Semantic Zoom View:
A Focus+Context Technique for
Visualizing a Document Collection

by

Dustin Dunsmuir
B.Sc., Simon Fraser University, 2008

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

In the
School of Interactive Arts and Technology

© Dustin Dunsmuir 2011

SIMON FRASER UNIVERSITY

Spring 2011

All rights reserved. However, in accordance with the *Copyright Act of Canada*, this work may be reproduced, without authorization, under the conditions for *Fair Dealing*. Therefore, limited reproduction of this work for the purposes of private study, research, criticism, review and news reporting is likely to be in accordance with the law, particularly if cited appropriately.

APPROVAL

Name: **Dustin Dunsmuir**

Degree: **Master of Science**

Title of Thesis: **Semantic Zoom View: A Focus+Context Technique for Visualizing a Document Collection**

Examining Committee:

Chair:

____
Dr. Halil Erhan
Assistant Professor
School of Interactive Arts and Technology

____
Dr. John Dill
Senior Supervisor
Professor Emeritus
School of Interactive Arts and Technology

____
Dr. Christopher Shaw
Supervisor
Associate Professor
School of Interactive Arts and Technology

____
Dr. Robert Woodbury
External Examiner
Professor
School of Interactive Arts and Technology

Date Defended/Approved: March 22, 2011

SFU SIMON FRASER UNIVERSITY LIBRARY

Declaration of Partial Copyright Licence

The author, whose copyright is declared on the title page of this work, has granted to Simon Fraser University the right to lend this thesis, project or extended essay to users of the Simon Fraser University Library, and to make partial or single copies only for such users or in response to a request from the library of any other university, or other educational institution, on its own behalf or for one of its users.

The author has further granted permission to Simon Fraser University to keep or make a digital copy for use in its circulating collection (currently available to the public at the "Institutional Repository" link of the SFU Library website <www.lib.sfu.ca> at: <http://ir.lib.sfu.ca/handle/1892/112>) and, without changing the content, to translate the thesis/project or extended essays, if technically possible, to any medium or format for the purpose of preservation of the digital work.

The author has further agreed that permission for multiple copying of this work for scholarly purposes may be granted by either the author or the Dean of Graduate Studies.

It is understood that copying or publication of this work for financial gain shall not be allowed without the author's written permission.

Permission for public performance, or limited permission for private scholarly use, of any multimedia materials forming part of this work, may have been granted by the author. This information may be found on the separately catalogued multimedia material and in the signed Partial Copyright Licence.

While licensing SFU to permit the above uses, the author retains copyright in the thesis, project or extended essays, including the right to change the work for subsequent purposes, including editing and publishing the work in whole or in part, and licensing other parties, as the author may desire.

The original Partial Copyright Licence attesting to these terms, and signed by this author, may be found in the original bound copy of this work, retained in the Simon Fraser University Archive.

Simon Fraser University Library
Burnaby, BC, Canada

Last revision: Spring 09

SIMON FRASER UNIVERSITY
THINKING OF THE WORLD

STATEMENT OF ETHICS APPROVAL

The author, whose name appears on the title page of this work, has obtained, for the research described in this work, either:

(a) Human research ethics approval from the Simon Fraser University Office of Research Ethics,

or

(b) Advance approval of the animal care protocol from the University Animal Care Committee of Simon Fraser University;

or has conducted the research

(c) as a co-investigator, collaborator or research assistant in a research project approved in advance,

or

(d) as a member of a course approved in advance for minimal risk human research, by the Office of Research Ethics.

A copy of the approval letter has been filed at the Theses Office of the University Library at the time of submission of this thesis or project.

The original application for approval and letter of approval are filed with the relevant offices. Inquiries may be directed to those authorities.

Simon Fraser University Library
Simon Fraser University
Burnaby, BC, Canada

Last update: Spring 2010

ABSTRACT

In the field of visual analytics, analysts need overviews of large amounts of data. This becomes a challenge when working with non-numerical data such as document collections. This thesis describes the design and use of a new visualization technique called Semantic Zoom View (SZV), which provides an interactive overview of a document collection combined with a detailed view of entities contained in the documents (people, places, etc.) and full text of each document. SZV lets analysts easily and quickly see the main topics of a document collection. Any subset of documents can be semantically zoomed to show increasing detail as the zoom level increases, while keeping surrounding documents visible to supply context. This tight integration of focus within context encourages and facilitates the iterative process of finding relevant documents and reading them. An evaluation compares the described technique to an overview+detail technique for finding answers within a document collection.

Keywords: semantic zoom; focus+context; document collection; visual analytics

ACKNOWLEDGEMENTS

Thank you to John Dill, Christopher Shaw, Maureen Stone, and Tamara Munzner for guidance, design feedback and ideas throughout the life of my thesis. Thanks also to Robert Woodbury and the entire CZSaw team past, present and future for constructive criticism and support. I would also like to acknowledge all those who participated in the evaluation or provided comments during demonstrations of Semantic Zoom View.

Lastly, despite not playing a role in this project, I would like to thank all my family and friends for their support leading up to my masters.

TABLE OF CONTENTS

Approval ii
Abstract iii
Acknowledgements iv
Table of Contents v
List of Figures vii
List of Tables ix

1: Introduction 1
1.1 Domain of Interest – Text Document Analysis 1
1.2 Research Objectives 3

2: Related Work 5
2.1 Document Collection Visualization 5
2.1.1 Information Retrieval 5
2.1.2 Overviews 7
2.2 Zooming 10
2.3 Focus + Context 13

3: Design 22
3.1 Zooming 22
3.1.1 Level 1: Simple Glyph 23
3.1.2 Level 2: Labelled Glyph 24
3.1.3 Level 3: Entity Glyphs 24
3.1.4 Level 4: Labelled Entity Glyphs 24
3.1.5 Level 5: Document Full Text 25
3.1.6 Interaction 25
3.1.7 Implementation 27
3.2 Overview 29
3.2.1 Layout Algorithm 30
3.2.2 User Interaction 33
3.2.3 Layout Adjustments caused by Zoom 34
3.3 Query 37
3.3.1 Highlight by Date 38
3.3.2 Search 39
3.3.3 Brushing Entities 40
3.4 Structuring a Document Collection 41

4: Evaluation 49
4.1 Approach 50
4.2 Interface Alternatives 53

4.3 Study Design....56
4.3.1 Method....57
4.3.2 Analysis Design....59
4.4 Results....60
4.4.1 Quantitative Results....60
4.4.2 Qualitative Results....64
4.5 Limitations....65
4.6 Related Studies....66

5: Discussion....68
5.1 A Component of CZSaw....68
5.2 Strengths and Weaknesses....70

6: Conclusion....74
6.1 Future Work....74
6.2 Summary....75

Appendices....77
Appendix A: Technical Description....77
Appendix B: Evaluation Documents....93
Pre-Study Questionnaire....93
Reading Exercise....94
Example Questions....95
Task Questions....96
Post-Study Questionnaire....98
Appendix C: Qualitative Notes....101

Reference List....106

LIST OF FIGURES

Figure 2-1. In-Spire’s views. a) Galaxy View. b) Theme view. (Hetzler et al. 2004)........ 8

Figure 2-2. Starlight's Data Sphere view in 3D space next to other views (Risch et al. 1997).. 9

Figure 2-3. The Pad system with 3 portals showing different locations and zoom levels. (Perlin et al. 1993)........ 11

Figure 2-4. The DataSplash system with the layer manager on the right........ 12

Figure 2-5. Zooming using the NaviQue prototype (Furnas et al. 1998)........ 13

Figure 2-6. Document Lens (Robertson et al. 1993)........ 15

Figure 2-7. The expansion of cluster nodes in the variable zoom (Schaffer et al. 1996)........ 17

Figure 3-1. The semantic zoom levels of a document from a simple glyph to the document full-text........ 23

Figure 3-2. How ZVTM is used by the SZV. Changes in camera altitude change glyphs’ onscreen sizes. 28

Figure 3-3. An example of a layout in Semantic Zoom View........ 30

Figure 3-4. Computing a weighted edge between two documents based on common entities........ 31

Figure 3-5. The available layout options........ 33

Figure 3-6. Left: A cluster layout before any zooming with the document to zoom circled. Right: The layout with the document zoomed in........ 35

Figure 3-7. The layout adjustment algorithm for an expanding document. 36

Figure 3-8. The Highlight by Date feature. 38

Figure 3-9. The Search options in Semantic Zoom View. 39

Figure 3-10. Brushing an entity to highlight the other documents containing it........ 41

Figure 3-11. A sequence showing the animation of documents caused by the gather function........ 43

Figure 3-12. Creating a group by drawing a rectangle around a subset of documents........ 44

Figure 3-13. A group’s document tab shows the document glyphs within........ 45

Figure 3-14. A group's entity tab displays all contained entities. 46
Figure 3-15. The dynamic entity filter changes which entities are visible in a group. 46
Figure 3-16. A group's text tab shows contained documents' full text. 47
Figure 4-1. The study's zoom interface condition immediately following brushing. 54
Figure 4-2. The study's popup interface condition immediately following brushing. 55
Figure 4-3. The accuracy per question for each condition. 61
Figure 4-4. Average completion time for each question by interface (one standard deviation shown). 63

LIST OF TABLES

Table 2-1. A summary of this chapter’s document visualizations. 19

Table 2-2. A summary of this chapter’s systems using zooming. 20

Table 2-3. A summary of this chapter's focus + context techniques............................. 21

Table 4-1. Results of one-tailed Fisher's exact tests on questions that differed in accuracy between the two conditions.. 62

1: INTRODUCTION

According to the research agenda for visual analytics, “Visual analytics is the science of analytical reasoning facilitated by interactive visual interfaces.” (Thomas et al. 2005). This new field was created initially to help intelligence analysis in the aftermath of the U.S. World Trade Center attacks on September 11, 2001 (9/11). Today visual analytics is useful for making sense of the massive amounts of data in virtually every industry such as business intelligence and the health sector. Visual analytics uses an interdisciplinary approach that combines information visualization, human computer interaction, data mining, statistics, cognitive science, and psychology. Visual analytics research and development provide tools to analysts. The visualizations provided by these tools must effectively meet human perceptual abilities without distorting the data. The automated reasoning of these tools must speed analysis but be controlled and understood. The two must be combined into an intuitive application that analysts will adopt.

While techniques for visualizing numerical and structured data have been around for centuries, the development of methods for visualizing unstructured text documents are relatively new. Text documents consist of natural language elements not easily understood by computers. Fortunately, in the last 15 years there have been some advances in using the statistical properties of text to provide overview visualizations of document collections (Olsen et al. 1993; Wise et al. 1995; Risch et al. 1997). In this thesis, I present a new visual analytics technique that combines a document collection overview with a flexible focus + context drill-down mechanism. Since it makes use of a selective semantic zoom interaction to drill-down into sets of text documents, the technique has been named Semantic Zoom View (SZV). The balance of this introduction focuses on SZV’s domain of interest and the objectives of this research.

1.1 Domain of Interest – Text Document Analysis

Everyday, journalists write newspaper articles, field agents write intelligence reports, pilots fill out incident reports, and more than 247 billion emails are sent (Radicati 2010). Intelligence and business analysts must make sense of these huge collections of text documents in their search for trends, patterns or suspicious events. Without visualizations, the analyst’s iterative process consists solely of refining queries and skimming or reading documents. Wise et al. (1995)

describe weaknesses of such a workflow. First, quick query refinement makes it easy to exclude high value documents. Second, reading documents is time consuming and in attempting to shorten their task, analysts may prematurely select a hypothesis without fully exploring alternatives. An effective document collection overview can change this workflow as Wise et al. demonstrate (1995). Overviews also reduce the number of missed relevant documents.

When reading text documents, analysts are often concerned with the people, places and events they describe. Identifying and tagging these elements allows algorithms to deal with them explicitly. For this purpose, the text analytics and natural language processing communities have developed entity extraction tools that work toward marking all words corresponding to people, places, dates or other specified important entity types. Each document's entity set provides analysts with a quick summary of the who, where, when, etc. focus of the document.

Algorithms can find the same entities within multiple documents. Documents may contain multiple entities (people, organizations, places, etc.) and specific entities may occur in multiple documents. The resulting "document contains entity" relationships constitute a network of documents and entities that can be viewed using graph (i.e. node-link) drawing and interaction techniques. Another visualization possibility is displaying documents as containers for entities. Semantic Zoom View (SZV) takes this latter approach.

SZV was created as part of a visual analytics system called CZSaw, developed for analysis of document and entity networks (Kadivar et al. 2009) and inspired by Stasko's Jigsaw (Stasko et al. 2008). CZSaw contains a *document view* for reading the text of documents and a *hybrid (graph) view* for visualizing the network in a node-link diagram. SZV provides overviews of the text document collection with an integrated quick drill-down ability, a perspective previously missing from CZSaw.

A separate CZSaw focus is on capturing and supporting the analysis process, allowing analysts to better understand and control it. CZSaw records the analysis process by saving all user actions in a human-readable and editable script. All aspects of CZSaw, such as views, query results, and visualizations, are represented as variables within the script language. CZSaw also creates a graph showing dependencies among these variables. A *history view* displays screenshots of the analysis allowing analysts to return to a previous state and annotate their analysis process (Kadivar et al. 2009).

1.2 Research Objectives

The Semantic Zoom View document viewing technique provides a flexible overview of a text document collection and quick drill down into these documents without losing a contextual overview. It can be assumed that at least two different perspectives on the documents are important to analysts. First, analysts need an overview of the document collection to understand the topics they contain and aid in exploratory search. Analysts also need to be able to read documents, so the text of any document or subset of documents must be quickly accessible. SZV was designed to provide not only these two capabilities but also to ease the transition between them. SZV was created to overcome the shortcomings of previous systems (e.g. Starlight (Risch et al. 1997), Jigsaw (Stasko et al. 2008)) while adhering to guidelines and principles developed in information visualization and visual analytics.

Spatial overviews of a document collection can greatly reduce the time needed to understand the topics of the collection compared to the traditional method of listing only documents resulting from a query (Olsen et al. 1993; Wise et al. 1995). SZV's first goal is to provide a starting point for investigation using a flexible overview. To accomplish this, it shows a document glyph layout that places more similar documents closer together where similarity is measured by the number of entities they have in common. The human eye can then perceive clusters within this continuous layout. SZV provides a search feature that highlights the resulting documents within the overview's clusters. Thus, having found one relevant document, the analyst can easily find other potentially relevant documents since documents that are semantically "near" appear spatially near. Section 3.2 describes how analysts can customize the layout to match their task. SZV follows Shneiderman's often repeated mantra, "Overview first, zoom and filter, then details-on-demand" (1996), where the overview is simply the first step in the process.

SZV's second goal is to provide quick access to document contents so analysts can immediately switch between investigating the collections' overall themes (shown by document clusters) and reading individual documents. Directly within SZV's overview, an analyst can semantically zoom into a subset of documents. Semantic zooming, described at the start of Section 2.2, is a form of zooming where more object details are revealed as it expands to fill more space (is zoomed). This provides quick details-on-demand through zooming document glyphs. As explained in Section 3.1, document glyphs reveal increasing detail as they are zoomed in including their set of entities and at full zoom, the document full text. Entity glyphs are displayed within document glyphs, matching the internal "document contains entity" model. This meets the Naturalness Principle, as outlined in *Illuminating the Path,* that a visual representation should match the represented information (Thomas et al. 2005). The visual metaphor should be intuitive. Expanding a document is similar to opening a file in most operating systems. Both actions are performed to see contained content. SZV also provides a mechanism to make persistent document groups, similar to folders in file management systems. Analysts can organize the SZV view as they would items on a desktop by manually moving documents into a group or creating a group based on search results (Section 3.4).

SZV differs from previous document overviews by providing document details directly within the overview. It uses a focus + context approach to expand focus document(s) in place directly within the view, adjusting other glyphs' positions so that they do not overlap. Models of analyst sense-making strategies show that displaying context is key when the information load is high (Woods 2002). Thus, SZV's third goal is to integrate focus (i.e. zoomed-in documents) directly within the overall context (i.e. surrounding zoomed-out documents) in order to provide analysts with context throughout their investigation. Analysts do not have to construct and maintain detail-to-context connections in their own memory. This in-place drill down capability allows a tighter integration between finding relevant documents and reading them, an important consideration because analysts frequently iterate between these two tasks. The content within documents they read leads to further queries and more documents being read. By combining all aspects of this iterative process into a single integrated document investigation visualization, SZV aims to reduce the cognitive load necessary for analysis. In addition, SZV's semantic zooming capability allows analysts to see as much or as little detail as they wish in order to match their current task. This meets the Appropriateness Principle described in *Illuminating the Path* (Thomas et al. 2005): visual representations should provide exactly the right amount of information needed for the task. Too much information may be distracting. Described in Section 3.3.3, brushing and linking entities within the view makes use of the visible context by allowing an analyst to highlight instantly all documents in the view containing an entity from documents they are currently focused on. Thus, when an analyst discovers an entity of interest, he can quickly investigate all documents containing that entity.

The many objectives of SZV outlined above are intended to provide a cohesive environment that efficiently allows an analyst to investigate, organize, and make sense of a text document collection. Chapter 2 will place this design in the context of related work. Chapter 3 describes SZV's design while Chapter 4 presents a user study to evaluate its use. Chapter 5 discusses strengths and weaknesses of the research. Finally, Chapter 6 concludes with suggestions for future work.

2: RELATED WORK

While there has been much progress in the visual analytics field, *Illuminating the Path* (Thomas et al. 2005), remains the best roadmap for the challenges facing the field. It also serves as a resource summarizing work in related fields of use in visual analytics.

In this chapter, I describe related work in three domains contributed to by this thesis research. First, I will provide a review of the work done on document collection visualizations including information retrieval and overviews. Second, I will describe the history of zoomable user interfaces and the development of semantic zooming within computer applications for providing an appropriate level of detail. Lastly, since Semantic Zoom View (SZV) uses a focus + context technique to embed semantic zooming into a document collection overview, I will describe past focus + context techniques.

2.1 Document Collection Visualization

This section begins with a brief history of information retrieval systems as the precursors to document visualization systems such as SPIRE (Wise et al. 1995). A later version, In-Spire (Hetzler et al. 2004), is the past system most similar to SZV's document collection overview. Both SZV and In-Spire's Galaxy view display a layout of documents as glyphs with similar documents placed close together, although as mentioned below and explained in Section 3.2, the model behind SZV's document layout is based upon entities instead of all document words. The Starlight (Risch et al. 1997) and Jigsaw (Stasko et al. 2008) systems are also described and compared to SZV below.

2.1.1 Information Retrieval

The Information Retrieval field was formed in the 1950s to use computers to search text archives (Singhal 2001). While the data querying algorithms are not directly related to this thesis, the resulting visualization techniques are. In the 1960s, evaluation methodologies were developed, and in the 1970s and 1980s, many new document retrieval models were developed and evaluated on small (several thousand article) collections. Since then, the information retrieval research community has extensively studied the display of search results including text document collection overviews. One aspect of this research was the clustering of search results, since documents that have similar content tend to be relevant to the same search requests (Rijsbergen 1979). Willet (1988) reviewed early work on this clustering of documents.

The focus of the information retrieval community has shifted to interactive techniques and the organization of search results (Leuski et al. 1994). Leuski et al. evaluated the clustered listing of retrieved document results (Leuski et al. 1996, 2001, 2004) and developed a document clusters visualization called Lighthouse. It visualizes each document result as a sphere that users can click to view the document text (Leuski et al. 2004). A standard spring-based layout places similar documents closer together. Users rate documents as relevant or not when viewing them, which then feeds back into the system. Spheres are displayed in shades of green or red if they are similar to the relevant or non-relevant documents respectively, which guide the user. Somewhat surprisingly, instead of an evaluation with human users, they used mathematical formulae to determine the system usefulness, assuming users would follow the strategies of the user relevance model. Although using a model instead of actual users reduces what they can claim about system utility, the model allows them to adapt the visualization (change the sphere's colours) based on user actions.

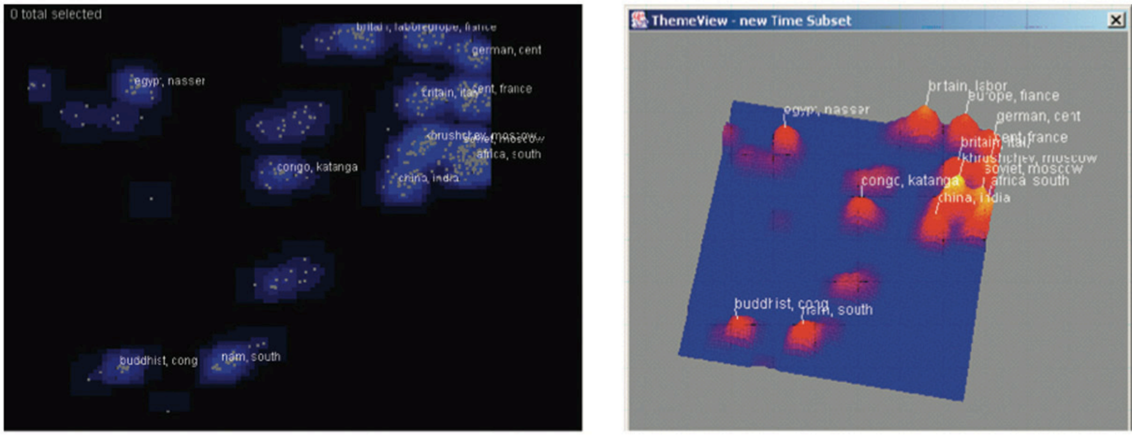
VIBE was a document retrieval system that displayed search results in a spatial visualization (Olsen et al. 1993). Unlike later systems, the document glyph layout was based on the terms used in the search queries rather than solely on the documents themselves. Users created multiple queries and defined them as points of interest (POI) in 2D space. Then search results (documents) were displayed as rectangle glyphs in positions relative to how much they met each query. By moving around POI and creating new ones, the users saw the location of search results in their defined space. This quickly allows identification of interesting document clusters. Thus, the document placement is easier to understand than more complicated statistical techniques.

Recently, Ahn et al. (2009) combined the ideas of Lighthouse and VIBE to provide an adaptive user interface within VIBE. They added POI based on a user model. The model comes from users' past queries and task notes. Documents more relevant to their task tend to be closer to the model-based POI than the normal POI. In SZV, there is no user model; however, analysts explicitly control which entity types determine the document layout in order to match their task (Section 3.2.2).

2.1.2 Overviews

The Spatial Paradigm for Information Retrieval and Exploration (SPIRE) system (Wise et al. 1995) developed at the Pacific Northwest National Laboratory (PNNL) contained some of the first interactive spatial visualizations providing a document collection overview. Traditional information retrieval techniques list only documents that match a query. During the iterative query refinement, relevant documents can be lost. SPIRE removes this problem by highlighting resulting documents within an overview of all documents. It takes advantage of the preattentive parallel processing powers of the human visual perception system (Treisman 1985), which allow users to understand a large layout of simple glyphs faster than reading text. Vectors are created for each text document with its statistical and semantic attributes such as word frequency. These vectors represent the documents in a high dimensional space before applying principle components analysis (PCA) or multi-dimensional scaling (MDS) to project the vectors down to two dimensions. SPIRE uses this process in its Galaxies display that represents each document as a point within the entire collection, a galaxy of points on a 2D plane. Documents with many of the same words tend to cluster and such clusters are labelled with their most common words. A tool called the temporal slicer partitions the document base into sections of time (from metadata) and then visualizes each period separately to show temporal change in the document collection.

SPIRE's ThemeScapes display visualizes themes (i.e. clusters) as a 3D landscape, displaying topics as labelled peaks. Close peaks represent topics with similar documents. Both the galaxy and landscape were chosen as metaphors that users would easily understand and be able to spot patterns within. Some informal use demonstrated this to be true with one user claiming to have found results within the data in 35 minutes that would normally have taken 2 weeks. The users found that the tool gave them more freedom to follow their analytical intuitions (Wise et al. 1995).



(a) **(b)**

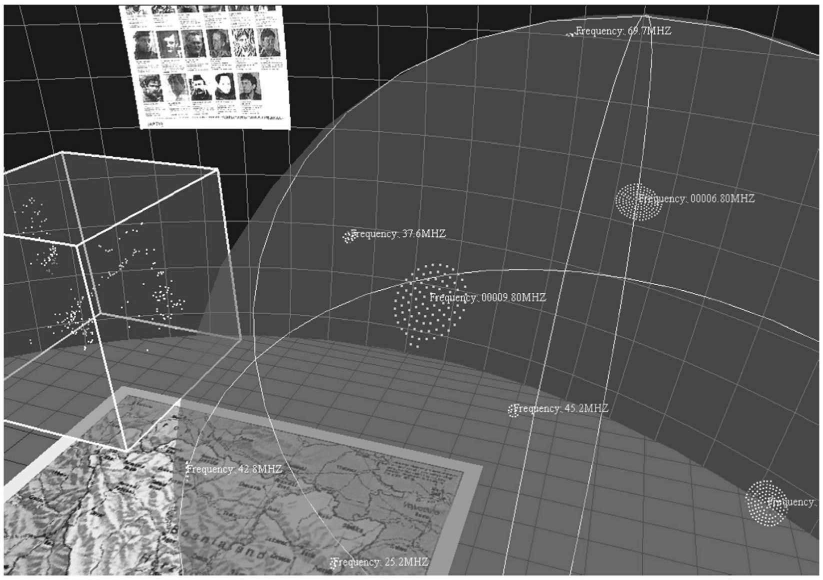
© 2004 IEEE, by permission

Figure 2-1. In-Spire's views. a) Galaxy View. b) Theme view. (Hetzler et al. 2004)

SPIRE's successor In-Spire has both Galaxy and Theme views (Figure 2-1). Hetzler et al. (2004) suggest that with limited time, analysts using traditional systems without visualizations may prematurely choose a hypothesis to support. They ran a formal user study with 24 highly skilled analysts from a variety of disciplines. With In-Spire, analysts were able to visualize collections as large as 70,000 documents and they claimed they found documents they would have otherwise missed. In-Spire was improved based on feedback from the study. One feature added was marking documents as outliers. This moves them outside the Galaxy view, which is then recalculated to adjust the layout for the remaining documents, yet these outliers can be easily restored, giving analysts more live control over the visualization.

Semantic Zoom View (SZV) resembles In-Spire's Galaxy view more than any other related work and as described in Chapter 3.2.2, it includes flexible controls to recalculate the visualization. One large difference between In-Spire's and SZV's layout algorithms is that SZV uses the document-entity model, introduced in Section 1.1. While this is a much sparser representation for a document than in In-Spire, it matches the internal data model of CZSaw (Kadivar et al. 2009) and can be more flexibly controlled (Section 3.2). As in most document overviews, In-Spire's Galaxy view allows users to read documents by clicking glyphs to show the document in a separate window. In-Spire also allows selecting a region of a view to open all documents within that region. SZV, as its name suggests, handles the viewing of documents quite differently by using a semantic zoom in a focus + context manner to expand documents directly within the main view.

Starlight (Risch et al. 1997), developed at PNNL, also provides document overview visualizations for intelligence analysis with many different data types, including structured and unstructured text, images or maps, and the relationships between them. Upon the importing of free text, Starlight applies statistical processing, such as Latent Semantic Indexing; and natural language processing, such as entity extraction. As mentioned in Section 1.1, entities are also a key element of SZV's (CZSaw's) data model.



© 1997 IEEE, by permission

Figure 2-2. Starlight's Data Sphere view in 3D space next to other views (Risch et al. 1997).

Starlight's visualizations can be viewed from within a three dimensional workspace. One of its visualizations is a star field view of documents, the Similarity Plot, similar to In-Spire's Galaxy view except in 3D. Similar documents are placed closer together. The Data Sphere view, in Figure 2-2, is for structured text or free-text that has some structure (for example entities). Information items are represented as glyphs on the surface of a sphere and are displayed in groups depending on the chosen field's value. The other fields' values can then be mapped to glyph shapes and colours. As in In-Spire, queries cause the resulting data items to be highlighted although in Starlight there is also a ranked results list. Starlight's strength is in its Link Nets visualization, a 3D visualization displaying the connections between multiple layers of data. Each layer can be images, maps, timelines, structured or unstructured text. While Starlight does not have a semantic zoom capability as in SZV, it does employ two different methods for reading the glyph's full text. The full text can be displayed next to the glyph or within a streaming label that updates to show parts of the document in sequence. Thus, an analyst may read the entire document one line at a time by waiting while it changes. The advantage of the first approach is that it displays all the text at once while the advantage of the second is that it does not cover up any of the other icons. As will be explained in Chapter 3, SZV has both of these advantages at once.

Jigsaw is a multiple view visual analytics system for investigation of text documents and their extracted entities (Stasko et al. 2008). It was the inspiration for CZSaw and so the systems have some similar visualizations, but CZSaw has the additional major capability of capturing the analysis process to visualize and reuse it (Kadivar et al. 2009). To support this, CZSaw has an underlying script and process visualizations not found in JigSaw, as described in Section 5.1. Jigsaw's document overview, Document Cluster view represents documents as small rectangles and analysts create document clusters by applying filters such as dividing the documents based on contained entities. Colour highlighting shows query results or brushing from other views. Documents must be transferred to the Document View to be read. In contrast, SZV has a continuous layout of the documents, which are read directly within the single view in a focus + context manner. Related work in zooming applications is discussed next followed by focus + context techniques.

2.2 Zooming

Semantic zooming is a class of techniques for showing a level of detail appropriate to the size of a displayed object. At a small size, a document may be represented by a simple rectangle, at a medium size as a summary, and at a large size by its full text. SZV uses semantic zooming to provide multiple levels of detail for documents. In this section, I will first describe zoomable user interfaces, many of which contain semantic zooming. Then I will describe systems that use semantic zooming or are tools for creating semantic zooming applications.

Bederson recently published a review of Zoomable User Interfaces (ZUIs) and defines them as, "systems that support the spatial organization of and navigation among multiple documents or visual objects" (2009). Essentially ZUIs are interfaces that organize information at varying scales in a large continuous space and then use zooming and panning as the main navigation modes. Such systems can employ semantic zooming to hide details of visual objects that are too small to be useful. In his review, Bederson outlines the history of ZUIs, describes their strengths and weaknesses found through past studies, and includes a table of many known ZUI systems, from photo browsers to presentation software.

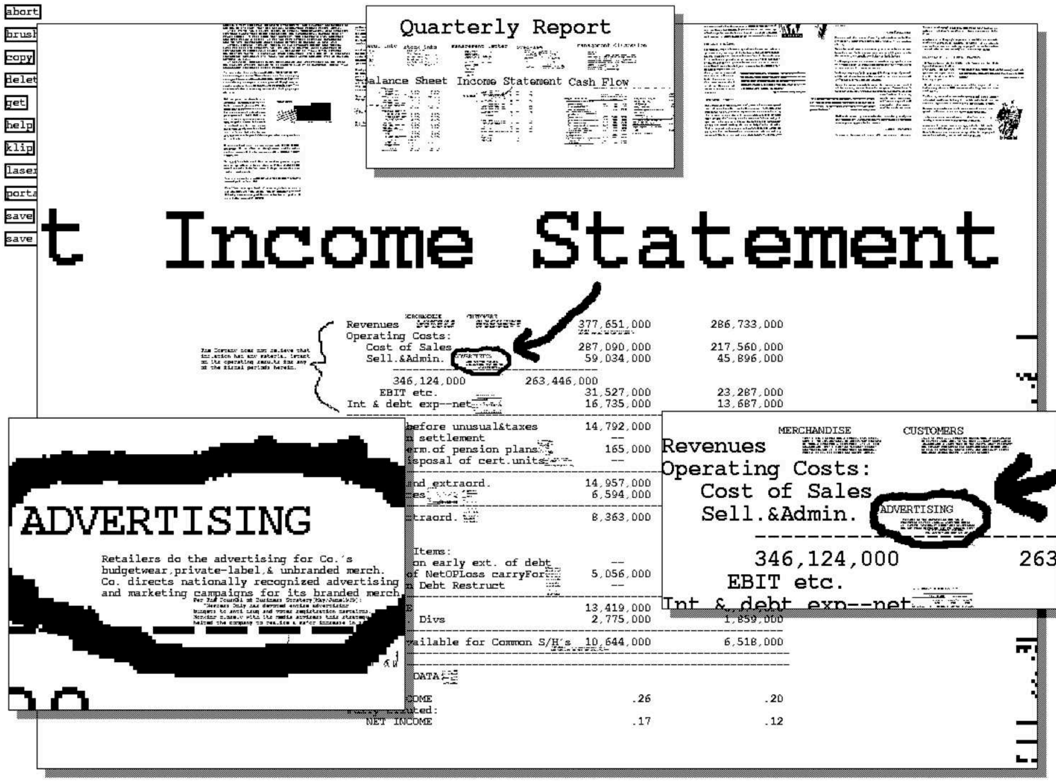
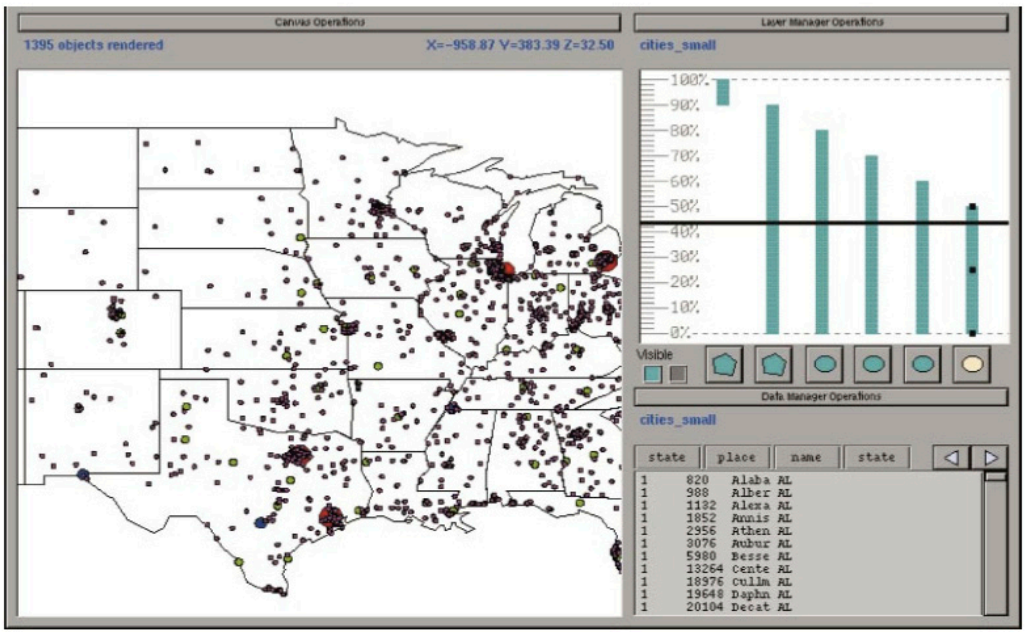


Figure 2-3. The Pad system with 3 portals showing different locations and zoom levels (Perlin et al. 1993).

The first ZUI system, Pad, was a complete information-managing environment with a spatial metaphor (Perlin et al. 1993). Figure 2-3 shows Pad displayed each piece of information in a specific location and scale on one large canvas. Users navigated this information space by panning and zooming. Multiple users could share and build zooming applications upon this information space. Pad used view ports called "portals" to enable the user to see multiple locations at multiple scales at once (see Figure 2-3). A portal acted as a view to a particular part of the canvas, as if a camera were placed over part of the canvas. Portals could also act as filters, showing only certain data, or showing all data in one visual form, such as showing all tabular data as bar charts. Pad was also one of the first systems to use semantic zooming. For example, in a calendar application using Pad, years are initially represented as rectangles containing the numeric year. As the user zooms towards one of these rectangles, it enlarges and reveals inner rectangles for each month. Similarly, the user can zoom into a month to see days. Pad was designed to take advantage of human visual memory, such as how humans remember object locations in the real world.

The Pa3D system, developed at the same time as Pad, was a 3D version of Pad (Bederson 2009) where every polygon was a zoomable surface. The first of many Pad successors was Pad++

(Bederson et al. 1994) which added smooth animation for zooming and search capability. When a user chose a search result, the system zoomed and panned automatically to its position in the space. In addition, many more applications were created for Pad++ including a hypertext browser and a file directory viewer. Pad++ developers used an "informational physics" strategy rather than traditional metaphors such as desktop and window. Moving through the Pad space is analogous to moving through the real world. We move closer to distant information in order to see detail. Pad++ successors include ZUI Java toolkits similar to what was used in SZV's implementation of zooming (Section 3.1.7).



Reprinted from *Journal of Visual Languages and Computing* 12, Woodruff, A., Olston, C., Aiken, A., Chu, M., Ercegovac, V., Lin, M., Spalding, M., and Stonebracker, M.., DataSplash: A Direct Manipulation Environment for Programming Semantic Zoom Visualization of Tabular Data, 551-571, Copyright 2001, with permission from Elsevier.

Figure 2-4. The DataSplash system with the layer manager on the right.

The DataSplash system (Figure 2-4) uses direct manipulation in controlling semantic zooming (Woodruff et al. 2001) to view multiple layers of tabular data. Data are mapped to nested virtual canvases (portals). A layer manager shows the range of camera elevation (zoom) each layer is visible within.

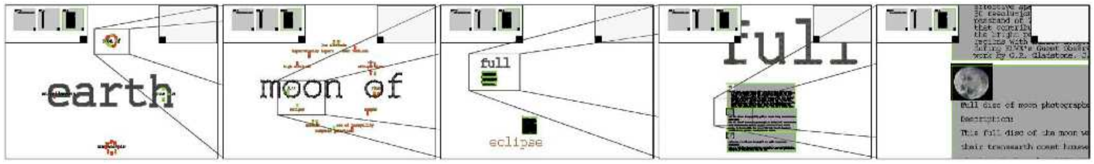


Figure 2-5. Zooming using the NaviQue prototype (Furnas et al. 1998).

The NaviQue prototype was implemented in Pad++ and is a ZUI for information gathering and organization (Furnas et al. 1998). As shown in Figure 2-5, it consists of one integrated environment that can be zoomed or panned and provides multiple views of the visual objects. Through direct manipulation, users can drag, drop, arrange, group, and resize visual objects in the interface. NaviQue allows user creation of sets of objects that can be semantically zoomed, and which support a form of brushing and linking of entities as does SZV (Section 3.3.3).

The Analytic Landscape (Anlan) tool (Rimey et al. 2006) is a ZUI similar to NaviQue that was designed for the spatial organization and storage of visualizations. Visualizations can be moved and resized and placed inside each other on the zoomable canvas. Sets of visualizations called piles can be created. Piles' metadata can be queried and enable lenses to display different views of the piles. Analysts can create edges for relationships between visualizations to organize hypotheses. Like most ZUIs, Anlan's goals were to take advantage of humans' abilities to remember where they put things and to support their thought process by allowing them to spatially layout their arguments. While SZV is limited to working only with documents, spatial arrangement of the documents aid an analyst's thought process. The different views of documents (Section 3.1) and groups of documents (Section 3.4) in SZV are similar to Anlan's lenses in giving analysts more flexibility in interpreting the document collection.

Next, I describe those systems that allow the user to perform a selective zoom on focus regions.

2.3 Focus + Context

Analysts must not only read the key documents in a collection but must also consider the key documents' context – related documents. They first need to determine which documents to read and overviews such as those described in Chapter 2.1 can help. They also need a quick way to drill down to the details of the documents they wish to focus on. To maintain a quick iterative loop between reading documents and navigating to further useful documents, SZV uses a focus + context technique to show simultaneously the focus documents in detail and the context of the surrounding documents as glyphs. Showing all documents in detail would require too much space and would be information overload for analysts. Focus + context techniques differ from detail + overview techniques by smoothly integrating focus within context rather than showing each in separate frames.

The fish-eye view concept developed by Furnas (1986) was one of the first forms of focus + context in computer interfaces. Furnas found that humans tend to have knowledge focused on their "neighbourhood" and then further away from their focus they only know major landmarks. This suggests a visualization showing a focus area in detail with less detail further away would be appropriate for viewing large structures. Furnas defined a "Degree of Interest" (DOI) function for each point or visual item that combines its "a priori" importance with its distance from the current focus point. Given this value for every item, a threshold is used to filter a visualization and only show items above a certain DOI level. Furnas' method allows multiple focal points. SZV and many tools before it allow users to choose multiple focal points so they may do detailed comparison tasks. SZV combines focus regions with the context of a full overview; however, it does not have a DOI function. Instead, analysts fully control each document's detail level. Unless specifically manipulated, documents are fully zoomed out to preserve space for documents the analyst has chosen to view. In addition, SZV does not filter out any documents, so that brushing for related documents (Section 3.3.3) may occur across the entire document set.

Twenty years later, Furnas reflected on how fisheye views and focus + context in general have evolved (2006). He categorized and described techniques based on what they were trying to show, how they were trying to show it, and why they were using a focus + context view. He also discussed the issue of balancing the amount of space used for focus versus context and how there has been little agreement on how to achieve this balance. SZV uses a semantic zoom to show text documents' content at various detail levels. Section 3.1.6 describes how the analyst controls the zoom level of all documents. Thus, SZV is a controlled user-defined version of focus + context, which is not based on geometrical distance from a focus point, and does not involve any distortion of glyphs. SZV handles the balance problem by allowing the user as much screen real estate as they like for the focus documents with remaining documents using the rest of the space. These tradeoffs are compared to related work below.

Recently, Cockburn et al. wrote a review of overview + detail, zooming, and focus + context interfaces that described each technique and strengths and weaknesses found by studies (2008). While overview + detail and zooming separate the focus from the context by space and by time respectively, focus + context techniques integrate them together. The review categorizes techniques based on selective presentation versus distortion, system- versus human-controlled focus + context, and single versus multiple focal points. SZV uses distortion, but only on the glyph locations, not their shape. It is human-controlled since the analyst chooses what to focus on, and it allows multiple focal points. The review described Furnas' work (1986) on developing the fisheye technique and the Sarkar et al. (1992) follow-up work that defined geometric transformations for creating fisheye views of continuous visualizations such as maps or graphs. They explore Cartesian and polar transformations and Lamping et al. (1995) used hyperbolic geometry in a hyperbolic tree visualization. These visualization techniques display the entire data set while changing the space distribution to display the focus larger than the context. SZV also changes the space distribution to give more space to focus documents; however, it does not distort the actual documents since this would render them difficult to read. Later visualizations such as SpaceTree (Plaisant et al. 2002) used the technique of hiding detail (branches in a tree) if they could not fit on the screen. This is a basic form of semantic zooming when detail can be selective hidden or shown.

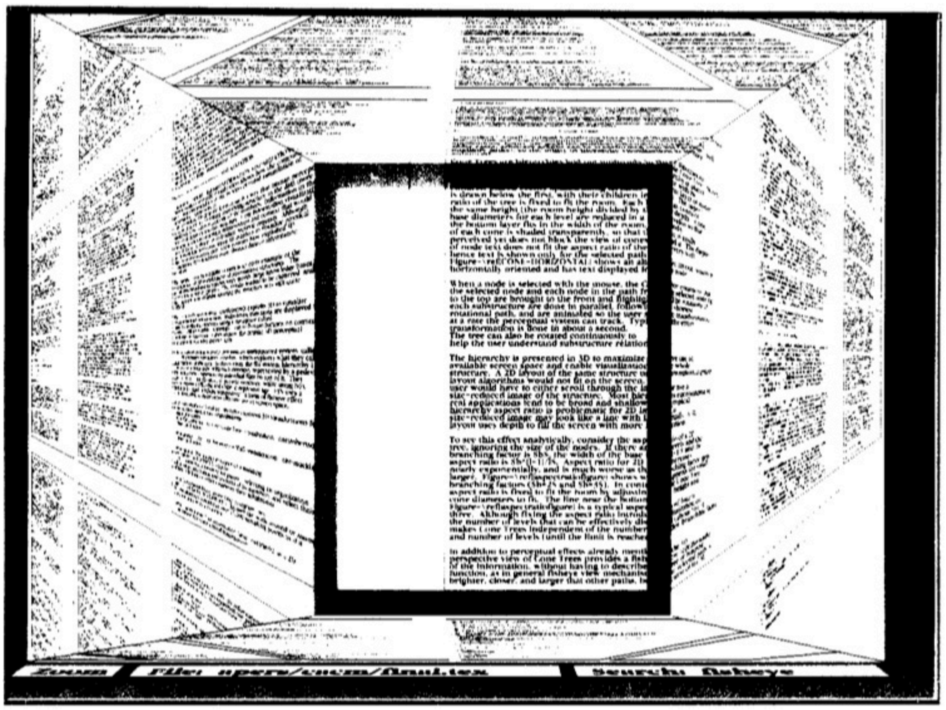


Figure 2-6.   Document Lens (Robertson et al. 1993) .

Cockburn et al.'s review also mentioned many applications of focus + context including the Perspective Wall (Mackinlay 1991), Document Lens (Robertson et al. 1993), TableLens (Rao et al. 1994) and DateLens (Bederson et al. 2004a). These applications work in similar ways, the main difference being the domain and the way in which they display context. All involve viewing the focus area as in a traditional application but reducing the size of the other items in order to make more fit on the screen. TableLens and DateLens are table visualizations that shrink the surrounding cells with distance from the focus row and column. Document Lens and the Perspective Wall use a 3D approach by making the non-focus area shrink perspectively (Figure 2-6). In contrast, SZV shows all non-focus documents in their smallest form. When more space is needed for the focus area, the space between document glyphs shrinks. This technique is not possible in these systems because they are table based. In SZV, the sharp boundary between focus and context maintains space for the entire document set to remain in view (Section 3.2.3). This allows brushing across the entire view for finding related documents. There was a similar reason to show context within these other visualizations. For example, a search in the Document Lens highlights passages in red throughout the context. Even though the text may be greeked (replaced with placeholder wavy lines), the user can still locate and then focus on the red text. Another way to handle this problem with large datasets is to treat the highlighted data as landmarks and guarantee their visibility, as in the TreeJuxtaposer focus + context technique (Munzner et al. 2003).

Baudisch et al. (2002) compared a mixed resolution focus + context display to zooming and overview + detail and found that the focus + context display allowed the subjects to complete tasks significantly faster. They claim that the overview + detail interfaces took longer because the participants needed to switch mentally between the overview and the detailed views. With the focus + context display users did not need as much mental effort to understand the spatial relationships between the focus and context as they did when the two are displaced spatially.

Another study mentioned within the Cockburn et al. (2008) review compared fisheye views to full zoom views (Schaffer et al. 1993, 1996). They designed a fisheye algorithm called the variable zoom (Zuo 1992) for use on hierarchically clustered graphs. Such graphs have nodes grouped into clusters that can be represented either as a single node icon or expanded to show their contents. Thus, this is a semantic zoom within a focus + context technique. With the variable zoom, users can select multiple focal points by expanding multiple clusters at the same time. When a cluster is expanded to show its contents, the other nodes and spaces between them are scaled down independently in height and width to give more space to the expanded node (Figure 2-7).

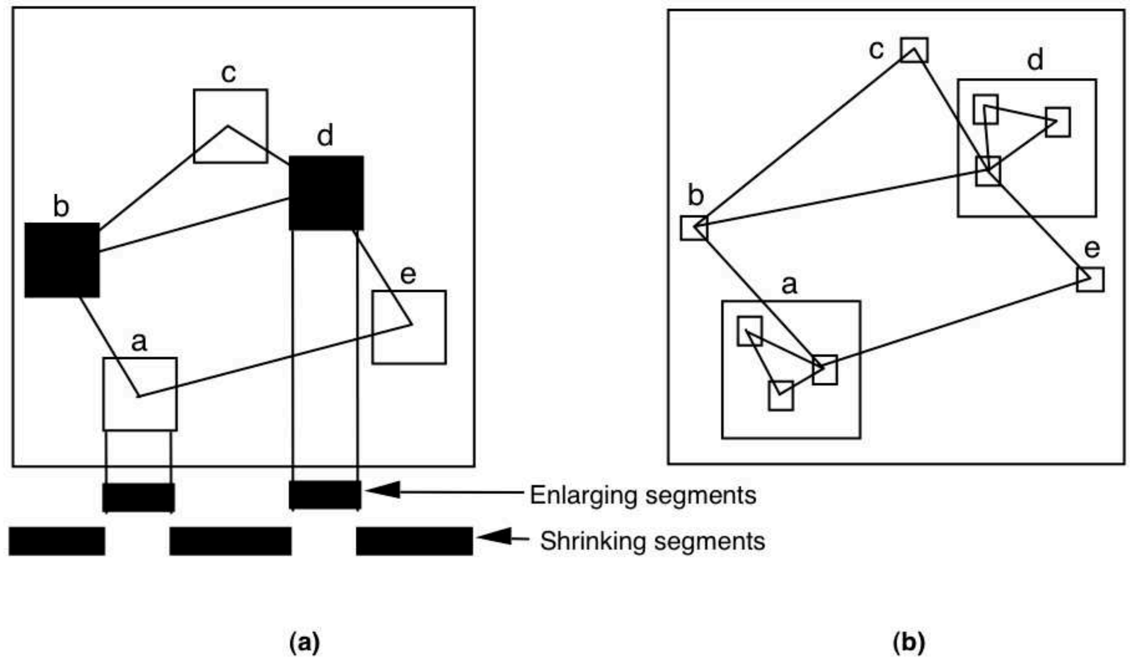


Figure 2-7. The expansion of cluster nodes in the variable zoom (Schaffer et al. 1996).

In the full-zoom condition used in their study a cluster node filled the screen when expanded. The study found participants completed navigation tasks faster and performed significantly less cluster expansions when using the variable zoom. Participants found the variable zoom context much better than having to remember the network. Two experienced control room operators tried the study and found the fisheye technique much less confusing for navigation; however, the full zoom technique was needed to finish tasks. For full details, the amount of space the fisheye technique gave to the focus was not enough. They concluded the space balance between the focus and the context must be better managed, or it needs to be user controlled. In the follow-up work of the Continuous Zoom, described next, and in SZV the user has fine control of the focus items' size.

The variable zoom technique led to the Continuous Zoom algorithm (Dill et al. 1994; Bartram et al. 1995), which is similar to the layout adjustment algorithms developed by Storey et al. (1996) many years later for SHriMP and used by SZV (Section 3.2.3). This algorithm improves upon the previous one by offering smooth transitions when expanding nodes, a DOI component to choose node sizes, and a mechanism for allowing the user to adjust these sizes. Unlike SZV, resizing a node will automatically move and resize the other nodes in the view. In SZV, all other document glyphs are moved but they are not resized since the analyst may want to maintain their current detail level.

The Continuous Zoom (CZ) algorithm was used in two applications: CZWeb (Collaud et al. 1996) and CZTalk (Lam et al. 2005). CZWeb created a map of web pages visited with each page represented as a node and clusters created based on web domain. Edges were drawn between web pages visited sequentially to show the exploration path. A study was done on CZWeb (Fisher et al. 1997) that found it was better than traditional tools (such as bookmarks) for revisiting web pages and understanding website organization. With CZWeb, users can move around page nodes to organize the view based on their mental model of the pages. SZV is similar in that documents are initially placed in a layout based on their similarity, but then analysts can move documents around and organize documents to match their mental model.

CZTalk uses the CZ algorithm to layout messages in a collaborative discussion environment (Lam et al. 2005), visualizing a discussion as an interactive hierarchical graph where the nodes are messages. While the CZTalk interface resulted in somewhat improved task completion time in a preliminary study, users did not like cluster nodes being closed automatically when they became too small. SZV never changes the detail level of any context documents or groups of documents automatically. The trade off is that analysts must manually reduce the size of some documents when they wish to have more space, but SZV aims to make this process as quick as possible.

Reinhard et al. (2007) improved upon the CZ algorithm for their ADORA tool, a plug-in to Eclipse for doing modelling of object-oriented software. They outlined six goals of a fisheye zoom algorithm for their purposes: 1) no overlapping nodes; 2) preserve the original layout as much as possible when zooming; 3) performing zoom operations and then their opposites should return to the original state; 4) allow the deletion, creation, resizing and moving of new nodes while preserving the layout as much as possible; 5) run in real-time; 6) allow multiple focal points. SZV meets all but the first condition. As explained earlier, the context documents (those not being zoomed) do not have their size changed automatically, resulting in potential overlap if screen space is insufficient for all expanded documents.

Another application using a focus + context technique is the Visual Understanding Environment (VUE), a concept map tool for viewing digital library repositories (Afram et al. 2007). The concept map is a graph with nodes showing varying detail levels due to a semantic zoom within a fisheye effect. As the fisheye focuses on nodes, their photos enlarge and more information is displayed. A study was performed comparing the fisheye technique to normal full zooming that had no semantic zoom and in which the context would move offscreen as you zoomed in. The node’s full details appeared in a popup in front of the main view. In the study, participants had to answer questions about the concept map content and speed and accuracy were measured. An extra set of questions took place after working with the interface in order to test participant's memory of the content. They found the full zoom interface significantly faster for the first question set. They claimed this was because the fisheye condition participants were not yet used to the interface. They also found a significantly higher accuracy for the fisheye condition on the recall task question set. They claim these participants did not need to remember what was in the rest of the concept map. Thus, they could concentrate more on remembering the current content. Though fully zoomed out documents in SZV have no labels, several features keep users from having to memorize document locations. These include moving documents to specific spots, grouping documents to create an easier to remember hierarchy, leaving important documents somewhat zoomed in, and brushing and linking for quickly relocating related documents.

Table 2-1. A summary of this chapter's document visualizations.

System	**Content**	**Overview Layout Method**
Lighthouse Leuski et al. 1996, 2001, 2004	Search results.	Spring-based layout places similar documents together.
VIBE Olsen et al. 1993, Ahn et al. 2009	Search results with added user-model based on task notes.	Documents placed relative to points of interest for queries they match and a user-model.
SPIRE & In-Spire Wise et al. 1995, Hetzler et al. 2004	A document collection.	Vectors based on document statistical properties in 2D & 3D
Starlight Risch et al. 1997	Documents, numerical data, images, maps.	Vectors based on document statistical properties in 3D and categorical view of structured data.
Jigsaw Stasko et al. 2008	Document collection with extracted entities.	Categorical view and clustered by query results.
Semantic Zoom View	Document collection with extracted entities.	Force-directed layout places documents with many entities in common closer together.

This chapter has described related work in the three areas of document visualization (Table 2-1, including information retrieval), zooming (Table 2-2), and focus + context (Table 2-3) systems. I have also directly compared SZV to these systems to place it within context before it is described in full detail. Chapter 3 describes the full design of Semantic Zoom View. First, I explain zooming in Section 3.1 and the document overview of SZV in Section 3.2 before explaining how these two combine in a focus + context manner.

Table 2-2. A summary of this chapter's systems using zooming.

System	Content	Description
Pad Perlin et al. 1993	Any digital drawings, text, tables, etc.	Large canvas with multi-scale content navigated by panning and zooming. It included portals and applications could be built upon it.
Pad++ Bederson et al. 1994	Any digital drawings, text, tables, etc.	Added smooth animated zooming and search to Pad.
DataSplash Woodruff et al. 2001	Tabular data and maps.	Tool for creating semantic zooming to show levels of details on a map.
NaviQue Furnas et al. 1998	Any digital drawings, text, tables, etc.	Implemented in Pad, contains groups, and brushing and linking.
Anlan Rimeyet al. 2006	Visualizations created in other tools.	Hierarchy of visualizations organized and linked on zoomable canvas.
Semantic Zoom View	Document collection with extracted entities	Semantic zooming and organization of documents including search, and brushing and linking.

Table 2-3. A summary of this chapter's focus + context techniques.

Technique	Description
Fisheye views Furnas 1986, 2006	Selectively filters a visualization based on a combination of a priori importance and distance from focus item.
Continuous fisheye Sarkar et al. 1992	Extends fisheye to continuous content such as maps or graphs.
Hyperbolic tree Lamping et al. 1995	Uses hyperbolic geometry to visualize a tree graph.
Document Lens Robertson et al. 1993	All pages are shown in one grid and a 3D perspective is used to display the center normally and then stretch the sides into the distance.
Mixed resolution focus + context Rimeyet al. 2006	A detailed area is embedded in a less detailed area by embedding an LCD screen within a projector screen.
Variable and Continuous Zoom Schaffer et al. 1993, 1996, Dill et al. 1994, Bartram et al. 1995	Hierarchical graphs have clusters that can be expanded or shown as a single icon. When these are expanded, compacted, resized or moved all other nodes are adjusted to use the remaining space.
CZWeb Lam et al. 2005	Uses the Continuous Zoom algorithm in a companion program to a web browser to show a hierarchical graph of the webpages visited.
CZTalk Lam et al. 2005	Uses the Continuous Zoom algorithm to show the threads of a collaborative discussion with messages as nodes in a graph.
ADORA Reinhard et al. 2007	Similar to the CZ algorithm, a tool for modelling object oriented software as a hierarchical graph where nodes can be expanded.
Visual Understanding Environment Afram et al. 2007	The user controls a fisheye focus over a concept map displayed as a graph with photos on the nodes.
Semantic Zoom View	Documents are glyphs on a plane that can be semantically zoomed to show various detail levels. As a subset of them is zoomed into, all other documents are moved outward.

3: DESIGN

Semantic Zoom View (SZV) demonstrates the application of a focus + context technique for drilling down to the details directly within a document collection overview. SZV is a unique application, but can be broken down into several aspects similar to the related work of the previous chapter. In this chapter, I describe each element of the technique and the design rationale. First, below I describe how SZV fits into the CZSaw system and the components of its data model. Then I describe the semantic zoom of a document to reveal layers of detail. Next, I explain the overview and how document zooming affects the layout. Following this, I describe the many mechanisms for locating specific documents and entities within the visualization. I end the chapter by explaining SZV features used for organizing a document collection.

SZV is one data view within the CZSaw visual analytics application (Kadivar et al. 2009), so it uses the CZSaw data model and instances of the view are opened within CZSaw. CZSaw's data model consists of text documents which have a name, a body text, metadata such as the date and a set of extracted entities. CZSaw imports data that already has the entities extracted from it, which is much different than In-Spire which does not involve entities. The entities in CZSaw are necessary for the overview layout algorithm (Section 3.2.1) and brushing and linking (Secton 3.3.3). After importing data into CZSaw, analysts can perform a search and display the results in a data view. Upon discovering a subset of documents they wish to see in a different perspective, analysts can drag and drop the documents from one data view to another. The analyst may also add the entire document collection to SZV. Thus, an analyst can use SZV to analyze either an entire document collection or only the portions of it that interest them.

3.1 Zooming

Many visual analytics tools, such as In-Spire (Hetzler et al. 2004), focus on providing an overview of the data so analysts see the big picture and instantly spot trends. To maintain the overview while allowing analysts to view object details, the simplest course is to display the details in a separate space. Unfortunately, there are disadvantages to this mainstream method. It reduces the amount of space available to the overview, especially if analysts wish to compare multiple detailed views. It also requires a mechanism for determining which objects from the overview are being viewed in detail. Analysts must mentally integrate this detailed information with the context surrounding its glyph in the overview. SZV uses zooming to access details directly within the main view, but to maintain the surrounding document collection overview, zooming is only applied to a small subset of the documents chosen by the analyst. Section 3.2.3 explains how zooming affects the document overview layout. Further, semantic zooming of documents enables efficient space usage while providing multiple perspectives on a document, each tailored to a different purpose. In this section, I illustrate the different semantic zoom levels and discuss the interaction and implementation of the zoom.

To achieve a smooth zooming of a document, each document is represented by a rectangle that smoothly changes size. The semantic zoom *level* of a document determines the document's detail level shown within the rectangle. There are five semantic zoom levels, each with slightly more detail than the preceding one. These are shown in Figure 3-1 and described below.

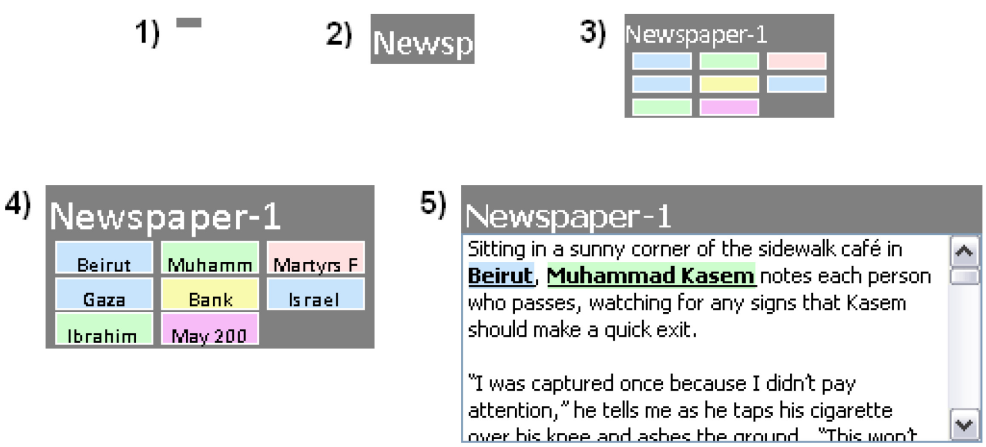


Figure 3-1. The semantic zoom levels of a document from a simple glyph to the document full-text.

3.1.1 Level 1: Simple Glyph

The first zoom level provides a small representation of a document so that a compact overview of a large document collection can fit onscreen. This initial representation is a 5 x 13 pixel rectangle. Using smaller rectangles would provide space for more rectangles, but would make documents more difficult to select with the cursor. Smaller glyphs would also make it more difficult to identify highlighted documents with a quick glance during brushing or a search (see Section 3.3). The performance of this glyph size and feedback from participants in a study are described in Chapter 4. Note that in this representation, documents are only different from one another by colour (if highlighted) and x and y coordinates but as explained in this chapter, these are enough dimensions to enable overviews, searching and brushing.

3.1.2 Level 2: Labelled Glyph

The second zoom level displays the name of the document and if the analyst has already read it, seeing the name may be enough for them to decide if they need to zoom any further. This level is reached when a glyph expands to a height of 14 pixels, large enough to hold some text at a size of 10 pt. The document name, or as much of it as will fit, appears within the rectangle (Figure 3-1). In addition to zooming, a mouseover shows its full name and date.

3.1.3 Level 3: Entity Glyphs

The next semantic zoom level provides the analyst a quick overview of the number and distribution of the entity types within the document so that, for example, they can instantly tell whether any people are present and if so how many. Thus, if the analyst seeks a specific person, glancing over many documents zoomed to this level easily suggests which documents to examine further. This zoom level displays entities as small rectangles in a grid, colour coded according to type: person, location, date, organization, etc. (Figure 3-1). To facilitate the analyst's ability to spot entities, they may order entities by their appearance in the text or alphabetically, with the option to group by type. Also, documents with more entities are assigned a larger display rectangle, making them easily distinguishable from those with fewer. The entity grid fills the document rectangle and since all entity rectangles are the same size, the grid is larger for a document with more entities, and thus the document rectangle is larger. The number of rows and columns of the entity grid are made as equal as possible with a partial last row of entities if necessary. The size allocated to the whole document is then determined by the size of a grid of these entities plus a space above the grid for the name of the document. Note that at semantic zoom level 1 all documents are the same size. The SZV uses a smooth interpolation to expand the larger documents (with more entities) faster when zooming in. The aspect ratio of each document is also adjusted as it is zoomed, as an interpolation from the standard zoomed out glyph dimensions, to an aspect ratio that will fit the document name and entity grid.

3.1.4 Level 4: Labelled Entity Glyphs

At the next zoom level, analysts can scan entity names to find specific entities or discover interesting ones. This level occurs when the document rectangle has increased in size enough to have space to display some or all of the entity's value. As seen in Figure 3-1, entity names may not fit within their rectangles, in which case the analyst can see an entity's full name on mouseover. In addition, clicking an entity rectangle triggers the highlighting of all other documents that contain this entity throughout the view. This entity brushing is described in Section 3.3.3 below. Brushed entities have a thicker border. The set of entities supply a summary of a document's contents, as they are the keywords related to who, when, and where. Based on what they see, analysts can then zoom to the full text of level 5 for more details such as what, why, and how.

3.1.5 Level 5: Document Full Text

Figure 3-1 shows the final semantic zoom level displays a scrollable text panel with the documents' body text. Further zooming of the document simply expands the rectangle to allow analysts to see more of the text at once. Entities are highlighted within the text. Similar to level 4, the highlighting is colour coded by entity type. Analysts can perform brushing at this level to highlight other documents containing an entity. Brushed entities are underlined in the body text.

3.1.6 Interaction

SZV provides a smooth interpolation between the five levels of semantic zoom described above. The mouse scroll wheel is used for zooming to offer fine control over the zoom level similar to many online map websites, such as Google Maps (2011). Although both SZV and zoomable geographic maps involve semantic zooming, the metaphors used differ, and thus I reversed the ”meaning” of the direction of motion of the mouse scroll wheel from that typically used for maps. In Google Maps, the user considers them self moving closer to the world as they increase the scale of the map, whereas in SZV the user is moving a subset of objects, documents, closer to them in order to see them in detail. To match SZV's metaphor, moving the scroll wheel towards the user (down) pulls the document(s) in (zooms in) and moving the scroll wheel away from the user (up) pushes the document(s) away (zooms out). This differs from Google Maps, where moving the scroll wheel towards the user (down) moves the user down towards the world (lowers the user's altitude over the world), while moving the scroll wheel away from the user (up) moves the user away from the world. Both SZV and Google Maps chose a mapping that they felt was a good match for their user's model of the movement being made; however, these mappings happen to be the reverse of each other.

Zooming into one or more documents causes the other documents to move away (towards the view boundaries) to supply the space needed, as described in Section 3.2.3 below.

The analyst may choose a single interesting document and zoom into it to get detail or they may zoom into multiple documents simultaneously (selected by any query technique in Section 3.3). They specify one or more documents to zoom and all other documents remain at the same zoom level. To zoom a single document, it is not necessary to first select the document; the analyst moves the cursor over the document rectangle and then uses the scroll wheel. To zoom more than one document simultaneously, an analyst can quickly zoom all the documents from one search or brushing of an entity that will all be selected in one colour, as described in Section 3.3. By holding down the shift key on the keyboard the zoom in or out action will apply to all documents selected in the same colour as the document under the cursor. Documents zoomed together do not need to start at the same zoom level; however, documents have a maximum zoom level so zooming in all the way will result in all documents reaching their maximum size.

As a document is zoomed in, it expands outward from its centroid so the cursor will remain within the document boundary. To avoid 'overloading' the scroll wheel, the analyst must use the scroll bar to scroll the full text[1]. Note that for large documents, since there is a maximum size a document can be zoomed to, it may not be possible to see all the text at once.

A multiple document zoom results in the documents moving outward from their average centroid[2]. The next section describes the implementation of the zoom and the toolkit it uses.

---

[1] Although computer users are used to scrolling text with a mouse scroll wheel, this mapping decision was made to enable analysts to not get stuck at the full text zoom level. Originally, with the scroll wheel causing text scrolling, users would zoom into a document with the mouse over the centre and then instead of zooming out the wheel would cause the text to scroll. In order to zoom back out, users would have to move the cursor to the document title area and then zoom back out. This frustrated early users and broke the consistency of the mapping, so it was changed. Users that wish to read more text of a document can either use the scroll bar or zoom the document further to expand the glyph further. Another possibility, would be to hold the *shift* key in order to overload the scroll wheel and use it for both, but than participants would have to remember this mapping.

[2] The document under the cursor may quickly move away leaving the cursor over white space. Originally, this would cause the zooming to halt and annoyed users since in order to zoom, they would have to chase one of the documents with the mouse. I changed the functionality so the documents being zoomed could continue to be zoomed in and out without the cursor over any of them, as long as there was no pause longer than a few seconds. By having a few users use the system during the early stages of development, I was able to improve it iteratively. In the future, the zoom could be centered on the document under the mouse to improve usability but this would require a more complex algorithm than described in Section 3.2.3.

3.1.7 Implementation

The zooming interaction and user interface were the first aspects of SZV to be developed. This section is a high level description of the semantic zoom implementation. Appendix A describes the actual implementation of an early prototype (Dunsmuir 2009).

SZV uses a Java toolkit called the Zoomable Visual Transformation Machine (ZVTM) (Pietriga 2005), similar to Pad++ successors. Jazz, the first successor to Pad++, is an open source Java 2D scene graph toolkit (Bederson et al. 2000). Java developers use the toolkit to make their own ZUI applications with multiple surfaces and cameras similar to portals, which are rendered to the screen within Java Swing components. Components including text, images, shapes, hyperlink nodes, and Swing components can be placed on the virtual surfaces. Multiple representations of the same Jazz components can be displayed and tied to specific scale ranges to implement semantic zooming. Cameras can be setup to point to different virtual surfaces and for each camera, layers within the scene can be selectively hidden or modified. "Sticky" visual objects can appear in a layer above the main display similar to a heads-up-display (HUD). The Jazz toolkit was designed to distribute the functionality across many different classes for objects in a scene rather than having one "monolithic" class hierarchy. While this makes programming with it more powerful, it also makes it harder to learn to use. This may be the motivation for the development of Piccolo, Jazz's successor, a much more traditional "monolithic" design similar to Java's Swing. Bederson et al. (2004b) compared Jazz and Piccolo and found that Jazz is better to use when there is much toolkit customization (as in creating another toolkit from it) and Piccolo is better to use when there is not.

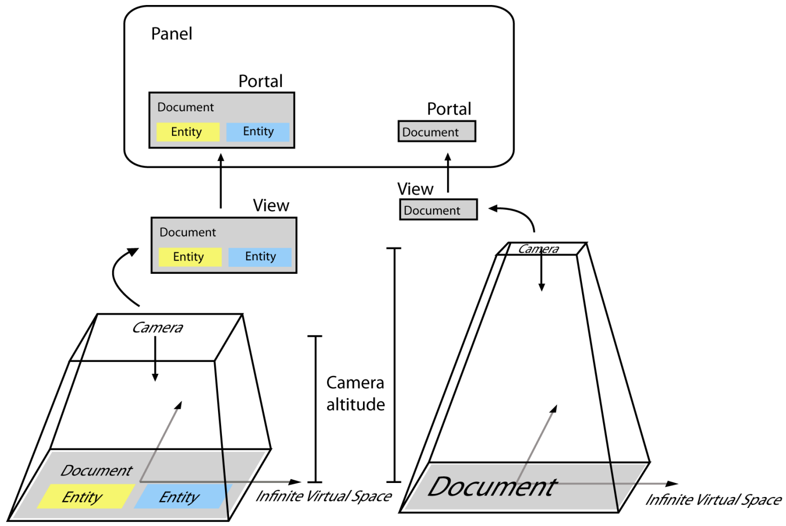


Figure 3-2. How ZVTM is used by the SZV. Changes in camera altitude change glyphs' onscreen sizes.

The toolkit used by SZV, ZVTM (Pietriga 2005) claims to have a more human-centric model and to be more lightweight than Jazz or Piccolo. For example, a program's own internal objects can make up the scene graph rather than managing a separate scene graph.

Much of the model is the same as the above Jazz description. In ZVTM, you can create virtual spaces and place glyphs on them such as text or shapes, but not Java Swing components. Thus, the scroll panes displaying full text for any documents are added to a layer above the ZVTM panel. For the rest of the visualization and zooming of a document, Figure 3-2 shows how each virtual space (one per document) is mapped to a portal, all of which are displayed in one onscreen panel. In SZV, each document is in its own virtual space with its own camera so it may be zoomed independently of all the other documents. By displaying each camera within a portal, all documents are shown onscreen at once in one panel containing all portals. To zoom into a document's glyphs, the camera altitude is adjusted using ZVTM. In ZVTM, a camera represents a viewpoint above the virtual canvas and the camera position controls the mapping from virtual canvas glyphs to the onscreen appearance of the glyphs. A camera at an altitude of 0 displays the glyphs onscreen with the same dimensions they have on the virtual canvas. The higher the altitude, the smaller the glyphs appear onscreen. In addition, a ZVTM camera can have a negative altitude that results in a magnification of the virtual canvas. In SZV, the semantic zoom level is also a function of the altitude, and glyphs are hidden or shown dependent upon the current level. As mentioned in 3.1.3, the sizes and aspect ratio of glyphs are changed to perform a smooth interpolation from the standard small document rectangle to a grid of entities. In addition, the portal bounds must be adjusted to crop the camera image to include exactly the document's glyphs. As explained in 3.4, documents may be moved around the screen. In SZV, this is done by moving the portals with the ZVTM panel. The next section describes the overview provided by SZV and how zooming affects the layout.

3.2 Overview

SZV provides analysts with a document collection overview as a starting point for their investigation. Clusters of similar documents (formed as described below) are displayed with the most frequent entities within them placed as a label at the cluster centroid (Figure 3-3). This provides analysts with the key entities from the document collection as well as the distribution of documents about these entities. Analysts may easily generate new layouts using only certain entity types, for example only people and organizations. From the overview, analysts may quickly access all the other SZV features. They may drill down into documents' contents as described in the previous section. To locate specific documents, analysts may brush entities or perform searches as described in the next section. Finally, to create and refine a visualization matching their mental model, the analyst may spatially organize documents and group them as described in Section 3.4. This section focuses on the flexible mechanism for generating a layout of the documents to provide an overview.

3.2.1 Layout Algorithm

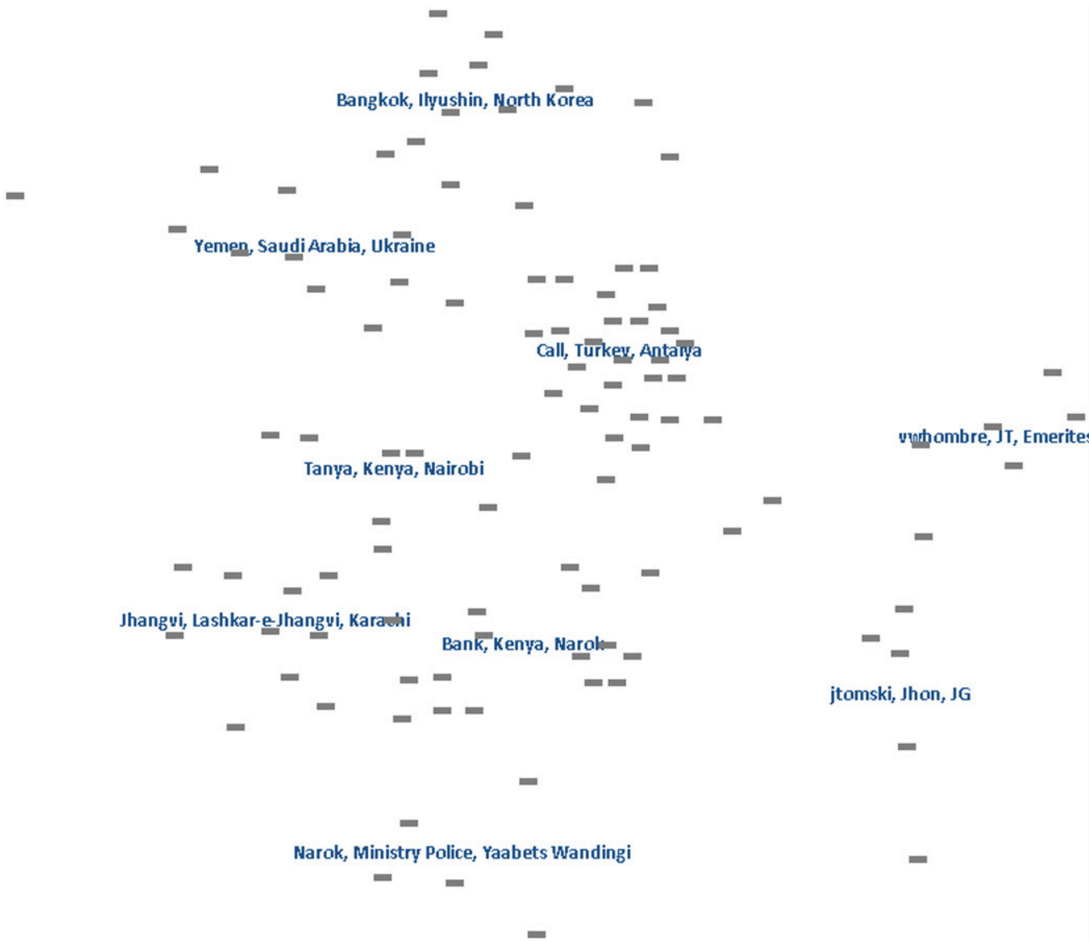


Figure 3-3. An example of a layout in Semantic Zoom View.

Figure 3-3 shows SZV's cluster layout. Documents with more entities in common are placed closer together so that an analyst may find similar documents around those they have already investigated. As mentioned in Section 2.1.2, SZV resembles In-Spire's Galaxy view (Hetzler et al. 2004) more than any other related research project; however, the layout algorithm used in SZV is far simpler as it is based on entities rather than all the document's words. It is less scalable, but more flexible.

In-Spire creates a vector for each document using all words in the documents minus a standard set of stop-words and any user-defined stop words. These vectors define locations in a high dimensional space for each document that are then projected into two dimensions using MDS techniques. Analysts have used In-Spire to study collections of up to 70,000 documents.

In contrast, SZV's layout algorithm only considers *entities*[3] within each document and starts by placing each document at a random location onscreen. It then computes weights for edges between the documents based on the percentage of their entities they have in common. Figure 3-4 shows an example of a weighted edge calculation and the procedure is described below. The current CZSaw data model does not contain information on how many times an entity occurs in a document, and so as not to break from this model, SZV only uses the binary fact of an entity appearing or not in a document. All entities within a document have equal weight, calculated as a percent of the document, so if a document contains only four entities, each entity is 25% of the document. When comparing two documents, the common entities will have a percent value for each document. The minimum of these two percent values is used and added to the minimum percent value for all common entities to compute the weight of the edge between the two documents (as demonstrated in Figure 3-4). Thus, the weight edge range is from 0 (no common entities) to 100 (identical entity sets). If the calculation evaluates to 0, then no edge is created.

Document 1	Percentage
Entity 1	33.30%
Entity 2	33.30%
Entity 3	33.30%

Document 2	Percentage
Entity 2	25%
Entity 3	25%
Entity 4	25%
Entity 5	25%

Entities in common are Entity 2 and Entity 3.

Edge Weight = E2 MIN % + E3 MIN % = MIN(33.3, 25) + MIN(33.3, 25) = 50

Figure 3-4. Computing a weighted edge between two documents based on common entities.

A force directed layout then uses the set of all weighted edges to pull and push documents from their initial random locations based on the edges between them. The algorithm used is a modified version of the Java Universal Network/Graph Framework 2.0 (JUNG) FRLayout2 class (Nelson et al. 2009), based on the Fruchterman-Reingold force-directed graph layout algorithm (1991). The algorithm iteratively moves each document glyph a small amount by applying repulsive forces between it and any document within a specific Euclidean distance (measured every iteration) and attractive forces between it and all neighbour documents (connected to it by an edge). Documents closer together assert a stronger repulsive force, and

[3] SZV's algorithm depends entirely on prior entity extraction since the layout will not run without entities. Thus, the layout algorithm's usefulness depends upon the accuracy of the entity extraction.

documents connected by a heavily weighted edge assert a stronger attractive force. The algorithm ends after 700 iterations or 10 seconds, whichever comes first. These are simple measures that could be replaced by instead stopping when the sum of the edge lengths or the net movement is minimized; however, it may be better to completely replace this algorithm with a faster one (see Section 6.1). How long an iteration takes depends upon the number of documents and edges. In the worst case, an iteration runs in $O(n^2)$ time where n is the number of documents, but this is only if all documents share at least one entity with all others. This algorithm is much slower than the In-Spire algorithm, taking the entire 10 seconds when run with thousands of documents. On a 100-document dataset, the layout is calculated in roughly 600ms.

Once the force-directed algorithm is complete, the overall layout is scaled and translated in x and y dimensions in order to fit the view dimensions (Figure 3-3). The force-directed layout results in weighted edges pulling subsets of related documents together into what the human eye detects as clusters. The layout algorithm is continuous though, it has no concept of clusters. Thus, labelling clusters with the most frequent entities requires a separate algorithm to determine which documents are in each cluster. In the future, this algorithm should use the underlying edges and graph clustering methods so that the cluster labelling algorithm matches the underlying model.

Currently, SZV uses Ward's hierarchical clustering method (1963), which only uses the final (post-layout) onscreen location of each document to determine cluster membership. It starts with all documents as their own clusters and repeatedly merges two clusters. At each iteration, the clusters to merge are chosen based on minimizing the Error Sum of Squares (ESS), which for SZV is the sum of the Euclidean distances between each document and their cluster's centroid. This algorithm is run until all documents are part of one cluster with the ESS increasing throughout. This creates a hierarchy of possible cluster schemes from which the desired number of clusters must be chosen. SZV chooses the number of clusters before the largest single increase of the ESS with a restriction of at least 5 clusters, since large jumps in ESS usually occur near the end of the algorithm.

Once document clusters are defined, SZV determines the three most frequent entities in each and labels the clusters with them in a manner similar to that of In-Spire. Labels provide the analyst with a quick preview of a cluster's contents to aid them in deciding which documents to investigate. The choice for how many clusters to use in the algorithm is not a clear one since the analyst could want more fine cluster labels depending on their task. A possible future addition to the layout controls described below is a slider that would allow the analyst to interactively change the desired number of clusters and view the labels.

3.2.2 User Interaction

SZV provides options the analyst can use to adjust the layout parameters. Figure 3-5 shows the options change entity types considered, initial placement of documents, and the set of documents to which to apply the layout.

In-Spire (Hetzler et al. 2004) allows an analyst to manually add stop-words and remove documents from the view, and then recompute the layout. In SZV, analysts can choose which entity types to include. Entity types not chosen are ignored when computing the weighted edges or determining cluster labels. This flexibility allows analysts to focus on entity types of interest, removing types that may be unnecessarily creating clusters. For example, if half the documents mention British Columbia and related places and the other half mention California and related places then using location entities may divide the layout into two separate clusters. This may hide other relationships. An analyst may already know about this division and wish to focus on connections among individuals instead. This option allows them to remove all location entities from consideration and instead choose just people and organizations to achieve a more useful layout.

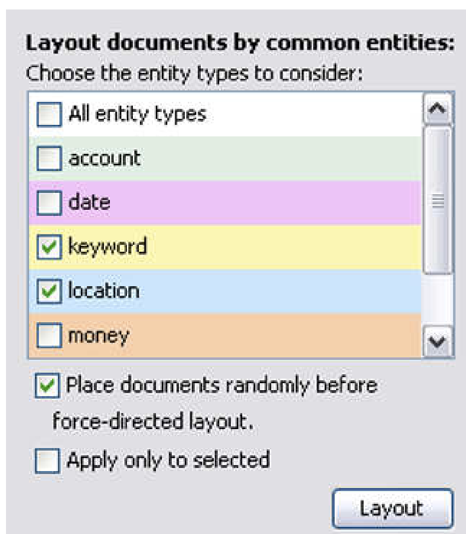


Figure 3-5. The available layout options.

When a new layout is determined, document movement is smoothly animated. When the “Place documents randomly” option is chosen, the current document layout is not considered in the algorithm and thus it will not bias the force-directed layout. If the analyst does not used this option, the current layout is used as a starting point for the algorithm, which will reduce the overall movement documents make to the new layout. Animation makes it possible for an analyst to follow the movement of a small subset of documents, so they will not have to scan for them afterward.

The last option, “Apply only to selected”, allows the analyst to run a layout on a subset of the documents. This allows an analyst to find finer clusters within a cluster. For example, an analyst with the two clusters British Columbia and California, may want to see the connections between people within each of these locations separately. When the layout is run on a selected cluster of documents, the resulting layout moves the documents only within the space in which they originally filled. One natural subset of documents to apply a layout to is all the documents within a group. See Section 3.4 below for a description of group creation that keeps a persistent document subset that can be manipulated and visualized as a unit.

3.2.3 Layout Adjustments caused by Zoom

The cluster layout provides a document collection overview that can then be interactively modified and organized by the user as described in Section 3.4. First, however, the analyst may wish to maintain the layout while they perform queries, drill down into subsets of documents to read them, and brush to find related documents. This subsection describes the layout adjustments made in order to keep the context visible while documents are zoomed.

As a document is zoomed, it occupies more space, but should not cover the nearby documents. Thus, they need to be moved away as the document expands. SZV makes use of a layout adjustment algorithm used by the SHriMP visualization technique (Storey et al. 1996) to move the other documents away while maintaining the same relative positions between documents. The aim is to maintain a layout that minimizes the changes to the clusters while providing more space for the focus document(s). The layout should remain consistent with the mental model the analyst may have formed while working with it. In the review of focus + context, overview + detail and zooming techniques, Cockburn et al. (2008) note that distortion oriented techniques may impair the user’s ability to make spatial judgements and cause a misinterpretation of the underlying data. SHriMP’s layout adjustment algorithm minimizes the distortion of the relative positioning of the documents (Storey et al. 1996). Note that since the view axes do not match any properties of the underlying data, only relative positions represent the underlying data.

The Simple Hierarchical Multi-Perspective (SHriMP) visualization uses a graph layout adjustment algorithm that has several variants and SZV uses the last variant described by Storey et al. (1996). This variant is designed to preserve proximities between objects in the layout, which helps maintain clusters. SHriMP uses the algorithm with a fish-eye view of nested graphs; however, the technique can apply to any visual objects in a layout. Figure 3-6 shows before and after the zooming of a document.

Figure 3-6. Left: A cluster layout before any zooming with the document to zoom circled. Right: The layout with the document zoomed in.

The layout adjustment algorithm moves each document in the layout along the line going through its center and the center of the expanding or shrinking document. Each document is moved away from the focus document if it is expanding or towards the focus document if it is shrinking. The distance along its line that each document moves is equal to the distance the document expanded or shrunk along the line. This is the distance along the line that the boundary of the document travelled as illustrated by Figure 3-7, which shows an expanding (bottom) document causing a smaller (top) document to move away.

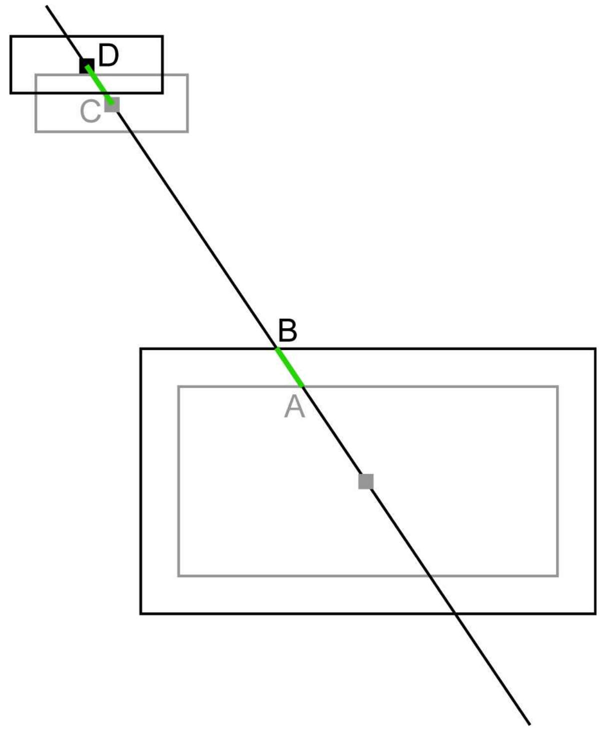


Figure 3-7. The layout adjustment algorithm for an expanding document. The distance between A and B (the green line segment) shows the distance the document's top boundary moved in this direction. The distance from C to D (also in green) is equal to the distance from A to B and this is the distance the smaller document moves away along the same line.

This algorithm preserves relative document proximities and enables the layout to revert to its original form when a document is zoomed back out. This variant of the layout adjustment algorithm has an optional last step, which SZV only partially follows. The last step is to scale down the expanding layout so it fills the same overall space (the dimensions of the view). SZV scales down the locations of all the documents to keep them within the original layout dimensions, but it does not scale the actual dimensions of individual document glyphs. We avoid automatic changes in the zoom level of non-focus documents because the analyst may have zoomed them to a specific detail level that should not be automatically changed. Thus without the automatic change in size, some documents may end up overlapping each other when the locations are scaled back. This should only happen on a small screen when a large number of documents are being zoomed. This description has only described the process for zooming a single document.

When multiple documents are zoomed simultaneously, the algorithm is simply run on each document separately resulting in a net movement for each document based on the zooming of all documents. This layout adjustment algorithm runs at interactive speeds for hundreds of documents.

3.3 Query

Semantic Zoom View provides multiple methods for an analyst to query the document collection. Here 'query' means any direct manipulation or method of using a control panel to locate documents meeting certain conditions without having to investigate every document serially. Query features are: highlight by date, search, and brush and link entities functions described below. Documents that match a query are highlighted within the view, no matter which semantic zoom level they currently display. Before describing the means of performing queries, I will describe the use of selection for highlighting documents within the view.

When documents are selected manually or through a query, they are highlighted in one of the multiple colours offered by SZV. Using multiple highlight colours, analysts can keep track of multiple subsets of documents at once in a less permanent manner than groups (described in the next section). The colours used were chosen to be easily differentiated from each other and quickly spotted among non-highlighted documents. For each query method, there is a method for first specifying which colour to highlight the results in. By using two different colours, the results of two queries can be compared, such as how results of each query are distributed across clusters. In-Spire contains a similar feature for specifying a colour for the query results (Hetzler et al. 2004); however, there are key differences between highlighting in In-Spire and in SZV. In-Spire allows a new colour for each search for an unlimited number of searches while SZV uses only seven different highlight colours. This makes In-Spire more flexible because analysts can keep track of more search results, but it can quickly become difficult to tell colours apart when there are more than seven. SZV ensures the analyst does not have this problem by restricting them to a carefully chosen set of seven colours that not only differ enough from each other, but also from the white background and the grey colour of unselected documents. To keep track of a subset of documents more permanently, a group can be created (see Section 3.4 below), and unlike colours, the number of groups is not limited.

SZV also treats overlapping query result sets differently than In-Spire, which uses a single colour to mark all documents that match more than one query. For example, purple could be used as the colour that marks all documents belonging to two or more coloured subsets of documents. This displays which documents represent overlaps, but not which colours are involved in the overlaps. SZV solves this problem differently by remembering a stack of highlighting colours for each document but only showing the top (most recent) colour. For example, assume a document is highlighted in red and then in green. If the green highlighting is removed, then the document will revert to being red. Thus, there is no visual marking of an overlap between groups; however, an analyst can always see all documents resulting from the last query including any overlap with previous groups, unlike In-Spire. In order to specifically see an overlap, an analyst can remove one set's highlighting and note which documents change to another colour instead of grey. With a large highlighted document set this will not be easy. An intuitive method of showing many set overlaps while maintaining a separate layout of objects is still an unresolved issue among information visualization techniques. To see overlap in SZV, analysts can combine highlighting with groups or manual layout of documents, both of which are described in Section 3.4.

Besides the query mechanisms described next, an analyst can also manually select or deselect documents using the cursor to highlight the document collection in multiple colours. All documents in the same colour can be moved together as described in Section 3.4 below. After choosing the current active colour, documents can be selected by clicking them or rubber banding (drawing a rectangle around them) and the *shift* or *ctrl* keys can be used to add to or remove from the currently selected set. The use of clicking, the keyboard keys, and rubber banding are well-recognized protocols for data visualizations and lists within most option panels of interactive computer programs.

3.3.1 Highlight by Date

Many document collections contain metadata such as the creation date of each document. SZV has a Highlight by Date feature shown in Figure 3-8.

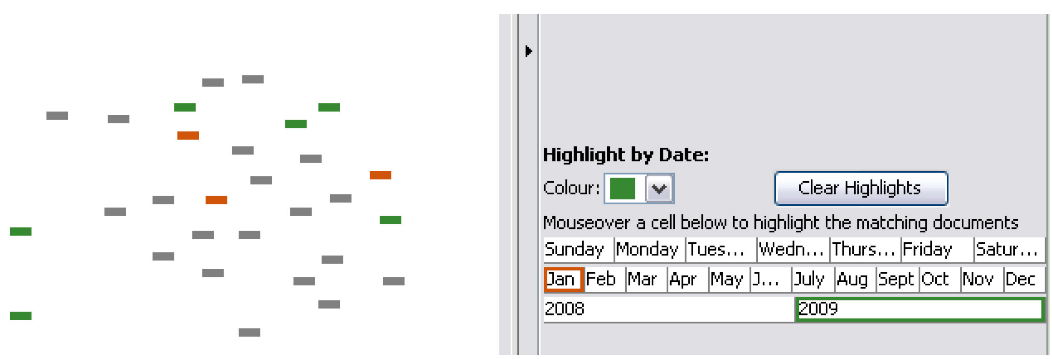


Figure 3-8. The Highlight by Date feature. View this image in colour to see the highlighting.

The Highlight by Date feature provides analysts with a quick method to see which time periods document clusters are from and discover temporal patterns present in the document collection. The interface is a table of days of the week, months and years that an analyst can mouseover to instantly see all documents with a date matching the given range. The days of the week cells may allow analysts to uncover weekly patterns, such as a cluster containing only documents created on the weekend. By clicking on a table cell, the analyst can assign the highlight colour to that date range and then mouseover other cells to add to the set of highlighted documents in the current colour. By choosing a different colour, the analyst can highlight the document collection in multiple colours based on their dates. This allows them to compare multiple time ranges to see the distribution of clusters over time. The table of dates only includes year cells for those years included in the document collection. The document collection used in Figure 3-8 only contains documents from 2008 and 2009.

The original SPIRE introduced a time-slicer feature that allowed analysts to filter the document collection based on document dates (Wise et al. 1995). Thus, it showed time change using actual time instead of colour. While this is a more natural way of displaying temporal changes, it makes it harder to compare many time periods at once since none of them can be simultaneously shown. Thus, the user may have to keep switching back and forth between time periods to compare them. SZV's Highlight by Date feature is a much less direct mapping, but more flexible since it shows multiple time ranges simultaneously.

3.3.2 Search

SZV provides a search feature for finding documents that contain entities or text that match a search string and meet conditions the analyst chooses. Figure 3-9 displays the search panel.

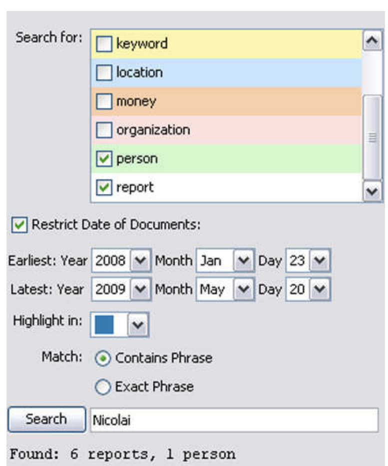


Figure 3-9. The Search options in Semantic Zoom View.

The analyst can choose to look only for documents with full-text or an entity that exactly matches the search string, or they can look for entities or text that just need to contain the string.

In addition, the analyst may restrict a search by document date by selecting a date range. The analyst can choose in which colour to highlight the results and by so choosing colours; they do not lose the highlighting of the previous search. Thus, they can compare multiple searches. When a search is performed, a message at the bottom of the search panel displays the number of documents and contained entities returned by the search. In Figure 3-9, 6 reports (documents) were found which contain 1 unique person entity. All documents found are highlighted in the view and all matching entities within these documents are also highlighted. This entity highlighting is only seen when the documents are displayed in semantic zoom levels 3, 4 and 5 (Section 3.1). Highlighted entities have thicker borders in a document's entity grid (Figure 3-10) and are underlined in its text.

Unlike search engines, the search function in SZV (and In-Spire) highlights the relevant search results within a layout of the entire document collection being searched. In search engines, only the search results are displayed and so it is not easy to understand their context. One could argue that you could not display the entire corpus over which a search engine is searching since this would be much too large a set. As described at the start of this chapter, an analyst using CZSaw can display only a subset of a document collection in SZV. Analysts must be aware that searches performed inside SZV are only searching across the documents already inside SZV. One way to search a large document collection in SZV without displaying all of them simultaneously is to create groups and close some groups to hide their contents. Groups are described in Section 3.4.

3.3.3 Brushing Entities

Brushing of entities allows an analyst to find all other documents containing a given entity. Unlike the other query methods, brushing takes place directly within the view by clicking on an entity. In the example in Figure 3-10, brushing (clicking on) the person entity Boonmee results in all documents containing the entity (Boonmee) being highlighted in the colour used. Since they are all highlighted together, they can then easily all be zoomed at once with the scroll wheel. The entity brushed is also selected within each document, which is displayed by a thicker border around the entity (Figure 3-10) and underlining the entity in the full text, the same way entities are highlighted from a search. Just as with the Highlight by Date feature, multiple brushed entities can be shown at the same time in the same colour or in different colours.

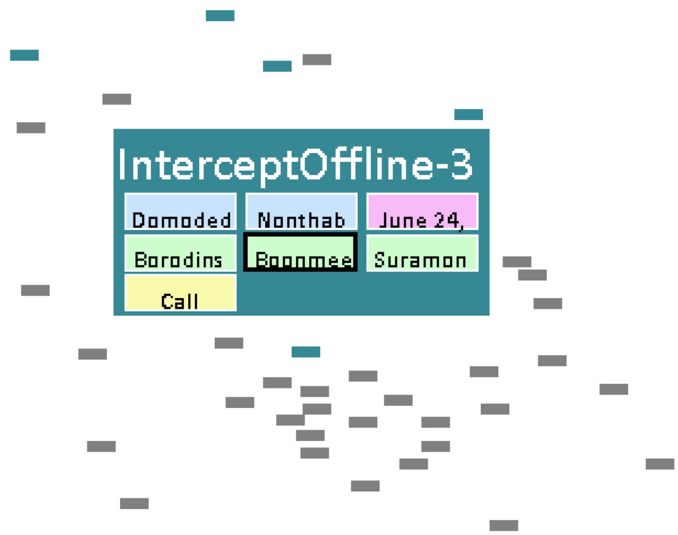


Figure 3-10. Brushing an entity to highlight the other documents containing it. View this image in colour to see the highlighting.

The ability to brush entities is what makes zooming into documents more than just a passive method of viewing them. It not only reveals their contents, but also reveals this method of interaction for finding related documents. Unlike other query methods, brushing happens directly within the view, without the need for an analyst to locate options in a side panel. This should better allow the analyst to maintain his/her train of thought and thus speed up an investigation.

Consider zooming into a document after running an overview layout using only people and location entities. Documents close to the focus document will contain many of the same people or locations. The analyst may not know which people or location entities are shared but can use brushing to find out quickly. Without brushing, zooming into all the surrounding documents would be necessary to find this same information. Brushing allows an analyst to focus easily on any entity of interest to them.

The next section describes how analysts can organize a document collection based on their hypotheses or simply structure the document collection into manageable smaller collections.

3.4 Structuring a Document Collection

As analysts start to build mental models of the structure and content of a document collection and to derive hypotheses about the contents, they often wish to impose their own structure on the document collection, for example grouping those documents that pertain to certain events. SZV provides capabilities to aid in this vital task, to assist analysts in keeping track of specific documents or document sets. These capabilities also support organizing a document collection into more manageable smaller collections. Described below are SZV's two main mechanisms for analysts to structure a document collection. These are creating hierarchies of groups of documents and placing documents and groups at specific locations in the view.

Within the layout, analysts may move documents manually in order to rearrange them into their own categories or simply to move them off to the side. After one of the query techniques (Section 3.3), relevant documents may be selected at various locations around the view and the analyst would like to keep this subset together. Thus, they will need to move them together to one location before grouping them or running a cluster layout on just that subset, to investigate them in more detail. Analysts can move documents by pressing and dragging them with the cursor, and if multiple documents are highlighted then they all move. All the documents being moved are translated by the same amount in x and y, so moving them in this manner does not make them any closer together. Thus, SZV has a gather function that the analyst can use to pull all highlighted (of one colour) documents into a tight cluster. Using the gather function, an analyst chooses a spot on the view to gather all documents highlighted in the active colour. Figure 3-11 shows the beginning, middle, and end of an animation that takes place when the analyst uses this function. The selected documents are pulled to one location, but maintain their same relative x-y positions with respect to each other. The algorithm for computing the documents' new locations involves moving the documents in order based on how close they are to the gather point. Each document is moved to the gather point and then moved back towards its start point as needed until it no longer overlaps an already moved document. Without the gather function, an analyst would have to drag each document one at a time to put them close together. This would make them less inclined to deal with any subsets of documents that are not already clustered together. Although the cluster layout is helpful, it is only one of many possible groupings; it should not prevent or discourage the analyst from forming other groupings.

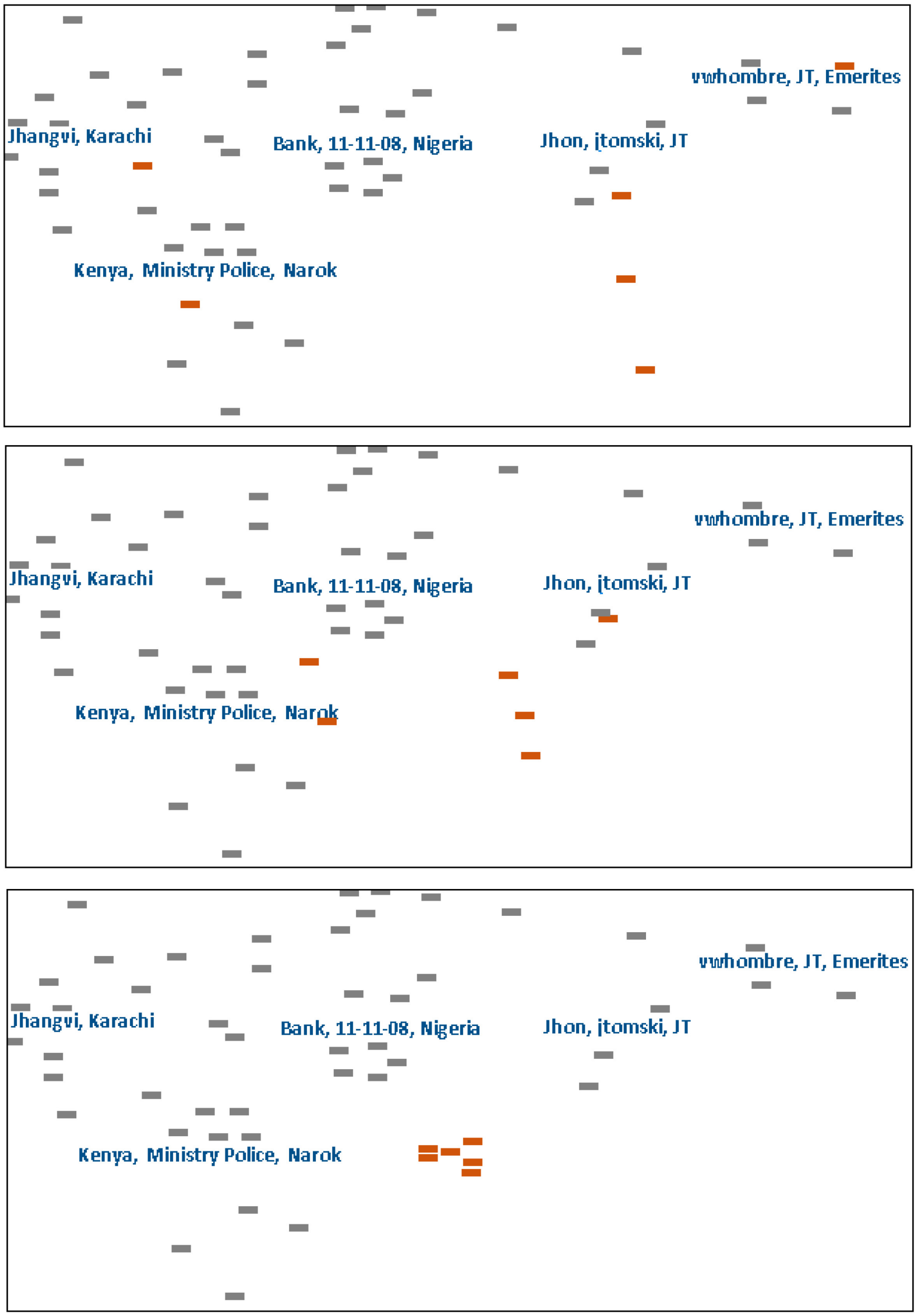


Figure 3-11. A sequence showing the animation of documents caused by the gather function. Given a set of highlighted documents, the animation is triggered by the analyst clicking the gather tool button and then a point (to move the documents to) on the screen. View this image in colour to see the highlighting.

As mentioned above, once a subset of documents is pulled aside, the analyst can form them into a group. A group is a persistent subset of documents shown within a rectangle frame within the main view. It can show different views of the contained documents, be moved around the screen, and opened and closed. Groups also have names, which allow analysts to keep track of many of them. Grouping is the strongest method SZV provides for marking a subset of documents. Humans are much better at distinguishing different locations than different colours (Cleveland et al. 1984), and in addition, a frame is displayed around the documents. Groups can be used for a variety of purposes. For example, an analyst may set aside some documents they do not currently wish to look at by putting them in a group and closing it. These documents will be easily accessible later but for the moment they do not use up space in the view. An analyst may create a group containing all the documents about a specific person or place. Then when they perform a search for anything else, the results are highlighted and they can instantly see how many documents both match the search and contain the person or place. Groups can be created to give more structure to the analyst's task. They can first investigate one group and then investigate another. Alternatively, they can give the CZSaw script file (Kadivar et at. 2009) to another analyst and have the other analyst investigate the second group. Thus, groups can aid in organizing who is investigating what in a collaborative analysis. By creating a group, an analyst no longer needs to use highlighting to keep track of the document subset; however, the arrangement of these documents in the overall layout is lost.

To create a group, the analyst chooses the group tool and then draws a rectangle around the documents to be grouped. Figure 3-12 shows this action and after the analyst enters a name for the group, it appears as in Figure 3-13.

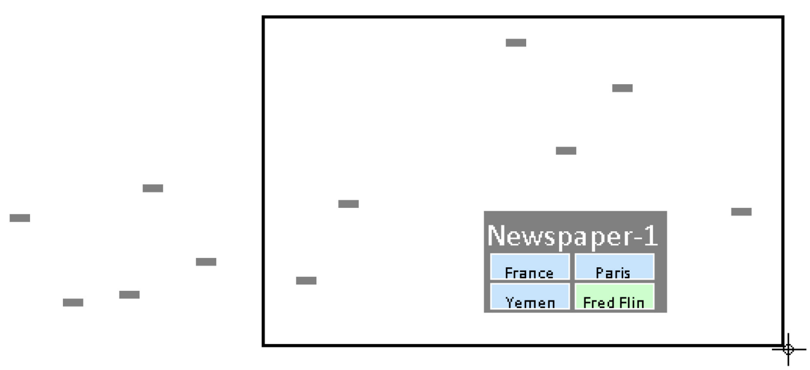


Figure 3-12. Creating a group by drawing a rectangle around a subset of documents.

A group can be moved as a unit and resized. Buttons at the top-right of the group let the analyst ungroup the documents or close the group. Closed groups hide the contained documents, showing only the group name. Groups have three tabs showing different views: the document glyphs (Figure 3-13), the entities (Figure 3-14) and the full text of each document (Figure 3-16).

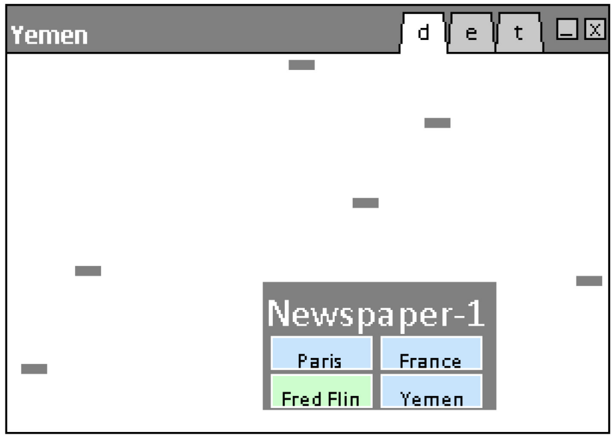


Figure 3-13. A group's document tab shows the document glyphs within.

A group's document tab displays document glyphs the same as if ungrouped and the documents can be zoomed and brushed normally. When zooming into grouped documents the layout adjustment algorithm is applied in two-steps to allow smooth zooming while keeping a group separate from the rest of the document collection. First, the layout adjustment algorithm (described in 3.2.3) is applied only to the group's documents and without the layout-scaling step at the end. Thus, the group boundaries expand or contract with the layout of the contained documents. Then the layout adjustment algorithm is run again on the rest of the view, but each group is treated as a single document. Thus, the expanding/contracting group is treated as if it were an expanding/contracting document. From this tab, analysts can also change a group's content by dragging or dropping documents or other groups into or out of it. The other group tabs are updated automatically when the group content is changed.

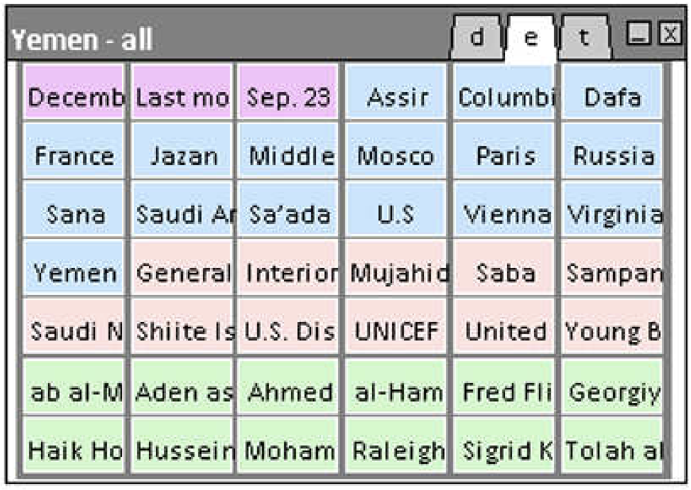


Figure 3-14. A group's entity tab displays all contained entities.

An overview of all entities in a group's documents is useful to give a quick overview of the group. A group's entity tab shows this in a compact, efficient manner. This tab shows a master grid of all the entities found in any of the contained documents, similar to the grids of each document glyph. If the group was created using all documents that contain an entity, then this view will be showing all entities that are in at least one document with the given entity. Analysts can also control the order of entities in order to find specific ones fasters. The entity display preferences for the main view (as mentioned in Section 3.1.3) also apply to this view of entities. For example, Figure 3-14 shows the entities grouped by type and then ordered alphabetically. Any entity type may be filtered out and the entities may alternatively be ordered by their appearance in the document's full text. Currently these options apply to the entire view; however, in the future they could be applied to only specific groups or documents.

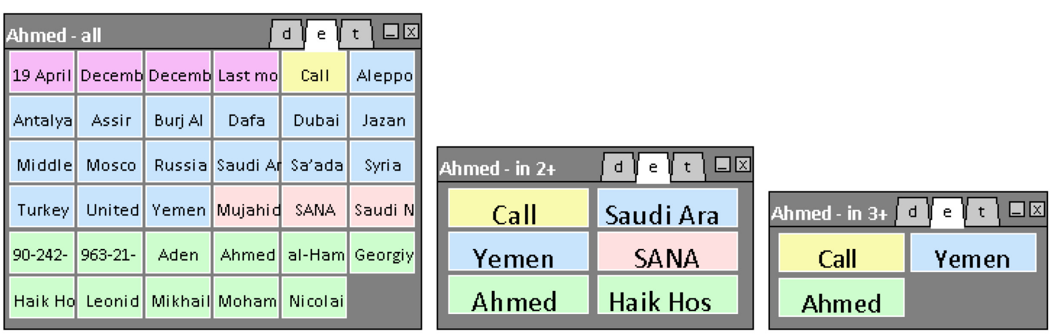


Figure 3-15. The dynamic entity filter changes which entities are visible in a group.

In addition, analysts can get a brief group summary and see its most frequent entities using the dynamic entity filter that filters out the less popular group entities dependent upon the available space. This feature considers the group's onscreen dimensions and the number of documents each entity is in to choose a threshold number of documents and then hide all entities not in at least that many documents. When this option is turned on, an analyst can resize a group to see quickly which entities in the group are the most popular, which are slightly less popular, etc. Figure 3-15 shows a group, called Ahmed, changing size with the dynamic entity filter applied. The title bar shows the threshold number of documents for the displayed entities.

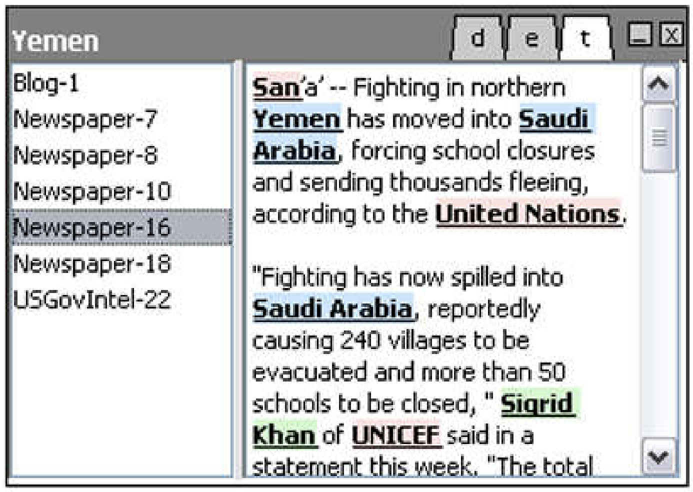


Figure 3-16. A group's text tab shows contained documents' full text.

The text tab gives the analyst an alternative mechanism for reading documents that is much more traditional. While an analyst cannot see the contents of multiple documents at once in this tab, they can quickly read documents in sequence using much less space than zooming into each. This group tab displays the full text of one of the documents in the group. The document being displayed can be chosen from a list of all the group's documents. In Figure 3-16, the text of Newspaper-16 is being shown. Within the text, all the document's entities are highlighted in their entity type colour. The document list can be sorted alphabetically, shown by date instead of by name, and sorted by date. In addition, an analyst has the option of choosing any of the documents in the list and having it highlighted in the group's document tab. In this way, they can move from one representation to another quickly. Note that this tab cannot be used to view simultaneously two documents' text. For that, the analyst must zoom both documents in the document tab.

SZV supports hierarchical groups. Groups can contain either groups or documents. All interaction methods mentioned above apply as expected when there are groups within groups. Moving a group moves all documents within the group and recursively within any group that is within the group. The layout adjustment algorithm applies recursively, moving up group levels until the outer layout is reached. Documents can be dragged and dropped from any inner group to an outer group or outside of all groups.

The creation of a hierarchy of groups allows the analyst to apply their own knowledge in a lasting manner to the document collection by providing additional structure to the collection. There are many strategies an analyst can use in their group creation. They can create a group from a cluster to keep the documents together permanently. They can create a group for all the documents that mention an entity of interest, a set of related entities of interest, or all documents supporting a given hypothesis. A group can be created for any set of documents they wish to keep together and to have quick access to a summary of using the group entity tab.

This chapter has described SZV's full design and how each feature should aid an analyst in their investigation of a document collection. The next chapter's focus is on evaluating the basic technique of SZV in order to identify its strengths and weakness with the goal of further improvement.

4: EVALUATION

The design of Semantic Zoom View (SZV) is based upon previous related work, which includes studies that validate the techniques as well suited to human perceptual capabilities and, specifically, analysts' needs. This does not guarantee that SZV will also be useful and intuitive while decreasing analysis time and increasing useful results. Thus, I chose to perform an evaluation of the main technique of a document overview with a focus + context embedded semantic zoom using a simplified version of SZV. This chapter describes the reasoning behind the approach taken, the design of the interfaces compared, the study design and the results found.

Following a brief description of the reasons for comparing a simplified version of SZV to an overview + detail technique, Section 4.2 outlines the differences between the two interfaces and how participants operated their basic functions. Section 4.3.1 describes the study method, which included a pre-study questionnaire, a reading exercise, a set of ten task questions to solve using the interface, and a post-study questionnaire rating the interface. All these study documents are included in Appendix B. The quantitative analysis plan is described in Section 4.3.2, including the alternative hypotheses and statistical tests used. Results 4.4, describes both the quantitative and qualitative results of the study with more focus on the qualitative results (further described in Appendix C), since the study resulted in many interesting process observations.

The main statistically significant result found was a higher accuracy in the SZV's zoom interface for one question. In Qualitative Results 4.4.2 and Appendix C, this difference between interfaces is explained including a number of strategies that the detail + context condition participants used that gave them a larger or more focused view of the details at the expense of the context. While participants in the SZV condition were more restricted and many wanted more freedom, the restrictions created less room for error. Also in Appendix C is the discussion of any difficulties found with the interaction mappings of the interface and its affordances. These problems can be addressed by adjusting the interaction mechanisms and performing further evaluations to test that the changes improved the interface. I begin this chapter by describing my basic approach in the study design.

4.1 Approach

SZV's many features all have the potential to contribute to its overall effectiveness and usability. At first, it would seem the best analysis of SZV would be to provide it to analysts who regularly use another tool. They could use it for their regular analysis and comment on its usefulness compared to the "other" tool. In addition, I could measure the difference in length and quality of the analysis between tools. The evaluation reported here is much different from this for three main reasons:

- SZV is not in a stage of development in which this would be useful.
- To understand effect, it is more appropriate to evaluate a specific technique rather than compare entire applications.
- A full comparison would require resources and time beyond the scope of this thesis.

Below is more detail on the above three points. In addition, I highlight ways in which this study relates to Plaisant's arguments presented in her discussion on evaluation of information visualization tools and techniques (2004). Lastly, I present qualitative comparison questions that the study was designed to answer.

Comparing the whole SZV application to a past system such as In-Spire (Hetzler et al. 2004) would involve measuring time and accuracy as participants use either or both systems for analysis. Unfortunately, SZV has been developed by a single graduate student, and thus is not at a stage where it can be compared to a polished application such as In-Spire, which has been developed and refined by teams of researchers at Pacific Northwest National Laboratory over more than 10 years. Although feedback has been received from informal users of the tool, a formal SZV evaluation was needed to identify any usability issues. Any such issues can then be addressed before comparing it to a commercial system. Early usability tests must focus on identifying areas for improvement, rather than on proving one application is better than another.

For the purposes of this thesis, I decided it was a better idea to investigate first whether the basic technique was useful to an analyst compared to more traditional approaches. It would not be wise to invest the necessary time and effort on an evaluation of the full SZV application, including debugging the application and training users, just to discover that they find the fundamental technique not intuitive and less useful than other methods. Thus, this chapter describes an evaluation of a semantic zoom within a document collection overview by using a simplified SZV version. It was compared to a more traditional approach that displays document contents separately in a popup frame instead of directly within the context. By using a simplified SZV version, the evaluation focused on the fundamental technique of semantic zooming without needing to train participants on the use of all the features of the application. Thus, I decided to evaluate the semantic zooming for viewing a document within its context and the use of brushing before further SZV development and testing.

The final reason for not conducting a long-term full evaluation with professional analysts is the lack of resources and time. In 2004, Plaisant remarked on the difficulties of evaluation of information visualization tools and techniques (2004). She noted that case studies of tools in realistic settings are the least common studies and are often difficult to replicate and generalize. Success using a tool may be hard to trace back to what aspect of the tool enabled them. By narrowing down the features included in the evaluation, I reduced the number of factors that could be responsible for participants' performance at the tasks. By evaluating the basic technique and providing detailed observations of participant use, I aim to provide researchers with useful knowledge about the technique that will allow them to adapt it into their systems. If instead I performed a full SZV evaluation, successes might be attributed to any combination of its features and thus difficult for another researcher to adapt.

Plaisant also notes that evaluations need to be improved by performing tasks more complex than just "locate" or "identify". She mentions a lack of more complex tasks such as compare, associate, distinguish and others. In the evaluation done for this thesis, I therefore had the participants perform complex tasks such as comparing multiple documents and associating one document with others through brushing. Each task differed from the rest by the steps necessary to answer the question (Section 4.3.1). Analysis of the results looked not only at each interface as a whole but also at the specific tasks and observations of why one interface performed better. Plaisant remarks that separating an analysis based on task allows potential adopters to find the right tool. For the purposes of this research, it is more useful to identify interface issues and the technique's strengths.

The evaluation reported here used undergraduate and graduate students as participants. Ideally, real analysts would have been used; however, it is essentially impossible for academic researchers to get access to real analysts. The In-Spire evaluation by Hetzler et al. (2004) was possible because they had access to many analysts working on a variety of problems. They also had a fully operational tool complete with access to online help, a printed user guide, and phone and email support. Analysts had free use of the tool rather than specific tasks and thus accuracy and analysis completion time were not measured. Clearly, case studies such as that performed with In-Spire are valuable; however, SZV is not at that stage in its development.

Given the above reasons, I concluded it was better to evaluate the basic technique of a document overview with an embedded selective semantic zoom. This was compared to a technique in which document contents are displayed in popup frames. One of SZV's main goals is to allow analysts to drill down quickly from an overview into a subset of documents using SZV's 5 semantic zoom levels. A quite different method is to jump directly from a glyph to the full document contents. The study was designed to test which is more useful to users. Questions include:

- Does a user find information faster semantically zooming to a specific detail level or when provided either all details or none?
- Does the scroll wheel mechanism for zooming work better than double clicking to see a document's contents?

A second goal of SZV was to enable analysts to see a document's content and the surrounding context simultaneously. Analysts should easily see the surrounding document glyphs to quickly see which ones are closer (and thus more related) and which are highlighted (from brushing). They also should be able to compare multiple documents' content easily. Questions include:

- Does a user find relevant documents faster when their focus documents are displayed among the closed (zoomed out) documents or when the focus documents are displayed separately from the layout?
- Does a layout adjusted by moving document glyphs outward so the focus documents do not overlap them provide a more useful context than an unaltered layout partially covered by the focus documents?

The interfaces differ in the method in which they provide the detail compared to the context; the study's goal was to determine which method of providing the detail makes the detail and context more accessible.

Participants used one of the two interfaces to answer a series of ten questions. Both time and accuracy were measured. Participants also completed a post-study questionnaire rating the interface they used and providing comments. Videos of the screen were captured to record issues participants encountered. This chapter describes the study design and results.

I begin by describe the two interfaces used for the study.

4.2 Interface Alternatives

First, I will describe the simplified version of SZV used for the study. All control panels were removed from the interface and the remaining main view was setup automatically for each of the ten tasks a participant had to accomplish (Listed in Appendix B with solution steps in Section 4.3.1). The setup consisted of an initial overview layout and for 8 of the 10 tasks, highlighted search results. Participants were told the meaning behind the layout clusters and what search was done but did not create their own layout or perform any searches.

Their task was to answer questions about information contained in the documents that required zooming into and reading documents. Participants could select multiple documents and zoom into them simultaneously as described in Chapter 3. For the more advanced tasks involving brushing, participants used the semantic zoom levels to determine which entities they had to brush. The selection of documents, zooming into and out of documents to see their contents, and the brushing of entities were the only lower level actions participants were given. Thus, besides no search or layout tasks, the participants also did not perform any highlighting by date or gathering or grouping of the documents. None of this functionality was available to the participants. In addition, this simplified version only used one colour for highlighting.

Some additions were made to the interface based on difficulties experienced by 7 pilot participants. The pilot participants had trouble locating the glyphs highlighted from brushing. Thus, the highlight colour was changed to pure red and immediately upon brushing, small red circles faded in and out around the newly highlighted documents for 2 seconds. The circles attracted participants’ eyes to the highlighted documents’ locations. A legend for the entity type colours was added below the main view, since normally the options panels each had this information (but they were removed for the study). Figure 4-1 shows this simplified SZV version and the highlighting circles. In this chapter, this version will be referred to as the zoom condition or zoom interface.

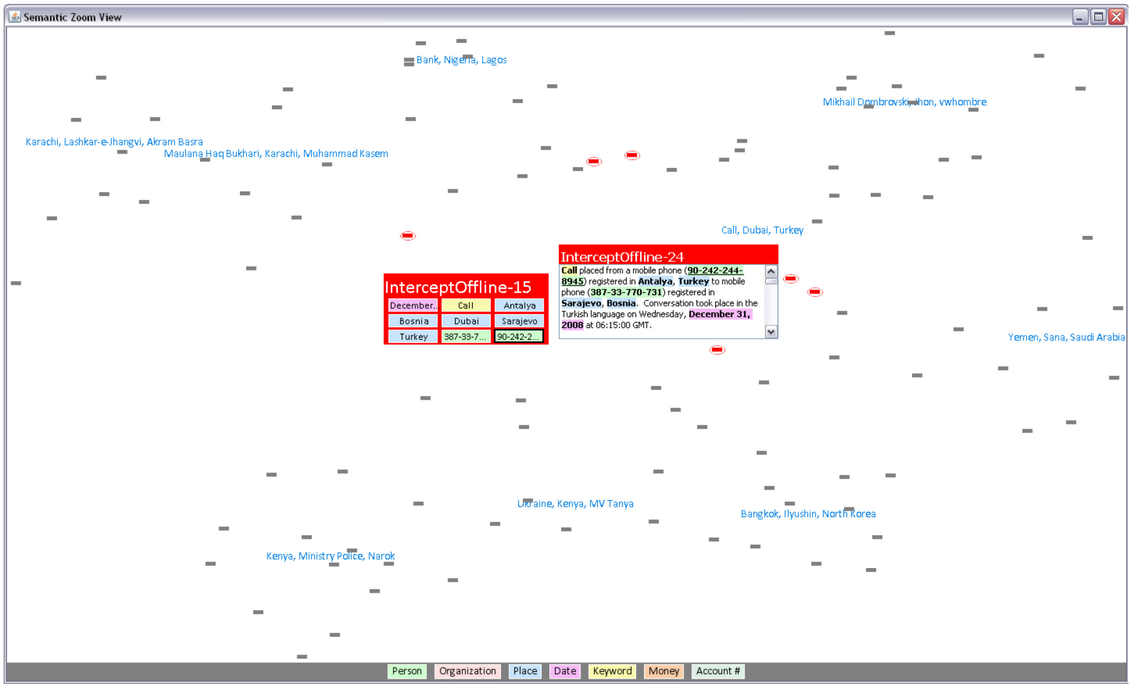


Figure 4-1. The study's zoom interface condition immediately following brushing.

For the other study condition, an interface was created that offered access to the same information but without a semantic zoom or displaying the focus documents directly within the context. Instead, it displayed the document entities and full-text within popup frames in a layer above the document overview. This will be referred to as the popup condition or popup interface. At the beginning of a task, the popup and zoom interfaces looked identical because no documents were open. The popup interface provided the same selection functions as the zoom interface; however, in order to view a document, the participant double clicked the glyph. When they did this, the document's contents were displayed in a new frame (the popup) in a corner of the view with a line connecting the popup to the glyph. Popups were displayed in a layer on top of the overview instead of within it, resulting in anything "underneath" the popup being hidden from view. Figure 4-2 displays the popup interface at the same task state as the zoom interface is in Figure 4-1.

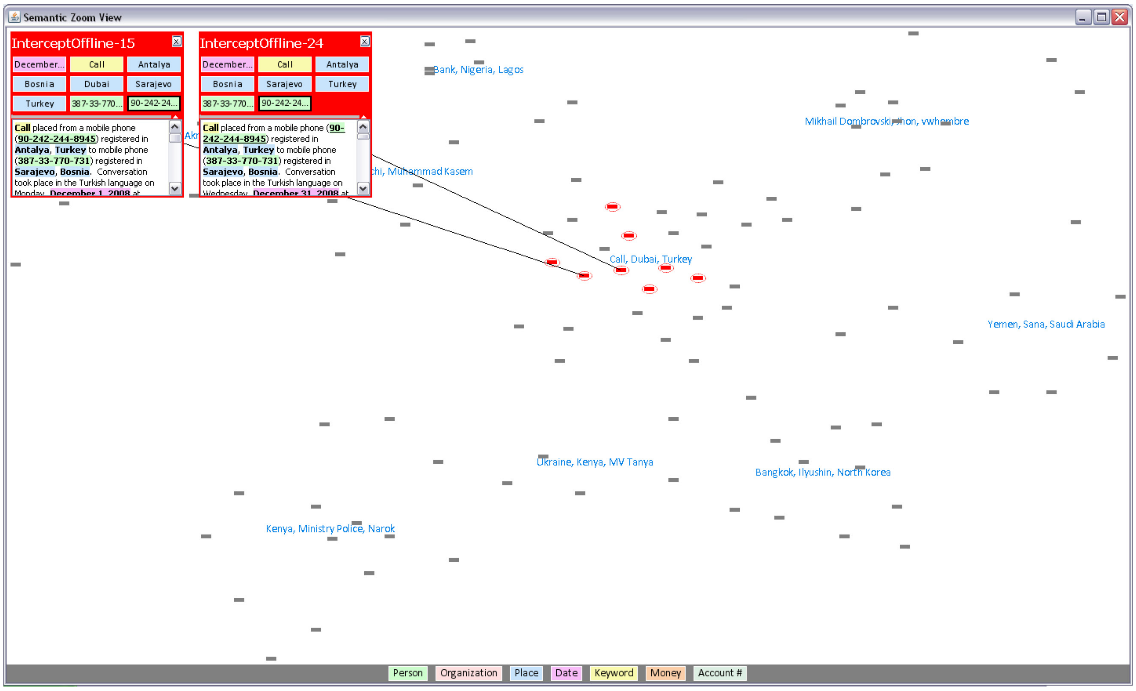


Figure 4-2. The study's popup interface condition immediately following brushing.

Just as in the zoom interface, participants could open multiple documents at once by holding shift and double clicking a document to open all highlighted documents or to close them all (if the document was already opened). Popups appeared from left to right across the top of the screen and then started a new row as needed.

Popups displayed all entity rectangles in a grid in their top half to provide a summary similar to semantic zoom level 4 (Section 3.1.4). Though participants could hide this top half, none chose to do so. The bottom half (which expanded when the top half was hidden) displayed the document's full text similar to semantic zoom level 5 (Section 3.1.5). Brushing on the entities could be performed exactly the same as brushing in the other condition. The document popups could be moved and resized in standard ways. The original size of a popup depended upon the entity grid dimensions, which in turn depended on the number of entities in a document, just as in the zoom condition (Section 3.1.3).

The two interfaces differed only in the above-mentioned ways to ensure that any differences in performance could be attributed to the document viewing mechanism or its interaction with other features. All the same information could be reached in both interfaces but there were differences in what could be seen simultaneously. In the zoom interface, a participant could not see a document's entity grid and the full text at once, but they could see the entire layout of all the documents. In the popup condition, the participants could see the entity grid and the full text at once but the popups covered part of the layout. Both interfaces provided the same brushing capability; however, in the popup condition there was the possibility of highlighted document glyphs being covered by the popups. Brushing often caused documents near the one being brushed to be highlighted since the layout places similar documents near each other. In the zoom condition, these documents are near the focus document, but in the popup condition, the participant's focus is on a document popup not necessarily near its matching glyph. The next section describes the design of the study.

4.3 Study Design

I designed the study with two goals. The first goal was to find a quantitative difference between the task completion times of participants using the two interfaces in accurately answering questions about a document collection. The second goal was to collect qualitative feedback on the strengths and weaknesses of the semantic zooming as a focus + context technique directly in the overview, to identify areas of improvement to investigate in further studies.

The study used a between-subjects design with 20 subjects such that each interface was used by 10 students. There were several reasons for the between-subjects design. First, because of the need for training and having each participant solve 10 questions, using both interfaces would require too much of the participant's time, and they may also be fatigued after the first interface. Alternatively, since concepts such as "entities" and actions such as brushing are the same across interfaces, participants would gain useful experience from the first interface. In addition, within-subjects would require multiple document collections with question sets that need to be proven to be similar difficulty. Using a between-subjects design limits the comparisons that can be made between groups; however, members of these groups were chosen randomly and the demographics collected from these students were not significantly different.

The document collection used in the study was a modified version of the Visual Analytics Science and Technology 2010 Mini Challenge 1 data set (Grinstein et al. 2010). Our CZSaw team solved this challenge (Chen et al. 2010) and the grand challenge (Dunsmuir et al. 2010) and was given an award for each solution. Our work on the solution included the use of SZV. This document collection is also featured in screenshots throughout this thesis. This collection consists of 102 documents, all fictional, that fall into one of five categories: US government intelligence reports, newspaper articles, website and blog posts, telephone and wired communication intercepts, and email and message board intercepts. Entities were extracted from the documents by Maxim Roy of SFU's Natural Language Lab using Alias-i's LingPipe system (LingPipe 2010). The documents range in length from 16 to 890 words and in content from 1 to 23 entities.

I created a set of 10 questions that could only be answered by viewing the entities or text of one or more documents. Each question could be solved by viewing a different set of documents with no overlaps between questions; however, participants saw all document glyphs for each question and so they could investigate a document by accident (not on route to solution) that they had already seen. This rarely happened.

The participants interacted with the interfaces using a mouse and keyboard. The mouse had a scroll wheel and participants only needed the keyboard for the shift key. The study used a single 30-inch 2500 x 1600 HD LCD monitor.

4.3.1 Method

A pilot with 7 participants was used to refine the design. In this section I walk through the phases of a participant's session, each lasting between 40 and 60 minutes. All participants were students at SFU who had no experience with the document collection used. Participants began the study by reading and signing the consent form and filling out a pre-study questionnaire to collect demographics and data on their information visualization experience level and frequency of non-academic reading. This questionnaire is featured in Appendix B along with the study documents described below.

Next, the participants completed a reading exercise where they were presented 5 short paper documents taken from the task dataset. Entities in the documents were highlighted by type using colours roughly the same as those in the interface (adjusted to account for the fact that colours appear much different on paper than a screen). In addition, the sheet of paper included a legend for the colours. I explained what an "entity" was to the participants. They then had to answer 3 questions using the documents' content. They wrote their answers on the document paper. This task was designed to be the same as the main study tasks except done on paper instead of using a visual analytics tool. Thus, by recording the participants' reading exercise times and accuracy (score out of 3), I could control for different reading skill levels. The documents from this exercise were not useful in solving any of the main task questions.

After the reading exercise, participants were given a short training session (under 10 minutes) on the basic interactions of the interface. A training script for each interface ensured each participant received the same training. I explained the following to participants:

1. How to zoom into or open documents.
2. The meaning of different zoom levels or popup parts, ie. that entity rectangles were entities in the document.
3. What the layout meant, how documents closer to each other were similar.
4. How to perform brushing.
5. How to select multiple documents and zoom into or open them together.

Following this description and a demonstration of how to use the interface, I walked through solving two sample questions that were in the same format as the main task questions. The first question was a simple question with only one part and the second question was an advanced two part question involving brushing. The demonstration showed no documents needed for the main task answers.

Then the main task began. Participants were given 10 questions on a sheet of paper. They answered the questions in sequence one at a time, with screen-recording software saving each solution process separately. The participants were given time to read each question and plan how to solve it before working with the interface. This planning time was not measured as we were only interested in how long participants took using the interface.

The interface was setup separately for each question-answering task, with all document glyphs in a clustered layout based on people, places, organizations and keywords. Also, for 8 of the 10 questions, a document subset was highlighted as search results. In these 8 questions, the search term used was stated (see Appendix B). The screen capturing software was started when the participants began to the use the interface and stopped when they found the answer, which they then wrote on the paper. The 10 questions each involve different steps to most directly solve. The set of quickest steps for solving all 10 questions in the zoom condition are listed below:

1. Zoom into a single document and the answer is an entity (highlighted).
2. Zoom into several documents and the answer (entity) can be found in two of them.
3. Zoom into several documents and compare entities in them to find the answer.
4. Zoom into a single document and brush an entity. This highlights another single document and then zoom into this document to find the answer (entity).
5. Zoom into one document based on its cluster, brush an entity and then count the number of highlighted documents.
6. Zoom into several documents and brush an entity from one of two of them. Then zoom into the several highlighted documents and find the answer (entity) in one of them.
7. The participant is told exactly which entity to brush but must find it within several documents by zooming into them. After brushing several documents are highlighted

and the participant can find the answer (entity) within one of two of them by zooming.

8. Zoom into multiple documents based on their cluster and find an entity. Brush it and then find the answer (entity) within one of the highlighted documents by zooming.
9. Zoom into multiple documents and determine which entity to brush (one of two of them) that is a phone number (more advanced than other questions). Zoom into the newly highlighted documents and find the answer (entity) in one.
10. Zoom into multiple documents and brush an entity that is in one of them. Zoom into the newly highlighted documents and find the answer (not an entity) in one of them.

The questions involving brushing specific entities and then finding more entities were asked in two parts on the task sheet as shown below in question 6.

6a. All documents containing "Nicolai" are highlighted. Nicolai Kuryakin is a key player in the network of arms dealers. He attended two events related to weaponry, one in June and one in September. Find the name of his companion at these events.

6b. Who did the companion meet with on June 26th?

Part *a* asked the participant to find an entity that met some condition and then part *b* asked them to find something else related to part *a*'s entity answer, that required brushing it. All tasks involved zooming into (or opening) at least one document. While participants worked with the interface, notes were taken by the experimenter on any interesting interactions. Accuracy was determined by the number of questions correct (out of 10) and time taken per question was recorded by the screen capture software.

Next, participants completed a post-study questionnaire containing 10 7-point Likert scale questions rating different aspects of the interface. Half the questions were phrased negatively such that participants had to clearly read each statement. Participants were also encouraged to write comments in the questionnaire.

4.3.2 Analysis Design

The null hypothesis was that there was no difference in speed or accuracy between the participants using each of the two interfaces for solving any of the 10 questions and that there would be no difference in the post-study questionnaire ratings. The alternative hypothesis was that the zoom interface would allow faster completion time with relatively the same accuracy and better post-study questionnaire ratings. I expected high (and thus similar) accuracy in both conditions because the 10 questions were straightforward. This was done on purpose in order to compare the completion time needed to correctly complete analysis tasks. Many incorrect answers may lead to widely varying question answering time and is much less useful to know than time to find a correct answer.

Analysis of the accuracy difference between the two interfaces was done per question using the mean accuracy from all 10 participants in each condition. A one-tailed Fisher's exact test was run for each question. Section 4.4.1 describes how this test provided the only statistically significant difference for one question with a higher accuracy in the zoom condition. I used a one-sided test here because from observation of participants it appeared there might be a difference in accuracy. In addition, I created a logistic regression model to find which variables were predictors of the question accuracy. For analysis of completion time, a linear model was created to determine which factors significantly contributed to the time taken. Finally, a one-tailed t-test was performed for each of the post-study questionnaire questions to compare the means of each interface condition. The significant results of this quantitative analysis are presented in Section 4.4.1; however, the most valuable results from this study were the qualitative results discussed in Section 4.4.2 and Appendix C.

4.4 Results

Much useful qualitative information was collected from the study through observation of the participants, recorded videos, and feedback written in post-study questionnaire comments. These turned out to be the most valuable results of the study since the quantitative results were of limited (statistical) significance, as described below. The only statistical significance found was a higher accuracy in the zoom condition for one of the task questions. The reason for this, as explained in Section 4.4.1 is that in the popup condition many participants needed to locate highlighted documents that were covered by popups. Section 4.4.2 explains the tendencies of many popup participants to cover up much of the view, but also how they had more freedom in controlling the space displaying the details (document contents). In contrast, the zoom condition provided a more restrictive interface. This section and Appendix C also describe difficulties participants had using the interfaces in both conditions. Many of the issues can be solved by adjusting the mouse mappings, fixing the colours used, or adding more advanced options to SZV.

4.4.1 Quantitative Results

In this section, I describe the quantitative results of the study that unfortunately did not support the hypotheses presented in the Analysis Design 4.3.2. First, there were no significant differences between the demographics of the two participant groups. In this section, I focus on the results of the analysis of accuracy, which found that on one question the zoom condition had significantly higher accuracy than the popup condition. This was due to the popups covering the context that was needed to use brushing and linking successfully. Only one question had lower accuracy in the zoom condition and reasons for this are hypothesized. Models were created (Section 4.3.2) to determine which variables, such as the interface condition, were predictors of the accuracy and log of the completion time; however, these found only the question number to be a significant predictor. Time to complete task questions varied greatly between questions as the difficulty increased. Finally, the post-study questionnaire found that the average scores were all on the positive half of the Likert scales with no significant difference between conditions.

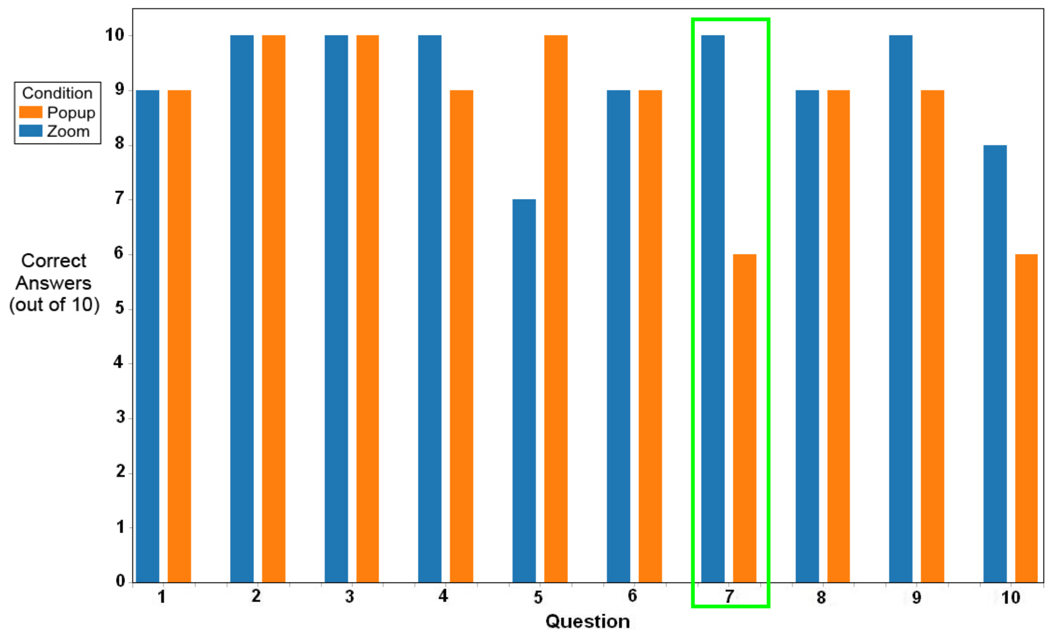


Figure 4-3. The accuracy per question for each condition.

Figure 4-3 shows the accuracy per question for each interface condition. A value of 6 means 6 out of 10 participants answered the question correctly. Average accuracy was high as expected (8.95 out of 10 correct). A one-tailed Fisher's exact test was used to analyze the accuracy for each question. Table 4-1 shows question 7 is the only one with a significant difference. The reason participants in the zoom condition performed significantly better for this question is clear from observing participants. This is the only question that was setup such that some of the documents highlighted from brushing were near the top of the screen and tended be covered by popups. Thus, some participants had trouble locating these documents and completing the task correctly.

Table 4-1. **Results of one-tailed Fisher's exact tests on questions that differed in accuracy between the two conditions. Note that accuracy was only higher in the popup condition for question 5.**

Question	Correct Zoom Answers	Correct Popup Answers	One-tailed p-value
4	10	9	0.5
5	7	10	0.11
7	10	6	0.043
9	10	9	0.5
10	8	6	0.31

Note that question 5, the only question with higher accuracy in the popup condition, required participants to locate the correct cluster based on its label and then zoom into at least one of its documents. Once zoomed in they needed to brush the correct entity (the one from the label) and count the number of highlighted documents that contained it. The 3 participants who answered incorrectly all began the question by simultaneously zooming into all 5 documents they considered to make up the cluster. Then 2 of the 3 decided these 5 documents must exactly be the document subset that contained the entity and thus wrote down the answer 5 (correct answer is 6). The third participant correctly brushed the entity and then counted the highlighted documents; however, the one unzoomed newly highlighted document was missed by their counting. I hypothesize two reasons that these participants failed the question. First, they did not understand that the entities listed in a label were only the frequent entities of the cluster and were not guaranteed to appear in all the cluster documents and only them. Allowing the analyst to brush these cluster labels and see all the documents that contained them would have removed this problem. Secondly, (for the third participant) the proximity of the large bright zoomed in and highlighted documents to the one unzoomed highlighted document caused the participant to not notice it. Due to popups being placed off to the side this was not a problem in that condition.

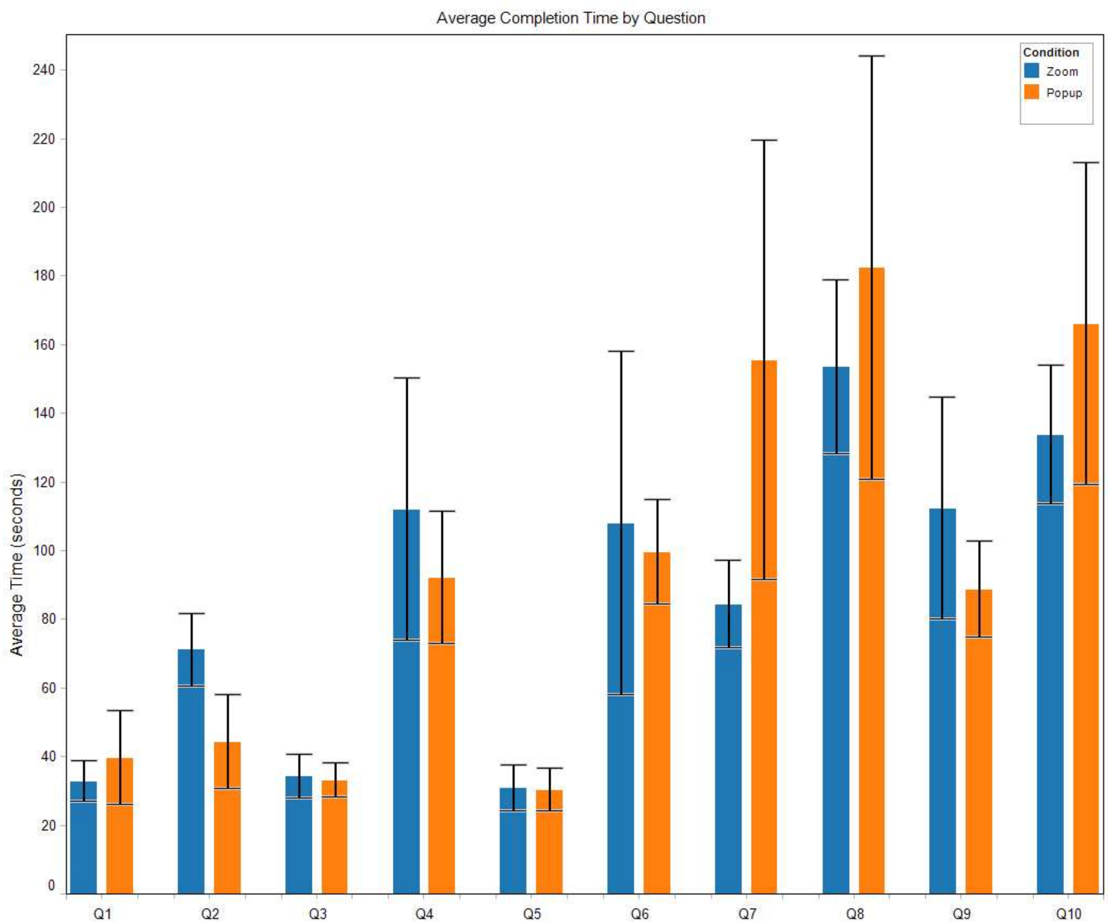


Figure 4-4. Average completion time for each question by interface (one standard deviation shown).

Figure 4-4 shows that average question completion time varied widely over the question set with less variation between the two interfaces; however, there are a few questions with noticeable differences, with the zoom condition participants completing the harder questions 7, 8, and 10 much faster on average. There is a larger standard deviation for many of the harder questions as some participants took much longer than average. The average time for all tasks completed by the zoom condition participants was less than three minutes. The more advanced tasks included looking through several documents in these three minutes. For the simpler tasks, the average completion time was around 30 seconds demonstrating participants were quickly able to use the interface even for their first tasks.

The post-study questionnaire is included in Appendix B and there were only small differences between the two interface conditions on all of the ratings. The average rating for all questions was above 4, and for question 4 was 6.8 and 6.9 for the zoom and popup conditions respectively. This means participants strongly agreed that brushing was useful for answering the questions. This is not surprising since brushing was the only query mechanism available. It is important to point out that brushing is not possible within In-Spire (Hetzler et al. 2004). The quantitative hypotheses presented in Section 4.3.2 were not supported by the study; however, the post-study questionnaire was useful for collecting qualitative feedback about the usability of the interface and technique.

4.4.2 Qualitative Results

This section is a brief summary of the study's qualitative findings. The full findings are described in detail in Appendix C.

First, in both conditions a few users found the glyphs too small to move the mouse over quickly. Second, due to the short training time, participants initially had minor trouble remembering how to control the semantic zoom or properly open the popups for multiple documents. To perform selection and document viewing, the popup condition involved double clicking, a mapping that was confused with single clicking for the selection of documents and entity brushing. The zoom interface involved less clicking and thus did not have these problems although participants were much less familiar with the use of a scroll wheel for zooming and the mapping of the scrolling direction went against conventions so participants had to adapt to it. Some zoom condition participants also commented that the scroll wheel was too slow and they would like to jump instantly between detail levels. Section 6.1 Future Work describes a new method of controlling the semantic zoom that a future study could compare to the current method.

For one question, popup condition participants had popups that covered glyphs that were highlighted by brushing, and so they had to move the popups to find these documents. This problem did not happen in the zoom condition because the focus documents appear within the context. In addition, participants in the popup condition had more freedom in how they viewed document contents, including moving and resizing popups. Unfortunately, this flexibility led to further covering of the context, making the problem worse. In the zoom condition, participants did not have these problems but instead commented they would have liked to be able to resize or move documents (available in full SZV).

There were some problems with the colours used in both interfaces that led to difficulties in quickly locating highlighted glyphs. The open documents displayed too much bright colour that attracted too much attention compared to the highlighted glyphs. In the zoom condition this was a worse problem since the open (zoomed in documents) were closer to the glyphs. Appendix

C contains more details about all the above issues, while the next section describes limitations of this evaluation.

4.5 Limitations

This section describes the main study limitations resulting from the choice of alternative interface and the set of participants used, along with some smaller issues in the reading exercise, interface training, the method of observing participants, and the data used.

I emphasize that SZV is a new combination of techniques and that there are many other techniques with which it can be compared. This study tested but two aspects of the technique, the use of focus + context and of semantic zooming. Further, it was compared to only a single specific alternate interface.

A few low level problems occurred during study execution. During the reading exercise, participants may have spotted answers while I was explaining what they had to do, which would have marginally decreased their reading exercise time. Unfortunately, I did not let participants have any free time to play with the interface before the task. Demos were given of solving example task questions, but putting the interface in the participant's hands for just a few minutes before the actual tasks could have increased their performance on the first real tasks. When participants were solving the tasks, I was watching them work to take note of the actions that they took. Though participants were asked to focus on quality answers rather than time taken, they may have felt pressured to work quickly because I was watching them. Thus, if I had not been watching they may have used the interface more easily.

This study was clearly far removed from a real life analysis session - participants were students rather than analysts and the task used artificial data. The difference between actual analysts and students was made clear when on two occasions participants located the correct document but then failed to write down the right answer from within it. Analysts, who are very familiar with reading these types of documents to find facts, would be much less likely to make this mistake.

Although the data were artificial, entity tagging was still imperfect. For example, some documents, not necessary to reach the solutions, contained unhighlighted entities.

One participant noted that much of their accuracy with the tool was due to the limited data plus knowledge that all the answers actually existed in the documents and could be found starting with the search results I gave them. This is a good point, but unavoidable in studies such as this, since tasks must have clearly defined answers. In fact, this problem also occurs in much larger contexts, including the VAST contest itself and even in case studies used to train real analysts such as in the US Joint Military Intelligence College (Dill 2011).

In the future, a case study may focus on real analysts using SZV with real data for many days without finding what they are looking for in the data. Such a study would require a polished version of SZV, access to analysts, and document collections in a format that SZV can use.

4.6 Related Studies

There have been some past studies similar to the one described in this chapter.

As mention in Section 2.3, a predecessor to the Continuous Zoom algorithm was the variable-zoom developed by Zuo (1992) for use with hierarchically clustered networks. A study by Schaffer et al. (1996) compared it to normal zooming (no context) for telephone network monitoring tasks. One finding from this study was that the balance factor that determines the screen real estate given to the focus area versus the context is very important to manage the visibility and size tradeoff. When an expanded node took up most of the screen, it strained the system's ability to display the hierarchies of the rest of the network. In the SZV study, this problem did not occur in the zoom condition because the documents had a maximum size they could be zoomed to; however, with a larger data set or more documents needing to be simultaneously zoomed, one would want a larger monitor or multiple screens to combat this problem. In the popup condition, participants could resize the popups and when they did this it covered much of the overview, which caused the only significance difference in accuracy between conditions.

Hornbæk et al. (2002) compared Zoomable User Interfaces with and without overviews and found that subjects who switched between overview and detail took more time. They hypothesized this was because integrating the two views required more mental effort. Their study did not include a focus + context condition.

The Fluid Links hypertext browser (Zellweger et al. 1998) displays descriptions of links as secondary text, called glosses, within the primary text as an alternative to placing the text in a popup or footnote. Fluid Links uses animation to place the glosses close to the link anchors while disturbing the primary text as little as possible. This is an example of a focus + context technique.

A study was done to compare the Fluid Links to popups or footnotes (Zellweger et al. 2000), and it was found that glosses placed close to the link anchors were displayed by the subjects for a much shorter time than those placed farther away. In addition they found that subjects kept the gloss open longer when it was in a footnote since they had to take longer to reorient their focus to it. The SZV study's popup condition was much like Fluid Links' popup and footnote conditions (because the popup is placed off to the side); however, the same effect was not seen. This is most likely due to large differences between the studies. Document contents in SZV are larger and are a more central aspect of the interface than the glosses in the Fluid Links system. Regardless of how they were displayed, the full document contents were always noticed by participants and looked at.

5: DISCUSSION

The main contributions of this thesis are the development and refinement of Semantic Zoom View (SZV) and the qualitative results of a study of its main technique. Building on previous work in document overviews, zoomable user interfaces, and focus + context techniques, SZV is designed to enable an analyst to see the big picture in a document collection with a seamless capability to drill down to the details of individual documents. Chapter 3 described the full Semantic Zoom View visualization while the evaluation in Chapter 4 focused on strengths and weaknesses of only SZV's basic technique; here we discuss strengths and weaknesses of the full application along with feedback received informally outside the study.

We begin by placing SZV in the context of its containing application, CZSaw, which provides an encompassing architecture that includes capturing and supporting the analysis process.

5.1 A Component of CZSaw

SZV is one of the data visualizations in the CZSaw Visual Analytics system being developed at Simon Fraser University by a team of faculty and graduate students (Kadivar et al. 2009). It is a visual analytics application inspired by Jigsaw (Stasko et al. 2008) for investigative analysis of collections of text documents. CZSaw differs from previous visual analytics applications because it also focuses on representing and visualizing the analysis process itself. These differences aided our team in solving the IEEE Visual Analytics Science and Technology (VAST) 2010 Mini Challenge 1 using CZSaw (Chen et al. 2010). This section describes the components of CZSaw that helped us gain an award for "Outstanding Interaction Model" in the challenge. Its dual focus is on SZV's use by the students and on SZV's use of aspects of CZSaw.

First, CZSaw has two data views other than SZV. The *document view* is similar to SZV's group's document tab. It displays a list of documents and for the selected document, shows a list of its entities and its full text. The other data view is the *hybrid view* with a node-link diagram showing relationships between documents and entities. Where SZV uses containment to show a document contains an entity, the *hybrid view* uses an edge between nodes to show this relationship. While SZV does not show as many types of relationships as the *hybrid view*, it integrates document's detail levels and contents directly into one view.

In CZSaw, the analyst can switch their perspective of data by dragging and dropping data from one view to another. For example, dragging a single document or group of documents from SZV into a *hybrid view* results in the display of a graph node for each of the documents transferred. The analyst can then expand the graph to show a social network of people and places in the documents. In this way the analysis can integrate all three types of data visualizations.

Quick generation of many visualizations will quickly complicate the analysis process, and so to track this history CZSaw records the analyst's actions within the interface in a custom-made script language. A history visualization visualizes this script by displaying screenshots of the process. In addition, CZSaw has an underlying dependency graph that maintains the dependencies between imported data, results sets from queries, and visualizations. This dependency graph enables the views to update instantly when changes are made to the root data or to any existing queries.

The *history view* and underlying script were useful to our team in solving the VAST challenge using SZV and the other data views. With SZV, I was able to take a divide and conquer approach to the dataset by dividing the document set into groups each with one country's activities. The groups I created in SZV were automatically recorded as variables in CZSaw's script language. Then I gave my CZSaw project folder to team members who ran it and generated the same groups and could see my process in creating them. This allowed us to divide the work of the analysis because a single student analyzed each group.

In our work on solving the VAST Mini Challenge, we often wished to merge entities such as two names of people that were actually the same person (slightly different spellings of the same name). To meet this need, we added several entity refinement features to CZSaw. For example, an analyst can identify ("extract") new entities by selecting text in the *document view* and choosing an entity type or creating a new one. This will find the entity within all documents in the collection. Similarly, an entity can be removed from a document or the entire collection. In both *hybrid* and *document views,* multiple entities can be merged into a single entity.

These entity refinement capabilities are essential for analysts to move forward in their analysis and make full use of the capabilities of each data view. SZV layout capabilities, for example, depend upon an accurate set of entities in the document collection. A lack of clusters in a layout, for instance, suggests very different documents (not about related events), a particularly diffuse problem, or a problem with the current set of entities.

Clearly, access to entity refinement functions must be a central feature of CZSaw accessible from every data view including SZV. This is just one example of the ongoing integration of SZV into CZSaw. In the future, the analyst will be able to add or remove entities in the documents in SZV just as they currently can in the *document view*. They will also be able to merge two entities into one and see the change instantly in SZV and every other data view.

5.2 Strengths and Weaknesses

This section will summarize Semantic Zoom View's strengths and weaknesses based on its application of past proven techniques, informal use by graduate students, formal evaluation of a simplified version, and feedback received from attendees at the 2010 IEEE Visual Analytics Science and Technology (VAST) conference.

SZV provides a flexible document collection overview using a simplistic document model, i.e. a set of entities. Document clusters result from commonalities in these entity sets. This model is much simpler than In-Spire's model, which considers most of the words in each document including their frequency within the document (Hetzler et al. 2004). However, SZV's entities are categorized by type and thus carry more information than In-Spire's words. SZV can provide the analyst the ability to turn on and off the influence of each entity type. In fact, a researcher from Oculus Info Inc. (the sole Canadian VA company) was very interested in SZV's capability to rerun the layout with different entity types. In addition, the model is consistent with the rest of CZSaw, which focuses on entities and relationships between them and allows refinement of entities. We note in passing that this entity refinement is a way for users to add their knowledge to the document set, and of course, the SZV layout will also change with this added knowledge.

Once a good level of entity accuracy was achieved in the VAST challenge, we found SZV to be highly effective at clustering together all the documents about each country, since the cluster layout placed many of these documents close together. The formal evaluation included two task questions where users were forced to choose which documents they needed to zoom based only upon the cluster layout. Unfortunately, for the purposes of keeping tasks short and approachable by non-analysts, the cluster labels matched words within the questions. Thus, the initial document selection task involved reading all the labels to find the cluster with the relevant label. Another study would have to be done focusing solely on the layout mechanism to determine more completely its value to analysts. Along these lines we note, a researcher from Defence Research and Development Canada (DRDC) commented that the cluster layout is as good as In-Spire's, a helpful and nice comment to have. We add that while SZV's force-directed layout is much slower than In-Spire's algorithm, layout performance improvements are certainly possible, and as described in Section 6.1, this can be addressed while retaining the entity model.

SZV applies a focus + context method to provide a new mechanism for viewing document content within an overview. The formal evaluation made apparent that although some refinement of the current interaction mechanism is needed, the visualization has some advantages over alternatives. The formal evaluation found some participants had difficulty either moving the mouse over a desired document (small targets) or quickly determining which direction to scroll to zoom in. In his review of zoomable user interfaces (ZUIs), Bederson includes "Standardized Navigation" as a key area for further research of ZUIs (2009). There currently is no standard, so it is difficult to provide familiar controls to a user. Fortunately, participants in the study quickly became familiar with the zooming controls and easily understood the concept of semantic zooming. Not all users found many of the zoom levels useful, and so these can be adjusted in the future (Section 6.1).

The focus + context technique displayed an advantage over the popup technique. Embedding the focus documents directly in the context did not hide necessary information. All documents could be seen at once, which made brushing more efficient. In addition, participants considered the focus documents and their context at once instead of glancing back and forth between popups and the layout. SZV provides a direct manipulation, instant access, details on demand approach interface. An analyst does not have to open up a separate window or even move the cursor over to a control panel in order to view a document. During a CZSaw demonstration, a researcher from SAP noted that the more effort it takes to view a document, the less often people will. Since document content viewing is an important reoccurring subtask, it is important to support quick drill down, as does SZV.

In the study, popup participants could resize the popups and move them around to compare multiple documents side by side. The zoom participants did not have these controls; however, several commented that they would have liked to move documents close together for easier comparison. Participants in the popup condition of the study performed much more moving and resizing then was needed for the information seeking task. This unnecessary fiddling led to more context being covered and lower accuracy. The full SZV allows analysts to take either approach because they can move documents around if they wish to place all the documents they are investigating closer together. The disadvantage is that this upsets the original layout. This trade off can be managed by the analyst dependent upon their task.

As described in Section 3.3, SZV provides several different query techniques. The query results are displayed within their context since they are highlighted documents within the clusters of an overview or analyst-defined groups. This is a strength of the system over classic search techniques (such as most online search engines), since they contain no context as to how the query results are distributed among the overall document collection and give no indication of what key entities these document collections contain.

SZV's search and highlight by date features were not evaluated in the formal study. The search options allow a search to be based on a document's text, entities and date. The search feature should be upgraded to allow regular expressions and expose more of the metadata of documents (such as their source). This will allow the same query flexibility provided by programs such as In-Spire that analysts use to refine queries by adding or removing terms (Hetzler et al. 2004). The highlight by date feature acts as a shortcut for quick searches by date. It should be useful for quickly finding patterns within the document collection, but this has not been formally evaluated.

The brushing capability within SZV is a fast details-on-demand query method integrated directly within the documents. Although brushing and linking is not a new technique, it is normally used with multiple views to highlight the same data within different visualizations. Instead, in SZV it is used within the single view to highlight the same entities within multiple documents possibly at different levels of detail. This is an alternative to connecting the documents or entities in them by edges. Brushing within a single view is rare but systems such as In-Spire (Hetzler et al. 2004) and NaviQue (Furnas et al. 1998) allow analysts quickly to highlight previously created sets of objects by using a control panel on the side of the view. In SZV, brushing can identify similar documents in a more targeted, controlled, and fluid manner than the cluster layout. Participants in both study conditions quickly learned how to use brushing to identify relevant documents. Brushing is a strength of the system only because of a combination of other features. Effective brushing depends upon the user viewing all documents at once, which was not maintained in the study's popup interface. Brushing entities also obviously depends upon the document-entity model, similar to many of SZV's features. If a person is missed in one document (not tagged as an entity), then that document will not be highlighted when the person entity is brushed. The search feature must be used instead. This is just another example of how difficult a problem data ingest is in the visual analytics domain. The CZSaw team is working on a better data ingest method that can help ensure a high quality entity model in each document collection.

SZV allows analysts to organize a document collection into groups and view the different semantic levels for a whole group. The grouping feature has not been formally evaluated but as noted above, it was fundamental in our work on the VAST challenge. Imposing groups on the document collection provides a structure that can aid analysts in managing their analysis.

In summary, Semantic Zoom View provides querying and details on demand directly within a tool for overview and organization of a document collection. The formal evaluation involved beginner users performing simple investigations on a document collection using a simplified SZV version. Participants in the alternate interface (popup condition) had more freedom in how and where they viewed document content, except this led to valuable context being covered (document glyphs). Participants using the simplified SZV took time getting comfortable controlling the zooming and these controls can be improved as described in Section 6.1. The flexible layout and entity brushing are possible due to an underlying document-entity model. The model provides structure to documents but also yields many features of SZV inaccurate when the extracted entities are inaccurate. SZV uses both colour and position to define subsets of the document collection in order to structure the analysis. SZV provides analysts with one integrated environment for all stages of their analysis from the initial overview, through iterative queries and document skimming to the organization of a document collection for supporting several hypotheses.

6: CONCLUSION

6.1 Future Work

The formal evaluation and informal use of the system have suggested many ways to improve Semantic Zoom View. This section describes the most important of these, along with suggestions for evaluative studies.

Visual analytics applications must be scalable. Due to the immense quantities of data being created daily, techniques that cannot scale to at least thousands of data items will not be useful. Thus, one goal of further development of SZV must be to handle larger document collections.

The goal of this thesis was to create a prototype in order to prove that the focus + context technique is effective before devising better algorithms for larger document collections. SZV's grouping feature helps analysts organize document collections to help deal with very large collections. The layout algorithm needs the biggest scalability improvement, since it is currently much too slow for large document collections. Instead, methods must be investigated involving multi-dimensional scaling techniques. The document-entity model should be maintained for reasons described in the previous chapter though adding entity counts might be an improvement. This would enable a vector determined by a document's entities to represent it more accurately in multi-dimensional space. After a new layout algorithm is implemented, an evaluation can be performed to investigate how useful the layout and clusters are for an analyst's tasks.

The focus + context technique should be adjusted based on the formal evaluation's participant performance and feedback. Participants had troubles quickly putting the mouse over the document glyphs; however, increasing the glyph size will reduce the number that can be shown at once. One possible remedy for this difficulty is to use a Visual-Motor-Magnifier cursor, as introduced by Findlater et al. (2010). Upon clicking, the area within a circle region is magnified, allowing the user to move the mouse over a much larger target and click again to select. A future study can compare this new cursor type to a traditional pointer for document selection.

In addition, the evaluation found some participants were not comfortable with the scroll wheel controls or the semantic zoom level granularity. They instead wished to jump to each semantic zoom level instantly. A new method of zooming could be to cycle through zoom levels using the right and left mouse buttons and animate or instantly relocate the other document glyphs. Again, a formal study should compare the new controls to the original technique.

It is essential that analysts have entity refinement capabilities within SZV so they may fix any problems with the original entity extraction without needing to remove their focus from the main view. The visualization should update immediately when these changes to the entities are made, which may include re-applying any altered brushing that will highlight more or less documents than before.

Based on feedback from the evaluation, an SZV version should be created that allows participants to resize the full-text document frames by dragging their edges instead of the current scroll wheel action (for further zooming/expansion). The scroll wheel could then be used for scrolling the text of the document. This interaction technique would fit well with the left click and right click to cycle through semantic zoom levels, since the scroll wheel would then be completely removed from controlling zooming. A formal study should then compare the resulting interface to the original.

Once many studies have been done to determine quantitatively the most effective interface and interaction techniques for SZV, case studies should be performed. Early usability studies will most likely involve graduate students as participants; however, actual analysts need to use SZV and CZSaw to analyze their real data. To begin, an analyst could take part in "Pair Analytics" sessions with a developer of the tool, as introduced by Arias-Hernandez et al. (2011). The developer would control the tool, while the analyst would suggest high-level actions. By working with the developer, the analyst would soon be ready to use the tool alone. We will never truly know how useful our applications can be until they are in the hands of the end user. Before the case studies, it will be useful to meet with some analysts and show them prototypes to get early feedback.

6.2 Summary

In this thesis, I have described the design and evaluation of Semantic Zoom View that uses a focus + context technique for providing quick drill down into a document collection overview. The technique combines ideas, metaphors and methods from related work in document overviews, focus + context techniques, and zoomable user interfaces. An evaluation was performed to compare the use of a focus + context technique (simplified SZV) to an overview + details technique for viewing a document collection. From this evaluation, valuable feedback was obtained on the effectiveness of both alternatives for finding information within a document collection. This feedback suggested many improvements in the visual analytics tool that can be evaluated by future formal comparison studies. Ultimately, Semantic Zoom View must be placed in the hands of real analysts with real data in order to determine how to meet the needs of analysts.

The main contributions of this thesis are the design of a new visual analytics visualization technique and associated interactions, and an evaluation of the technique. This thesis revealed some advice for development of similar tools. First, support direct manipulation including drill down and queries directly within visualizations, but take special care to design simple interaction techniques that do not directly conflict with familiar conventions. Analysts should have quick access to information, but the way in which they access it should also be intuitive and easy to remember. Second, the method of drill down into an overview, whether an overview + detail or focus + context approach, should maintain visibility of the entire overview while showing the detail. This is especially important if some form of brushing and linking is applied between the focus (detailed) content and the overview. Third, a clear data model with useful categories of data (such as entity types) can lead to great flexibility in an overview if the user can not only filter the resulting visualization but also filter the categories of data that are used by the layout algorithm. This thesis described recommendations for future development of Semantic Zoom View, which should be compared to the current version through formal evaluations to iteratively develop a visual analytics tool for analysts.

APPENDICES

Appendix A: Technical Description

(Dunsmuir 2009)

The University of British Columbia - Fall 2009

CPSC 533: Information Visualization

Final Project Report

Selective Semantic Zoom of a Document Collection

Dustin Dunsmuir

University of British Columbia

Abstract

Analysts are challenged to make sense of huge document collections containing text that cannot be easily summarized. Text analytics can help uncover relationships within the data but there is a need for visualizations which smoothly integrate an overview of the document collection with the details of these relationships. This overview should allow the analyst to organize the document collection as their investigation progresses. I introduce the Semantic Zoom View which is designed to do all of the above through the use of nesting entities within documents and using a selective semantic zoom. This zoom reveals the details on demand of a document while keeping the context of the document collection. This context, which is present in the same view as the details, can be organized quickly by the analyst.

Keywords: Document collection, semantic zoom, hierarchical layout, focus+context.

1 Introduction

Techniques within the field of Information Visualization have been used for intuitively visualizing attributes of data and aggregations of data. Many techniques excel at showing the whole picture which is an essential task with the increasing size of datasets. Unfortunately this task is made much harder when the data is not numerical such as with news articles or intelligence reports. Intelligence analysts often take on the challenging task of making sense of a large collection of such documents. Analysts are interested in the activities of certain people or organizations mentioned within the documents so entity extraction systems have been developed that automatically extract the key people, places, dates, etc. from a large document collection (Calais [4] and MALLET [10] are two examples). Once this process is complete the result is a set of entities contained within each document. Each entity also has an entity type such as person, place or date. Then the visualization techniques can focus on visualizing the relationships between these entities and the documents, rather than the full text of each document. The full-text is still important but it is usually only accessible within a separate window.

Applications for sense-making across text documents typically involve multiple views for different perspectives of the document collection and different levels of detail. In his review of overview+detail, zooming and focus+context displays, Cockburn found that disadvantages with the overview+detail (a form of multiple views) technique is the additional use of screen space and the added time and mental effort required by the user to integrate the information from the views [5]. Focus+context displays allow the user to see all the information seamlessly within one view and in multiple focus point systems the level of detail can be adjusted at many points across the view. In addition, semantic zoom is a technique in which the user sees a different representation of the data at different zoom levels. As Chris

Weaver puts it, "*Semantic zoom is a form of details on demand that lets the user see different amounts of detail in a view by zooming in and out*." [21]. These two techniques can be combined to allow quick access and re-access of detailed information directly within a visualization without losing the context.

The main contribution of this paper to the field of Information Visualization is to introduce the design of a new visualization technique for getting an overview of a document collection, inspecting the details of each document and organizing the documents all within one view. The implementation of this design is called the Semantic Zoom View. It uses a selective semantic zoom similar to the multiple focus point fish-eye views of previous work [2, 15, 19] and applies it to this field of sense-making across text documents. Entities are nested within documents to intuitively illustrate that they are mentioned within the document.

The Semantic Zoom View will become a part of the CZSaw system [9]. This system is a multi-view application designed for sense-making across text documents. The main focus of CZSaw has been on capturing and supporting the analysis process through an underlying script of all actions, a history view generated from the script and a dependency graph that preserves the dependencies of the variables created in the analysis and allows quick propagation of changes. My efforts as part of the CZSaw team have been to develop hybrid visualizations within CZSaw which allow the analyst to focus on a single powerful and flexible visualization. The Semantic Zoom View is one of these visualizations.

For the purposes of testing the Semantic Zoom View, I have used the VAST contest dataset from 2006 called Alderwood [7]. The screenshots throughout this paper show this dataset. The documents are news articles from the fictional town of Alderwood, Washington. Each article consists of paragraphs of text and is the typical length you would expect from a newspaper article. The entities within each document are usually 1-3 word phrases. Each news article also has a name and the date it was written. Unfortunately within the Alderwood dataset the name is not the news article name, but rather a unique number.

2 RELATED WORK

In this section, past applications for sense-making across text documents will be discussed and compared to the current technique. Then some layout algorithms are described which are similar to those implemented within this project.

2.1 SENSE-MAKING

The Jigsaw system [17] is a visual analytics application designed to be used by intelligence analysts for sense-making across text documents. It provides multiple views each designed to emphasize a specific aspect of the documents and entities, but each within its own separate window. The CZSaw project's main data views are based around those views present in Jigsaw [9].

The view within Jigsaw most used to see connections between documents and entities is the Graph View. While the Graph View shows the "document contains entity" relationship with an edge, the Semantic Zoom View shows this relationship through actual containment of the entity glyph within the document glyph. This should be more intuitive than an edge. The Graph View, however, only contains one copy of every entity whereas in the Semantic Zoom View each entity is contained once within each document that it occurs. The zooming ability as explained in this paper means this repetition does not lead to a large use of screen real estate.

One view in Jigsaw but not yet in CZSaw, which has some similarity to this visualization, is the Document Cluster View. This view has two tabs, standard and group, and each visualizes documents as small coloured rectangles. The standard tab shows one rectangle for every document and colours these documents according to filters created by searches. The documents can be moved around the view and there seems to be no limit to the number of colours available. When a new filter is created a new colour appears for the filter. The group tab has the same functionality except that it also groups the documents by filter and has multiple copies of documents that match more than one filter. Although these documents can be moved around they are put back to their original locations the next time a filter is added. In contrast the Semantic Zoom View combines both the colouring and grouping into one view along with the ability to zoom in to see the details of each document. In Jigsaw another view must be used to see the relationships of the entities within a document (Graph View) or read the text (Document View). Starlight [13] also offers the same representation in which colour coding can be applied to the documents based on any values of the entities. Again, reading the text or getting any other details besides the filters matched is only possible in another window.

IN-SPIRE [12] offers the Galaxy View where each document is represented by only 4 pixels and situated in space dependent on the keywords within it. Clusters of documents are shown around common keywords. Documents can be filtered from the view and then the layout recomputed. This layout is not present currently within the Semantic Zoom View but will be added in the future. The IN-SPIRE system allows the user to view the document text in another window.

The design discussed in this paper differs from the above applications by the use of a zoomable user interface to embed the document details within the same view. To enable this, an algorithm is required to handle the selective zooming.

2.2 Zooming Layout Algorithms

Various research projects have investigated the use of focus+context with semantic zooming acting as a fisheye view. The main idea is to allow the user the ability to quickly zoom any part of the view to see the details while smoothly adjusting the rest of the view. One of these techniques is the Continuous Zoom developed by Bartram et. al. [2] that can be used on hierarchical graphs. A related method called variable zoom was used in a study done by Schaffer et. al. [16] involving subjects navigating a simulated telephone network. The fisheye view was compared to a full-zoom view and found to be faster to use and for some tasks allowed better performance. Eleven years later the ADORA system was developed by Reinhard et. al. [15] which built upon many of the features of the Continuous Zoom algorithm to make an improved fish-eye zoom algorithm that was more flexible and easier to reverse. The ShriMP system has also been developed for looking at nested graphs (software architecture) [18] and like the above algorithms is designed for adjusting a graph given the zooming of a node [19].

These algorithms have the goal of maintaining the user's mental model by changing the view as little as possible, but still making sure there is no occlusion. They also have the goal of being able to reverse the zooming operations to get back to the original layout. The ADORA method outperforms the other in its flexibility in being able to restore the view but this is not surprising as it was designed many years afterwards. The SHriMP algorithm may have an advantage over the other two in that it works in both dimensions simultaneously while the others use interval structures along the X and Y axes. This means they suffer from many small documents being within the projected shadow of a larger one. All of these algorithms also involve shrinking other items to give screen real estate to the one being expanded, a side effect that I wished to avoid in the Semantic Zoom View. In the SHriMP algorithm this rescaling is an optional last step so for these two reasons the algorithm used in this project is most similar to SHriMP. In Section 4.2 the details of the algorithm and its differences from these is explained.

3 Design

This section will describe the design of all the functions of the Semantic Zoom View while emphasizing how they are tailored to meet the perceptual abilities of humans. Then Section 4 provides details on the implementation of the system.

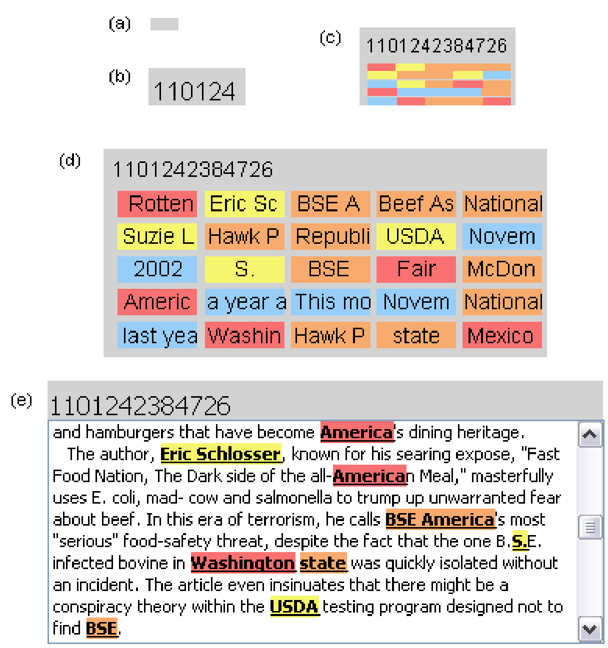


Figure 1. The semantic zoom levels: (**a**) Fully zoomed out: represented as a small rectangle. (**b**) Name only: A rectangle labeled with the document name. (**c**) Entities shown: Same as above but now containing other rectangles (one for each entity, coloured by type). (**d**) Entity names shown: Same as above but now the entity rectangles are labeled with their value. (**e**) Full Text: A small window into the full document text which is scrollable to move around within the document.

The semantic zoom of a document is the central component of the view and is what makes it unique within this problem domain. The view initially displays all documents zoomed completely out so that each is represented as a small rectangle of 50 pixels. Any document can then be zoomed independently of the rest of the document collection. The size of the document increases smoothly during the zooming but there are five semantic zoom levels at which the detail within the view differs. These levels are illustrated in Figure 1.

These levels of detail enable an efficient use of space for displaying document details. As a document is zoomed, it not only takes up more space on the screen but also displays more detail in that space. When zoomed out a document has a very small size but also shows very little information, while a document fully zoomed in is at its largest size but also shows the most information.

Unlike a traditional zoomable canvas, each document is zoomed independently of its surroundings. That is the other documents do not change their level of detail or size. To prevent occlusion the surrounding documents are moved outward when zooming into a document. Since other documents are not zoomed out, the analyst can still see all of their current detail provided the screen is big enough. After the analyst has read a document or seen enough of a document, they can easily zoom it out to provide more room for other documents. The zoom out operation reverses the movement of the other documents from before so that each will return to their original location. This moving of documents back inward reduces the screen space used that is important when dealing with document collections containing more than a thousand documents. More importantly the result is that the original layout is recreated, as it is desired to have the least impact on the analyst's mental model of where documents are located in the view. The view allows the analyst to move documents around the space in order to organize them and so the layout they create should be maintained as much as possible.

The method of zooming into and out of documents quickly enables the analyst to start an investigation by reading some of the documents and looking for any suspicious activity within them. It is quite possible; however, that the analyst already has an idea of what they are looking for. For this reason, there is a search feature within the view. With this feature the analyst can search for a string of text within the full-text of all reports or only within a specific entity type (only people or only places, etc.). The result of a search is that all of the documents matching the query are highlighted within the view.

Brushing and linking across entities can also be done in the view. An entity will be repeated within the view in all documents it is contained within although at any given time it is likely that the majority of these documents will be zoomed out so the entity cannot be seen. By clicking on an entity, all other documents that contain this entity will be highlighted and the entity will be highlighted within them. Document highlighting is done in the view's active colour which is explained below. The scenario of Section 5 features searching as well as brushing and linking.

3.1 DOCUMENT ORGANIZATION

The large number of documents in a collection clearly means an analyst does not have time to read all of the documents. Thus one goal of the system is to allow the analyst to quickly organize the collection into those documents that are relevant and those that are not. They may also wish to create several distinct groups of documents relating to different parts of their investigation. The Semantic Zoom View provides three main methods for visually distinguishing a set of documents from the rest. These are highlighting, clustering, and grouping.

3.1.1 Colour

Within the view, colour is an important visual channel used to make a set of documents stand out from the rest of the collection. Colour is preattentively processed and so all of the items of one colour can pop out to the analyst. Unfortunately as the number of colours increases the pop out effect decreases substantially [20]. Thus in the Semantic Zoom View there are only five colours an analyst can choose from for selection and highlighting search results. The palette was chosen from the Color Brewer [3] website among those that are distinguishable by colour deficient people and these colours were also checked using the Vischeck website [6]. Two different shades of green and two different shades of blue were chosen so that the different shades may be distinct from each other but still used for two groups that contain documents more similar to each other than others (as decided and organized by the user).

One colour is always the active colour within the view. The active colour can be changed at anytime (using the drop down menu) and the current active colour is what is used to highlight the results of a search. The active colour is also used when selecting documents in the view by clicking on them or using a rubber band rectangle. In essence, selecting and highlighting documents are one and the same in this system as they both add to the set of coloured documents. To deselect all documents of the active colour the analyst clicks on the whitespace within the view. When this happens all the documents that were previously a different colour are reverted to that colour rather than becoming the default unselected grey. This memory of the highlighting of a document allows an analyst to quickly reverse the action of a search and as done in the scenario of Section 5 it can enable them to find documents matching multiple queries.

Entity highlighting does not directly use colour. Any entities which match a search query are outlined in black while the document is highlighted in the active colour. This is due to the fact that the entities are themselves colour coded by their entity type and although this palette also meets the requirements of the Vischeck site, the combined scheme does not. This is why if a document is zoomed in enough to show entities, highlighting of the document is done through changing the border colour rather than the background. This keeps the palettes separate. Another reason for only highlighting the border when the document is zoomed in is because I wish to avoid having large areas of saturation as they stand out far too much [20]. The entity type colour scheme is used throughout the CZSaw system and is specific to the dataset as different datasets contain different entity types. With the Alderwood dataset there are 6 different entity types and those that are similar in meaning (for example date and time) were given the most similar colours (blue and purple) although these are still distinguishable by everyone.

Colour is useful for highlighting when trying many searches as it can be easily reversed by simply reverting to the default colour. It can also be used when documents are already located in a meaningful location, but to more permanently mark a set of documents spatial position should be the number one choice.

3.1.2 2D Position

2D position has been found to be preattentively processed [20] and also is perceived more accurately than any other visual channel (such as saturation, shape or area) for quantitative and qualitative data [11]. Thus, within the Semantic Zoom View the analyst may move one or more documents to a new location by the normal click and drag or rubber band and drag method. When documents are moved they may be placed in such a way that they overlap other documents. In this case the other documents are moved to remove the overlap as explained in Section 4.2. In this way, documents can be quickly moved around the screen without causing occlusion. However, in order to quickly organize the document collection more advanced methods are needed than simply translating all highlighted documents by the same vector.

Figure 2. The group operation first performs a cluster but then also places the items within a named rectangle that can be moved, resized or closed. (**a**) The view before performing the grouping of all green documents. (**b**) The view with the named group added to it.

For example, when the highlighted documents are spread across the view with large space between them (as is likely from a search), translating the set of documents inevitably leads to some of the documents moving off screen. Thus it is desired to move all the relevant documents closer together. The cluster feature within the Semantic Zoom View is designed to do just this. To cluster the set of documents highlighted in the active colour, the analyst clicks the cluster button and then clicks a location to cluster at. Then the active documents are moved such that they cluster around the point but maintain the same relative positioning between each other. It can be seen as a scaling down of the original layout of these documents with a gravity force applied to pull every document towards the cluster point. More information on the clustering algorithm is present in Section 4.2.

The group function in the view allows the analyst to more strongly distinguish a set of documents from the whole collection. A document group has a name and a bounding rectangle within the view as shown in Figure 2. To create a group for a set of highlighted documents, the analyst one again can choose a location within the view. Then the documents are clustered together and the analyst is prompted for a name for the new group. The documents are then shown within a rectangle with the name of the group at the top. To save space in the view this group can be closed which hides the documents, showing only the name of the group. Unlike a cluster, documents in a group do not have to be selected to move together. Instead the group may be clicked and dragged to move to a new location. Each document appears only once within the view, so documents may only be part of a single group. In contrast, two clusters may be placed close to each other with some documents near both. Once a group is established documents may be easily added or removed from a group by dragging and dropping them inside or outside the bounds of the group. Groups are also resized based upon changes made to their contents such as zooming in on a document.

To provide an overview of the entire document collection all documents are zoomed completely out in the initial layout. They appear all the same size in a grid layout which is ordered by the document date if one is present in the dataset. As of this writing, one layout has been developed to provide an overview of the document set. This is the date layout which goes beyond the normal grid by showing documents in a calendar format for each month of each year. This layout is currently for ungrouped, zoomed out documents as it rearranges the position of all documents and assumes they are all the same size. The date is taken from the metadata of the document, rather than date entities within it, so each document has a single date; however it is quite possible that multiple documents have the same date. Thus the date view stacks documents with the same date diagonally. This leads to occlusion but since the documents are zoomed out there is no loss of information and large stacks can quickly be seen representing those days that have the most documents. The date layout can be used by the analyst to find weekly patterns of highlighted documents (as in Figure 7 of Section 5) or seasonal patterns as the summer of each year appears directly below the summer of the previous year. A similar technique of lining up dates to find temporal patterns was found useful in the hotel visitation visualization created by Chris Weaver in Improvise [22].

In addition, although this layout rearranges all the documents, the previous layout can be instantly re-obtained. At the same time those documents that were moved while using the date layout are not returned to their original location. This allows an analyst to pick out a set of documents around a given date but still have all other documents return to their previous location when the date layout is turned off.

4 IMPLEMENTATION

I have created the Semantic Zoom View as an independent Java application although in the future it will be a view within the CZSaw system. As such, the data query methods (with MySQL) were taken from CZSaw rather than being re-implemented. How the results of the queries are displayed in the view and all of the visualization code were written specifically for this project and use the Zoomable Visual Transformation Machine (ZVTM) Java library [14].

4.1 USE OF ZVTM

Figure 3 illustrates how the visual components of the ZVTM library are connected and used within the Semantic Zoom View. The ZVTM library allows the creation of infinite canvases called virtual spaces which can contain a variety of glyphs. A virtual space can be seen with a camera which can focus on different areas of the space and can zoom in or out of the space. Finally the image of each camera is connected to a view which is shown in a panel on the screen. For the Semantic Zoom View, a glyph is created for each rectangle and the document and entity labels; however, each document is added to a separate virtual space. This is so that they may be zoomed independently. Thus there are many virtual spaces, each with its own camera and view, but these must all be displayed in the same panel.

To accomplish this, the ZVTM portal object was used. A portal is an inset in the panel that has bounds and its own camera connected to a virtual space. The example of a portal used in the ZVTM documentation is that of an overview map that sits in the corner in a map application. Thus it does not appear to be originally intended to be moved around the screen. Some work was needed to accomplish this, although ZVTM provide a listener for when the mouse enters and exits a portal. There was also no built in functionality in ZVTM for having a portal change size automatically in response to changes of the camera or the glyphs on the virtual space being viewed. Thus to keep the camera only viewing the document within a portal some calculations were necessary to resize the portal as a document is zoomed.

The semantic zooming of a document is done by changing the visibility of glyphs depending on the new altitude of the camera. Additionally some changes of size on the virtual space are also needed to smooth the transitions. For example, the background rectangle of a document is changed in aspect ratio from showing just the label to fitting all the entities. Other than this, most changes in size seen within the view are due to a portal's camera zooming in on the document.

I extended the compound glyph class of ZVTM to create a class for a document and a class for an entity. I also extended the portal class to create an abstract zoom portal which was extended for a document portal and a group portal. There is no support in ZVTM for portals within portals as a portal is always directly on the panel. Thus the group feature is implemented by a portal that is drawn before the document portals are drawn in front of it. All of the glyphs within the view are created when the application is first run so that they may be available when any document is zoomed in. This means that the only wait time for the building of glyphs is on startup. For the Alderwood data set, which contains 1,182 documents and 13,356 entities, it takes 15 seconds to load the view in Parallels using 1.4 GB memory on a MacBook Pro. This time includes not just creating the glyphs but also connecting to the database, performing the queries necessary to get all the documents and displaying the user interface.

Also provided by ZVTM was a mouse listener for actions on the panel. The methods I implemented using this listener in the view are for mouse clicked, moved, pressed, dragged, or released actions as well as scrolling with the mouse scroll-wheel (or trackpad). These allow the analyst to interact directly with the documents in the view using the mouse to do such things as zoom (scroll-wheel), select and move documents.

The ZVTM library has support for animations of properties of cameras, portals, glyphs, etc. I used the translation portal animation to animate the forming of clusters and groups.

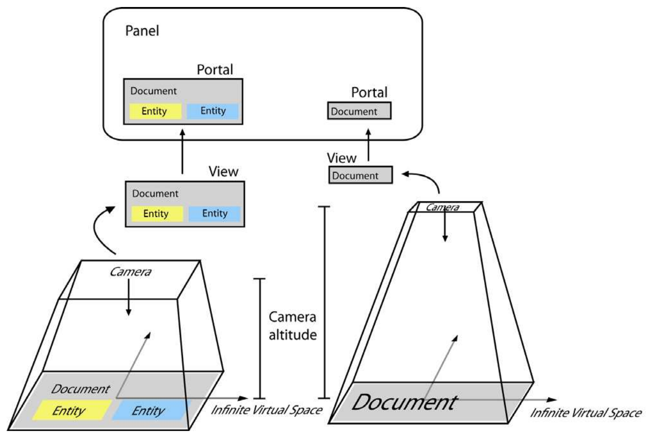


Figure 3. How the visualization model of ZVTM is used within the Semantic Zoom View. Changing the camera altitude causes a document to appear smaller within its portal as defined by ZVTM. Then I have added the methods to make the portal change size to show only the document and the code to make the zoom semantic by changing the glyphs visible.

4.2 ALGORITHMS

In order to work easily with documents at multiple zoom levels, and thus at multiple sizes on the screen, three different algorithms were developed. The first algorithm is used to move documents and groups in response to the zooming in or out of a document. The second algorithm removes overlap caused by moving one or more documents and groups. The third algorithm determines the location of documents when a new cluster is formed. Since this project's main focus is on the design of this new data view, the following descriptions of the algorithms will be kept relatively brief.

The Continuous Zoom [2], ADORA [15], and SHriMP [19] algorithms were mentioned in the related work section. These algorithms are applicable because they also involve zooming into items and the changes that are made to the rest of the view as a result. I started by implementing the SHriMP algorithm. Unfortunately with all the resulting white space and opting not to automatically rescale other documents (so zoom levels stay independent), the layout expands quickly. Thus, I first sort all the other documents by their distance to the focus document (the one being zoomed). Then I move each in turn according to the SHriMP algorithm only if they are occluded by one of the ones already moved. The resulting layout starting from a grid is shown in Figure 4. This logic works for zooming into a document (making it larger) but does not work for zooming out since no occlusion occurs. Since all the documents start zoomed out, I simply store whether a document was affected by the zoom, in order to move all the same documents again for the zoom out. This data is needed for each zoomed document although it is cleared when the document is manually moved since it no longer applies. The time the algorithm takes to run when zooming in is $O(n(n+1)/2)$ and when zooming out it is $O(n)$ where n is the number of ungrouped documents plus the number of groups. The original SHriMP algorithm runs both ways in $O(n)$. This variant of the SHriMP algorithm runs in under a second when working with the Alderwood dataset.

This modified SHriMP algorithm improves the compactness of the layout over the original algorithm but lacks some of the mental map saving properties of the original as seen in Figure 4. The original SHriMP algorithm preserves orthogonal orderings and proximities between nodes while this variation does not preserve either of those properties. The original layout however adjusts the entire view even if there are many disjoint parts while this new algorithm completely preserves clusters of documents that are disjoint from what is being zoomed as long as there is space. Currently in the Semantic Zoom View the initial layout is far less meaningful then any clusters formed through the process. Thus I argue that less use of screen space and maintaining the exact position of already sorted documents are much more important then minimizing the distortion of those documents surrounding a zoomed document, that are yet unsorted. Ultimately an experiment must be performed in the future to study this trade off.

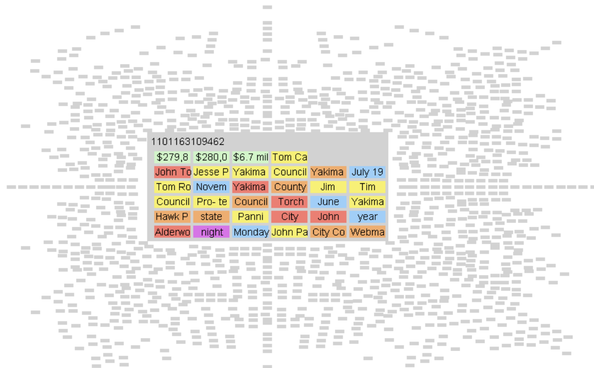


Figure 4. The layout that results from the selective variant of the SHriMP algorithm which the Semantic Zoom View uses.

When one or more documents or groups of documents are manually moved a different algorithm is used to remove any overlap that this causes. Once again no items are changed in size. They are only moved so most likely this causes the overall bounds of the document layout to grow. The reason that a different algorithm is used here is because it is no longer as important to support a reversal of this action. An analyst is much less likely to move a document across the view and then back again then they are to zoom in and then back out. Thus an implementation of the Force Transfer Algorithm is used [8]. The speed at which this algorithm runs is more dependent upon the number of overlaps then the total number of documents but the worst case is $O(n^2)$ where n is the number of ungrouped documents plus groups in the view. This number should be fairly small since any overlaps will have occurred directly from the last move of documents. The algorithm also does not need to run quickly multiple times (unlike the zoom algorithm) because it is only applied when the analyst finishes dragging the documents and groups to their new location.

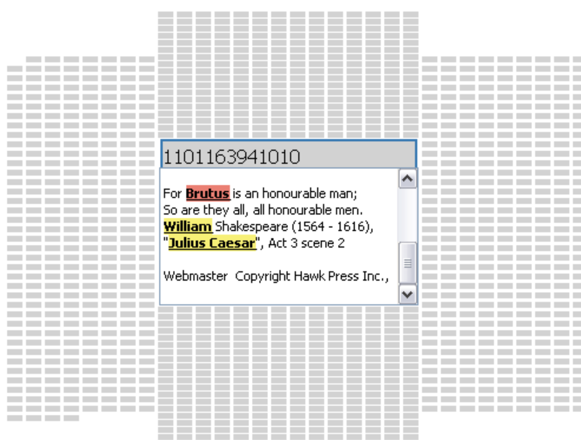


Figure 5. The layout that results from the force transfer algorithm after placing a zoomed in document into the center of the grid. In the future the documents will be displaced in all four directions rather than just up and down.

This Force Transfer Algorithm is more effective than the SHriMP algorithm at keeping changes to the layout to a minimum; however, it takes longer to run. The main problem with the current implementation of the algorithm occurs when a large document is placed over many small documents. The algorithm fails to move all of them in the direction that minimizes the distance moved. Figure 5 demonstrates an instance of this problem where some documents should have been moved left or right. I have worked out a solution that will increase the complexity of the algorithm but make it more effective. The details of the solution are beyond the scope of this paper.

The third algorithm was used for the clustering to move documents from across the view to be clustered around one location except still in the same relative position. Initially when the cluster feature was designed the main goal was to collect all highlighted documents of one colour to a specific location and so they were packed into a new grid. There are two problems with this. The first is that their original layout is completely lost. Some of the documents may have been already clustered or in a date layout so it is desired not to completely destroy these encodings. Secondly, the documents may be at different zoom levels which means they are different sizes within the view. While a decent solution for the packing rectangles of different sizes into a larger rectangle is not difficult, finding the minimum bound of the layout needed is an NP-hard problem [1]. Thus I allow some white space and constrain the problem by attempting to keep the documents in the same relative position. The first step in the clustering is to translate every involved portal (document or group) by the same vector so that the set is centered on the cluster point. I then order the portals by their distance to the cluster point. Then in turn each portal is moved inward along the line connecting its location to the cluster point until they can no longer move because they would occlude a portal already moved. This final position is where each portal is animated to from the original position. This algorithm gives the appearance of a gravity point that all involved documents and groups are sucked into. I use the same algorithm for the clustering involved in forming a group except that all selected groups are ignored since currently groups cannot contain other groups. The algorithm runs in O(n(n+1)/2) where n is the number of documents or groups being clustered.

In the clustering algorithm some spaces still exist between documents depending on how they were originally positioned in the view. If in the future it is determined that tightly packing the clusters is more important than resembling the original layout some random jittering of positions could be added. Then reapplying the algorithm could reduce the space used.

Groups add another level to the zoom algorithm and must be considered when determining what to move within the other two algorithms. When a group is moved, all of the documents within it are moved as well. Thus if the document being moved or zoomed is not within a group then the zoom and move algorithms only consider groups and documents outside of groups, ignoring those documents inside groups. The cluster algorithm also ignores these grouped documents. If the document being zoomed is inside a group then the zoom algorithm is first applied only to the documents in the group, then the bounds of the group are adjusted and it is applied again on all the ungrouped documents and groups based on the change in bounds. In this way, groups make the zoom algorithm multi-level.

5 Scenario

Now that the current state of the system has been fully explained I will narrate and illustrate a scenario that an analyst would take within the system. The goal is to show that someone quickly trained with the system (perhaps simply by reading this paper) can carry out the organization of a document collection and narrow down their investigation and sense-making process to the more relevant documents. The full task of an intelligence analyst of discovering plots or suspicious trends is not an easy or quick task.

The VAST contest with the Alderwood dataset was to determine if any inappropriate activities were happening in the town of Alderwood, so there were no real clues as to where to start the investigation [7]. Thus after loading the view, I begin the scenario by simply zooming (using the mouse scroll wheel) into the first document to read it. This document is the oldest news article as they are ordered by date. It turns out this article is just about the weekly lottery numbers, something not useful to my investigation, so I can filter it out. However I should first determine if there are any other articles about lottery numbers. To do this I search for “lucky numbers”, a string that appears in the document. All the documents containing lucky number appears highlighted in the active colour (Figure 6).

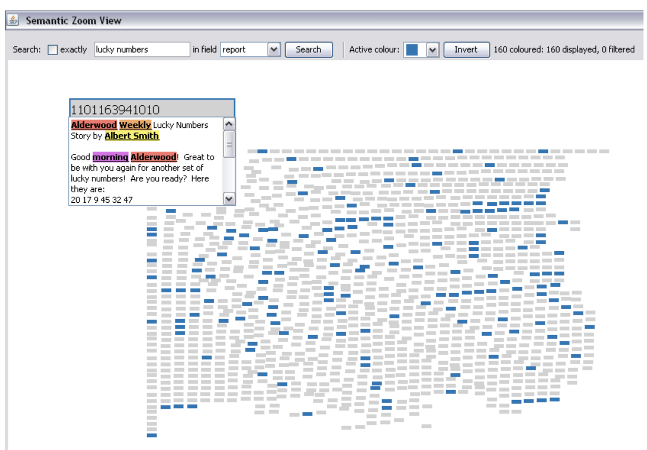


Figure 6. All 160 “lucky numbers” documents are highlighted in the view by a search.

According to the text at the top of the view there are 160 documents containing “lucky numbers”. To confirm that these are probably all about the weekly lucky numbers I zoom the document back down and then switch to the date layout using the drop down menu. Immediately I see that a highlighted document occurs once a week on the same day each week (Figure 7).

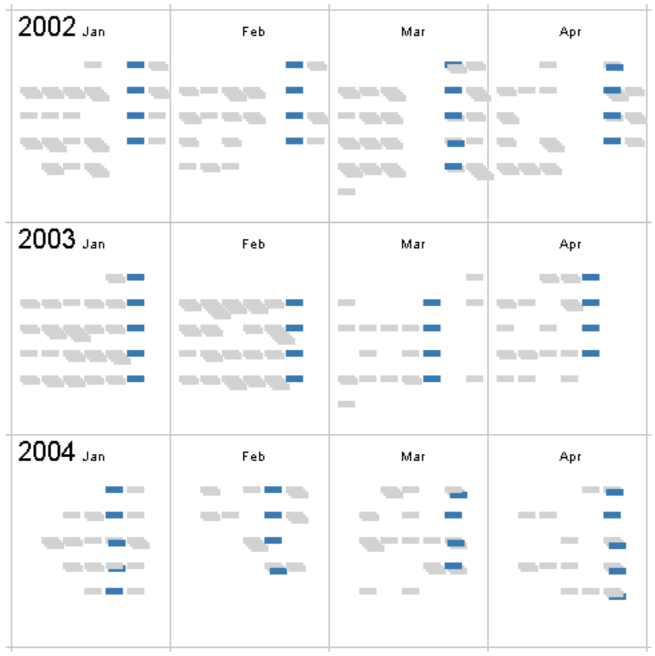


Figure 7. A portion of the date layout when looking at the temporal pattern of the lottery number news articles.

Switching back to the normal layout I then filter out all the highlighted documents. I then zoom in the second document and read it. It is about the finding of mad cow disease (BSE) within one cow recently shipped from Canada. The document seems to be about breaking news and it is the second document in the collection so perhaps there is more on this topic. "BSE" is an entity within the document so clicking on it in the full text performs brushing and linking by highlighting all those other documents containing it (Figure 8).

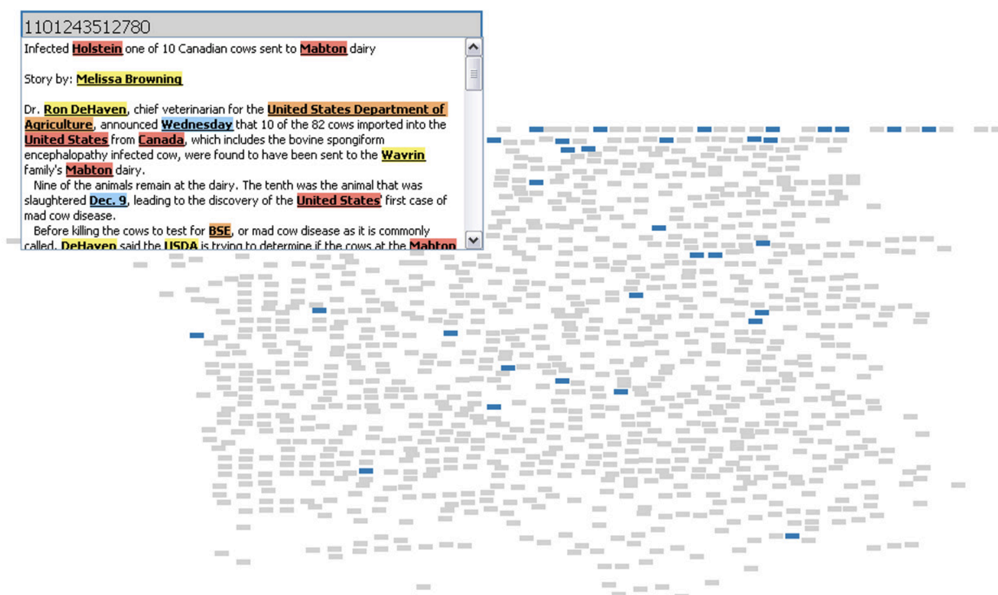


Figure 8. Brushing and linking on the BSE entity within the document's text highlights all those documents that contain it.

One of these is in the bottom right corner indicating it is one of the last documents in the dataset. Zooming into this document I find out by reading it that it is an article strongly criticizing a magazine article titled "America's Beef is Rotten and Washington Couldn't Care Less" (Figure 9). Although this document claims that the one BSE infected cow found in Washington state (from the previous article) was quickly isolated, it also mentions that the author of the magazine article "insinuates that there might be a conspiracy within the UDSA testing program designed not to find BSE". This sounds worthy of investigating, but there is more information about the BSE in the first document.

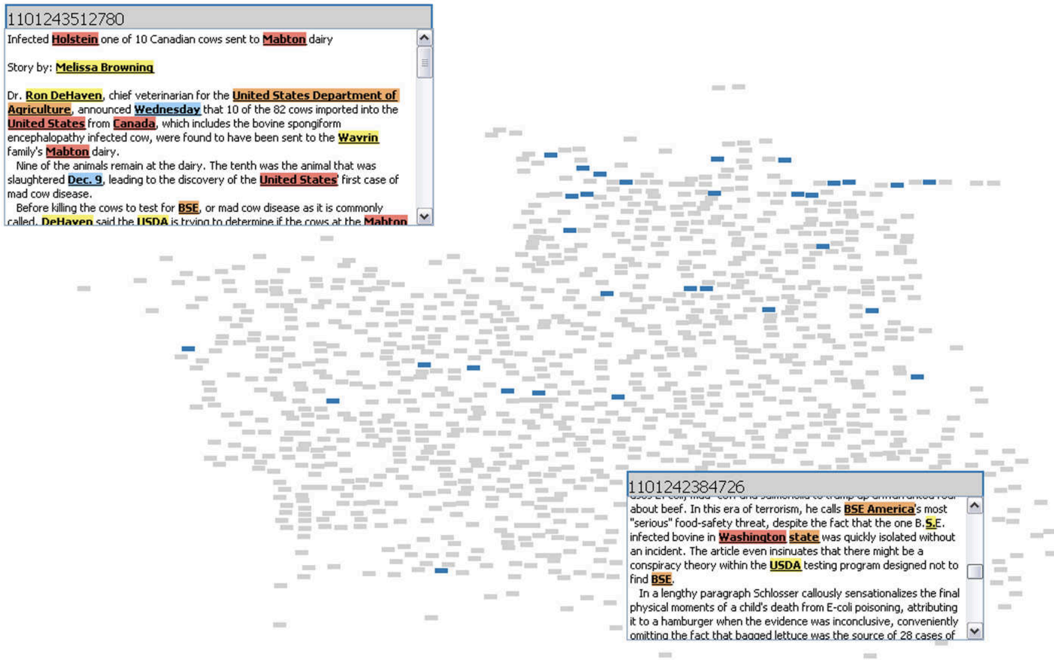


Figure 9. Investigating the last document that mentions BSE and finding mention of a conspiracy.

Thus I zoom this document out and consider the first document which lists many people and organizations involved in the USDA investigation. By clicking on each of the these entities in turn I can see how many documents they occur in and I can spot documents containing multiple entities by changing the colour and noting documents that change colour. I already have the BSE documents highlighted in blue and so I switch to yellow and click the USDA entity before switching to green and clicking DeHaven (Figure 10).

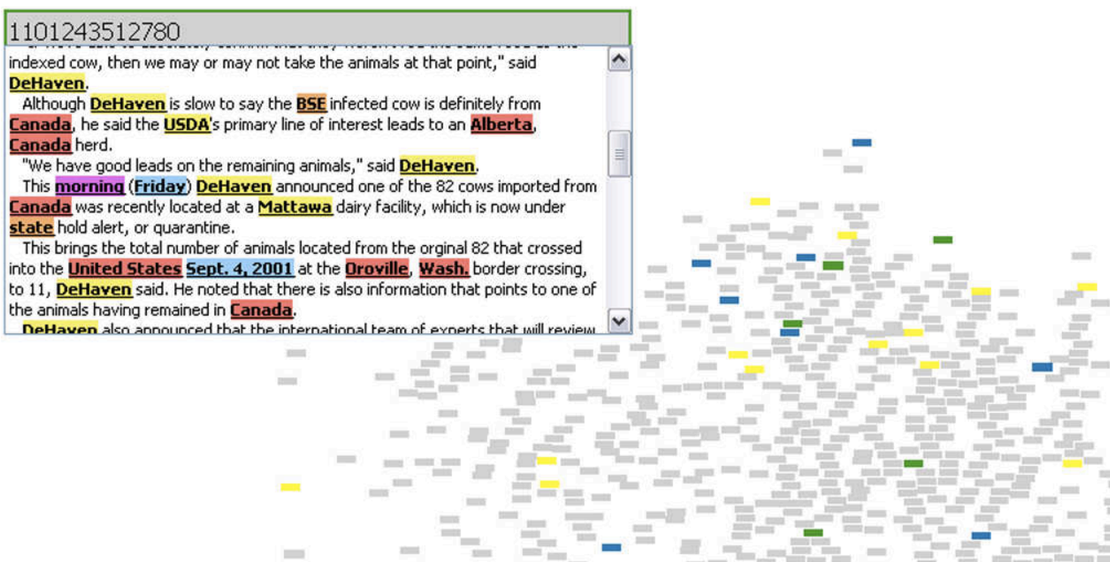


Figure 10. Through brushing all DeHaven documents are highlighted green, USDA documents that don't include DeHaven are highlighted yellow and BSE documents not including the other two are highlighted blue.

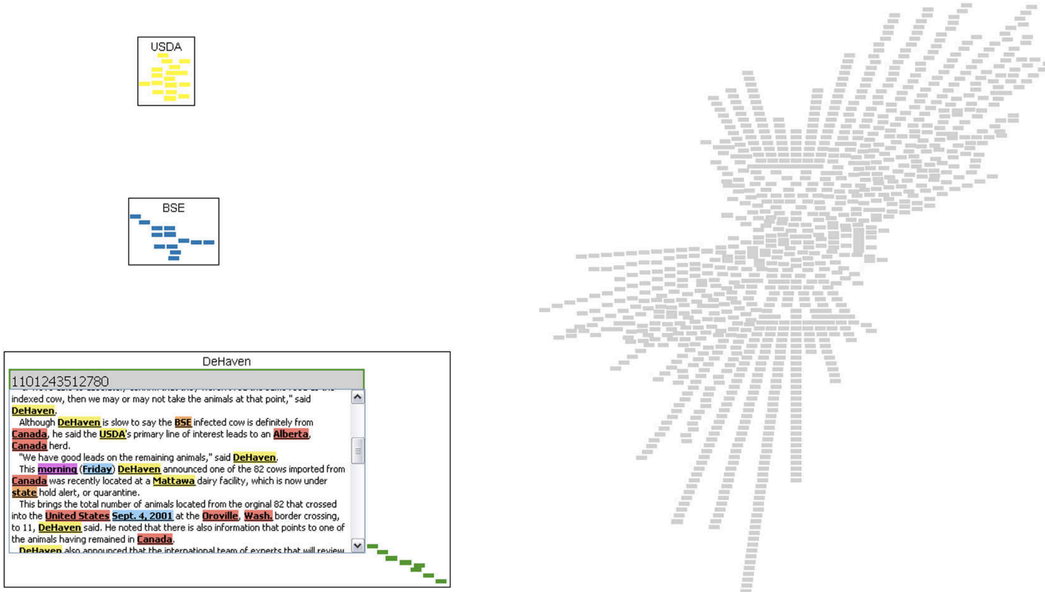


Figure 11. The DeHaven, USDA and BSE documents grouped and the rest of the documents clustered on the side.

At this point I decide to do some grouping in order to organize the documents and more strongly emphasize those documents worth viewing. In turn, I select each of the colours used so far, blue, green and yellow and group each set of documents, naming them after the entity contained in them. To organize the view further I select all the uncoloured documents in a fourth colour and cluster them away from the documents I am focused on. Then I deselect them (Figure 11).

Now I deselect the green DeHaven documents to discover that in fact all of them mention either BSE of UDSA as they are all yellow or blue (Figure 12). Perhaps DeHaven is not an important character as he is never mentioned without BSE or UDSA.

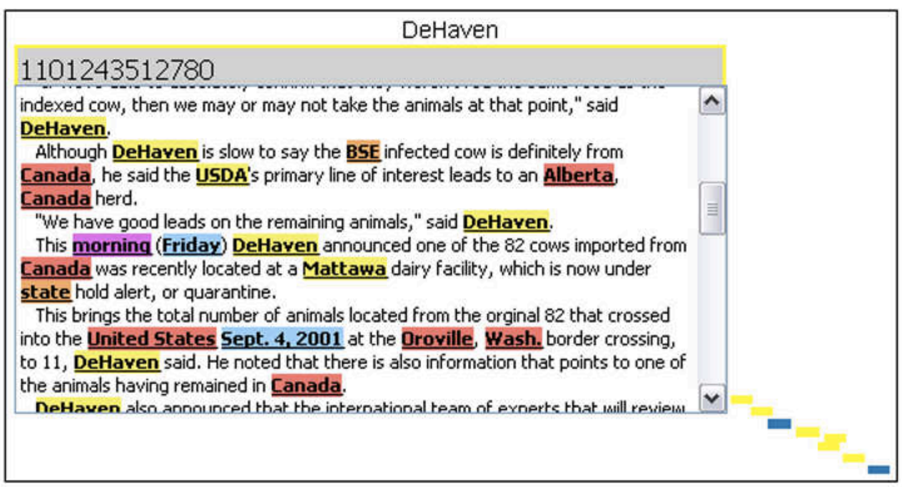


Figure 12. When green is unselected it is revealed that all the DeHaven documents are part of the other two searches.

Deselecting all yellow documents shows that all of the DeHaven documents mention BSE and most of the USDA documents do since they are also highlighted in blue (Figure 13). Perhaps the USDA documents not referring to the BSE should be investigated.

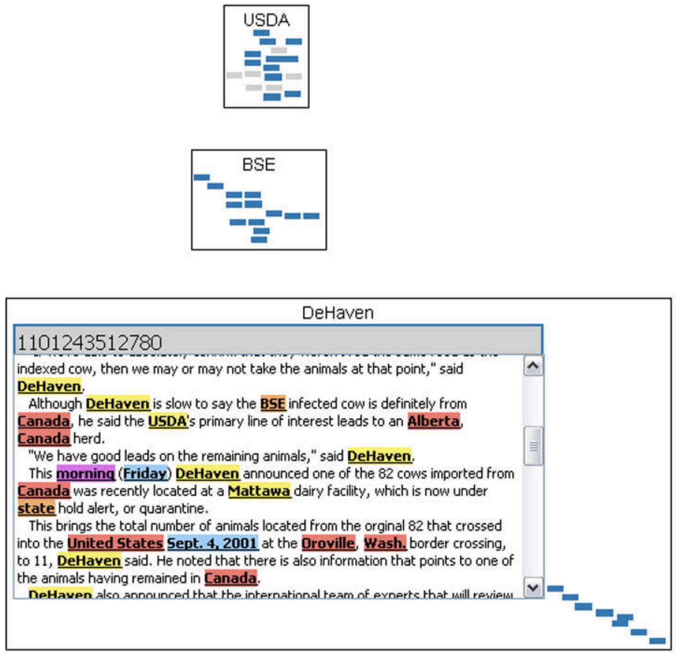


Figure 13. All the DeHaven documents are also documents containing BSE (all blue) and most of the USDA documents also mention BSE.

Based on these observations, I now decide to combine the DeHaven group with the BSE group. I select the DeHaven documents in another colour temporarily to drag and drop them into the BSE group before I delete the DeHaven group. Then I zoom in on one of the USDA grey documents. The story mentions a congressman named Doc Hastings who stopped for a short visit in Alderwood. Clicking on his name with active colour yellow highlights all the documents he is in which include 2 other USDA documents and a BSE document (Figure 14).

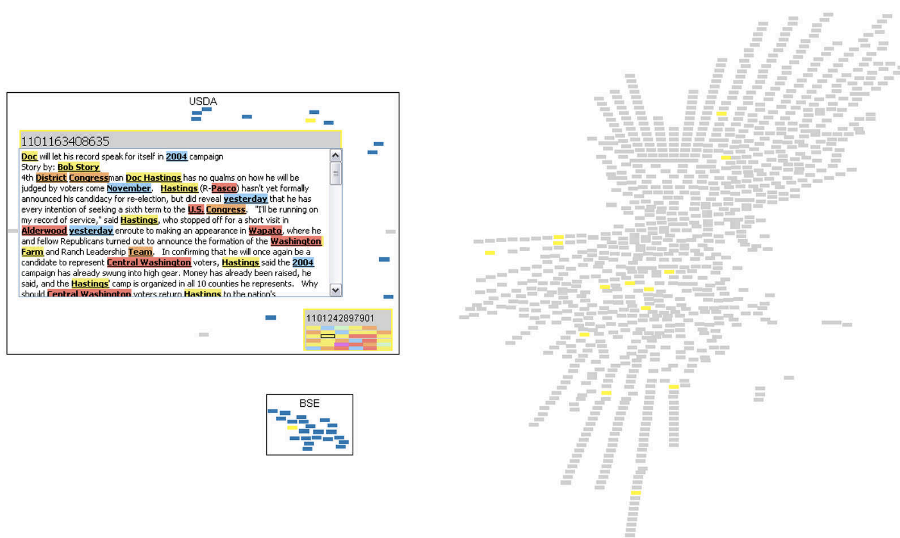


Figure 14. The investigation moves onto reading documents on Doc Hastings to see if and how he is connected to the BSE disease and the USDA.

The investigation can continue from here, looking into the BSE, USDA, and Doc Hastings and further categorizing the documents while looking for any suspicious activity. The analysis task is a long and tedious one not easily made shorter. The groups working on this challenge had several months to determine the plot. This scenario has been illustrated to demonstrate the flexibility of the features of the Semantic Zoom View.

6 Discussion

Some strengths and weakness of the design of the Semantic Zoom View have been mentioned already but in this section more will be discussed.

Spatial position and colour combine in this design to facilitate two stages of categorizing documents. Colour can be quickly applied to the view to find search results and more permanent organization can be created smoothly using groups. The use of groups almost negates the need for the filter function. Why filter documents when you can easily hide them within a group for easy access in case you need them in the future? Spatial memory of where groups are placed along with the name the analyst gives a group should both aid in quickly being able to find documents previously visited. These visual encodings that match the humans' perceptual system are a strength of the design.

Using the same colour for highlighting and selection was done to add flexibility but may be a weakness of the design. It can get in the way of quickly interacting with the view. A feature for quickly moving a single document only even if it is highlighted should be added.

One weakness of the current new design is that although the majority of features are based upon related work or the perceptual abilities of humans, the combination of features is unique and it remains to be verified that they match the tasks of the analyst effectively. Thus, experiments and interviews are needed to determine the applicability of the technique. These would preferably be with actual analysts. There are a number of questions needed to be answered. One of these concerns the initial layout of the visualization. Given no numerical data for a location for each document what is the best way to layout the documents? I have used a grid of documents placed in the center of the view so that there is space to move documents outside the grid in organizing them. Perhaps it would be more useful if the documents were spaced out so the grid filled the screen. This would mean zooming any document in place would results in less overall change to the view. Alternatively each document could start further zoomed in so that the grid filled the view with small spaces between documents. Given a small enough collection this would allow an analyst to see the names of all the documents. This could be useful but could also present an unnecessary amount of detail. This is a trade-off that is currently unanswered and the solution may only be found by allowing a potential user to try each method.

Aside from whether the operations present in the view are useful is the question of whether the ways in which they are currently performed are intuitive and easy to remember. For example, is it more intuitive to use the scroll wheel to zoom a document or to zoom a document by dragging the corner? Usability studies will need to be conducted to solve issues such as this.

6.1 Lessons Learned

I have learned a great deal from working on this project, both about some of the problems with visual encodings as well as some of the unsolved problems in the areas I researched for the implementation.

Choosing colours which are easy to spot among many other colours is not easy, especially if all the colours should be distinguishable to colour deficient people. Spatial position is a much better method of separating categories. Unfortunately the problem of how to nicely show overlapping groups of items for any number of groups and overlaps is an unsolved problem both spatially and with colours. It does not seem likely to be solved anytime soon. Most systems with colour coding, such as IN-SPIRE [12] use a specific colour to specify that an item should really be two or more colours. In other words one colour is dedicated to indicate any kind of overlap of sets. This originally did not strike me as being very effective as it does not show what the overlap is, but there does not seem to be an easy alternative.

Another thing I learned may be both a strength and weakness of my design. In investigating the document set and acting as an analyst I found I zoomed all the way into documents often. To get the real context of the document I had to read the full text and since brushing could be done directly from the full text I rarely zoomed just to the entity level. While it is good that it is so easy to zoom in and read the text it is not good that the other zoom levels did not factor into my scenario. One possibility is that this simply relates to the stage of the analysis I was at. It is also true that I am not a professional analyst and may perform quite differently from them.

There is a really hard problem of how to use space effectively when trying to lay items out by date. Clearly if there is a gap between the dates in a set of items this should be represented within the view but how can you have space for many gaps of different sizes while effectively visualizing items that have similar dates?

Finally, as mentioned in Section 4.2, I discovered that sometimes when a problem is NP-hard such as the packing problem [1], adding some constraints that make sense intuitively to the application and relaxing the problem a little can help to solve it. I am referring to the constraint of maintaining a similar relative positioning of the documents within the cluster algorithm.

6.2 Future Work

The experiments and usability studies mentioned in Section 6 are an important part of the future work on this project; however, there are also a number of extensions and more advanced features that may prove useful to analysts.

As mentioned in Section 3.1.1, documents remember previous highlighting so that they may display it when the current highlighting is removed. This is done through a stack of highlight colours within the implementation of a document. Although it is useful, this feature can get confusing if the stack becomes large or the highlights within it are fairly old. This confusion is because it is not currently possible to see the stack of highlighting colours or to change their order. A function for adjusting the global stacking of colours should help this issue. Just by adjusting the colour order with this feature the analyst could pick out documents with both highlight colours by seeing which ones change colour.

Another feature that should be added to help emphasize when documents meet two search queries is the ability to place groups within groups thereby unrestricting the number of levels of the view. Although groups cannot overlap, at least an analyst can place documents that meet two queries within a subgroup of one of the groups formed by the queries. Given the object oriented nature of the implementation (in Java) adding this feature would not require much effort.

6.2.1 Visualizing Entities

Some additional visualization techniques involving entities should be developed. The current techniques mostly focus on the documents and the people and places mentioned within them are often more important to the analyst. Currently brushing and linking allows the analyst to quickly see how many documents an entity is contained in but it is not easy to compare this for multiple entities or to find entities that share many of the same documents. In contrast it is really easy to see all of the entities within one document as they are clearly displayed nested within it. Thus providing a facility to temporarily merge multiple documents together and see all the entities within the new super document may be useful. A one level treemap could be created within the super document where the size of an entity is relative to how many of the documents it is present in. A two level treemap could be created where the first level is split by entity type and the second as described above. Within this super document and within documents in general the analyst should be given the ability to layout the entities to fit their thought process and to filter out any entities they don't wish to see just as both these activities are currently possible at the document level.

6.2.2 Spatial Layout

The date layout is currently rather limited as it does not handle groups or zoomed in documents. Handling groups could be done by applying a separate date layout to those documents within a group and placing the groups off to the side. As mentioned in Section 6.1, this may not be easy. Accurately placing documents by date in a calendar format may not be possible within a small space especially if the documents have vastly different dates. Regardless, this view should be made more flexible to better integrate with the other features of the view.

The use of 2D position allows the analyst to organize the documents to further their investigation and understanding. Unfortunately aside from the date layout the rearrangement of the documents is currently only specified manually by the analyst. This lack of facilities for meaningfully automatically performing a layout of documents is due to a lack of numerical data within the documents. As mentioned in the related work section, there have been algorithms developed for placing documents on a plane according to their keywords [12]. In the future one of these techniques will be used to perform a layout of the document collection. The analyst should be able to recompute the layout after filtering out some documents as with IN-SPIRE but should also be able to keep some documents in the view that are unaffected by the layout so they may still manually organize them.

7 Conclusion

The Semantic Zoom View is a promising new information visualization design using a focus+context method for investigating a document collection. There are clearly many ways in which it can be expanded upon and this is the focus of my thesis. A large part of my thesis work will be concentrated on determining which encodings and functions are most useful to analysts in the investigation process. This will be an iterative process in which feedback from users informs changes to the design. Although there is much work ahead this paper has introduced the basic visualization design and concepts around which the view is focused. These are providing a flexible overview capability and document organization environment and then using semantic zooming to quickly get details on demand for any document.

References

[1] N. Bansal, J. R. Correa, C. Kenyon, and M. Sviridenko. Bin packing in multiple dimensions: inapproximability results and approximation schemes. *Mathematics of Operations Research,* 31(1):31-49, Feb. 2006.

[2] L. Bartram, A. Ho, J. Dill, and F. Henigman. The continuous zoom: a constrained fisheye technique for viewing and navigating large information spaces. *Symposium on User Interface Software and Technology,* pages 207-215, 1995.

[3] C. Brewer. Colorbrewer. http://colorbrewer.org, 2009 (accessed Dec 2, 2009).

[4] Calais. http://www.opencalais.com/, 2009 (accessed Oct 25, 2009).

[5] A. Cockburn, A. Karlson, and B. B. Bederson. A review of overview+detail, zooming, and focus+context interfaces. *ACM Computing Surveys*, 41(1): 2:1-2:31, 2008.

[6] R. Dougherty, and A. Wade. VisCheck. http://www.vischeck.com, 2002 (accessed Dec 2, 2009).

[7] G. Grinstein, T. O'Connell, S. Laskowski, C. Plaisant, J. Scholtz, and M. Whiting. The VAST 2006 contest: a tale of Alderwood. *In Proc. IEEE Symposium on Visual Analytics Science and Technology* (VAST), pages 215-216, 2006.

[8] X. Huang, W. Lai, A. S. M. Sajeev, and J. Gao. A new algorithm for removing node overlapping in graph visualization. *Information Sciences*, 177:2821–2844, 2007.

[9] N. Kadivar, V. Chen, D. Dunsmuir, E. Lee, C. Qian, J. Dill, C. Shaw, and R. Woodbury. Capturing and supporting the analysis process. *In Proc. IEEE Visual Analytics Science & Technology (VAST),* pages 131-138, 2009.

[10] A. K. McCallum, MALLET: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002. (accessed Oct 25, 2009).

[11] T. Munzner. Visualization, chapter 27. In Peter Shirley, Steve Marschner, authors, *Fundamentals of Computer Graphics, Third Edition*, pages 675-707. AK Peters Ltd, 2009.
[12] Pacific Northwest National Laboratory. IN-SPIRE. http://in-spire.pnl.gov/getacopy.stm, 2008. (accessed Oct 26, 2009)
[13] Pacific Northwest National Laboratory. Starlight. http://starlight.pnl.gov/, 2008. (accessed Oct 23, 2009)
[14] E. Pietriga. A toolkit for addressing HCI issues in visual language environments. *In Proc. IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'05)*, pages 145-152, Sept. 2005.
[15] T. Reinhard, S. Meier, and M. Glinz. An improved fisheye zoom algorithm for visualizing and editing hierarchical models. *Second International Workshop on Requirements Engineering Visualization (REV)*, Oct. 2007.
[16] D. Schaffer, Z. Zuo, S. Greenberg, L. Bartram, J. Dill, S. Dubs, and M. Roseman. Navigating hierarchically clustered networks through fisheye and full-zoom methods. *ACM Transactions on Computer-Human Interaction,* 3(2):162-188, 1996.
[17] J. Stasko, C. Gorg, and Z. Liu Jigsaw: supporting investigative analysis through interactive visualization. *In Proc. IEEE Visual Analytics Science & Technology (VAST),* pages 118-132, 2008.
[18] M-A. D. Storey, C. Best, and J. Michaud. SHriMP views: an interactive environment for exploring Java programs. *International Conference on Program Comprehension*, 2001.
[19] M-A. D. Storey, and H. Muller. Graph layout adjustment strategies. *Proc. Symp. on Graph Drawing*, 1027:487-499, 1996.
[20] C. Ware. *Information visualization: perception for design*, chapter 4-5. Morgan Kaufmann/Academic Press, 2nd edition, 2004.
[21] C. Weaver. Building highly-coordinated visualizations in Improvise. *In Proc. of Information Visualization*, 2004.
[22] C. Weaver, D. Fyfe, A. Robinson, D. Holdsworth, D. Peuquet, and A. MacEachren. Visual exploration and analysis of historic hotel visits. *Information Visualization (Special Issue on Visual Analytics)*, 6:89-103, 2007.

Appendix B: Evaluation Documents

Pre-Study Questionnaire

Subject #: _______

Gender: _________ Age: _____

Role at the University:

- ☐ Undergraduate student
- ☐ Graduate student
- ☐ Staff
- ☐ Faculty

What is your experience with Information Visualization software or web pages?
Check the closest to correct.

- ☐ None
- ☐ Used only a small amount
- ☐ Used many times in the past
- ☐ Regularly use now (used in the past week and plan to use again in the coming week)

If you have used Information Visualization software or web pages, what did you use them for?
Check all that apply.

- ☐ For fun
- ☐ For school
- ☐ For work

How often do you read news articles, novels, blogs, or similar materials about recent or past events and people? Do not count academic papers. Check the closest to correct.

- ☐ Almost never
- ☐ A few times a month
- ☐ A few times a week
- ☐ Daily

Reading Exercise

Person   Organization   Place   Date   Keyword   Money   Account #

Subject #: ________

Read the questions below and then find the answer in the paragraphs below them. Answers are a few words or less. The entities in each paragraph have been highlighted.

1. How much money did the Somali pirates demand?

British intelligence sources report that a cargo ship, MV Tanya, has been captured by pirates off the coast of Somalia. The MV Tanya is owned by Waterways AG, based in the Netherlands, sails under a Panamanian flag, and is managed by the Southern Shipping Company based in the Ukraine.

British intelligence sources report that the group of approximately 50 Somali pirates who have captured the MV Tanya off the Somalian coast call themselves the Waterways Protection Regional Guard. It is reported that they have demanded a ransom of $35M US dollars for the release of the ship.

The boat is still under power toward an unknown destination.

---

2. Approximately how many TOS-1 rocket launchers were on board the MV Tanya?

---

3. How did the captain of the ship supposedly die?

The US Fifth Fleet reports that the Newport News based destroyer USS Leonard is present and in visual range of the MV Tanya, which has now anchored off the Somalia cost near Hobyo. The Belize-flagged ship was headed to the Kenyan port of Mombasa when it was seized last Thursday with a crew of 21. Nairobi said the shipment was part of a contract with Ukraine to update some of its military hardware.

According to the Ukrainian defense ministry, the MV Tanya is carrying over 30 Soviet-type TOS-1 multiple rocket launcher vehicles as well as other military supplies. It is not known if the pirates had advanced information of the cargo.

The captain of the MV Tanya is reported dead. The ship's first mate stated that the captain was a victim of an unfortunate accident, when he fell from one deck to another onto a metal walkway.

Example Questions

The documents have been clustered based on the people, places, organizations and keywords they have in common. Each cluster is labeled with the entities that are most popular in it. Before each question some of the documents may be highlighted (in red) based on a search done on the document's full text. All answers require only a few words.

1. All documents containing "pirates" have been highlighted. How much money did the Somali pirates demand?

2a. All documents containing "Medellín" have been highlighted. A phone call took place to Medellín, Colombia from another city in South America involving setting up a contact for an arms deal. Find the name of this city.

2b. Later someone from this city made a phone call to setup a money transfer. How much money (US$)?

Task Questions

Subject #: ________

The documents have been clustered based on the people, places, organizations and keywords they have in common. Each cluster is labelled with the entities that are most popular in it and these labels are needed for some questions. Before each question some of the documents may be highlighted (in red) based on a search done on the document's full text. If the question asks for someones name, only the first name is needed. All answers require only a few words.

1. All documents containing "Minsky" have been highlighted. What year did the Rome customs police first start surveillance on Leonid Minsky?

____

2. All documents containing "Kasem" have been highlighted. Muhammad Kasem is the leader of an organization in Gaza with initials MFJ. What does MFJ stand for?

____

3. All documents containing "jt" are highlighted. These are all message board posts. What is the earliest date of any of these posts?

____

4a. All documents containing "Raleigh" are highlighted. Mr. Raleigh was arrested for trying to sell firearms to an arms dealer. Find the name of this arms dealer.

4b. The arms dealer was mentioned in one phone call. Who received this phone call?

____

5. How many documents mention the entity MV Tanya?

____

6a. All documents containing “Nicolai” are highlighted. Nicolai Kuryakin is a key player in the network of arms dealers. He attended two events related to weaponry, one in June and one in September. Find the name of his companion at these events.

6b. Who did the companion meet with on June 26th?

7a. All documents containing “Lashkar-e-Jhangvi” are highlighted. Find Maulana Haq Bukhari, who is a suspected leader of this terrorist group.

7b. There is a bank account suspected of being owned by him. What are the first 3 letters of the account?

8a. An Ilyushin cargo plane was seized in Bangkok. Find the name of the person who owns the company that operates this plane.

8b. This person made a phone call on June 24th. What airport did the person call from?

9a. All documents containing “Bosnia” are highlighted. Find the phone number in Antalya, Turkey that called Bosnia to buy “farming equipment”.

9b. On December 16th, 2008 this same phone number phoned someone in Istanbul. What is the name of the person they phoned?

10a. All documents containing “Nairobi” are highlighted. Find the name of the arms dealer who died on May 1st in the Nairobi hospital.

10b. Guns were once seized in a raid on his home. How many guns?

Post-Study Questionnaire

This questionnaire gives you an opportunity to tell us your reactions to using the information visualization tool. Your responses will help us understand what aspects of this version of the tool were useful to you in performing the task and those that need improvement. Think about your experience using the tool while you answer the questions below. Please read each statement and indicate how strongly you agree or disagree with the statement by **circling a number on the scale**. Feel free to write comments to elaborate on your answers. Thank you!

Subject #: ________

1. I was able to answer the questions accurately using this system.

strongly disagree ←――――――→ strongly agree

1 2 3 4 5 6 7

Comments:

2. Answering the questions using this system was a slow process.

strongly disagree ←――――――→ strongly agree

1 2 3 4 5 6 7

Comments:

3. I found it difficult to brush across an entity in a document and see the other documents containing the same entity.

strongly disagree ←――――――→ strongly agree

1 2 3 4 5 6 7

Comments:

4. Brushing across the entities of a document was useful for answering the questions.

strongly disagree ←→ strongly agree

1 2 3 4 5 6 7

Comments:

5. I found it easy to view the text of documents.

strongly disagree ←→ strongly agree

1 2 3 4 5 6 7

Comments:

6. I found it difficult to see all the entities within a document.

strongly disagree ←→ strongly agree

1 2 3 4 5 6 7

Comments:

7. I found it difficult to close the documents.

strongly disagree ←→ strongly agree

1 2 3 4 5 6 7

Comments:

8. I found it easy to open multiple documents at once.

strongly disagree ←→ strongly agree

1 2 3 4 5 6 7

Comments:

9. When two related documents were found, I found it difficult to examine their contents simultaneously.

strongly disagree ⟷ strongly agree

1 2 3 4 5 6 7

Comments:

10. Overall, I am satisfied with the system's ease of use.

strongly disagree ⟷ strongly agree

1 2 3 4 5 6 7

Comments:

11. Please add any additional comments you would like to make about the Information Visualization tool.

Appendix C: Qualitative Notes

Notes were taken on difficulties participants had using either interface, based both on interaction with the interface and verbal comments they made. Participants were not required to think aloud, but some participants did so anyway. In addition, notes were taken on watching screen capture videos recorded per task per participant. These notes reflect many of the reasons for each participants' resulting accuracy and task completion times. Finally, much valuable feedback was gained from the written comments in the post-study questionnaire. This section is organized according to technique aspect.

Overview issues: For 2 of the task questions, participants had to use the cluster labels and layout to determine which documents to investigate and for the other 8, they used the highlighted search results. In both conditions, multiple participants clicked on the cluster labels in an attempt to highlight all the documents containing those entities. This does not work but may be a useful feature to add. For question 5, the whole task was determining the number of documents that contained one of the entities that was in a cluster label. As mentioned in the Section 4.4.1, 2 participants in the zoom condition did not use brushing to verify which documents contained the entity. Thus, the cluster label was perceived more strongly than its actual meaning, since participants thought the entity was guaranteed to be in and only in every document in the cluster. If the interface allowed clicking the cluster labels to brush the entities in the label, users would understand the distinction.

Selection issues: In the zoom condition, multiple participants forgot that it is not necessary to click a document or even have it selected in order to zoom into it (using the scroll wheel). Thus, these participants would first select it by clicking or drawing a rectangle around it. This is understandable because most interfaces require selection before interacting with an item. Counter examples of scroll wheel use not requiring prior selection include Google Maps (2011) and most web browsers. Allowing zooming a document without prior selection does not add complexity complications to the interface and does provide faster drill down document access. Prior selection would require the participant to use at least one selection colour, taking away from its use for some other highlighting in the view. The disadvantage is that this is counter to convention for many users.

In most applications, clicking an icon once selects it and double clicking it opens the associated file. Despite this common convention and training, some popup condition participants single clicked a document glyph when they meant to double click it. Unfortunately, this resulted in all other highlighted documents being deselected, thus slowing down the solving of the task. In addition, some participants double clicked a document glyph with shift to try to open all highlighted documents, but instead they all closed, since the one they had clicked was already open. They figured out quickly that they could open them all by double clicking again, but this was frustrating to them. Although it is a convention to first select items before interacting with them, it seems that in this specific application it caused problems not found in the zoom condition, which cleanly separated the selection and zoom interaction mechanisms.

Participants had minor difficulty quickly getting the mouse over one of the small document glyphs. When fully zoomed out, the glyphs are too small targets. There is a trade off in that larger glyphs mean less of them can be simultaneously displayed. With larger screens and multiple screens, this will be less of a problem. One participant commented that putting the mouse near a document could act the same way as placing it on the document. A future study would have to be done to determine the ideal glyph size and if allowing just putting the mouse near glyphs is beneficial.

When trying to open multiple documents, several participants in each condition attempted to do so with the mouse over whitespace. Since the shift key specifies opening multiple documents and these documents are already highlighted, they assumed the mouse location did not matter. When nothing happened, participants quickly figured out that they needed the mouse over one of the selected documents. This restriction exists because in the full SZV a participant can use multiple selection colours, and so the mouse is necessary to specify which set of documents to zoom. For zooming a single document, it is necessary to specify which document using the mouse.

Semantic Zoom issues: When zooming multiple documents, they expand outward from their centroid. Thus, the mouse may quickly end up over whitespace; however, as long as the participant kept scrolling without a break of a few seconds, they still zoomed the documents. This was specifically described in training; however, a few participants still chased one of the documents with the mouse. With more experience with the technique, participants would understand that this is unnecessary.

Several participants began the first task by scrolling in the wrong direction when trying to zoom in. This is most likely because they were used to the map zooming metaphor in which scrolling up moves you closer to the world. As explained in Section 3.1.6, the opposite direction is used since you are instead pulling one or more documents towards you. Participants quickly realized they needed to scroll down to zoom in. One suggestion for making the metaphor stronger was to use 3D and make the document look as though it is popping out. A future study could investigate the use of 3D techniques similar to Document Lens (Robertson et al. 1993), but it may be better to keep the visualization simple and uncluttered.

Some participants had feedback on the semantic zoom levels. One participant commented that certain zoom levels were not useful. Some participants commented that they would like to jump instantly to different zoom levels because it would be faster than scrolling. These intermediate stages are present for multiple reasons explained throughout this thesis. If a document opened to all of its contents instantly then the layout would change instantly making it hard to track individual glyphs. The alternative is for a document to zoom smoothly, but rather than have animation, SZV puts this zoom in the hands of the analyst. The scroll wheel allows fine control of the zoom speed to efficiently reach any detail level. In the study, participants sometimes needed to look for a specific person among multiple documents. Some of them determined that they could zoom into the documents until all were at zoom level 3 (Section 3.1.3) with entity rectangles but no text for the entities. At this point, they stopped and noticed that only two of the documents had green entity rectangles (people). They then just zoomed into each of the two documents. Thus, a zoom level that does not seem useful to some users can be useful given the right task.

SZV rarely has overlapping documents, so unlike the popup interface document glyphs are not covered up; however, there is less freedom for those users who wish to see more text at once. The documents have a maximum size they can be zoomed to after which the analyst must use a scrollbar. This maximum stops participants from zooming documents in too large accidentally when zooming multiple documents. This restriction could be removed for experienced users. It is worth remembering though that context can serve little purpose when there is not enough room to display it.

Popup issues: Some participants had a strategy of resizing many popups to see more text, which resulted in much of the main view being covered. This occasionally led to key document glyphs being hidden (in task 7, described in Section 4.4.1). Most participants who had this problem realized they needed to move the popups around to find highlighted document glyphs underneath them. In the zoom condition, this was simply not a problem since focus documents were shown within the context instead of covering it.

Some participants had problems with resizing popups since the border area that needed to be dragged for a resize was smaller than it should have been. Participants missed this area with the mouse despite the icon for the cursor changing. One zoom condition participant wanted to resize documents by dragging an edge or corner. This is a potential extension to SZV. In the full SZV, participants can resize groups in this manner. With a single document, once it displays the full text, the scroll wheel causes the text area to expand, which is no longer a zoom (ie. the text font does not get bigger). Thus, a study should be done to determine if performance could be improved by changing the interaction mechanism for this resizing. If this interaction were changed then the scroll wheel could match convention and be used to scroll the full text.

Entity Display issues: Several zoom condition participants commented that since the entity rectangles cut the labels for the entities to 5 or 6 characters, it was difficult to scan through the entities. This could be addressed by making the standard entity rectangle longer or by replacing it with a tag cloud formed of the entity values; however, this would take up more space. Thus, less documents could be in focus and there would be less room for the context. It is interesting that no one in the popup condition made this comment about the entity labels. This is most likely because the full text was available to them within the popup. Thus, they ended up scanning the text to see the full entity names instead of using the entity grid.

Brushing issues: For 7 of the 10 task questions, brushing an entity was necessary to solve the problem. Several popup condition participants attempted to double click entities to perform brushing. This acted as two single clicks that performed the brushing but then hid it again. Participants figured this out and learned to use only a single click. This problem did not occur in the zoom condition since double click was not used for anything.

Colour issues: Adjustments made to the interfaces after the study's pilot participants may have caused a few problems for participants when performing brushing. A pure red colour was used for selection in order to make the highlighted document glyphs stand out. Unfortunately, this resulted in the zoomed documents and popup frames standing out even more as well. The large area of red colour in these focus documents draws the eye away from the rest of the view. Sometimes participants performed brushing but then read the open documents they had already read instead of noticing highlighted document glyphs. It was even harder for participants to spot the highlighted document glyphs located close to the popups or zoomed in documents since they stuck out less than those farther away. Such a bright colour should not be used for the whole background of a zoomed in document. Instead, the border could be all that is highlighted to reduce the amount of colour. Also, on the large monitor used for the study the pure white background of the view may have made it harder to spot the highlighted documents. An off-white colour may have made the highlighted documents stand out better.

Some more comments from participants matched functionality that is present in the full SZV, but was removed from the interface for the purposes of the study. This included brushing multiple entities, searching, and changing the ordering of entities.

Some other features requested by participants are available or under construction within the larger CZSaw application (Kadivar et al. 2009). Multiple participants mentioned the desire to undo actions within the view. They wish to undo the de-selection of entities or undo the closing of popups. CZSaw focuses on capturing the analysis process in a script that a user can rerun to a different point in order to return to an earlier state of analysis. Work is still being done to record all interactions with SZV and the other data views as statements in CZSaw's script. Multiple participants commented that they would like to change which entities are highlighted in the documents. If they could have added entities for items such as weapons or numbers then they could have solved some of the task problems easier. This entity refinement is possible in CZSaw where new entities can be extracted on the fly. This editing of the data is important for analysts working with messy data; however, this is a feature that was unrelated to what was being tested by this study.

In summary, the qualitative findings found some large differences in use of the two interfaces that included strengths and weaknesses of both approaches. Feedback on both interfaces is useful for improving SZV in the many ways discussed throughout this section. In general, the document popups provided greater flexibility to participants since they could move them around and resize them, yet this flexibility hindered the use of context. It resulted in more of the overview being covered. Participants in the zoom condition did not have this freedom, but at the same time they did not run into the difficulties that it caused. Instead participants in the zoom condition were faced with a more novel interface and had to learn how to semantically zoom documents. They performed quite well under these circumstances and provided much feedback for improvement of the tool.

REFERENCE LIST

Afram, A.J., Briedis, J., Fujiwara, D., Jacob, R.J.K., Cao, C.G.L., and Kahle, D. (2007) Evaluation of Semantic Fisheye Zooming to Provide Focus+Context. *Proceedings of the Human Factors and Ergonomics Society 51st Annual Meeting* 459-463.

Ahn, J. and Brusilovsky, P. (2009) Adaptive Visualization of Search Results: Bringing User Models to Visual Analytics. *Information Visualization* **8**(3), 167-179.

Arias-Hernandez, R., Kaastra, L., Green, T.M., and Fisher, B. (2011) Pair Analytics: Capturing Reasoning Processes in Collaborative Visual Analytics. *Proceedings of the Hawaii International Conference on System Sciences.*

Bartram, L., Ho, A., Dill, J., and Henigman, F. (1995) The Continuous Zoom: A Constrained Fisheye Technique for Viewing and Navigating Large Information Spaces. *Symposium on User Interface Software and Technology*, 207-215.

Baudisch, P., Good, N., Bellotti, V., and Schraedley, P. (2002) Keeping Things in Context: A Comparative Evaluation of Focus Plus Context Screens, Overviews, and Zooming. *CHI Conference on Human Factors in Computing Systems.*

Bederson, B.B. (2009) The Promise of Zoomable User Interfaces. *CHI Conference on Human Factors in Computing Systems.*

Bederson, B.B., Clamage, A., Czerwinski, M.P., and Robertson, G.G. (2004a) DateLens: A Fisheye Calendar Interface for PDAs. *ACM Transactions on Computer-Human Interaction,* **11**(1), 90–119.

Bederson, B.B., Grosjean, J., and Meyers, J. (2004b) Toolkit Design for Interactive Structured Graphics. *IEEE Transactions on Software Engineering,* **30**(8), 535-546.

Bederson, B.B. and Hollan, J.D. (1994) Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics. *Proceedings of the ACM Symposium on User Interface Software and Technology,* 17-26.

Bederson, B.B., Meyer, J., and Good, L. (2000) Jazz: An Extensible Zoomable User Interface Graphics Toolkit in Java. *Proceedings of the ACM Symposium on User Interface Software and Technology,* 171-180.

Chen, V., Dunsmuir, D., Alimadadi, S., Lee, E., Guenther, J., Dill, J., Qian, C., Shaw, C.D., Stone, M., and Woodbury, R. (2010) Model based Interactive Analysis of Interwoven, Imprecise Narratives. *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology,* 275-276.

Cleveland, W.S. and McGill, R. (1984) Graphical Perception: Theory, Experimentation and the Application to the Development of Graphical Models. *Journal of the American Statistical Association*, **79**(387), 531-554.

Cockburn, A., Karlson, A., and Bederson, B.B. (2008) A Review of Overview+Detail, Zooming, and Focus+Context Interfaces. *ACM Computing Surveys*, **41**(1), 1-31.

Collaud, G., Dill, J., Jones, C.V., and Tan, P. (1996) The Continuously Zoomed Web - A Graphical Navigation Aid for WWW. *IEEE Visualization Late Breaking Hot Topics Papers,* 1-3.

Dill, John. (2011) Personal Communication.

Dill, J., Bartram, L., Ho, A., and Henigman, F. (1994) A Continuously Variable Zoom for Navigating Large Hierarchical Networks, *Proceedings of the IEEE Conference on Systems, Man and Cybernetics*, 386-390.

Dunsmuir, D. (2009) Selective Semantic Zoom of a Document Collection. *Report for the University of British Columbia course CPSC 533: Information Visualization, Fall 2009.* Accessed on February 23, 2011. http://www.cs.ubc.ca/~tmm/courses/533/projects.html.

Dunsmuir, D., Baraghoush, M.Z., Chen, V., Joorabchi, M.E., Joorabchi, M.E., Alimadadi, S., Lee, E., Dill, J., Qian, C., Shaw, C.D., and Woodbury, R. (2010) CZSaw, IMAS & Tableau: Collaboration among Teams, *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology,* 267-268.

Findlater, L., Jansen, A., Shinohara, K., Dixon, M., Kamb, P., Rakita, J., and Wobbrock, J. O. (2010). Enhanced Area Cursors: Reducing Fine Pointing Demands for People with Motor Impairments. *Proceedings of the ACM Symposium on User Interface Software and Technology*, 153–162.

Fisher, B.D., Agelidis, M., Dill, J., Tan, P., Collaud, G., and Jones, C. (1997) CZWeb: Fish-eye Views for Visualizing the World Wide Web. *Proceedings of the International Conference on Human-Computer Interaction,* 719-722.

Fruchterman, T.M.J. and Reingold, E.M. (1991) Graph Drawing by Force-directed Placement. *Software: Practice and Experience,* **21**(11), 1129-1164.

Furnas, G.W. (2006) A Fisheye Follow-up: Further Reflection on Focus + Context. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* 999-1008.

Furnas, G.W. (1986) Generalized Fisheye Views. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* 16-23.

Furnas, G.W. and Rauch, S.J. (1998) Considerations for Information Environments and the NaviQue Workspace. *Proceedings of the ACM Conference on Digital Libraries,* 79-88.

Google Maps. (2011) Accessed on February 25, 2011. http://maps.google.com/.

Grinstein, G., Plaisant, C., Scholtz, J., and Whiting, M. (2010) Text Records – Investigations into Arms Dealing. *Visual Analytics Benchmark Repository: VAST Challenge 2010.* Accessed on November 24, 2010. http://hcil.cs.umd.edu/localphp/hcil/vast/archive/task.php?ts_id=148.

Heer, J., Mackinlay, J. D., Stolte, C., and Agrawala, M. (2008) Graphical Histories for Visualization: Supporting Analysis, Communication, and Evaluation. *IEEE Transactions on Visualization and Computer Graphics,* **14**(4), 1189-1196.

Hetzler, E. and Turner, A. (2004) Analysis Experience using Information Visualization. *IEEE Computer Graphics and Applications,* **24**(5), 22-26.

Hornbæk, K., Bederson, B.B., and Plaisant, C. (2002) Navigation Patterns and Usability of Zoomable User Interfaces with and without an Overview. *ACM Transactions on Computer-Human Interaction,* **9**(4), 362-389.

Kadivar, N., Chen, V., Dunsmuir, D., Lee, E., Qian, C., Dill, J., Shaw, C., and Woodbury, R. (2009) Capturing and Supporting the Analysis Process. *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology*, 131-138.

Lam. H., Fisher, B., and Dill, J. (2005) A Pilot Study of CZTalk: A Graphical Tool for Collaborative Knowledge Work. *Proceedings of the Hawaii International Conference on System Sciences.*

Lamping, J., Rao, R., and Pirolli, P. (1995) A Focus+Context Technique based on Hyperbolic Geometry for Visualizing Large Hierarchies. *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 401–408.

Leuski, A. (2001) Evaluating Document Clustering for Interactive Information Retrieval. *Conference on Information and Knowledge Management,* 33-40.

Leuski, A. and Allan, J. (2004) Interactive Information Retrieval Using Clustering and Spatial Proximity. *User Modeling and User-Adapted Interaction,* **14***,* 259-288.

Leuski, A. and Croft, W.B. (1996) An Evaluation of Techniques for Clustering Search Results. *Technical Report IR-76*, Department of Computer Science, University of Massachusetts, Amherst.

LingPipe. (2010) Alias'i. Accessed on November 24, 2010. http://alias-i.com/lingpipe/.

Mackinlay, J., Roberston, G., and Card, S. (1991) The Perspective Wall: Detail and Context Smoothly Integrated. *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 173–179.

Munzner, T., Guimbretiere, F., Tasiran, S., Zhang, L. and Zhou., Y. (2003) TreeJuxtaposer: Scalable Tree Comparison using Focus+Context with Guaranteed Visibility. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* 453-462.

Nelson, T., White, S., Boey, Y-B., and Fisher, D. (2009) Java Universal Network/Graph Framework. Accessed on November 24, 2010. http://jung.sourceforge.net/.

Olsen, K.A., Korfhage, R.R., Sochats, K.M., Spring, M.B., and Williams, J.G. (1993) Visualization of a Document Collection: The VIBE System. *Information Processing & Management,* **29**(1)*,* 69-81.

Perlin, K. and Fox, D. (1993) Pad: an Alternative Approach to the Computer Interface. *International Conference on Computer Graphics and Interactive Techniques,* 57-64.

Pietriga, E. (2005) A Toolkit for Addressing HCI Issues in Visual Language Environments, *IEEE Symposium on Visual Languages and Human-Centric Computing*, 145-152.

Plaisant, C. (2004) The Challenge of Information Visualization Evaluation. *Proceedings of the Working Conference on Advanced Visual Interfaces*, 109-116.

Plaisant, C., Grosjean, J., and Bederson, B. (2002) SpaceTree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation. *Proceedings of the IEEE Conference on Information Visualization.*

The Radicati Group, Inc. (2010) Accessed on December 9, 2010. http://www.radicati.com/.

Rao, R. and Card, S.K. (1994) The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus+Context Visualization for Tabular Information. *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 318–322.

Reinhard, T., Meier, S, and Glinz, M. (2007) An Improved Fisheye Zoom Algorithm for Visualizing and Editing Hierarchical Models. *Second International Workshop on Requirements Engineering Visualization.*

Rijsbergen, C.J. (1979) *Information Retrieval,* Second Edition. London, Butterworths.

Rimey, R.D. and Bolme, D.S. (2006) Re-Using Millions of Visualizations. *Intelligence Tools Workshop,* 19-27.

Risch, J.S., Rex, D.B., Dowson, S.T., Walters, T.B., May, R.A., and Moon, B.D. (1997) The Starlight Information Visualization System. *Proceedings of the IEEE Conference on Information Visualization,* 42-49.

Robertson, G.G. and Mackinlay, J.D. (1993) The Document Lens. *Proceedings of the ACM Symposium on User Interface Software and Technology,* 101-108.

Sarkar, M. and Brown, M. (1992) Graphical Fisheye Views of Graphs*. Proceedings of the ACM Conference on Human Factors in Computing Systems*, 83–91.

Schaffer, D., Zuo, Z., Bartram, L., Dill, J., Dubs, S., Greenberg, S., and Roseman, M. (1993) Comparing Fisheye and Full-Zoom Techniques for Navigation of Hierarchically Clustered Networks. *In Proceedings of Graphics Interface,* 87-96.

Schaffer, D., Zuo, Z., Greenberg, S., Bartram, L., Dill, J., Dubs, S., and Roseman, M. (1996) Navigating Hierarchically Clustered Networks through Fisheye and Full-Zoom Methods. *ACM Transactions on Computer-Human Interaction,* **3**(2), 162-188.

Shneiderman, B. (1996) The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. *IEEE Proceedings of the Symposium on Visual Languages,* 336-343.

Singhal, A. (2001) Modern Information Retrieval: A Brief Overview. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering,* **24**(4), 35-43.

Stasko, J., Görg, C., and Liu, Z. (2008) Jigsaw: Supporting Investigative Analysis through Interactive Visualization. *Information Visualization,* **7**(2), 118-132.

Storey, M-A.D. and Müller, H. (1996) Graph Layout Adjustment Strategies. *Proceedings of the Symposium on Graph Drawing,* **1027**, 487-499.

Text Retrieval Conference (TREC) Home Page. (2010) Accessed on February 24, 2011. http://trec.nist.gov/.

Thomas, J.J. and Cook, K.A. (2005) *Illuminating the Path. The Research and Development Agenda for Visual Analytics.* IEEE.

Treisman, A. (1985) Preattentive Processing in Vision. *Computer Vision, Graphics, and Image Processing,* **31**(2), 156-177.

Ward Jr., J.H. (1963) Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association,* **58**(301), 236-244.

Ware, C. (2004) *Information Visualization: Perception for Design,* Second Edition. San Francisco, CA, Morgan Kaufman.

Willett, P. (1988) Recent Trends in Hierarchic Document Clustering: A Critical Review. *Information Processing and Management*, **24** (5), 577-597.

Wise, J.A., Thomas, J.J., Pennock, K., Lantrip, D., Pottier, M., Schur, A., and Crow, V. (1995) Visualizing the Non-Visual: Spatial Analysis and Interaction with Information from Text Documents. *Proceedings of the IEEE Conference on Information Visualization*, 51-58.

Woodruff, A., Landay, J., and Stonebraker, M. (1998) Constant Information Density in Zoomable Interfaces. *Proceedings of the Conference on Advanced Visual Interfaces*, 57-65.

Woodruff, A., Olston, C., Aiken, A., Chu, M., Ercegovac, V., Lin, M., Spalding, M., and Stonebraker, M. (2001) DataSplash: A Direct Manipulation Environment for Programming Semantic Zoom Visualizations of Tabular Data. *Journal of Visual Languages and Computing,* **12**, 551-571.

Woods, D.D., Patterson, E.S., Roth, E.M., and Christofferson, K. (2002) Can We Ever Escape from Data Overload? *Cognition, Technology and Work,* **4**, 22-36.

Zellweger, P.T., Chang, B-W., and Mackinlay, J.D. (1998) Fluid Links for Informed and Incremental Link Transitions. *Proceedings of the Conference on Hypertext and Hypermedia,* 50-57.

Zellweger, P.T., Regli, S.H., Mackinlay, J.D., and Chang, B-W. (2000) The Impact of Fluid Documents on Reading and Browsing: An Observational Study. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* 249-256.

Zuo, Z. (1992) Variable Zoom: A new Display Technique for Large Hierarchical Networks. *Thesis (M.A.Sc. (Engineering Sc.)), Simon Fraser University.*