that need to be examined.

2. **Scalability:** Scalability is one of the major challenges we face in the field of visualization and is one we address in this thesis. Administrative areas of maps feature an assortment of complex boundaries, but the granularity of the areas can vary, leading to large amounts of complex borders. Administrative areas can be so small that it becomes difficult to interpret any data from them. For example, the United Kingdom holds over 180,000 areas of varying sizes, making an overview of the data recorded almost impossible without first transforming the data to a lower resolution. Our first and major challenge is to adress this challenge by using dynamic visualization to intuitively represent appropriately sized areas based on the user's current view.

3. **Perception:** Our second challenge lies with the representation of areas. Cartograms already fill the niche of distorting area size to present administrative areas and their values. However, these techniques can either make it difficult or almost impossible to diagnose particular areas at just a glance. In order to avoid this, we need to produce a technique that minimizes any changes to the representation, offering an easier understanding of the underlying data.

4. **Multivariate Geospatial Data:** If we consider scalability as the breadth of data points, the we consider multivariate aspect as the depth per data point. If we want to present something a scalable we need to consider both of these aspect. In order to to do this, we need to create a multivariate geospatial map that presents meaningful data to the user.

5. **Performance:** Our final challenge lies with performance. The amount of computation necessary to update the display during rendering is counter-intuitive. We need to ensure that the technique we develop does not interfere with the user's exploration of the data.

1.3 Research Methodology

In order to approach these challenges, we will use the following approach. First, we will gather a large amount of literature to gain a keen understanding of the visualization landscape in the form of a literature review. This should provide evidence on the impact of scalability challenges, and approaches that have already been considered. Once we have confirmed these elements, we can move on to the design and implementation of a technique to tackle the challenge set forth. With this, our goal will be to create a set of maps that will manipulate to aid the user at different levels of zoom. Once we have done this, we will study the impact to gather more insight into how our algorithm is used, and the positive and negatives associated. Finally, we will implement the algorithm into a more robust piece of software, including real-world multivariate data.

This methodology relies on the literature review as the first step in order to gain an in-depth understanding, rather than focusing on our studies. We leave this until we have a stronger understanding of our approach in order to confirm or deny the impact of our research. At this point, we aim to present a stronger piece of software to present how the algorithm can aid in the visualization of data.

1.4 Contributions

The contributions of this thesis include:

1. **Volume of Literature:** The first Survey of Surveys covering the landscape of Information Visualization [**?**]. The survey summarizes and classifies other survey papers in order to enable both newcomers and experienced researchers to obtain a greater understanding of the landscape of literature that has, and has not yet been published, as well as a broader understanding of future research challenges in the domain. We also provide a new educational paper as a guide to create your own survey paper.

2. **Scalability:** The development of a novel technique for presenting choropleth maps at multiple levels of detail through user interaction [**?**]. This is resolved by building a hierarchical data structure in which each area and its data is unified with one of its smallest neighbors recursively until only one polygon covers each contiguous region. The benefits are that the viewer can always view area-based data contained in the map regardless of how small any individual area becomes during interactive zooming.

3. **Perception:** An experiment to verify the relationship between choropleth maps, their underlying color map, and a user's perceivability [**?**]. We do this by testing a user's perception of color relative to an administrative area's size within a choropleth map, as well as user-preference of fixed-locale maps with enforced minimum areas based on the algorithm outlined in the Dynamic Choropleth Map algorithm.

4. **Multivariate Geospatial Data:** A multivariate map that uses scale-awareness to position multivariate glyphs and enables users to interact with both the map and glyphs to show meaningful data at different levels of detail [**?**]. We discuss the algorithm pipeline for this process, as well as how the user can review and interact with the data. We present some user options to facilitate the exploration process and provide observations to support how the application can be used. We finish by reviewing the utility of the algorithm by looking at three case studies and comparing the map against an existing grid placement strategy. The result is a novel glyph placement solution to support multivariate maps.

1.5 Thesis Structure

The remainder of this thesis incorporates the following structure: In Chapter **??**, we present a survey of related work in the form of a survey of survey papers (SoS), that examines how research areas relate across different fields of Information Visualization. In Chapter **??**, we describe a novel technique to present choropleths at multiple levels of detail using a hierarchical data structure to unify administrative areas. Chapter **??**, uses the algorithm presented in Chapter **??** to investigate how error relates to the scale of administrative areas on a map via a user study. The results provide guidance on the relationship between size and error, size and performance time, and some first steps in reviewing user preference in map design, concerning size. Chapter **??** also builds on Chapter **??** by applying the choropleth algorithm to multivariate maps, using glyphs to present data for administrative areas. We improve upon the original method by adding smooth transitions and uncertainty indicators, while also comparing our algorithm against existing glyph-placement strategies. We look at some of the exciting aspects and challenges we ran into during the process of developing the software in Chapter **??**. Finally, in Chapter **??** we draw our conclusion on the topic and discuss potential future work that can be applied to the themes in the thesis. We also include an educational tutorial discussing how to write a survey paper in Appendix **??**.

Chapter 2

Survey of Surveys (SoS)

[**?**]

“ But nowadays there’s lots of other journals and it takes more and more effort to make sure that you know what’s happening.”

— Jim Blinn, 1998

Contents

Chapter Abstract

Information visualization as a field is growing rapidly in popularity since the first information visualization conference in 1995. However, as a consequence of its growth, it is increasingly difficult to follow the growing body of literature within the field. Survey papers and literature reviews are valuable tools for managing the great volume of previously published research papers, and the quantity of survey papers in visualization has reached a critical mass. To this end, this survey chapter takes a quantum step forward by surveying and classifying literature survey papers in order to help researchers understand the current landscape of Information Visualization. It is, to our knowledge, the first survey of surveys (SoS) in Information Visualization. This chapter classifies survey papers into natural topic clusters which enables readers to find relevant literature and develops the first classification of classifications. The chapter also enables researchers to identify both mature and less developed research directions as well as identify future directions. It is a valuable resource for both newcomers and experienced researchers in and outside the field of Information Visualization and Visual Analytics.

2.1 Introduction and Motivation

When we first discussed potential survey papers in 2016 for the thesis, we considered Smart City Visualization. After an initial search, we found a book already published with a survey on this topic by Ciuccarelli et al. [**?**]. As a result, we decided to narrow the scope of the literature review to public transport and found a survey paper on the topic by Chen et al. [**?**]. After this, we discussed two different potential survey topics including geospatial visualization and human movement visualization. Subsequently we found survey papers for each by Nusrat and Kobourov, and Gavrila [**?**, **?**]. At this point of surprise, we decided that a survey of surveys would be a logical direction forward. We looked at the benefits of surveying different survey papers which would allow us to gain a good understanding of unsolved problems across the whole landscape of information visualization, specifically for scale and zooming problems which we discuss throughout the thesis.

"It used to be that SIGGRAPH was the only place that would publish computer graphics papers, and so all you had to do was read the SIGGRAPH Conference Proceedings and you knew you were up to date. But nowadays there's lots of other journals and it takes more and more effort to make sure that you know what's happening." This quote is taken from Jim Blinn's renowned keynote speech at SIGGRAPH 98 [**?**]. Decades later this theme is still considered one of the most important challenges for research in any field.

Information Visualization is a rapidly evolving research field defined as "the communication of abstract data through the use of interactive visual interfaces" [**?**]. Because of this, many researchers spend countless hours on research and development of Information Visualization techniques only to discover that research on a given topic has already been published. Survey papers and literature reviews are valuable and critical tools for managing the great number of previously published research papers. However even the number of survey papers themselves has reached a critical mass, thus inspiring a quantum step forward.

In this first undertaking of a survey of surveys (SoS), we aim to present the landscape of rapidly evolving research within Information Visualization. In order to emphasize open directions for future work, we present literature reviews of papers that survey research topics, and extract essential information from them systematically as a guide for both newcomers and experts in the field and beyond. We then classify over 80 survey papers to examine trends and themes that have recently been published. Our contributions to the field include:

- A quantum step in literature reviews presenting the first meta-survey, i.e. a 'Survey of Surveys' (SoS).

Conferences & Journals	Related Papers
The Annual EuroVis Conference	20
IEEE TVCG Journal	18
IEEE Pacific Visualization Symposium	2
IEEE VAST Conference	3
The Annual Eurographics Conference	3
Journal of Visual Languages & Computing	2
Information Visualization Journal	5
Computer Graphics Forum	3
ACM Computing Surveys	0
Other	30
Total:	86

Table 2.1: A list of literature sources we search for survey papers, with the quantity of papers identified from each. For paper searching, we use IEEE Xplore [**?**], ACM Digital Library [**?**], Google Scholar [**?**], and Vispubdata [**?**]

- A novel classification of survey literature in the field of Information Visualization, which can be used as a guide for new researchers or a tool for field experts.
- The first classification of literature classification schemes.
- A structured overview of both mature and less developed future research directions that cover the domain of Information Visualization.

2.1.1 Literature-based Challenges in the Field

There are at least three major difficult challenges within this field:

1. *Understanding what has been already been done*: Many researchers end up losing time due to challenges associated with literature searches. As the Information Visualization landscape grows, so does the number of papers, conferences, and journals. This makes it increasingly difficult to find topic related papers.
2. *Understanding what areas in the domain have yet to be explored*: A researcher may not be truly sure whether a paper does not exist. This is a logical uncertainty. Until the point a paper is discovered, a researcher may still be unsure whether their development has been explored or not.

3. *Making sure your discoveries are not ignored*: Papers are published to present discoveries in their field. However, due to the volume of published material, papers can be missed, forgotten, or reinvented. This challenge is also highlighted by Jim Blinn [**?**].

The SoS aims to address these challenges by surveying a collection of over 80 survey papers to enable a quick overview of the scope of research directions, what has already been done, and which directions are more open for research. We provide systematic summaries of these survey papers for those with interest in the field. This serves as a valuable starting point for young researchers and a practical reference guide for field experts. We also believe this survey of surveys will reach audiences beyond the field of information visualization and visual analytics, and entice more researches towards the area.

2.1.2 Literature Search Methodology

Our survey search methodology includes a combination of linear-search and relation-search. Our starting point includes reviewing previous EuroVis State-of-the-Art (STAR) papers. The linear search focuses on looking at each journal or conference and checking each paper that includes keywords such as 'Survey', 'Taxonomy', or 'State-of-the-Art'. The relation-search includes searching the references of each survey paper for related survey papers.

Sources that are searched within our SoS following this methodology are summarized in Table **??**. Our survey chapter search lasted over a year.

2.1.3 Classification Overview

In order to classify each survey, we develop categorical dimensions. The dimensions are derived from previously published and well-known literature based on:

1. An adapted Information Visualization pipeline model originally presented by Card et al. [**?**]. A visual aid to this pipeline can be found in Figure **??**.
2. Subject-based clusters guided by SurVis [**?**] and the survey paper topics themselves.

The Information Visualization Pipeline

The Information Visualization pipeline model we use to classify the surveys is based on that presented by Card et al. in their classic book *'Readings in Information Visualization'* [**?**] (See Figure **??**). The pipeline describes the transition of raw data into a visualization which is

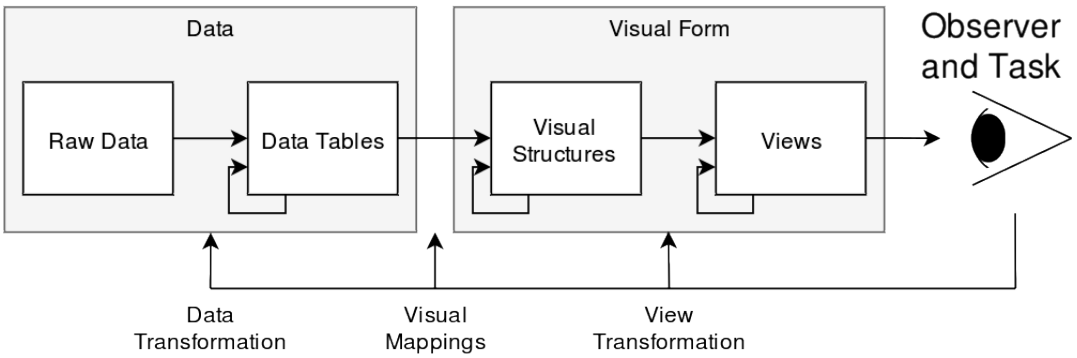


Figure 2.1: The original Information Visualization Pipeline model created by Card et al. [**?**] which we adapt to design our modified classification.

visible to a user. This consists of (1) raw data transformed into data tables via the use of data transformations. (2) Data tables transformed into visual structures via the use of visual mappings. (3) Visual structures transformed into views via the use of view transformations. The final step is (4) User input manipulating the data in order to feed back into the pipeline. For the purpose of the SoS, the pipeline is adapted in order to facilitate the categorization process. The following pipeline stages are used.

1. **Data Enhancement & Transformation** - Data Enhancement and Transformation is used to describe the raw data that is transformed or enhanced in order to derive a data structure(s) that can be used for visualization. The classification also includes how the data is captured and how the data is classified. Survey papers that are data-centric are placed in this category.

2. **Visual Mapping & Structure** - Visual Mapping and Structure defines the techniques to visualise data or data structures. This section also examines how a visualization is structured, such as how the mapping is used or the facets that are included. This category involves mapping the enhanced data to visual primitives, for example, color, opacity, textures, and geometry such as points, edges, as well as 2D and 3D shapes. Survey papers with an emphasis on visual mapping and structure are categorized here.

3. **Exploration and Rendering** - The three common types of view transformation are location probes that use location to reveal additional information, viewpoint controls which are used to scale or translate a view, and distortions which modify the visual structure [**?**]. Exploration and Rendering looks at these transformations along with the rendered representation views and projections. This is the presented state a visualization takes upon completion and aims to present the data to the user. Survey papers with a focus on rendering and exploration are classified here.

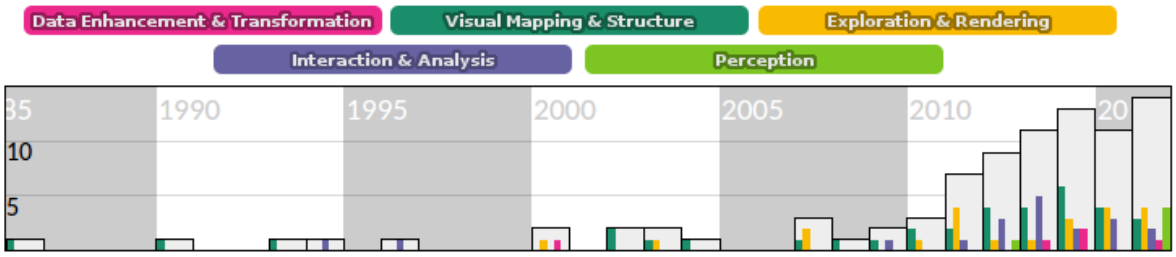


Figure 2.2: A histogram providing the frequency of surveys found for each year mapped along the x-axis. each bar represents one year whilst the y-axis provides the amount of survey papers found for each year that meet our scope (Section **??**). The colored bars represent a further breakdown of the survey papers based on their given classification dimension (discussed in Section **??**). The visualization is taken from the SoS literature browser [**?**].

4. **Interactive Analysis** - Analysis refers to how the user provides feedback to a visualization. A user can connect with a visualization manually, by modifying or transforming a view state, or by reviewing the use, effectiveness, and their knowledge on the visualization. This also includes selection protocols and mapping techniques such as task taxonomy or other viariations of selection. Survey papers with a focus on interactive analysis or tasks are placed here.

5. **Perception** - Perception examines the cognitive interperatation of a visualization from the perspective of a user. Perception can be viewed through the design and creation of user studies or papers relating the visual system to Information Visualization. Survey papers emphasizing perception and user-studies are placed in this category.

These components form the basis of the first dimension of our classification. Figure **??** provides a breakdown of these classification by year.

Subject-based Clusters

Literature and subject-based clusters group similarly focused survey papers. SurVis [**?**] is used as an aid in order to discover and diagnose suitable clusters, exploring keywords related to each paper. Survey papers that cite previous survey papers create natural topic clusters that are taken into account within our subject-based clusters. The subject clusters are as follows – Data-Centric contains literature that focus on types of data, or data itself. Multivariate

& Hierarchical focus on structured data, or data that visualizes many dimensions. Graphs and Networks focus on literature that discusses nodes and edges used in visualization, usually standard 2D views. Geospace+Time review surveys with that look at dimensional data. Co-

• Data-Centric	◇ Data-Types
	◇ Text-Focus
• Multivariate & Hierarchical	◇ Hierarchical
	◇ High-Dimensional Overview
	◇ Parallel Coordinates
	◇ Glyphs
• Graphs & Networks	◇ Graphs
	◇ Networks
• Geospace + Time	◇ Temporal
	◇ Geospatial
• Coordinated Multiple Views	
• Real-World & Applications	◇ Finance
	◇ Healthcare
	◇ Security
	◇ Systems
	◇ SoftVis
	◇ Frameworks
• Overview	◇ Focus+Context
	◇ Provenance
	◇ General

Table 2.2: Hierarchy of subject-based clusters.

ordinated Multiple Views (CMVs) centers around literature that examines the coordination or linkage between multiples. Real-World and Applications focuses on literature that reviews topics with an emphasis on practical data or with real-world users. Overview contains literature that provide a more broad survey of the information visualization landscape. A full breakdown of the topics can be found in Table **??**.

2.1.4 SoS Scope

Restrictions are used in order to define, manage, and constrain the scope of the SoS.

Firstly, only survey papers with a focus on Information Visualization and Visual Analytics found within the field are included. This means Scientific Visualization does not fall within the scope. There are several recent scientific visualization surveys not included. For purposes of this survey, we define scientific visualization (SciVis) as the following:

"Data that describes a physical phenomenon is defined as scientific data. Examples of this are fluid flow, living organisms, and data from the natural world." For the purpose of this survey, Euclidean space-time coordinate data is considered Scientific Visualization. Lipsa et

al. survey visualization in physical sciences, this does not meet the criteria of our survey [**?**]. Edmunds et al. present a framework for flow visualization [**?**]. Flow visualization is considered SciVis. Blascheck et al. 's survey on Eye-Tracking data is a review of literature focused on data from a physical phenomenon and can therefore be classed as SciVis [**?**].

Papers that focus on Computer Vision are not considered within scope. Datondji et al. present a survey focused on vision based traffic monitoring of road intersections. This is considered a computer vision topic and therefore does not meet the scope of this review [**?**].

Computer graphics and Graph theory literature surveys are not considered in the scope of the SoS. Ghosh and Goswami present a paper reviewing unsolved problems in visibility graphs [**?**]. Although the paper works on graphs, the review focuses on the mathematics of graph theory and not visualization. Biomedical and computational biology are beyond the scope of this SoS. They are covered in an affiliated SoS called the SoS-MDV [**?**].

Secondly, publication date is considered. This enables a clear emphasis on recent advancements, and what can be done at this time to increase our advancement in the field of Information Visualization. This is important as it allows us to provide a clearer message about future research fields, as the older survey papers may discuss research areas that are now mature. The emphasis for the SoS is between the years 2010 and 2016. As prior papers are still important, we include surveys that fall out of this time-frame onto our classification table, however, we do not include a detailed description of them. Older surveys that are included in the table only can shed historical light on where authors chose to publish visualization paper before the visualization community developed (prior to 1990).

Finally, the SoS emphasizes literature reviews, as opposed to comparison-oriented papers. Shneiderman et al. look at the innovation trajectories of treemaps, conemaps, and hyperbolic trees, but focus on the global state of each techniques' citations and papers, rather than a comparison of individual papers [**?**]. Sedlmair et al. look at scatter-plot and dimension reduction technique choices, as well as multiple reduction techniques for scatter-plots. These techniques are compared as abstractions [**?**]. Papers such as these are not given a detailed summary, however, we include these in the classification table for completeness.

2.1.5 Background

To the best of our knowledge, there are no previous papers that attempt to review literature in this way. Other papers use alternative methods to address the literature explosion challenge. Laramee et al. provide an in depth review of unsolved problems in human-centered visualization [**?**]. The review provides an in-depth understanding of challenges identified for each paper which differs from the solution our chapter uses, that appropriates important research

		Information Visualization Pipeline				
		Data Enhancement & Transformation	Visual Mapping & Structure	Exploration & Rendering	Interaction & Analysis	Perception
Data-Centric	Data-Types		[?] [?] [?]	[?] [?]		
	Text-Focus	[?]	[?] [?] [?] [?] [?]	[?]	[?] [?]	
Multivariate & Hierarchical	Hierarchical	[?]	[?] [?] [?]			
	High-Dimensional Overview			[?] [?]	[?]	
	Parallel Coordinates		[?]		[?]	[?]
	Glyphs		[?] [?]			[?]
Graphs & Networks	Graphs	[?]	[?] [?] [?] [?]	[?] [?]	[?] [?] [?]	
	Networks		[?]	[?]	[?]	
Geospace + Time	Temporal		[?]	[?]		
	Geospatial		[?] [?] [?]	[?]	[?]	
Coordinated Multiple Views (CMVs)			[?] [?]	[?] [?]		
Real-World & Applications	Finance	[?]		[?]		
	Healthcare		[?] [?]	[?]		
	Security			[?] [?]		
	Systems		[?]		[?] [?]	
	SoftVis		[?] [?] [?]	[?] [?]	[?]	[?] [?]
	Frameworks		[?]	[?] [?]		
Overview	Focus+Context		[?]	[?] [?]	[?]	
	Provenance				[?]	
	General	[?]	[?] [?]	[?] [?]	[?] [?] [?] [?] [?]	[?] [?]

Table 2.3: A 3-Dimensional hierarchical classification table depicting the categorization of all the survey papers. Green Highlighting represents survey's summarised within the SoS. Yellow Highlighting represents surveys that were not summarised in detail due to prioritization of journals or size constraints. Pink Highlighting represents survey's not reviewed in detail within the literature review due to year constraints discussed in the Scope (Section **??**).

challenges by looking at the frequency of each challenge across papers. Henry et al. review 20 years of conference publications from CHI, UIST, AVI and InfoVis [**?**].

Isenberg et al. present a novel visualization of a database of papers across InfoVis, SciVis, VAST and Vis [**?**]. We provide a different view of the data by clustering papers together to quickly understand domains, with a focus on survey papers. Isenberg et al. present topic popularity using research paper keywords across four conferences and provide related papers [**?**]. Our literature review differs by providing analysis of related papers with a focus on surveys to provide an overview and explore possible research areas in the field.

2.1.6 Organization of The SoS

Survey Literature is organized using a 2D matrix that incorporates the two classifications discussed in Section **??** and **??**. Each survey paper is placed at the most relevant intersection of each classification criteria matched. Color is used to signify the depth of which the literature is reviewed within. This is shown in Table **??**.

The SoS is structured using the subject-based clusters as the primary organization. This groups related papers together. The modified pipeline classification is ignored in favor of chronological order within the paper's organization. This enables us to describe a natural progression within each section for papers that are intrinsically related.

2.2 A Classification of Classifications

Classifications are an integral and important part of a survey paper. Table **??** systematically indicates how each survey paper's classification of literature is represented. We provide three characteristics of classifications: **dimension**, **structure**, and **mapping schema**. For this discussion C denotes a classification topic.

The **dimensionality** organizes the space in which the classification is laid out. We subdivide the dimensionality in three ways. One-dimensional (1D) classification presents the classification topics (C) in linear fashion. Two-dimensional classifications (2D) usually present more than one classification dimension, one on each axis (C), and are usually presented in the form of a table. The third category represents classification topics (C) with three or more dimensions. Common ways to represent additional attributes are through the use of color, shape, or symbols.

	Dimensions			Structure		Mapping Schema		
Literature:	1D	2D	3D+	Linear	Hierarchical	Indirect Mapping	Unique-Mapping	1-N Mapping
[?]		✓			✓	✓		
[?]			✓		✓			✓
[?]	✓				✓	✓		
[?]		✓			✓		✓	
[?]	✓			✓			✓	
[?]		✓		✓				✓
[?]			✓		✓			✓
[?]		✓			✓		✓	
[?]		✓			✓		✓	
[?]			✓	✓		✓		
[?]	✓				✓			✓
[?]		✓		✓		✓		
[?]		✓			✓			✓
[?]	✓				✓		✓	
[?]		✓		✓				✓
[?]			✓	✓				✓
[?]	✓				✓	✓		
[?]		✓		✓				✓
[?]		✓			✓			✓
[?]	✓			✓			✓	
[?]			✓	✓				✓
[?]			✓		✓		✓	
[?]		✓			✓		✓	
[?]		✓		✓		✓		
[?]		✓		✓				✓
[?]	✓			✓		✓		
[?]	✓				✓		✓	
[?]		✓			✓		✓	
[?]		✓		✓			✓	
[?]		✓		✓			✓	
[?]		✓			✓			✓
[?]		✓			✓	✓		
[?]			✓		✓		✓	
[?]		✓			✓	✓		
[?]		✓		✓				✓
[?]			✓		✓	✓		
[?]	✓				✓	✓		
[?]		✓		✓				✓
[?]		✓		✓				✓
[?]			✓		✓			✓

Table 2.4: A Categorization of classification tables found within each primary survey paper (highlighted green in Table **??**). The table examines how many dimensions each survey table features, the structure of each survey classification, and the type of mapping schema it incorporates. This table uses the paper's visual representation of the classification. If there is more than one classification, the primary classification is shown. This table itself corresponds to the classification example shown in Figure **??** (B).

Structure represents the organization of the classification. This category is sub-divided into two columns, flat or hierarchical. Flat structures usually represent classification topics (C) with a discrete linear ranking or order. A hierarchy provides the classification topics (C) with a more complex structure by grouping similar items together.

Mapping schema describes how the survey's reviewed literature (L) is mapped to classification topics (C). We introduce L to refer to a reviewed item (in most cases, the literature being reviewed). This is split into two categories, Unique-mapping and 1-N mapping. Unique-mapping schema map each reviewed item (L) once for every topic (C). This mapping schema is best for finding areas in the field with extensive or limited work, which may guide researchers to immature areas for new research possibilities. Figure **??** presents some examples of **unique mapping**.

	C_1		
C_2		L_1	
	L_n		
			L_2

(A)

	C_1			C_2		
L_1		✓		✓		
L_2			✓			✓
L_n	✓				✓	

(B)

(C)

C_1	L_1,L_4,L_5
	L_3,L_6,L_7,L_8
	L_2

Figure 2.3: Examples of classification schemes using **unique-mapping**. C refers to a classification topic and L refers to a reviewed item (in most cases, the literature reviewed). Examples (A) and (B) map L to each of C once. However, example (A) structures the table such that both classifcation topics are represented by an axis and map L to the appropriate intersection. Example (B) maps L to the Y-Axis and each classification topic C on the X-Axis. Example (C) links each of the reviewed items (L) to the appropriate classification topic in the form of a list. Examples (A) and (B) show the same information.

Kerracher et al. use a unique mapping schema to plot the design space of temporal graphs [**?**] by mapping classification topics to the x and y axis, and placing L at the intersection of the two criteria (Figure **??**). Nusrat and Kobourov present a task taxonomy for cartogram visualization that conveys how different tasks can be classified. The tasks are uniquely mapped to 4 different classification topics (Figure **??**) [**?**]. Wagner et al. present a malware visualization taxonomy and map reviewed literature directly to the appropriate classification category (Figure **??**) [**?**]. Our main taxonomy (Table **??**) also uses a unique-attribute mapping schema to map our two classification topics, the modified InfoVis pipeline and the subject-based clusters to L. The table also displays how the literature is ordered in the SoS by mapping a unique color to each L.

1-N Mapping differs from the unique-mapping schema by allowing a reviewed item (L) to be mapped up to N times for each classification topic (C) where N is the number of available attributes. Examples of N-mapping can be found in Figure **??**. Multiple-Attribute mapping matrices are most suited to comparing different elements, such as techniques or frameworks, against one another. These papers usually offer a checklist and present the criterion each paper fulfills or does not.

Borgo et al. compare different glyph-based visualization techniques using a multiple-attribute mapping matrix and identify which papers exemplify the proposed design guidelines (Figure **??**) [**?**]. Tominski et al. compare different magic lens approaches, and what data or tasks are applicable for each [**?**] using a N-mapping schema (Figure **??**).

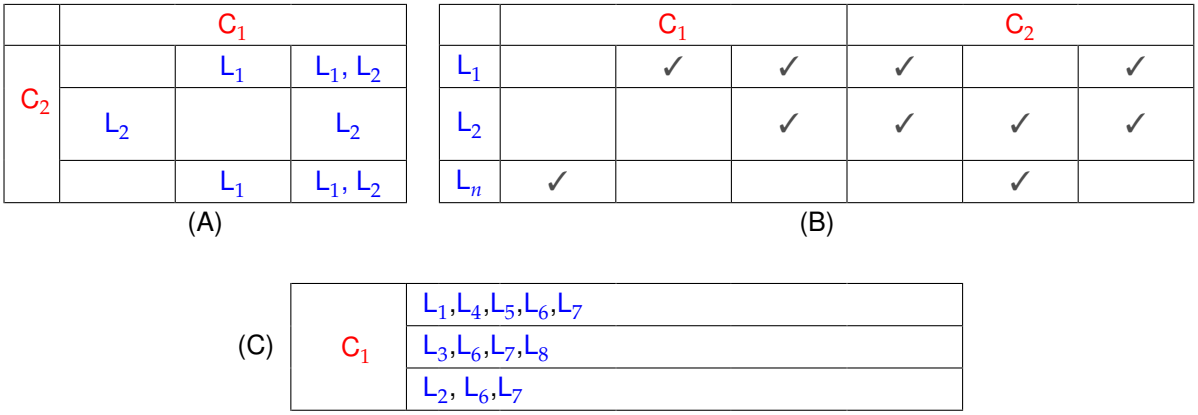


Figure 2.4: Examples of classification schemes using **1-N mapping**. C refers to a classification topic and L refers to a reviewed item.
Examples A and B can map L to each of C multiple times. Example A structures the table such that both classifcation topics are represented by the X and Y axes and map the reviewed topics at their appropriate intersection. Example B plots reviewed items to the Y-Axis and each classification topic on the X-Axis. This examples gives a clear comparison of reviewed item's (L). Example C links each of the reviewed items (L) to the appropriate classification topics in the form of a list. Examples A and B show the same information.

Some papers do not map L explicitly in their categorization and choose to display just their classification. We identify these survey papers as incorporating an **indirect mapping**. Some examples of this can be found with Sedlmair et al.'s taxonomy [**?**] which classifies data characteristics between two different classification topics, Class-Factors and Influences (Figure **??**). Another example of this is Heinrich and Weiskopf's state-of-the art report for Parallel Coordinates [**?**], which presents a hierarchical view of the important topics within the field. This representation does not explicitly show how literature fills the specified topics (Figure **??**).

The SoS aims to provide researchers with an understanding of open research areas. We use a 2D, Hierarchical, Unique-mapping table which follows the example found in Figure **??** (A). Our table is used to present a taxonomy which clearly conveys what areas are less developed in terms of survey papers (Table **??**). We use a separate table to compare different types of classifications used (Table **??**). This follows the same classification scheme but follows Figure **??** (B).

2.3 Survey Papers

This section provides a collection of summarised survey papers (see Table **??**). Each paper is broken down to present the survey's concept, their classification schema, and open areas of research discovered.

2.3.1 Data-Centric Survey Papers

The Data-Centric section contains literature that focus on types of data or data itself. Of the survey papers reviewed, two categories were identified within as subtopics which include data-type papers that emphasize the type of data surveys and papers that focus on text.

Data-Type Focused Surveys

This subsection presents a diverse collection of survey topics including visual distance measures between images, hardware and software performance, and color maps.

Sedlmair et al. present a taxonomy of visual cluster separation factors in scatter-plots, as well as a qualitative evaluation of recently proposed separation measures [**?**]. They provide a brief introduction to the area of dimension reduction, as well as their motivation behind the literature survey. The paper discusses chosen cluster separation measures: cluster identification and verification. This is followed by a section on related work before discussing their taxonomy which includes a qualitative data study. The paper presents its taxonomy on visual cluster separation factors, and evaluates them, before discussing the results of the study.

The proposed taxonomy includes four main categories: scale, point distance, shape, and position. These categories are examined during different states of observation. Within-class factors is the first state which includes variables such as density, curvature, and clumpiness. The second state is between-class factors which arise from the variance between two or more properties. For example, a variance of size between two time slices.

Sedlmair et al. suggest that the taxonomy could be extended with new dataset characteristics.

Isaacs et al. present a survey focused on reviewing hardware and software performance data as well as performance visualization techniques [**?**]. Performance data is defined as data generated to measure the effectiveness and behavior of a process. The survey develops a taxonomy to aid selection of the appropriate techniques to display performance data for both analysts and developers. They introduce the concepts behind performance visualization,

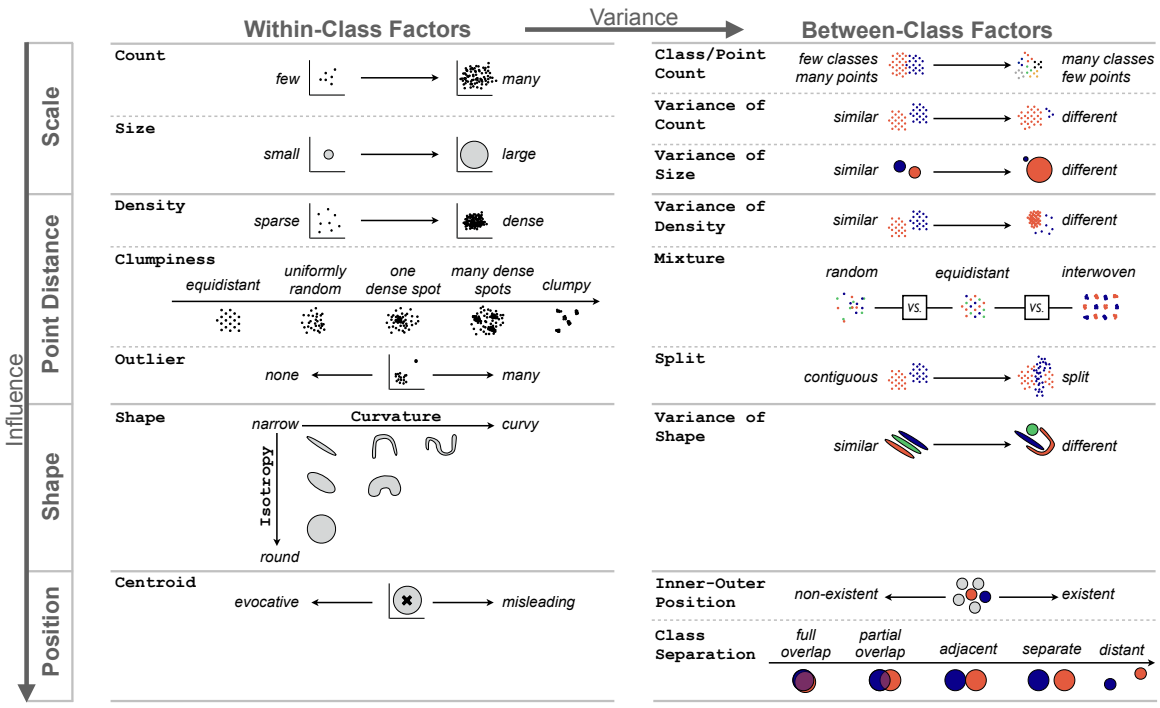


Figure 2.5: An indirect mapping taxonomy of data characteristics with respect to class separation in scatterplots. Courtesy of Sedlmair et al. [**?**]

including how it is acquired, what form it can take, and the goals of using performance data. The paper discusses the taxonomy by discussing visualization types based on the source of performance data. Hardware visualization examines performance data of hardware and aims to visualize complex hardware topology. Software visualization describes performance data of software, such as software maintenance. Task performance investigates performance of tasks. The final category, application visualization, discusses context-specific performance data. Isaacs et al. also provide an interactive literature browser related to this topic. We provide a full list of these browsers in Table **??**.

Isaacs et al. present their design space by looking at the context, scale and goal of each paper. How each paper fits in the presented taxonomy (hardware, software, task, application), scalability, what it visualizes, and what it presents are discussed (see Figure **??**).

The paper presents future challenges that are found within the landscape of performance visualization. These include, scalability, the increase in system complexity and it's result on performance visualization, comparing performance differences and ensemble datasets, integration of performance visualization across each taxonomy point, and the creation of performance visualization that facilitates software development and debugging.

		Taxonomy				Demonstrated Scale*		Global Compre.		Problem Detection			Diagnosis/ Attribution	
Visualization Techniques	Papers	H	S	T	A	Data	Parallel	Program Structure	Resource Usage	Anomalies	Bottlenecks / Imbalance	Resource Misuse	Software	System
Radial Tree	Bhatele et al. [BGI*12]			X		NR	10^4 processes	X	X		X			
Node-Link Graph	Boxfish [LLBT12, ILGT12]	X				NR	10^4 nodes		X		X			X
Radial Tree, Animation	Choudhury and Rosen [CR11]	X	X			10^7 transactions	N/A	X	X		X	X	X	
Layered Node-Link	DOTS [BKS05]	X		X		NR	NR	X	X		X	X	X	
Clustered Node-Link, Animation	Frishman et al. [FT05]	X		X		NR	10^2 objects	X	X	X				
Node-Link Graph	Heapvis [AKGT10]		X			10^3 nodes		X	X				X	
Radial Tree	Kim et al. [KLJ07]		X	X		NR	10^3	X		X			X	
Node-Link Trees, Indented trees	Lin et al. [LTOB10]		X			NR	NR	X	X				X	
Node-Link trees	DeRose et al. [DHJ07]		X	X		NR	10^2 cores				X		X	
Node-Link graph, Animation	Sambasivan et al. [SSMG13]		X			NR	NR	X					X	
Radial Tree	Sigovan et al. [SMMT13a]	X				10^1 resources	10^3 processes		X	X		X		
Node-Link trees	STAT [AdSLT09]		X	X		NR	10^5 tasks	X			X		X	
Clustered Node-Link, Animation/Real Time	Streamsight [DPA09]		X	X		streaming	10^3 tasks	X	X	X	X			
Layered None-Link	Threadscope [WT10]	X		X		10^3 events	10^1 threads	X	X	X	X			
Node-Link Graph, Treemap	Weidendorfer et al. [WKT04]		X			NR	1	X					X	
Timeline, Stacked Graph, Small Multiples	de Pauw et al. [DPWB13]	X		X		streaming	10^3 tasks	X	X		X		X	
Shared Timeline	Muelder et al. [MGM09]			X		NR	10^4 processes	X			X			
Gantt Charts, Timeline, Matrix, Scatterplot	Muelder et al. [MSMT11]	X	X	X		NR	10^3 cores	X	X		X			
3D Parallel Gantt Chart, Treemap/Force-directed layouts	Triva [SHN10]	X		X		NR	10^3 processes	X	X		X			
Parallel Gantt Chart, Node-Link Tree, Bar Charts	Zinsight [DPH10]		X	X		10^5 events	10^2 processes	X	X		X		X	
1D Color-Coded Array, Histograms	Cheadle and Field [CFAT06]	X	X			10^1 memory groups	N/A	X					X	X
1D Color-Coded Array Stacked By Time	Moreta and Telea [MT07]		X			10^5 allocations	N/A	X				X	X	
Edge Bundling, Gantt Charts, Hierarchies	Extravis [CHZT07]		X			10^5 events	NR	X					X	
Parallel Gantt Chart, Indented Trees, Code view	HPCToolkit [ABF*10, TMCF*11, LMC13]	X	X	X		10^1 gigabytes	10^4 processes	X	X		X	X	X	X
Stacked Barcharts, Stacked Timelines	Lumière [BBH08]	X	X	X		10^6 decisions	NR	X	X	X	X		X	
Parallel Gantt Chart, Small multiples, Plots, Ensemble	Projections [KZKL06, LMK08]		X	X		gigabytes	10^4 processes	X	X		X	X	X	
Stacked Barcharts, Scatterplot, Histograms, Code Coloring	TraceVis [RZ05]	X	X			10^7 instructions	NR	X	X		X	X	X	
Icicle Timelines, Coordinated views	Trumper et al. [TBD10]		X	X		10^4 events	10^1 threads	X			X	X	X	
Parallel Gantt Chart, Icicle Timeline, Adjacency, Indented Trees, Ensemble Timeline, Plots	Vampir [NAW*96, BW12, ISC*12, VMa13]	X	X	X		terabytes	10^5 processes	X	X		X			
Abstract Diagram	Choudhury et al. [CPP]	X	X			10^1 buffers	N/A				X	X	X	
Dot Plot, Bar Charts	Iviz [WYH10]		X	X		10^6 events	2 jobs	X			X		X	
Scriptable	ParaProf [SML*12]			X		NR	10^4 processes		X		X			
Indented Trees, Matrix	Scalasca [GWW*10, WG11]		X		X	terabytes	10^5 cores	X	X		X		X	
Color-coded 2D matrix, histograms, 3D graph layout	Schulz et al. [SLBT11]	X		X	X	NR	10^4 cores		X	X	X	X		
Bubble Chart, Animation	Sigovan et al. [SMM13]			X		NR	10^4	X	X		X			
City Metaphor	SynchroVis [WWF*13]		X	X		10^2 objects	10^1 threads	X		X			X	
Icicle Timeline, Bundles	SyncTrace [KTD13]		X	X		10^7 events	10^2 threads	X	X		X	X	X	
Sunburst, Matrix, Dendrogram	Trevis [AH10]		X			10^3 nodes	NR	X					X	

Figure 2.6: Isaacs et al. present a 1-N design space that classifies literature based on the context, scale and goal of each paper. Image courtesy of Isaacs et al. [**?**]

Interactive Literature Browsers	URLs
Cartogram Visualization	[**?**]
Dynamic Graph Visualization	[**?**]
Financial Visualization	[**?**]
High-Dimensional Visualization	[**?**]
Matrix Reordering	[**?**]
Performance Visualization	[**?**]
Scientific Literature & Patents Vis	[**?**]
Set Visualization	[**?**]
Software Reuse Tasks	[**?**]
SoS Literature Browser	[**?**]
Space-Time Cube Visualization	[**?**]
Text Visualization	[**?**]
Time Visualization	[**?**]
Visualizing Group Structures in Graphs	[**?**]

Table 2.5: The tables provides an overview of interactive literature browsers found during the literature search. Sorted in alphabetical order

Zhou and Hansen present a comprehensive review of color-map generation techniques, and provide a reference for readers who are faced with color mapping decisions [**?**]. The paper aims to provide a comprehensive overview of various color-map generation techniques. The paper also presents a hierarchical taxonomy of color-mapping literature to guide readers when choosing appropriate literature to review, and classifies representative visualization techniques that discuss usage of color-maps.

The hierarchical classification divides papers into various different color-map generation techniques based on the type of data used in the form of a flow chart. The taxonomy subdivides the data into either discrete or continuous. The discrete data type is split into either nominal or ordinal data types, another division of data comprehension. Both ordinal data and continuous data are linked to color-map transformation literature. The taxonomy also takes into consideration the type of paper and examples.

Zhou and Hansen recommend more research into multivariate or high-dimensional color-mapping as a future research direction. Further research into aesthetically pleasing color maps that still provide insight into data, and the use of meta-data to optimize color-mapping suggestions are also potential future research directions in the field.

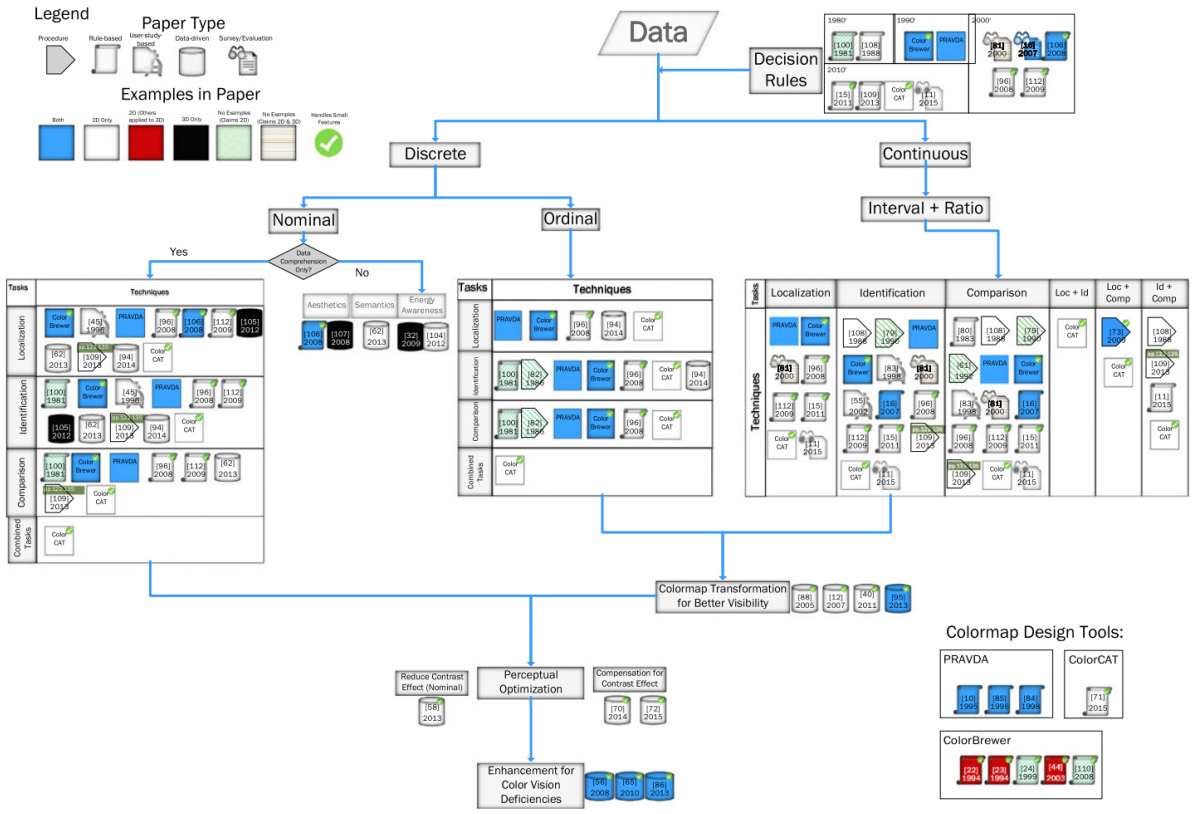


Figure 2.7: The hierarchical taxonomy presents literature by looking at the type of data mapped, and the task of the visualization. The taxonomy uses icons to present the paper-type, and color to signify which examples are present within. Courtesy of Zhou and Hansen [**?**] .

Text-Focused Surveys

The text visualization literature is evolving rapidly. This section includes surveys focused on understanding and visualizing text. The section contains four survey papers. The first paper analyses how different text sources are used in visualization with event detection methods. The second focuses on advancement in the field of close and distant reading. The third presents a classification of text visualization techniques. The last survey focuses on the visualization of 'scientific literature and patent' text sources. Table **??** cites a number of other text surveys as well.

With the growth of social media and micro-blogging, Wanner et al. take the opportunity to explore the use of analytical processes on real-time data and divulge key event-detection approaches that can be used for textual data [**?**]. They review the different available data sources when working the text-data streams including news, email, micro-blogging, research papers,

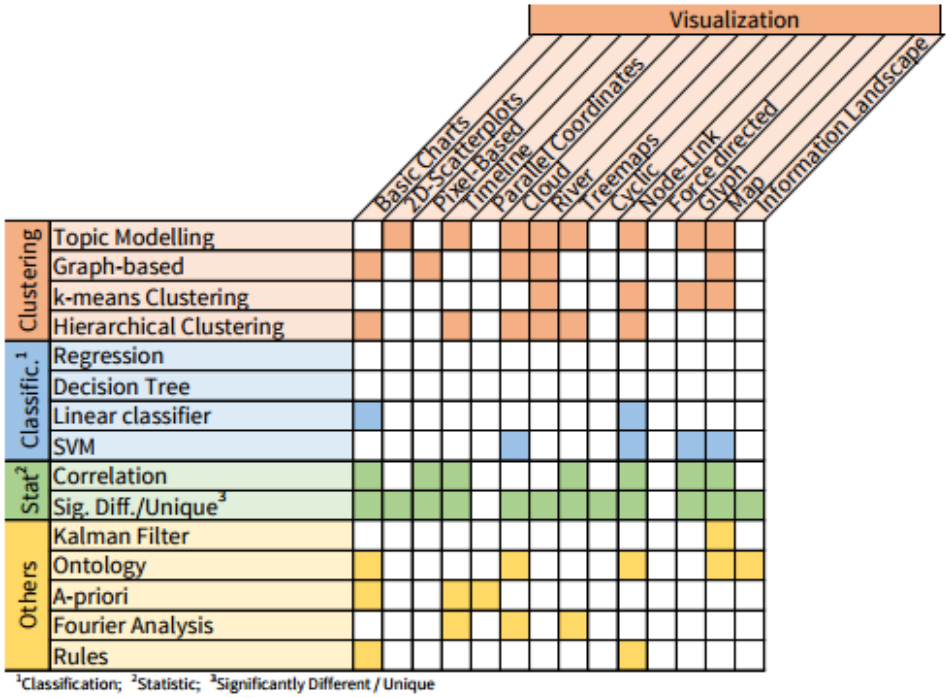


Figure 2.8: This 1-N table identifies event detection techniques and displays the the use of visualization with each. The table reveals that clustering techniques are often represented using the river metaphor. Image courtesy of Wanner et al. [**?**]

and metadata. They discuss methods that are used to process text. The study examines the event detection methods that are grouped including: clustering techniques, classification-based, statistic based, and miscellaneous techniques such as the Kalman Filter and Fourier Analysis. They finish by discussing evaluation methods that can be applied to text-data.

Wanner et al. classify 51 papers across their survey. They classify these in various ways including: data source, text-processing methods, automatic event detection methods, visualization methods, and tasks supported. The main taxonomy displays the use of automatic event detection methods on the y-axis, and then characterizes each research paper via the visualization technique along the x-axis. By doing this, they can investigate the correlation between these two fields. (Shown in Figure **??**).

The reviewed papers had little focus on discussion forums. Demand for more sophisticated techniques such as topic modeling is something that will become more apparent in the future, and event detection algorithms seem to exclude important information for deciding whether items are newsworthy or not. This could be considered new research in this area.

Jänicke et al. present recent advancements in the field of visualizations that support close and distant reading of textual data [**?**]. Nancy Boyles defines close reading as "reading to

uncover layers of meaning that lead to deep comprehension" [**?**], whilst Moretti describes distant reading with the statement, "a little pact with the devil: we know how to read texts, now let's learn how not to read them" [**?**]. The paper focuses on quantitative literary text analysis using statistical analysis methods for visual analytics and visualization. Literature in the digital humanities is also covered. These are categorized using a taxonomy for applied methods [**?**]. The paper looks at different types of techniques, such as color mapping or heat-maps, for close-reading analysis, distant-reading analysis, and combinations of both.

Jänicke et al. classify papers based on the type of reading analysis provided, which is broken down for each paper. This is compared to the method of analysis, which includes single text analysis, parallel-text analysis, and corpus analysis.

They provide a large collection of areas for future research which includes novel techniques for close reading, visualizing transposition in parallel texts, geospatial and temporal uncertainty, usability studies, and the development of design guidelines for scholars. All these areas are ripe with unsolved problems in the field. Jänicke et al. published an extended version of this survey [**?**].

Text visualization is becoming a more mature field within information visualization and Kucher and Kerren aim to classify the literature using an interactive browser [**?**]. See also Table **??**. They examine the field of text visualization and closely related surveys. Kucher and Kerren then present the taxonomy they have created to look at different areas within text visualization and how they can be subdivided (the figure is provided in the supplementary material).

Kucher and Kerren categorize text visualization techniques into 5 main categories: Analytic Tasks, Visualization Tasks, Domain, Data, and Item Visualization.

Federico et al. present a survey that addresses literature papers with an emphasis on visual approaches for scientific literature and patents. Literature is classified looking at the data-type and the fulfilled task of each [**?**]. The paper identifies four types of data found within scientific literature papers and patents. These include text, citations, authors, and meta-data. The data types create the main structure of the paper, with each data-type divided to explore how tasks are analyzed for each category. The paper also provides a breakdown of literature that handles multiple data types, which focus on different tasks.

Their indirect classification uses two tables. The first table displays the total number of publications that match the criteria at the axis intersections, which examine the four data-types and different tasks such as lookups, relation seeking, and patterns. The second table is similar but records how multiple data-types are mapped to a new set of tasks. This includes aggregation, labeling, composition, tight integration, and multiple views. The table shows that most literature in the field attempts to analyze patterns within text.

			Close Reading					Distant Reading						
			Plain	Color	Font size	Glyphs	Connections	Structure	Heat maps	Tag clouds	Maps	Timelines	Graphs	Miscellaneous
Single Text Analysis	enhanced text views	[Pie10], [CGM*12], [Pie13], [GWFI14]				x								
		[PSA*06], [CTA*13], [Ben14], [BJ14]		x										
		[ARLC*13]				x	x							
		[WMN*14]		x	x									
	both	[VCPK09], [BGHJ*14], [KJW*14]		x				x		x				
		[WJ13b], [CMLM14], [KZ14]		x			x							
		[Cay05]	x					x						
		[CDP*07]		x					x					
		[WV08]		x									x	
		[MFM13]	x										x	
		[RSDCD*13]	x											x
	abstract text views	[KO07], [FS11], [CTA*13], [OKK13], [Ben14]							x					
		[Pie05]						x						
		[PBD14]												x
Parallel Text Analysis	section alignments	[WH11], [HKTK14]		x										
		[Cor13], [WJ13b]		x			x							
		[JRS*09]		x				x						
		[GCL*13]		x					x					
		[JGBS14b]		x			x		x					x
	sentence alignments	[BGHE10]		x			x							
		[JGBS14a]			x		x							
Corpus Analysis	statistics for textual entities	[Bea08], [Bea11], [Bea12], [BJ14]								x				
		[WJ13a], [HCC14]												x
		[CWG11]		x	x				x					
		[Mur11]		x					x					
		[FKT14]	x						x	x				
	relationships between texts	[EX10], [Gal11], [WH11], [Joc12], [CEJ*14], [Ede14]											x	
		[RRRG05]							x					
		[OST*10]		x										x
		[Wol13]		x									x	
	relationships between textual entities	[RRRG05], [AGL*07], [vHWV09], [KKL*11], [MLSU13], [WJ13a], [Arm14]											x	
		[GZ12], [RFH14]		x									x	
		[MH13]		x					x				x	
		[AKV*14]		x					x					
	social networks	[Cob05], [CSV08], [BDF*10], [RD10], [BHW11], [Kle12], [Boo13], [KOTM13], [Tót13], [Pet14]											x	
		[KLB14]	x										x	
	space and time	[JHSS12], [JW13], [DNCM14], [GDMF*14], [ÓML14]									x	x		
		[Wea08]							x		x	x		
		[BPBI10]		x							x	x		
		[DWS*12]	x							x	x	x		
		[HACQ14]		x						x	x	x		
	space	[MBL*06], [DFM*08], [Tra09], [GH11b], [EJ14]									x			
	time	[KBK11], [ARR*12], [LWW*13]										x		
		[CLT*11], [CLWW14]								x		x		
		[HSC08]	x									x	x	
		[DWS*12]	x							x	x	x		
		[ESK14]								x		x		x
		[HPR14]		x								x		

Figure 2.9: A 1-N Taxonomy by Jänicke et al. to map reading techniqes found within different analysis methods. Image courtesy of Janicke et al. [**?**]

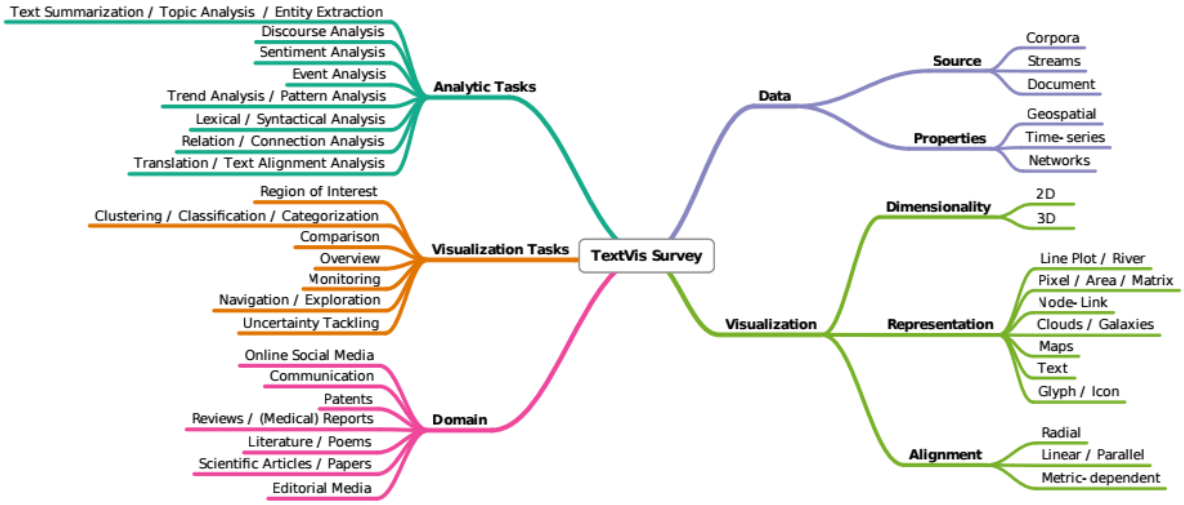


Figure 2.10: Kucher et al. present a hierarchical taxonomy used to classify text visualization techniques [**?**] .

	Elem. lookup & comparison	Elem. relation seeking	Synoptic (Patterns)	Synoptic (Temporal patterns)
Text	8	7	20	5 (6)
Citations	2	7	9	10
Authors	2	1	2	7
Metadata	2	5	8	2 (3)

a Approaches for data types: *Text*, *Citations*, *Authors*, and *Metadata* (supporting *Elementary* and *Descriptive Synoptic* tasks).

	Aggregation	Labelling	Composition	Multiple views	Tight integration
Multiple/Connectional	11 (6)	4 (3)	3 (1)	12 (3)	5 (2)

b Approaches for data type: *Multiple* (supporting *Connection Discovery* tasks), by level of integration.

Figure 2.11: Table *a* presents the distribution of papers for each single data category, whilst *b* contains the distribution papers which look at multiple data-types. Both tables distribute papers based on tasks. Numbers in parentheses are papers identified as a secondary classification. Image courtesy of Federico et al. [**?**] .

Federico et al. break down future research directions for each data-type. Research on text data identifies contextual identification in compact space. Research into author data suggests uncertainty analysis of ambiguities with synonyms and homonyms. Limited work is provided with citation data and meta-data, with citation data focusing on citation count and meta-data ignoring many pieces of gathered data, which narrows the fields significantly. Some other broader examples include quantitative and qualitative evaluations, scalability, user interaction, and research into user-tasks.

2.3.2 Multivariate & Hierarchical

This category discusses multivariate, high-dimensional, and hierarchical visualization. These are grouped together due to their association within large datasets. The reviewed content on this subject can be broken down to look at hierarchical visualization, an overview of high-dimensional visualizations, parallel coordinate plots, and glyph-based visualization.

Hierarchical Surveys

This section includes surveys that have an emphasis on hierarchical structures. The first survey focuses on the classification of hierarchical aggregation strategies for visualization. The second survey provides a design space of implicit hierarchy visualization to compare literature in the field. The final survey looks at set-typed data and how set-typed data visualizations relate to different tasks.

Elmqvist and Fekete review the use of hierarchical aggregation within Information Visualization. Hierarchical Aggregation is based on iteratively building a tree of aggregate items. Elmqvist and Fekete use this review to present a model that enables augmentation of existing techniques with multiscale functionality [**?**]. They first describe related reading before discussing hierarchical aggregation and presenting various related techniques (scatter-plots, parallel coordinates, etc). The paper then presents examples of hierarchical aggregation within visualization before presenting their classification and guidelines (the figure is provided in the supplementary material). Elmqvist and Fekete end by describing design guidelines for hierarchical aggregation.

Elmqvist and Fekete derive a classification of visual aggregation strategies. The table looks at the data-structure used, visualization mapping type, type of visualization, type of aggregation, the visual aggregate, and what is being visualized.

Elmqvist and Fekete discuss future work which includes model refinements, as well as investigating the trade-off between accuracy and usability. Finally, they look at the idea of reviewing different hierarchical structures such as Directed Acyclic Graphs (DAGs).

Schulz et al. construct a survey categorizing the design space of techniques for hierarchy visualization, with the aim of guiding researchers to unexplored research areas within the field [**?**]. Examples of techniques for implicit hierarchy visualization include spatial dimensions and node representation. Schulz et al. present their aims, before presenting the design space for hierarchy visualization (see Figure **??**). They follow this by discussing some of the limitations of the design space such as techniques with *mixed Treemaps*. Using the design space, they

Data structure	Visualization	Type	Aggregation	Visual aggregate	Metadata visualized
multidimensional	scatterplot	O/L	hierarchical clustering	points [19, 74]	average
multidimensional	scatterplot	O/L	hierarchical clustering	boxes [82]	extents (axis-aligned), average
multidimensional	scatterplot	O/L	space-filling subdivision	boxes [80]	extents (axis-aligned), average
multidimensional	scatterplot	O/L	hierarchical clustering	hulls [4]	extents (convex hull), average
multidimensional	scatterplot	O/L	hierarchical clustering	blobs [6, 15, 20, 44]	extents
multidimensional	parallel coordinates	O/L	hierarchical clustering	lines [75]	average
multidimensional	parallel coordinates	O/L	hierarchical clustering	bands [34, 82]	extents, average
multidimensional	parallel coordinates	O/L	hierarchical clustering	color histograms [29, 30]	distribution, extents
multidimensional	parallel coordinates	O/L	hierarchical clustering	beads [4]	distribution, extents
multidimensional	starglyphs	O/L	hierarchical clustering	lines [75]	average
multidimensional	starglyphs	O/L	hierarchical clustering	bands [34, 82]	extents, average
multidimensional	starglyphs	O/L	hierarchical clustering	color histograms [29, 30]	distribution, extents, average
tree	treemap	S/F	existing tree hierarchy	treemap nodes [63]	extents, average
tree	node-link diagram	O/L	existing tree hierarchy	thumbnails [17, 59]	extents, count, depth
graph	node-link diagram	O/L	hierarchical clustering	metanodes [2, 8]	extents, average
graph	node-link diagram	O/L	—	edge bundles [47]	link extents, average
graph	node-link diagram	O/L	data cube aggregation	metanodes [78]	node and link counts
graph	adjacency matrix	S/F	recursive edge merging	edge blocks [1, 28]	distribution, average
spatial	2D/3D geometric	—	recursive data merging	quad/octree blocks [4]	extents, average

Figure 2.12: A hierarchical classification of aggregation strategies for Information Visualization techniques. Image courtesy of Elmqvist and Fekete [**?**] .

present novel techniques that are not explored with visual representations, using their own rapid visualization prototyping software.

The survey paper investigates four main classification topics. Spatial dimensionality, how nodes are represented (such as their shape), how edges are represented (Do they overlap? Are they included?), and the layout (subdivision or packing).

Schulz et al. propose that the design space presented in the paper can be used to create more surveys within the field of hierarchy visualization. New layouts can also be created using the characterization of implicit hierarchy visualization.

Visualization of sets can be a demanding task due to the wide variety of possible relations between them. Sets are defined as items that are grouped into sets based on specific properties. Alsallakh et al. present an overview of state-of-the-art techniques for visualizing different forms of set data (defined as a collection of unique objects called set elements) which can be used to select appropriate techniques for different scenarios [**?**]. They first define set-type data before looking at some common tasks. The tasks are either related to elements, element attributes, or relationships between sets. The paper then provides examples of different ways to visualize set-typed data such as using euler diagrams, bubble-sets, pivot-paths, and scatter views.

Alsallakh et al. classify literature by constructing an overview of different tasks and techniques that are supported, partially supported, or supported with an interaction requirement. These techniques include Euler-based techniques, overlays, node-link's, matrices, aggregation, and scatter techniques (the figure is provided in the supplementary material).

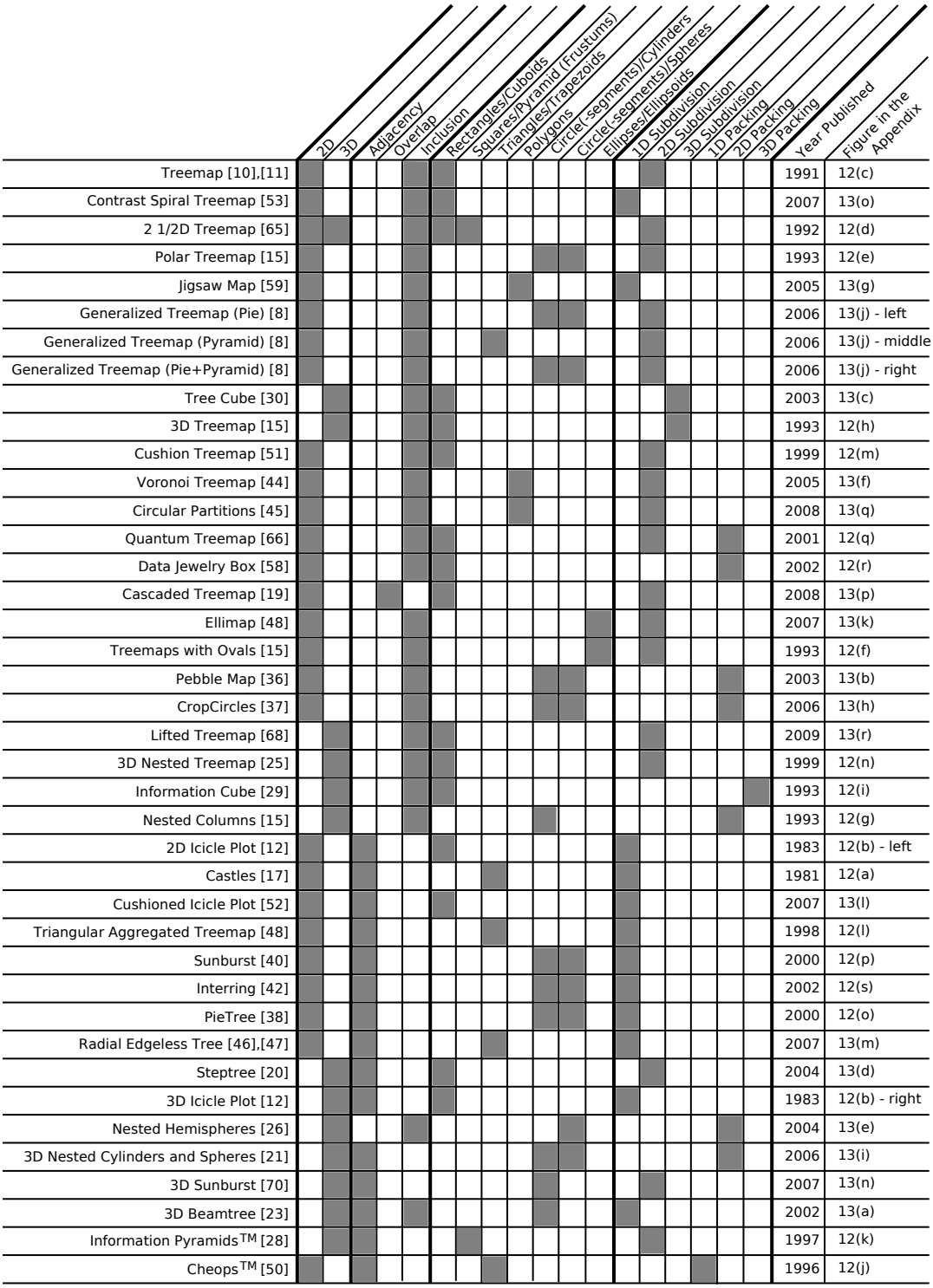


Figure 2.13: Design space for implicit hierarchy visualization created by Schulz et al. to compare techniques in the field. Image courtesy of Schulz et al. [**?**]

	Technique	Element-related Tasks						Set-related Tasks												Attribute-related Tasks				Scalability	
		A1	A2	A3	A4	A5	A6	A7	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	C1	C2	C3	C4	C5	in # of sets	in # of elements
Euler-based	Euler diagrams	●	●	●	○			○	○	●	●	●	●	○	●	●	○	●	○	○	n/a	○	○	○	○			about 10	hundreds / ∞
	ComED	●	●	●	○			●	○	●	○	○	○	●	○	○			○	○	●							10 to 20	hundreds
	DupED	●	○	○	○			●	●	●	●	●	○			○			○	○	●							about 10	tens
Overlays	BubbleSets	●	●	○	○			●	○	○	○	○	○	○	○		○	○		○	○		○	○				about 10	tens
	LineSets	●	●	○	○			●	○					○	○	○				○	○		○					10 to 100	hundreds
	Kelp diagrams	●	●	○	○			●	○	○	○	○	○	○	○	○	○		○	○	○		○	○				about 10	tens
	Colored glyphs	○	●	○	○				○				○	○		○				○	○					○		10 to 20	hundreds
	Icon lists		●					●												○			●			○	○	tens	large list
Node-link	Linked lists	🖱	🖱	🖱	🖱	🖱	🖱	🖱	●												○		●	○	○	🖱	🖱	hundreds	hundreds
	Anchored maps	○	○		○		○		●				○		○	○		○		●	🖱		🖱					20 to 50	hundreds
	PivotPaths	🖱	🖱		🖱		🖱		●					○						🖱	●		🖱					50 to 100	hundreds
Matrix	ConSet	●	●	○		🖱	🖱	🖱	●			🖱	🖱	○							🖱	🖱	●					about 100	about 100
	PixeLayer	●	🖱	🖱					●			🖱	🖱									🖱	🖱					tens	hundreds
	Frequency grids	●	●	○	○	○	○	○	○	●			●	●	●	●	○	●		●	●		🖱					3 to 5	hundreds
	Overlap matrix								●			●	●			●	●	○	○	○	●							about 100	not applicable
	KMVQL	🖱	🖱	🖱					●			○	○	○	●	●					●				●			4 to 6	not applicable
Aggregation	Mosaic displays			🖱					●			○	○	○	●	●		○		○	●				●			up to 4 sets	large (agg.)
	Double-Decker	○		●					●		🖱		●	○	●	●		●		●	●			●	●			4 to 6	large (agg)
	Sets'o'grams	🖱		🖱	●	🖱	●		●				🖱	○	🖱	🖱				○						🖱	🖱	50 to 100	large (agg.)
	Radial Sets	🖱		🖱	●	🖱	●		●			●	●	●	🖱	🖱	●	●	●	●	●	🖱		🖱	●	🖱	🖱	20 to 30	large (agg.)
Scatter	Scatter view								○										○									hundreds	not applicable
	Cluster view								○										●									hundreds	not applicable

● Task is supported

○ Task is partially supported

🖱 Task requires interaction

A1: Find/Select elements of a specific set
A2: Find sets containing a specific element
A3: Find/Select elements by set memberships
A4: Find/Select elements by their degrees
A5: Filter out elements by set memberships
A6: Filter out elements by their degrees
A7: Create a set out of certain elements

B1: Find the number of sets in a family
B2/3: Inclusion relations / hierarchies
B4/5: Exclusion / intersection relations
B6: Identify intersections between k sets
B7: identify sets involved in an overlap
B8: Identify intersections of a set
B9: Identify the set with largest / smallest number of pair-wise set intersections
B10: Analyze & compare cardinalities
B11: Analyze & compare set similarities
B12: Analyze & compare set exclusiveness
B13: Highlight specific sets, subsets, etc.
B14: create a set by set-theoretic operation

C1: Find an element's attribute values
C2: Attribute distribution in a set / subset
C3: Compare attribute values between subset
C4: Set memberships for specific attr. values
C5: Create a set of elements by attributes

Figure 2.14: A 1-N taxonomy of set-types data showing a comparision between tasks and techniques. Courtesy of Alsallakh et al. [**?**]

The paper presents an abundance of future research in this area. These include scalability within set-typed data, re-ordering sets to reveal clusters, a user study on the effectiveness of techniques, visualizing uncertainty, temporal set-typed data, generating euler diagrams with specific properties, visualizing set in context of other data types, and comparing multiple set families. There is also a discussion of improvements with coordinated multiple views and matrix-based representations of set-type data.

High-Dimensional Surveys

The high-dimensional section focuses on surveys that provide a broad overview of the field of high-dimensional visualization. "High-Dimensional" is defined as ‘any data set with a dimensionality that is too high to easily extract meaningful relations across the whole set of dimensions’ by Bertini et al. [**?**]. The section covers two surveys papers. The first paper reviews quality metrics for high-dimensional visualization. The second survey discusses the recent advancements for visualization of high-dimensional data.

Bertini et al. review techniques that use quality metrics, defined as *"a metric calculated at any stage of the information visualization pipeline, that captures properties useful for the extraction of meaningful information about the data"*, to find meaningful results for the exploration of high-dimensional data. By analyzing literature related to the topic, Bertini et al. provide a systematization of approaches that use quality metrics in high-dimensional data. [**?**]. They begin by describing the background and search methodology used within the survey. The paper presents the quality metrics pipeline, a modified version of Card et al.’s information visualization pipeline [**?**], which we also use, before discussing their systematic analysis of the literature. Bertini et al. then give examples of quality metrics factors such as what is measured, where, and its purpose.

The survey classifies each work of literature by examining the visualization technique, the metrics measured which range from clusters to image quality, where the measurements take place, the purpose of the metrics, and any interaction techniques available.

Four main areas for future research are highlighted in the paper. Evaluation is poorly represented within the field with a small representation of the papers using a real-world setting and data. This could pave a path for future research. Perceptual Tuning is another area with weak focus, as well as scalability of data representation. Metrics systematization is the final area for directed future research. There are a wealth of different metrics which makes them very hard to compare and without guidance could end up as redundant metrics usage. Composite visualization of quality metrics is also discussed.

Figure 2.15: A 1-N classification created to systemise quality metrics factors for high-dimensional data. Courtesy of Bertini et al. [**?**]

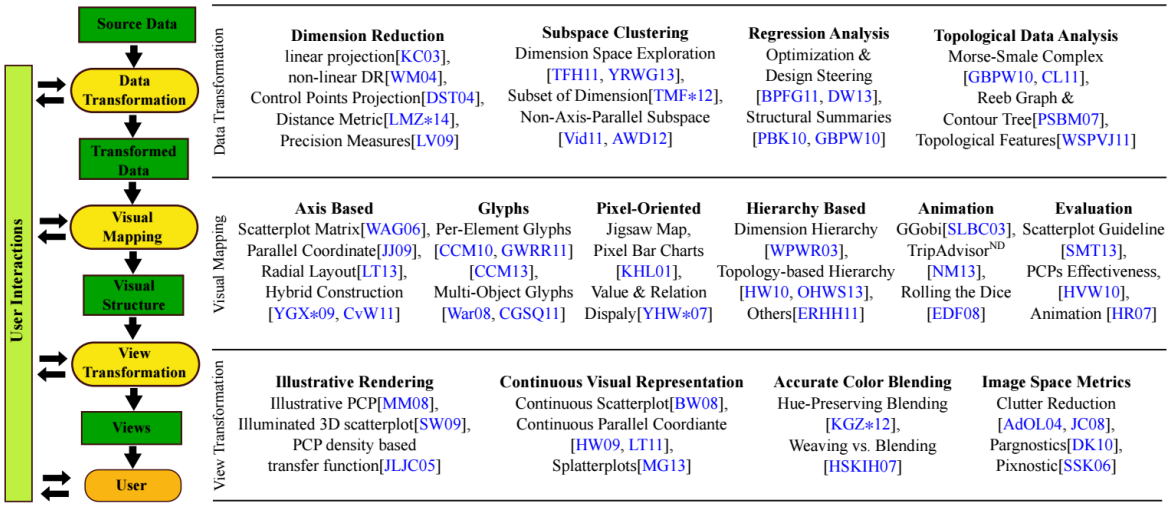


Figure 2.16: A 2D classification designed using the information visualization pipeline for the taxonomy of high dimensional data. Courtesy of Liu et al. [**?**]

Liu et al. provide a comprehensive survey focused on the advances of high-dimensional visualization techniques between 2000 and 2014 [**?**]. The study aims to help practitioners understand the recent advances in this area with the hope of inspiring the creation of new visualizations and the comprehension of future opportunities with high-dimensional data. They first discuss their classification and methodology before discussing various sections or the information visualization pipeline. Similar to ours, this includes data transformation, visual mapping, and view transformation.

The taxonomy of the survey is based on the information visualization pipeline (the figure is provided in the supplementary material). The classification clearly represents when techniques are to be used within the pipeline, as well as some action-driven classification signifying the time of use for each technique. Some examples of this are histograms, jigsaw maps, and glyphs. Liu et al. have also created custom action-driven classifications to further improve their taxonomy.

Liu et al. take great care to expose multiple directions for future research. Subspace clustering is an important technique used when visualizing high-dimensional data, and exploring non-axis-aligned methods may lead to new view-selection techniques. Understanding uncertainty within data is also an important research area. As the amount of data increases, so does the understanding of the quality of data and the best way to visualize this high-dimensional data will become more apparent. Liu et al. also discuss model manipulation, topological data analysis, as well as problems occurring with multivariate volume visualization and machine learning with a link to high-dimensional data visualization.

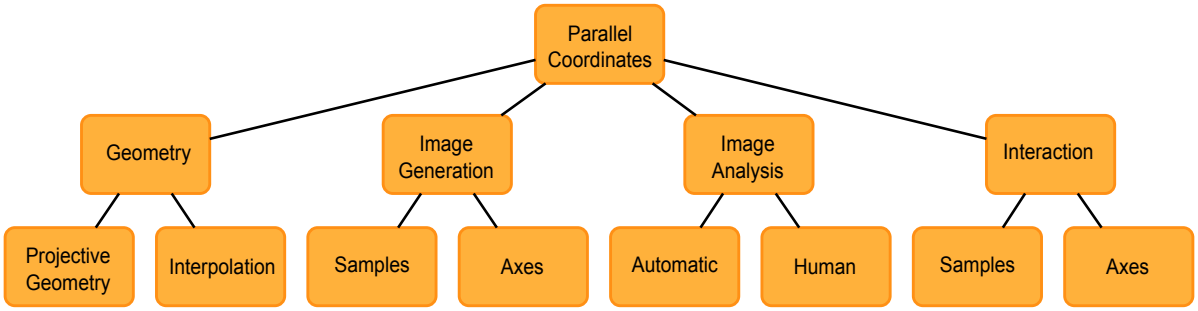


Figure 2.17: An indirect hierarchical taxonomy of important topics withn Parallel Coordinate creation. Created by Heinrich and Weiskopf, the taxonomy reflects the different sections of their survey. Courtesy of Heinrich and Weiskopf [**?**] .

Parallel Coordinates Surveys

Parallel Coordinates are a technique used to visualize multivariate and high-dimensional data. Parallel coordinates are constructed by placing axes in parallel, the choice of layout depends on the number of axes, and the range of data [**?**]. Included here are two survey papers focusing on Parallel Coordinates. One focuses on the design and structure of Parallel Coordinates and the second collects the findings of user-evaluation studies, comparing the use of a standard 2D parallel coordinate plot with other variations.

Heinrich and Weiskopf present the state of the art in visualization techniques for parallel coordinates, which is well-known for exploratory data analysis [**?**]. The paper describes the parallel coordinate plane, and reviews different variations of the methodology, with the hope of directing research in new areas related to the topic. The paper aims to provide (1) a taxonomy of techniques related to parallel coordinates, (2) a review of challenges in the field, (3) a reference to important literature in the domain, and (4) a guide to the use of parallel coordinates for applications.

Heinrich and Weiskopf split their hierarchical taxonomy into four main topics: Geometry which focuses on the coordinate system creation; Image Generation maps data to the coordinate system; Image Analysis highlights visual perception of the mapped parallel coordinate; and Interaction emphasizes manipulation of the parallel coordinates (see Figure **??**).

They state that an evaluation of existing tools would be required to identify issues in the implementation of parallel coordinate techniques. Additionally, more in-depth studies can test each category in order to find underrepresented research areas. They avoid evaluating techniques based on their applicability, correctness, usability, and performance. These are considered areas for future work within the field.

Johansson and Forsell provide a thorough literature review survey studying user-centered evaluations that investigate use and usability issues in the field of parallel coordinates. The paper aims to address issues within the domain and provide a set of guidelines for future research [**?**]. They first divide the 23 papers identified into 4 categories: evaluating axis layouts, comparing clutter reduction methods, showing practical applicability of different parallel coordinates, and comparing parallel coordinates with other analysis techniques. These categories are discussed in detail with reference to the evaluated papers. The categories include cluster analysis, correlation analysis, outlier search, value retrieval, pattern detection, and line tracing.

The paper provides a classification of how each paper compares with standard 2D Parallel Coordinates (2DPC). The table (provided in the supplementary material) maps the selection of aspects evaluated. The third dimension uses color to indicate how the advanced feature compares to a standard 2D Parallel Coordinates: Yellow identifies no significance variation in the results, Green signifies the extension is reviewed to be better than 2DPC, and red signifies the reviewed item compares unfavorably with 2DPC.

Johansson and Forsell provide a plethora of different research directions in the field. The paper proposes that existing axis configurations in parallel coordinate plots have not been thoroughly studied, with an emphasis on 3D parallel coordinates. They discuss the lack of conclusive clutter reduction comparisons, and propose research in evaluating clutter reduction for complex patterns in parallel coordinates. Some other areas for future research discussed include longitudinal studies, temporal views in 3D parallel coordinates, and the aesthetic design of parallel coordinates.

Glyph Surveys

A glyph is defined as *'a small independent visual object that depicts attributes of a data record'* [**?**]. Glyphs use visual primitives to represent different attributes in multivariate data. This section contains two literature surveys. The first presents design guidelines for the creation of glyphs. The second paper presents a selection of glyph comparison papers to provide an evaluative perspective on glyph usage.

Borgo et al. examine the fundamental concepts and design guidelines of glyph-based visualization, and how current implementation techniques adhere to them [**?**]. They first discuss the concepts and history of glyph usage. The paper covers the design and usage guidelines for glyphs, data mapping, shape design, glyph appearance, glyph placement, rendering, and glyph interaction. Borgo et al. then discuss the application of glyph-based visualization in different visualization scenarios.

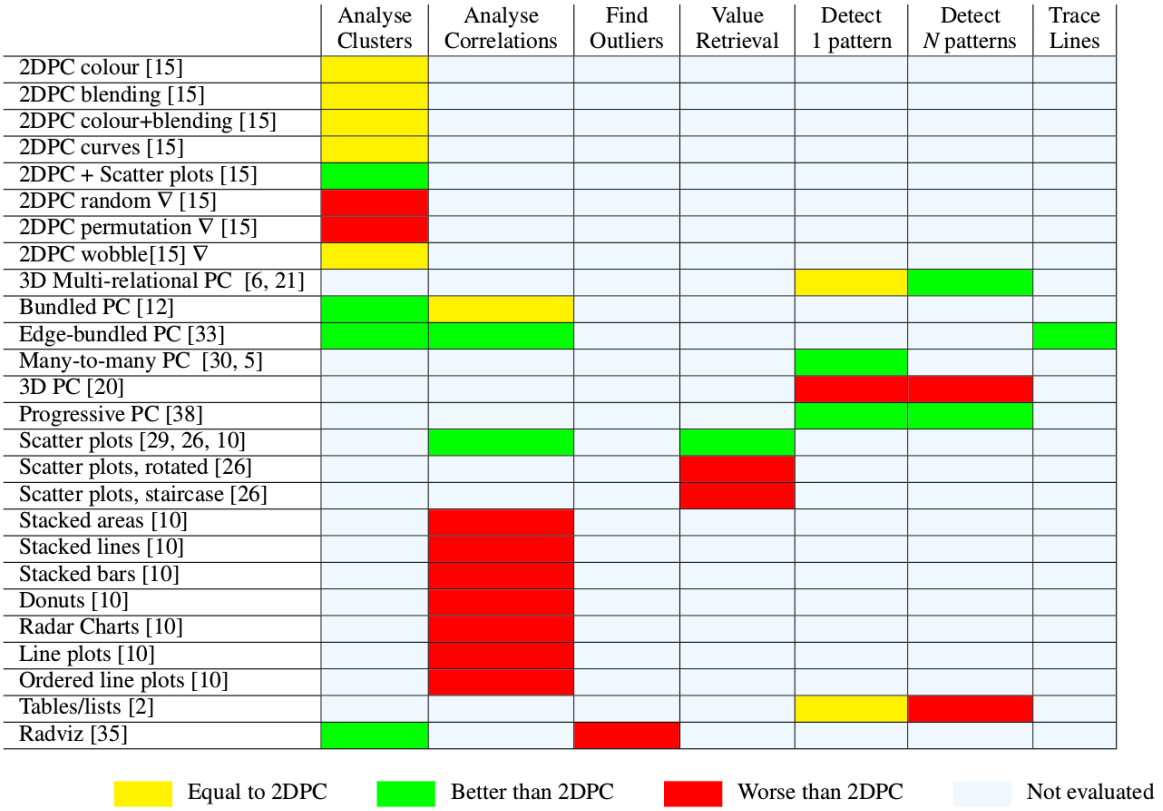


Figure 2.18: A 1-N classification of 26 techniques performed in relation to standard 2D parallel coordinates. Yellow colour indicates no significant difference in performance. Green colour means that the technique outperforms 2DPC for the specific task. Red colour shows the technique performs worse than 2DPC. Light blue colour reveals no evaluation has been found in the literature. ∇ denotes that the technique is based on animation. Courtesy of Johansson and Forsell [**?**] .

Borgo et al. categorize the literature by examining each technique and determining whether they fulfill the design guideline criteria. There are 13 tasks created that the papers are categorized according to, which range from complexity and prioritization to design and balance. The papers are also categorized according to the focus on different visual channels including color, shape, size, orientation, texture, and opacity (see Figure **??**).

Borgo et al. suggest the need for a common framework to direct more energy to the field. They also suggest evaluating extended data dimensions using glyph representations.

Fuchs et al. aim to assist researchers in gaining a stronger understanding of user-studies within the glyph design space [**?**]. Data glyphs have a wide variety of designs and uses, with many studies carried out across different glyph-types, however an overview of these studies has never been recorded. They present how the expanse of user-studies in the field compare. There are many types of data glyphs analyzed within the paper including: orientation-based,

Authors / Technique	design guideline													visual channel					
	[DG1] visualization space	[DG2] complexity vs. density	[DG3] hybrid visualizations	[DG4] perceptually uniform properties	[DG5] redundant mapping	[DG6] importance-based mapping	[DG7] view point independence	[DG8] simplicity and symmetry	[DG9] orthogonality and normalization	[DG10] intuitive / semantical mapping	[DG11] balanced glyph placement	[DG12] facilitate 3D depth perception	[DG13] interactive occlusion control	color	shape	size / height / length	orientation	texture	opacity
Brewer [Bre99]: Color use guidelines				■		■				■				■					
Cleveland & McGill [CM84]: Graphical perception	2D/3D			■					■		■			■	■	■	■	■	
Crawfis & Max [CM93]: Vector field visualization	3D	2	■			■	■	■		■	■	■		■			■		■
de Leeuw & van Wijk [dLvW93]: Local flow probe	3D	-3	■			■			■	■				■	■	■	■		
Healey & Enns [HE99]: Combining textures and colors	2.5D	1	■	■				■	■	■	■			■	■	■	■	■	
Healey et al. [HBE96]: Preattentive processing	2D		■	■				■	■		■			■	■	■	■	■	
Kindlmann & Westin [KW06]: Glyph packing	3D	2	■			■	■	■	■	■	■	■		■	■		■	■	
Kindlmann [Kin04]: Superquadric tensor glyphs	2.5D	1.5	■				■	■	■	■					■		■	■	
Kirby et al. [KML99]: Concepts from painting	2D	1	■			■		■	■	■	■			■	■	■	■	■	■
Laidlaw et al. [LAK*98]: Stochastic glyph placement	2D	2	■			■		■	■	■	■	■		■	■	■	■	■	■
Li et al. [LMvW10]: Symbol size discrimination	2D			■					■						■	■			
Lie et al. [LKH09]: Design aspects of glyph-based 3D visualization	3D	2		■	■		■	■	■	■		■	■	■	■	■	■		■
McGill et al. [MTL78]: Variations of box plots	2D	-3		■	■	■		■	■	■					■	■			
Meyer-Spradow et al. [MSSD*08]: Surface glyphs	2.5D	0	■			■	■	■	■	■	■		■	■	■	■			■
Peng et al. [PWR04]: Clutter reduction using dimension reordering	2D	1		■	■			■	■		■				■	■			
Pickett & Grinstein [PG88]: Stick figures	2D	3		■	■			■	■								■	■	
Piringer et al. [PKH04]: Depth perception in 3D scatterplots	3D								■	■		■	■	■		■			■
Rogowitz et al. [RTB96]: How not to lie with visualization	3D		■	■		■			■	■				■					■
Tominski et al. [TSWS05]: Helix glyphs on geographic maps	2.5D	-2	■		■					■	■	■		■		■			
Treinish [Tre99]: Task-specific visualization design	2.5D	-2	■	■		■			■	■	■			■	■		■	■	
Ward & Guo [WG11]: Shape space projections	2D	3		■	■			■						■	■	■			

Figure 2.19: A 1-N categorization of glyph-based approaches created by Borgo et al. In Desgin Guideline 2, -3 represents a small amount of complex glyphs with +3 displaying a large number of simple glyphs. Courtesy of Borgo et al. [**?**]

which include angular representations to represent dimensions; color saturation, which uses various saturation's to represent dimension values; positional or length glyphs, which plot dimensional data to bars or lines; and Faces, which map dimensional data to different attributes of a face. They focus on studies with measurable tasks using data glyphs. Tasks are discussed within the classification. The three main goals identified are 1) a comparison of glyph designs based on observed performance with the aim of ranking each design. 2) A comparison of variation within glyph types to detect important features within the data type. 3) The comparison of glyphs as opposed to raw data tables in order to stimulate the use of visual representation rather than textual representations [**?**]. Fuchs et al. present the results of their survey by reviewing the influence of background information or the layout, the quantity of data, data dimensionality, the influence of the task, and the effect of metaphoric glyph design.

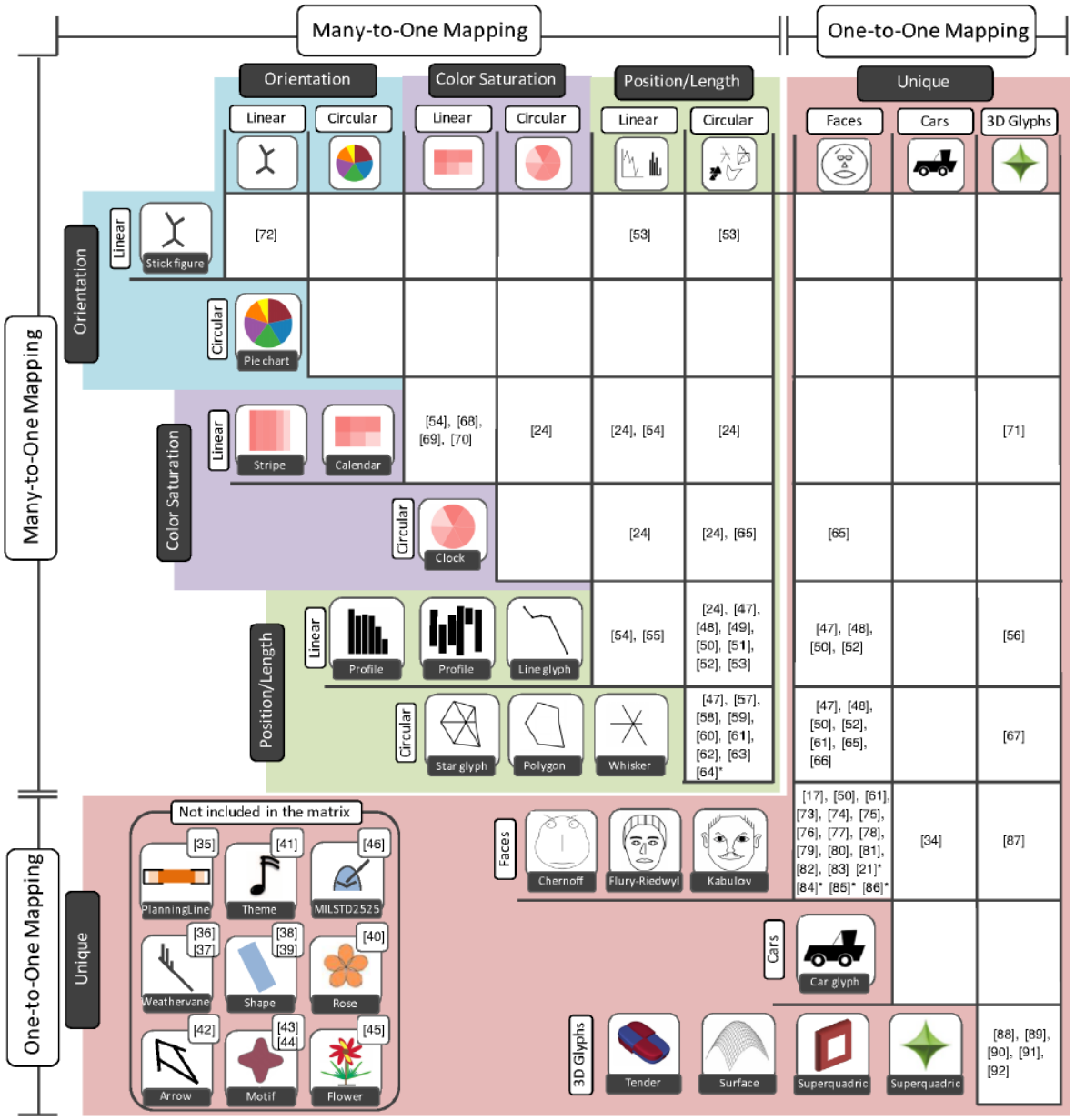


Figure 2.20: A 2-Dimensional table showing the classification of the literature in the glyph-based user-study survey. Courtesy of Fuchs et al. [**?**]

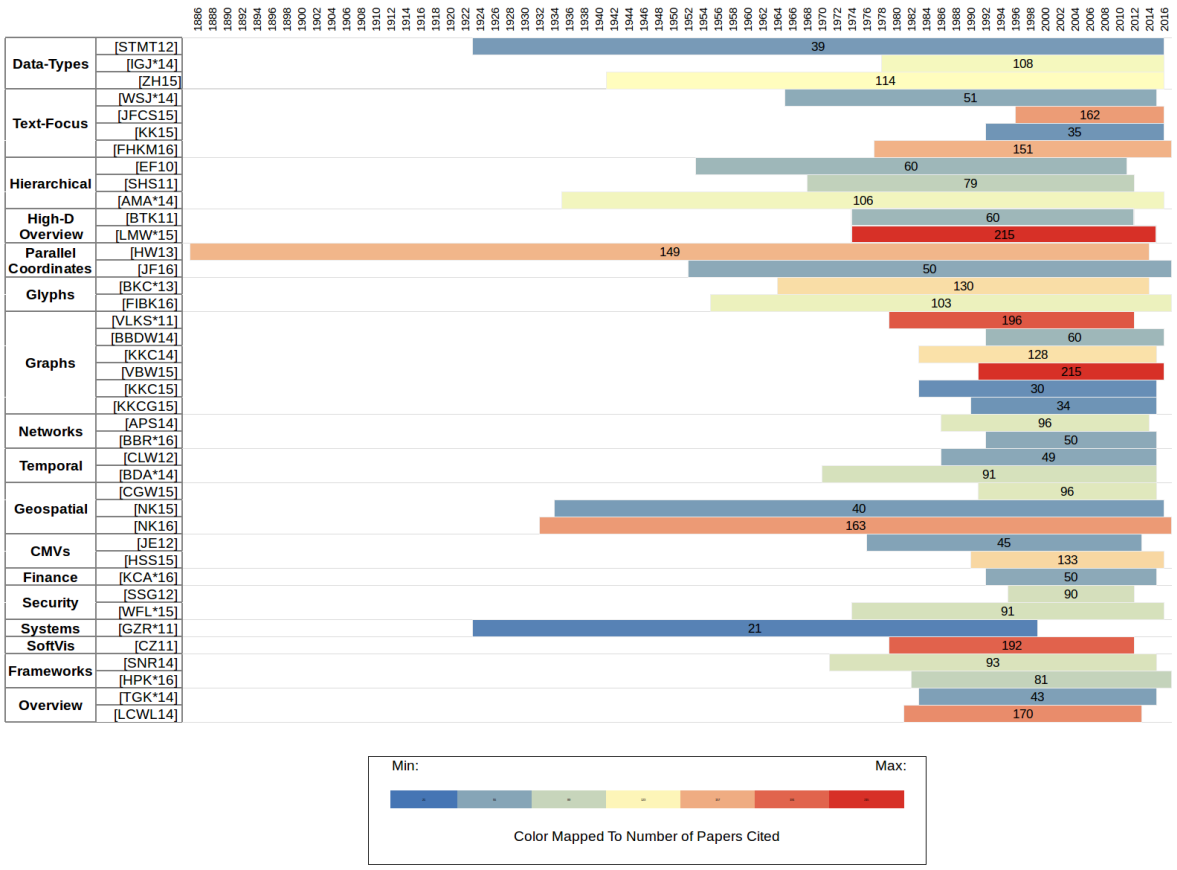


Figure 2.21: Visualization of the number of citations within each survey paper discussed, where the years spanned is mapped to the length of each bar along the x-axis. The color represents the number of papers cited within each survey.

The literature is classified according to data-to-visual primitive mapping. This classification allows Fuchs et al. to understand some comparisons with limited discussion and discuss the benefits of further work within the area (see Figure **??**).

Fuchs et al. provide many open research areas within their survey. There is a large contrast in the observations of data glyphs for quantitative data compared to observations with qualitative data. There is also little research on how data glyphs differ between synthetic and real data. Fuchs et al. find a high percentage of studies fail to look at the use of data-glyphs in exploring data and extracting information, but focus on presentation and simple output in it's stead. They propose that data glyphs can be explored in more complex layouts in order to understand further use and glyph influence in applications.

2.3.3 Graphs & Networks

There is a large body of work published on graphs and networks with over 10 surveys about the topic. Graph papers tend to have a concise range of citation years, which can be seen in Figure **??**. This figure shows the quantity and range of citations found for each survey paper.

Graph Surveys

This section includes surveys with a focus on graph papers. A graph is defined as *'a diagram showing the relation between variable quantities'* [**?**]. The section includes six survey papers: an analysis of large graphs, a classification of dynamic graphs, the visualization of grouped structures in graphs, and three papers providing an understanding of temporal graph visualization.

Von Landesberger et al. examine the visual analysis of graphs designed for large data sets. The survey reviews current techniques, with regards to the types of graph supported, and aims to present open research challenges in the field [**?**]. They give some key definitions for graphs and preprocessing techniques that are related to the survey, including different types of graphs (directed, dynamic, compound, etc.). They introduce some visual representations of these graphs, as well as how interaction in graphs is incorporated. Von Landesberger et al. finish the survey with graph analysis.

Von Landesberger et al. classify graphs according to both time-dependency and structure. Time dependency can be split into either static graphs, or time-dependent graphs, whilst the structure can be classified as a tree structure, generic graph structure (directed, undirected) or compound graphs (the figure is provided in the supplementary material).

They provide extensive insight into the future challenges within the field. These include scalability, uncertainty, perceptual evaluation, interaction, task evaluation, data-type analysis, generic frameworks, user studies, and graphing benchmark systems.

Using the SurVis system [**?**], Beck et al. create a paper deriving a hierarchical taxonomy of dynamic graph visualization which is achieved by categorizing and tagging publications within the area, as well as evaluating and comparing the use of animated diagrams against time-line diagrams (see Table **??**). The paper first classifies a static graph as $G := (V, E)$, where *V* represents the vertices and *E* defines the edges. Understanding this allows a representation of a dynamic graph as $G_i := (V_i, E_i)$, where *i* defines the number of time steps for the graph [**?**]. The paper could also be considered a survey within our temporal categorization. The three main section discussed are animation, time-lines, and a hybrid of the two (the figure is provided in the supplementary material).

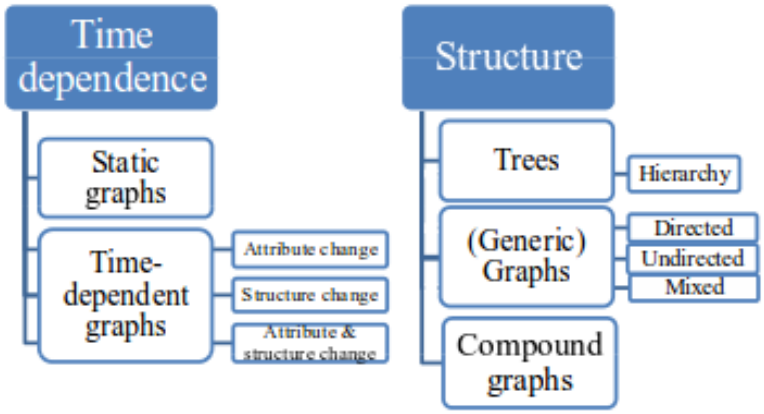


Figure 2.22: Classification of graphs with respect to the temporal or structural characteristics. Courtesy of Von Landesberger et al. [**?**] .

The hierarchical classification of dynamic graph visualization provides the density of reviewed papers per category, as well as some example techniques for each subsection. Beck et al. also provide an interactive system using SurVis [**?**] to provide extra categorization for user-exploration of the dynamic graph visualization field.

Beck et al. point out a number of open research areas in the field including the evaluation of dynamic graph visualization techniques, scalability of dynamic graphs, a larger effort in hybrid visualization, extended dimensionality of data, new interaction approaches, and new applications for dynamic graph visualization itself.

Kerracher et al. present a paper which identifies the design space of temporal graph visualization [**?**]. They aim to identify this design space, expanding the classification to create an open research area, and raising awareness of the temporal graph visualization field. The design space of temporal graph visualizations is divided into two categories, a graph structural dimension and a temporal dimension, before looking at a combined view. After showing a topology of their survey results, they discuss their findings.

The design space presented specifies a correlation between specific temporal encodings and specific graph structural encodings. Kerracher et al. create a 7x5 matrix yielding a total of 35 different combination of temporal graph visualizations (Figure **??**). Of these 35, 14 are not filled. The design space also color shades each cell based on the density of papers having this combination. This design space informs the reader that node-links within a sequential view are the most common type of temporal graph visualization.

Kerracher et al. suggest that there could be more research on temporal visualizations using space filling techniques, as well as matrices for dense networks. There are also many

			Graph structural encoding				
			Space filling	**Node-link**	**Matrix**	**Compound**	**Other**
Temporal encoding	**Multiple timeslices**	**Sequential view**	TS08 (5)	BdM06 (47)	FQ08 (1)	RPD09 (4)	
		Juxtaposition	TJ92 (1)	RM13 (20)	PS12 (3)	BD08 (4)	*Graph statistics* GGK*11 (4) *Alluvial diagrams* RB10 (3)
		Additional spatial dimension		ADM*04 (12)	BPF14 (1)		GHW09 (1)
		Superimposition		FAM*11 (5)	MGK11 (1)		
	Embedded	**Merged**		GdBG11 (5)	BPF14 (1)		
		Nested	HDKS05 (2)	SLN05 (2)	YEL10 (3)		
		Time as a node		TDKB07 (2)	Hoe11 (1)		

Figure 2.23: Design Space for Temporal Graph Visualization. Shading represents the number of papers found per combination (number in brackets). Courtesy of Kerracher et al. [**?**]

unexplored areas found within their design space. An example of this would be a space-filling graph with superimposed time-slices (refer to Figure **??**).

'Graphs or networks are used to model relationships between objects of any kind' [**?**]. Vehlow et al. survey the use of these group structures within graph visualization in order to gain a meaningful insight into the underlying data. They provide a classification of techniques used to visualize these group structures. Vehlow et al. present an overview of group structures in graphs and their encodings before presenting their classification table. The survey discusses node attributes, juxtaposed visualization, superimposed visualization, and nested visualizations.

Vehlow et al. classify the surveyed papers based on group structure and group visualization. The group structure can be categorized as disjoint flat, overlapping flat, disjoint hierarchy and overlapping hierarchy (the figure is provided in the supplementary material). Of these four, disjoint hierarchy is the most densely populated. Color nodes, glyph nodes, juxtaposition, superimposition, and nested visualizations are identified visualization techniques (see Figure **??**).

Vehlow et al. interview 7 domain experts to identify 5 research challenges in the domain. These are time-varying groups and comparison, data complexity, scalability, and the use of interaction techniques such as providing alternative group structures based on users' instructions. The final research challenge identified focus on tasks and evaluation. This involves evaluating group structures based on given tasks in order to analyze the most appropriate group structure to visualize.

Kerracher et al. look at the quickly-developing research area of visual representations of temporal graphs. This taxonomy reviews the design space in a way that enables the reader to review overlapping task categories for the purpose of formalizing graph decisions based on these criteria. [**?**, **?**]. Kerracher et al. incorporate the Andrienko Task Framework (ATF) [**?**]

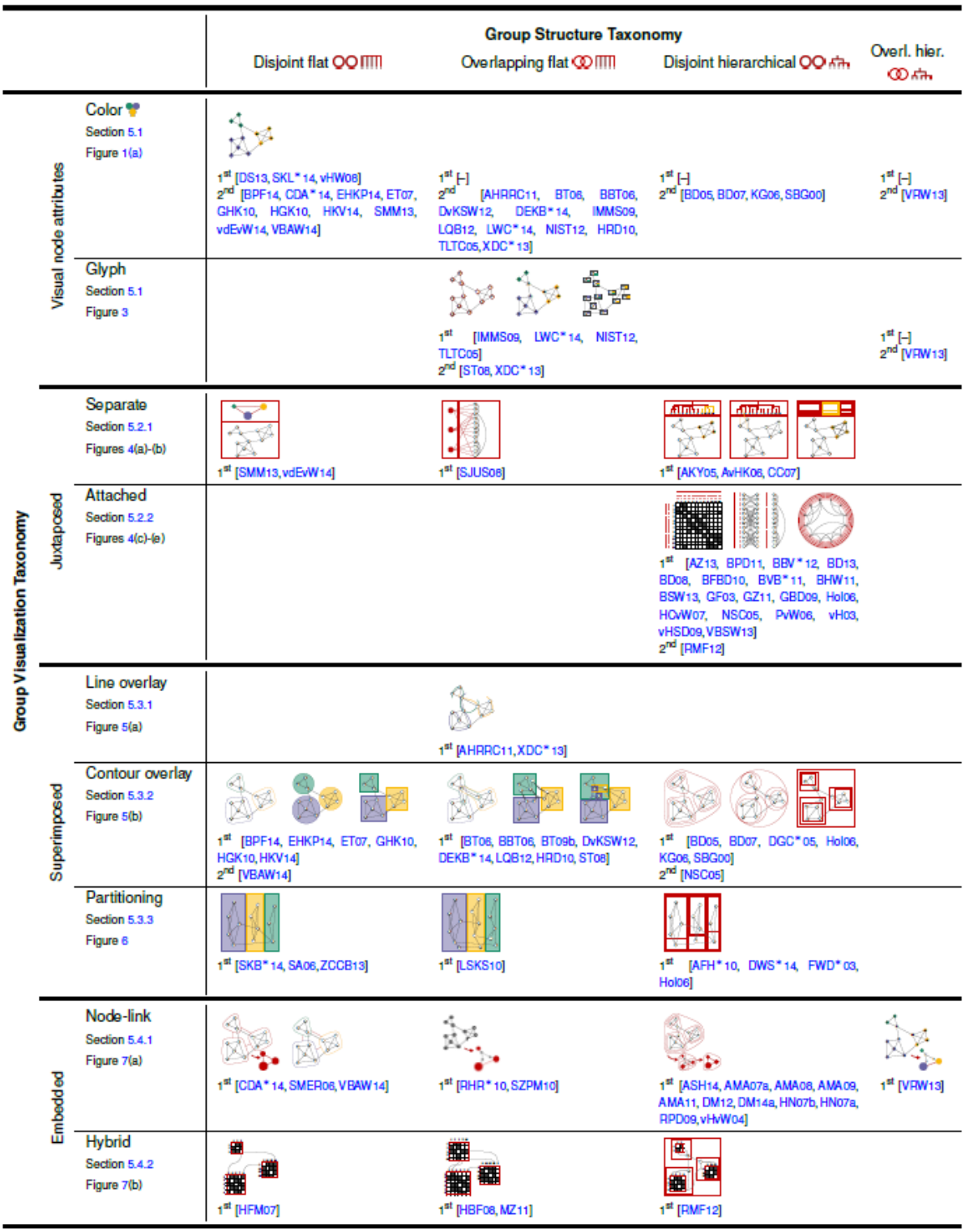


Figure 2.24: Taxonomy table created by Vehlow et al. correlating group visualizations and group structures. Courtesy of Vehlow et al. [**?**]

as the foundation of their classification (the figure is provided in the supplementary material). They analyze the limitations of the framework and propose an extension to consider structural tasks. They then present a summary of tasks primarily for the use of temporal graphs.

Kerracher et al. use the Andrienko Task Framework (ATF) [**?**] to create an indirect classification for their survey. This task categorization includes lookup tasks, comparison tasks, and relationship seeking tasks. This is combined with a classification of data items (the figure is provided in the supplementary material) discussed in their previous work, *The Design Space of Temporal Graph Visualization* [**?**].

Kerracher et al. examine the quickly-developing research area that is visual representation of temporal graphs, with a focus on techniques to support exploratory analysis tasks [**?**]. Kerracher et al. discuss a task classification which is used to create groups of exploratory tasks based on data attributes. The paper then maps visualization techniques to the quadrant classification as well as the tasks categories and provides examples of the related systems (the figure is provided in the supplementary material).

Kerracher et al. point out a few areas for future research in the field. Using the task taxonomy, a new classification can be made for more than graph areas, such as static graphs, multivariate graphs, and graph comparison. The taxonomy also opens up more research into temporal graphs, with a focus on classification of visualization techniques and mapping tasks to real world scenarios. These could lead to further research on edges cases or unsupported tasks.

They note that more research needs to be aimed at incorporating techniques from a wider range of research areas than the ones typically associated with temporal graph visualization. In particular, techniques used to support the comparisons of data items in temporal graph visualization.

Network Surveys

This section provides an understanding of the survey landscape for network visualization. Bertin defines a network as the following: '*when the correspondences on a plane can be established among all the elements of the same component, the graphic is a network*' [**?**]. This SoS discusses two surveys on this topic which include a task taxonomy with focus on network evolution analysis and a classification of matrix reordering methods for network visualization.

Ahn et al. provide a literature review with a focus on visualization tasks for network evolution analysis. The paper surveys 53 existing systems and creates a taxonomy with the aim of providing suggestions for designing future visualization tools in the domain [**?**]. This pa-

per could also be categorized in the temporal space of the SoS but is presented here since networks are the primary focus. The paper identifies three aspects of the systems: entities, properties, and temporal features. These aspects are broken down to create the design space for network evolution analysis. The paper uses the design space to analyze task frequency within network evolution analysis and surveys domain experts on their views of the design space. The results show 67% of domain experts rated the design space as very positive (see Figure **??**).

The three aspects of the system identified (entities, properties, and temporal features) are used to create the design space for network evolution analysis. The entities are broken into three subcategories including node/link, groups, and networks. The entity signifies what is being analyzed. Temporal features are displayed on the Y-axis, looking at individual events, shape of changes, and the rate of changes. The properties of the task are displayed inside the table and provide guidance on when a task can be applied, and what information is needed.

Future research directions for this field include reviewing the importance of domain properties, additional research on temporal features such as *rate of changes*. Granularity or the scale for analysis had few related papers and is an option for future research. Finally, more research into compound tasks is considered a critical research direction by experts in the field.

Behrisch et al. provide an overview of algorithms used to reorder visual matrices of tabular data. A visual matrix is defined as a visual representation of tabular data used to depict graphs and networks [**?**]. Their survey provides a guide to reordering algorithms in a unified manner to enable a wide audience to understand their differences and subtleties, and provides an overview of how, and when these algorithms are used. Behrisch et al. start by providing an introduction to the visual matrix. They discuss the different pattern types which can be used with matrices before deriving their taxonomy. They review some examples of each, before comparing the performance, and how they were tested. The paper finishes by describing directions on algorithm selection.

The matrix reordering algorithms are classified into seven families. These families are grouped by the type of algorithm which includes Robinsonian, Spectral, Dimension Reduction, Heuristic Approaches, Graph Theoretic, Bi-clustering, Interactive User-Controlled.

Some of the open research directions include hybrid solutions to ‘global vs local’ algorithms, and research into similarity, or distance calculation. Frameworks to assess the quality of patterns within matrices and craft objective functions to optimize algorithms’ performance, and human assisted reordering are important.

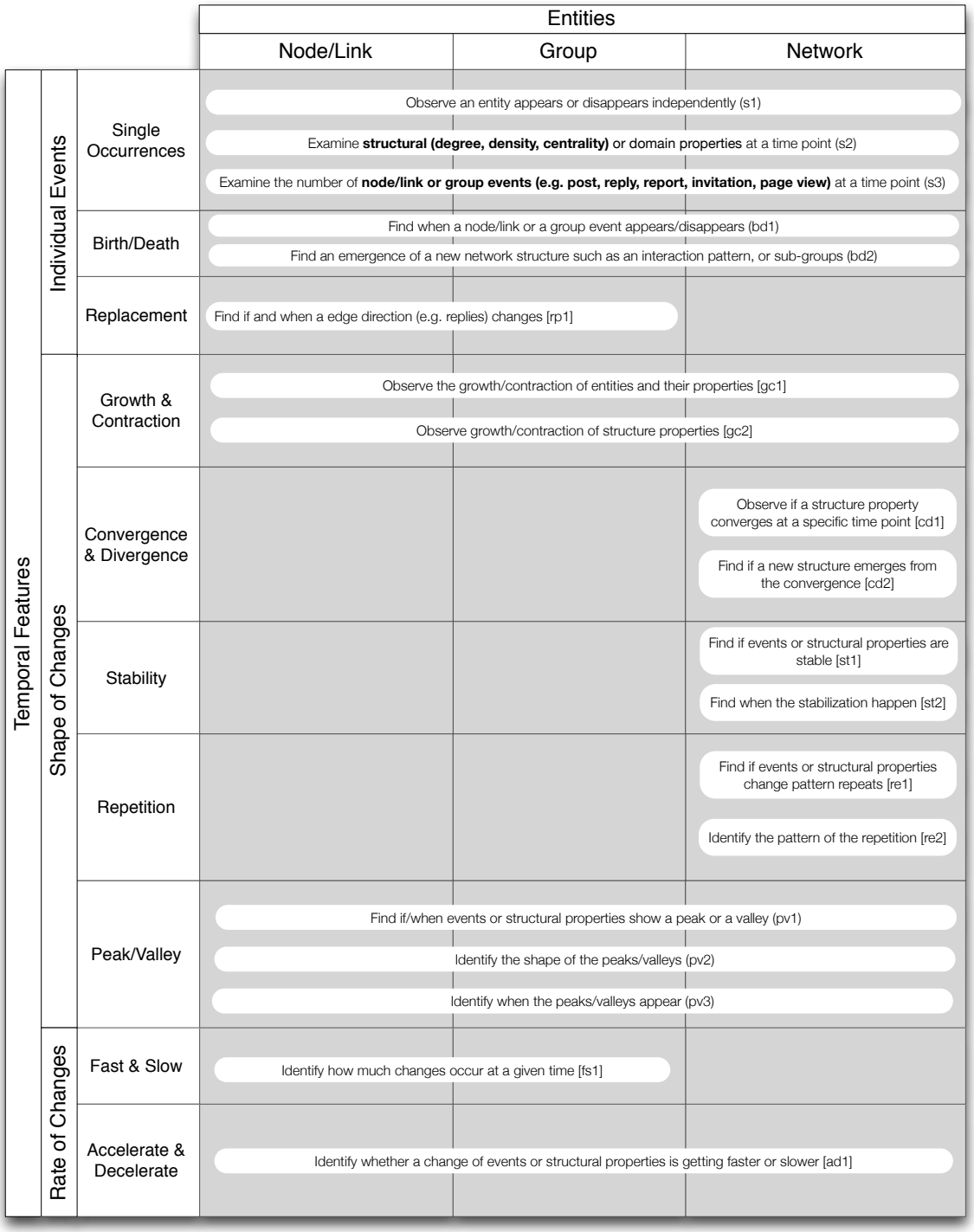


Figure 2.25: Design Space of network temporal evolution tasks courtesy of Ahn et al. [**?**]

2.3.4 Geospace + Time

The section presents at two different visualization types, geospatial visualization and temporal visualization. We place time and geospace together since they are both traditionally dimensional types of data.

Time Oriented Surveys

Here we cover surveys with a primary emphasis on time-series data or visualization across multiple time-slices. The SoS summarizes two surveys. The first paper provides a classification for visualization of dynamic data. The second paper provides an understanding of different ways to review slices of data within a Space-Time Cube.

Cottam et al. review the impact of dynamic data on Information Visualization, and how this data change can influence a visualization's discernability. This is done via the creation of a taxonomy that categorizes dynamic visualization techniques. The paper defines dynamic visualizations as "visualizations that change over time" [**?**].

Cottam et al. present the dimensions of their classification before presenting their technique matrix. Each cell of the technique matrix is reviewed, which gives an understanding of how the axes interact with each other. These are then grouped into higher-level identity groups, which represent how each classification cell would update to a new state. The paper ends by matching techniques to task scenarios.

The classification has three dimensions. The first dimension envelopes retinal (visual-based) categories, which include: (1) unchanged (immutable) scales, (2) a known scale, (3) extreme bins which categorize catch-all bins such as "100+", and (4) mutable scales which are dynamic scales. The second dimension identifies spatial categories such as fixed spatial dimension, mutable spatial dimension, new spatial elements (create), and create and deleted spatial elements. The third dimension depicts the higher-level identity groups: identity preserving changes, transitional changes, and immediate changes (the figure is provided in the supplementary material).

Cottam et al. consider that the taxonomy could be extended by looking at additional spatial categories, such as "delete but not create." The categories are not distributed evenly, so a study into why is also suggested.

Bach et al. survey a variety of temporal data visualization techniques and discuss how their operations can be used with space-time cubes in order to create a simple visualization from the 2D+time model. [**?**] The paper discusses common static space-time cube operations

which includes time-cutting, time flattening, time juxtaposition, space cutting, space flattening, sampling, and 3D rendering. Bach et al. then present the taxonomy of space-time cube operations that they have designed before giving the reader a selected sample of multi-operation systems.

The taxonomy (see Figure **??**) presents a classification of elementary space-time cube operations such as drilling, cutting and chopping. These are broken down into sub-sections with schematic illustrations in order to enable the user to easily understand what effect the operation has. For example, the flattening section is broken down into planar flattening and non-planar flattening. Planar flattening is broken down into orthogonal flattening and oblique flattening.

There are many open research areas that are discussed within the paper. Some of these include interaction techniques such as focus+context to use with different operations, research into operations for extended data dimensions, and understanding which operation is most appropriate for a given task.

Geospatial Focused Surveys

This section focuses on surveys that examine geo-spatial visualization. The section provides an understanding, and classification, of tasks for cartograms, a view of geospatial traffic data, and another review of the use of cartographic visualization in information visualization.

Chen et al. analyze various ways traffic data can be recorded as well as some different approaches that are brought forward to depict a combination of spatial, temporal, dimensional, and categorical visualization [**?**]. They systematically review how traffic data is captured. This is divided into three unique categories: Location-Based, which records data as it appears at a fixed point, or sensor range; Activity-Based, which may record data when a specific event is started or finished; Device-Based, recording from a device which records information periodically such as a GPS. Each of these has their own unique benefits and uses. Each data type can be broken down into the four types mentioned previously: spatial; temporal; dimensional; and categorical. Chen et al. begin by looking at how the traffic data can be captured and discuss the different ways this data can be processed. The main focus of the survey is how the data is visualized. The paper provides examples of usual design for time, spatial properties, spatio-temporal data, and multi-property data.

To further improve the taxonomy of traffic data, Chen et al. structure the visual design types into the three main goals: Situation-Aware Exploration and Prediction; Pattern Discovery and Clustering; and Visual Monitoring of Traffic Situations. These goals are clearly explained and then presented with relevant examples [**?**].

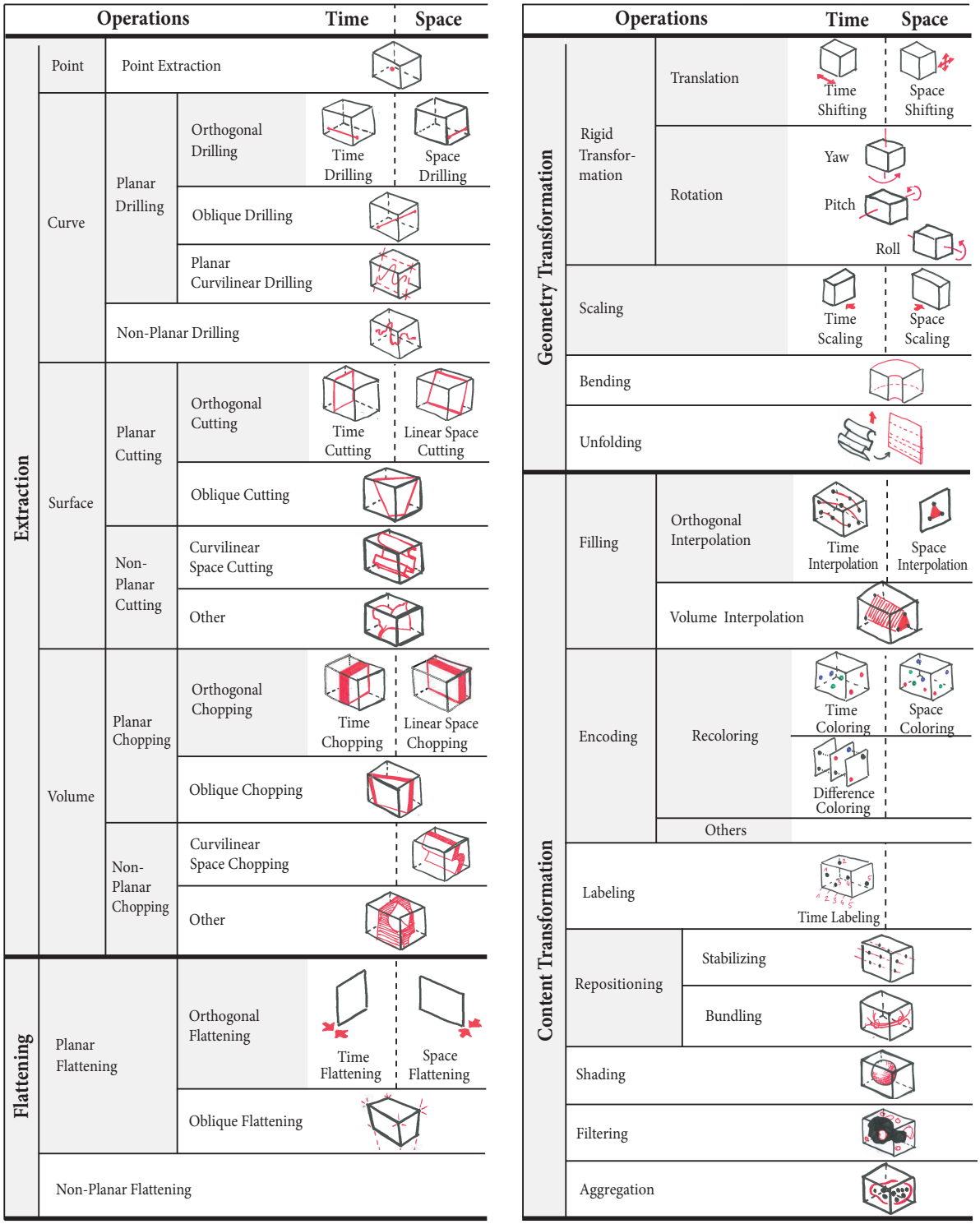


Figure 2.26: Taxonomy of Space-Time cube operations created by Bach et al. [**?**] Each operation gives a representation of how the operation may work. Bold font indicates complete operations. Gray shading indicates non-leaf nodes. Image courtesy of Bach et al. [**?**]

	Data	Properties	Data Types			Representative Datasets
			N	C	T	
Trajectory	Shipping trajectories	Time	✓			Vessel traffic data [4]
		Location	✓			
		Ship type		✓		
		Destination			✓	
		Velocity	✓			
	Aircraft trajectories	Location	✓			Flight in France [5], Europe 24 [6]
		Flight level	✓			
		Time	✓			
		Velocity	✓			
		Aircraft ID			✓	
	Automobile trajectories	Time	✓			Taxi GPS data of Beijing [7], [8], Shenzhen [9], Shanghai [10], [11], [12], San Francisco [13], New York City [14], Wuhan [15], [16], and Sweden [17]; Traffic monitoring cells data in Nanjing [18]; GPS data in Louisiana [19]
		Location	✓			
		Direction		✓		
		Change of direction		✓		
		Velocity	✓			
		Acceleration	✓			
		Pick-up/drop-off		✓		
	Train/Metro trajectories	Location	✓			Train data in France [20], Boston's metro data [21]
		Time	✓			
		Station			✓	
	Pedestrian trajectories	Location	✓			Human mobility traces [22]
		Time	✓			
		Velocity	✓			
	Mixed trajectories	Object type		✓		Intersection count [23]
		Position	✓			
		Velocity	✓			
		Direction		✓		
		Time	✓			
Incident	Tunnel incident	Stateful events		✓		Incident detection system (IDS) data [24]
		Stateless events	✓			
		Video				
	Highway incident	Location	✓			Maryland highway & traffic information [25], Traffic Management Centers Data [26], traffic incident in Singapore [27]
		Time of date	✓			
		Weather conditions		✓		
		Vehicle involved			✓	
		Incident type		✓		
	Metro incident	Time	✓			Metro smartcard records in Shenzhen [28], urban rail transit system data [29]
		Station			✓	
		Check in/out		✓		

Figure 2.27: The taxonomy displays different data types with their potential properties. These are then categorized into three data types: Numerical; Categorical; or Textual. Examples of related literature are also given. Courtesy of Chen et al. [**?**]

Chen et al. present analysis of situation-aware and immersive environments, as well as the design of huge spatio-temporal analysis of online or streamed data, as open challenges in the field. Visual Analysis of heterogeneous data, social transportation as an example, is another area that requires more focused research.

A cartogram is a type of visualization that aims to combine statistical and geographical information where areas are scaled dependent on statistical proportions. Nusrat and Kobourov study the effectiveness of cartograms as a visualization tool, as well as compare the effectiveness of different cartogram methods. The paper presents a set of cartographic visualization tasks and their application to information visualization [**?**]. Nusrat and Kobourov begin by providing an overview of cartograms, with related literature for those who want to expand their knowledge of the subject. They then present their design space for cartographic tasks and

	Goals			Means				Characteristics		Cardinality		
	Query	Search	Extract	Map Relation	Data Relation	Navigation	Derive	Low Level	High-Level	Single	Multiple	All
Recognize	√	×	×	√	×	×	×	√	×	√	×	×
Detect Change	√	×	×	√	×	×	×	√	×	√	×	×
Compare	√	×	×	×	√	×	×	√	×	×	√	×
Find top-*k*	×	√	×	×	√	×	×	×	√	×	×	√
Filter	×	√	×	×	√	×	×	×	√	×	×	√
Cluster	×	√	×	×	√	×	×	×	√	×	×	√
Locate	×	√	×	×	×	√	×	√	×	×	×	√
Find Adjacency	×	√	×	×	×	√	×	×	√	×	√	×
Summarize	×	×	√	×	×	×	√	×	√	×	×	√
Identify	×	×	√	×	×	×	√	√	×	√	×	×

Figure 2.28: Task taxonomy for cartogram visualization courtesy of Nusrat and Kobourov [**?**].

visual goals for cartogram usage. Both of these are coupled with clear examples of when they could be selected.

Nusrat and Kobourov classify tasks by reviewing two hierarchical dimensions, analytic tasks such as identification, location, sorts and clustering (see Figure **??**). The second axis maps visualization goals which includes the goal of the visualization, how a task is carried out, features of the task, and the cardinality.

Nusrat et al. follow-up their previous survey with an extended version [**?**]. They start by presenting a history of cartographic visualization, beginning with their origins in 1870. The paper examines the literature surveys related to cartograms, discussing what is presented in each, before discussing the different design-types of cartographic layouts as well as a task taxonomy extension of '*Task Taxonomy for Cartograms*' by Nusrat and Kobourov [**?**], their applications, and their effectiveness.

They introduce the three major design dimensions of cartographic visualization: Statistical accuracy, geographical accuracy, and topological accuracy. In addition, cartograms are sub-divided into four different types which include contiguous, non-contiguous, Dorling, and rectangular cartograms.

Nusrat and Kobourov pose a multitude of areas for future cartogram research. Firstly, some of the design dimensions are under-utilized, which could facilitate a paper comparing usage with less used cartographic layouts. Cartograms normally only excel in one of the three design dimension, and a study to mitigate errors in the other two dimensions may allow for

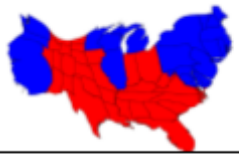
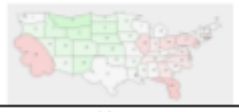
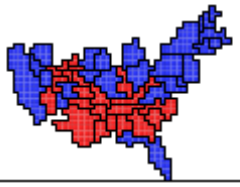
Type	Statistics	Contiguity	Geography	Topology	Example
Diffusion-based cartograms [GN04]	Almost accurate	Contiguous	Distorted	Topology-preserving	
Circular-arc cartograms [KKN13]	Not Accurate	Contiguous	Shape mostly preserved	Topology-preserving	
Optimal rubber sheet method [Sun13b]	Almost accurate	Contiguous	Distorted	Topology-preserving	
Fast, free-form rubber-sheet method [Sun13a]	Almost accurate	Contiguous	Distorted	Topology-preserving	
T-shape cartograms [ABF*13]	Accurate	Contiguous	Shape not preserved	Topology-preserving	
Non-contiguous cartograms [Ols76]	Accurate	Not contiguous	Shape preserved	Topology not preserved	
Demers cartograms [BDC02] (figure from [NYT12])	Accurate	Not contiguous	Shape not preserved (squares)	Topology not preserved	
Mosaic cartograms [CBC*15]	Not accurate	Contiguous	Shape mostly preserved	Topology-preserving	
Table cartograms [EFK*13]	Accurate	Contiguous	Shape not preserved	Topology not preserved	

Figure 2.29: A 2D systematic overview of different types of cartograms, displayed with their categorizations. Courtesy of Nusrat and Kobourov [**?**]

Technique	Visualization A	Visualization B	Spatial Relation	Data Relation
ComVis [24] (Figure 2)	any	any	juxtapose	none
Improvise [39] (Figure 3)	any	any	juxtapose	none
Jigsaw [36]	any	any	juxtapose	none
Snap-Together [30]	any	any	juxtapose	none
semantic substrates [34] (Figure 4)	node-link	node-link	juxtapose	item-item
VisLink [11] (Figure 5)	radial graph	node-link	juxtapose	item-item
Napoleon's March on Moscow [37]	time line view	area visualization	juxtapose	item-item
Mapgets [38] (Figure 6)	map	text	superimpose	item-item
GeoSpace [22] (Figure 7)	map	bar graph	superimpose	item-item
3D GIS [8]	map	glyphs	superimpose	item-item
Scatter Plots in Parallel Coordinates [45] (Figure 8)	parallel coordinate	scatterplot	overload	item-dimension
Graph links on treemaps [14] (Figure 9)	treemap	node-link	overload	item-item
SparkClouds [21]	tag cloud	line graph	overload	item-item
ZAME [13] (Figure 10)	matrix	glyphs	nested	item-group
NodeTrix [17] (Figure 11)	node-link	matrix	nested	item-group
TimeMatrix [44]	matrix	glyphs	nested	item-group
GPUVis [25]	Scatterplot	glyphs	nested	item-group

Figure 2.30: Classification of common composite visualization techniques [**?**] .

a better understanding of cartogram usage. Some other areas of interest are the mapping of multivariate data, memorability and recall within cartograms, uncertainty within cartograms, and 3D cartographic visualization.

2.3.5 Coordinated Multiple View (CMV) Surveys

Coordinated Multiple Views (CMVs) surveys focus on literature that examines the coordination or linkage between multiple views. This section summarizes two papers related to the subject. The first reviews the use of composite visualization and the second provides an understanding of multi-faceted graph visualization.

Javed and Elmqvist examine different Composite Visualization Views (CVVs), which are defined as *"a visual composition of two or more visual structures in the same view,"* and present their CVV design patterns created via a literature survey [**?**]. They start by discussing different CVVs, including their design patterns and existing formalisms. This is followed with an in-depth look at the different types of views (juxtaposed, integrated, superimposed, etc), with examples of each. They then present their design space and guidelines, as well as their 1-N classification table.

The authors look at different techniques and how the visualization's relations are classified. The grouping looks at the result of merging two visual designs, the composite relation (juxtaposed, integrated, superimposed, overloaded, nested), as well as what data-relation is created (item-item, item-group, etc).

Javed and Elmqvist suggest their design patterns are limited to literature reviewed and therefore work can be invested into extending their framework. The design pattern is also

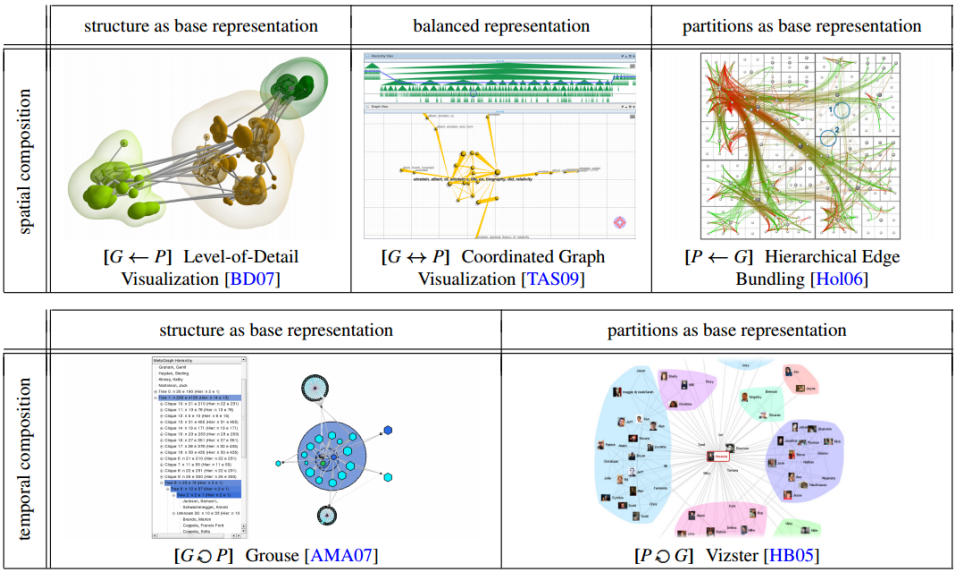


Figure 2.31: Examples of multiple facet representations within visualization. Courtesy of Hadlak et al. [**?**]

limited to only spatial relation and does not look at other composite visualization views, such as interaction or animation.

Many surveys focus on only a single additional facet in order to classify graphing techniques. Hadlek et al. aim to build on existing surveys in order to create a more in-depth observation of four common facets: partitions, attributes, time, and space. Each of these characteristics are discussed based on their relationship as well as examples of how these graphs can be represented depending on the hierarchy. Hadlek et al. focus on an output oriented perspective, and optimize facet selection by focusing on their composition. These compositions are given a representative visualization (seen in Figure **??**) to discuss in detail in the content of the survey. Hadlek et al. analyse visual design of the graph structure with a single additional facet and graph structure with multiple additional facets. These are sub-divided for each common facet. This is followed by analyzing multiple instances of graph facets.

Each graph structure is split into five combinations. Whilst looking at the spatial composition, an example is given for structure as the base representation, partitions as the base representation, or a balanced representation. A temporal composition has an example for either structure as the base representation or partitions as the base representation.

The paper notes the exploration of geo-spatial graph visualization, by reviewing output or task taxonomy, has yet to be published. Hadlek et al. also point out that some of the facets discussed had very sparse usage such as temporal compositions. Finally, only four facets were examined but there are many other extensions such as provenance, uncertainty,

heterogeneity, or text/annotations that have little-or-no exposure which could be a new thread of research to investigate.

2.3.6 Real-World and Applications

Many literature surveys focus on real-world scenarios and applications. This area covers a wide range of surveys including finance, health-care, security, systems, software visualization, and visualization frameworks.

Finance Focused Surveys

This section has a focus on survey papers related to finance visualization. The survey summarizes one paper focused on different sources of financial data and how they are visualized.

Ko et al. perform and present a study of visualization and visual analytics of financial data. Economy is an important field for any business which has led to financial data being a popular topic for visualization in industries. They aim to utilize existing papers in order to help researchers design better systems and understand new research fields within the area [**?**]. After discussing the survey scope, Ko et al. explore the types of data analyzed and some data sources for each. Some examples of this include stock data, transaction data, and fund data. The paper derives a classification of techniques and provides examples of these uses before examining the interaction methods and evaluation methods. This is followed by an evaluation of the papers.

Ko et al. examine multiple ways to classify the gathered information. The first classification tests the type of data used by each paper, which indicates a heavy focus on stock data. The second classification discusses papers based on automated techniques, such as K-means. The third looks at visualization techniques, based on Keim's technique taxonomy [**?**] (see Figure **??**). The fourth categorization considers interaction methods and the final organization examines evaluation methods.

They propose that there are nine more business domains which could have their visual analytics reviewed such as economic analysis, financial risk management, and portfolio management. They also discuss the lack of research into company performance with financial data and suggest this as an open field. An important area for research is with automated visualization techniques, which industry experts believe are important to facilitate a richer depth of information. The final point they discuss is the use of heterogeneous data to enable improved prediction models.

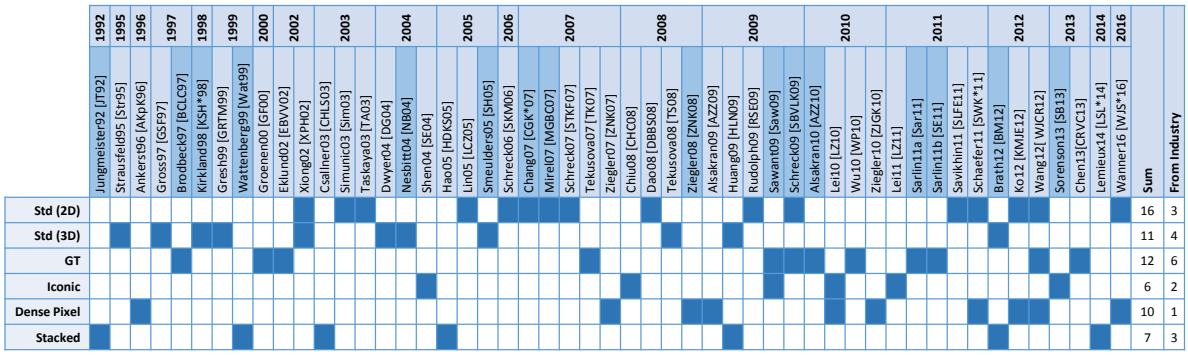


Figure 2.32: Ko et al.'s Categorization of surveyed papers via Keim's visualization techniques taxonomy [**?**, **?**] .

Security-based Literature Surveys

This section includes survey papers that present security systems. One features a focus on visualization systems for network security and another on malware analysis.

Shiravi et al. provide a comprehensive overview of network security visualization and present data sources for each. The paper also provides a taxonomy that includes literature across five use-cases [**?**]. They present a table that provides potential data sources for security visualization before giving an in-depth view of five use-cases for reviewing network security and their related papers. The use-cases are host/server monitoring, internal/external monitoring, port activity, attack patterns, and routing behavior.

The taxonomy takes the form of a 2D table, sub-divided into five sections representing the five use-cases. The taxonomy reviews the type of visualization techniques and data source of each research paper. It also includes the number of citations a paper has to emphasize systems that have more references, as well as signifying whether the system is available online (the figure is provided in the supplementary material).

Shiravi et al. present future research topics including situation awareness in presenting information, user experience evaluation, scalablity, occlusion in network security, privacy preservation and novel ways to provide 3D imagery of the data.

Wagner et al. present a systematic overview and classification of malware visualization systems used for visual analysis. The field is gaining more and more interest due to the increasing threat of malware attacks on user systems. Malware is defined as *"any software that does something that causes harm to a user, computer or network"* [**?**]. They focus on malware systems for visual analysis. They first provide tools, discussed as *data providers*,

Visualization System	Visualization Technique(s)	Data Source(s)	Number of Citations	
Host / Server Monitoring				
Erbacher et al. [4][5]	Glyph	Server Logs	106 | 7	
Tudumi [6]	3D Node Link	Server Logs	38	
NVisionIP [7,8]	Scatter Plot	NetFlows	145 | 20	Available Online
Portall [9]	Node Link	Packet Traces	21	
HoNe [10]	Node Link	Packet Traces	8	
Perlman et al. [11]	Node Link | Glyph	Packet Traces	5	
Radial Traffic [12]	Radial Panel	Packet Traces	23	
Mansmann et al. [13]	Node Link	Packet Traces	2	
Internal/External Monitoring				
VISUAL [14]	Scatter Plot | IP Matrix	Packet Traces	93	
VizFlowConnect [15]	Parallel Coordinates	NetFlows	111	Available Online
Erbacher et al. [16]	Radial Panel	Packet Traces	8	
TNV [17]	IP Matrix | Color Map	Packet Traces	48	Available Online
Port Activity				
Abdullah et al. [18]	Histogram	Packet Traces	30	
Cube of Doom [19]	3D Scatter Plot	Packet Traces	99	Available Online
PortVis [20]	Scatter Plot	NetFlows	112	
NetBytes Viewer [21]	3D Scatter Plot	NetFlows	7	
Existence Plots [22]	Scatter Plot	Packet Traces	3	
Attack Patterns				
Giardin [29]	Color Map	Packet Traces	60	
NIVA [30]	Node Link | Glyph	Intrusion Alerts	51	
Snort View [31]	Scatter Plot | Glyph	Intrusion Alerts	67	Available Online
IDGraphs [32]	Scatter Plot	NetFlows	29	
IP Matrix [33]	Scatter Plot | Color	Intrusion Alerts	21	
Visual Firewall [34]	Scatter Plot	Packet Traces	24	
IDS Rainstorm [35]	Scatter Plot	Intrusion Alerts	60	
Vizalert [36][37][38]	Radial Panel	Intrusion Alerts	38 | 35 | 29	
Rumint [39][40]	Parallel Coordinates	Packet Traces	15 | 35	Available Online
Ren et al. [41]	Flying Term	DNS Traces	10	
Xiao et al. [42]	Scatter Plot	Packet Traces	23	
Svision [43]	3D Scatter Plot	Packet Traces	9	
Mansmann et al. [44]	Treemap	Packet Traces	20	
SpiralView [45]	Radial Panel	Intrusion Alerts	5	
NFlowVis [46]	Treemap	NetFlows	17	
Avisa [49]	Radial Panel	Intrusion Alerts	2	
Routing Behavior				
BGPlay [50]	Node Link	BGP Traces	22	
Wong et al. [51]	Node Link	BGP Traces	9	
LinkRank [52]	Node Link	BGP Traces	16	
Teoh et al. [53][54][55]	Histogram | Node Link	BGP Traces	54 | 28 | 35	
BGP Eye [56]	Color Map	BGP Traces	8	

Figure 2.33: Taxonomy of Security Visualization Systems, divided into different use-cases. Created by Shiravi et al. [**?**] .

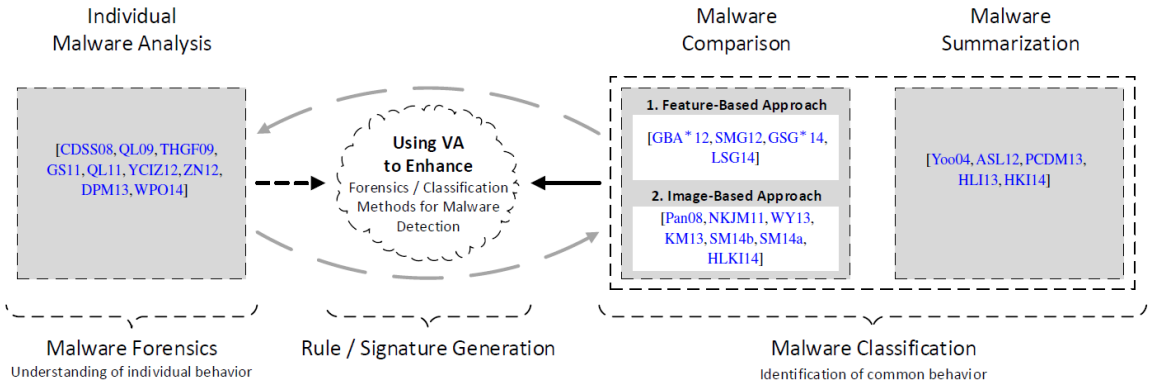


Figure 2.34: The 1D malware visualization taxonomy courtesy of Wagner et al. [**?**]

and provide a comparison of their usage. They present their malware visualization taxonomy and categorize each data provider in a number of different classification groups.

Wagner et al. create three categories for malware visualization systems: (1) Individual malware analysis which enables the system to look at a single malware sample and learn about its individual behavior. (2) Malware comparison which facilitates comparison of a range of malware for viewing. (3) Malware summarisation that outlines the behavior of different malware samples (see Figure **??**).

They find that malware visualization is cleanly partitioned between each classification category and believe more work needs to be done in creating a connection between categories. Using different systems could cause unnecessary overlap for users which could be minimized with a system that could move through this categorization. Some other challenges in the field include the integration of more data sources, a stronger understanding of the requirements for malware visualization, enabling expert analysis and externalization, and an increased focus on the relation between analysis and visualization for malware.

Systems-based Surveys

This section focuses on Surveys that have an emphasis on classifying systems. We summarize one survey paper in this section which looks at performance visualization of large-scale systems.

Gao et al. present a review on papers designated for researching performance visualization on large-scale systems. Gao et al. define performance visualization as *'the use of graphical display techniques for the visual analysis of performance data'* [**?**]. The paper aims to shed light on open research areas, and increased discussion on the design of visual tools

Category	Performance Visualization Techniques	Example applications and studies
Simple visual structures	Pie charts, distribution, box plots, kiviat diagrams	ParaGraph [2], PET [20], SvPablo [16], VAMPIR [21], Devise [22], AIMS [9]
	Timeline views	Paje [23], AIMS [9], Devise [22], AerialVision [24], Paraver [25], SIEVE [14], Virtue [13], utilization and algorithm timeline views in [17]
	Information typologies	SHMAP [26], Vista [4], Voyeur [27], processor and network port display in [28], hierarchical display in [12]
	Information landscape	Triva [29], Cichild [30]
	Trees & networks	Paradyn [18], Cone Trees [31], Virtue [13], [32]
Composed visual structures	Single-axis composition	AIMS [9], Vista [4]
	Double-axis composition	Devise [22], AerialVision [24]
	Case composition	Triva [29]
Interactive visual structure	Interaction through controls (data input, data transformation, visual mapping definition, view operations)	Paje[23], data input, filtering, and view manipulation in [28] and [32]
	Interaction through images (magnifying lens, cascading displays, linking and brushing, direct manipulation of views and objects)	Virtue [13], Cone Trees [31], Devise [22], direct manipulation of the 3D cone and virtual threads in [32]
Focus + context visual structures	Macro-micro composite view	Microscopic profile in [4], PC-Histogram in [24]

Figure 2.35: A classification of performance visualization techniques courtesy of Gao et al. [**?**].

for these systems. The paper provides a brief background of performance visualization and how it functions.

Gao et al. classify performance visualization techniques using four main categories. Simple visual structures which include statistical charts with one or two variables, composed visual structures that include a combination of simple chart views, interactive visual structures featuring structures that provide a variety of user interactions, and focus+context which refers to visualizations with mapping that is automatically modified without the need of user interaction (the figure is provided in the supplementary material).

Some future work areas presented by Gao et al. include scalability, user studies, and the synthesis of high-level context with low-level detail.

	Level	Focus	Section	Visualization Technique	Representation	References	Year
Time T Visualization	Line	Line properties	2	Seesoft	2D colored pixel	[1], [2]	1992
				Sv3d	3D colored cuboid	[3], [4]	2003
	Class	Functioning, Metrics	3	Class BluePrint	2D layers and graph	[5], [6], [7]	1999
	Architecture	Organization	4.1	Treemap	2D/3D colored nested boxes	[8], [9], [10]	1991
				Circular Treemap	2D/3D colored nested circles	[8], [11]	1991
				City/Cities	3D city metaphor	[12], [13], [14], [15]	1993
				Sunburst	2D colored radial display	[16], [17], [18]	1998
				Solar System	3D solar system metaphor	[19], [20]	2003
				Voronoi Treemap	2D colored irregular shapes	[21]	2005
		Relationships	4.2	Dependency Structure Matrix	2D table	[22], [23], [24]	1981
				UML	2D diagrams	[25]	1996
				Geon	3D geon diagrams	[26], [27], [28]	1998
				Solar System	3D solar system metaphor	[19], [20]	2003
				Landscape	3D landscape metaphor	[29], [30]	2004
				Hierarchical Edge Bundles	2D graph with bundled edges	[31]	2006
				City/Cities	3D city metaphor with edges	[32], [33], [34]	2007
				3D Clustered Graph	3D clustered graph	[35]	2007
		Metrics	4.3	Polymetric views	2D graph	[5], [36], [37]	1999
				Solar System	3D Solar system metaphor with edges	[19], [20]	2003
				UML MetricView	2D UML diagrams with charts on top	[38]	2005
				Treemap metrics	2D nested boxes with color and texture	[39]	2005
				City	3D City metaphor	[40], [41], [42], [43]	2005
				UML Area Of Interest	2D diagrams with area of interest	[44], [45]	2006
Visualizing Evolution	Line	Changes	5.1	Code Flow	cable-and-plug wiring metaphor	[46], [47]	2007
	Class		5.2	TimeLine	3D building metaphor	[48]	2008
	Archi.	Organizational Changes	5.3.1	Hierarchical Edge Bundles	2D graph with bundled edges	[49]	2008
		Metrics Evolution		Evolution Matrix	2D matrix	[50], [51]	2001
			5.3.2	RelVis	2D Kiviat diagrams and graph	[52]	2005
				City/Cities	3D city metaphor with animation	[48], [53]	2008

Figure 2.36: Caserta and Zendra present a table that classifies methods that visualise the static aspects of software and the associated literature [**?**] .

Software Visualization Surveys

Software Visualization papers focus on visualizing aspects of software creation. Diehl defines the topic as follows:*'Software visualization encompasses the development and evaluation of methods for graphically representing different aspects of software, including its structure, its execution, and its evolution'* [**?**]. The SoS summarizes one recent survey paper in this section that focuses on the static aspects of software visualization.

Caserta and Zendra categorize visualization techniques that represent the static aspects of software and it's evolution. The paper defines visualization of the static aspects of software as '*visualizing software as it is coded, and dealing with information that is valid for all possible executions of the software*' [**?**]. Evolution of software adds a temporal dimension to the visualization of the static aspects of software. The paper provides a guide to papers that feature code-line-centered visualization, class-centered visualization, architecture visualization, and the visualization of software evolution. They provide examples of different types of visual design for each and how they may be applied.

The hierarchical, 1-N, classification presents the representation and visualization techniques used for each paper, and shows how they fit into their taxonomy (see Figure **??**).

Caserta and Zendra propose that it would be beneficial to invest research into usability evaluation to find the most effective ways to visualize the static aspects of software. There is also limited research in navigation and interaction for 3D visualization within the field.

Surveys of Frameworks

This section examines the review of frameworks, which is defined as *'a basic structure underlying a concept'* [**?**]. The section covers two surveys: the first presents a design framework survey for bi-cluster visualizations and the second presents a framework for emphasis in visualizations.

Sun et al. provide a survey focused on bi-cluster visualization, design considerations, and applications. Bi-clusters *"provide a rich high-level abstraction that represents coordinated relationships between groups of entities of different types"* [**?**]. The advantages and disadvantages are compared, and a five-level relationship is presented to assist in design options that support user-tasks. The paper describes the concept of bi-clustering and the five relationship levels of the bi-cluster visualization design framework. These include: entity level (single entity relationships), group level (entity group relationships), bi-cluster level (coordinated relationships), chain level (chained coordinated relationships), and the schema level (schema level relationships). The paper also examines four levels of interaction design: readability, navigation, parameter, and object level.

Sun et al. provide a summary of the five-level design framework for bi-cluster visualization. This incorporates the interaction design level, major tasks, design choice, and trade-offs.

They identify three challenges in the field. The first challenge is the creation of an example that implements design options across all five levels of the framework. The second challenge discusses traversal between the different levels. The final challenge suggests optimal layout of bi-cluster chains in visualization.

Hall et al. present a mathematical Framework for Information Visualization Emphasis (FIVE) by reviewing existing emphasis literature and frameworks [**?**]. Some examples of emphasis provided include highlighting regions of interest, animating data points, and altering the size of data points. They first present a language for emphasizing sub-sets of data, and present a table displaying how their framework compares to previous solutions (see Figure **??**). The paper then discusses different types of emphasis effects and how they can be used such as position, color, motion and transparency. Finally, the paper discusses the opportunities to use FIVE and some future directions for research.

Relations	Major Tasks	Design Choices			Pros	Cons
		Visual Representation	Supplementary Visual Technique	Interaction Design		
Entity Level	1. Show an entity 2. Show a group of entities 3. Show entity level relations (a single case or multiple cases) 4. Show single entity vs. groups of entities level relations (a single case or multiple cases) 5. Find relevant entities for a specific entity 6. Verify relations between some entities 7. Discriminate some entities from others 8. Mark important entities or relations	The Node-Link Diagram	1. Edge bundling 2. Use spatial distance (e.g. the force-directed layout) 3. Use spatial distance + hiding links 4. Color coding to separate nodes of different domains or selected and unselected nodes or links 5. Visual marks (e.g., shapes) to separate nodes and/or links	1. Select nodes/links 2. Highlight nodes/links 3. Drag nodes/links	1. An intuitive way to show either an entity or multiple entities and relations between entities 2. Customizable spatial layout for users 3. Links clearly show specific relations between entities	1. Entities are randomly placed in the space, so it may be difficult to find an entity if there are many entities 2. The number of links exerts much impact on the readability of the diagram 3. Without links, relations between entities cannot be identified easily 4. Color coding and visual marks are not efficient to visually separate domains
		A Simple Matrix	1. A single cell to represent Entity 2. A row or a column to represent Group 3. Use a heatmap	1. Select cells 2. Highlight cells 3. Extract a cell 4. Merge cells	Avoid visual clutters caused by too many links	1. Not as easy as node-link diagrams to perceive 2. Columns or rows rearrangement is the only way to change the layout
		Parallel Coordinates with Two Domains	1. Edge bundling 2. Using curved lines to indicate links	1. Select entities 2. Highlight polylines/entities 3. Brushing 4. Axes rearrangement 5. Entities reposition in axes	1. Place entities of the same group together 2. Relatively easy to find entities 3. Efficiently select multiple entities/polylines	1. The number of links exerts much impact on the readability of the diagram 2. Without links, relations between entities cannot be identified easily 3. Sometimes entity reposition (e.g., moving relevant entities to the top) is necessary to understand grouping
Group Level		Tree Visualizations	1. Icicle 2. Bubble trees 3. Treemaps	1. Select nodes/links 2. Highlight nodes/links 3. Path extraction	Clearly represent hierarchical relations	1. Not all Groups are hierarchical relations 2. Cannot represent biclusters and bicluster-chains
Bicluster Level	1. Show a bicluster 2. Show all biclusters 3. Find biclusters of interest 4. Mark biclusters of interest	Matrices	1. Use a heatmap 2. Reorder rows or columns 3. Repeat rows or columns 4. Color coding the region of a bicluster	1. Reorder rows/columns 2. Select biclusters 3. Highlight biclusters 4. Replicate rows/columns	1. A visual representation that is easy to understand biclusters 2. Efficiently reduce visual clutters caused by many links	1. It is difficult to display all biclusters without replicating rows and/or columns 2. Replicated rows or columns may cause confusion 3. Overlaps may obscure biclusters with less entities
		Parallel Coordinates with Two Domains	1. Edge bundling 2. Use curved lines 3. Wrap entities with polylines 4. Tile-based parallel coordinates	1. Select entities 2. Brushing 3. Highlight polylines/entities/ribbons 4. Axes rearrangement 5. Entities reposition in axes	1. Place entities of the same group together 2. Relatively easy to find entities 3. Efficiently select multiple entities/polylines	1. The number of links exerts much impact on the readability of the diagram 2. Without links, relations between entities cannot be easily identified 3. Sometimes entity reposition (e.g., moving relevant entities to the top) is necessary to understand the relation
		Zoned Node-Link Diagram	1. Wrap nodes of a bicluster in a colored region 2. Use force-directed layout 3. Hide links between nodes	1. Select nodes/links 2. Highlight nodes/links 3. Drag nodes/links	1. Customizable spatial layout for users 2. Links clearly show relations between specific entities 3. Easily find entities that are shared between biclusters	1. Entities are randomly placed in the space, so it may be difficult to find an entity if there are many entities 2. Without links, relations between entities cannot be identified easily 3. Biclusters with less entities may be obscured in the overlapping region
Chain Level	1. Show a chain 2. Show all chains 3. Find chains of interest 4. Mark chains of interest	Node-link Diagram + Matrices	Combine all supplementary visual techniques that the node-link diagram and matrix based visualizations can use and the Bubble Sets technique	Combine all interactions that the node-link diagram and matrix based visualizations can use and path extraction	1. Efficiently reduce the number of links 2. A customizable spatial layout for users 3. Show the overview of the data based on bicluster-chains 4. By following links, users can find out how a bicluster-chain is formed	1. Entities may replicate many times in multiple matrices 2. Not a trivial visualization for users to understand connections across several biclusters 3. Which bicluster to choose to start a bicluster-chain is a problem
		Parallel Coordinates + Matrices	Combine all supplementary visual techniques that parallel coordinates and matrix based visualizations can use and the Bubble Set technique	Combine all interactions that parallel coordinates and matrix based visualizations can use and path extraction		
Schema Level	1. Show the overview of a dataset 2. Guide the exploration of chains or biclusters	The Node-Link Diagram	1. Clutter Map 2. The PivotGraph technique 3. Color coding to indicate different domains 4. Visual marks (e.g., shapes) to separate nodes and/or links 5. Use spatial distance (e.g. force-directed layout) 6. Use spatial distance + hiding links	1. Select nodes/links 2. Highlight nodes/links 3. Dynamic path extraction	1. An intuitive way to show relations between domains 2. The size of nodes and the thickness of links can be used to encode the information of biclusters and/or chains	1. The layout of PivotGraph cannot be easily changed by users 2. Depend on links to perceive relations across several specific domains.
		The Chord Diagram	1. Color coding of chords to indicate different domains 2. Use ribbons between chords to indicate connections	1. Select chords/ribbons 2. Highlight chords/ribbons	1. An intuitive way to show relations between domains 2. The length of chords and the thickness of ribbons can be used to encode the information of bicluster and/or chains	1. Not efficient for a dataset with many domains 2. Ribbons inside the diagram may form visual clutters 3. Paths inside the diagram may be obscured by too many crossing ribbons

Figure 2.37: Design framework associated with bicluster visualization. Courtesy of Sun et al. [**?**].

The frameworks are split into three categories. (1) *Magnification* - papers that describe magnification emphasis effects. (2) *Beyond magnification* - papers describing non-magnification emphasis effects. (3) *Data suppression* - papers that focus on the creation of emphasis effects through data suppression.

They provide four areas for future work: creating emphasis effects using under-explored visual variables and time variation, exploring alternative ways to vary data point prominence to create emphasis effects, providing a richer space of how to define and implement emphasis effects, and conducting empirical studies of emphasis effects.

2.3.7 Overview Surveys

Surveys that attempt to cover information visualization as a whole are presented here. This section summarizes two papers that review recent advancements in information visualization.

We summarize one survey that reviews the use of interactive lenses in visualization. Tominski et al. aim to analyze the use of interactive lenses in the context of visualization by

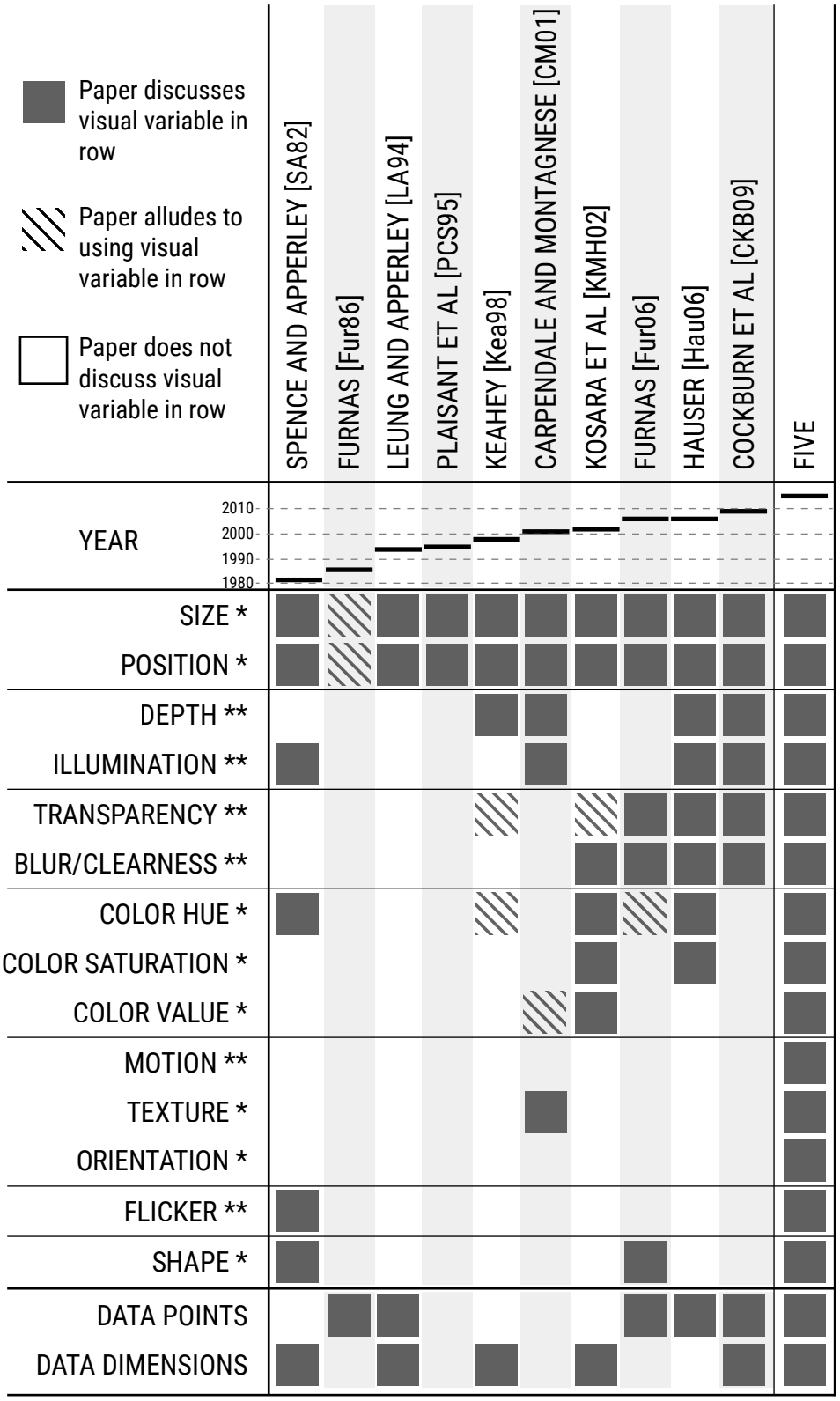


Figure 2.38: Hall et al. present a table used to classify previous emphasis frameworks to diagnose what types of emphasis are discussed, and compare them to FIVE [**?**] .

reviewing the different techniques used to create lenses, whilst also helping researchers identify when to use interactive lenses. They discuss applicable data types such as geo-spatial data, why incorporating lenses is beneficial to the user experience, and some important techniques to be aware of if you are interested in this type of visualization. An interactive lens is defined as *'to provide an on demand alternative visual representation of the data underlying a local area of the screen'* [**?**]. After introducing the model of the interactive lens as well as the discussion of lens usage, they segment the research into interaction types. This includes examination of mouse and keyboard interaction, touch and multi-touch interaction, tangible interaction, tangible views and spatial interaction, gaze-based interaction and head tracking. [**?**]

The taxonomy of Tominski et al's survey examines both the data types that each visualization technique demonstrates (temporal, geospatial, flow, etc.), as well as the task that is achieved using it (across a total of 43 papers). This taxonomy enables discussion of possible future work such as the use of lenses for multi-user work (see Figure **??**).

After reviewing some survey notes, they suggest that the need for more dynamic lenses with flexibility is an important design note as well as useful for a more varied use of functionality. Although mentioned in the survey, Tominski et al. would like to further investigate the development of multi-user or shared lenses, due to the growth in high-resolution and interactive displays. Finally, the idea of lens tool kits is an important focus area. Lenses are globally recognized visualization types but are in low use due to how interwoven they are with visualizations. With big data becoming a strong focus, this is something that needs further development. Tominski et al. also provide an extended version of the classification (see Figure **??**) [**?**].

Liu et al. create a comprehensive study on the domain of information visualization. The aim of the paper is to derive an organization of the field, describing features, goals and state-of-the-art approaches for each category [**?**]. The paper opens by examining the visualization pipeline and classification schemes. They proceed to present their 1-N taxonomy of the literature landscape within recent years, and present each topic whilst giving examples of papers in the related area. This continues on to communicating some technical challenges.

Liu et al. break their taxonomy down into four main categories: empirical methodologies, interactions, frameworks, and applications. These categories are sub-divided into subcategories (the figure is provided in the supplementary material). The classification has a lot of overlap with our organization.

Liu et al. describe an abundance of open research areas including usability, scalability, heterogeneous data, real-time visualization, and uncertainty.

	Data							Tasks						
Techniques	Temporal	Geospatial	Flow	Volume	Multivariate	Graph	Document	Select	Explore	Reconfigure	Encode	Abstract & Elaborate	Filter	Connect
[SB92, SB94] Fisheye Views		●				●						●		
[RM93] Document Lens							●					●		
[CMS94] MagicSphere				●				●			●	●	●	
[RC94] Table Lens					●							●		
[VCWP96] 3D Magic Lenses				●								●	●	
* [FG98] Lenses for Flow Visualization			●								●	●		
[FP99] Excentric Labeling					●				●			●		●
[SHER99] Interactive 3D Lens			●	●	●		●				●	●	●	
[LHJ01] Volume Magnification Lens				●								●		
[SFR01] Time Lens	●								●					
[BCPS02] Fuzzy Lens, Base-Pair Lens, Ring Lens						●				●	●	●		
[MTHG03] 3D Flow Lens			●								●	●		
* [WCG03] EdgeLens						●				●				
* [vHvW04] Graph Abstraction Lens						●						●		
[RHS05] Magic Lenses in Geo-Environments		●									●		●	
* [EBD05, ED06b, ED06a] Sampling Lens					●								●	
[RLE05] Temporal Magic Lens	●								●					
* [WZMK05] The Magic Volume Lens				●								●		
[TAvHS06] Local Edge Lens						●							●	
[KSW06] ClearView				●								●	●	
[TGBD08] 3D Generalization Lenses		●										●		
* [TAS09] Layout Lens						●				●				●
[BRL09] Enhanced Excentric Labeling					●				●			●		●
* [MCH*09] Bring &Go						●				●				●
[ACP10] High-Precision Magnification Lenses	●	●	●	●	●	●	●	●				●		
[JDK10] Network Lens						●					●			
* [KCJ*10] Detail Lenses for Routes		●										●		●
[Kin10] SignalLens	●											●		
* [SNDC10] PushLens						●				●				
* [STSD10] Tangible Views		●			●	●		●	●	●	●	●	●	●
* [EDF11] Color Lens	●	●	●	●	●	●					●			
[GNBP11] FlowLens			●						●		●		●	
[HLTE11] SemLens					●						●		●	
[HTE11] MoleView		●			●	●				●				
[LWG11] Facet Lens							●				●		●	
[PBKE11] EdgeAnalyser					●	●					●	●		
[ZCB11] MagicAnalytics Lens	●													●
* [ZCPB11] ChronoLenses	●												●	●
[TSAA12] Time Lens		●							●		●			●
[PPCP12] JellyLens		●										●		
* [KTW*13] TrajectoryLenses		●											●	
[PPCP13] Gimlenses				●								●	●	
[UvK13] Magic Lenses for Hypergraphs						●							●	
* [CC13] Lens for Querying Documents							●						●	●
[AACP14] RouteLens		●										●		
[BHR14] PhysicLenses		●										●		
* [MW14] Bubble Lens					●			●				●		
[DMC14] VectorLens	●				●			●					●	
[KRD14] Multi-touch graph Lenses						●				●			●	●
[DSA15] 3DArcLens		●				●				●			●	

Figure 2.39: Lens Techniques categorised according to data types and task. Courtesy of Tominski et al. [**?**]

2.4 Future Work

At the end of most survey papers, it is common for the authors to discuss future areas of work that are discovered over the course of the survey. These challenges and research directions have been compiled using SurVis, an online literature browser [**?**, **?**]. This enables us to find a number of future research areas within Information Visualization.

The research topics found to have a high frequency (over 15%) are listed in this section. Each topic provides a percentage of recently *summarized* papers (40) that address a challenge found within the surveys.

1. **Evaluation (50%):** The most frequent topic discussed for open-research directions is visualization evaluation. This includes user studies, qualitative studies, quantitative studies, and longitudinal studies. There is a strong focus on perceptual surveys, that would clearly show where studies are limited within each topic.

2. **Missing Scenarios (40%):** Many topics point out vacant research scenarios within their classification tables. All of these topics are subject specific, and can be viewed using our literature browser [**?**].

3. **Scalability (35%):** Scalability is still a very important trend in visualization design at the moment. This includes large datasets, the ability to move between views clearly, reducing clutter, and improving visual understanding of complex views.

4. **Interaction Techniques (28%):** The use of interaction techniques continues to be an essential part of visualization design in the recent decade. Research that discusses ways to filter or manipulate visualization seem to be an important topic in the upcoming future.

5. **Extended Data Dimensions (25%):** Many surveys suggest that new data dimensions can be explored in the field. This differs from Item 2 by discussing the extension of a taxonomy.

6. **Tools (20%):** A number of papers describe the need for new tools for their domain which includes a need for both generic tools or frameworks to enable users to quickly use techniques over multiple pieces of software, or context sensitive tools to allow specific test or an understanding within the field.

7. **Tasks (20%):** 20% of papers suggest that a stronger focus needs to be placed on exploring different tasks within each topic. This would enable researchers more understanding

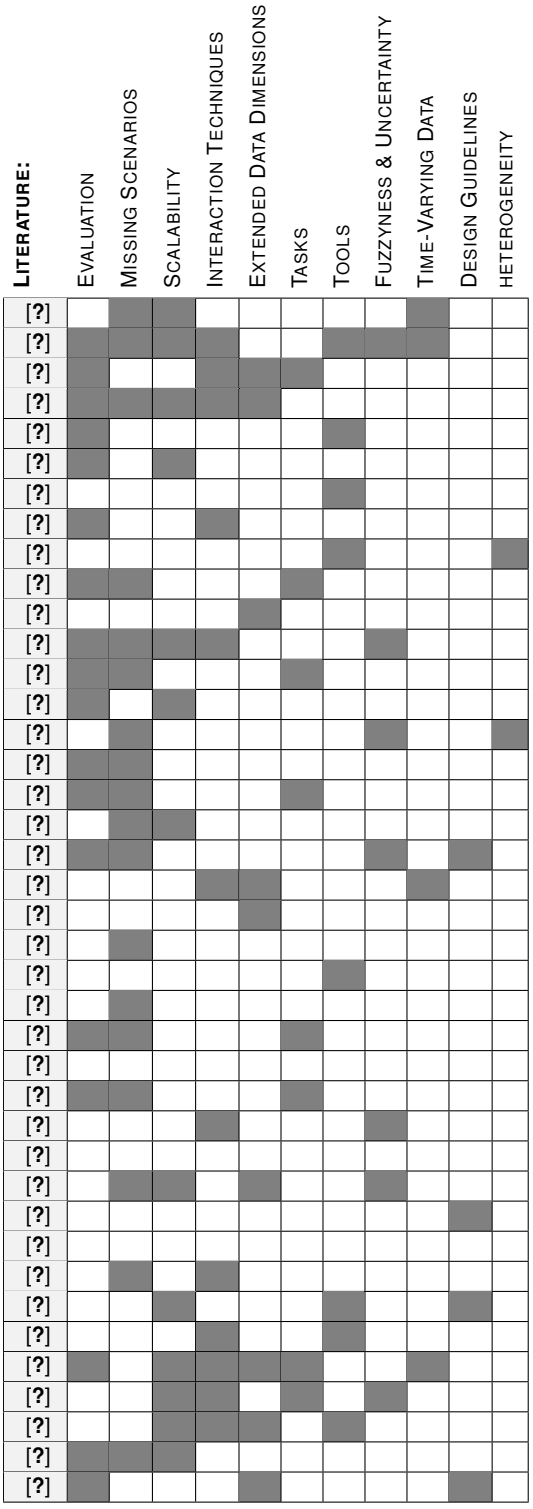


Table 2.6: The tables shows a breakdown of research directions discussed for each primary survey paper (highlighted green in Table **??**). The directions displayed represent research areas that are discussed in more than one survey. This table corresponds to the 1-N classification example shown in Figure **??** (B).

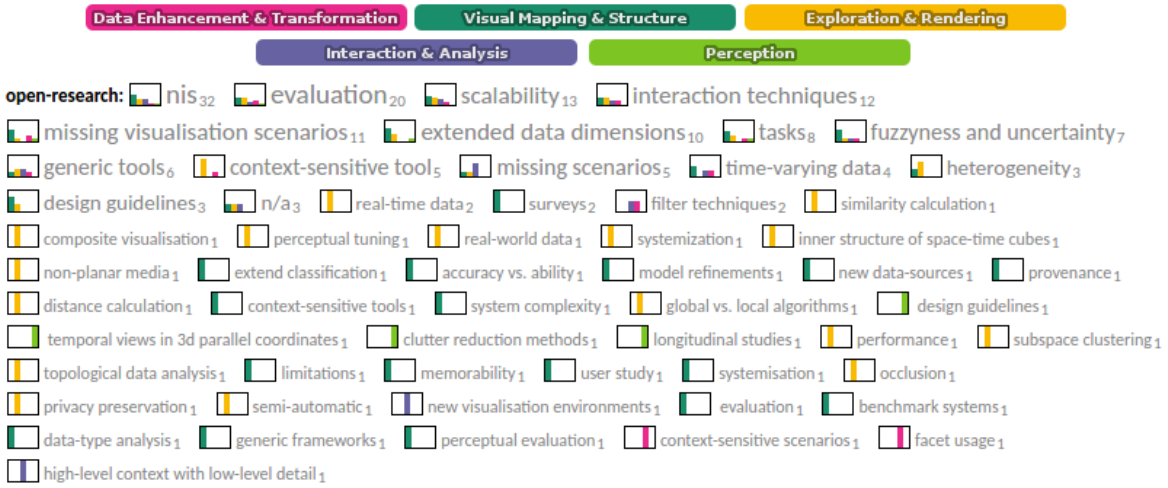


Figure 2.40: Open research keywords collected across all recent, reviewed survey papers (2010-2017). *'nis'* refers to papers not summarised. *'n/a'* refers to summarised papers with no explicit open-research. Collected Keywords are reviewed using the SurVis Literature Browser [**?**, **?**].

when a technique is appropriate, what approaches would produce the best output, and what can be gathered from visualization design choices.

8. **Fuzziness and Uncertainty (18%):** Fuzziness and Uncertainty are a growing topic within Information Visualization and the results of our survey shows that this is a positive step. Visualization aims to represent clear findings and it is therefore essential that and uncertainty is represented. This open research topic was mainly suggested for text-focused surveys and multivariate surveys. Although there are uncertainty surveys published, only one of these fulfills one of our topics [**?**], while the other two are SciVis papers [**?**, **?**], so this research area is still open for research.

Our chapter presents some interesting findings. Graph surveys and text surveys have a large quantity of survey work in recent years. This enables quite a large overview of the current landscape of the topics but will also make it difficult to justify the creation of new surveys in the field. There is a large quantity of user studies across many fields yet there is little evidence of their use. Perceptual surveys are a great tool to analyze and document user-studies within a field, with the two papers summarized in this category giving a greater understanding of their benefit and contribution to the field [**?**, **?**] (see Table **??** and figure **??**). We also look how these papers are used in the field. We found that over 50% of survey topics show a positive trend between surveys before and after 2010 (see Figure **??**).

2.5 Limitations

There are some important considerations when looking at the implementation of the Survey of Surveys. The SoS uses natural topic clusters to classify literature in the field of information visualization. This means that topics reviewed are naturally biased towards surveys that have been published. A second limitation is that open-research is only based on what is discussed within each survey and this does not necessarily fully represent the current landscape of the domain, as there is a possibility that papers have been presented that fulfill open research directions between the publication of the survey and the publication of the SoS. This means that the older the paper, the more uncertain we are that the open research areas have matured.

2.6 Conclusion

The SoS contributes a quantum step forward in literature surveys. We present a novel classification of survey papers that enables the reader to find recently published literature among a wide variety of topics. The classification also enables users to easily spot areas of open-research for survey publication, as well as an understanding of broad open research topics in the field of information visualization. The literature review provides a basic systematization of classification tables among the existing survey literature. It provides a valuable starting point for both newcomers and experienced researchers in visualization. We also believe it provides a valuable resource to readers outside of the information visualization and visual analytics communities.

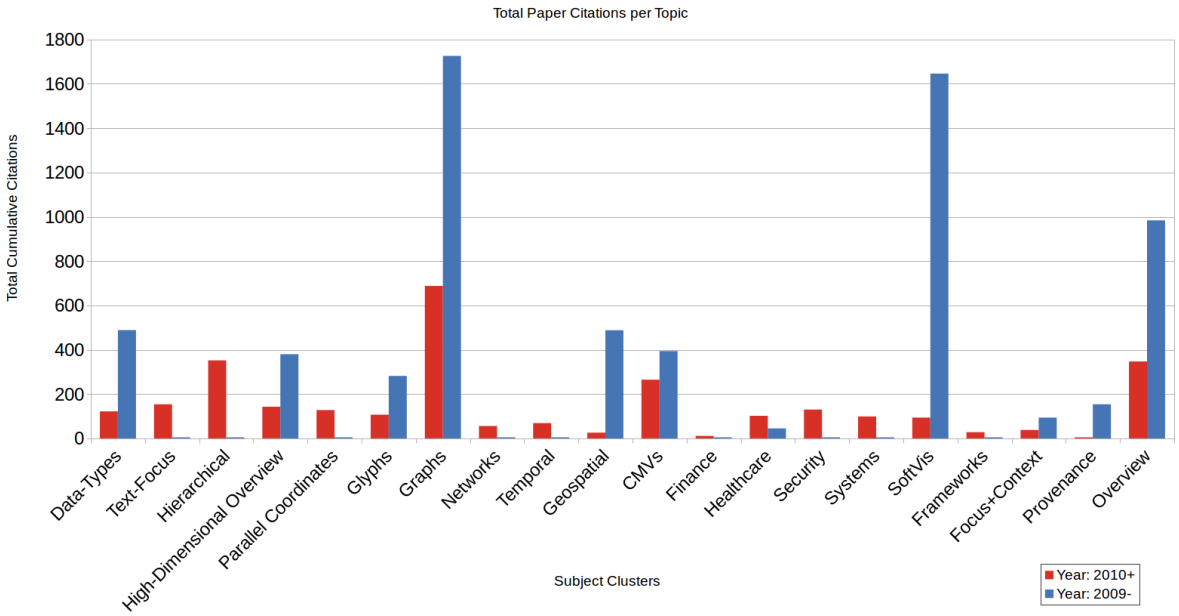


Figure 2.41: The graph provides a visualization of paper citations, ordered by topic, highlighting the difference between surveys published before and after 2010. The graph shows that 11 of the 20 show a positive trend, 10 of which are considered new subject clusters.

Chapter 3

Dynamic Maps

[**?**]

"A problem of choropleth maps is that the most interesting values are often concentrated in densely populated areas with small and barely visible polygons, and less interesting values are spread out over sparsely populated areas with large and visually dominating polygons."

— Ward et al, 2010

Contents

Chapter Abstract

Choropleths are a common and useful way of depicting area-coupled data on a geo-spatial map. One advantage they provide is combining area-based data accurately with geo-space. However perceptual problems arise when areas are too small, i.e when they only cover a few pixels or less. This is a very common occurrence when zooming or in densely populated areas like capital cities. We present a novel algorithm that ensures the user is able to observe area-based data coupled to geo-space based on their interactive level of zoom without distorting the original geo-spatial map. This is resolved by building a hierarchical data structure in which each area and its data is merged with one of its smallest neighbor recursively until only one polygon covers each contiguous region. The benefits are that the viewer can always view area-based data contained in the map regardless of how small any individual area becomes during interactive zooming. We break down each step of the algorithm and provide pseudo-code to enable reproducibility. We also discuss unique test cases that challenge the robustness of the algorithm with 30,000 polygons and 4,652,800 vertices as well as the performance.

3.1 Introduction and Motivation

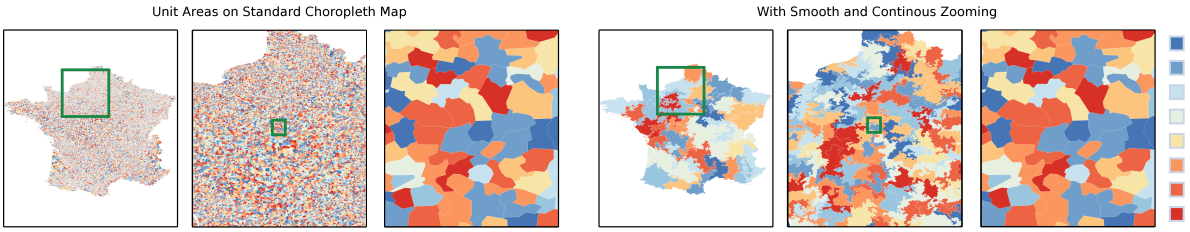


Figure 3.1: A comparison between a shape file representing France with over 30,000 administrative units and 729,565 vertices before and after the implementation of smooth zooming at 3 different levels of zoom, with minimum required screen space (m) of 1%. Mapped colors from colorbrewer color palette [**?**].

Learning that scale is such a prevalent unsolved problem in the the information visualization domain, this theisis examines the creation interaction techniques to synergise with zooming functionality. The topic our initial investigations lies with is one of geospatial visualization's oldest and most common technique, the choropleth map.

Choropleth maps can be defined as displays where data is aggregated using administrative units and normalized values [**?**]. Choropleths are ubiquitous for conveying area-based data on a geo-spatial map because they are intuitive and preserve geo-spatial information. However, because they do not distort geo-spatial boundaries, areas may be too small to perceive any data (see Figure **??**). This is especially true in the context of zooming where an area may not even cover a full pixel. Area-based data is often too dense to perceive in capital city regions. Ward *et al.* state, "*A problem of choropleth maps is that the most interesting values are often concentrated in densely populated areas with small and barely visible polygons, and less interesting values are spread out over sparsely populated areas with large and visually dominating polygons*" [**?**].

We focus on maintaining perceivable areas without map distortions by developing an area-merge algorithm that provides a user-controlled parameter, m, to display area units or area unit clusters that meet a minimum screen-space requirement. Rao and Card define such an adjust operation as "*...change the amount of contents viewed within the focus area without changing the size of focus area*" [**?**]. By introducing a hierarchical representation of the choropleth, we can update the display quickly and enable changes to the level of detail for the best visual experience. We call this a dynamic choropleth map. Our zooming is smooth and continuous. By this we mean there are no jumps, distortions, or disruptions during the zooming. The level of detail changes dynamically and interactively without distorting the geometry. Changes in zoom level must be smooth and not rely on distortion of the geo-space or any areas contained within.

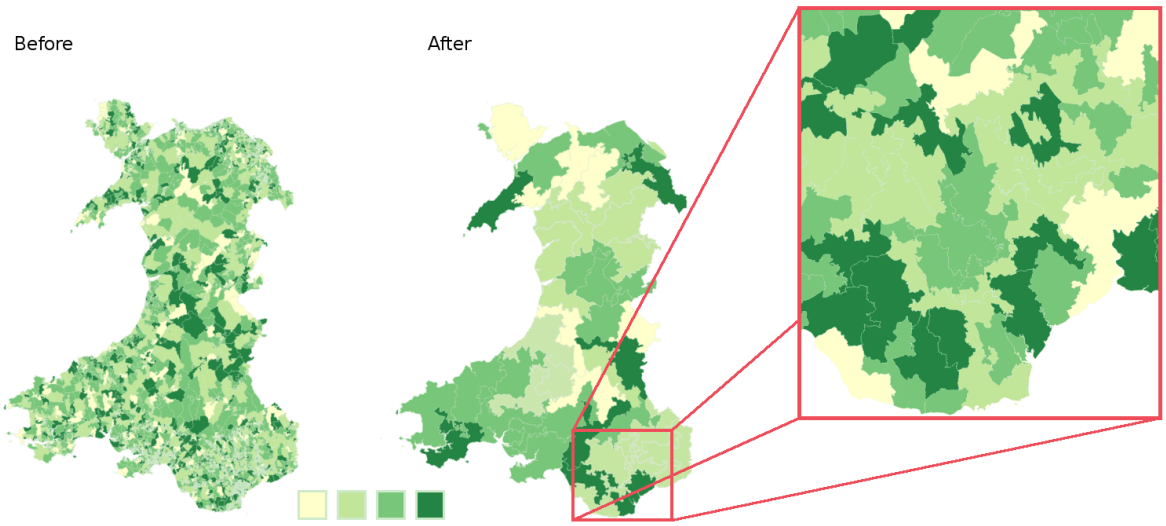


Figure 3.2: Example of the procedure applied to Wales [**?**]. The left image shows the original image with over 10,000 output areas having $4,652,800$ vertices [**?**], where we can see a dense clutter of indistinguishable areas in the south-east section. The right images shows the effects of the procedure at two different zoom levels (indicated by the red box), where m is 2%. Areas are color-mapped using colorbrewer color palette [**?**].

Our contributions include:

- A novel algorithm to interactively zoom smoothly, providing appropriate and perceivable levels of detail for choropleth maps.
- Providing a set of pseudo-code to enable reproducibility of the method.
- The application of our algorithm to complex, real-world shapefiles including those with over 10,000 unit areas and over 4.5 million vertices.

To provide this functionality, challenges must be overcome including developing an algorithm that detects when unit areas become too small, joining boundaries, building an appropriate area hierarchy, and zooming dynamically and continuously whilst preserving the traditional choropleth properties.

In Section **??**, we review previous work on interactive zooming and choropleth maps. Section **??** discusses the proposed methodology of the algorithm, a general overview of the procedure and the individual steps required. Section **??** discusses results and performance including benefits and limitations. Section **??** looks at potential future work and conclusions.

3.2 Background

d The Survey of Surveys for information visualization, found in Chapter**??**, identifies one related survey paper on clutter reduction, no related surveys on the topic of choropleths or surveys focused on geo-spatial zooming, and one survey focused on hierarchical aggregation [**?**]. Ellis and Dix provide a taxonomy of clutter reduction for information visualization and review 11 clutter reduction techniques including clustering, space-filling, and animation [**?**].

3.2.1 Zooming

Cockburn *et al.* review pan+zoom used in over 15 research papers, and examine overview+detail, zoom, and focus+context [**?**]. Rao and Card discuss the use of zooming for tabular information in the context of interactive manipulation of focus (zoom, adjust, and slide) [**?**]. We require that the view and geometry are not distorted in any way in our work. Jog and Shneiderman present the zoom bar and introduce a zooming approach based on zooming towards a fixed line within a starfield visualization [**?**]. This differs from our work that focuses on choropleths. Van Wijk and Nuij provide an algorithm for smooth and efficient zooming across 2D planes [**?**] and extend on this idea by looking at non-uniform scaling between two planes [**?**]. They derive an optimal camera path for smooth zooming and panning. This is likely the previous work most similar to ours. Their work does not consider regions that may be too small to perceive which differs from our work. Also the choropleth map is dynamic in our case. Javed *et al.* present a zooming technique titled PolyZoom where a user progressively builds a hierarchy of focus regions to zoom between [**?**]. Polyzoom focuses on different scales of maps separately whereas we endeavor to provide a continuous zooming method. Axelsson *et al.* tackle challenges addressing visualization between large scales of information for astronomical data using scale scene graphs [**?**] which differs from our work that focuses on a single scene that must be smooth and continuous. Google Maps provides a map of the earth which enables the user to zoom on user-selected areas. Moving between zooming levels comes with sudden, discontinuous transitions between levels of detail which we avoid [**?**]. Both Akelsson *et al.* and Google Maps process image data broken up into rectangular tiles. Our algorithm processes original unit areas and handles geo-spatial boundaries composed of vertices and edges.

Blanch and Lecolinet provide zoomable treemaps that pan and snap-zoom between different levels within a tree map [**?**]. Roberts *et al.* extend Van Wijk and Nuij's zooming work applying their smooth zooming algorithm to tree maps, and combine this with a smooth transition between levels of detail [**?**]. Our work differs from Roberts *et al.* as our approach maintains a smooth and continuous transition between zoom levels, and selects what to display based the

zoom level and a user-specified parameter. In addition, our work handles much more complex area-unit boundaries because it processes choropleths.

3.2.2 Choropleths

Digital choropleth maps have been produced prior to 1970 with the U.S Department of Commerce citing 10 choropleth mapping systems [**?**]. From our related work literature search we find previous work on choropleths focus on class intervals (or systems) rather than zooming. A class is defined as a mutually exclusive and non-overlapping set of grouped data whilst a class interval is defined as the selected width (or range of data) of each class [**?**]. Tobler questions the use of class intervals within choropleth maps by reviewing the use of inked area vs. white area to display values [**?**]. Brewer and Pickle provide a qualitative study on class intervals for choropleth maps comparing seven different methods [**?**]. Zhang and Maciejewski detect critical boundary cases within choropleth maps where statistical measures fall near the selected classification bounds [**?**]. This informs them of optimal selection of class intervals for data representation. Pickle presents a guideline for map design including color selection, legend design and smooth transition between color within area-units [**?**]. Slocum *et al.* provide a full chapter on Choropleth Mapping which includes 58 references [**?**] spanning 1957 [**?**] to 2006 [**?**]. They discuss decision-making behind classed and un-classed maps, appropriate color schemes, and designing the legend of the map [**?**]. Dykes and Brunsdon introduce new techniques for geographically weighted visualization using scalograms [**?**]. Each of these papers places emphasis on class intervals, whilst our chapter focuses on perceivable individual areas on a dynamic map.

Andrienko and Andrienko briefly survey the overall spatial distribution of data with diverging color scales in choropleth maps, and provide an example of animated choropleth map displays with small multiples [**?**]. We do not review color scales or the use of temporal data in choropleth maps.

Jern *et al.* use linked views to observe regional development data using both a choropleth map and tree map [**?**]. Our chapter focuses on adding a new dynamic feature to choropleth maps rather than combining them with other techniques. Dang *et al.* present a generalized map-based information tool for dynamic queries and brushing on choropleth maps [**?**]. Our work focuses on zooming rather than brushing. Li and Han look at applying the Lorenz curve to choropleth mapping to identify numerical trends [**?**]. We focus on user perceivability rather than new trends in data. Johansson *et al.* present a web-based visualization tool that combines the use of choropleth maps with dashboard functionality in order to review multifaceted information on climate change and adaption measures [**?**]. We focus on perceivability of unit areas, rather than the use of a choropleth map for climate change data. Speckmann and

Verbeek present necklace maps which present choropleth maps with juxtaposed proportional symbol maps that allow the user to understand size data without distorting the topological view [**?**]. We develop interactive, smooth zooming in order address similar issues.

Rittschof and Kulhavy present a user-study which includes a comparison of choropleth maps and cartograms. Cartograms are a different class of related work considering a wide range of techniques (Gastner-Newman [**?**], Dorling [**?**], etc.) which use distortion to convey data. We want to avoid introducing geo-spatial error into the map in our technique. Their results found choropleth maps were associated with greater recall of information [**?**]. Kasper review the effectiveness of Gastner-Newman diffusion cartograms [**?**, **?**] for the representation of population data, which includes a comparative experiment against thematic maps (choropleth with overlayed circle maps). The results report that the thematic maps are more efficient and effective, specifically with complex tasks [**?**]. Sun and Li review the effectiveness of cartograms for the representation of spatial data, which includes a comparative experiment against thematic maps including choropleths. The results indicate that the thematic maps are more effective representing quantitative data, whilst cartograms were more effective with qualitative data [**?**].

To the best of our knowledge, no previous work focuses on dynamic and continuous zooming of choropleth maps while maintaining perceivable area units without distortion.

3.2.3 Cartographic Generalization

Slocum *et al.* provide a full chapter on Cartographic scale and generalization [**?**]. The chapter defines generalization as: *"the process of reducing the the information content of maps because of scale change, map purpose, intended audience, and/or technical constraints"*, and reviews models of generalization include the models of Robin *et al.* [**?**] and McMaster and Shea [**?**]. Slocum *et al.* define the fundamental operations of generalization as simplification, smoothing, aggregation, amalgamation, collapse, merging, refinement, exaggeration, enhancement, and displacement. Our algorithm uses recursive amalgamation on a per-area basis.

Elmqvist and Fekete provide a survey on hierarchical aggregation for information visualization [**?**]. The survey only provides one spatial aggregation techniques by Andrienko and Andrienko (discussed below). Andrienko and Andrienko briefly discuss aggregation with earthquake occurrences in Turkey [**?**]. They use a density map to aggregate the occurrences per rectangular grid cell. Andrienko and Andrienko's generalization approach looks at point data, whilst we focus on areas. Zhang *et al.* present a novel visualization technique titled 'TopoGroups' [**?**] used to group spatial data into hierarchical clusters to minimize visual clutter.

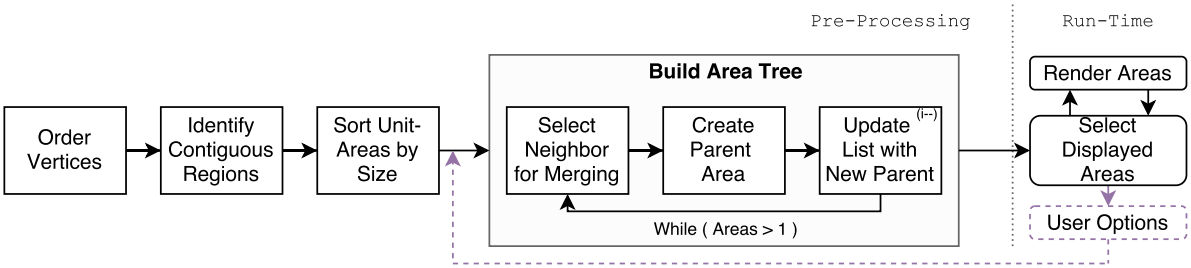


Figure 3.3: The pipeline for the area amalgamation algorithm. After loading the shapefile, polygons are partitioned based on area contiguity, and sorted within islands (or land masses) based on their size. A recursive function is then used to identify new parent areas and their boundaries until their are no remaining neighbors to merge. See section **??** for details.

Boundaries are used to present data topics as a stipple line, where the ratio of a stipple represent that of the data. We focus on polygon unification rather than point data.

Regnauld and Revell discuss their automatic amalgamation method used in producing the ordnance survey's scale maps [**?**]. The paper uses a number of generalization techniques to select clusters (triangulation, proximity, and edge filtering) and manipulate the clusters to give a visually clear representation of amalgamated buildings. Our chapter looks at areas rather than buildings and is used for only contiguous areas. Li *et al.* review amalgamation of buildings based on the Gestalt principles of design [**?**] which include separation, length, and area thresholds as well as similarities in shape, size and orientation. Our amalgamation technique does not allow for any separation and unites two areas instead.

3.3 Methodology

We begin with an overview of our methodology before discussing each step in detail. The algorithm is based on the premise that each area, starting with the unit areas, can be merged with its closest neighbor from smallest to largest to create a smooth and continuous transition for perceptible areas.

3.3.1 Method Overview

In order to effectively enable smooth and continuous zooming at run-time, we use pre-processing. We build a hierarchical data structure before displaying the choropleth. For this we have created a pre-processing pipeline shown in Figure **??**. We first load each unit area represented

by a polygon, p. A polygon p is a list of vertices: $p = \{v_0, \ldots, v_n\}$. We then update the order of each unit-area's list of vertices to ensure that they are in clockwise order. The next step is to identify contiguous regions. Here we separate contiguous regions into islands (or land masses) which enforces topological continuity. Once each contiguous region is identified, each unit-area within the same contiguous region is sorted by size since scale is an important part of the algorithm. It is more efficient to sort before building the hierarchical data structure.

The hierarchy construction is a recursive algorithm broken down into three sub-routines. As the regions are pre-sorted from smallest to largest, we know the first area merge candidate (p_1) is at the front. We must then find the second merge candidate (p_2) by selecting one of p_1's neighbors using a distance function. When we have found a merge pair (p_1, p_2) we identify both the shared (b_s) and non-shared (b_{ns}) boundary of each, and combine such that p_1 and p_2 unite using only their shared boundary to create a new area P.

$$P = (p_1 \cup p_2) - (p_1 \cap p_2) \tag{3.1}$$

This is stored as a parent in the hierarchical data structure. When this is done, we can then remove the p_1 and p_2 from the merge candidates list and insert the new parent P into the list preserving sorted order (by size). When there are no remaining neighbor candidates, the hierarchy is complete. When this is done for each contiguous region, we have the necessary hierarchical data structures for smooth zooming and clustering.

With the hierarchies built, display is relatively simple. By specifying a desired minimum screen space, m, using the current zoom level and comparing that to each tree node's size using a depth-first search (DFS) in the hierarchy, we can select the appropriate polygons to display. An example of the results can be found in Figures **??** and **??**.

3.3.2 Order Area Polygon Vertices

Our first step is to order the original vertex data from the shape file. This is important in order to reduce complexities in later stages. It allows us to simplify the identification of and unification of boundaries ($p_1 \cup p_2$). For this we use the shoelace formula (also known as Gauss's Area Formula or Surveyor's Formula), which allows us to derive both the area (useful for later) and the orientation [**?**].

$$a = \frac{1}{2}\left|\sum_{i=1}^{n-1} x_i y_{i+1} + x_n y_1 - \sum_{i=1}^{n-1} x_{i+1} y_i - x_1 y_n\right| \tag{3.2}$$

The notation x and y refer to the coordinates of each vertex and n refers to the number of vertices in p. If we remove the absolute value, we can deduce that if the area is negative, the vertex list is counter-clockwise, and we can reverse the list order. Unit-area's with multiple contiguous regions are also split up to enforce topological continuity. We process these islands (or land masses) as individual areas. We must also test for uncommon inner rings or any other vertices related to the shape. These can be saved in a separate list to aid in rendering, however these must also be searched during boundary processing, as a ring found in unit-areas is usually formed as a result of a fully surrounded unit-area. In our Wales example (Figure **??**) we find 31 instances of inner rings out of 30,000 polygons.

3.3.3 Identifying Adjacent Neighbors & Contiguous Regions

After ordering each unit-area's vertex lists, we can identify the contiguous regions. This is important for us in order to prevent a merge of two islands. The most important consideration is identifying what is classified as a neighbor. We provide pseudo-code for this in Algorithm **??**.

Algorithm 1 - Are polygons neighbors?

Input–
p_1 : polygon one
p_2 : polygon two

```
 1: procedure ISNEIGHBOR(p1, p2)
 2:     if isOverlapping( p1.boundingBox(),
 3:                       p2.boundingBox() ) then
 4:         return commonVertices(p1, p2)
 5:     endIf
 6:     return FALSE
```

Local Variables– $counter$: number of matching vertices
$MIN = 2$: minimum number of matching vertices required to be neighbors

```
 1: procedure COMMONVERTICES(p1, p2)
 2:     counter = 0
 3:     for i = 0; i < p1.length(); i++ do
 4:         if p2.intersects(p1[i]) then
 5:             counter++
 6:             if counter ≥ MIN then
 7:                 return TRUE
 8:             endIf
 9:         endIf
10:     endFor (i)
11:     return FALSE
```

Desc: *Compares two polygons and tests for overlapping boundaries,* $p_1 \cap p_2$. *Returns true if the minimum number of common vertices are found.*

We first test p_1 and p_2's bounding boxes for overlap. By comparing Axis Aligned Bounding Boxes (AABB's) which use the maximum and minimum values for each axis of the areas p_1 and p_2 [**?**], we ensure the in-depth neighbor checking is applied to as few areas as possible.

Algorithm 2 Contiguous Regions

Input– L_p : non-empty list of polygons
Output– $L_{islands}$: list of contiguous islands (or land masses)
Local Variables–
island: current island
neighborFound: flag designating if neighbor is part of existing island

```
 1: procedure IDENTIFYCONTIGUOUSREGIONS(L_p)
 2:     // For each polygon
 3:     while !L_p.isEmpty() do
 4:         // Assume Island
 5:         island = L_p.popFirst()
 6:         // For each island
 7:         for i = 0; j < L_islands.length(); i++ do
 8:             // For each polygon on each island
 9:             for j = 0; j < L_islands[i].length(); j++ do
10:                 if isNeighbor(island, L_islands[i][j]) then
11:                     neighborFound = true
12:                     break
13:                 endIf
14:             endFor (j)
15:             if neighborFound then
16:                 island.appendList(L_islands[i])
17:                 L_islands.removeIslandAt(i)
18:                 i- -
19:             endIf
20:         endFor (i)
21:         L_islands.append(island)
22:     endWhile
23:     return L_islands
```

Desc: *Partitions a list of non-contiguous polygons into separate contiguous regions such as islands and land masses. A contiguous region has connected neighbors where no area is completely separated by water.*

If p_1 and p_2's AABB intersect, we test their vertex lists for common points, where common points are considered identical coordinates. Algorithm **??** uses a simpler approach where we assume that all points have a matching point in a neighbor's vertex list. If areas with long straight edges (like some US states) are used to define unit-areas, we find cases where we need to use a second test to identify whether a point intersects a boundary edge (examples of this include T-junctions). We define neighbors as two polygons with at least two unique common vertices. We do not consider one common vertex as a boundary edge. The start and end of a shared boundary b_s must also be considered the end and start of a non-shared boundary $b_n s$ to enforce topological continuity of the unit areas.

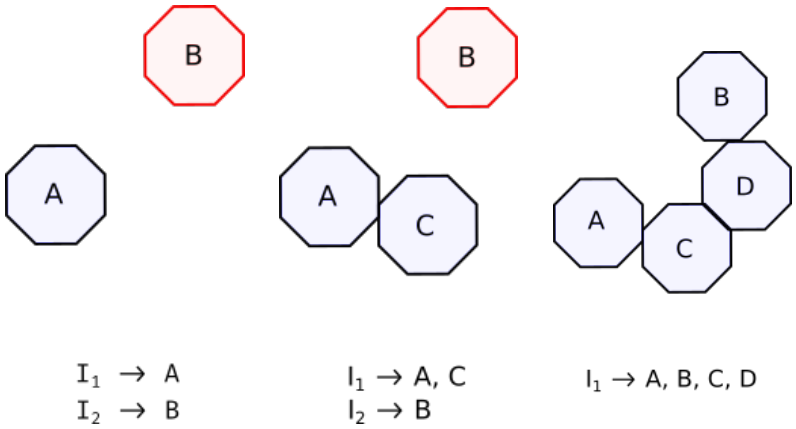


Figure 3.4: Visual example of the contiguous regions procedure. This shows how a potential contiguous region can be derived over three steps. See Section **??**.

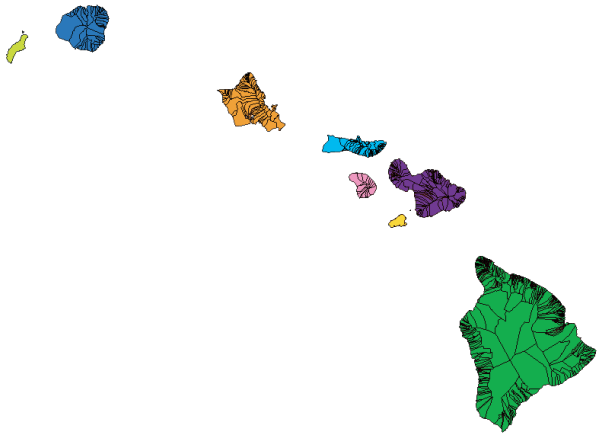


Figure 3.5: Example of the contiguous regions procedure applied to the Ahupua'a boundaries of the state of Hawaii [**?**]. There are 10 visible contiguous islands each with their own color. See Section **??**.

Now that we can identify adjacent neighbors, we identify the contiguous regions. Pseudo-code is provided in Algorithm **??**. We assume that our first unit-area is an island and test this against every other island. If an island contains a neighboring unit-area, we know that every other region on that island is also linked. Knowing this, we can merge the two polygon lists and continue our search. See Figure **??**. It is important that we do not finish the search here as our new unit-area may connect multiple islands together. Once this is done for each unit-area, we have identified each contiguous region and each of these can be sorted based on their size. Figure **??** provides an example of the procedure, whilst Figure **??** shows a visual result of this step.

3.3.4 Building the Hierarchical Data Structure

We use a recursive procedure to create a hierarchical data structure. A hierarchy is created for each contiguous region, where each area (p_1) is merged with it's closest neighbor (p_2). Distance is measured using a general and flexible metric described in Section **??**. We start with a merge candidate list filled with the sorted unit-areas (for one contiguous region). The list is sorted by size. As mentioned in Section **??**, there are three main sub-routines: neighbor selection, creating the parent area (P), and updating the merge candidate list. If only a single unit-area remains in the merge candidate list, no further merges can be processed and we have finished the procedure. Here we denote p_1 as the first area merge candidate, p_2 as the second merge candidate and parent P (Equation **??**).

3.3.5 Boundary Neighbor Selection & Amalgamation Criteria

In order to select an appropriate neighbor to join, we use a general and flexible distance metric for amalgamation evaluated between neighboring areas. We use this to measure a distance where the closest distance is considered the optimal selection for a neighbor. The measure consists of four constituents: Smallest area (a), euclidean distance between centroids (d), value variance (α), and shared boundary resolution (b_s). We formulate the measure as:

$$D = w_a.\frac{a}{a_{max}} + w_d.\frac{d}{d_{max}} + w_\alpha.\frac{\alpha}{\alpha_{max}} + w_{b_s}.(1 - \frac{b_s}{b_{s_{max}}}) \tag{3.3}$$

The distance metric includes weight co-efficients which enable the user to customize the importance (w) of each criteria as an option, with a default weight 0.5 for a, and a $\frac{50}{3}$ weight for d, α, and b_s. We define the criteria as:

- Smallest area (a). The criteria tests the size of a neighbor. Searching for small areas is the primary objective of the procedure and it is therefore important to take this into account during the distance measure. By doing this we reduce the number of small areas at a faster rate. We discuss how the area is calculated in detail in Section **??** (Equation **??**). a_{max} is considered the area of the canvas' bounding box.
- Euclidean distance (d). This represents the shortest distance between two centroids. By taking the distance between centroids into account, we can enable more natural polygon formations to form. To calculate this we can use ($\sqrt{(|p_1(c_x) - p_2(c_x)|)^2 + (|p_1(c_y) - p_2(c_y)|)^2}$). The term d_{max} is the largest distance between all centroids.

- Data Value Similarity (α). Data is an important aspect of cartography and is considered when agglomerating areas. In order to factor it in the distance metric we look at the variance between the values of p_1 and p_2 ($|p_1(\alpha) - p_2(\alpha)|$). α_{max} is the largest data value in the data range.

- Shared Boundary Resolution (b_s). Unlike the other criterion, we favor a larger shared boundary resolution. The shared boundary resolution refers to the topological length of a shared boundary, where a larger shared boundary defines a closer unification between two areas. This is calculated by running our merge algorithms early (refer to Section **??** for more detail) and normalizing it over the largest resolution area in the tree ($b_{s_{max}}$). Once this is done, we subtract the normalized value from 1 to impose a stronger weight for larger shared boundaries.

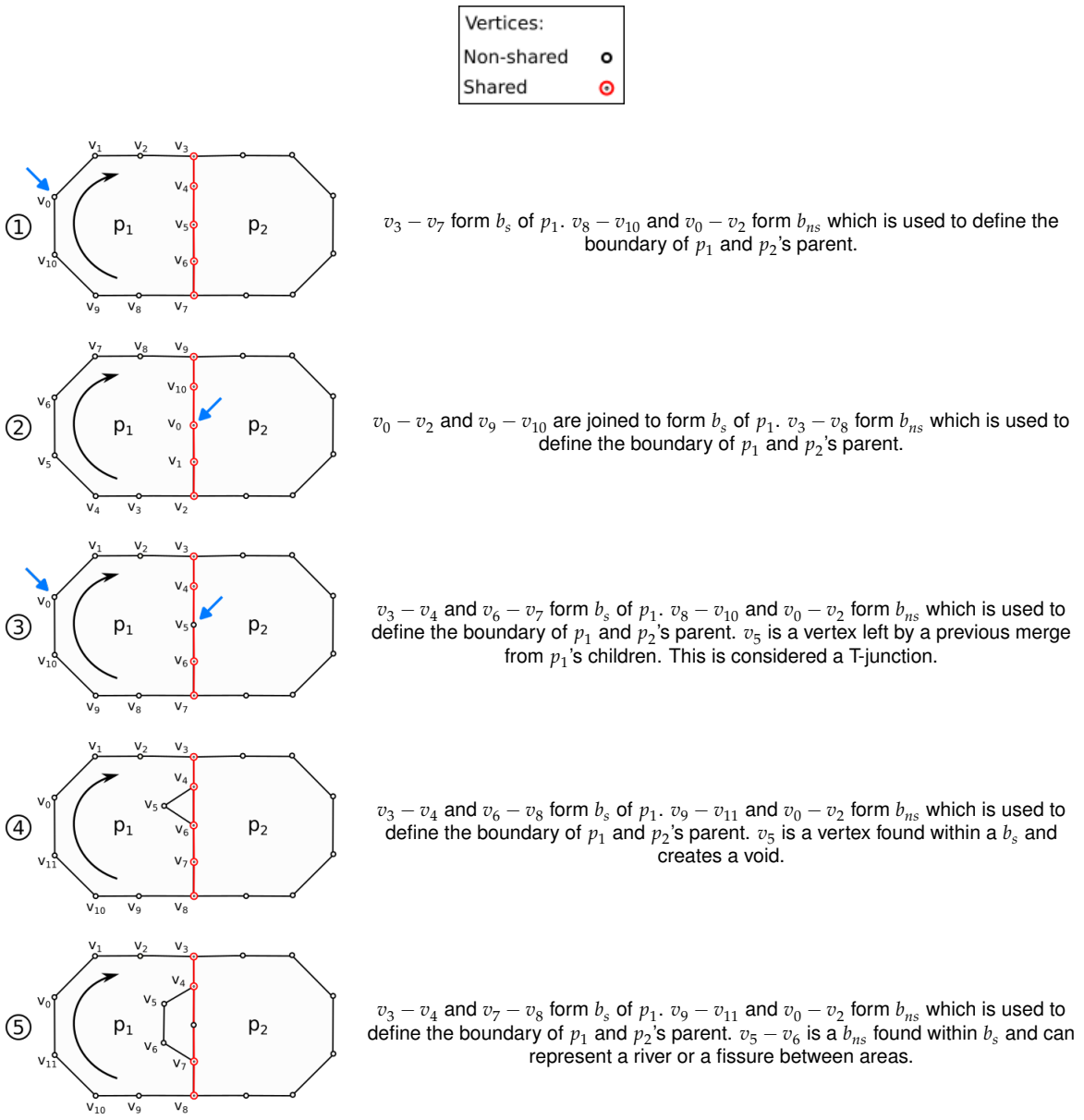
Using these criteria, we can select an optimal amalgamation candidate. We also provide the user the freedom to modify the criteria by using weighted coefficients. These can be modified after the procedure has been completed. This is a general and flexible distance metric because the distance metric itself is not a focus of the chapter. Many such metrics have been studied in great detail [**?**].

3.3.6 Creating Parent Area

Creating P includes 3 steps: (1) identify b_s and b_{ns} of each area's merge pair, (2) combining b_{ns} of the p_1 and p_2 for the boundary of the parent area P, (3) linking p_1 and p_2 to P for use in the rendering stage.

There are configurations which can cause unexpected challenges with the boundary identification. Firstly, the vertex list of each area is ordered but there is no given information about shared boundaries. This means that b_s can be found at any point within a vertex list, and can also start at any point with a vertex list. If our boundary search starts on b_{ns} and we search the vertices in clockwise order, as in case 1 of Figure **??**, we can assume that the first common vertex is the boundary start. This is not the case for a first vertex found on b_s. In order to render the boundary correctly, we must not only identify b_s but also identify the start and end points of the boundary. Figure **??** illustrates various cases identified for b_s identification between two neighboring areas p_1 and p_2.

Due to voids and fissures representing by rivers or other geographical features, finding the start and end points of b_s can become complicated even when testing the entire vertex list. For example, if a vertex list begins on b_s that includes a fissure of n vertices, the selection of the b_s' beginning and end indexes becomes less obvious.



① $v_3 - v_7$ form b_s of p_1. $v_8 - v_{10}$ and $v_0 - v_2$ form b_{ns} which is used to define the boundary of p_1 and p_2's parent.

② $v_0 - v_2$ and $v_9 - v_{10}$ are joined to form b_s of p_1. $v_3 - v_8$ form b_{ns} which is used to define the boundary of p_1 and p_2's parent.

③ $v_3 - v_4$ and $v_6 - v_7$ form b_s of p_1. $v_8 - v_{10}$ and $v_0 - v_2$ form b_{ns} which is used to define the boundary of p_1 and p_2's parent. v_5 is a vertex left by a previous merge from p_1's children. This is considered a T-junction.

④ $v_3 - v_4$ and $v_6 - v_8$ form b_s of p_1. $v_9 - v_{11}$ and $v_0 - v_2$ form b_{ns} which is used to define the boundary of p_1 and p_2's parent. v_5 is a vertex found within a b_s and creates a void.

⑤ $v_3 - v_4$ and $v_7 - v_8$ form b_s of p_1. $v_9 - v_{11}$ and $v_0 - v_2$ form b_{ns} which is used to define the boundary of p_1 and p_2's parent. $v_5 - v_6$ is a b_{ns} found within b_s and can represent a river or a fissure between areas.

Figure 3.6: Different cases for b_s and b_{ns} identification. Case 1 displays the basic case where a whole boundary is found in contiguous order. Case 2 provides a contiguous order, but is split due to the location of p_1's vertex list start index. Case 3 displays a T-junction which splits b_s into two segments. This could be resolved by point-line intersection testing. Case 4 and 5 represent voids and fissures which cannot be resolved by point-line intersection, with the fissure having a possible size of $b_s.length - 2$. We look at the length of common vertex chains to determine the start and end of b_s detailed in Section **??**.

Algorithm 3 Identify Boundary Range

Input–
$start$: starting index of shared boundary line
end : last index of shared boundary line
V_c : current polygon vertices in clockwise order
V_n : neighbor polygon vertices in clockwise order
Local Variables–
$longestC$: found by comparing distance between common vertices
$common$: list of found commonVertices

```
 1: procedure IDENTIFYBOUNDARYRANGE(start, end, V_c, V_n)
 2:     for int i = 0; i < V_c.length(); ++i do
 3:         if V_n.contains(V_c[i]) then
 4:             common.append( V_c[i] )
 5:         endIf
 6:     endFor (i)
 7:     longestC = longestSharedBoundaryChain( common, V_c )
 8:     *end = V_c.indexOf(common[longestC])
 9:     *start = V_c.indexOf(common[longestC.next()])
10:     return
```

Desc: *Identifies b_s for V_c with V_n as a neighbor. Required for parent node.*

We provide our boundary identification process in Algorithm's **??** and **??** which identify the start and end vertices of b_s. Firstly, we search and identify every common vertex between the area neighbors. As discussed in Section **??**, we assume that every common vertex has a matching vertex in their neighbor's vertex list, whilst shape files with simpler boundaries may need an additional point to line intersection test (T-junctions). From these vertices we can identify the beginning and end indexes of b_s (a common boundary between p_1 and p_2) by looking at the length of each common vertex chain. We use a heuristic that any voids and fissures found on b_s will be smaller in length compared to b_{ns} and therefore the longest chain between two common vertices signifies the chain between the end and the start of b_s. Figure **??** provides a visual presentation of boundary identification on some test cases encountered. This method handles cases with voids and fissures between neighboring polygons, as well as complications that can be caused by the T-junctions that may arise. For our Wales example in Figure **??** with over 10,000 unit areas (over 20,000 merges) and 4.5 million vertices we found 11,112 individual error cases caused by voids, fissures, and T-junctions. This means a non-trivial case is found in over 55% of the merges between p_1 and p_2.

Knowing b_s's start and end indexes, we can easily separate the boundaries into b_s and b_{ns}. We can then combine the b_{ns} of an p_1 and p_2 in clockwise order to create the new parent area P. An example can be found in Figure **??**.

Once P's vertex list is updated, we create pointers that enable P to find it's children. This is important to enable traversal and selection within the hierarchical data structure. Algorithm's **??** and **??** detail this process.

Algorithm 4 Longest Shared Boundary Chain

Input–
$common$: list of found commonVertices
V_c : current polygon vertices in clockwise order

```
 1: procedure LONGESTSHAREDBOUNDARYCHAIN( COMMON, V_c )
 2:     if isLongestChain( V_c, &longest, V_c.indexOf(common.last()),
 3:                                V_c.indexOf(common.first()) ) then
 4:         longestIndex = common.length()-1
 5:     endIf
 6:     for i = 1; i < common.length(); i++ do
 7:         if isLongestChain( V_c, &longest, V_c.indexOf(common.at(i-1)),
 8:                                V_c.indexOf(common.at(i)) ) then
 9:             longestIndex = i
10:         endIf
11:     endFor (i)
12:     return longestIndex
```

New Input–
$longestL$: The current longest distance between two common vertices
$currI$: current index to test
$nextI$: next index to test
Local Variables–
$length$: length of current chain

```
 1: procedure ISLONGESTCHAIN(V_c, LONGESTL, CURRI, NEXTI)
 2:     length = nextI - currI
 3:     if length < 0 then
 4:         length = length + current.size() - 1
 5:     endIf
 6:     if *longestL < length then
 7:         *longestL = length
 8:         return true
 9:     else
10:         return false
11:     endIf
```

Desc: *Identifies the longest absence of a common vertex. We can assume that this signifies the beginning and end points of* b_s.

3.3.7 Updating the Sorted List with the Parent

We update the list preserving the sorted areas. We first remove the p_1 and p_2 from our merge candidates list as each area can only be merged with one other area. Then we can insert P into the list in sorted position based on its size. The procedure for building the hierarchical structure is found in Algorithm **??**.

Algorithm 5 Build Binary Tree

Input–
L_{contig} : contiguous list of polygon sorted by area
Local Variables–
$neighborI$: index of selected neighbor
p : parent node of two neighbor areas

```
 1: procedure BUILDBINARYTREE(L_contig)
 2:     if L_contig.length() > 1 then
 3:         //Neighbor Selection
 4:         neighborI = selectNeighbor(list) //neighbor of list.first()
 5:         //Create Parent Area
 6:         p = new Node()
 7:         p.identifyVertices(L_contig.first(),L_contig.at(neighborI)))
 8:         p.setLeftChild(L_contig.first())
 9:         p.setRightChild(L_contig.at(neighborI))
10:         L_contig.first().setParent(p)
11:         L_contig.at(neighborI).setParent(p)
12:         //Update Sorted List with Parent
13:         updateList(L_contig, p, i)
14:         return buildBinaryTree(L_contig)
15:     endIf
16:     // Base Case -> L_contig == 1
17:     return L_contig
```

Desc: *Builds hierarchy of polygons using a list of merge candidates recursively.*

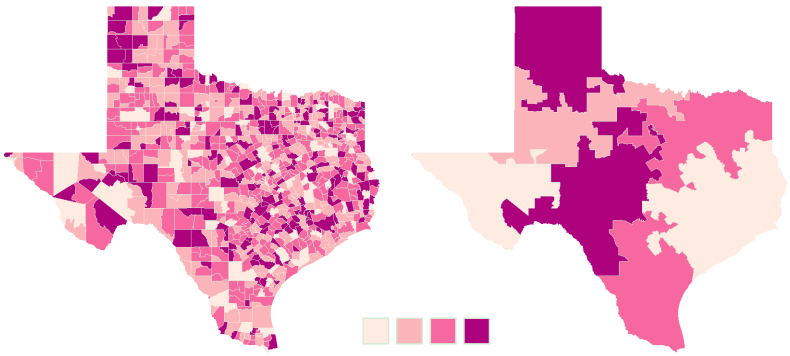


Figure 3.7: Example of derived parent area's from original unit-areas. The example uses the State of Texas and shows multiple boundary merges [**?**] so that m is 15%. See section **??**. Our example uses a color palette from colorbrewer [**?**].

3.3.8 Selecting Visible Boundaries

We select visible areas and boundaries based on a minimum area requirement, m, relative to the current screen space. As the screen space coverage changes based on the movement of the dynamic zoom level, we render different areas based on a zoom level and area size. The DFS identifies the smallest nodes in the tree that meet the minimum area size requirement. If any parent node is larger than the m, we test two criteria. (1) If the area is a leaf node, we can render the current node. (2) If either the left child or right child is smaller than m, then the current parent area is the smallest unit that meets the area requirement and is rendered. Completing the DFS will render only the smallest area within each branch that is larger than m. An illustrated example of this search can be found in Figure **??**.

3.3.9 Storing Values of Amalgamated Areas

The Modifiable Areal Unit Problem (MAUP) [**?**] is an important aspect to consider when discussing the modification of boundaries or values. We address this by providing the user options to modify calculation of aggregated values as well as the weighted distance metric discussed in Section **??**. The data is linked to the administrative areas during the initial loading of the shape files. Before the area tree is built, the user can select the type of value amalgamation. This enables the user to choose options of sums, qualitative frequencies, and averages. When amalgamating values using sums, the value of P can be calculated as $P(\alpha) = p_1(\alpha) + p_2(\alpha)$. Qualitative values are calculated using frequencies. Using a DFS, P can count the frequency of each value for each leaf node and use the value of the most frequent of the leaf nodes. This is useful for categorical data. The average and weighted av-

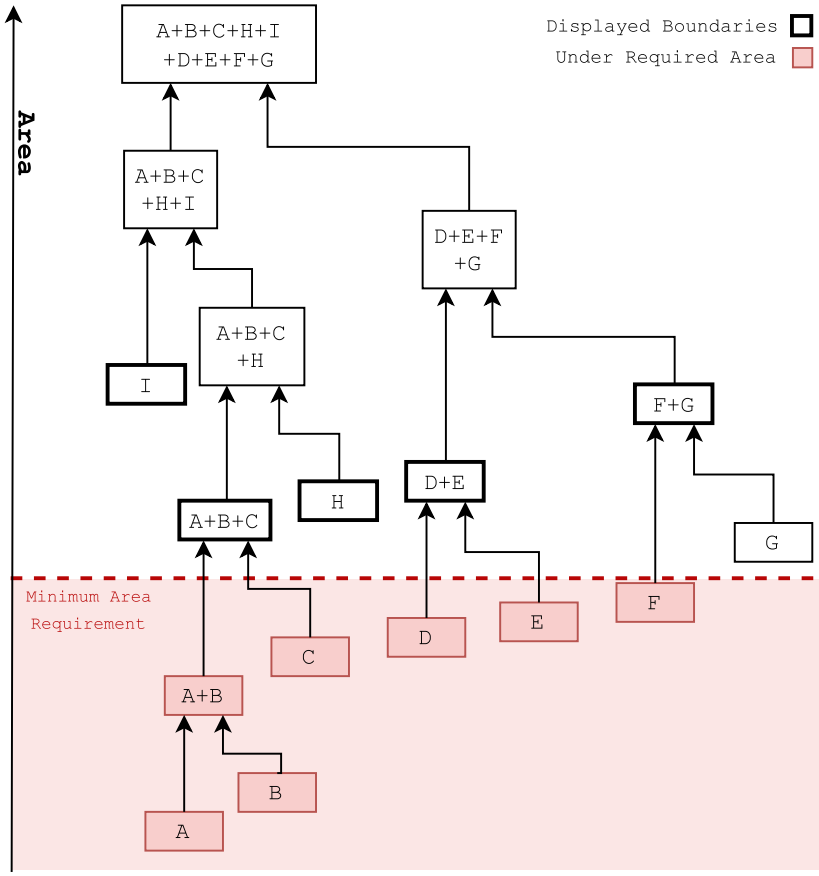
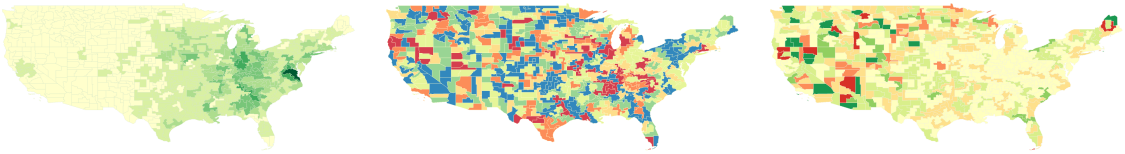


Figure 3.8: An example of areas being selected and rendered. An area is only rendered if one or both child nodes are smaller than the minimum area requirement, m. Otherwise, perform a depth-first search until a leaf node is identified. In this example, **I, A+B+C, H, D+E,** & **F+G** are selected to be rendered. See Section **??**.



(a) Sum **(b)** Frequency **(c)** Average

Figure 3.9: 1 value-set displayed using 3 different base-calculation types using US counties (m=0.3%). (a) Represents using the sum to calculate the new values (sums). (b) Uses the highest frequency to represent values (qualitative data). (c) Uses the average of the value from all leaf nodes. See Section **??**.

erage can also be calculated using a DFS, by calculating the sum, $P(\alpha) = \sum_{i=0}^{i=n} \frac{p_i(\alpha)}{p_i(a)}$, where p denotes a leaf node in the tree. Examples are shown in Figure **??**.

As well as these value criteria, these can be normalized at the rendering stage. Some examples of these normalization techniques include area ($\frac{P(\alpha)}{P(a)}$), population ($\frac{P(\alpha)}{P(\kappa)}$), as well as any ratio ($\frac{P(\alpha)}{P(\delta)}$).

Although the normalization can be turned on and off after the area tree is built. In order to change value representations, the build area tree procedure is re-run.

3.4 Results and Performance

The desktop used to test this implementation features an Intel i5-4460 at 3.2GHz with 16GB of RAM and a GeForce GTX960. The implementation is developed using the Linux Mint 18 environment and the C++ framework of Qt. The software uses the Geo-spatial Data Abstraction Library to read the Shape File's unit-area information [**?**] and the OpenGL library to render the results.

We test 5 different shape files of varying resolution including US Counties, Japan, Italy, Wales and Germany found using the Global Administrative Areas website [**?**]. There is a large variance in the number of areas, average number of vertices, total contiguous regions, and coordinate space range. We know of no closely related previous algorithm that we can compare performance with. See Figures **??** to **??** for results imagery. See the accompanying video for more dynamic results.

The performance is not only reliant on number of unit areas but also the complexity of unit areas, and the total number of contiguous regions. A summary is found in Figure **??**. Pre-processing can require a few minutes however it is only a one-time cost.

We found that different shape files for the same region would garner inconsistent topologies, which even includes the contiguity of the unit-areas. This makes it impossible to compare our fully merged areas to already existing shape files as a way of testing the topology preserving nature of our implementation.

3.5 Conclusion

We introduce a novel method of smooth and continuous zooming by exploiting a hierarchical data structure to merge areas based on their sizes and shared boundary. The shared boundary is found by first comparing the vertex list of two neighboring areas and finding the longest vertex chain between common vertices. We then render only the perceivable areas

Shape File	Number of Areas	Total Vertices	$\frac{\text{Vertices}}{\text{Area}}$	Average FPS, m = 5%
US Counties	3,134	51,891	16.56	30
Japan	3,223	869,386	269.744	21
Italy	8,946	966,206	108.004	9
Wales	10,355	4,652,800	449.32	5
Germany	12,416	1,934,800	155.779	6
France	37,227	729,556	19.597	4

Table 3.1: The results of performance. We present some attributes of each shape file, performance times broken into separate sections of the procedure, and the average FPS. The FPS is set to a minimum required screen space of 5% for polygon rendering.

or area clusters based on the current zoom level and screen space. This method of rendering improves perceptability whilst still providing an understanding of the underlying data without distorting the map. This enables the user to zoom without any distortion to the geometry and enables clear perceivable choropleth data for the user.

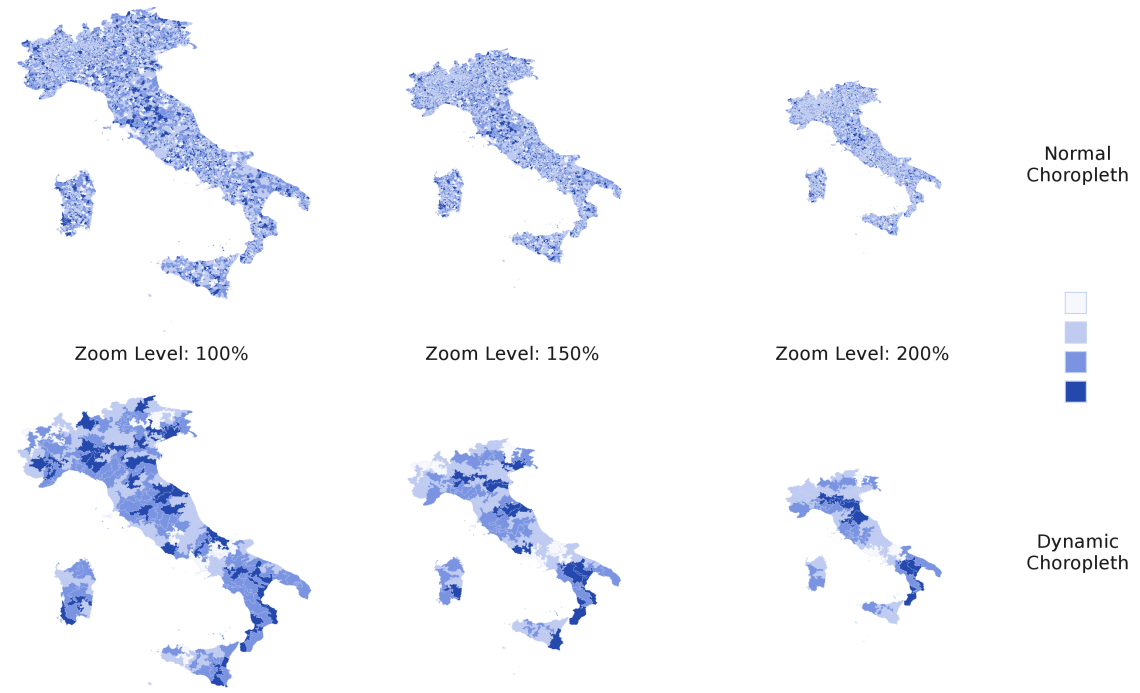


Figure 3.10: An example of zooming out of Italy's administrative units where m = 1%.

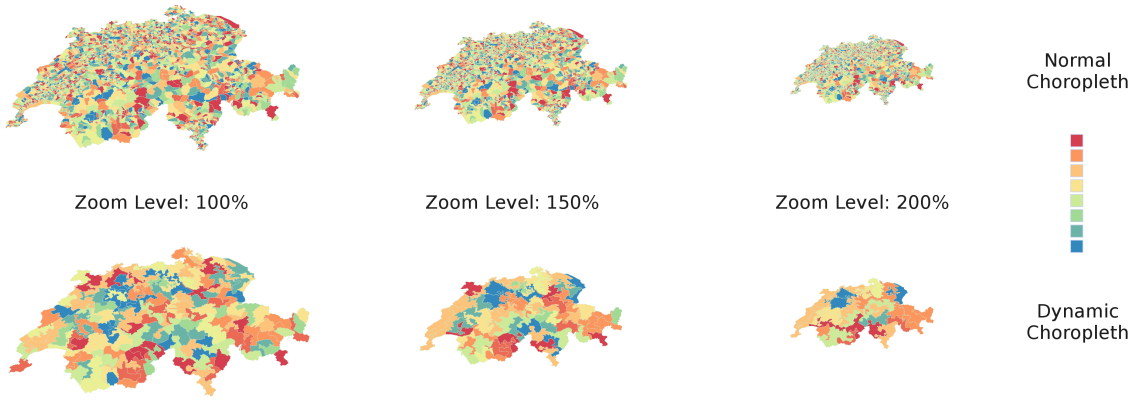


Figure 3.11: An example of zooming out of Switzerland's administrative units where m = 1%.

Chapter 4

User Study

[**?**]

“If you can’t read the scoreboard. You don’t know the score. If you don’t know the score, you can’t tell the winners from the losers.”

— Warren Buffet, CEO of Berkshire Hathaway

Contents

Chapter Abstract

Choropleth maps are an invaluable visualization type for mapping geo-spatial data. One advantage to a choropleth map over other geospatial visualizations such as cartograms is the familiarity of a non-distorted landmass. However, this causes challenges when an area becomes too small in order to accurately perceive the underlying color. When does size matter in a choropleth map? We conduct an experiment to verify the relationship between choropleth maps, their underlying color map, and a user's perceivability. We do this by testing a user's perception of color relative to an administrative area's size within a choropleth map, as well as user-preference of fixed-locale maps with enforced minimum areas. Based on this initial experiment we can make the first recommendations with respect to a unit area's minimum size in order to be perceivably useful.

4.1 Introduction

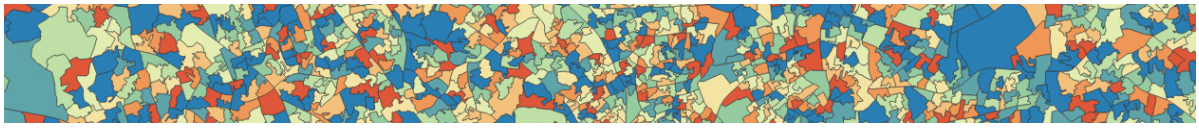


Figure 4.1: Presents a subset of administrative areas in London at 1.5% of the scale of the full map.

Having a completed algorithm for unifying areas based on scale, we verify its effectiveness for choropleth maps. First, we design a user study to present why it is helpful. We can do this by testing user task performance accuracy for small scale areas on a choropleth map. Secondly, we can review user preference of scale exploiting the algorithm in Chapter **??**.

Size is an important facet of visual design in visualization. Our survey of surveys finds scalability as the 3^{rd} most cited future work direction across 40 visualization survey papers, only being preceded by evaluation and missing scenarios in classifications (Chapter **??**). In relation to choropleth maps, size is considered a widely recognized challenge in the field. Ward et al. state "*A problem of choropleth maps is that the most interesting values are often concentrated in densely populated areas with small and barely visible polygons*" [**?**]. Lee *et al.* suggest that improving tiny classes would be interesting future work in the field [**?**].

We conduct a preliminary user-experiment to determine the minimum screen size of a unit area on a choropleth map in order to be accurately perceivable. We also measure the range of unit area sizes where an increase in perceivability error occurs. The experiment focuses on the scalability factors 'human perception' and 'monitor resolution', presented by Eick and Karr [**?**]. We base our user-study on the following hypotheses:

H1 The smaller the size of a unit area, the higher the error rate of perceiving the correct underlying color category.

H2 The smaller unit area, the more time required to perceive its color.

H3 The user will prefer choropleth maps with a larger minimum size over those with smaller sizes. Particularly, users will prefer a trade-off between area resolution in favor of legibility.

The contributions of this chapter include:

1. The first user-study of its kind to evaluate perceivability of color on a choropleth map relative to unit area size.
2. The first recommendation on the minimum screen space size a unit area should occupy to maintain perceivability on a choropleth map.

3. The first recommendation on the minimum screen space size a unit area should be to optimize user satisfaction.

The rest of the chapter is organized as follows. In Section **??**, we explore work in the related fields of perception of choropleths, color, and size. Section **??** breaks down the tasks that we will cover in our user study. Section **??** describes the important variables that are considered in the user study, this is divided into dependent variables and controlled variables. In Section **??**, we discuss the procedure of our user study. We provide depth for each task in order to allow the study to be reproduced. In Section **??**, we discuss the stored data from the experiments to give an understanding of the results. We then discuss the results of our three tasks in Section **??** in detail, and present our user-study findings. Finally, we give our conclusions.

4.2 Background

We first describe previous research involving user-studies and perception. Chapter **??**, as well as Kijmongkolcahi *et al.* [**?**] provide papers that aid in the search for related work in the areas of human-centered evaluation of choropleths, color, and perception of size.

Perception of Choropleths: Rittschof and Kulhavy present a user-study which includes a comparison of choropleth maps and cartograms. Cartograms are a different class of geo-visualization because they use distortion to convey data. They study the recall of data using different types of maps, firstly testing maps against the raw data and secondly studying two different map types (cartograms and choropleths) with two variations of exposure time. Their results indicate that choropleth maps are associated with greater recall of information [**?**]. Kasper reviews the effectiveness of Gastner-Newman diffusion cartograms [**?**] for the representation of population data which includes a comparative experiment against thematic maps (choropleth with overlayed circle maps) [**?**]. The experiment is performed using two informatically equivalent maps using different designs, testing response times and accuracy for varied levels of question difficulty (using Bertin's map reading levels [**?**]). The results report that the thematic maps are more efficient and effective, particularly with complex tasks. Sun and Li review the effectiveness of cartograms for the representation of spatial data which includes a comparative experiment against thematic maps, including choropleths [**?**]. The test takes place online for 100 subjects, who rank thematic maps (including choropleth maps) against cartograms. The results indicate that the thematic maps are more effective representing quantitative data whilst cartograms are more effective with qualitative data.

Perception of Color: We find two survey papers related to color mapping and perception in visualization. Zhou and Hansen present a survey of color maps and color-map generation

techniques in visualization, providing a helpful reference for readers who are faced with color mapping decisions [**?**]. Silva *et al.* present an overview of color, color scales, and tools to guide expert and non-expert users [**?**]. The paper examines different domains of visualization individually.

Heer and Stone investigate how a model of color-naming can enable user interfaces to meaningfully mimic a link between visual perception and symbolic cognition. Color saliency is used to define the degree to which a color value is uniquely named. This is tested against a number of popular qualitative and quantitative color palettes to find the average salience which is calculated using the algorithms provided in the paper [**?**]. Lee *et al.* present a color optimization algorithm for visibility measures over a range of color palettes [**?**]. They look at classes and the area the class will envelope, and modify the color palette based on this. Smaller classes are given higher saturation and luminance, for example. Fang *et al.* provide an algorithmic approach for maximizing the perceptual distances among a set of colors, with a focus on color re-assignment, which may occur when values change, and the phenomenon of local maxima, which becomes a problem when new colors need to be added [**?**]. Szafir presents quantitative studies measuring color difference perception for points, bars, and lines [**?**]. The experiment for each of the three marks is conducted using Amazon's Mechanical Turk with items of varying size in terms of pixels, and recorded the distinguishability of the points, lines, and bars. The goal of the paper is a first step in discerning quantitative understanding of color perception in visualization. Our paper is closely related to this, with a focus on the size of perceivable areas in color, and also focuses on maps rather than mark types.

Perception of Size: Borgo *et al.* present a new order of magnitude mark (OOMM) and an empirical study comparing logarithmic, linear, scale-stack, and their own custom marks to test magnitude estimation, target identification, ratio estimation and trend analysis [**?**]. The results found the OOMM marks outperform state of the art techniques in quantitative analysis. Gramazio *et al.* present a study looking at the relationship between perceivable size, grouping, and search performance. The user is asked to find a distinct mark amongst a set of distractor marks which was applied to a grid layout as well as a scattered layout [**?**]. The study finds that grouped, or single color layouts have a much faster response time, and visual marks below 0.508° showed a large drop in performance time. Jansen and Hornbæk provide a psychophysical investigation of size as a physical variable, where participants estimated size of objects between solid bars and spheres. Solid bars were found to be estimated based on their length whilst spheres were measures based on their surface area [**?**]. Ronne Jakobsen and Hornbæk produce an experiment reviewing the use of overview+detail, focus+context, and zooming over 3 different display sizes. The results indicate that performance time for each technique was worse for smaller screens [**?**]. Healey and Sawant conduct an experiment to review identified pixel resolution and visual angle needed to distinguish different values of

luminance, hue, size, and orientation [**?**]. Our study differs by testing on a set of non-uniform shapes, as would be found on many choropleth maps.

4.3 Experimental User Tasks

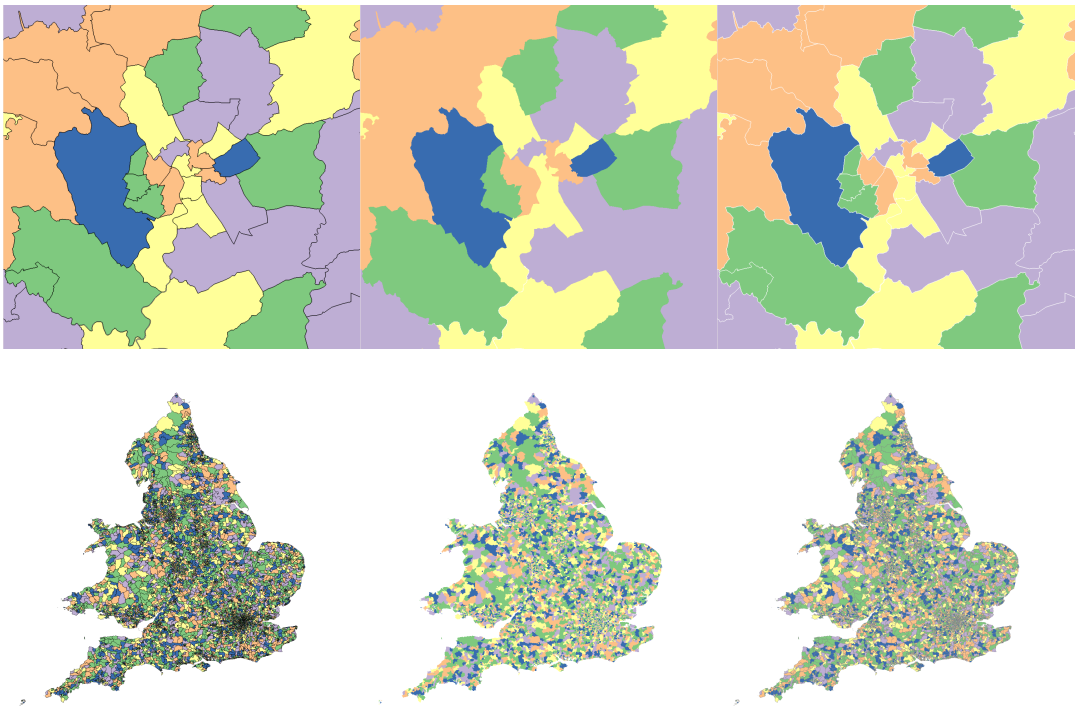
We research the hypotheses presented in Section **??** by testing singular interpretation, comparative interpretation, and comparative preference using the following user tasks:

T1 Identify the color of a given unit area on a choropleth map.

T2 Compare the relative color of two unit areas.

T3 Select the preferred minimum size unit area on a dynamic choropleth map using a slider.

Details about how the tasks are performed can be found in Section **??**.



(a) Black Boundary **(b)** No Outline **(c)** Lower Alpha

Figure 4.2: Three examples for boundary design at two different scales. The selected process is to use boundaries with a low alpha level.

4.4 Experimental Variables

In this section, we discuss the user-study variables, and some of the factors we consider in the design of the experiment. These are split into the dependent variables which we test, and variables we control.

4.4.1 Dependent Variables

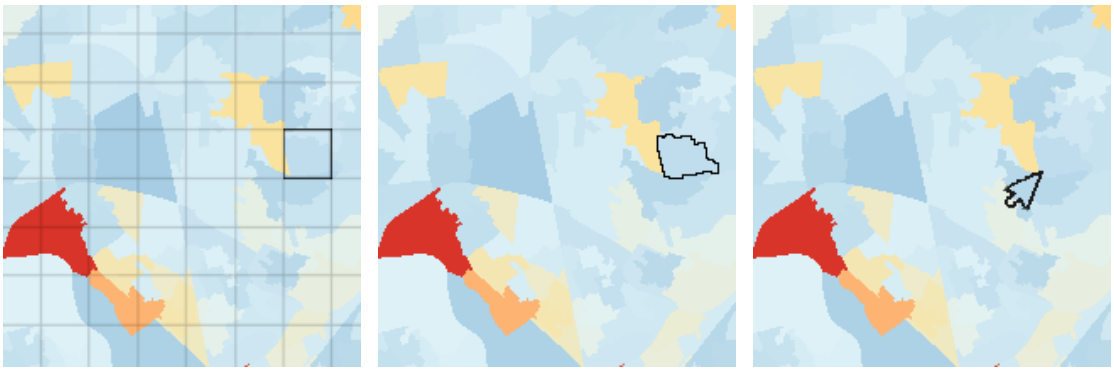
Our primary dependent variable is size of a unit-area on a choropleth map. Unit area size is an important attribute to study due to it's common use and variability in standard choropleth maps. We test variable unit-area sizes by looking at how small a unit area can be in order to perceive the color it encodes based on a given color-map. We measure size in terms of the number of color-mapped pixels in a given unit area. This measure is somewhat screen dependent since different displays feature different pixel densities, thus we use the same display throughout the experiments (see Figure **??** for pixel size demonstration). The hardware specification of the display is given in Section **??** . We also measure the size of each unit area as a percentage of screen space.

4.4.2 Control Variables

We discuss visual variables that we control in order to study the perception of unit-area size. Some user-study design decisions have to be made carefully.

Area Uniformity: The first thing we must decide is whether to test uniform shape areas, where all areas are the same shape during a given task, or non-uniform areas where areas have distinct shapes. For the experiment, we focus on non-uniformly shaped areas used in real choropleth maps. This decision is made to study and emulate real-world scenarios, and therefore we use real-world administrative areas for this experiment. We attempt to collect enough experimental test data to reduce the influence of unit area shape on the experimental findings.

Figure 4.3: Area Size approximation demonstration. (left) 1 pixel. (middle) 50 pixels. (right) 200 pixels



(a) Grid Concept **(b)** Contour Highlight **(c)** Cursor Concept

Figure 4.4: Three concepts for communicating requested area. The selected process was the contour highlight

Area Boundary Design: We must decide how differentiation between neighboring areas is implemented. A boundary line is useful for discerning where one area begins and another ends. Although this can be considered an important feature in a regular choropleth map, we must also consider the perceptual influence associated with different types of boundaries. First is border thickness. When the user examines smaller unit areas, the border can become relatively thick and obscure a unit area's color, causing unnecessary error increase. Three choropleth design options are (a) to include boundary lines, due to their popular use in choropleth maps, (b) remove the outline of boundaries to avoid obscuring any perceivable color, or (c) add boundary lines at a reduced opacity with the goal of reducing their influence. Refer to Figure **??**. We must also consider boundary color an important variable when designing boundaries. We use a black line with an alpha value of 20 (7.5%), to keep a standard profile, using Figure **??**(c) as our boundary design. We use a boundary thickness of one pixel.

Communicating the target unit area to the user: One of the most important factors of the experiment is to verify whether the user can accurately interpret the underlying color of an area of screen space. In order to do this, we need to accurately communicate the target area itself without obscuring or enhancing the area's color. This becomes non-trivial when perceiving an area of screen space at less than 1%. We considered the use of a grid to quickly convey the target area. Although the grid could be effective with uniform size areas, the grid would become less accurate with non-uniform size areas. Grid granularity would also need to be taken into account and conditions in which cell labeling would become harder to communicate. Using the mouse cursor to communicate the area to the user may allow the user to intuitively see the target area. The disadvantage of this is it could cause a wider variance in performance time, and the mouse pointer itself could cause occlusion. A superimposed cursor could be added to the image to control mouse movement, however this could

(a) (b) (c)

Figure 4.5: 3 options for color mapping. (a) Colorbrewer's 5 color palette (qualitative). (b) Colorbrewer's 5 color palette (diverging). (c) Tableau 10 color palette (reduced to 5 for consistency). We use (a) for this study [**?**, **?**].

still result in scenarios with multiple small areas close together being difficult to diagnose. Refer to Figure **??** for examples.

We use Robinson's design criteria [**?**] in order to select an identification criteria that does not alter or accentuate the value itself. We use style reduction highlighting using the lower alpha boundaries discussed in the area boundary design, and give contour highlighting to outline the communicated area using a thicker black outline. We use a pixel thickness of 4 pixels to aid the use for smaller areas in order to reduce the effects of framing areas. On piloting the identification criteria, this alone was deemed to be insufficient causing users to overlook small areas. On top of this criteria, we added a blinking effect to help the user locate the target area, and render the highlighted color to something that stands out over the existing boundary. In this case we use a black boundary which stands out from our color palette. The black outline blinks at 1Hz for a period of 1 second. In order to further aid the search for the highlighted area, we place a grey circle around the target area, where the target area sits in the center of the circle, and fills 10% of the screen, and blinks with the black contour. See Figure **??** (a).

Choice of Color Map: Selecting the appropriate color map is an important decision when testing perception (refer to Section **??**). We use colorbrewer [**?**] due to it's far-reaching application across geography and cartography, and acceptance with basic and advanced users [**?**]. Colorbrewer is also created by geospatial visualization experts. We would like to ensure perceptual distances and therefore avoid the use of a sequential color map focused on a single hue. We can incorporate qualitative and diverging color maps for the color palette. Using Heer and Stone's color saliency test, we can see that qualitative color palettes tend to have better saliency, and can therefore use this as a guideline [**?**]. Although Tablaeu's color palette provides marginally better saliency, we use Colorbrewer's qualitative color map [**?**, **?**]. See Figure **??**.

Display Screen Resolution: Keeping resolution consistent is essential to experimenting with perception. Both digital resolution (display resolution) and physical resolution (monitor size) need to be consistent. In order to ensure this, we must forgo a mechanical turk testing

environment and work in a controlled lab test environment. This is further explained in Section **??**.

Lighting Conditions of Environment: We must consider the lighting as an important control variable when conducting this experiment. We make sure that the lighting environment stays the same by using a static study location, that uses interior lighting as the only lighting source. We also must consider the monitor's lighting settings, we do this by calibrating the screen using an International Color Consortium (ICC) profile. We provide a sample of presented question designs for T1 and T2 in Figure **??**.

4.5 Perceivability Experiment

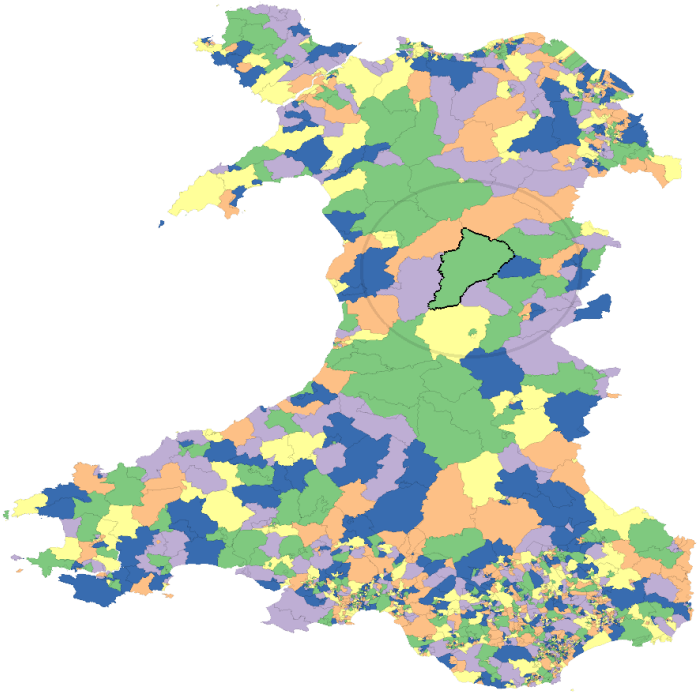
In this section, we discuss the nature of the experiment including our designated equipment, experiment design, and stored experiment data.

4.5.1 Equipment

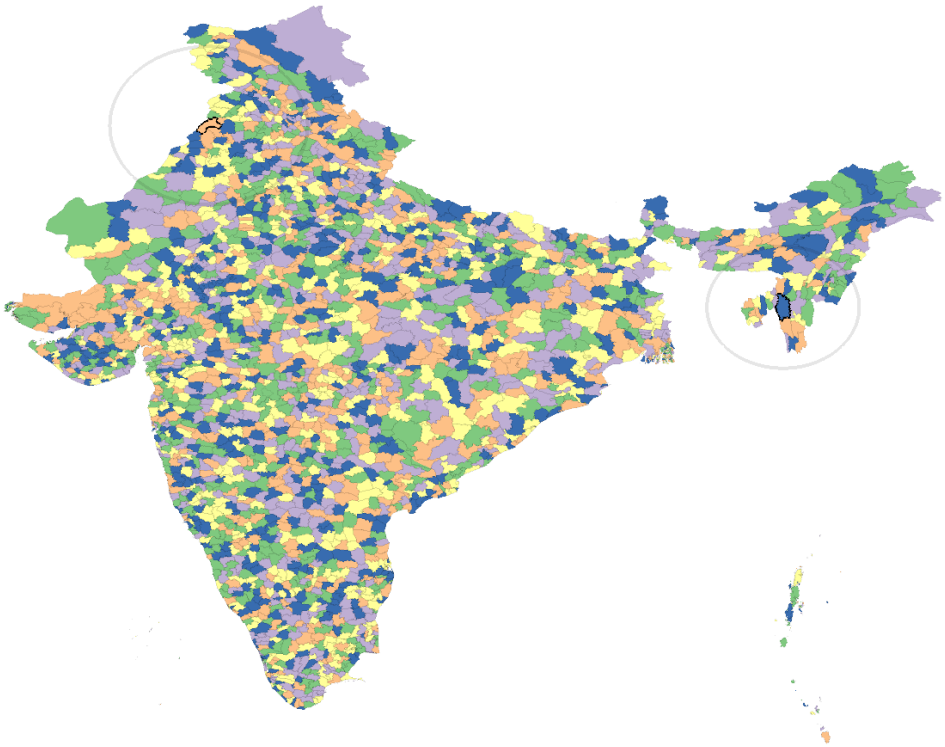
For this process, we use a standard Dell U2412M display resolution of 1920x1080p, on a screen size of 21". We then display the choropleth images to the subject as a static image 1200x960p. The desktop used to test this implementation features an Intel i5-4460 at 3.2GHz with 16GB of RAM and a GeForce GTX960. The test is run using the Qt Framework [**?**]. For the ICC color pofile, we use the Dell U2412M (Custom) color profile with suggested settings [**?**].

4.5.2 Procedure

The experiment requires approximately 30 minutes per participant. We solicit ten participants who individually perform the tasks in this study. The subjects were gathered using an advertising campaign. The experiment is designed to test perception of color conveyed by areas with a small screen space. An instructive introduction explains what the test entails, what a choropleth is, how a task is completed, and sample questions. Some sample questions are used to test for basic understanding of the tasks as well as a test for color blindness or eyesight problems. The sample questions provide a task where the largest area is used as the target. Test subjects who fail the training questions do not have their results used. For T1 a standard question presents a choropleth map image mapped by a color palette. There is one highlighted area in which the subject diagnoses the conveyed color. Each section of the color



(a) T1 Sample



(b) T2 Sample

Figure 4.6: Sample questions for T1 and T2.

(a) Original **(b)** 0.100% Minimum Area Size **(c)** 0.250% Minimum Area Size

Figure 4.7: Sample map designs using dynamic maps presented in Chapter **??** for T3.

palette is presented as an option for the subject to select. See the Supplementary Material. The subject's color-choice answer is saved to a database for further analysis along with the true color of the area. These are saved as well as the performance time, choropleth identifier, area identifier, and the area's size in terms of pixels and screen space percentage. The experiment is run in task sets of 10, where each set provides a set of random choropleth maps and random unit area targets with a screen space ranging from 1 pixel upwards. When the set is complete we produce a short transitional state where we ask the participant for a confidence rating, and ease of understanding rating. See Figure the Supplementary Material. Each task is associated with a random choropleth, random color from the color map, and randomized choice of unit area on the choropleth in order to prevent learning effects. We run this for five sets.

T2 is presented indicating two unit areas on the same choropleth map. The user is asked to diagnose whether the units are the same color, or differ. After selecting their answer, the answer is saved as well as the the same set of information in T1, for each area. This is run in sets of five, with ten tests per set. See Figure **??** for sample images.

T3 is presented by showing a dynamic map presenting the same data using the Dynamic Choropleth Map procedure presented in Chapter **??**. We let the user dynamically select their preferred minimum screen space. The map for each position on the slider is split to enable options for the default map, moving up in increments of 0.001% up to 0.01%, and an increment of 0.01% minimum screen space up to 0.25%. The user will simply select the preferred size using the slider and a new map will replace it. This is done for five different comparisons, with the selected options being stored for further analysis. Refer to Figure **??**. These maps will be given sequentially as we want to gather data for each of these maps, where the slider of each represents different constrained minimum areas.

4.5.3 Stored Experimental Data

Refer to Table **??** for a visual breakdown of the stored experimental data. Data stored per task in T1 for analysis includes: map identifier, target area identifier, user-selected area color, target area color, task time, task identifier, and set identifier. T2 saves all of the T1's information, but with the additional data for the second target area. For both T1 and T2, we store the set identifier, confidence level, and difficulty level per set. For T3 we store: task identifier, task time, and preferred screen space identifier. T1 does not store any information for target two as the task only focuses on one target area. T2 includes the most data saved. T3 does not save any information for sets, screen space of targets (as targets already refer to screen space), pixels, or the target answer, as this is used to test preference. Therefore we save the value selected using a the interactive slider. See Table **??**.

4.6 User-Study Results

In this section we discuss the results of three tasks which include diagnosing the color value of areas on a choropleth map, comparing the color of areas on a choropleth map, and de-

Stored Data:	**T1**	**T2**	**T3**
User ID	✓	✓	✓
Set ID	✓	✓	✗
Test ID	✓	✓	✗
Performance Time	✓	✓	✓
Map ID	✓	✓	✓
Target ID	✓	✓**x2**	✓
Screen space for Target	✓	✓**x2**	✗
Pixels for Target	✓	✓**x2**	✗
User Selected Color	✓	✓	✗
Target Color	✓	✓	✗
Set Confidence	✓	✓	✗
Set Ease	✓	✓	✗
Slider Value	✗	✗	✓

Table 4.1: Table indicating the data saved for each task. Refer to Section **??**.

termining user preference for choropleth maps with respect to minimum area size. The task included 10 participants with 50 data samples per participant for both T1 and T2, and 5 data samples per participant for T3 (from a selection of 130 potential candidates). Of the participants, we use an unconstrained sample of users from both genders, a range of ages (18-35), at different expertise levels. Candidates were provided with £10 amazon vouchers after the completion of the study's completion.

4.6.1 T1 Analysis

Task 1 involved the user determining the color of highlighted areas. Each user completed the task with 50 data samples, excluding outliers for a total of 500 data samples across all users. Of these 500, we use 465 of these for analysis, filtering out areas that are over 200 pixels in size, or outliers in terms of performance time (see Section **??**). For T1 we present our findings using a scatter plot and area chart (Figure **??**). The unit plot displays the relationship between the performance time, unit area size, and user results. We can see a large distribution of results where the user correctly interpreted the color mapped to the area. For incorrect results we see that the results primarily fall very close to 0. On top of this, we identify when a user selects the 'couldn't tell' option. This seems to fall along the lower half of the incorrect distribution. In terms of performance time, there is a clear gap between 0-2 seconds. This seems to be related to user search time, and the highlight animation. This is followed by another 2 seconds where there is a noticeable gap in incorrect answers. This might be influenced by pre-attentive processing . The area chart presents the accuracy of T1 error based on unit area. We also provide a histogram to present the frequency of size representation in the study. For T1, we see clearly that accuracy is particularly low below 5 pixels in size. With a range of error just over 90% for 1 pixel. We see from this chart that the error becomes prevalent under 10 pixels.

Statistical Analysis: We use our results of T1 in our analysis of H1 (size vs error) and H2 (size vs time) using the Pearson correlation co-efficient where α = 0.05. As our sample size favors smaller pixels due to the random nature, we use normalized error rate to calculate the relationship between area size and error. We conclude using Pearson's correlation co-efficient that there is a relationship between area size and error rate, where r(50) = -0.77. This supports our hypothesis H1. For H2, we conclude there is a relationship between area size and performance time, where r(466) = -0.325. This supports our hypothesis H2. Due to this we can verify that our first two hypotheses are supported statistically by our results for T1.

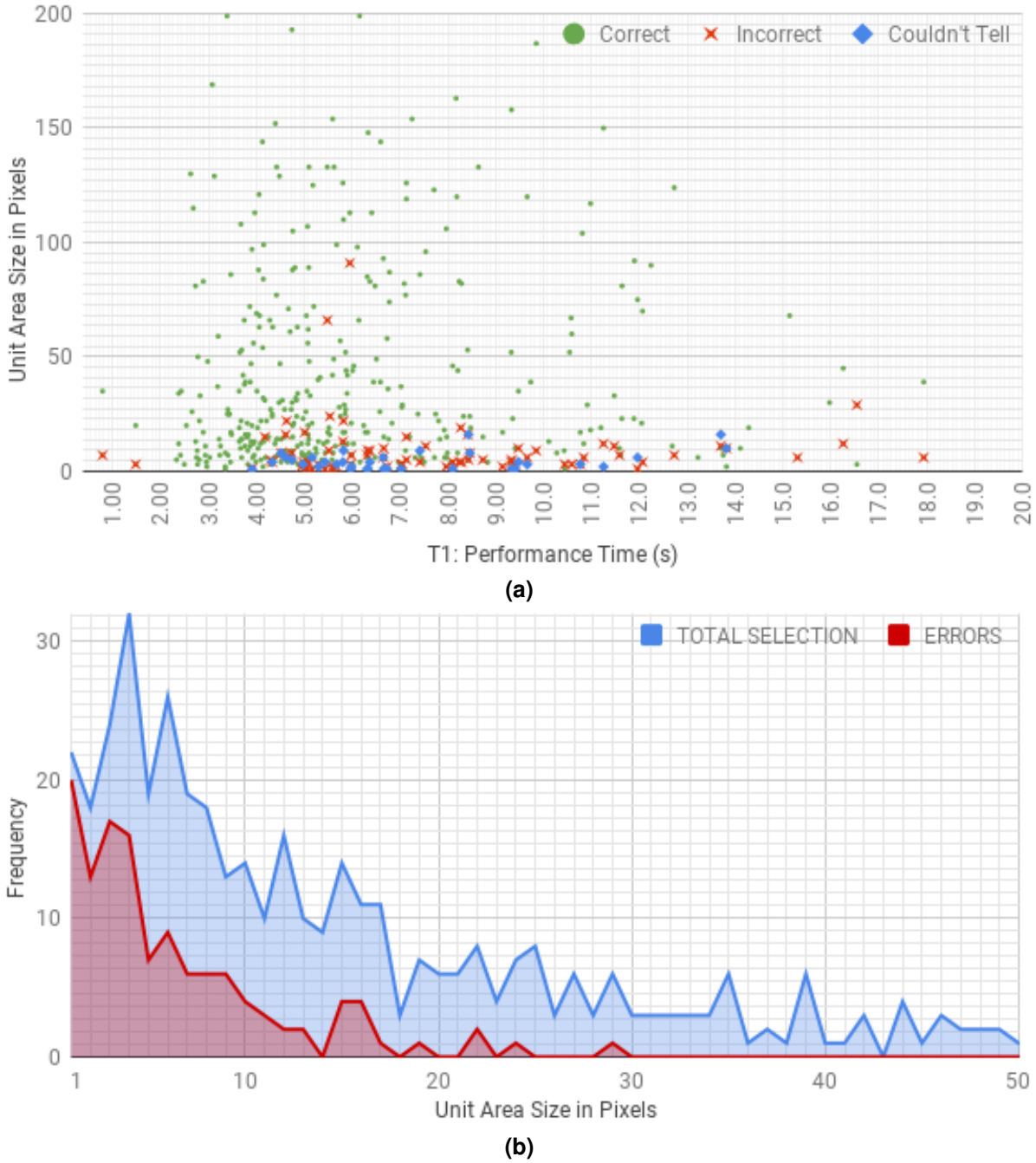


Figure 4.8: T1: (a) A scatter plot depicting the correlation between performance time (seconds) and area size (in pixels). Color is mapped to Answer Category (green for correct answer, red for incorrect answer, blue for 'couldn't tell'). (b) Area chart presenting the T1 error rate based on area size measured in pixels, where the blue area represents the total samples per area size, and the red area represents the error per sample.

4.6.2 T2 Analysis

Task 2 involves comparing 2 highlighted areas in order to diagnose whether they are mapped to the same color value. Each user completed the task with 50 data samples for a total of 500 data samples across all users. Of these 500 we use 490 of these for analysis, filtering out areas that are over 200 pixels in size, or outliers in terms of performance time (see Section **??**). We produce similar graphs to facilitate comparison between T1 and T2 (see Figure **??**). Similarly to T1, correct answers exhibit a wide distribution of pixels and performance time, however we find that users were far more liberal with the use of of the 'cannot tell' option. This suggests that the binary comparison was more difficult than the selection of colors, confirming the assumption that color salience may have been an important factor to consider.

In terms of accuracy, we can see that the error rate for T2 was less extreme for small area sizes, but has a slightly higher distribution which leads to error up to 39 pixels (although 38 pixels has only one test case, 35 pixels has a better sample size). The lower error rate at smaller sizes could be related to a couple of visual features. First, as there are 5 map-able colors, when diagnosing colors in T2, a decision can be made by testing the contrary. A user may not be able to distinguish a color in T1, however if they can distinguish that the two colors are not the same without distinguishing the correct color, they can effectively answer the question. We must also consider that although we provide and encourage the use of the 'couldn't tell' button, some users compared this to a test and therefore ignored the button. For T1, guessing a correct answer has a 1/5 chance of being correct, however, selecting the 'no' option has a 4/5 chance of being a correct answer in T2.

Statistical Analysis: We use our results of T2 in our analysis of H1 (size vs error) and H2 (size vs time) using the Pearson correlation co-efficient where α = 0.05, similarly to our analysis of T1. We follow the same convention to analyse error as in T1, by using normalized error. From this analysis we conclude using Pearson's correlation co-efficient that there is a relationship between area size and error rate, where r(50) = -0.433. This supports our hypothesis for H1. For H2, we conclude there is a relationship between area size and performance time, where r(490) = -0.358. This also supports our hypothesis H2. As T1 and T2 both support the hypotheses of H1 and H2, we can be confident that our hypotheses for each are evident.

4.6.3 T3 Analysis

For T3, we allow the user to select the most "useful" map from a set of fixed-locale choropleths which constrain areas size, using the Dynamic Choropleth Map procedure presented in Chapter **??**. We defined "useful" as being 'your preference of area size against the data relayed, where larger areas provide more uncertain data'. A histogram is used to visualize

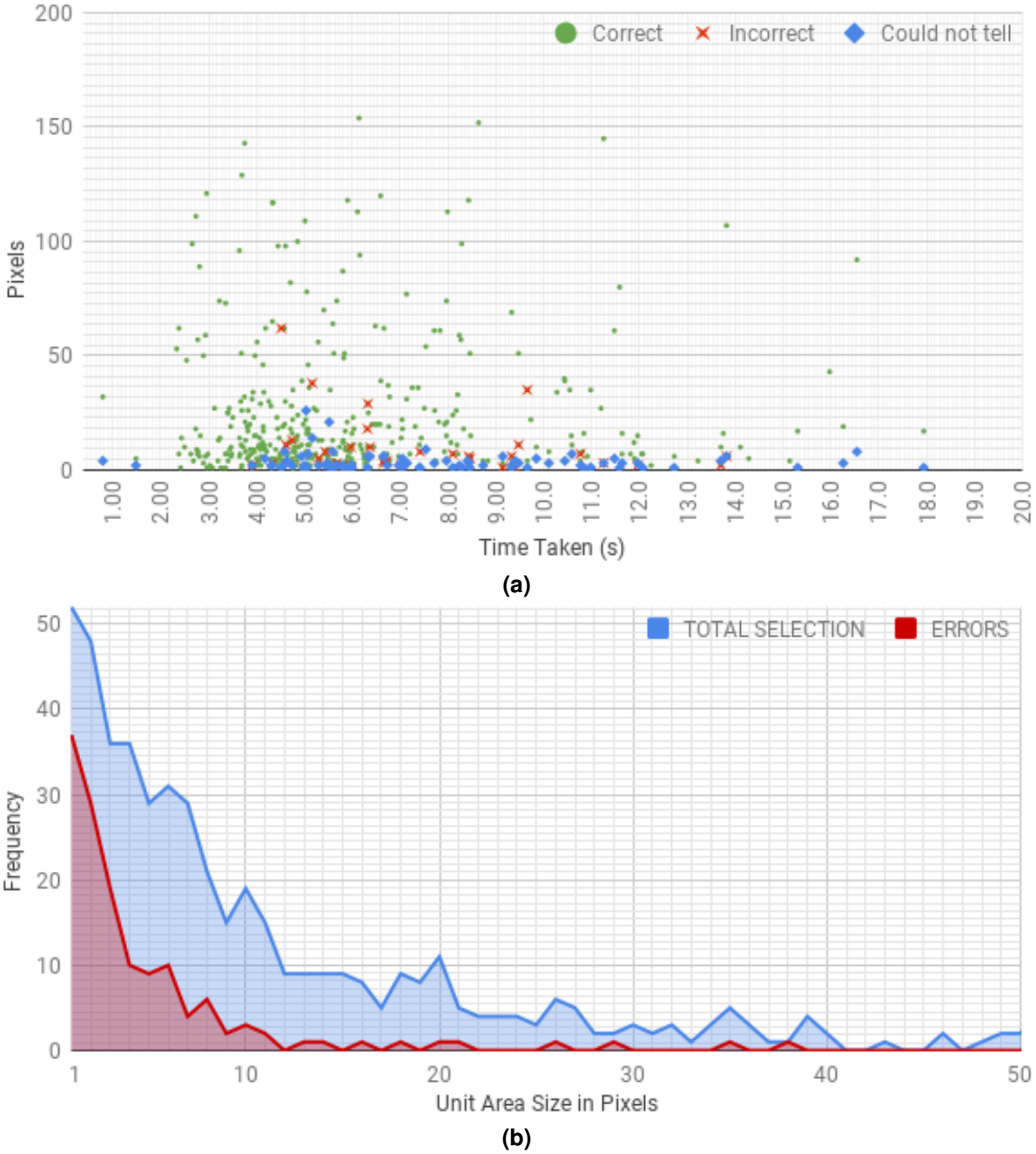


Figure 4.9: T2: (a) Scatter Plot depicting the correlation between performance time (seconds) and area size (pixels) of the smallest target area. Color is mapped to user performance (green for correct answer, red for incorrect answer, blue for could not tell). (b) Area chart presenting the T2 error rate based on area size measured in pixels, where the blue area represents the total samples per area size, and the red area represents the error per sample.

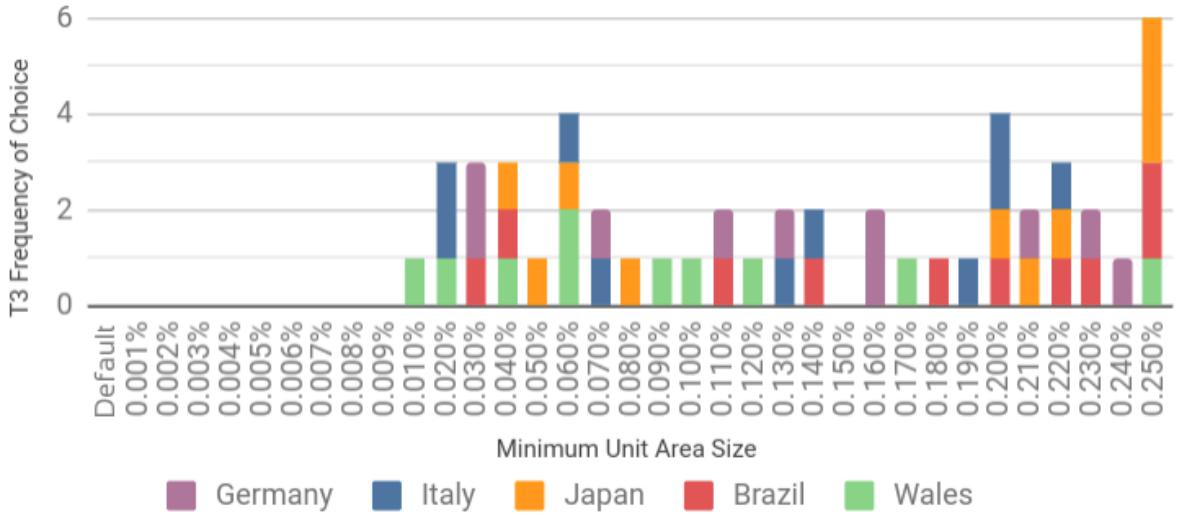


Figure 4.10: T3: Histogram presenting the frequency of minimu unit area size preference, where the preference is defined as 'your preference of area size against the data relayed, where larger areas provide more uncertain data'.

the frequency of selection for T3 and the results of this task are quite unexpected (see Figure **??**). We expected that the preference would be found at the lower end of the spectrum, however over half of the study participants selected the largest possible enforced minimum screen space. On top of this, our pilot study prompted us to increase the number of selectable areas closer to the default screen space which did not reflect any results in the study. This may have been caused by the discrepancy in size variation between the first 10 selections, and the following selections. In order to rectify this, a follow-up study may be necessary that shows screenspace on a logarithmic scale to remove that type of bias, or a larger selection of choices, that are quantized into bins. Unlike H1 and H2, we cannot effectively run statistical analysis on our finding due to the subjective nature of the task. However, our observations do show evidence to support hypothesis H3. A more thorough investigation into this task is a good area for future work.

4.6.4 Anecdotal Results

During the period of testing, there were a few interesting observations we noticed. We found that users were generally caught off guard by the difficulty of the test, although the random selection of areas was mentioned, some users believed that this was not the case, leading to the conclusion that many are unaware of the frequency of small areas. To continue with this, users tended to feel like they had performed poorly, making statements remarking the difficulty. Some users regarded surprise at the difficulty of T2, suggesting that they believed it would be easier to identify whether two colors would be the same, but found themselves in situations

where it was difficult to identify either color. For T3, one user explained that his decision for larger areas was based on the fact that they prefer islands to consist of only one area, which suggest the users' preferred minimum screen space could be related to area contiguity.

4.7 Conclusion

We conduct a preliminary study to evaluate perceptual evaluation of performance error, performance time, and preference of area's relative to size in choropleth maps. We hypothesize that the smaller the size of a unit area, the higher the error rate of perceiving the correct underlying color category; the smaller the unit area, the more time required to perceive its color; and the user will prefer choropleth maps with a larger minimum size over those with smaller sizes. Particularly, users will prefer a trade-off between area resolution in favor of legibility. For these three statements, we support the first two using Pearson's r to show a relationship between the variables, with strong observations for the third that give an initial understanding of how users' perceive the relationship between area size and data on a choropleth map. We give a recommendation of at least 10 pixels displayed for each area to be accurately interpreted.

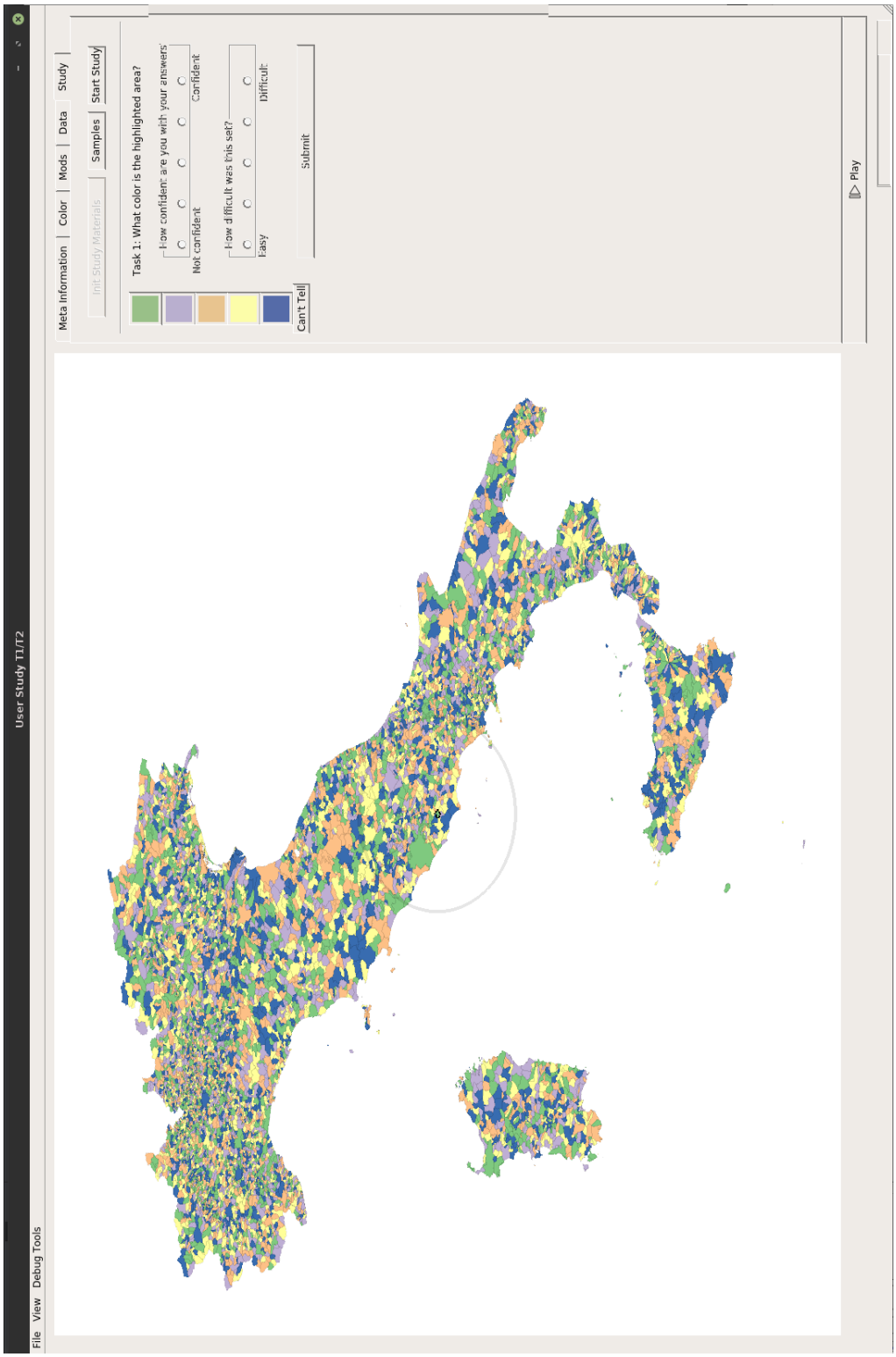


Figure 4.11: Sample question for T1 presenting in application, including user interface. This sample shows a more realistic question than the samples shown in the Main text (selected to present clearer examples).

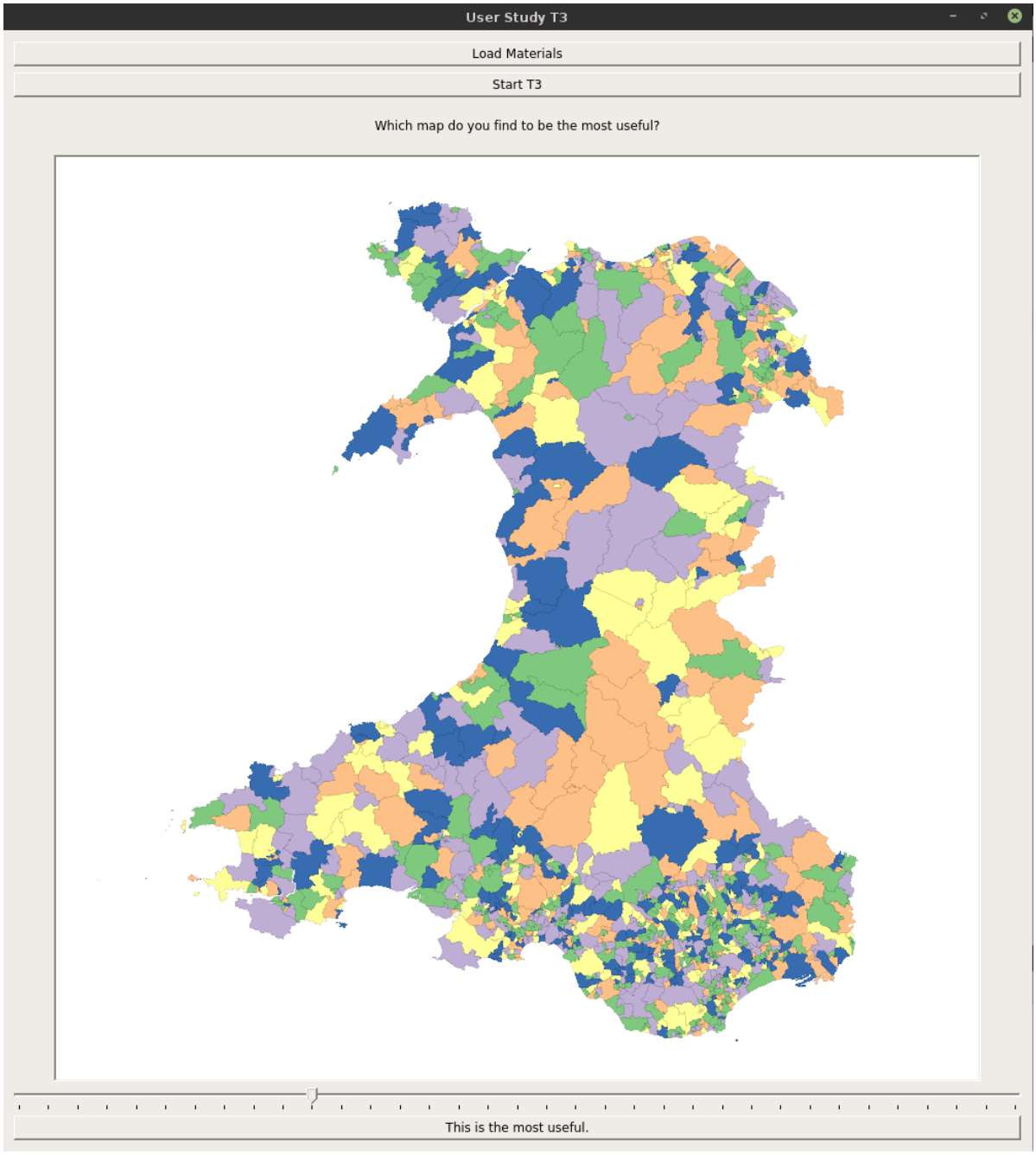


Figure 4.12: Sample question for T3 presenting in application, including user interface.

Chapter 5

Multivariate Maps

[**?**]

"A glyph representing multiple attributes may need simplifying when reduced in size, resulting in a loss of data."

— Ellis and Dix, A taxonomy of clutter reduction for information visualisation [**?**]

Contents

Chapter Abstract

Maps are one of the most conventional types of visualization used when conveying information to both inexperienced users and advanced analysts. However, the multivariate representation of data on maps is still considered an unsolved problem. We present a multivariate map that uses geo-space to guide the position of multivariate glyphs and enable users to interact with the map and glyphs, conveying meaningful data at different levels of detail. We develop an algorithm pipeline for this process and demonstrate how the user can adjust the level-of-detail of the resulting imagery. We present a selection of user options to facilitate the exploration process and provide case studies to support how the application can be used. We also compare our placement algorithm with previous geo-spatial glyph placement algorithms. The result is a novel glyph placement solution to support multi-variate maps.

5.1 Introduction and Motivation

In this chapter, we move toward extending our existing algorithm to support new features, including multivariate data and glyph placement. We also present the robustness of our design using real-time filters and animation.

Maps are useful for conveying information to both inexperienced and advanced users. There are many types of maps designed to present data but the underlying maps often come with other challenges such as the how the areas are segmented. Fairbairn et al. suggest scale, level of detail, and multivariate data as common challenges for the representation of geo-spatial data [**?**]. Ward et al. state, *"A problem of choropleth maps is that the most interesting values are often concentrated in densely populated areas with small and barely visible polygons, and less interesting values are spread out over sparsely populated areas with large and visually dominating polygons."*

The challenge of perception (**C1** – size perceivability) is a fundamental one associated with digital maps. Even when trying to rectify this for a univariate map, few solutions enable opportunities to convey multivariate, high-dimensional data. For example, geo-spatial designs (choropleths, cartograms, symbol maps, etc.) only depict uni-variate, or occasionally, bivariate data. This is a challenge regarding the conveying of multi-variate geospatial data (**C2** – multivariate geospatial data). One possibility is glyphs to support multivariate visualization options. However, even if we can present multivariate geospatial data using glyphs, we still run into challenges. If we plot glyphs in their geospatial context, then we risk overlap and over-plotting. In other words, if we place a multivariate glyph at the center of each unit area on a map, the glyphs will either overlap in many cases or be too small to perceive, especially in densely populated areas (see Figure **??**) (**C3** – occlusion). Ellis and Dix state *"a glyph representing multiple attributes may need simplifying when reduced in size, resulting in a loss of data"* [**?**], suggesting that reducing the size of a scalable multivariate glyph can be problematic (**C1** – size perceivability). Another option to address **C3 – occlusion** is to employ structure-driven glyph placement guided by a Cartesian grid. However this common solution de-couples the glyphs from the original geospatial areas they intend to represent. This is the challenge of geo-spatial glyph-placement (**C4** – glyph placement).

In order to address all four challenges, **C1–C4**, we introduce scale-aware maps, a process of presenting geo-spatial multivariate data based on a desired screen space, that enables dynamic modification to the level of detail shown using both zooming functions and custom scale options. We integrate this with glyphs to present multivariate data in a geo-spatial context to

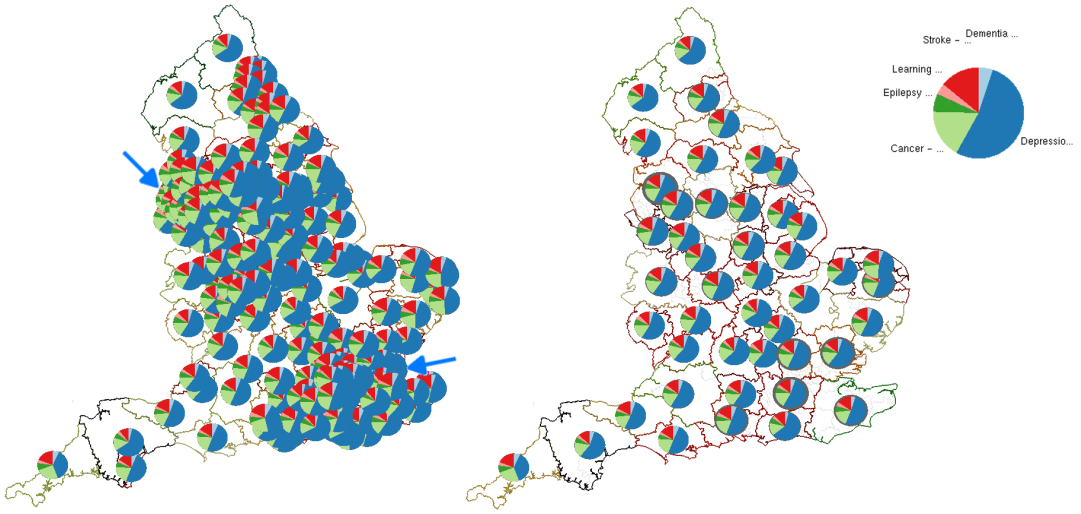


Figure 5.1: (left) The representation of population health data based on the Clinical Commisioning Groups (CCGs) of England [**?**]. Refer to Section **??** for a case study. Glyphs that are simply placed at the centroid of each region are over-plotted and occluded around London, Manchester, and Liverpool (indicated by blue arrows). (right) Our result using level-of-detail scale-aware maps. Even at a small scale for the figure, we can still clearly differentiate each area's glyph.

enable interactive exploration, and facilitate easier comprehension with area context using both smooth transitions and uncertainty indicators. We refer to our work as using glyphs as opposed to symbols guided by the definition from Borgo et al. who define glyphs as, *"...an independent visual object that depicts attributes of a data record"* [**?**]. Our contributions include:

1. A multivariate map with scalable glyph rendering and presentation (in the form of scale-aware maps) (**C1** – size perceivability, **C2** – multivariate geospatial data, **C4** – glyph placement).
2. Dynamic glyphs that support zooming, and user-controlled level of detail. (**C2** – multivariate geospatial data, **C3** – occlusion, **C4** – glyph placement)
3. Interactive filters to improve analysis and exploration of multivariate data and comparison of geo-spatial areas. (**C2** – multivariate geospatial data)

In order to do so, we develop solutions that address the four major challenges, **C1–C4**.

5.2 Background

Our literature review in Chapter **??** includes a section of glyph-focused survey papers, as well as geospatial surveys. Borgo et al. present a survey of glyph design criteria [**?**]. Fuchs et al. provide a systematic review of experimental studies on data glyphs [**?**]. Ward presents a taxonomy of different glyph placement strategies (discussed further in the glyph placement section) [**?**]. We find three survey papers on cartograms [**?**, **?**, **?**]. We do not consider univariate cartograms within the scope of our work as they distort the boundary geometry of the geo-spatial data, which we avoid in our process.

Related Work with a focus on Multivariate Maps: Multivariate maps have been used in cartography for over 100 years. For example, Minard depicts a multivariate map using pie charts to present cow consumption across France [**?**]. The pie charts are placed manually. Kahrl et al. present a range of imagery focused on California's water supplies including irrigation applied to crops in the form of dense pixel displays across geo-spatial points [**?**]. Approaches to add more dimensions to choropleths include bivariate color maps [**?**, **?**]. Dorling visualizes local urban changes across Great Britain [**?**]. The paper uses multivariate options to review industry distribution, owner-occupied housing, as well as a set of attributes plotted using Chernoff faces as a equal area representation. Brewer and Campbell present point symbols for representing quantitative data on maps, including bi-variate options [**?**]. Although their paper does not focus on placement. Their examples place symbols on a centroid, and show minor occlusion. Andrienko and Andrienko [**?**] contains a range of examples of multivariate maps using glyphs for thematic maps, including temporal glyphs, and multivariate pie glyphs for forest data. Slocum et al. provide a chapter on multivariate maps, describing techniques to consider when displaying bivariate, trivariate, and multivariate data [**?**]. Slingsby et al. capture the geo-spatial context and transform their results into a grid, which is then represented by a treemap where the hierarchy is based on temporal data [**?**]. Slingsby et al. present a rectangular cartogram showing the postcodes in Great Britain, where postcode district and unit postcodes form the hierarchy [**?**]. Cartograms distort geo-space, which we avoid using our procedure. Elmer review symbol consideration for bivariate thematic maps, but do not support more than two variates [**?**]. Our algorithm supports an arbitrary number of variants depending on the glyph design. Kresse and Danko present geographic techniques from basic principles to applications [**?**]. They present a table of visual variables to represent data, applied to a given map and symbols.

Tong et al. develop Cartographic Treemaps to explore multivariate medical data provided by Public Health England [**?**]. This is extended to time-varying data [**?**]. Tsorlini et al. present a taxonomy of thematic cartography symbols, including multivariate options [**?**]. The symbols are presented as a hierarchy, focusing on the amount of attributes, and arrangement.

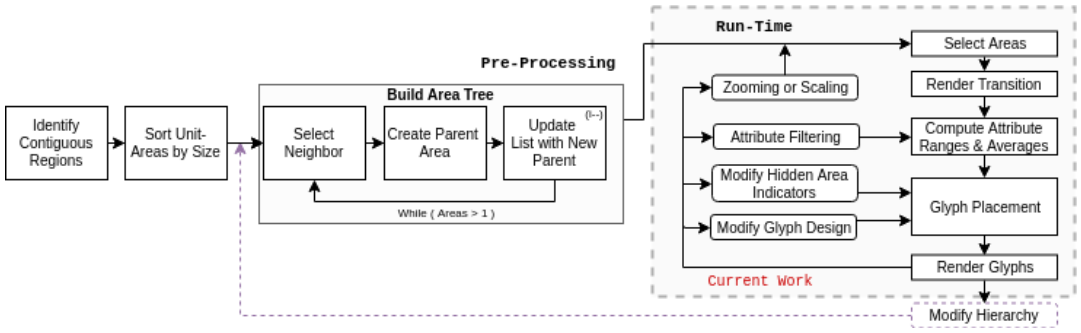


Figure 5.2: The flow chart of the procedure. The right square represents what is discussed in the scope of this paper, the pre-processing steps are discussed in Chapter **??**.

Beecham et al. visualize trends to explain the UK's vote to leave the European Union. They use a juxtaposed view to presented equal area cartograms for different variants [**?**]. Nusrat et al. produce a cartogram that presents bi-variate data using a ring encoding, where the color presents the leading statistic, and the ring thickness presents the value the leading statistic leads by [**?**].

Related Work with a focus on Glyph Placement: Ward and Lipchak create a software tool for cyclical, temporal multivariate data. Glyphs are placed in an ordered grid structure to enable easy comparison between similar months, or entire years [**?**]. They also use a radial structure. Our work differs from this work by focusing the glyph placement coupled to geo-spatial areas. Ward presents a taxonomy of different glyph placement strategies [**?**]. They introduce glyph designs that can be used, and 15 glyph placement strategies together with a flow chart of how the glyph placement is driven (data-driven or structure-driven). Our placement strategy is considered geo-spatially data-driven. As the modifications are made before the placement process, it falls into original->derived->data-driven. This is expanded by a subsection in a further Ward survey [**?**]. Andrienko and Andrienko present glyph placement in a selection of ways, for example standard symbol placement for US states and a Cartesian grid to represent forest data over Europe [**?**]. They discuss the importance of the link between identifying a symbol and the geo-space it represents (on the map) (**C4** – glyph placement). Ropinski and Preim present a taxonomy of usage guidelines for glyph-based medical visualization [**?**]. As opposed to Ward's placement taxonomy, they suggest glyphs should be placed based on physical characteristics or anatomical features. Borgo et al. provide a section on glyph placement which extends on both of the previous taxonomy by suggesting user-driven placement [**?**]. Chung et al. discuss glyph sorting strategies and present horizontal axis bins, applying it to sport-event analysis glyphs [**?**]. Our work differs by guiding our glyph placement strategy based on a geospatial, 2D context.

5.3 Design Goals and Tasks

We derive six mains tasks to motivate our design process.

T1 – Overview: Provide a glyph-based overview of multivariate data on a map free from occlusion (**C3** – occlusion).

T2 – Multivariate Map: Offer a selection of informative multivariate glyphs to compare trends between regions (**C2** – multivariate geospatial data).

T3 – Glyph Placement: Clearly couple encoded glyphs to their geo-spatial contexts (**C4** – glyph placement).

T4 – Occlusion Leverage scale-aware maps to enable exploration of the data at multiple levels of detail (**C1** – size perceivability).

T5 – Filtering: Support the exploration of multivariate geo-spatial data with user options with varying glyph designs and filters (**C2** – multivariate geospatial data).

T6 – Fluid Interaction: To provide smooth and fluid transitions between the different levels of detail for fluid interaction (**C4** – glyph placement).

5.4 Overview

This section provides the pre-processing steps used to create the scale-aware maps, the run-time process for transitioning between glyph densities, and the options we provide to enhance the exploration of the data. The pre-processing steps are based on previous work in Chapter **??**. The purpose of the pre-processing step is to build a dynamic map whose areas are always perceivable, unit areas that are too small (refer to Chapter **??**) are unified until they reach a minimum area threshold set by the user. We build a hierarchical area-based data structure before displaying a dynamic choropleth map. We separate regions into islands (or land masses) for topological continuity. Once each contiguous region is identified, each unit-area within the same contiguous region is sorted in order of increasing size, since scale is an important part of the algorithm. The area-based hierarchy construction is a recursive algorithm broken down into three sub-routines. In these three steps, we select the optimal neighbor for merging, we identify the shared boundary between the given area and its neighbor, and unify them to create a new area which is then inserted back into the list of areas sorted by size. A flow chart of the procedure is found in Figure **??**.

5.4.1 Pre-processing

Contiguous Regions: In order to reduce the complexity of the hierarchy construction, we group unit areas into contiguous regions. If a given land mass or island identifies a neighboring unit-area, we recognize that every other region belonging to the land mass is also linked contiguously. We can then merge the two areas and continue our search. It is important that we do not terminate the search here as our new unit-area may join multiple land masses or islands together. Once this process is complete for each unit-area, we have identified each contiguous region.

Build Hierarchy: We use a recursive procedure to create a hierarchical area-based data structure. An area hierarchy is created for each contiguous region, where each area is merged with its closest neighbor identified using a customizable distance metric (refer to Chapter **??**). We start with a merge candidate list filled with the sorted unit-areas (for one contiguous region). There are three main sub-routines: (a) neighbor selection, (b) creating the parent area, and (c) updating the merge candidate list. If only a single unit-area remains in the merge candidate list, no further merges can be processed and the procedure terminates. (a) In order to select an appropriate neighbor to join, we use a general and flexible distance metric for amalgamation evaluated between neighboring areas.

We use the closest distance considered as the optimal selection for a neighbor, $D = w_a \cdot \frac{a}{a_{max}} + w_d \cdot \frac{d}{d_{max}} + w_\alpha \cdot \frac{\alpha}{\alpha_{max}} + w_{b_s} \cdot (1 - \frac{b_s}{b_{s_{max}}})$. The measure consists of four constituents: Smallest area (a), euclidean distance between centroids (d), value variance (α), and shared boundary resolution (b_s). We search and identify each common vertex between neighboring areas to identify the shared boundary. We update the sorted area list by removing the two merged areas, and inserting the newly created parent, which may be used as a new merge candidate. This is repeated until only one area remains in each contiguous region.

Value calculation for unified areas: The Modifiable Areal Unit Problem (MAUP) [**?**] is an important aspect to consider when discussing the modification of boundaries or values. We address this by providing the user options to customize calculation of aggregated values as well as the customizable distance metric used to evaluate area merge candidates. The multivariate data is linked to the unit areas during the initial loading of the shape files. Before the area tree is built, the user can select the type of value amalgamation. This enables the user to choose options of sums, frequencies, and value averages. When amalgamating values using sums, the value can be calculated using aggregation. Qualitative values are calculated using frequencies. For a detailed description of parent value calculation, see Chapter **??**.

Figure 5.3: An example of a smooth transition made between two child glyphs that translate towards the new parent node. Both child glyphs decrease in opacity, whilst the new parent glyph increases in opacity. Refer to Section **??**.

5.4.2 Geospatial Glyph Placement

We select visible areas and glyphs based on a minimum area scale requirement (a percentage), m (see Section **??**). When the map is rendered, the tree is traversed using a depth-first search (DFS) to identify which areas are rendered. If an area is larger than m we test two criterion: if the area is a leaf node, or if either the left or right child is smaller than m. If either of these true, we render the area. For each area displayed, we create a glyph using the area's centroid to position the glyph. We create a glyph that reflects the given area's multivariate data values (based on the user's selection, see Section **??**). As the zoom level of the map changes, different areas may meet m and therefore be presented, creating a dynamic presentation of glyphs. This addresses **T1 – Overview** and **T3 – Glyph Placement**, by providing a clear overview of the map with no occlusion, and clearly encoded geo-spatial context.

5.4.3 Glyph Selection

We provide the user four common glyph design options to represent the data (see Table **??**). We chose these four typical options due to their common occurrence in geo-spatial literature [**?**]. However, the principles we describe can be applied to any multi-dimensional glyph. The user can switch between each glyph design at any point once the hierarchical data structure has been built. These glyph options are:

Pie Chart: Pie charts are an easily recognizable and practical design, making it a suitable option to present multivariate data. Pie charts are primarily used to present distribution per geospatial area, where the angle of a segment is mapped to each data dimension proportionally. See Table **??**.

Polar Area Chart: Originally published by Nightingale [**?**], a polar area chart is another radial plot but with equal segment angles. The radius or each slice corresponds to the values of each dimension, which facilitates comparison between geo-spatial areas. The polar area chart features different names including the wheel, coxcomb, or wing chart. See Table **??**.

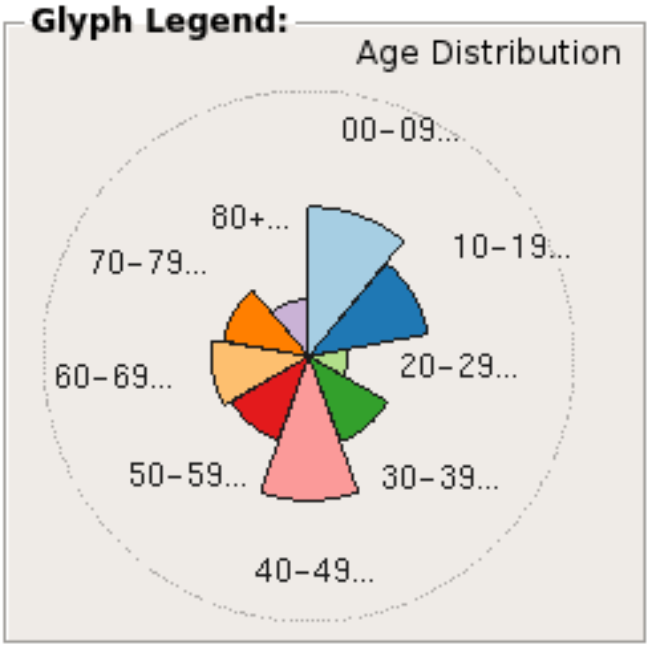


Figure 5.4: A representation of a glyph legend. The data represents the prevalence of population per age range. The dotted circle represents the full scale of the glyph or the largest value for each dimension. The glyph legend shows the average values over the whole data set. Refer to Section **??**.

Bar Chart: The bar chart is one of the most visually recognizable visual designs. Values are assigned to bar heights. aligned to the horizontal axis for easy value comparison. See Table **??**.

Star Glyph: Originally presented by Siegel et al. [**?**], a star glyph presents values using lines originating from the same point, at equal angles. The endpoints connect to form a unique polygon based on the length of each line. See Table **??**.

This addresses the requirements for **T2 –Multivariate Maps**. We choose four standard glyph designs as a proof-of-concept. Glyph placement, not glyph design is the focus of this paper. The principles we present can be extended to any multivariate glyph.

5.4.4 Adjusting Level-of-Detail with Glyph Density

Adjusting glyph density can be handled in two different ways. First, we give the user a slider which depicts m, a minimum area requirement. The parameter m represents a percentage of screen space. This is used as the primary variable for the depth-first search (DFS) discussed in Section **??**. We also allow the user to interactively zoom in or out of the map. This changes the visible extents of the map, modifying the screen space covered by each area.

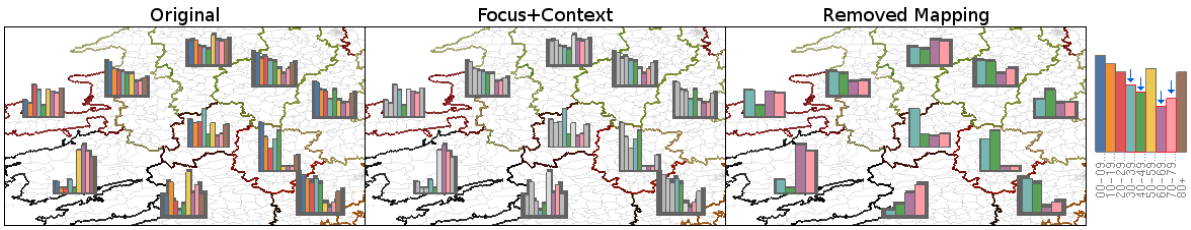


Figure 5.5: Our filtering options. The left shows our original image. The center shows our focus+context rendering, which represents the context in greyscale. The right image shows our mapping filter, which redraws the data based on the focus dimensions. The legend indicates the focus dimensions using the blue arrows. Refer to Section **??**. The Munster area refers to the bottom-left glyph, which is notable for Case Study 3, Section **??**.

These options enable the rendering of perceivable glyphs, meeting the requirements for **T4 – Occlusion**.

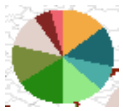
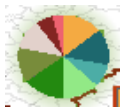
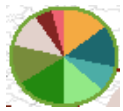
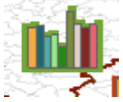
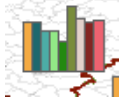
Glyph Design	Hidden Density Indicators			
	Outline	Size	Shadow	Size + Outline
Pie Chart				
Polar Area Chart				
Bar Chart				
Star Chart				

Table 5.1: Previews of the different glyphs, and the hidden density indicators provided in the application. Each glyph represents the same area, reflecting the same hidden indicator values, and attributes. Refer to our third case study, Section **??**, for more details on the values.

5.4.5 Smooth Merging and Splitting Transitions

In order to increase the fluidity of user interaction and changes to glyph size when zooming or manipulating m, we apply smooth transitions to child glyph merging and parent glyph splitting. When the user reduces the number of glyphs by either zooming out of the map or increasing the level of detail, glyphs translate towards the origin of their parent in the hierarchy while the opacity is reduced until it is no longer visible. The parent increases in opacity until it is fully opaque, creating a smooth transition. When adding new glyphs (zooming in or reducing minimum scale), the new child glyphs translate away from their parent and increase in opacity to provide a similar effect. Using this technique, we fulfill the requirement for **T6 – Fluid Interaction**. See Figure **??**.

5.4.6 Dynamic Average Glyph Legend

We provide a dynamic average glyph legend to present how the multivariate data dimensions of the glyph are encoded. Each variate is given a label, which provides context to the user about what is presented. The data used to present the glyph is made meaningful by visualizing the average value of each dimension. In Figure **??**, we can see that there seem to be some extreme values for the 80+ and 20–29 range, causing the average per area to be quite small overall.

5.4.7 Attribute Filtering

Our first filter option is to re-calculate the glyph design with only the toggled dimensions. Each data dimension can be toggled using a check-box incorporated directly into the glyph design. This allows the user to focus on or emphasize data dimensions that may reveal trends. We support user filtering using focus+context rendering. We provide a gray-scale option which removes the color from context data dimensions, enabling easier comparison. This aids in the requirements we set forth in **T5 – Filtering**. See Figure **??**.

5.4.8 Unit Area Density Indicators

We present unit area density indicators that provide a visual queue indicating how hidden unit areas are distributed, and encourage the user to explore the visualization through multiple levels of detail. When two child glyphs merge to form a parent, the child glyphs are then hidden. Our glyph design maps the number of merges to a range of different visual indicators

that generally surround the glyph. See Table **??**. We offer four options:
Outline: Outline maps the unit area quantity around each glyph to thickness. The thickness of the outline grows as more areas fall underneath a glyph.
Size: Rather than provide an outline, the glyph's overall size increases as the glyph represents more unit areas. This works especially well with pie charts, that emulate a proportional map.
Size+Outline: Size + Outline uses a combination of the two previous options.
Shadows: Rather than an outline with a constant opacity, we enable for the user to choose a gradient, enabling less occlusion in the representation.
These unit area density indicators are inspired by the work of Chung et al. [**?**] where the indicator was effective, but used to represent another data dimension (as opposed to the density of a map). We also give the user an option to represent the indicator mapped to color. The color represents the scale the glyph encodes, as opposed to other visible encodings. This enables the user to gain an understanding of how manipulation of glyph density can affect the map if a transition is made. See Figure **??**. This addresses our requirements of **T4 – Occlusion**.

5.4.9 Interactive User Options

We provide additional user options to support **T5 – Filtering**. We present a range of user options including value range filters, advanced focus+context rendering options, estimated glyph placement, and context administrative areas.
Data Range Options: We provide data range filtering to enable customized local and global design options for dimension encoding. On a local range, the user can shift the value range to present the data dimensions based on the values found in the leaf nodes (the original dataset), or clamp the ranges amongst those that are currently being rendered to enable a more accurate data range to compare data dimensions. We also support global range options by enabling the user to depict each variant based on its own range, or by creating a range based on the highest and lowest value of all mapped dimensions.
Advanced Filters: We include two advanced filters to render focus+context for the user. For numerical values, the user can present focus+context based on values higher or lower than the average value per data dimension.
Color Map: We provide the user with a variety of color maps, selected from published research papers, including ColorBrewer [**?**] (Refer to Table **??**) and Colorgorical [**?**] (Refer to Figure **??**).
Glyph Scaling: We allow the user to scale the size of the glyph. This enables the user to explore a ratio between the minimum scale and size of glyphs that meets their own data.
Naive estimated glyph placement: Using the size of the glyph, we can support the user to

make an estimation of the minimum screen space necessary to remove occlusion with the use of a button. This makes it easier to obtain a starting point, in order to decide the design of the map they would like to use.
Context Administrative Areas: We can provide additional context behind the areas by rendering every leaf area in a context view, which is shown in Figure **??**.

5.5 Evaluation

We evaluate our glyph placement for multivariate maps in two ways. First, we provide three cases for the use of the multivariate map with varying data sources. We then provide a comparative evaluation of our glyph placement strategy against a standard Cartesian grid-based glyph placement to evaluate its effectiveness and any advantages or potential drawbacks against pre-existing techniques.

5.5.1 Case Studies

In order to evaluate our glyph placement strategy, we incorporate three case studies. In our first case study we examine health indicators coupled with CCGs within England. Secondly, we examine the average income of US counties over 10 year periods. Finally, we look at the age distribution across the electoral divisions of the Republic of Ireland.

Case 1: England's Clinical Commissioning Groups (CCGs)

Our first case uses a dataset focused on England's Clinical Commissioning Groups, which represent areas of NHS practices, meaning that all people who reside in the area are generally expected to use the same practices. We explore the prevalence of afflictions per CCG area, including Dementia, Depression, Cancer, Epilepsy, Learning Disabilities, and Stroke. Refer to Figure **??** to show an example of the CCGs represented [**?**]. There are over 200 CCGs.

For this example, overlapping glyphs are prevalent around London, Liverpool, and Manchester, if we simply render a glyph at each centroid (Figure **??**, left). We start with pie chart glyphs to obtain an overview of the data (**T1 – occlusion**). As pie charts always feature as maximum radius, combining our level-of-detail glyph placement algorithm combined with the estimated minimum size placement removes most of the occlusion, enabling visual comparison between the points (**T2** – multivariate maps, **T3** – glyph placement). The first trend we notice is that depression has a majority prevalence across most CCGs, although we can ob-

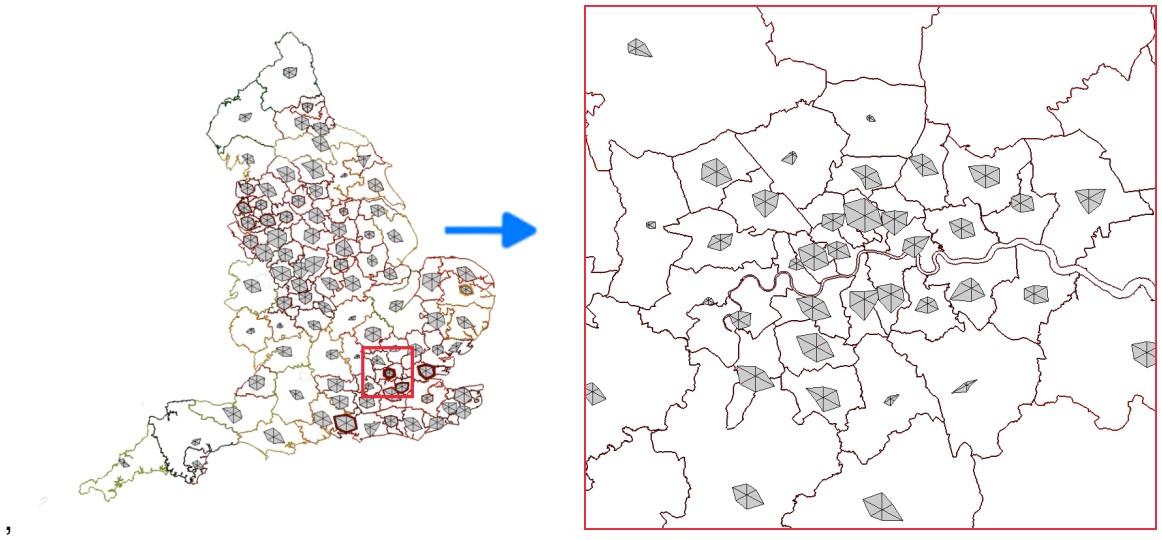


Figure 5.6: After identifying the southwest of London as having lower prevalence rates than the rest of London (red box), we zoom in to see the cause. We can see low prevalence rates are more frequent among the northwest, with some low prevalence rate in the southeast. We can also now identify the particular CCGs. Glyph scale increased for zoomed in view. See Section **??**.

serve that the Kernow CCG exhibits an uncommon distribution, caused by a larger distribution of cancer as opposed to other pie glyphs (Figure **??**). At this point, we can filter out depression prevalence, however we can glean a bit more information by switching glyph design. If we transition to the star glyph, we can see that this is due to both the larger prevalence of cancer and low rate of depression in comparison to other prevalence values for CCGs (Figure **??**(a)). As the star glyphs have varying extents, we can reduce m down to 0.8% to increase the level of detail with no occlusion (**T4** – occlusion). At this scale, London is split into 3 zones, where we can clearly see the northwest point has lower prevalence overall (Figure **??**). We can investigate this by zooming in to London (**T6** – fluid interaction). We zoom in to see a larger number areas (rendered by m). Not only do we find Barnet, Enfield, Hillingdon, and Hounslow to have low prevalence rates overall, but Bromley and Croydon in south London also show these signs (dementia, stroke, and cancer prevalence in particular). See Figure **??**.

Case 2: Counties of the United States

Our second example explores counties in the US. We look at the average income over 40 years for each county in the United States from 1979, 1989, 1999, to 2009 in 10 year increments [**?**]. The US consists of over 3,000 counties.

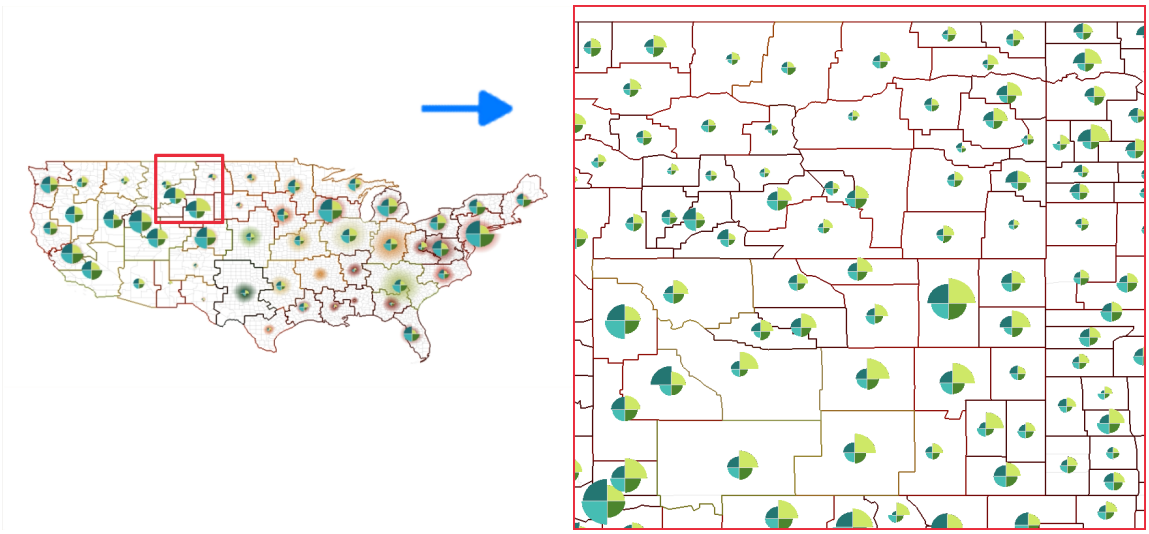


Figure 5.7: We notice counties around Wyoming and Montana have are higher average income in 1979 and 20019 than usual, We zoom in, and can verify this amongst particular counties. Glyph scale increased for zoomed in view. See Section **??**.

Rendering the glyphs presents a large frequency of occlusion and therefore we use the estimated minimum size, m, to reduce the large number of glyphs to something more accessible. Starting with the pie chart shows a standard distribution where the average income increases per time period (**T1** – overview). Since each glyph represents a number of areas, we adjust the range indicator to represent areas that are rendered, and switch to a polar area chart to visualize the data (Figure **??**(b)). The wheel glyph shows higher income on the east and west coasts, with the lowest value glyphs across the center of the United States. Wyoming and Montana have some uncommon behavior, where 1979 and 2009 show much stronger average income than their other variants. Zooming in, Wyoming exhibits a tendency to exhibit a higher average income in 1979 over the 40 years, independent of their standing amongst the rest of the US counties. The counties of Sublette and Teton are the exceptions to this which hold stronger mean incomes in 1999 and 2009. See Figure **??**.

Case 3: Electoral Divisions of the Republic of Ireland

For our final case, we look at the electoral divisions of the Republic of Ireland. Our data set looks at population distribution across each division, which is split into nine groups, 0–9 years old, 10–19, 20–29...up to 80+ [**?**]. There are over 3,400 electoral divisions in the Republic of Ireland.

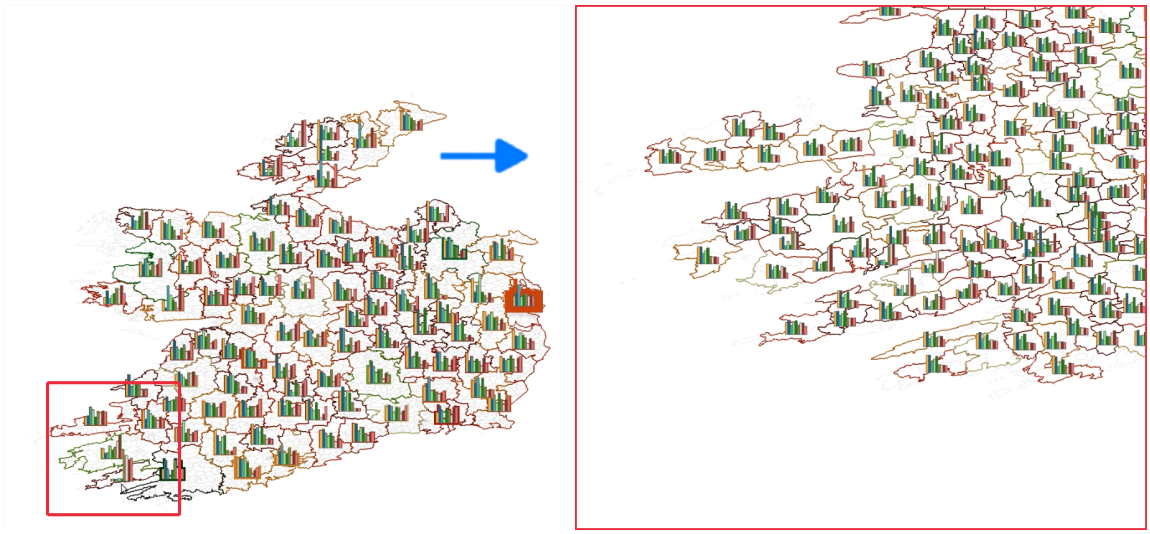
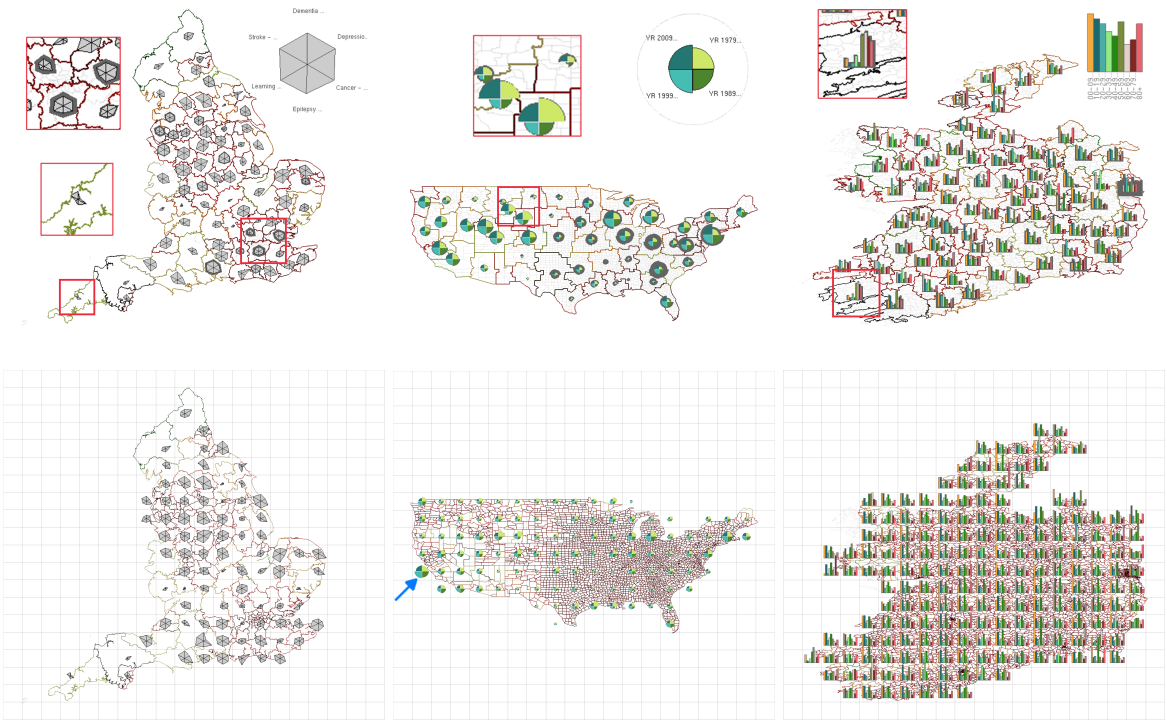


Figure 5.8: After noticing a strange inconsistency in the south-east of the Republic of Ireland, we zoom in and can verify that this trend can be found amongst a selection electoral divisions. Glyph scale increased for zoomed in view. See Section **??**.

Similar to Case 2, there are a large number of electoral divisions so we immediately choose to reduce the visible areas to a perceivable number using the estimated scale glyph placement, and adjust a filter to represent a clamped range. In this example, we use bar charts to represent the data. If we look towards Munster, we can see an unusual population distribution, where the proportion of the population, 50 or above, is uncommonly high, and the proportion of people, under 50, is uncommonly low (Figure **??**). Zooming in, Glanmore, Canuig, Tahilla, Derriana, Dawros, Ardea, Castlecove, and Caher seem to be the leading factors in this trend. See Figure **??**.

5.5.2 Comparative Evaluation: Grid-Placement vs. Dynamic Placement

We evaluate user interpretation of the data against a typical grid structure for glyph placement. We use a Cartesian grid (20^2) structure that places glyphs at approximately the same size and resolution as our presented process. Each area is assigned to a cell of the grid, closest to its centroid, where glyphs are derived using the same process as our algorithm. In terms of design, we try to keep both structures similar. In our algorithm, we use a thickened outline to signify the unified area the glyph presents which is not possible for the grid placement version because unit areas are arbitrarily split using a Cartesian grid. We therefore show the presented areas using a lower line width to avoid over representation. Other than this,



(a) **(b)** **(c)**

Figure 5.9: Glyph-placement comparison: top row – our method, bottom row – grid-based. (a) Comparison of CCGs (b) Comparison of US Counties (c) Comparison of Ireland's Electoral Divisions. Glyphs based on grid-placement are often de-coupled from the geo-space the represent. The blue arrow signifies the cause of the incomparable data values. See Section **??**.

all design elements are the same and we allow the user to adopt filters and user options identically. However, we use a standard grid structure and therefore the grid structure does not necessarily handle multiple levels of detail. Examples are shown in Figure **??**.

We look at two main aspects of placement, geo-metric coupling, and value representation. First we examine geo-metric coupling. As the grid is uniform, in all instances, the grid allows for a larger number of glyphs, however, this can be seen as a clear positive. If we start with Figure **??**(a), it is sometimes difficult to verify where a glyph is when areas reside within a corner of multiple grid slots. If we look at the central-east coast of England, identifying even large areas becomes difficult. This is because the areas are considered uniform, and therefore are distorted uniformly, as opposes to our presented algorithm which attempts to avoid this as much as possible. In examples Figure **??**(b), our placement uses fewer glyphs due to the large variance in wheel glyph extents, as opposed to the grid placement that presents roughly twice as many. The grid results in the same limitation as above, with a strong difficulty in understanding how values are mapped to their glyph counterparts. We run into a second problem with the density of the areas, where administrative areas make it difficult to perceive where a grid cell covers, providing little understanding of the context. Both of these problems follow on to Figure **??**(c).

For value representation, the algorithms do show differences, which is to be expected, in accordance with the modifiable areal unit problem [**?**]. For Figure **??**(a), both representations pointed to the same observation in our case study. Figure **??**(b), exhibits a significant difference. The grid-placement greatly skews the value representation of the US counties, due to some grid elements covering secluded cells. The west coast contains a grid cell with San Francisco, which causes most of the other glyphs to be quite small, independent of the data range option selected. It becomes difficult to compare the two placement options. Although that is the case, both placement schemes lead to the observation found that the time period of 1979 maps to a larger segment in Wyoming. Figure **??**(c) also presents the observations found in our case study, although the concentration is more spread out, which can be considered better for examinations. If we consider the ability to zoom, we feel that only the Cartesian grid representation of the Republic of Ireland can lead to a fair comparison for observation, and this is only based on trial-and-error.

5.6 Conclusion

We present a glyph placement algorithm supporting multivariate geospatial visualization at different levels of detail. We discuss how we create scale aware map, and apply the process to glyph placement. We also discuss the different glyph options and filters we have designed

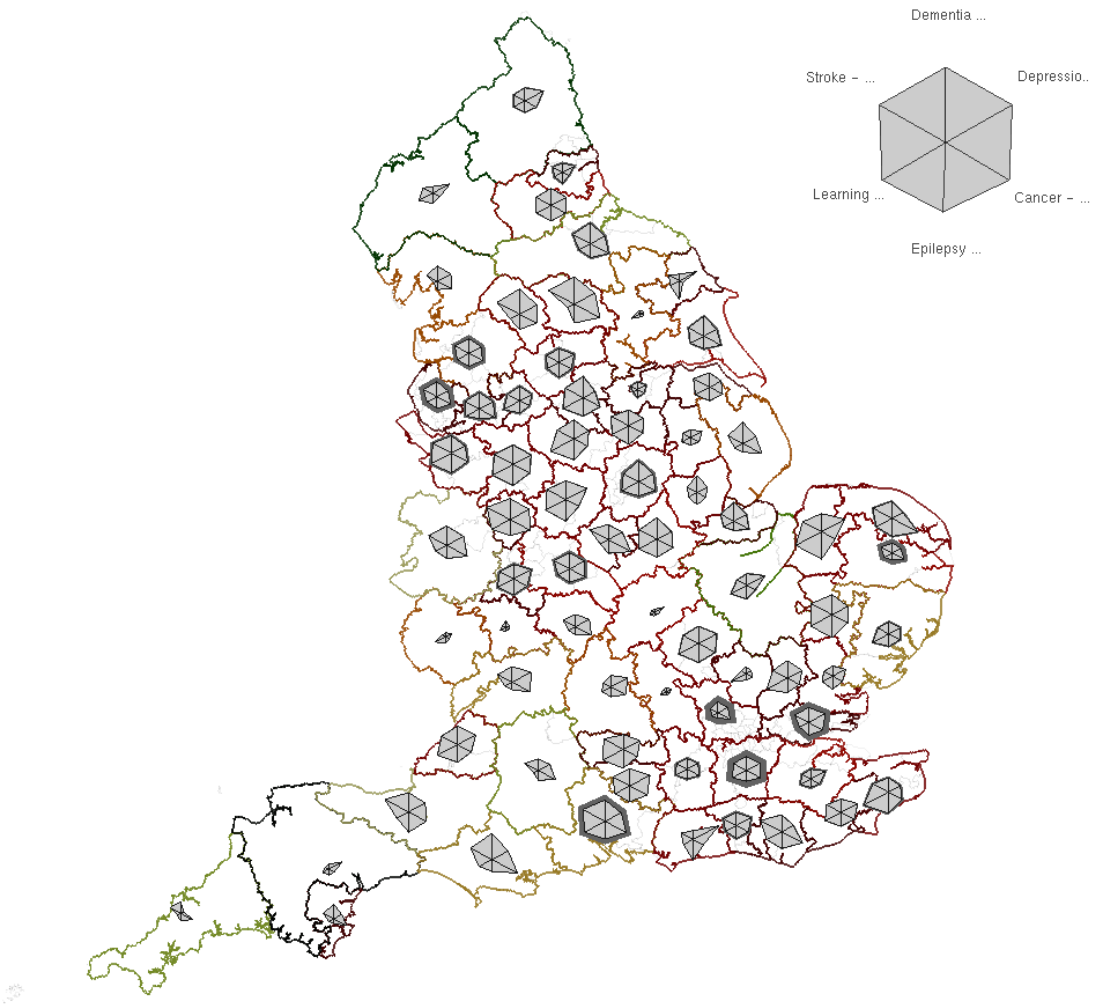


Figure 5.10: Standard representation of the glyph placement algorithm using the CCGs of England [**?**], with star glyphs. The glyph represents the prevalence of afflictions per CCG area. The legend represents the average prevalence across all glyphs.

to support exploration of multivariate data. Finally, we review the algorithm by examining the separate use cases, and compare against a pre-existing glyph placement strategy.

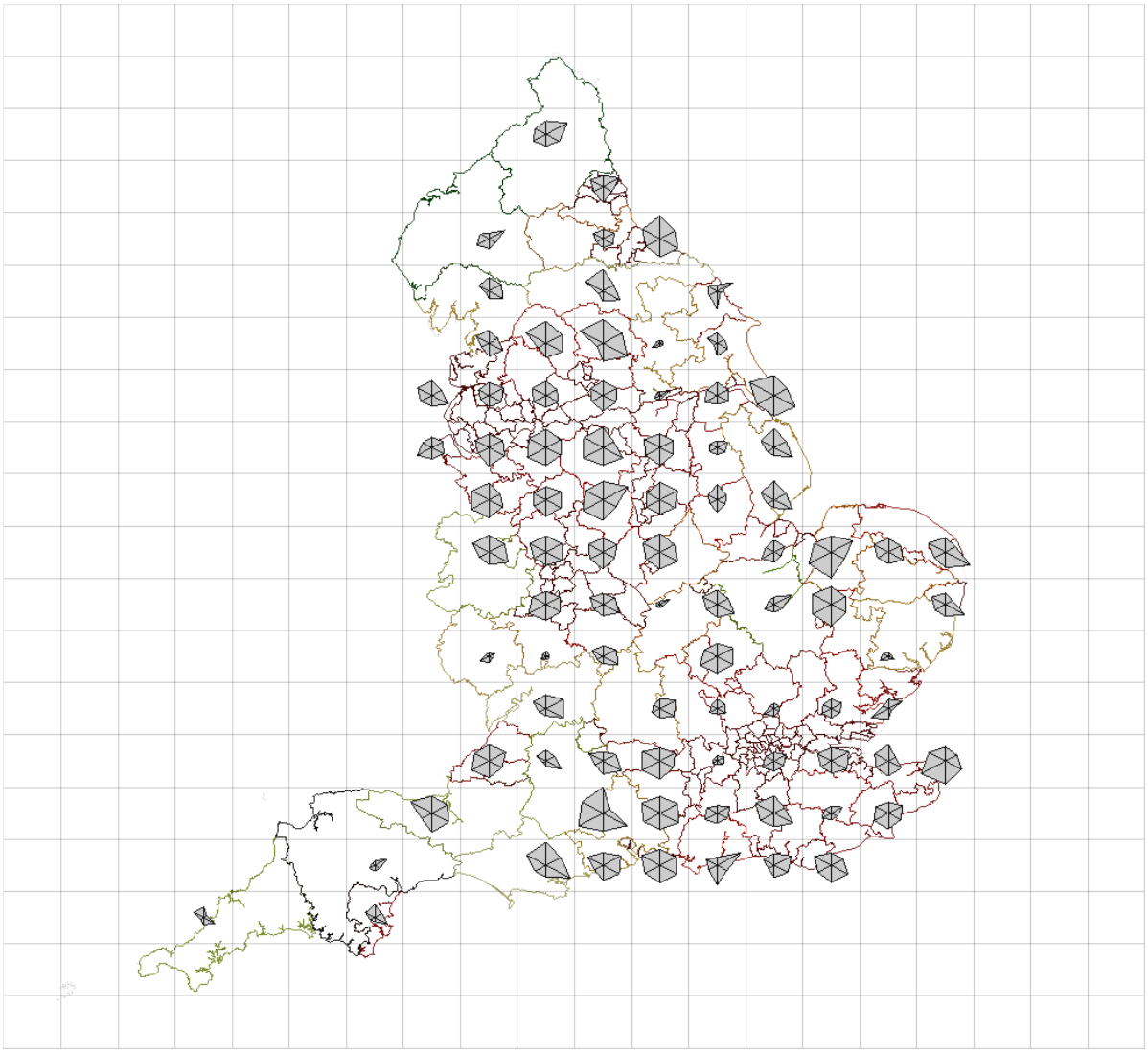


Figure 5.11: A comparison piece for our CCG sample in Figure **??**. The grid-placement is a 20^2 grid and assigns values in a similar fashion to our concept.

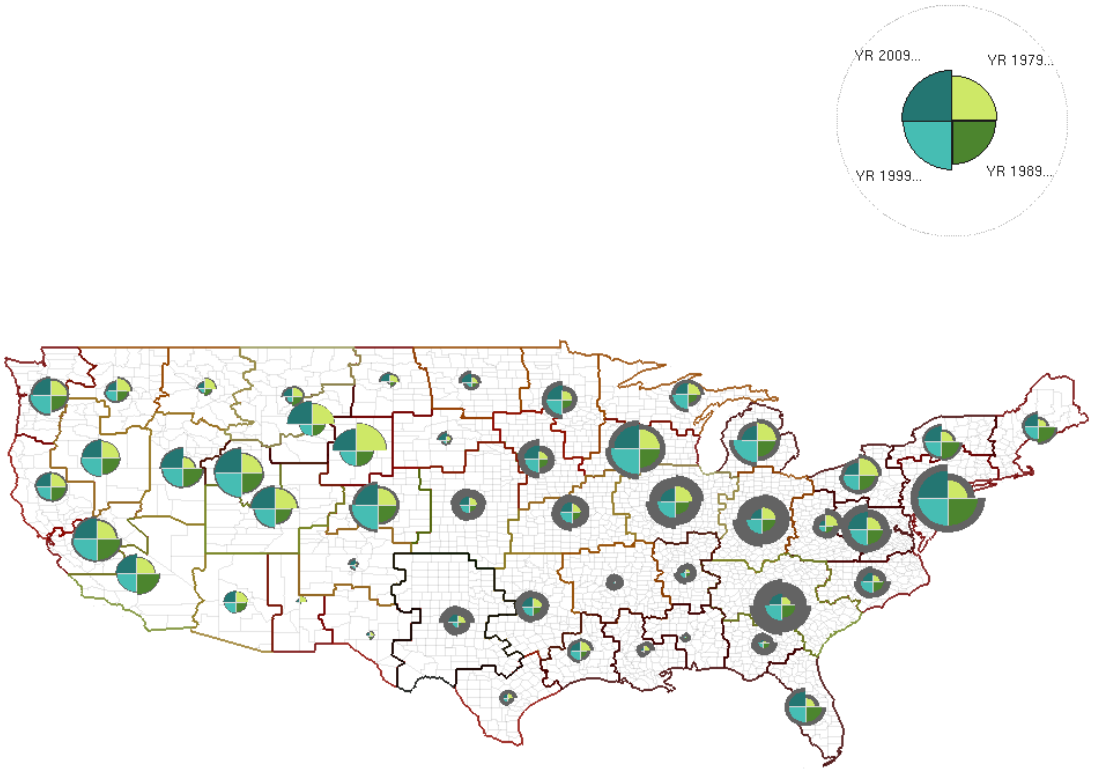


Figure 5.12: Standard representation of the glyph placement algorithm using counties of mainline United States. [**?**], with polar area glyphs. The glyph represents the average income per household over four time periods, 1979, 1989, 1999, and 2009. The legend represents the average income across all glyphs.

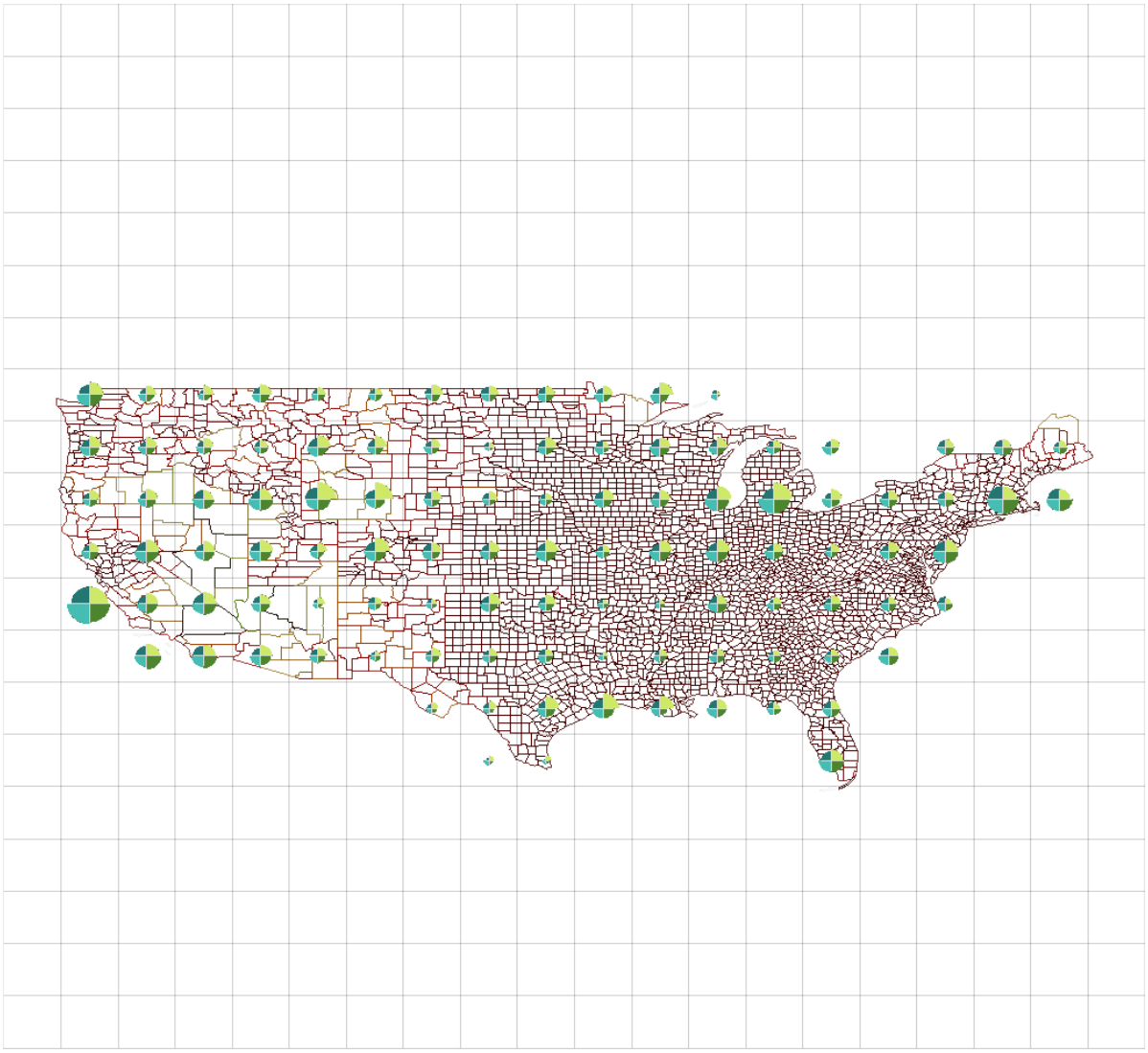


Figure 5.13: A comparison piece for our US counties sample in Figure **??**. The grid-placement is a 20^2 grid and assigns values in a similar fashion to our concept.

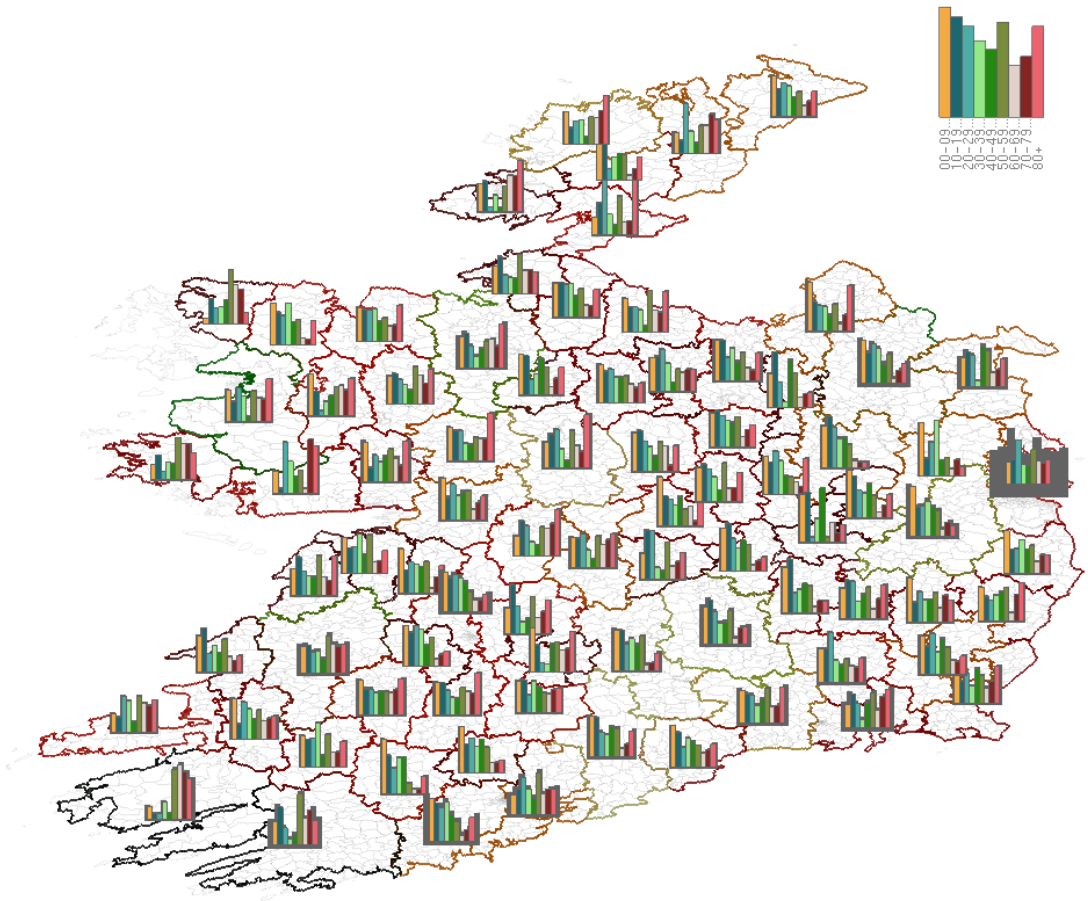


Figure 5.14: Standard representation of the glyph placement algorithm using electoral divisions of the Republic of Ireland. [**?**], with bar glyphs. The glyph represents the age distribution per electoral division in 10 year increments up to 80. The legend represents the average age representation across all glyphs.

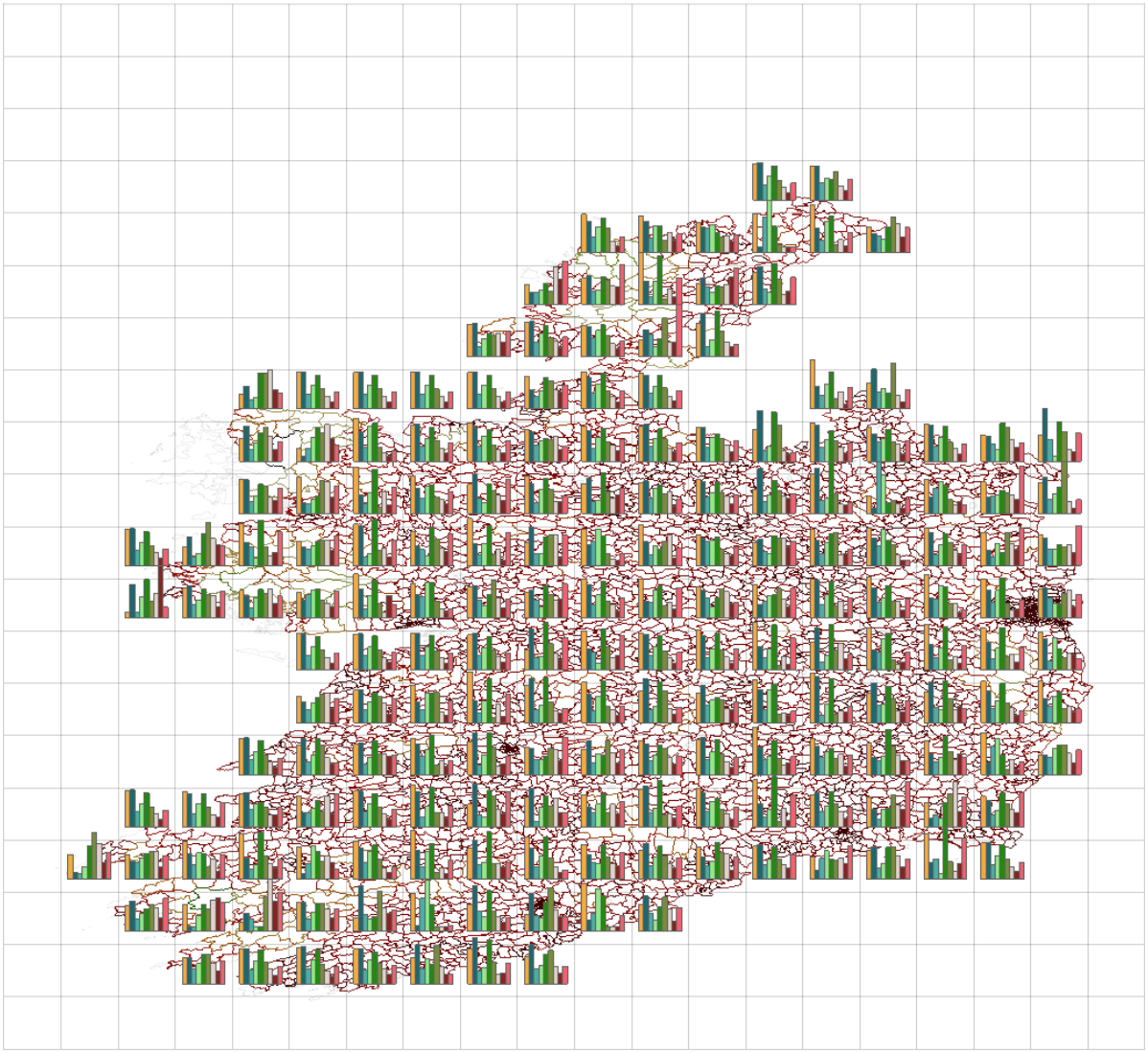


Figure 5.15: A comparison piece for our electoral divisions sample in Figure **??**. The grid-placement is a 20^2 grid and assigns values in a similar fashion to our concept.

Chapter 6

Software Design and Development

"It should be trivial."

— Dr. Robert S Laramee, 2016

Contents

6.1 Introduction and Motivation

Throughout the course of this thesis, we implemented two software-driven chapters as well as a user-study developed using custom software. There are many design challenges and obstacles that ocurred throughout the development of software components and this chapter highlights some of the major themes. We focus on three specific topics for this chapter: debug visualization, geometric primative testing, and the complexities of boundary identification.

6.2 Debug Visualization for Geospatial Maps

When creating complex algorithms for diverse data sets, there are many opportunities for errors or oversights to present themselves. Even when we discover them, it can be difficult to determine the cause without efficient debugging strategies. This concept is important when working with geospatial data. We present some debug visualization techniques that are used to process the representation of geospatial data for the dynamic choropleth map algorithm (Chapters **??**, **??**, and **??**). Although Laramee has already presented some work on this topic [**?**], we extend this with concepts specific to geospatial visualization, and our specific experience.

6.2.1 Debug Geospatial Visualization Concepts

In this section, we will discuss some concepts which are incremental but still important to efficient debug visualization.

Step Function: One of the most important tools when using debug visualization is the proper use of a step function. A step function is used to stop-and-go through different stages of a visualization process incrementally. In our project, a step function can be toggled on and off using a checkbox. This enables us to check whether the intermediate results are as intended for each section of the algorithm (discussed in the proceeding sub-sections), at each interval (intervals discussed in detail below).

Color Mapping: It is important to distribute roles clearly using color. Color is an essential tool in any visualization and debugging is no exception. For our algorithm, we used grey to signify context and blue to signify the focus of the current processing step. If there is no obvious way of determining the number of colors needed, the use of seeded colors can be useful. This is shown in Section **??**.

Test Data sets: Testing an algorithm on multiple data sets is a key aspect of debugging your software and is a concept not exclusive to the visualization process of debugging. We worked

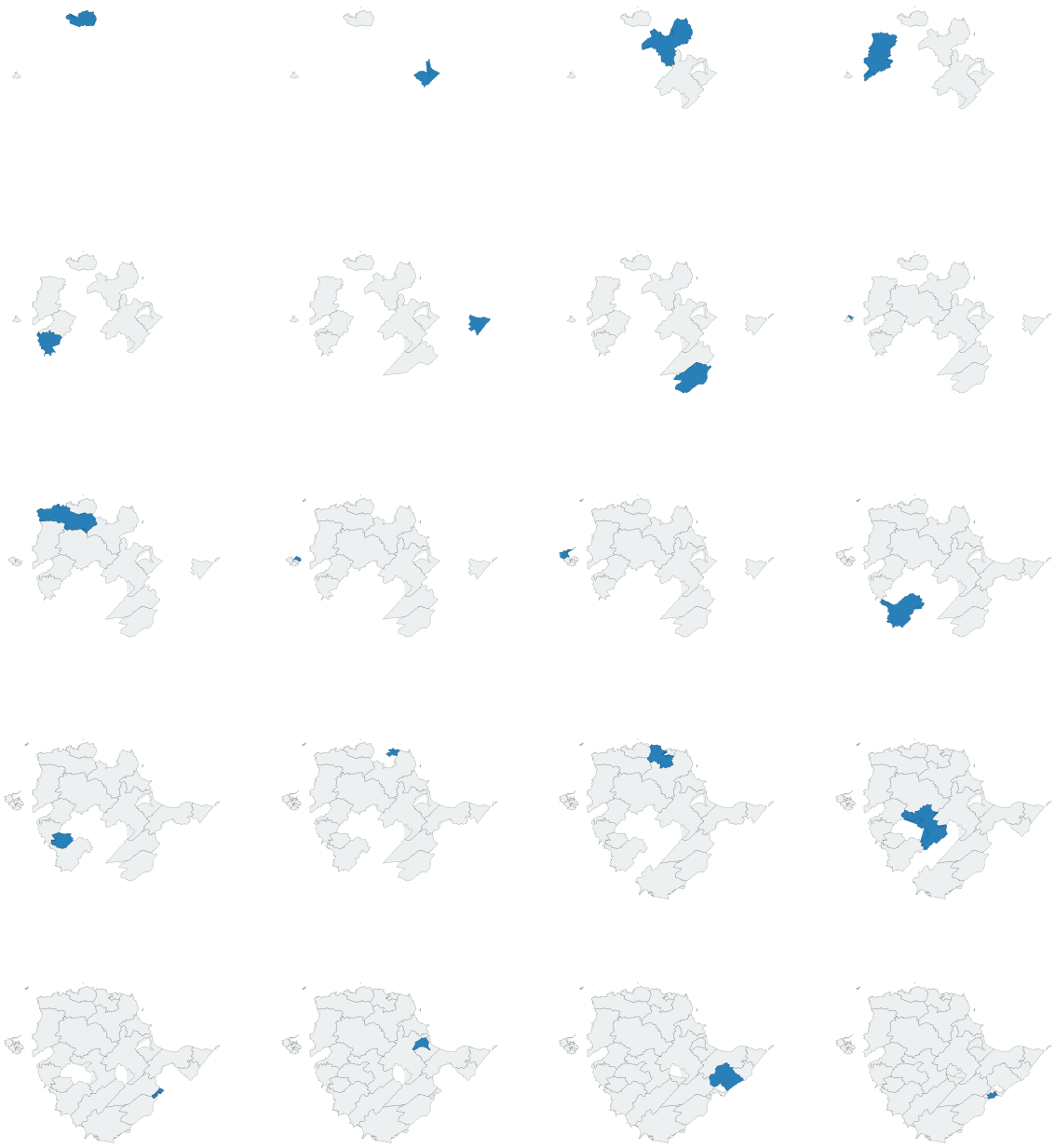


Figure 6.1: A matrix depicting time slices of the temporal debug visualization for geospatial debug loading. The blue presents the last loaded polygon, whilst they grey represent polygons that have already been loaded. The file represents a test data set taken from the LSOA's of Wales, specifically the contiguous mainland of Anglesy [**?**].

with over ten test example data sets. These datasets are split into smaller, more manageable sub-sets. The data sets ranged from 20 polygons up to over 30,000 polygons. If an error occurred during the algorithm run, the area in which the error occurred was identified, investigated and split into a smaller data set to add to our test runs.

Comparative Software: Software that handles similar data can be an essential tool in interpreting errors. We thank QGIS [**?**] for helping us understand why errors may occur, and what should be expected from the test data sets.

Multiple Views: In order to gain the most out of our debug visualization, we enable two different viewpoints to be toggled. The first viewpoint we used is an overview which presents the map in full. This allows the user to understand how far the algorithm has progressed accurately. We also allow the user to move the extents to follow the current processing step's result, which we identify as a focus processing step. This allows for a clearer representation of the current processing step, letting us observe errors quickly. Using both of these views are essential in the progression of the algorithm. Figures **??** and **??** are examples of how the views can differ during the debug process.

6.2.2 Geospatial Data Loading Verification

Debug verification of polygon loading is a simple concept. Each step presented the last loaded polygon mapped to the color blue, while previously loaded areas are rendered as grey context. See Figure **??** for a visual representation.

Adding visualization techniques to this step was more important than is first apparent. We used open source library Geospatial Data Abstraction Library (GDAL) [**?**] to load shapefiles. However, we noticed that during the representation of geospatial data, some areas would not load, which we learned using comparative software, QGIS. It was clear the general shape of maps differed between our software and our comparitive software. Using loading verification, we found data was not being loaded from the file, rather than it being lost or corrupted in some processing steps. This was due to the mishandling of polygon types due to the way GDAL handles "polygons" and "multi-polygons" seperately. "Inner" rings, as labeled by GDAL, caused a second challenge that we ran into. They are used to separate internal area's vertex lists from those of the primary outer boundary. Although the problem was identified later using visualization concepts, we noticed that inner rings were not represented in the loading visualization step. After further research, we found the additional vertex lists. Refer to Figure **??** for an example.

These mistakes may have been avoided by a more experienced user of the library, but geospatial debug visualization proved an essential tool for new users.

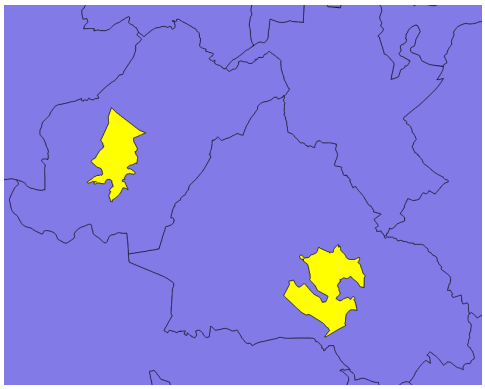


Figure 6.2: Two examples of inner rings, highlighted in yellow. These can be found in the Lower Super Output Areas (LSOAs) of Wales in “Carmarthenshire” [**?**].

6.2.3 Contiguity Visualization

We reduce the complexity of hierarchy building by introducing a constraint that all hierarchies are complete when exactly one area remains in our merge candidate list. See Chapter **??**. This is done after the areas are loaded. In order to do this, we need to make sure that every instance of the hierarchy is formed as one contiguous region. We debug this aspect by mapping a random color to each list of polygons (representing a contiguous region) at each intermediate step, where a step is presented when each polygon has its contiguity tested. As there may be an arbitrary number of contiguous regions, pre-designating identifiable colors may be difficult. Therefore, in order to address this challenge, we altered the program slightly. When a polygon is assigned to a contiguous region, a new color is assigned to the region, This allows for a visible understanding of the change without the need for the step function. We present a matrix of images showing a concise contiguity visualization in Figure **??**. The file represents a test data set taken from the LSOA’s of Wales, specifically the contiguous mainland of Anglesy [**?**].

6.2.4 Hierarchy Building Visualization

In Chapter **??** we state that over 50% of merges have at least one error case, which is why this section is so important. Our hierarchy building visualization is used to support the identification of a shared boundary between two areas. For each step, we present the newly unified area in blue and every other area that is considered a merge candidate as grey context. This enables us to observe the hierarchy at run time, as well as how it is built. This visualization type enables the user to perceive the majority of errors found during the creation of our algorithm. We present many of the archived errors, in Figures **??**–**??**.

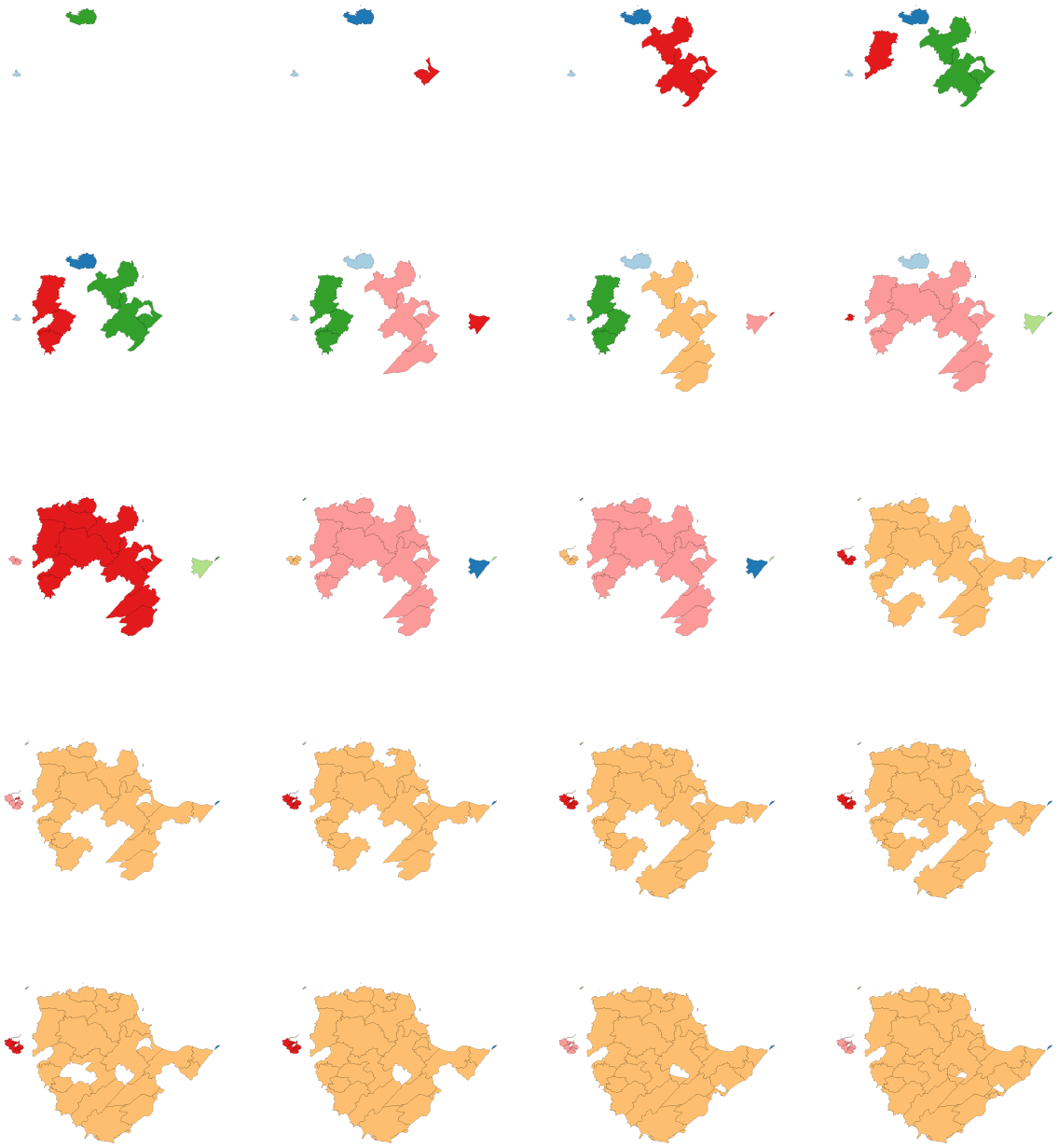


Figure 6.3: A matrix depicting processing steps of the intermediate debug visualization for contiguity debugging. Each color rerpresents a contiguous region. The color considers the number of contiguous regions, and sets the last updated contiguous region to the most recent color. Test data set is relative to Figure **??** for comparison. The file represents a test data set taken from the LSOA's of Wales, specifically the contiguous mainland of Anglesy [**?**].

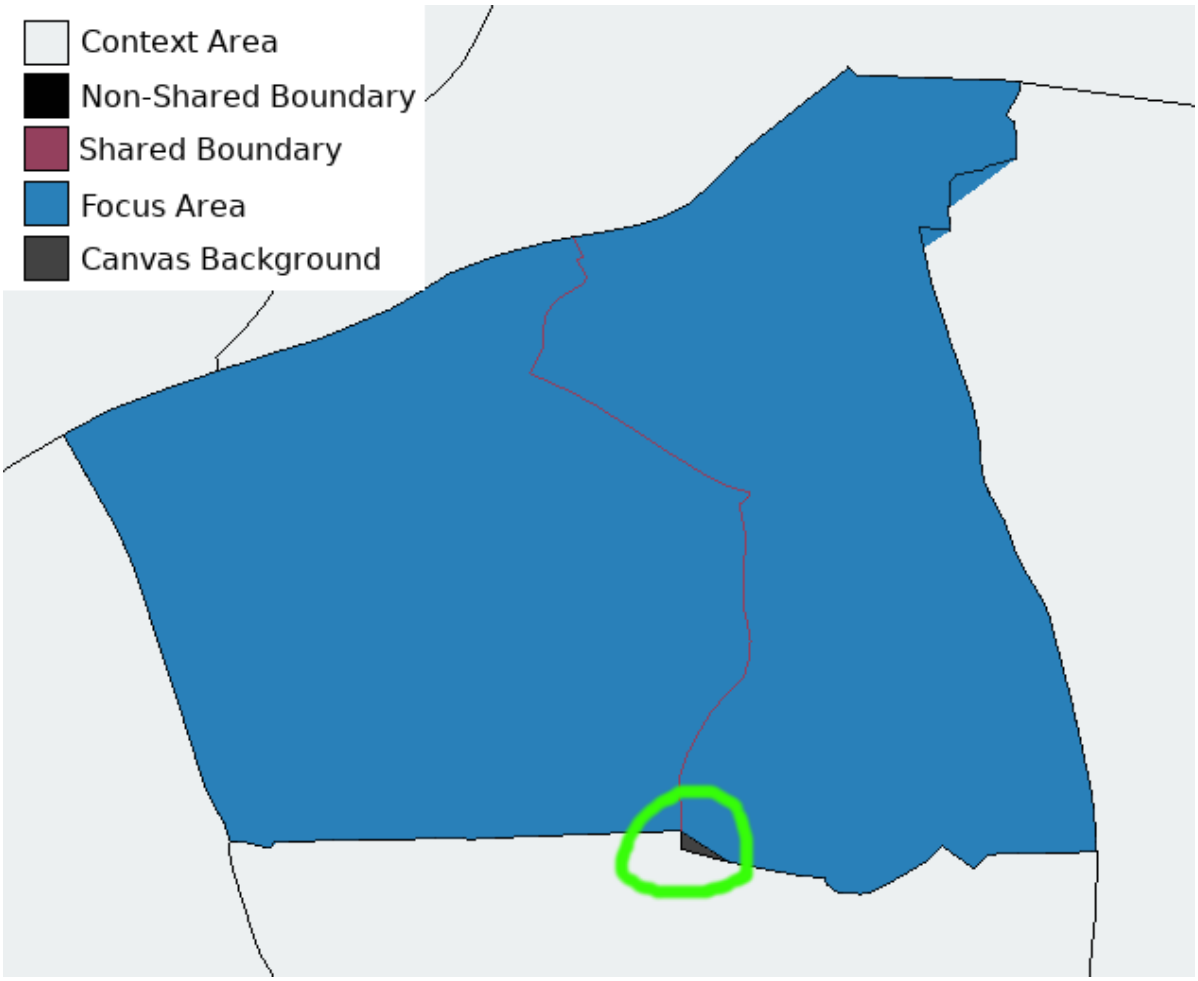


Figure 6.4: The first instance of shared boundary matching. The red shared boundary refers to the assumed shared boundary. the blue color refers to the focus area, the background color is represented by the dark grey. The red boundary seems to stop just short of the shared boundary, however there are no vertex points to join at the end causing an error. After dscovering this, the algorithm was upgraded to check both sides of a shared boundary and match them using a point-to-line intersection. We highlight the problem with a green circle.

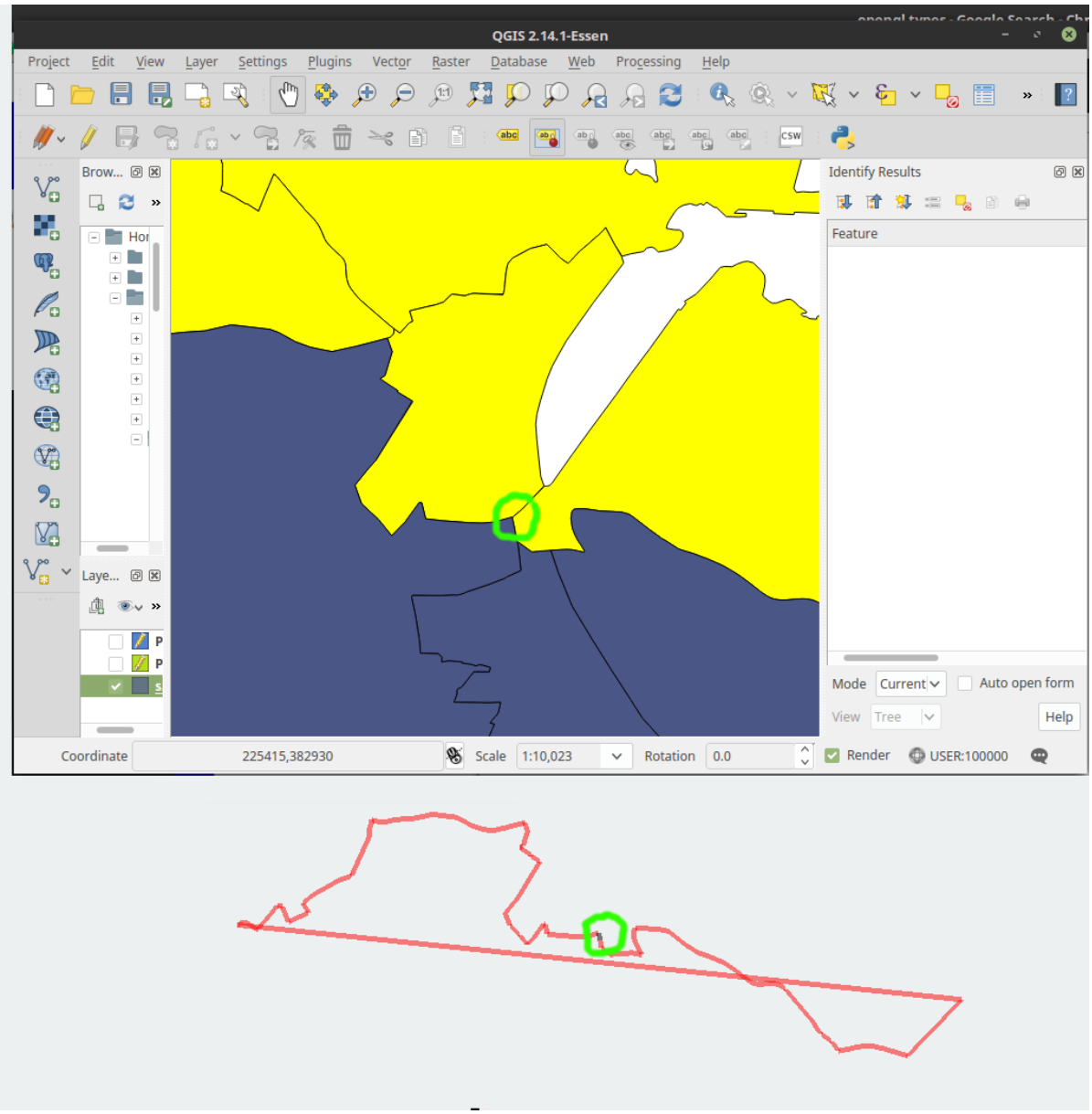


Figure 6.5: The red color represents the assumed shared boundary. The first instance of the T-junction challenge discussed in more detail in Chapter **??**. Using the debug visualization for hierarchy building, along with comparative software tools, we were able to identify the exact vertex point (circled green) that caused the problem. This Lead us to re-evaluate the handling of T-junctions. We identify the boundary as the introduction between the following areas within a LSOA of Wales test data set in the “Isle of Anglesey” [**?**].

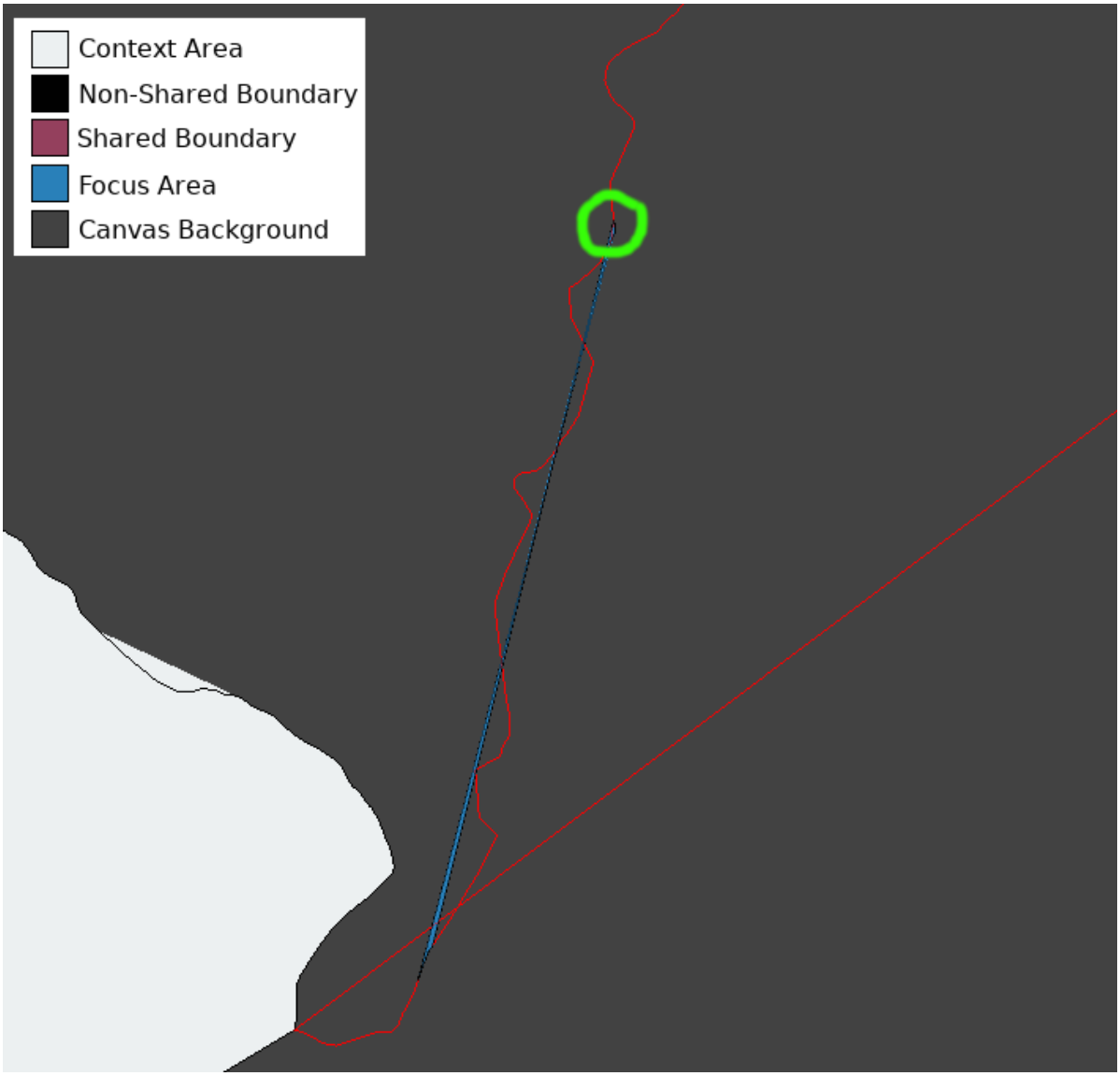


Figure 6.6: This presents a focused step presenting an error that occurred in the processing of the geospatial data. The red line presents the assumed shared boundary, the white presents a context polygon, the grey represents the default background, and the blue signifies the unified polygon. This error was caused by a fissure along the shared boundary which result in a unification error. This is because an early version of our algorithm assumed that voids would only occur in singular instances. The fissure made the logic believe we rejoined a shared boundary shape, causing most of the boundary to be lost in the unification process, depicted by the straight red line moving off the right size of the canvas which then meets up to complete a loop (see Figure **??** to see an example of how the boundary loops). Refer to Figure **??** for the location and cause, where we match the areas using a green circles.

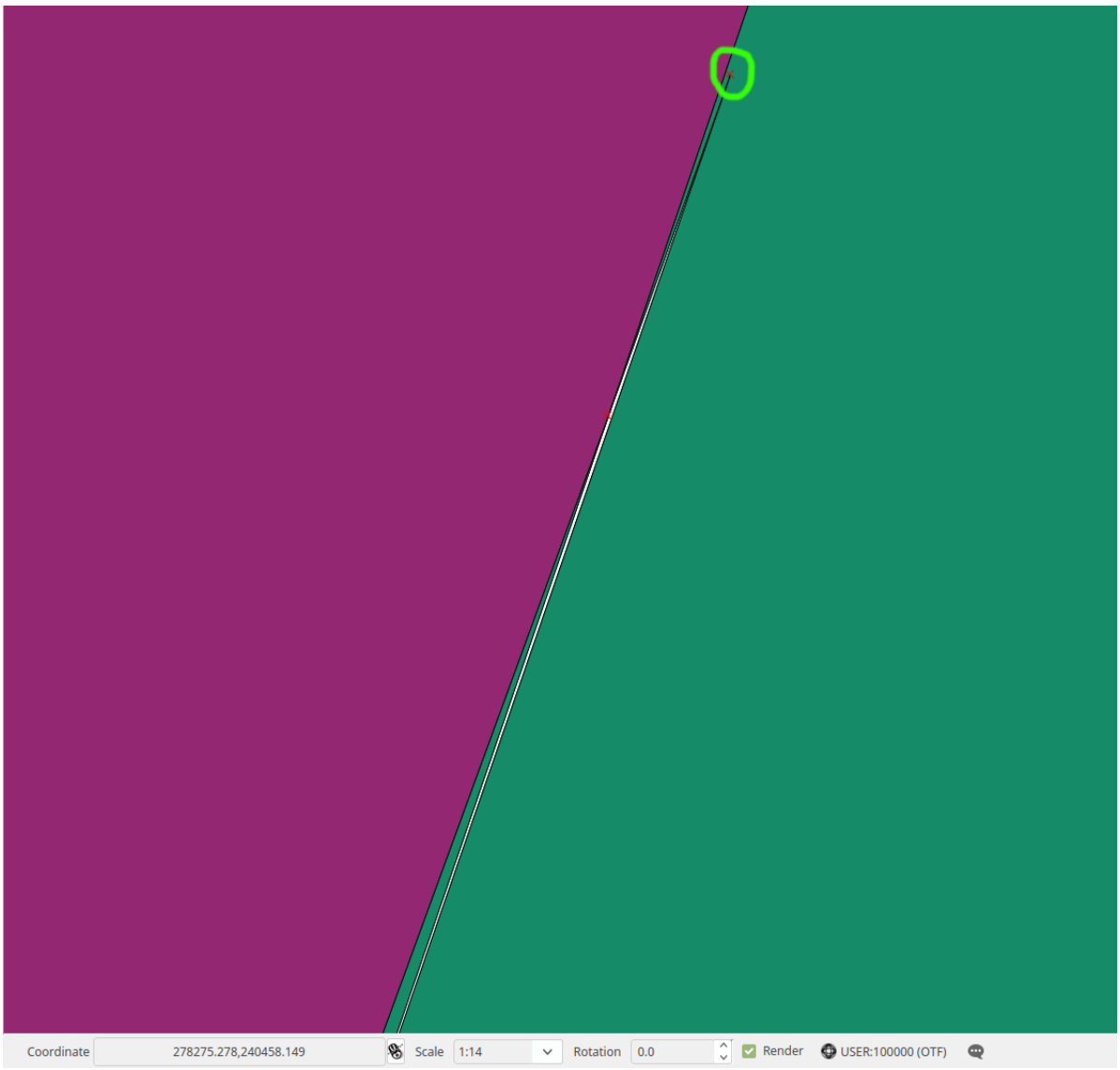


Figure 6.7: This is a screenshot taken from QGIS to verify an error found in Figure **??** The polygons contain a strange break in between them which was only visible at a 1:14 scale of the original data set. Something that would be impossible to identify if reviewing the data as an overview image. The white is the background of the software, signifying a lack of any polygonal data. The green circle relates to a point relative to the green circle in Figure **??**. The void is so thin that we can not clearly identify the all the voids in one image. This error occured between the community wards of "Cynghordy" and "Rhandirmwyn" [**?**].

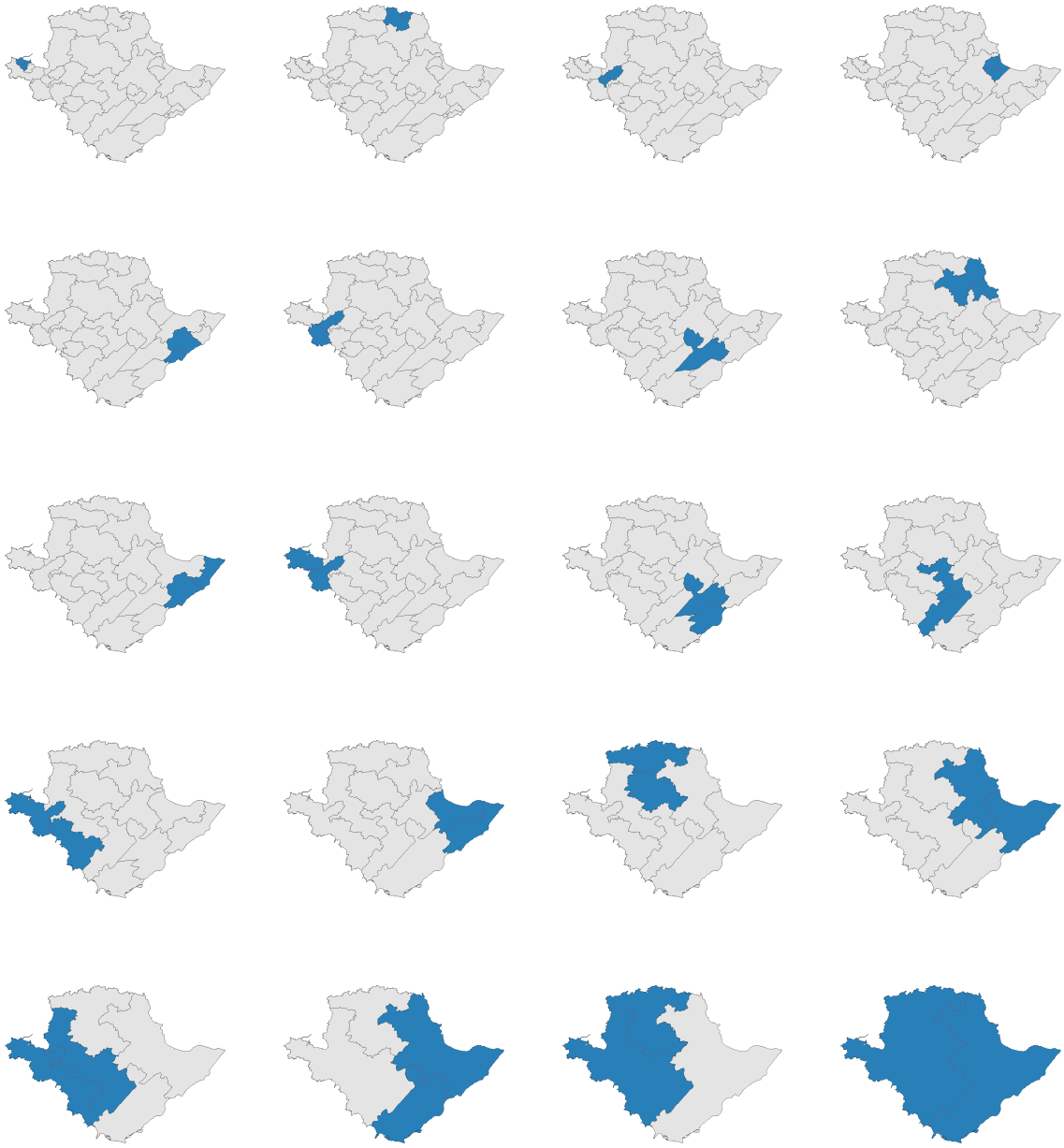


Figure 6.8: A matrix depicting processing steps of the intermediatel debug visualization for hierarchy debugging. The blue color rerpresents the most recent unified area. Grey polygons represents potential merge candidates, the red line represents the shared boundary found to unify the new area. This matrix presents the overall view of the hierarchy visualization. See Figure **??** which shows a focus view, for clearer boundary interpretation. The file represents a test data set taken from the LSOA's of Wales, specifically the contiguous mainland of Anglesy [**?**].

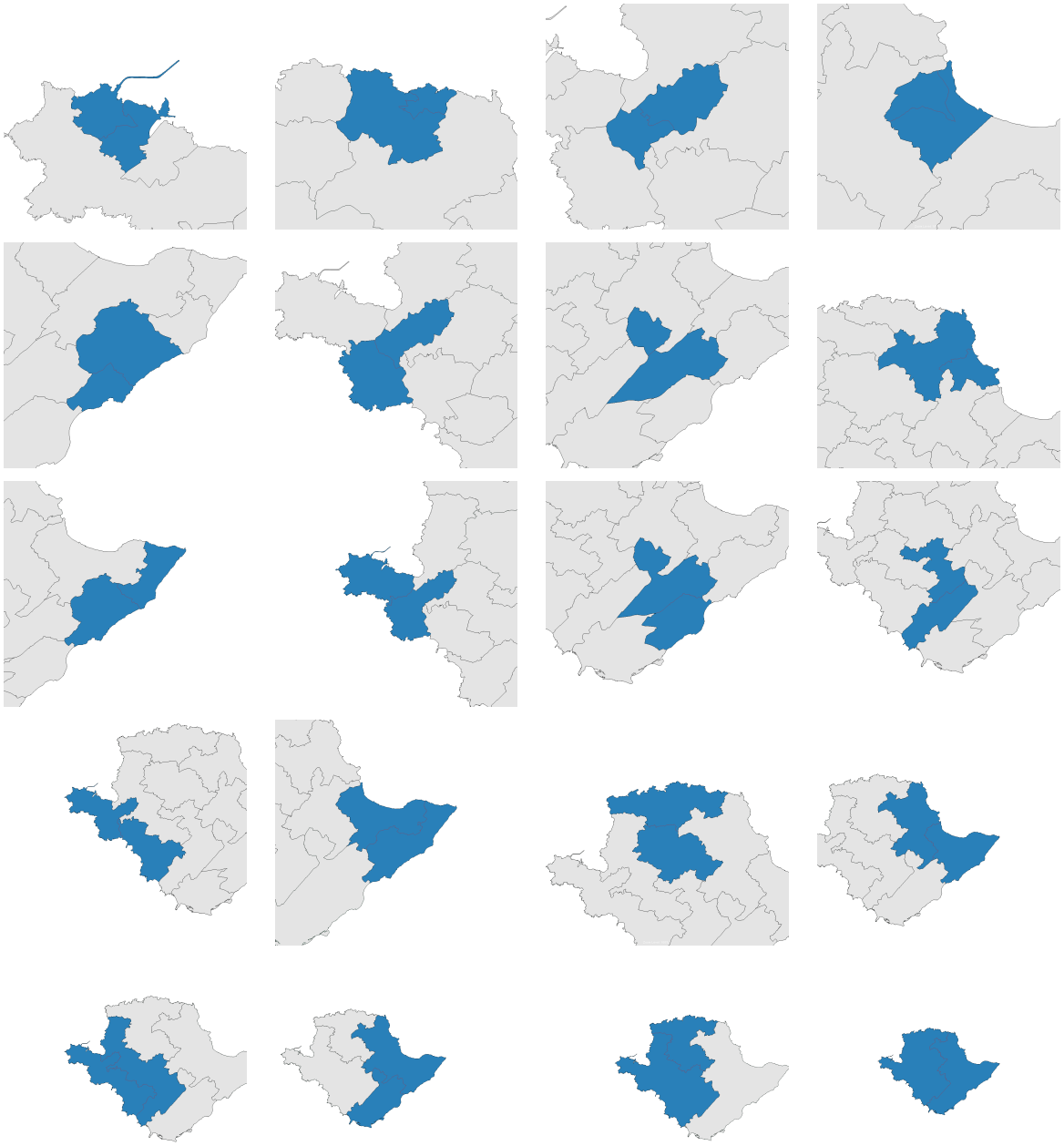


Figure 6.9: A matrix depicting processing steps of the debug visualization for hierarchy debugging. The blue color rerpresents the most recent unified area. Grey polygons represents potential merge candidates, the red line represents the shared boundary found to unify the new area. This matrix presents the focus view of the hierarchy visualization. Relative to Figure **??** which depicts the same visualization using the overall focus to clearly depict progress of the process. The file represents a test data set taken from the LSOA's of Wales, specifically the contiguous mainland of Anglesy [**?**].

The first error (Figure **??** on Page **??**) found was identified using our focus processing step, which enables the user to see minor errors in boundary design. This was due to the corresponding vertices not always overlapping resulting in shared boundaries being two different topological lengths. After noticing the problem, we added a point-to-line test to identify the closest point to the boundary, and duplicate points to ensure a consistent and clean shared boundary.

Although the second error (Figure **??** on Page **??**) is similar to the previous example, the cause is slightly different. We found that we would encounter T-junction challenges during a merge. This is caused when a shared boundary's vertices do not line up with a point that may be used on a later shared boundary. Once identified, we added the option to include point-to-line tests within a shared boundary entirely. Although this would reduce performance time, we rectify this with the ability to save and load build instructions which speed up performance during a later run of a shape file.

The next case (Figures **??** & **??** on Pages **??** & **??**) we look at is caused by what we call a fissure, which refers to space between boundary vertices. Although we considered voids as single anomalies along a shared boundary, learning that it was common to have a set of continuous voids (caused by rivers or mountains) made it impossible to ignore these anomalies. This lead us to an extensive re-factoring of the algorithm, that was used to diagnose the difference between these voids, and a non-shared boundary (refer to Section **??**).

6.3 Intersection Testing for Primatives

Intersection testing is one of the key areas that we examine when reviewing performance. Two tests, in particular, result in a high frequency of calls: the point-to-line and bounding box intersection test. The first is used as a deep search into accurate boundary representation, while the second is a filter for disjoint, non-intersecting polygons. Although there are many methods online to help manage these tests, we did not find a developer-friendly library that managed to guide us with primitive testing. Therefore, we created a small open-source library to test the intersection between five different primitives. The primitives we test are:

- Point - Defined as an (x, y) coordinate, $p(x, y)$
- Line Segment - Defined as two points (assumed to be connected), $(p_1, p_2))$
- Circle - Defined by a center point, and a radius, $(p_c, r))$
- Triangle - Defined by three points (p_1, p_2, p_3)

- Axis-Aligned Bounding Box (AABB) - Defined by four points, assumed to be aligned.

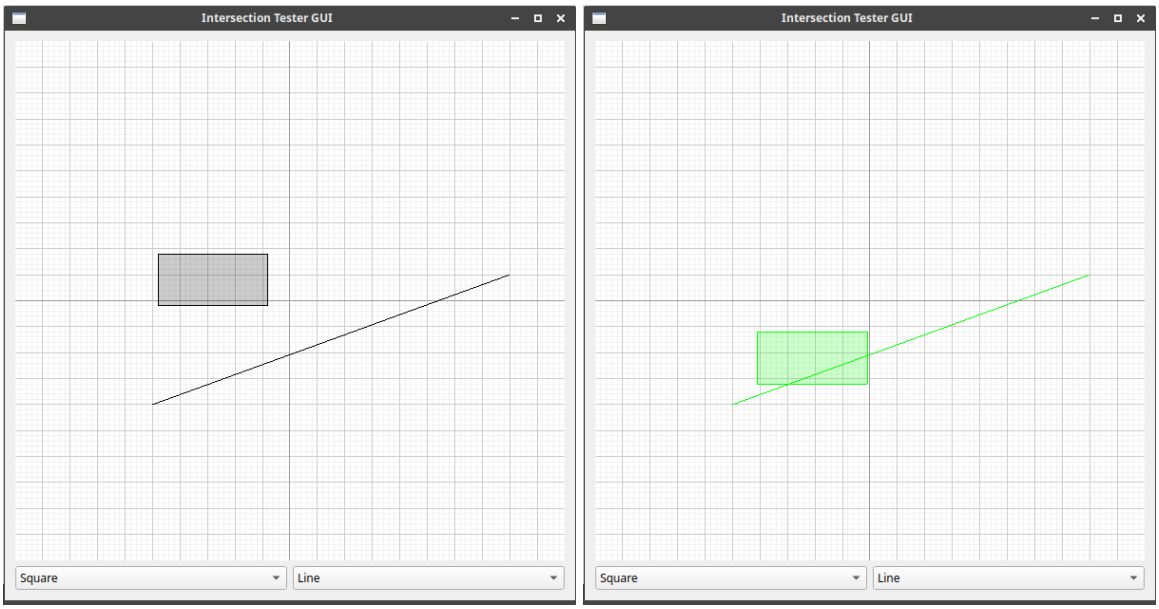
Other than the triangle, which is added for completeness, all of these tests are necessary for the software project. Points are used to present vertices, as well as the mouse pointer, Line Segments are used to test precise boundaries for shared boundary testing, circles are used to test likely collision with glyphs, and axis-aligned bounding boxes are used to test possible collisions with boxes.

6.3.1 Motivation

Generic shape intersection testing is a major component within the algorithm presented in this thesis. We present a table depicting meta data for different tested shape files, and the number of intersection tests we gathered for some of our tests. See Table **??**. For this table, we present a number of examples that are used in Chapters **??**, **??** & **??**. Polygon Count records the AABB-to-AABB tests, whilst the vertex count records the Point-to-Line tests.

Meta Data			**Intersection Tests**		
ShapeFile Name	Polygon Count	Vertex Count	Point-to-Line (% of tests)	ABBB-to-AABB (% of tests)	Total Tests
CCGs	541	2,890,934	5,208,849 (99%)	52,284 (1%)	5,261,133
Ahupuaa Boundaries of Hawaii	725	284,252	547,303 (70%)	239,024 (30%)	786,327
LSOA of Wales	1,949	2,417,391	4,648,604 (80%)	1,175,523 (20%)	5,824,127
US Counties	3,134	51,976	126,841 (6%)	2,007,756 (94%)	2,134,597
Republic of Ireland Electoral Divisions	4,407	419,368	935,468 (16%)	4,742,824 (84%)	5,678,292
Germany	5,385	1,456,469	6,287,363 (38%)	10,093,710 (62%)	16,381,073
Italy	8,946	966,206	1,463,588 (4%)	31,817,661 (96%)	33,281,249
Brazil	11,181	1,288,007	6,124,539 (11%)	47,995,943 (89%)	54,120,482

Table 6.1: A table recording some results for intersecting testing. We record the two major tests from the examples in Chapter **??** as well as a few more represented in Chapters **??** and **??**.



(a) No Intersection **(b)** Intersection

Figure 6.10: Sample image representing the intersection tester GUI. (a) refers to an example of no intersection, while (b) presents an example of an intersection. When an intersection occurs, the primitives change to green.

6.3.2 Graphical User Interface

In order to present our library header, the files includes a graphical user interface for testing purposes using Qt [**?**]. The user can select any of the primitives to move using their mouse with a dropdown box and create a static primitive to test against using a second dropdown box. See Figure **??**.

6.3.3 Using Primatives

The library provides classes for each primitive with basic functionality. The idea behind this library is that testing for intersections is simple. Objects are created or cast into our own primitive objects, and the function *isIntersecting(Object a, Object b)*, returns a boolean value. Although this library is used within our project, we also provide a graphical user interface using Qt to test the library. This enables us to test each primitive against any other primitive conveniently.

```
int HelloWorld::main( )
{
    Point p( 0.0f, 0.0f );
    LineSegment ls( Point( 10.0f,-10.0f ), Point( -10.0f, 10.0f );

    if( IntersectionTester::isIntersecting( p, ls ) )
        std::cout << "Intersect at" <<
            IntersectionTester::closestPoint(ls,p).toString();

    return 0;
}
```

We also give the user access to more information for their own testing. For example, we use a method to test the closest point on a line, to a second point. Rather this allows the user to find that point and make other tests. For example, in our Graphical User Interface, we depict the closestPoint while testing some primitives.

```
Point IntersectTester::closestPoint(LineSegment ls, Point c)
{
    Point p1 = ls.getStart();
    Point p2 = ls.getEnd();
    Point p3 = c;

    float distX = (p1.getX() - p2.getX());
    float distY = (p1.getY() - p2.getY());

    float len = sqrt( ( distX * distX ) + ( distY * distY ) );
    float dot = ( dotProduct( p3, p1, p2 ) ) / ( len * len );

    float closestX = p1.getX() + ( dot * fabs( p2.getX() - p1.getX() ) );
    float closestY = p1.getY() + ( dot * fabs( p2.getY() - p1.getY() ) );
    return Point( closestX, closestY );
}
```

6.3.4 Open Source

In order to maximize the potential of this library, we upload a public copy to GitHub. The library can be found with the two main headers, the GUI testing project, test cases and full Doxygen documentation [**?**]. We plan to update this in the future with any performance optimzations, and additional shapes that may be necessary. We also hope to include 3D functionality in the future as well as other shapes. As well as some extra return procedures for the point(s)

of intersection, or points clamped outside of the primatives. For more work on this topic, we recommend work by Ericson [**?**].

6.4 The Complexities of Identifying a Boundary

Correctly identifying boundaries was one of the biggest challenges througout the course of the PhD. There are many considerations that are necessary to create a robust procedure which can be used. In order to explain some of these challenges, this section covers some of our previous attempts and the road blocks that we encountered.

For our first attempt, we implemented a simple vertex matching algorithm. For a merge pair, we test each vertex against every other vertex for overlap. Common vertices would be saved to a seperate vertex list and this was used to remove those vertices from both areas to identify a shared boundary. There were two shortcomings with this approach.

6.4.1 Unification of Area Pair

The first challenge is identification of a boundary seemed simple, but unifying the two areas along this boundary is more difficult. In order to unify the merge pair along with their non-shared boundary, it is necessary to identify where in the vertex list the boundaries originates. In order to reconcile this, we made some changes to the original implementation. First, we saved the identity of the beginning and ending points of a shared boundary. This enables us to recognize where and when we need to unify the two areas. The second change we make is to control vertex direction. This enables us to increase identification of unification order as follows: $P_aOrigin- > P_aStart- > P_bEnd- > P_bStart- > P_aEnd- > P_aOrigin$. Refer to Figure **??**.

The challenge with this assumption is with the origin. At this point, we assume that the origin is not already on the shared boundary. This is not always the case. A shared boundary can span an arbitrary number of vertices, and can also be positioned at any point within the vertex list. This means that a start vertex can reside within the shared boundary. In order to rectify this situation, we order the vertex list of the merge pair in order to pre-emptively sort the vertex list. When we diagnose the end of a shared boundary, we make this the beginning of the non-shared boundary. Refer to Figure **??**.

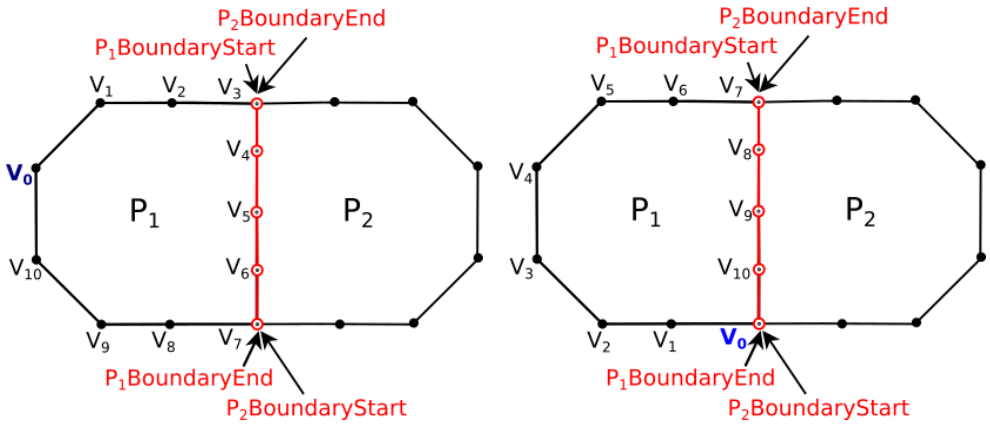


Figure 6.11: (left) Our first attempt to unify areas. (right) The improved process to unify areas. We re-order the vertex list to start at the end of a shared boundary.

6.4.2 Identifying the Start and End of a Shared Boundary

Now that we understand why finding the start and end of a shared boundary is critical, we discuss the challenges for finding these start and end points. Attempting to use our previous procedure to identify common vertices is not appropriate, although we did not fully understand why up to this point. This was due to voids and fissures in the non-shared boundary. Because of the frequency of these configurations, they gained an accumalative effect on further identification processes which caused the early versions of the algorithm to crash. At this point, we had not implemented debug visualization to inform the cause of the errors. However, we did identify what we considered as rendering errors, points along the shared boundary that were still rendered (which were later identified as voids and fissures, causing error at later cycles).

In order to rectify these errors, we modified our extraction algorithm to include a bi-directional search. From the origin, we search both clockwise and anti-clockwise. In this case, we can assume that the route can fall into two cases. First the origin resides on a non-shared boundary. If this is the case (Case A, Figure **??**), we search bi-directionally if the clockwise search encounters a common vertex, we can use that as the start of the shared boundary whilst the first common vertex in the anticlockwise search will be considered the end of the shared boundary. The second case (Case B, Figure **??**) is if we start on a shared boundary. In this case. our configuration can be flipped, where the last common vertex before a non-common vertex in the anti-clockwise direction is the start of the shared boundary, and the last common vertex before a non-common vertex in the clockwise position is the end of the shared boundary. However, this approach also falls apart when we add new exceptions. Refer to Figure **??** for a visual representation.

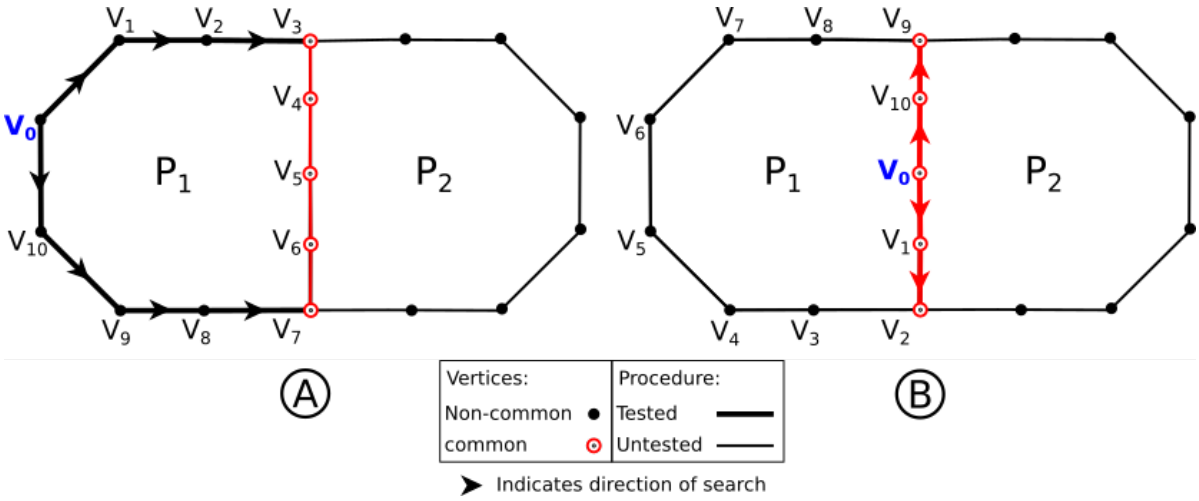


Figure 6.12: The second version of our boundary identifier using a bi-directional search. We identify two cases, Case A refers to the origin (V_0) residing on a non-shared boundary whilst Case B refers to the origin (V_0) residing on a shared boundary. The arrows refer to the direction tested and length of the clockwise or anti-clockwise search.

The first condition we found was a T-junction. In order to verify a non-common vertex is not part of a shared boundary, we now have to test if this point falls on a line segment. This increases the complexity of common vertex testing, but the algorithm handles this exception well. The second exception is more of a challenge. We now detect voids which we describe as an extra point on a shared boundary that does not lie on shared boundary line. This complicates Case B, as now we cannot consider the first non-common vertex to accurately depict the start or end of a boundary. For this, we added void tests, to see if the shared boundary contiuned. The other challenge with a void is that it now has potential to falsely identify the Case B as case A. If the start vertex falls on a void, the algorithm would assume that we now sit within Case A (Case A of figure **??**) which presents a false postive boundary selection.

This becomes more complex when you consider fissures which feature an arbitrary number of linked voids, allowing for more oppurtunities for this exception to present itself, and nullifies any void testing which could be used to depict it. Finally we also find examples of voids and fissures with multiple instances within one shared boundary to complicate this. Refer to Figure **??** for a visual representation.

In Chapter **??**, we discuss the fact that over 50% of one of our test cases contain voids, T-junctions, or fissures, when identifying a shared boundary. This causes a major concern when identifying shared boundaries. We could have cleaned the shapefiles of voids and fissures, however the shape files we work with are considered industry standard. It would be

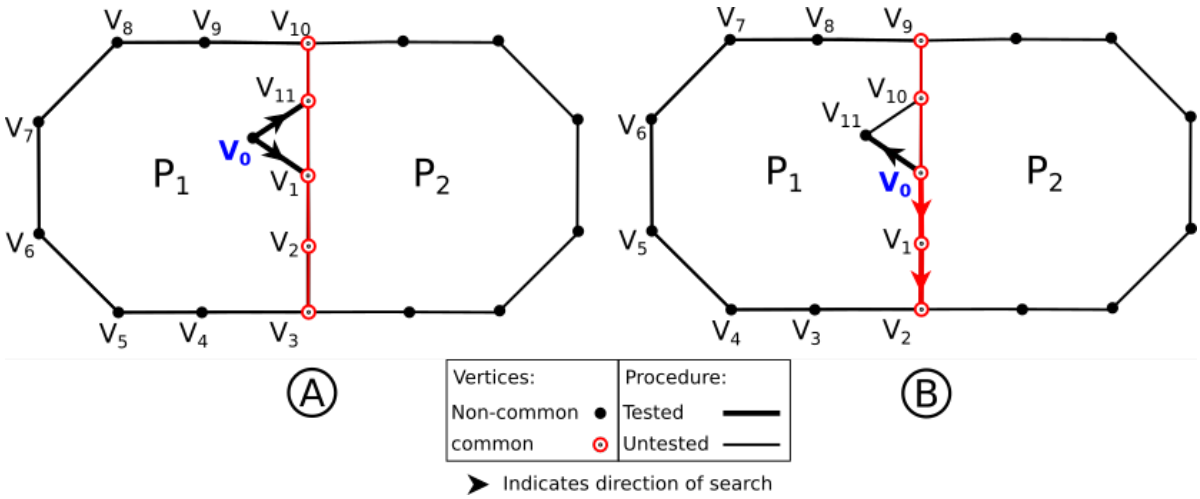


Figure 6.13: Errors that occur when voids are considered. We run the same cases (A and B) from Figure **??** and implement voids which causes two distinct errors. As voids can be replaced with fissures, and can be found multiple times. This represents only the simplest cases for error. Case A begins on a void, while Case B holds a void on the shared boundary.

disingenuous to believe that our algorithm is useful when we hold the only shapefiles that the algorithm worked with.

In order to rectify this problem, we added two hueristics, and develop a more advanced method. First, we assumed that boundaries between two areas are not split, and are one contiguous boundary. Split shared boundaries between areas are already rare, however, we made this assumption due to our focus on scale. Our algorithm's distance metric is biased towards merging the smallest area with it's smallest neighbour. Even amongst cases where a split boundary could occur, it would be highly irregular for a boundary that caused a split boundary to not be considered the first area to be merged. Our second assumption is that a shared boundary has a shorter length than a non-shared boundary. It is more likely for this to occur, however just like with our first assumption, we mitigate the chances of our assumption being false by considering the smallest neighbour above others. See Figure **??**.

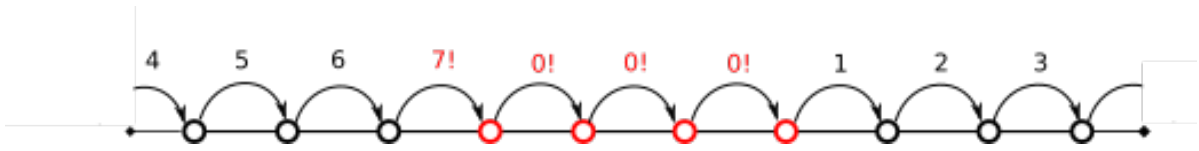


Figure 6.14: Depiction of topological distance test of non-shared boundaries for a non-void vertex list. We search the list as loops until all vertexes have been tested.

Chapter 7

Conclusion

“Overview first, zoom and filter, then details-on-demand”

— Ben Schneiderman, 1996

Contents

7.1 What are you hiding from me? On data resolution, expectations and fears, and realistic counter-measures in visualization.

7.1.1 Introduction and Motivation

With big data rapidly becoming ubiquitous within the field, many have been wondering what they must do to represent all the data given to them. At the EuroVis conference 2017, in his capstone presentation, Helwig Hauser brings up the idea of human vision bandwidth and discusses the brain's capability to interpret information. Hauser states that color and transparency are only temporary solutions and may not be useful when our data sets are 10 orders of magnitude larger than current data sets [**?**]. We consider this while looking at Schneiderman's information seeking mantra, *" Overview first, zoom and filter, then details-on-demand"*, specifically the first step, gaining an overview of the entire collection [**?**]. If we can no longer rely on color, how can we enable a full overview in one image? The answer may be simpler than expected, and that is to reduce the resolution of data to present.

Wong has already begun work on the topic of multi-resolution data [**?**], however his work emphasizes scientific visualization. We suggest *low resolution* visualization. Low resolution visualization describes a representation of a visualization with a reduced level of detail. We believe that low resolution can be used to present an overview of the data while (a) still presenting an authentic view of the underlying data (b) conveying close approximations of the

underlying data, and (c) still allowing the user to clearly understand where and how zooming and filtering will be useful. However, implementing a low-resolution view can be interpreted negatively. By creating approximations of data, value error may be introduced, and when values are associated with error the observer may be subject to false insight and false results. This issue is a well-known problem even outside of the visualization landscape. The Modifiable Areal Unit Problem (MAUP) considers how aggregating values confined to a 2D space can vastly change the results depending on which areas are grouped [**?**]. We also consider that many people assume that something unseen, has been left unseen intentionally.

We believe that low resolution is a viable approach that can improve any visualization tool and new techniques can be used such as clutter reduction for visualization that would otherwise find difficulty with common reduction techniques. We explore why low-resolution visualization should not be considered a barrier, and preferably a useful tool to avoid cognitive overload. We also discuss common negative sentiment and ways to address challenges that may be coupled with the idea.

7.1.2 Hidden Data

It is important to consider that there are many cases where data you visualize has already been transformed to reduce complexity. If we look at census data, by law, before any data is made public, the data needs to be both anonymized annd grouped in order to protect the identity of persons both in name and location.

7.1.3 Visual Perception

Perception is a large research area within visualization and is a leading consideration when designing software and visualization techniques. This large amount of work can be broken down into discrete categories such as color [**?**, **?**, **?**], size [**?**, **?**, **?**], and motion [**?**, **?**, **?**]. However, research on this topic can vary widely to consider the work comparing ease of perception against other techniques in the form of user studies [**?**] or guidelines on perceivability within specific topics [**?**]. When considering a visualization technqiue, recognizing perceivabilty as a key challenge can transform your software into a critical analysis tool.

Visual perception is more than just an understanding of digital phenomenom. Physical constraints can also play an important role on how techniques evolve. Pixel density is an important consideration when depicting data. In Chapter **??**, we depict that when an area becomes smaller than 10 pixels, error becomes prevalent when identifying color, and perfomance time also increases. This has lead to many advances in space-filling algorithms,

however what happens when a space-filling algorithm is forced to present some data marks within a small pixel threshold.

7.1.4 Overview without full detail

Let's reconsider the information seeking mantra once more. An overview is considered the first step to understanding, and because of this, it is easy to consider this as everything that needs to be represented. However, this is not the case. If everything could be understood from an overview, there would be no need for a user to zoom in or filter in the first place. If we compare the overview of a visualization to a modern shop, we must consider the overview as our initial view of a shop. If we conclude the outside of the shop is the overview, then our main point of emphasis is that which is displayed in the at the entrance or on display in the window. If we consider the interior, we are likely to see signs which lead us to where we would like to go. An overview should be considered a gateway to enable zooming and filtering.

In order to make sure that reducing the resolution does not affect exploration, we can use techniques to minimize the effects of aggregation. In this section, we discuss two key areas: resolution indication and uncertainty visualization.

Geometric Channels	**Optical Channels**	**Topological and Relational Channels**	**Semantic Channels**
• size / length / width / depth / area / volume • orientation / slope • angle • shape • curvature • smoothness	• intensity / brightness • colour / hue / saturation • opacity / transparency • texture (partly geometric) • line styles (partly geometric) • focus / blur / fading • shading and lighting effects • shadow • depth (implicit / explicit cues) • implicit motion / motion blur • explicit motion / animation / flicker	• spatial location • connection • node / internal node / terminator • intersection / overlap • depth ordering / partial occlusion • closure / containment • distance / density	• number • text • symbol / ideogram • sign / icon / logo / glyph / pictogram • isotype

Table 7.1: Visual channels presented by Borgo et al. [**?**]. Refer to Section **??**

Resolution Indication

If area resolution varies across 2D space, it is important to visualize where resolution may lie. We can use visual design variables in order to depict spatial frequency of data. Borgo et al. combine multiple visual encoding variables to present a table of visual channels including geometric, optical, topological and relational, and semantic channels [**?**]. Refer to Table **??**. In Chapter **??**, we present low resolution indicators by fixing the indicator to outline thickness, size or shadow. In this instance, hidden indicators are mapped to administrative area density. This makes the tool less useful for those already aware of the geospatial structure, however, this could be intrinsic for new users.

Uncertainty Visualization

Uncertainty visualization is a growing aspect of information visualization and applying these techniques to low resolution data is a logical process. Common uncertainty visualization normally uses visual mappings to convey error such as fuzzyness to detract careful analysis. However, when approaching uncertainty in low-resolution visualization, the idea sits in contrast with our attempt to coerce further analysis, causing fuzzyness to be a counter intuitive mapping technique to use. We propose mitigating uncertainty by applying an advanced structure approach to your visual design of choice. For example, cateorical data could be mapped to present the majority, however a second axis (saturation or opacity for example) can be used to present the percentage of the majority hold. We could increase this by modelling a tri-variate color map that depicts the weighting of objects between the three values, presented in Figure **??**. There is also work on this technique for higher dimensional data by Cheng et a [**?**].

7.1.5 When to Consider Low Resolution Overviews

In order to use low resolution overviews, we need to be able to identify when they are appropriate. We use Figure **??** as our first example. **Occlusion** is likely the easiest criteria to spot. Our example plots glyphs to Clinical Commisioning Groups representing healthcare data. It is possible to reduce occlusion with other methods. For example, Dwyer et al. present the Fast Node Overlap Removal (FNOR) algorithm to avoid this [**?**]. However, this may not always be useful. In this case, we want to ensure that the glyphs are easily coupled to the geospatial context whilst the FNOR algortihm would remove this entirely in densely populated areas. In this case, areas are amalgamated to make both the glyphs and the underlying values perceivable.

Our second criteria is **data density**. Even if there is no occlusion, it is still possible for it to become difficult to perceive information. Space-filling techniques such as treemaps can avoid occlusion, but when you have to map millions of events, then the output can result in simple pixel representations of the data. Fekete and Plaisant present a paper working on a treemap for a million items represented [**?**]. We can see Figure **??** makes it very difficult to see the granular data, which could hide some interesting information.

7.1.6 Conclusion

We discuss the concept of reducing data resolution to create low resolution visualization. We look at perception as well as the idea of hiding or anonymising data, and some considerations that need to be taken into account when creating a low resolutions visualization.

7.2 Future Work

During the journey of the thesis, there were many avenues that the thesis could have followed. As such, we dedicate this section to discussing some of these ideas, as well as potential directions a subsequent candidate could follow during their PhD.

With regards to our literature review, We feel we have opened the doors to a range of potential survey papers under the SoS, or meta-survey branch. During the process of the thesis, other related surveys have been published [**?**, **?**, **?**]. However, other avenues could be presented such as a Survey of Surveys for Geospatial, Scientific or Computer Graphics.

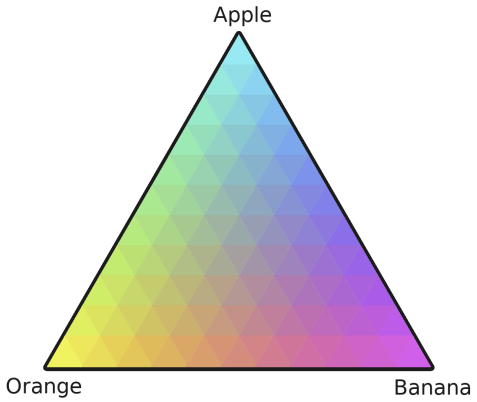


Figure 7.1: A depiction of a trivariate color map. The color map clearly expresses the distribution of values. By using this with low resolution visualization, we could clearly depict what each area holds at multiple levels of detail. Courtesy of McConchie [**?**]

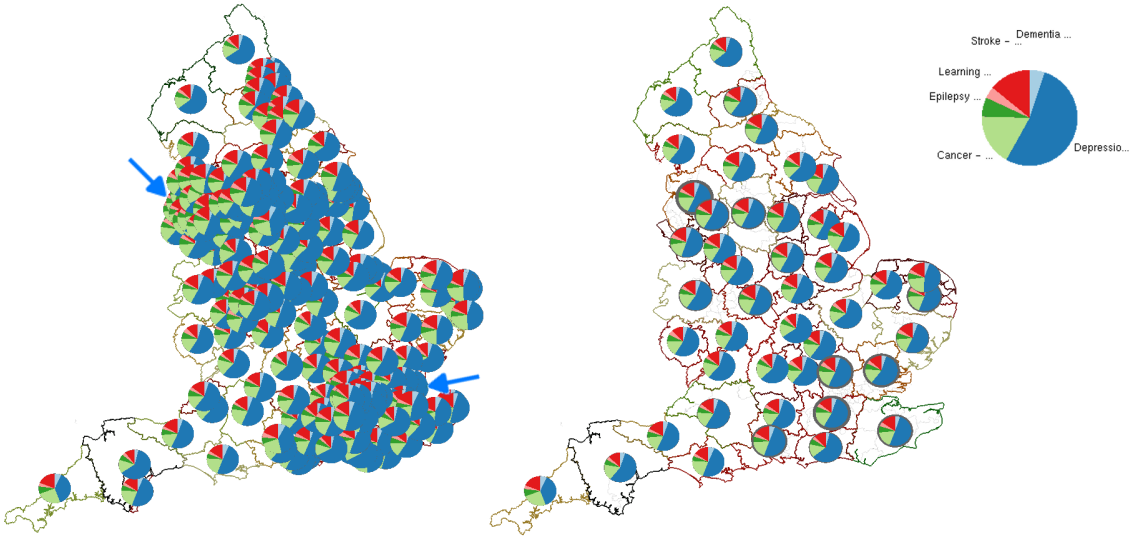


Figure 7.2: (left) The representation of population health data based on the Clinical Commisioning Groups (CCGs) of England [**?**]. Glyphs that are simply placed at the centroid of each region are over-plotted and occluded around London and Liverpool (indicated by blue arrows). (right) The result using level-of-detail scale-aware maps. Even at a small scale for the figure, we can still clearly differentiate each area's glyph.

There are many avenues for future work for the merging algorithm presented in the thesis. Although we use real unit-areas, we would like to test with a broader range of choropleth data. The algorithm still has performance optimizations which could accelerate the speed even further, such as schematization [**?**] which could be used to enable better optimization with the drawback of topological continuity being reduced. Other existing formats such as TopoJSON [**?**] look at reducing geometry redundancy and could be an excellent subsequent format for the procedure. We worked with 2D coordinate-spaces. A 3D coordinate space would be an exciting direction to take the algorithm and could open new applications for the process. The current termination method revolves around the idea of one area per contiguous region. Updates in the procedure could allow for more user control when it comes to stopping the merge procedure such as for categorical data where the most abstraction is introduced. Alternatively to this, a bi-variate color map could be implemented to display more accurate concordance of underlying values. The algorithm potentially can apply to any data-sets with geometric boundaries and is open to new data structures.

There are some limitations to the study we review in Chapter **??**. We found boundary design to be a more significant factor for small area sizes than anticipated. In the pilot study, we found that users had a much better accuracy overall, and comments were made about the area's color changing. This was actually due to the boundary's framing of the pixel, enhancing the perception of color based on the surrounding black contour. We try to control this as

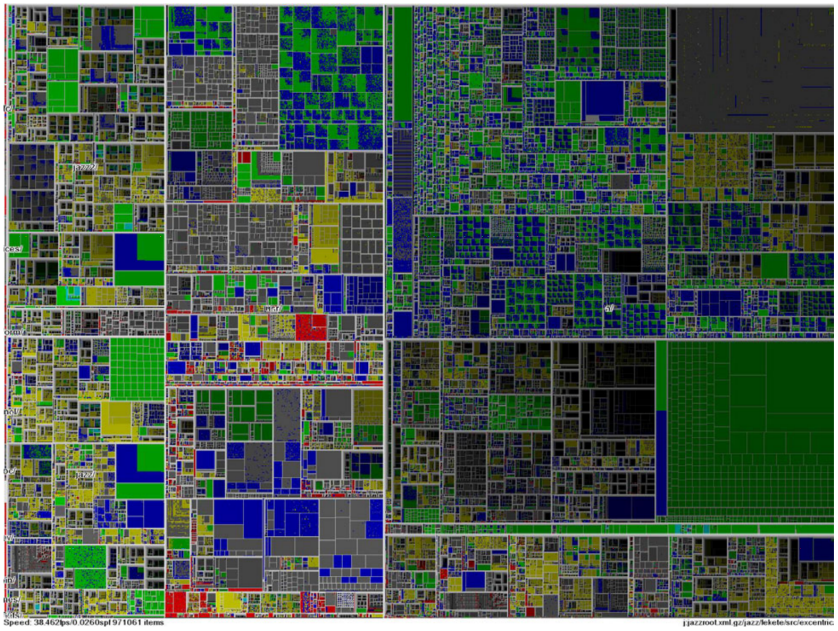


Figure 7.3: A treemap depicting a file system of over a million items. Color represents the file type. Image courtesy of Fekete and Plaisant [**?**].

carefully as possible, however, this could be examined further in a follow-up study. We use real-world maps and areas. This means we do not control the aspect ratio of the areas, which may lead to more error in some situations such as narrow areas. This is an aspect that can be explored in an alternative study. T1 and T2 could point to a large amount of future research. For example, since we hold data about the location of the presented area and the color seeding, we could look back at what could have influenced the color perception and decision for the area.

For our multivariate implementation in Chapter **??**. At the moment, we use the raw derived centroid as a placement strategy. Although this removes a lot of density and occlusion, there is still some wasted space. We believe that by adding some overlap removal, we could use space more efficiently, while still avoiding any decoupling problems. Although we present some case studies, the algorithm could be more carefully compared to other glyph placement strategies with a user-study evaluation. We think the use of transitions is a great tool for understanding variation, but it is not always necessary. We feel that there are many avenues for exploring at multiple levels of detail. For example, directly zooming to a glyphs unit area extents may not need to represent zooming, to speed up exploration.

Now that we can improve performance on a second pass-through by saving build instructions to reduce the calculation of neighbor and boundaries, the next step would be to pipe multiple areas to be merged. This would allow the user to constrain area representation to

the desired representation further. For example, if we look at the UK, OA's would only merge within their LSOA, and LSOA's would only merge within their MSOA's.

At the Vis 2015 conference, Ribarksy states "When you add dimensions to a problem, all of a sudden it becomes unsolved" [**?**]. Although we extend our technique to take advantage of multivariate data. There are still opportunities to increase the dimensionality by asking for both statistical and temporal data at once.

In order to move full circle (refer to Chapter **??**), we must consider scalability as our future work. Although throughout the thesis, we made large leaps in both the scale and performance, we still met some challenges in the area. During our initial testing of the algorithm, we attempted to run the algorithmn on some large datasets, including the output areas of the Wales and England. This dataset held over 180,000 polygons with high complexity vertex lists. Although our algorithm did not fail, the perfomance time became such an issue (days) that we had to terminate before we could see any result. If there was more time, I believe that search for ways to optimize the algorithm in order to make this dataset achievable would be a good next step. Some options for this would be more focus on GPU implementation, multi-threaded processing, and a way to hold contiguous stuctures seperately from the build hierarchy to avoid unecessary recalculation.

Appendix A

How To Write A Visualization Survey Paper: A Starting Point

[?]

Abstract

This paper attempts to explain the mechanics of writing a survey paper in data visualization or visual analytics. It serves as a useful starting point for those who have never written a survey paper or have very little experience. A literature review or survey paper is often considered the starting point of a PhD candidate's scientific degree. However, there are no dedicated papers that focus on guidelines for the planning or writing of a survey paper or literature review in visualization or visual analytics. We provide guidelines and our recommendations for a foundational structure on which to build a survey paper, whilst also considering intermediate goals, and offer helpful advice to improve the survey process and literature analysis. The result is a useful starting point for those wishing to write a survey paper or state-of-the-art (STAR) review in visualization or visual analytics. The guidelines and recommendations we make can also be generalized to other areas of computing and science.

An abstract is a required feature of a survey paper and should identity the topic of the literature review. A good abstract addresses why the given topic is interesting and why it is helpful. A good abstract features the following elements: (1) topic introduction, (2) the motivation, (3) the goal of the review, and the benefits the review provides to the reader. A good literature survey offers a helpful classification of the literature, mature areas of research, and open, unsolved problems in visualization or visual analytics.

A.1 Introduction and Motivation

A survey paper is an incredibly useful tool for both newcomers and experts of a given field. Twenty years ago, Jim Blinn stated *"There's lots of other journals and it takes more and more effort to make sure that you know what's happening."* [**?**] One of a survey paper's primary goals is to assist the reader in the hunt for previously published research papers on a given topic.We discuss a general approach to planning and writing a typical survey paper section-by-section, and provide more details and guidelines as to appropriate content for each section.

The target audience of this paper is a PhD student in their first year and their supervisor. This paper presents and follows a versatile template which the reader can follow to accelerate and facilitate the creation of a survey paper. In addition, we discuss important considerations in the preparation phase of a survey paper. For this, we use the symbol (◆) to designate aspects that are examined as part of the preparation, but are not necessarily discussed in the actual text itself. The guidelines and recommendations are based on our experience of reading and writing survey papers in visualization [**?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**] as well as meta-surveys, or surveys of surveys (SoS) [**?**, **?**, **?**, **?**].

A well written introduction motivates the topic, including applications of the topic, why the research direction is important, and what the contributions of the survey or state-of-the-art (STAR) report are. Here we use the term "STAR" to refer to literature reviews with a special focus on recent literature, e.g, within the last 10 years. Survey papers are more comprehensive and may include literature from all years.

The introduction and motivation describe some of the big research challenges that are faced by the topic covered by the survey. Some generic aspects that can be considered common challenges are: the rapidly increasing size and complexity of the data, heterogeneous data sources, uncertainty, challenges associated with equipment such as cost or speed, cognition and perception, understanding or representation of observations, the limitations of existing software, and hardware or software performance limitations.

A.1.1 Contributions

The contributions are clearly presented in either the introduction (and motivation) section or a subsection. A reader can recognize the importance and novelty of any paper by the end of the introduction and the insight and benefits that can be gained from reading it. We recommend authors strive for approximately three contributions. These contributions are described in conjunction with the rest of the first section, to make it clear how the survey paper fits into the visualization field's landscape. Examples of a typical contribution include:

- A novel classification of the literature (how your classification differs from previous surveys, or whether the survey is the first of it's kind in the field).
- A compilation of future challenges or trends in the domain.
- The identification of both mature and less explored research directions in the field.

A good review paper considers key questions in the field. What has been published so far? Are there any controversies, debates or contradictions that should be brought to light? Which methodologies have researchers used, and which appear to be best? Who are the leading experts in the field? And how the topic fits into the landscape of visualization. By analyzing questions like these, your survey presents some clear contributions to discuss.

The contributions of this paper include:

1. The first guidelines (to our knowledge) on how to write a survey paper in data visualization or visual analytics.
2. Guidelines on the process of preparing a literature survey.
3. A structured survey paper template that can be followed, with in-depth guidelines describing the content of each section.

We believe a high quality, full survey paper can take approximately a full year (part-time) to incrementally prepare and write including the literature search. A significant portion of this time concentrates on gathering the related literature on a given literature review topic. Due to the length of full survey papers (20-30 pages), it is time-consuming and difficult to undertake multiple internal full paper reviews and revisions, therefore it is helpful to distribute the preparation, discussion and intermediate feedback sessions periodically over the preparation time frame, to reduce the drafting and corrections process in the final stages. A tested strategy can separate the individual paper browsing and summarization process from the main survey paper organization [**?**]. Individual research paper summaries can be written on a weekly basis for the first six months, yielding roughly 24 summarized topic papers before any final decisions have been made on the organization, or literature classification. This provides a good basis for potential paper classifications to develop.

A.1.2 Challenges of Writing a Survey

We identify seven main challenges associated with writing a survey paper.

1. Managing the amount of previously published literature (discussed throughout this paper)

2. Identifying a starting point (the purpose of this paper)
3. Deciding on a topic (see Scope, Section **??**)
4. Performing a search (see Search Methodology, Section **??**)
5. Interpreting individual research papers (see Section **??**)
6. Deriving a classification of literature on the given topic (see Section **??**)
7. Determining related unsolved problems and future challenges (see Section **??**)

In the following sections, we address some of these central challenges.

A.1.3 Literature Search Methodology ◆

It is important to clearly describe how you search for the papers cited in the survey. When a reader browses the literature review, it is likely that you have found research papers that they may not have seen. A new PhD student usually has not yet discovered all of the relevant conferences and journals to search. The literature search methodology provides the names of digital libraries, search engines, search terms, and literature sources used to find literature in your survey paper. If we are looking for research papers on the topic of treemaps, we can use the Google Scholar search engine for example [**?**] to search the term "treemap". This gives us (at the time of writing) over 16,000 related search results. By doing the same using the IEEE Xplore Digital Library [**?**], we get 115 items, and using vispubdata [**?**], we get 58 items. For visualization purposes, the three previously-mentioned search engines are a great tool. Combined with the use of Google Scholar's "Cited by..." option to find related work, you should be able to gather a fairly complete set of papers. A complete list of sources to search is provided in Table **??**.

The other search consideration is a manual search. When you have found one matching paper, it is likely that you will find a number of related research papers in the related work of the given match. This can be especially useful if there are related survey papers. If you find the majority of papers this way, providing a breakdown of conferences and journals may be a beneficial method of presenting your literature search. The goal is to provide enough information to make your literature search thorough and reproducible.

Literature Sources
Google Scholar [**?**]
IEEE Xplore Digital Library [**?**]
ACM Digital Library
Vispubdata [**?**]
The Annual EuroVis Conference
IEEE TVCG Journal
IEEE Pacific Visualization Symposium
IEEE VAST Conference
The Annual Eurographics Conference
The Eurographics Digital Library
Journal of Visual Languages & Computing
Information Visualization Journal
Computer Graphics Forum
Computer & Graphics
ACM Computing Surveys

Table A.1: A shortlist of literature sources.

A.1.4 Literature Classification Overview

Organizing the research papers in your survey is a central topic in any literature review. Classification dimensions can be derived for the task. Each of your classification dimensions is presented in this section. One dimension of a literature classification is often a list of (subjects or) topics. Describe each topic in the classification, how each research paper is classified, and possibly some exceptions where research fits into multiple subject categories. A high-quality literature classification is often composed of more than one dimension, e.g. subject category and data dimensionality. By talking about dimensions separately, you allow the reader to understand what is presented in the classification before discussing the individual topics and papers. Images or tables are a good way of conveying an overview.

Refer to McNabb and Laramee's Survey of Surveys for an example [**?**]. One axis consists of an adapted Information Visualization pipeline. A second dimension is a set of topic clusters. A section is presented to discuss how their pipeline differs from Card et al.'s original [**?**], and why modifications have been made. The topic clusters are discussed in a separate section. How the subjects were gathered and selected is explained. Both sections provide a visual representation to aid in the understanding of each main axis.

It is important to clearly explain each axis of your proposed literature classification as this is one of the most important aspects of a survey paper which can clearly impact whether a

survey paper makes it through the review process. Please review Section **??** for a detailed guide to developing a classification.

A.1.5 Survey Scope ◆

Defining the scope of a survey – subject categories or the topics it covers (and does not cover) can be a very challenging aspect of writing any literature review. The scope defines the topic boundaries of literature that are included or excluded from the survey. If the scope is too broad, then the survey includes too many research papers to manage. If the scope is too narrow, then not enough literature may be included for a good classification from which deductions may be drawn. Therefore, it is important to accurately designate what does and does not meet the scope of your paper. The scope is flexible as long as you clarify the scope boundaries clearly. For example, if you paper reviews a specific topic of interest (i.e. a technique or application), then it makes sense to explicitly state that this is the case.

Aim to create a scope that encompasses roughly 40-50 research papers. If you cannot see any avenues of narrowing your scope, year of publication is always an option and can be used as a soft-cap, for example, Alsallakh et al. with their state-of-the-art in sets and set-typed data [**?**]. Limiting your survey to the most recent 10 years also turns it into a STAR. Doing this can be a way of reducing the focus research papers while still including older papers for other types of analysis if necessary. Provide examples when pointing out what is and isn't in scope. Lipsa et al. provide an informative example of a scope [**?**]. The survey clearly presents papers that do or do not meet the scope criteria, with given examples of papers.

A very common problem is defining a scope that is too broad. We recommend a scope that includes approximately 50 research papers per PhD student and supervisor pair on the author list. You can use Google Scholar and the other search engines described in Section **??** to make early assessments of scope. For example, if you enter "Isosurface" into a search engine for research papers, you will get a broad scope including over 100 research papers. The scope could be narrowed by focusing on isosurfaces for time-dependent data.

Survey Type ◆: It is important to consider the type of survey that you would like to present within your scope. A *'Literature Review'* refers to a more comprehensive list of papers on a given research direction whilst a *'State-of-the-Art'* (STAR) report refers to the more recent research or techniques. Some examples of this include Beck et al. with their state-of-the-art in visualizing dynamic graphs [**?**] and Laramee et al. with the state-of-the-art in flow visualization [**?**]. On top of this surveys can focus on task taxonomies rather than the field itself. Examples of this include Kerracher et al. and their task taxonomy for temporal graph

visualization [**?**], or Schulz et al. with their design space of visualization tasks [**?**]. These are important considerations to discuss with any potential co-author.

A.1.6 Survey Paper Organization

The organization of the survey refers to the order in which research papers and subject topics are presented in the main body of the literature review. Organization of a survey can be trivial when the classification has been developed. Research topics and papers can be discussed in linear order based on the primary dimension in your classification, where the secondary dimension will represent the sub-sections of your survey. In each section, a sub-organization can be applied either based on another classification or publication year. Tong et al. present their classification dimensions and follow chronological order when presenting their summarized research papers in each sub-section [**?**].

A.2 Related Work

There are a number of other related educational papers readers may find very useful. Sastry and Mohammed develop a tool named a summary-comparison matrix (SCM) to aid in the organization and extraction of information in research papers [**?**]. Our paper differs to this by contextualizing the entire survey writing process. Laramee presents a starting point on how to write an individual visualization research paper [**?**]. The paper covers each aspect of the individual paper including the introduction, method, implementation, and performance. The paper also covers important considerations such as planning, procedure diagrams, figures and images, and supplementary material. This paper is considered sibling reading and is encouraged for discussions on paper writing, titles, latex, and collaboration. Laramee also presents guidelines on how to read and summarize a visualization research paper [**?**]. We extend this in Section **??** to guide uses in extracting the essentials of a survey paper. Daniel Patel provide an educational paper focused on the requirements of an article-based PhD in visualization [**?**]. Munzner provides an overview of pitfalls authors may fall into, leading to rejection during the reviewing process [**?**]. Although closer to complimentary material, Laramee produces an audio-visual representation of the paper topic. The video gives a clear structure to design your survey paper [**?**]. There are several pieces of software that can aid in the collection and usage of literature reviews including Mendeley, JabRef and Papers [**?**, **?**, **?**, **?**].

A.2.1 Background

Depending on your chosen topic, it may be appropriate to include a background section. A background section reviews the history of a topic such as how a technique originally emerged and it's evolution, or how its use has changed. Nusrat and Kobourov give an excellent example by looking at the evolution of cartograms in the 19th and 20th century [**?**]. Borgo et al present the history of glyphs and their applications [**?**].

A.3 Presenting the Main Literature Survey

Section **??** discusses individual research papers, developing a literature classification, and different types of paper classification.

A.3.1 Summarizing An Individual Research Paper

This sub-section is adapted from the paper by Laramee and focuses on extracting essential information from a single visualization paper [**?**]. This is a helpful exercise during both the preparation and writing phases of a literature review. It also facilitates the development of a literature classification (Section **??**). We extend this to provide guidance when summarizing an individual survey paper based on the experience gathered writing the SoS [**?**]. This is helpful for describing survey papers that may be included in the related work section (Section **??**). When summarizing an individual research paper, the core elements to extract are: the concept, related work, data characteristics, visualization techniques, and application domain.

The focus elements of a survey paper vary in what may be considered the important essentials or aspects. For example, it is unlikely that a survey has a regular set of data characteristics. For a survey we focus on: the concept, the scope, the classification, classification dimensions, and unsolved problems or future work. Here is a sample survey summary of space-time cube operations by Bach et al. [**?**]

Title: *A Review of Temporal Data Visualizations Based on Space-Time Cube Operations* by Bach et al. [**?**]

1. *The Concept:* Bach et al. survey a variety of temporal data visualization techniques and discuss how their operations represented by space-time cubes are used in the context of a volume visualization from the 2D+time model. [**?**]

2. *The Scope:* The paper discusses common space-time cube operations including time-cutting, time flattening, time juxtaposition, space cutting, space flattening, sampling, and 3D rendering.

Bach et al. then present the taxonomy of space-time cube operations they designed before providing the reader a selection of multi-operation systems.

3. *Classification:* The taxonomy (<Figure 24 of original paper>) presents a classification of elementary space-time cube operations such as drilling, cutting and chopping. These are broken down into sub-categories with schematic illustrations in order to enable the user to easily understand what effect the operation has. For example, the flattening section is broken down into planar flattening and non-planar flattening. Planar flattening is sub-divided into orthogonal flattening and oblique flattening.

4. *Figure: <Figure 24 of original paper>*

5. *Classification Dimensions:*

X-Axis: Operations, Time, Space

Y-Axis: Extraction: [Point Extraction, Planar Drilling, Planar Cutting, Non-Planar Cutting, Planar Chopping, Non-Planar Chopping],
Geometry Transformation: [Rigid Transformation, Scaling, Bending, Unfolding],
Content Transformation: [Recoloring, Labeling, Re-positioning, Shading, Filtering, Aggregation],
Flattening, Filling.

- **Supplementary URL:** http://spacetimecubevis.com/
- **Papers:** There are 91 Papers cited in the survey (1970-2013)

6. *Unsolved Problems/Future Research:* There are many open research areas discussed. Some of these include interaction techniques such as focus+context applied to different operations, research into operations for extended data dimensions, and understanding which operation is most appropriate for a given task.

You can see the initial summary follows a template. This is useful for treating the survey papers systematically including collecting meta-data in a consistent and helpful manner. When the papers start to be placed into the survey, the template format can be adjusted into more naturally written text.

A.3.2 Developing a Literature Classification (How To) ◆

Deriving a literature classification is one major challenge of writing a survey. A classification categorizes each research paper such that similar papers fall into the same group. Deriving groups, categories, and dimensions for your classification requires careful thought.

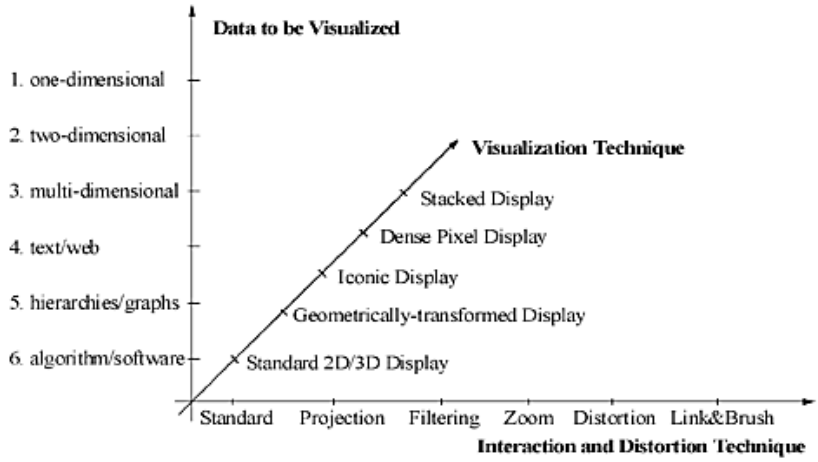


Figure A.1: Keim's Classification of information visualization techniques. Courtesy of Keim [**?**].

One property of a good classification is that it is easy to properly place research papers into categories. If you have great difficulty placing individual research papers into the categories identified in your classification, this may be a sign that it requires adjustment.

A.3.3 Identifying Classification Dimensions ◆

There has been a lot of work invested in generating taxonomies. A good classification dimension e.g. subject-category, data-dimensionality etc, is descriptive and easy to communicate. Your classification may change during survey drafting. If you find it difficult to insert literature into a classification, modification is always an option. We recommend aiming for a 2D classification to begin with. If you are looking for ideas, adapting existing taxonomies or principles can yield useful classification topics. For example, Keim's technique taxonomy [**?**] is used to group visualization techniques into 5 key categories (standard 2D/3D displays, geometrically-transformed displays, iconic displays, dense pixel displays, and stacked displays) (See Figure **??**). This is used by Ko et al. [**?**] in their survey of financial data visualization, where each paper is mapped to these different techniques that are used within the literature. Another option is automatic taxonomy generation. There is a lot of work on text extraction [**?**, **?**] and taxonomy generation [**?**, **?**, **?**, **?**, **?**, **?**].

We recommend looking for natural recurring topic clusters. If you follow the temporal planning guide suggested in Section **??** and extract meaningful meta-data, you may produce some classification candidates by brainstorming. Some other candidate classifications are: subject category, data dimensionality, visualization technique, design dimensionality, field challenge

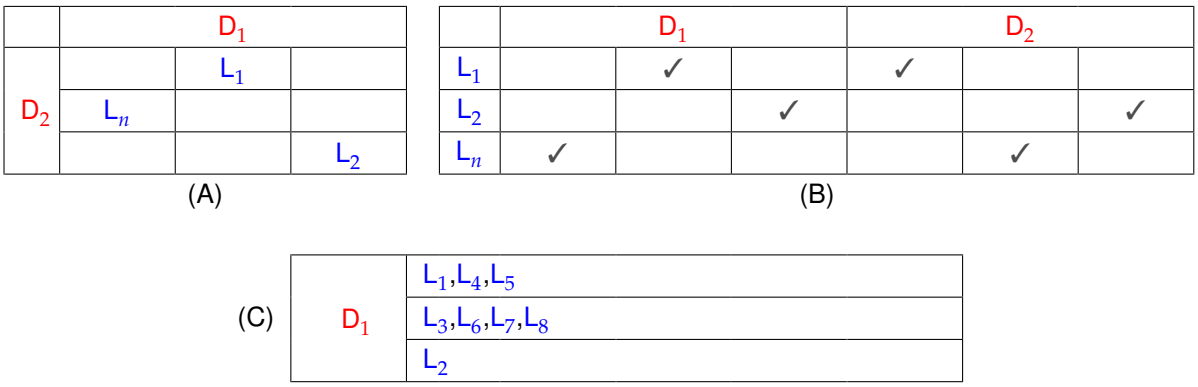


Figure A.2: Examples of classification schemes using unique-mapping. D refers to a classification dimension and L refers to a reviewed item (in most cases, the literature reviewed).

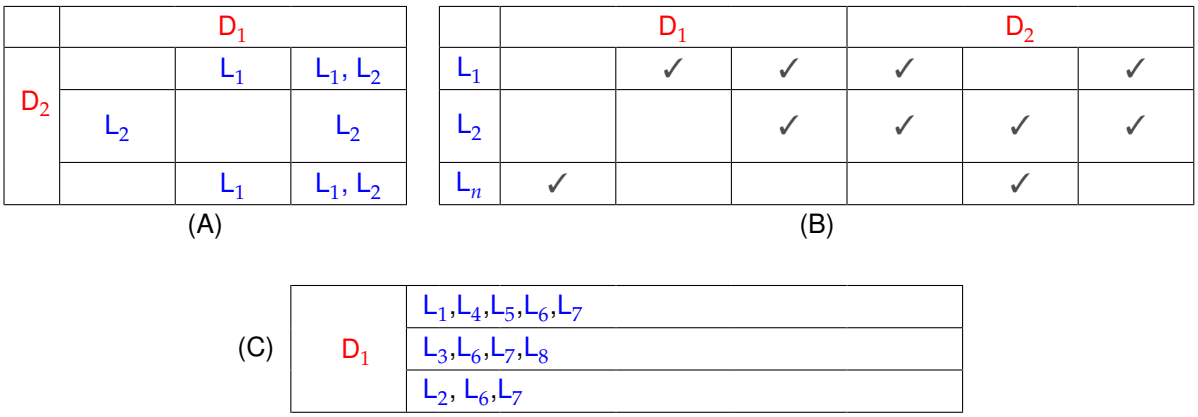


Figure A.3: Examples of classification schemes using 1-N mapping. D refers to a classification dimension and L refers to a reviewed item.

type, user task type, application domain, data processing size, performance, visualization design type, data type, and field of view.

User tasks are a useful and frequently-used classification dimension. They are the main focus of many survey papers [**?**, **?**, **?**]. For task taxonomies, we recommend reading Kerracher and Kennedy's work that focuses on the process and considerations for visualization task classifications [**?**]. As a good starting point for tasks, Schneiderman's task by data-type taxonomy is recommended [**?**], where *overview, zoom, filter, details-on-demand, relate, history,* and *extract* are presented as major visualization tasks.

A.3.4 Literature Classification Types ◆

We base this discussion on the work of McNabb and Laramee [**?**]. We identify three important characteristics of classifications: dimension, structure, and mapping schema. For this discussion, D denotes a classification dimension.

The **dimensionality** organizes the space in which the classification is laid out, for example in a table or matrix. A typical classification usually has no more than 3 dimensions (2 axes + 1 additional visual attribute). Common ways to represent an additional attribute are through color, shape, or symbols. More than 3 dimensions is definitely possible, however it may be worth considering multiple representations at this point, if the classification becomes too complex.

A **structure** represents the organization of the classification. This category is sub-divided into two types, flat or hierarchical. Flat structures usually represent subject categories (D) with a discrete linear ordering. Johansson and Forsell present an example of this with their evaluation of parallel coordinates [**?**] where the user-task is mapped for each reviewed literature. A hierarchy provides the subject categories (D) with a more complex arrangement by grouping overlapping subjects together. Draper et al. use this to categorize radial visualization techniques [**?**].

Mapping schema describes how the survey's reviewed literature (L) is mapped to classification dimensions (D). We introduce L to refer to a reviewed item (in most cases, the literature being reviewed). This is split into three categories, Unique-mapping, 1-n mapping, and indirect mapping. A unique-mapping schema assigns each reviewed item (L) once for every dimension e.g. subject category, data dimensionality, etc (D). This mapping schema is suitable for finding areas in the field with extensive or limited work, which may guide researchers to immature areas for new research.

Figure **??** presents some examples of **unique mapping**. Examples (A) and (B) map L to each of D once. However, example (A) structures the table such that both classification dimensions are represented by an axis and map L to the appropriate intersection. Example (B) maps L to the Y-Axis and each classification dimension D on the X-Axis. Example (C) links each of the reviewed items (L) to the appropriate classification dimension in the form of a list. Examples (A) and (B) show the same information.

An example of Figure **??**(A) would be Vehlow's taxonomy of group visualizations and group structures [**?**]. An example of Figure **??**(B) is with Nusrat and Kobourov with their task taxonomy for cartograms [**?**]. An example of Figure **??**(C) is with Behrisch et al. and their review of matrix re-ordering methods in network visualization [**?**].

1-*n* Mapping differs from the unique-mapping schema by allowing a reviewed item (L) to be mapped up to *n* times for each classification dimension (D) where *n* is the number of chosen attributes. Multiple-Attribute mapping matrices are most suited to comparing different elements, such as techniques or frameworks, against one another. These papers usually offer a checklist and present the criterion each paper fulfills or does not.

Examples of 1-*n* mapping can be found in Figure **??**. Examples (A) and (B) can map L to each of D multiple times. Example (A) structures the table such that both classification dimensions are represented by the X and Y axes and map the reviewed topics at their appropriate intersection. Example (B) plots reviewed items to the Y-Axis and each classification dimension on the X-Axis. This example provides a clear comparison of reviewed item's (L). Example (C) links each of the reviewed items (L) to the appropriate classification topics in the form of a list. Examples (A) and (B) show the same information. We do not recommend (A) or (C) as they can cause some confusion with literature being listed multiple times. An example of Figure **??**(B) can be found with Tominski et al. and their look at interactive lenses in visualization [**?**].

Some papers do not map L explicitly in their categorization and choose to display just their classification, using symbols rather than explicit citations. We call this **indirect mapping**. Some examples of this can be found in Sedlmair et al.'s taxonomy [**?**] which classifies data characteristics between two different classification dimensions, class-factors and influences. Another example of this is Heinrich and Weiskopf's state-of-the art report for Parallel Coordinates [**?**], which presents a hierarchical view of the important topics within the field. This representation does not explicitly show how literature maps the specified topics.

A.3.5 Paper Centered vs Topic Centered ◆

If your goal is to help users understand an area, you can produce a survey focused on underlying topics rather than the research literature. We consider surveys that dedicate more space and content to topics (as opposed to individual research papers) to be topic centered. In this case, the literature is surveyed in the hope of creating a novel framework that can be applied to a field. Landesberger et al. provide a good example of this is their state of the art in the visual analysis of large graphs [**?**]. See Tong et al. for an example of a paper-centered survey where the content is more focused on research papers [**?**].

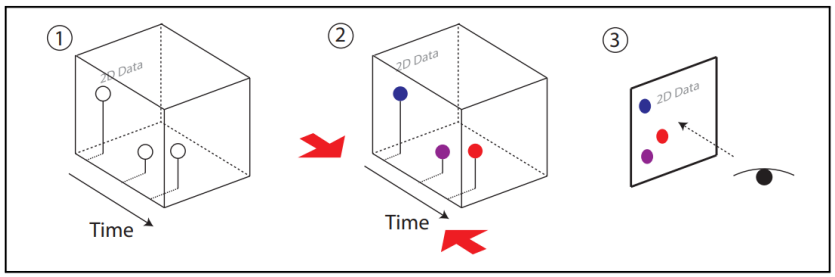


Figure A.4: The colored time flattening operation for space-time cubes. An example depicting a technique using visual rerpresentation. Image courtesy of Bach et al. [**?**]. Refer to Section **??**

A.4 Complementary Material

Although the core of your survey resides in the classification and surveyed material, your paper does not need to end there. In this section, we discuss some additional options to improve your survey paper. We first focus on collective meta-data and some additional figures to improve the comparison of papers, followed by some options derived from the literature.

A.4.1 Figures

As well as presenting some of the work from the summarized papers, figures can be used to enhance understanding of the presented concepts. Bach et al. provide an excellent example with hand-drawn illustrations of operations for space-time cube flattening operations (Figure **??**) [**?**]. Hadlak et al. present different facets of graph-structured data that are commonly visualized on the survey of multi-faceted graph visualization [**?**]. Figures can be used to present more than just an introduction to a concept. Caserta and Zendra, present a comparison of similar systems using different visual techniques to give a comparative view [**?**]. Schulz et al. present a similar example in their design space of implicit hierarchy visualization [**?**] (Figure **??**). Borgo et al. use comparisons to present visual variables, as well as their glyph design criteria in their glyph-based visualization survey [**?**]. Javed and Elmqvist use figures to present the differences between composite visualization using a scatter plot and bar graph as examples [**?**]. Vehlow et al. present something similar in their state of the art in visualizing group structures in graphs [**?**]. Tominski et al.'s duology of surveys on interactive lenses in visualization provide good examples of comparative views of techniques and a depiction of an interactive lens technique [**?**, **?**].

Mentioned in Section **??**, figures can be used to break down your dimensions. For example if you use a pipeline, presenting it as an image is a useful approach in making sure

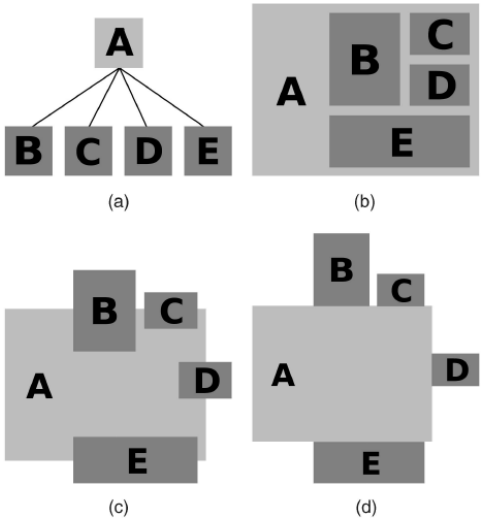


Figure A.5: (a) explicit node-link layout. (b) implicit node-link by inclusion. (c) implicit node-link by overlap. (d) implicit node-link by adjacency. These figures display the same system. Courtesy of Schulz et al. [**?**]

the reader understands the concept. Chi use this concept to present the information visualization data state reference model, which they use to create a taxonomy of visualization techniques [**?**]. Cottam et al. superimpose sources of dynamics onto Chen et al.'s information visualization pipeline [**?**, **?**]. Landesberger et al. present their scope and organization in the form of a venn diagram, looking at the main components of visual analysis of large graphs [**?**]. Wagner et al. present a pipeline depicting the scope and organization of the paper reviewing malware systems [**?**]. For more information on frameworks, Wang et al. present a survey on visual analytics pipelines which may be a good starting point [**?**].

A.4.2 Collective Meta-Data for Comparison

A full survey can occupy more than twice the length of most research papers, and therefore pacing is very important. In order to facilitate comparison of research literature and enhance the sections, you can use the collective meta data to present interesting observations and trends in the literature and make comparisons.

A common set of meta-data to visualize is a *distribution of the literature*. Beck et al. present a histogram to present the number of literature papers per year, with the publication type distribution mapped to color (Figure **??**) [**?**]. Federico et al. also present a histogram of literature distributed by the year [**?**]. Both of these are very prevalent and are often presented together, for example, Blumenstein et al. present a stacked bar chart to display the selected literature's venue, stacked based on the publication year [**?**].

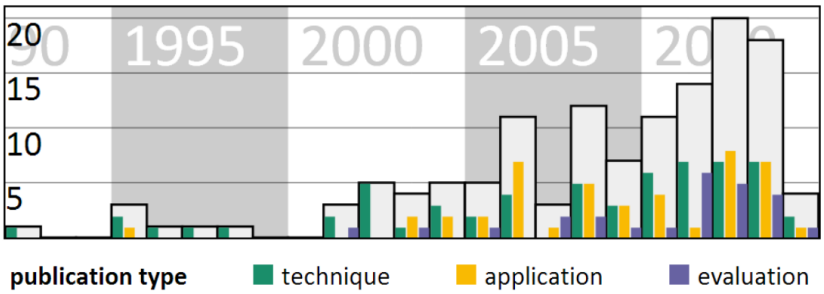


Figure A.6: Yearly number of publications on dynamic graph visualization according to our literature database; light gray bars indicate the total number of publications, colored bars distinguish the publications by type. Courtesy of Beck et al. [**?**]

Another common example comes with the *venues* (conferences or journals for example). Ko et al. use a histogram to present this in their survey paper [**?**]. Lipsa et al. present a line chart to present the distribution of papers amongst years and their venues [**?**]. Rather than looking at the venue, Janicke et al. break up the reviewed literature by the *field origin* (visualization or digital humanities) [**?**]. Alharbi and Laramee compare the number of methods presented in a paper against the *number of citations* using a bar graph. They also present a word cloud of their focus papers to present the differences in the vocabulary used within each [**?**]. Isenberg et al. produce a line chart depicting *paper counts per year*, showing trends within each publication venue [**?**]. Schneiderman et al. provide a similar example, presenting paper counts for three types of tree innovations [**?**]. McNabb and Laramee present a gantt chart depicting the *time frame for citations* across their reviewed literature, with number of citations mapped to color [**?**].

Edmunds et al. classify their papers using *relationship diagrams* to indicate how concepts are built upon each other [**?**]. Laramee et al. present a similar hierarchy of related literature, with their presented technique's dimensionality mapped to glyphs [**?**].

Merino et al. produce a sankey diagram to present the relationship between data collection and empirical evaluation in their survey software visualization evaluation [**?**]. Henry et al. present a wide arrange of meta-data visualization on timelines of paper scope, citation counts, and collaboaration diagrams, as well as more [**?**].

A collection of literature can often aid in the creation of a new framework. Even if you do not use this as a dimension in your classification, their is no reason to not include one. Chen et al. present a *conceptual pipeline* of traffic data visualization for the survey on the same topic [**?**]. Dasgupta et al. end their paper by presenting their visual uncertainty pipeline which they believe extends past the scope of their parallel coordinates survey [**?**]. Liang and Huang provide multiple models, including a conceptual model of highlighting, and a framework of

Meta-Data types	Examples
Publication year	Beck et al. [**?**], Federico et al. [**?**]
Publication venue	Ko et al. [**?**], Henry et al. [**?**], Isenberg et al. [**?**]
Data Origin	Janicke et al. [**?**], Wanner et al. [**?**]
Number of citations or impact	Alharbi et al. [**?**], Isenberg et al. [**?**], Schneiderman et al. [**?**]
Year span of cited literature	McNabb and Laramee [**?**]
Evaluation Methods	Isenberg et al. [**?**], Lam et al. [**?**]
Literature/Author Relationships	Edmunds et al. [**?**], Laramee et al. [**?**]
Test result comparisons	Lam et al. [**?**], Zhang et al. [**?**]
Task Types	Fuchs et al. [**?**], Ahn et al. [**?**], Nusrat and Kerrarcher [**?**]
Additional frameworks	Chen et al. [**?**], Dasgupta et al. [**?**]
Interactive Literature Browsers	Beck et al. [**?**], Kucher et al. [**?**]
Any un-used classification dimensions (refer to Section ??)	
Examples: Visual Design [**?**], Data Set Size , Data Characteristics [**?**, **?**], Data Dimensionality [**?**, **?**] Keywords [**?**], Pipeline classification [**?**], Feature Comparison [**?**], Supplementary Material Classification [**?**, **?**, **?**].	

Table A.2: A shortlist of potential collective and comparative meta-data.

viewing control [**?**]. Lu et al. present a predictive visual analytics pipeline for their survey of the same topic [**?**]. Mattila et al. design a pipeline to present the stages of research [**?**]. Zhou et al. present a framework depicting an edge-bundling framework for their survey on the same topic [**?**].

If you have clearly labeled a scope, you may be able to present some data about options outside of your scope. Fuchs et al. use a stacked bar chart to display the ratio of two different *evaluation types*, where a lower saturation indicates experiments evaluating design variations of the marks (those being reviewed) and a higher saturation for other experiments (not reviewed) [**?**].

Lam et al. review studies presented in their papers [**?**]. A bar chart is used to show the distribution between process scenarios and visualization scenarios. They use lines to further evaluate visualization and process scenarios between 1995–2010, by breaking down the scenarios. Isenberg et al. review *evaluation scenarios* using both histograms and line charts [**?**]. Zhang et al. perform stress tests on the different commercial systems they review, and present them in the form of a bar graph [**?**]. See Table **??** for a breakdown of collect meta-data samples.

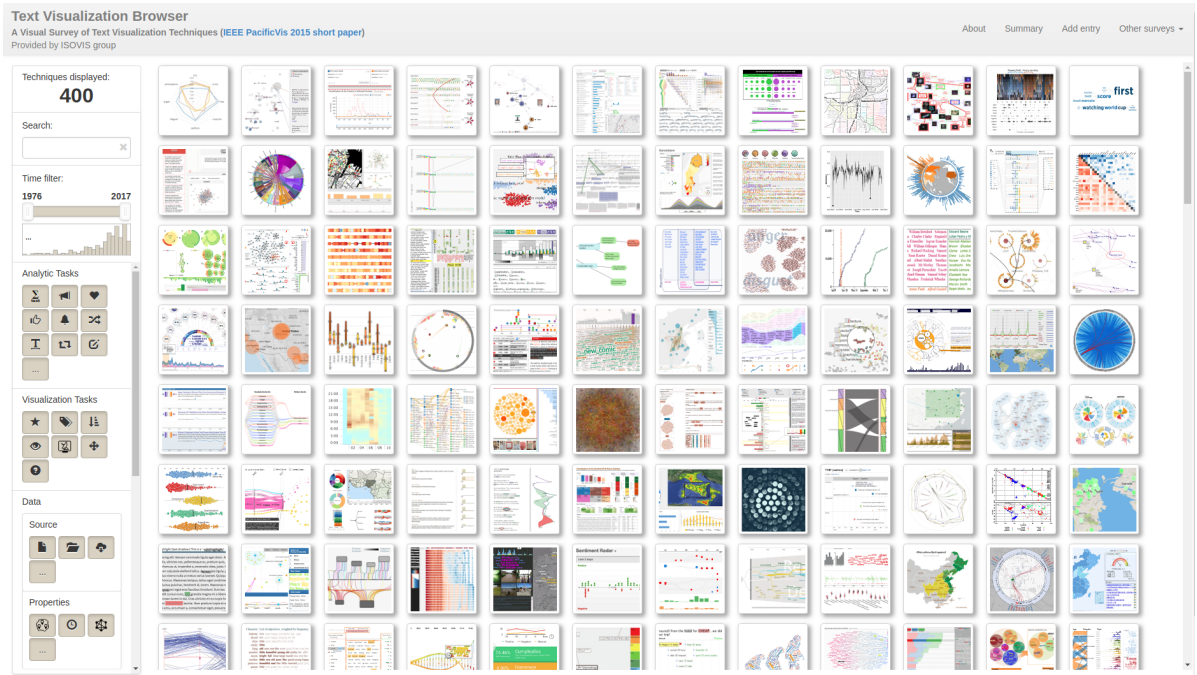


Figure A.7: Kucher and Kerren's interactive literature browser . [**?**]

A.4.3 Interactive Literature Browsers

Interactive literature browsers have become increasingly popular in the realm of visualization survey papers. The browser enables readers to interactively browse the findings of the paper to aid exploration. McNabb and Laramee's SoS paper find 13 literature browsers from the last 5 years, making this a worthwhile consideration. Although it is possible to produce a unique browser design, we recommend SurVis [**?**] as an option, due to it's open source, and easy-to-implement design.

Kucher and Kerren create their own interactive browser providing the opportunity for users to filter through different text visualization techniques, as well as allowing for user submitted entries (Figure **??**) [**?**]. Behrisch et al. provide a complimentary website to help readers compare matrix plots, as well as present additional meta data on the reviewed literature [**?**]. Dumas et al. present a website to review visualization focused on financial data [**?**].

A.5 Discussion and Future Challenges

The discussion and future challenges sections are critical areas within survey papers. The section summarizes any challenges presented in the research papers in generalized terms.

By the end of this section, readers could clearly understand what directions should be taken to further the field. If you are having trouble finding the content, domain expert feedback can be used to draw out more areas with less work undertaken.

We recommend referring to the classification that you have designed. It is likely that you have noticed trends throughout the compilation process such as missing or less mature areas within your classification dimensions. Two dimensional classifications are very good at pointing out both mature and immature research directions. Look for holes in your classification table or matrix. Your paper summaries can also be used to identify future research topics. If you notice key topics that seem to be missing or less mature (other potential classification dimensions for example), this is also worth mentioning. Look out for the following when developing your discussion: 1) Empty spaces in the classification table, 2) mature areas with dense research, 3) temporal trends such as early pioneers in the field and trends in only the last few years and 4) trends in publication venues.

A.6 Future Work

There are many avenues of future work that we could invest our efforts into for further papers. For this paper, we have focused on our own expertise and experience, however there our many different survey authors with many different styles and pieces of advice to share. Therefore, we would like to expand our guidance to hold more perspectives. We also think that exploring more example figures to discuss what makes them effective, and how they are used. Finally, we only briefly talk about the journey you follow in the temporal planning section. We believe discussing the intermediate stages and milestones could aid prospective writers overcome mental barriers in writing a successful survey paper.

A.7 Conclusions

We provide starting point guidelines with a flexible template for the reader to follow in order to produce a full visualization survey paper. The paper offers step-by-step instructions in order to guide the reader through each section of a typical survey, as well as additional considerations that need to be made during the writing cycle. The paper reviews what we consider essential topics and supplementary options that can be used to improve the quality of a survey paper and increase the chance of succeeding through the review process.

Appendix B

Miscellanious

B.1 Intersection Tester Doxygen Sample

◆ areLineSegmentsIntersecting()

bool IntersectTester::areLineSegmentsIntersecting (LineSegment a,
LineSegment b
)

static private

areLineSegmentsIntersecting we check the vector direction between the points. If the both ends of the line segment sit on seperate sides of the opposing line,than they intersect. Otherwise we also check if the lines sit on top of each other.

Parameters

a LineSegment a

b LineSegment b

Returns

if line segments intersect, return true, otherwise return false

Definition at line 96 of file IntersectTester.cpp.

```
 97 {
 98     int or1, or2, or3, or4;
 99     or1 = direction( a.getStart(), a.getEnd(), b.getStart() );
100     or2 = direction( a.getStart(), a.getEnd(), b.getEnd() );
101     or3 = direction( b.getStart(), b.getEnd(), a.getStart() );
102     or4 = direction( b.getStart(), b.getEnd(), a.getEnd() );
103 
104     if (or1 != or2 && or3 != or4 )
105         return true;
106     if( or1 == 0 && isIntersecting( b.getStart(), a ) )
107         return true;
108     if( or2 == 0 && isIntersecting( a.getStart(), b ) )
109         return true;
110     if( or3 == 0 && isIntersecting( b.getEnd(), a ) )
111         return true;
```

B.2 Table of Data

We add a table of data with some useful websites that are used during the process of the thesis.

Name	Description	Reference
Shapefiles		
GADM	GADM holds a large selection of Shape Files for countries across the world. It was always the first place I looked when reviewing potential shape files. (Germany, Italy, Brazil, UK, Japan, etc.) http://www.gadm.org/country	[?]
Texas	Shapefile used to represent Texas https://catalog.data.gov/dataset/	[?]
Hawaii	Shapefile used to represent Hawaii http://planning.hawaii.gov/gis/download-gis-data/	[?]
LSOA Wales	Extra boundaries used to represent Wales https://data.gov.uk/dataset/output-areas-oa-boundaries	[?]
US Counties	Counties of the United States http://www.census.gov/geo/maps-data/data/cbf/cbf_counties.html	[?]
Datasets		
LSOA Wales	Population Data https://bit.ly/2hkV0a6	[?]
US Counties	Assortment of county data for the us counties https://www.census.gov/library/publications/2011/compendia/usa-counties-2011.html	[?]
Public Health England	Public Healthcare data for the CCGs of England https://fingertips.phe.org.uk	[?]
Ordnance Survey Ireland	Census Data for the Republic of Ireland https://bit.ly/2Dyg7Ac	[?]

B.3 Introduction to QGIS (Guidelines for QGIS Tutorial for CGVC 2018)

We include an appendix piece used to guide a tutorial that introduces QGIS as a useful visualization tool at the Computer Graphics and Visual Computing Conference for 2018. The tutorial aims to give an understanding of how QGIS can be used to explore two depicitions of data. Example data is provided.

Introduction to QGIS with Liam McNabb

Basics:

(0) Download QGIS: https://qgis.org/en/site/

(1) Download the data: http://cs.swan.ac.uk/~csmcnabb/CGVC2018/Tutorial-QGIS/

(2) Uncompress the data a folder of your choice

(3) Understand the data of Example 1

-- "OSOpenMapLocal (ESRI Shape File) SS" & "OSOpenMapLocal (ESRI Shape File) ST"

Holds extra shape file information to aid searching (from OSOpenMap ordanance survey free version)

-- "2016-01-south-wales-street.csv"

The data file we will be exploring

-- "lsoaWales.*"

File retaining information needed to display the lower super output areas (LSOA) of Wales.

QGIS Browser:

(4) Open QGIS Browser

QGIS is a lightweight browser for geo-spatial files.

(5) Find “lsoaWales.shp” in the file browser (located on the left)

You can see the metadata held by the file

(6) Click the “Preview” tab

Clicking the preview tab provides an overview of the shape file’s form

(7) Click the “Attributes” tab

Data held for each polygon in the shape file

Example 1 (Exploring Point Data):

(8) Open QGIS Desktop

(9) Go to taskbar -> project -> New

(10) Add LsoaWales to the canvas. Go to taskbar -> Layer -> Add Layer -> Add Vector Layer

Or use the browser panel

(11) Add the data,

-- Go to taskbar -> Layer -> Add Layer -> All Delimited Text Layer

We user the geometry definition “point coordinates”, using the longitude and latitude to plot the points.

-- Add a coordinate reference system WGS 84, EPSG:4326.

The coordinate reference system can be split into two types: Geographic or Projected, we need to use the geographic to match the point data’s coordinates to the underlying map of Wales, which uses a separate coordinate system. We are using the World Geodetic System 1984 for this projection

(12) Make your map beautiful

-- Right click lsoaWales in the layers panel (below the browsers panel) and hover of styles.

The “edit styles” option gives more advanced settings.

-- Go to taskbar -> project properties. Select the general tab, where you can change the background color (if desired).

(13) Go to taskbar. View -> Zoom In. Draw a box around the desired area (Swansea in this example)

(14) Increase understanding of the crime data.

-- Right Click the crime data in the layers panel. Go to properties. Go to Open Attribute Table.

This shows the data stored in the CSV data file.

-- Right Click the crime data in the layers panel. Go to properties. Go to style. Top-left is a drop-down signifying “Single Symbol”. Switch to “Categorized”

We are going to color the points based on an attribute in the data.

-- Use the dropdown to select “Crime Type”. Click the button “Classify”.

This will set a random color to each crime type, and color the points appropriately. If you want, you can also change the color map, and symbol style which are under the Column dropdown.

-- Click on the “+” symbol by the Crime Data in the layers panel to see the legend

You can use the check boxes to filter the points

(15) Increase the understanding of the underlying geography

-- Click the first button wihtin the layers panel to add a group.

-- Open the folder “OSOpenMapLocal (ESRI Shape File) SS”, drag both shp files over the group in the layer panel.

-- Click on the group, and drag above LsoaWales

Although not necessary adding these files allows for easier understanding of the spatial area. These can be recolored in the same vein as lsoaWales (Right click -> Styles)

(16) Review a single point

-- Open the attribute table

-- Make sure the crime data is highlighted in the layers panel (clicked)

-- User the “Select Features button

-- Drag a box around the point you are interested in

-- Select the 9th button in the attribute table to bring the selected item to the top of the database.

Add the second folder of example one to explore Cardiff in the same way. Once you are done. Save and start a new Project.

Example 2 (Creating a choropleth map):

(17) Understand the data of Example 2

(18) Drag us_county shp file to the layers panel

(19) Change the projected view

-- Go to the bottom right corner and look for the globe symbol. Click on it.

-- Enable 'on-the-fly CRS transformation'

-- Change the CRS to NAD83/ Texas Centric Albers Equal Area, EPSG:3083

We used a project view for this. Thanks to the size of the US, representing it on a flat plane isn't ideal, we use a projected view to emulate the projection onto a sphere (like the globe). The system refers to where you are viewing the globe from. In this case we are telling the projection the Texas is in line with our viewpoint.

(20) Add the "IncomeData" to the layers panel

We don't have any point information for this layer, so we just add it as a file-in-use.

(21) Link the two files together

-- Open both Attribute tables and find common fields (in our case, STCOU in the data file, and GEOID in the shp file.

-- Right click the US shape file, go to properties. Select the Joins tab. Click the "+" button.

-- Join Layer = "IncomeData", Join Field = STCOU, Target Field = GEOID. Click OK.

By doing this, we can use fields in the csv file, with the data in the shapefile. We're going to use this to visualize the income data for each county in the US.

(22) Present the Joined Data

-- Go to the Styles Tab. Click the "Graduated" option in the top-left dropdown box

We use this instead of categorized to present number values

-- Click the ε button

-- Use the middle widget to create an expression to convert the field to a number. Conversions -> to_real, Fields and Values -> IncomeData_INC110179D, close the bracket.

-- Click the "Classify" button

You can change the color map using the color ramp. There are also different mode for distributing the values in the drop down menu "Mode"

(23) Export

-- Go to the toolbar -> project -> new print composer

-- Use the "Add new Map" button

The map thats added reflects what is shown in the QGIS Desktop canvas.

-- Modify the extents in the right widget, "Item Properties".

-- Go back to QGIS desktop and zoom in on alaska

-- Use the "Add new Map" button again

-- Use the "Add legend" button

-- Use the toolbar, Composer Export as… (Your choice)

END OF TUTORIAL

Colophon

This thesis was made in LaTeX 2_ε using the “hepthesis” class [**?**].

List of figures

List of tables