Workshop Proceedings

LIVVIL: Logging Interactive Visualizations & Visualizing Interaction Logs

October 23rd, 2016

IEEE VIS 2016 conference, Baltimore, MD, USA

http://livvil.github.io/workshop/

Organizers

Romain Vuillemot (LIRIS, École Centrale de Lyon, France)
Jeremy Boy (United Nations Global Pulse New York Lab, USA)
Aurélien Tabard (LIRIS, Université Lyon 1, France)
Charles Perin (Innovis, University of Calgary)
Jean-Daniel Fekete (Inria Saclay, France)

Contents

Challenges in Logging Interactive Visualizations and Visualizing Interaction Logs

Romain Vuillemot* Ècole Centrale de Lyon
Jeremy Boy† UN Global Pulse
Aurélien Tabard‡ Université Lyon 1
Charles Perin§ University of Calgary
Jean-Daniel Fekete¶ INRIA

ABSTRACT

A growing number of visualization tools are now publicly released on the Web. While this has many benefits, such as reaching more users without any installation time or procedure, it is often unclear how those tools are being used. The most common method to remotely observe usage is remote logging through a web server. Analyzing recorded logs has already been successful to improve the usability of tools, assess the performance of users and even to enrich the user interface with histories or logs visualizations. However, from our own practice of recording and analyzing logs, we have found a lack of methodology to support this process and use the results consistently. Our goal is to raise awareness of the potential of logging to improve visualization tools and their evaluation, as well as paving the way for a long term research agenda on the use of logs in Information visualization (Infovis).

1 CONTEXT AND MOTIVATION

Logging is a mechanism for automatically capturing the behavior of a program or of a user. It is usually invisible, non-obtrusive, and can be set up remotely for long periods of time [1]. Logging can be particularly useful for information visualization research, as it can serve to debug a visualization, to test its usability, or to evaluate a user's behavior while interacting with it. Although logs usually aim at capturing system events resulting form e. g., user interactions, they can also record other valuable information like a visualization's state at specific moments of a user-session—typically what data is being used, what window layout shows up on the user's UI, etc. These recordings can be set up explicitly (e. g., using a log tracker), or can be indirectly generated using web server logs or proxies [7].

We, the authors of this article, have used logging mechanisms for almost a decade now. We have mainly conducted system evaluation and user behavior analysis using logs, but we have also started to explore novel ways of visualizing logs themselves to facilitate their analysis. We have developed a variety of tools to track user-activity, which we have deployed in various online visualizations and tools, some of which have reached great masses of users (+100 000). This experience has led us to appreciate the need for developing structured ways of making sense of logs, and is what drives the questions and discussions we raise in this proposal. As so far we have failed to find proper documentation on best practices in this area in the Infovis literature, we hereby intend to encourage a community effort to share best practices, resources, and outline promising directions for future research and developments.

Logging is difficult because it provides only a partial view of users' behavior. This is the trade-off to accept in order to remotely track them in their own settings, such as computer, desktop and real collaboration environment. Logs cannot capture everything as most users are distracted by other applications, emails and social network notifications, and coffee breaks with colleagues. Recording every single event the user generates is also not reasonable as the volume of logs will be too important and the signal drowned in the noise especially as we have said since users are often multi-tasking. Finally, if logging spreads over long periods of times, Infovis software may have been upgraded during the period, the user may have worked offline and her environment has changed (new input or output device such as mouse or screen). For all those reasons, logging is a non-trivial problem but have a huge potential if done properly.

As far as we know, there hasn't been any attempt of tackling Infovis logging research and technical questions head-on. The workshop BELIV (Beyond Time And Errors: Novel Evaluation Methods For Visualization) has been running every 2 years for 10 years now, and a series of articles [1, 6, 4] investigate logging as an evaluation mechanism. Over the same period of time, the VAST challenge also released many datasets related to logs. For instance in 2011 the challenge contained firewall, IDS (Intrusion Detection System) and syslog (System) logs. Last year's workshop on Personal Infovis at IEEE Vis gathered researchers analyzing and visualizing human behavior data. Other research communities have organized workshops focused on logging user activity for specific contexts, such as such as WWW [3] for Web browsing. None of those workshops address the characteristics of Infovis interaction techniques and evaluation procedures.

2 RESEARCH AGENDA PROPOSAL

Our agenda focuses primarily on five issues associated with logging: 1) defining logging format(s); 2) reporting and analyzing logs; 3) setting up logging infrastructures; 4) reflecting on the legal issues and necessary ethical practices associated with logging; and 5) applications related to logs, such as their visual representations. In the following subsections, we briefly develop on each of these issues, and we propose a series of open questions intended as 'food for thought' for future research directions.

2.1 A Standard Logging Format

The first step when setting up a logging process is to ask what should be recorded, when it should be recorded, and how (by the web server, by the application itself, etc.). For example, even a simple and ubiquitous interaction, like a mouse dragging, requires carefully considerations as it can generate a lot of noisy events resulting in very large and thus difficult to interpret log files.

- How to record low-level interactions (mouse moves, keystroke, ...) and data-intensive interactions (dynamic queries, brushing and linking, ...) efficiently?
- How to track multiple and coordinated views? How to track the view the user currently focuses on?
- What is the scope of the context that should be recorded beyond user's interaction? Desktop UI configuration? Computer and office setup?
- How to record collaborative and multi-device activities?

*e-mail: romain.vuillemot@gmail.com
†e-mail:myjyby@gmail.com
‡e-mail:aurelien.tabard@univ-lyon1.fr
§e-mail:charles.perin@ucalgary.ca
¶e-mail:jean-daniel.fekete@inria.fr

- How to reduce the size of data intensive interactions? Should there by a low frequency / interaction sampling, filtering and/or aggregation to shrink log files? Should there be any buffering strategy?
- Is the Common Log Format (IP, User ID, Timestamp, etc.) generated by web servers, expressive enough to be the standard for Infovis? What are the related and upcoming standards (W3C, others)? Should Infovis define its own log format?

2.2 Logs Reporting and Analysis

Logs reporting in academic papers varies with high discrepancies. In PivotSlice [14] authors report *"interaction logs were recorded by the software"*. While in À Table [11], authors provide a detailed *"Participation Logs"* analysis of the 185636 interaction from 648 visitors. This raises the need to improve logs analysis reporting to allow sound conclusions, and reproducibility of the evaluation.

- What relevance have vanity metrics (# users, # visits) to assess the success of an Infovis tool/technique?
- What should be the standard procedure or the best practices in logs reporting, for applications ranging from usability testing to evaluation?
- How to improve the reproducibility of research results and interoperability between logging tools and techniques?
- What are the specifics of logs for controlled experiments versus in the wild ones?
- How do user behavior framework like the HEART framework translate into logs? (and vice versa)

2.3 Logging Infrastructure

As we have mentioned earlier, a series of tools log users by default (e.g. proxies, web servers). However, from the authors' practical experience, it is oftentimes necessary to build its own tools for the sake of control over the logging format and flexibility in types of events to tracks.

- What is a simple and affordable setup for logging in Infovis?
- How to deal with offline tracking, synchronization? How to merge collected logs with other data sources, e.g. to clean, validate or enrich them with more contextual information?
- How updating an Infovis technique impacts previously collected/legacy logs?
- How existing APIs (Google Analytics, KissMetrics) can be used to track Infovis techniques? And perform tests such as A/B testing, perform cohort analysis, and real time monitoring?
- Beyond remote servers: what logging device or tracker can be used for logging? Can logging be manual and self-reported by users, instead of automated?

2.4 Legal and Ethical questions

As log collection and analysis is related to behavioral research involving humans, it requires approval from researchers' employer.

- How to make logging comply with IRB applications? How those applications shape the logging collection and evaluation procedure?
- What are the disclaimers best practices to notify users of logging activity?
- What would be the design of logging respecting privacy (e.g. logging that doesn't enable to reveal people's identity)?

2.5 Application Related to Logging

Finally, we think that agreeing upon a logging format and infrastructure, would have spillovers such as data interoperability and allow more applications building upon logs. Letting users visualize logs, whether it is their own or others, is a rich and promising area to identify patterns [10, 12], insights [13] of large logs collections [9]. Logs may also enrich the user experience with enhanced history navigation [2], browsing [5] and monitoring [8]. More Infovis and Visual Analytics application already make sense of logs and further research is need to tackle challenges with the growing complexity of data types, user tasks, and the need for scalable solutions as logs volume increases exponentially.

REFERENCES

[1] Ben Shneiderman and C. Plaisant. Strategies for evaluating information visualization tools: multi-dimensional in-depth long-term case studies. In *Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization*, pages 1–7. ACM, 2006.

[2] F. Chevalier, P. Dragicevic, A. Bezerianos, and J.-D. Fekete. Using Text Animated Transitions to Support Navigation in Document Histories. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 683–692, New York, NY, USA, 2010. ACM.

[3] A. Edmonds, K. Hawkey, M. Kellar, and D. Turnbull. Logging traces of Web activity: The mechanics of data collection. In *Workshop presented at the 15th International Conference on World Wide Web (WWW 2006)*, 2006.

[4] J. Gerken, P. Bak, C. Jetter, D. Klinkhammer, and H. Reiterer. How to use interaction logs effectively for usability evaluation. 2008.

[5] J. Heer, J. Mackinlay, C. Stolte, and M. Agrawala. Graphical Histories for Visualization: Supporting Analysis, Communication, and Evaluation. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1189–1196, Nov. 2008.

[6] N. Henry, N. Elmqvist, and J.-D. Fekete. A Methodological Note on Setting-up Logging and Replay Mechanisms in InfoVis Systems. In *BELIV'08*, Florence, Iran, Apr. 2008. ACM.

[7] J. I. Hong, J. Heer, S. Waterson, and J. A. Landay. WebQuilt: A Proxy-based Approach to Remote Web Usability Testing. *ACM Transactions on Information Systems*, 19(3):263–285, July 2001.

[8] S. Huron, R. Vuillemot, and J. D. Fekete. Visual Sedimentation. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2446–2455, Dec. 2013.

[9] K. E. Isaacs, P. T. Bremer, I. Jusufi, T. Gamblin, A. Bhatele, M. Schulz, and B. Hamann. Combing the Communication Hairball: Visualizing Parallel Execution Traces using Logical Time. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2349–2358, Dec. 2014.

[10] A. Makanju, S. Brooks, A. N. Zincir-Heywood, and E. E. Milios. LogView: Visualizing Event Log Clusters. In *Sixth Annual Conference on Privacy, Security and Trust, 2008. PST '08*, pages 99–108, Oct. 2008.

[11] C. Perin, R. Vuillemot, and J.-D. Fekete. A Table!: Improving Temporal Navigation in Soccer Ranking Tables. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, CHI '14, pages 887–896, New York, NY, USA, 2014. ACM.

[12] R. Vuillemot and C. Perin. Sports Tournament Predictions Using Direct Manipulation. *IEEE Computer Graphics and Applications*, 36(5):62–71, Sept. 2016.

[13] K. Wongsuphasawat and J. Lin. Using visualizations to monitor changes and harvest insights from a global-scale logging infrastructure at Twitter. In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 113–122, Oct. 2014.

[14] J. Zhao, C. Collins, F. Chevalier, and R. Balakrishnan. Interactive Exploration of Implicit and Explicit Relations in Faceted Datasets. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2080–2089, Dec. 2013.

Interaction Log and Provenance for Sensemaking

Phong H. Nguyen, Kai Xu, B. L. William Wong

ABSTRACT

This paper describes two visual analytic tools designed to support sensemaking through the visualisation of interaction log and analytic provenance. The first tool, SensePath, aims to reduce the time required for the transcription and coding during qualitative analysis such as thematic analysis (making sense of the experiment data). The second tool, SenseMap, is designed to help online sensemaking with everyday tasks such as buying a digital camera. User evaluation leads to early insight of how the visualisation of interaction log and analytic provenance can help these sensemaking tasks.

Keywords: analytic provenance, sensemaking, visualization, interaction logs

Index Terms: K.6.1 [Management of Computing and Information Systems]: Project and People Management—Life Cycle; K.7.m [The Computing Profession]: Miscellaneous—Ethics

1 INTRODUCTION

There are many possible applications of interaction logs, and we are particularly interested in supporting sensemaking. In this paper, we describe two visual analytics tools designed to support sensemaking with the help of interaction log, which we call *analytic provenance*.

Sensemaking is the process of comprehension, finding meaning and gaining insight from information, producing new knowledge and informing further action [11]. It is the construction, elaboration and reconciliation of representations that explain the information we receive about the world [6]. The outcome of the sensemaking process is important, but the process itself also contains valuable information [10]. Analytic provenance captures both low-level user interaction with visual exploration systems and high-level user reasoning process. It supports reproducibility, accountability, training, collaboration and can help us understand what we can trust from possibly uncertain data [12].

Given the rapid increase in data volume and complexity, more tools are needed to support sensemaking, which in many cases remains a slow and laborious process performed by human analysts. The design of such tools requires a deep understanding of the sensemaking process, which is a reoccurring goal of qualitative research conducted by many HCI researchers. Common methods for such qualitative analyses are grounded theory [3] and thematic analysis [5]. Typically, researchers need to design a study, collect observation data, transcribe the screen capture videos and think-aloud recordings, identify interesting patterns, group them into categories, and build a model or theory to explain those findings. Unfortunately, this process largely remains manual and thus very time consuming. Thus, the first tool we development, SensePath [9], is designed to help this process, supporting the transcription and coding of the observation data of online sensemaking.

Another common issue in sensemaking is that people often get lost when solving complicated tasks using big datasets over long periods of exploration and analysis. They may forget what they have done, fail to find the information they have discovered before, and do not know where to continue. In the World Wide Web context, this is known as the *disorientation* problem [2]. One approach to address this problem is through a graphical browser history [7]. It visualizes visited web pages and the linking relationships between them to help users to quickly see where they are in the network and to navigate to the page they want. However, when solving a sensemaking task online, which requires gathering, restructuring and reorganizing lots of information to gain insight, the disorientation problem becomes more severe and difficult to address. They do not just get lost in the hypertext space but also get lost in the task space. Our second tool SenseMap [8] captures the sensemaking process (through interaction logging) and provides an overview to support information collection and curation.

2 SENSEPATH

First, we conducted two sets of observations to understand the characteristics of qualitative analysis of sensemaking activities. The insight from the observations led to the design user requirements for our tool. We decided to support qualitative researchers in using thematic analysis, specifically to improve the efficiency of its transcription and coding stages. Other requirements can be found in the original paper [9].

Our tool – SensePath – is implemented as a Chrome extension consisting of two components. The first one is a background process running in the participant's browser to automatically capture all the required analytic provenance during the observation stage of the qualitative study. The second component includes a set of four linked visualizations of the captured provenance data (Fig. 1), designed to facilitate transcription and coding.

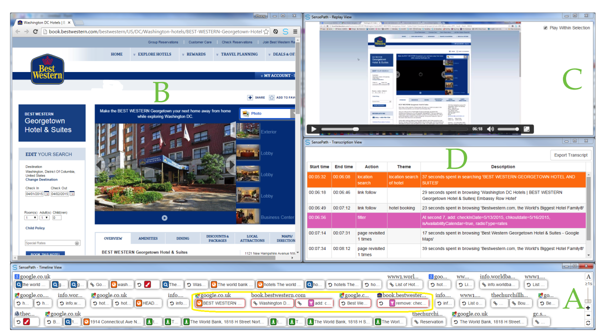

Figure 1: Four linked visualizations of SensePath. **A:** The timeline view shows all captured sensemaking actions in temporal order. **B:** The browser view displays the web page where an action was performed. **C:** The replay view shows the screen capture video to provide additional context. **D:** The transcription view details selected actions (highlighted in the timeline) and generates their transcript.

2.1 Interaction/Provenance Capture

We capture the analytic provenance corresponding to the *action* level in the Gotz and Zhou's model [4]. This capture can be done automatically yet still provides reasonable amount of semantics to the researchers. The following four aspect of actions are captured.

- **Type**: The type of action such as *search* and *filter*.
- **Timing**: The start and end time of an action.
- **Context**: Page title, URL, screenshot and contextual information such as "keyword" for search and "selected text" for highlight.

- **Relationship**: Providing how a web page was activated including *revisit* an already opened page, directly *link* from an existing page, manually *type* a new address, and open from a *bookmark*.

2.2 Timeline View

This view provides an overview of the entire sensemaking process, showing all the captured actions in their temporal order (Fig. 1A).

An action is represented as a bar, presenting all four aspects of provenance information discussed earlier (Fig. 2). The page URL (context) is displayed atop the bar. In the bar, the first icon shows that this action revisited a previously opened page (relationship). Next is the page title (context); only part of which is shown because of the limited space. This is followed by an icon indicating the type of that action such as a "filter". The last part is the specialized context for each action type, which is filtering parameters in this figure. The width of the action bar corresponds to the length of time spent in browsing the web page, and the relative position of the action type icon marks when the action happened.

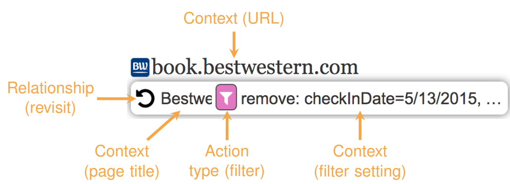

Figure 2: An action bar showing all four aspects of provenance information.

Zooming Action bars can reduce their widths through zooming to accommodate more actions. At the smallest level, only the action type is visible, and more details will become available when zooming in. Fig. 3 shows three zoom levels of action bars with the details increasing from top to bottom.

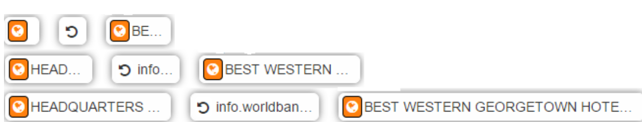

Figure 3: Three zoom levels of action bars with the details increasing from top to bottom.

Aggregate Action Instead of showing individual actions, adjacent ones happened on the same web page are merged to save space. It may also help researchers quickly understand the participant's process. Fig. 4 shows an aggregated action with eight highlights, which were made on the same Google Plus page.

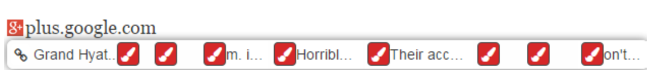

Figure 4: An aggregate action bar. It combines eight adjacent highlights made on the same Google Plus page.

Because the action bar is short, a timeline can show multiple rows. This, in combination with aggregation and interaction (described next), allows SensePath to display a reasonably large sensemaking session within a limited space. Fig. 1A shows about 50 actions out of a total of 70 actions from a 30-minute long session.

Selective Zooming SensePath implements focus+context technique [1] through *selective zooming*: when a zoom is executed, only a selected set of actions affects. This enables researchers to concentrate on certain actions without losing their context. However, they may forget the difference in zoom levels of actions, thus misunderstand the action lengths indicated by the bar widths. SensePath provides a reset button to change the zoom levels of all actions to the default value. Fig. 5 illustrates this technique.

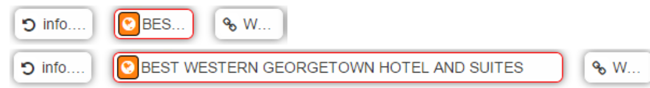

Figure 5: Selective zooming. Selected action bars are with red borders. Top row: before zooming. Bottom row: after zooming – only the selected action has its zoom level changed.

Filtering Researchers can filter actions based on duration, enabling them to focus on the range of actions they want. For example, if researchers think actions that last only a few seconds are trivial, they can be filtered out using a slider (Fig. 6), which sets a minimal length for visible actions. When the slider moves, actions that will be removed fade out, before disappearing when the slider stops. This enables researchers to preview the effect of filtering.

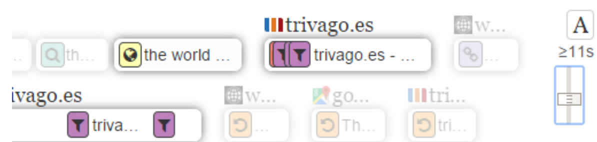

Figure 6: Actions filtering. The slider (on the right side) controls the minimal length visible actions. Actions fall below the threshold fade out first before completely disappearing.

Coding In traditional qualitative analysis, researchers analyze transcripts to identify common themes and assign suitable names or codes to them. In SensePath, the timeline view provides a succinct summary of the sensemaking process and allows researchers to drill down to explore more specific actions. Representing action types with icons and visualizing a sequence of actions next together may also help researchers to quickly identify patterns of the data, compared to watching videos or reading transcripts. Coding feature is available through a menu button when hovering an action bar.

2.3 Browser View

When an action is selected in the timeline, its associated web page is showed in the browser view (Fig. 1B). This enables researchers to examine the web page that the participant was looking at when performing a sensemaking action. If the action is an annotation or highlight, the browser view will automatically navigate to the location of the web page where the annotation or highlight was made, informing researchers which part of the page the participant was interested in.

2.4 Replay View

SensePath links the timeline to an externally captured screen video to provide additional information about the participant's behavior during the sensemaking session. When a researcher selects an action in the timeline, the replay view automatically jumps to the corresponding part of the screen video when the action is about to start. This avoids manual search within the video, which can be time consuming. After selecting an action in the timeline, a researcher can first check the web page in the browser view and then start the video playback in the replay view if she wants to find out more. The playback automatically stops when it reaches the end of an action, avoiding watching other irrelevant part. Alternatively, the researcher can choose to allow the video to continue; if so, the corresponding action in the timeline will be highlighted as the video progresses.

2.5 Transcription View

Detailed information of an action can be revealed by mouse over; however, it is inconvenient to do so for a set of actions. The transcription view addresses this issue by simultaneously presenting the details for all selected actions, in a tabular format (Fig. 1D). For each action, this view shows its starting and ending time, action type, assigned themes, and an automatically generated description such as "37 seconds spent in searching Best Western George Town Hotel and Suites". This description is based on a predefined template for each different action type with advise from the aforementioned participatory design session. The researchers are allowed to edit the description to better reflect what they think. Row backgrounds match the color of action type icons in the timeline view. The design of this view resembles the transcript interface of popular video transcribe software packages to reduce the learning efforts required.

2.6 Evaluation

We conducted a user-centered evaluation of the SensePath tool to establish an understanding of its use by an experienced qualitative researcher. We first conducted a number of user studies of participants carrying out an online sensemaking task, and we then recruited an HCI researcher with 7 years of experience in qualitative research to carry out an analysis of the sensemaking process of the users using SensePath. The researcher found the tool intuitive to use. The timeline view provided a useful overview of the participant's sensemaking process, enabling her to quickly identify recurring patterns of the participant and his rough strategy in conducting the task. The replay view complemented the timeline view with screen recording, enabling the researcher to investigate more fine-grained and continuous interaction.

3 SENSEMAP

While SensePath targets HCI/Visualisation researchers, SenseMap is designed for average users and everyday sensemaking tasks. To understand the requirements, we first conducted a semi-structured interview with nine participants to explore their behaviors in conducting online sensemaking for their daily work activities. These behaviors led to a sensemaking model for user behaviors on the web: users iteratively *collect* information sources relevant to the task, *curate* them in a way that makes sense, and finally *communicate* their findings to others. This is a simplified version of Pirolli and Card's sensemaking model [11]. We conducted a series of design workshops to derive requirements using these user behaviors and model and discuss design options to address them. All the requirements can be found in the original paper [8].

SenseMap is implemented as a Chrome extension with three linked views as show in Fig. 7.

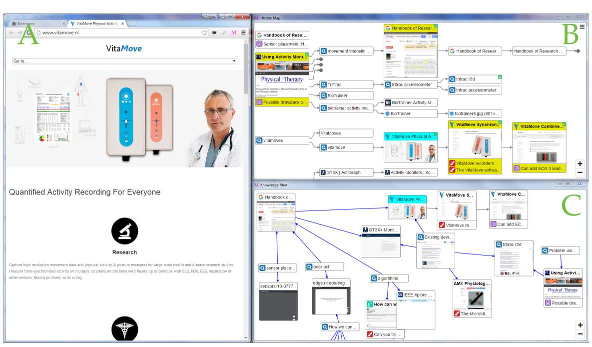

Figure 7: Three linked views of SenseMap. **A:** This is the standard browser with additional sensemaking and provenance support. **B:** The history map captures and visualizes user actions to provide an overview of the sensemaking process. **C:** The knowledge map enables users to curate and make sense of the most relevant information to their tasks.

3.1 Browser View

This is a standard web browser with additional sensemaking support such as highlight and annotation (Fig. 7A). User interaction is also captured using the same mechanism discussed in Sect. 2.1.

3.2 History Map

This map provides an overview of the sensemaking process using the captured actions and their provenance (Fig. 7B). An action is represented as a bar with an icon indicating its type and text showing the contextual information similarly to Fig. 2. Highlights and annotations of the same web page are grouped together as in Fig. 8. They are located in separate rows below the web page title. By default, just a few highlights and annotations are shown to ensure a reasonable height for the page. All of them can be revealed using a menu available when hovering on any highlight or annotation.

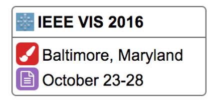

Figure 8: A page with one highlight and one note.

To help provide a connection between the history map and the browser view, the action bar corresponding to the active browser tab is highlighted in cyan. Pages that have been opened but have not seen yet (could be the result of opening links in new tabs) are shown with a dashed border, which may help to remind the user on reading them. Fig. 9 shows an example of pages with these two states.

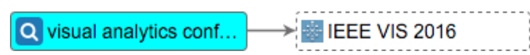

Figure 9: The user is active on a search result page (left bar) and opens a link in a new tab (right bar).

The history map displays all captured actions; however, probably not all of them are equally important and relevant to the sensemaking task. Therefore, it is necessary to allow users to assess the relevance of the collected information. We use the term *node* to refer to either a simple search action bar or a page containing many highlights. Three levels of relevance are provided, all through the menu available when hovering a node.

1. If a node is completely irrelevant, the user can *remove* it.
2. If a node is not quite relevant but the user wants to keep it to have a look at some point, they can *minimize* it.
3. If a node is very relevant, the user can *favorite* it.

When a node is removed, it and its links are removed from the map. When a node is minimized, it is collapsed into a small circle. This enables users to focus on other nodes and also save the display space. Favorite nodes are displayed with a yellow background and a thumbnail of the captured screenshot to increase their recognizability. Fig. 10 shows an example of minimized and favorite nodes.

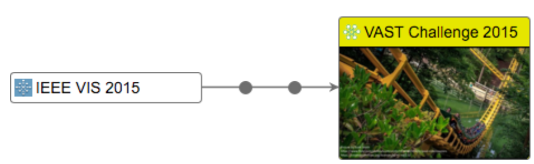

Figure 10: Nodes are pre-curated: two irrelevant nodes in the middle are minimized, whereas the last one is set favorite.

Nodes can reduce their size through zooming to accommodate more nodes within the visible part of the history map. By default, all nodes have the same width and the same maximum height, which allows a few words of the contextual text visible, and a reasonably large thumbnail image, which may help users recognize the visited pages. For each smaller level, both the node width and the number of highlights are reduced. The maximum height should be adjusted so that the ratio between it and the node width remains unchanged. At the smallest level, only the action type icon or a small thumbnail image is shown. Fig. 11 shows an example of different zoom levels applied onto the same node.

Figure 11: The same node with four zoom levels.

Node zoom level is explicitly controlled by the user using simple plus/minus buttons. When the collection of nodes exceeds the visible area, the user can pan the map to see them.

3.3 Knowledge Map

This map allows users to curate the information displayed in the history map (Fig. 7C). The curation process starts by adding nodes from the history map to the knowledge map. This is done via the *Curate* button in the menu available when hovering over a node. Nodes in the knowledge map have the same visual representation with those in the history map. The only difference is that thumbnail images of curated nodes are always made visible to improve their recognizability.

The limit of single dimensional ordering tabs from left to right is addressed in the knowledge map through the spatial organization of nodes. The user can freely move nodes by simply dragging them around. This enables the user to spatially group nodes and to assign different meanings to them. Fig. 12 shows an example of a knowledge map with three clear groups based on their locations.

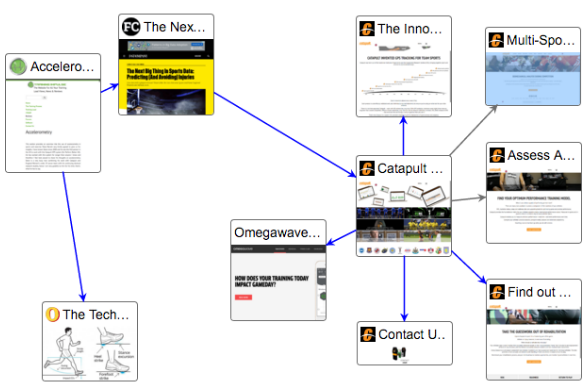

Figure 12: A knowledge map with three clear groups of nodes as the result of free movement.

Besides spatial grouping, seeing the casual relationships between collected information is also important to users in supporting sensemaking. A conventional representation is used to show this relationship: an arrow pointing from the cause to the effect. The user can add a casual relationship by clicking on the "cause node", holding it for half a second until the cursor changes to an arrow, then releasing the mouse on the "effect node".

When nodes are added to the history map, the provenance links among them are also copied to the knowledge map to provide an initial understanding of existing relations. Different colors are used to distinguish user-added links from provenance links.

Currently, SenseMap does not provide support for any formal argumentation methods. However, we think that the flexibility of spatial organization and relationships establishment can help the user apply their reasoning strategies. For instance, users can draw a link from a "hypothesis" node to its evidence. Then, they can move all supporting evidence nodes to one area and all counter evidence nodes to a different location to distinguish the two groups.

3.4 Communication

The final organization of curated information provides a complete picture of solving the sensemaking task, which makes it ideal for the user to present their findings. If the process is of interest, the history map can be used alongside the knowledge map. Moreover, the user can refer to raw data, via node revisitation, to support their presentation.

Both the history and knowledge maps can be saved as local files and loaded. This allows users to share their maps. Also, the user can create multiple copies of knowledge maps based on the same history map allowing customizing for various presentation purposes.

3.5 Evaluation

To explore how SenseMap is used, we conducted a user study in a naturalistic work setting with five participants completing the same sensemaking task related to their daily work activities. All participants found the visual representation and interaction of the tool intuitive to use. Three of them positively engaged with the tool and produced successful outcomes. It helped them to organize information sources, to quickly find and navigate to the sources they wanted, and to effectively communicate their findings.

References

[1] A. Cockburn, A. Karlson, and B. B. Bederson. A Review of Overview+Detail, Zooming, and Focus+Context Interfaces. *ACM Computing Surveys*, 41(1), 2009.

[2] J. Conklin. Hypertext: An Introduction and Survey. *Computer*, 20(9):17–41, sep 1987.

[3] J. Corbin and A. Strauss. Grounded theory methodology. *Handbook of qualitative research*, 17:273–285, 1994.

[4] D. Gotz and M. X. Zhou. Characterizing users' visual analytic activity for insight provenance. *Information Visualization*, 8(1):42–55, jan 2009.

[5] G. Guest, K. M. MacQueen, and E. E. Namey. *Applied thematic analysis*. Sage, 2011.

[6] G. Klein, J. K. Phillips, E. L. Rall, and D. A. Peluso. A Data-Frame Theory of Sensemaking. In R. R. Hoffman, editor, *Expertise out of context: Proceedings of the sixth international conference on naturalistic decision making*, pages 113–155. Mahwah, NJ: Lawrence Erlbaum Associates, 2003.

[7] N. Milic-Frayling, R. Sommerer, and K. Rodden. WebScout: support for revisitation of Web pages within a navigation session. In *International Conference on Web Intelligence*, pages 689–693. IEEE Comput. Soc, 2003.

[8] P. H. Nguyen, K. Xu, A. Bardill, S. Betul, K. Herd, and B. L. W. Wong. SenseMap: Supporting Browser-based Online Sensemaking through Analytic Provenance. In *IEEE Conference on Visual Analytics Science and Technology*, 2016.

[9] P. H. Nguyen, K. Xu, A. Wheat, B. L. W. Wong, S. Attfield, and B. Fields. SensePath: Understanding the Sensemaking Process through Analytic Provenance. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):41–50, jan 2016.

[10] C. North, R. Chang, A. Endert, W. Dou, R. May, B. Pike, and G. Fink. Analytic provenance: process+interaction+insight. In *ACM Transactions on Computer-Human Interaction*, pages 33–36. ACM, may 2011.

[11] P. Pirolli and S. Card. The Sensemaking Process and Leverage Points for Analyst Technology as Identified Through Cognitive Task Analysis. In *Conference on Intelligence Analysis*, 2005.

[12] K. Xu, S. Attfield, T. J. Jankun-Kelly, A. Wheat, P. H. Nguyen, and N. Selvaraj. Analytic provenance for sensemaking: a research agenda. *IEEE Computer Graphics and Applications*, 35(3):56–64, jan 2015.

Logging in Visualizations: Challenges of Interaction Techniques Beyond Mouse and Keyboard

Tom Horak* Ulrike Kister* Konstantin Klamka* Ricardo Langner* Raimund Dachselt*

Interactive Media Lab, Technische Universität Dresden, Germany

Abstract

Due to the deployment of novel interaction techniques, additional challenges for logging purposes in information visualizations arise. In this position paper, we discuss specific challenges regarding four different example setups illustrated with projects of our own. In each setup, various aspects need to be considered to enable, e.g., a meaningful logging of (multiple) input streams or the replaying of logs. We do not aim to provide a technical solution for logging interaction in the various setups, but rather want to share our insights and experiences from a set of projects that apply novel interaction techniques and multi-display setups to visualizations.

1 Introduction

As an ongoing upward trend in research, various novel interaction techniques are deployed to information visualizations and enhance the way we work with those visualizations [5]. Through the usage of additional modalities (e.g., touch, gaze, spatial position) or multi-display environments, the visualization systems are getting more complex. Of course, this also affects the logging of interactions and raises several questions, such as: What data can be logged and in which form could it be stored? What data has to be logged to allow making sense of the logs at a later time? How to handle large amounts of data and noise?

In this position paper, we aim to raise awareness of these challenges by considering four example setups using various input modalities. Although the given questions are also relevant in classic WIMP interfaces, the capturing process itself is in such systems relatively easy as there is typically a single event source. Also, it is possible to replay or simulate user interactions, since they do not depend on an outer context. In contrast, enabling novel interaction techniques causes more complex setup-driven challenges to arise. We provide four of our own projects going beyond WIMP interfaces to discuss the specific challenges of logging when merging multiple input streams, the possibility of replaying logs for interpretation, and the complexity and size of logged data.

2 Touch Input

We previously presented several multi-touch concepts to enable fluid interaction for star plot visualizations [4]. This involves rearranging axes via drag (Fig. 1a), splitting up axes on double-tap (Fig. 1b), scaling axes via pinch, or resetting the visualization via a wiping gesture. Thus, the interaction comprises single-touch, multi-touch, and gestural input.

Regarding the logging, single-touch as well as some multi-touch inputs (e.g., two finger drag) can be treated similar to mouse input in classic WIMP interfaces: They can be discrete events (e.g., tap) fired on a specific visualization element or continuous events (e.g., drag). For the latter, the interaction can be logged with all individual events, selected events (i.e., sub-steps), or as a single event. This consideration affects how granular a history or replay functionality can be realized.

*e-mail: {tom.horak, ulrike.kister, konstantin.klamka, ricardo.langner, raimund.dachselt}@tu-dresden.de

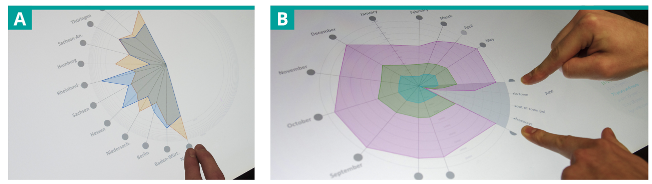

Figure 1: Fluid multi-touch interactions for star plot visualizations [4] require logging of both discrete and continuous input events (e.g., tap, drag).

In contrast, touch interactions like pinching have two or more input points defining a parameter (e.g., scaling factor) and the target. These multiple event sources require synchronized time stamps or storing as a combined event. Again, the granularity of stored (sub-)events can vary. Finally, gestural inputs (e.g., wiping) consist of a sequence of events from one or multiple input points, which are recognized as discrete gestures by the system. Therefore, they can either be logged as the recognized gesture or as the event sequence, which however requires the same gesture detection at the time of replay.

3 Spatial Arrangement of Mobile Devices (2D)

In a second example focusing on spatial interactions, we investigated how visualizations in multiple coordinated views can get tangible by distributing them to mobile devices [3]. We propose to use the spatial arrangement of these mobile devices to combine different visualization views and thus to enhance, e.g., visual comparison. For instance, this can be reached by reordering elements, flipping the view (Fig. 2a), or scaling of axes (Fig. 2b).

Figure 2: When adapting and synchronizing tangible visualization based on the spatial arrangement of mobile devices [3], individual data streams have to be joined for logging.

Besides keeping track of interactions on each device, it is now also necessary to know the applied combination or linkage of visualizations (similar to multiple coordinated views). Hence, the visualization state can now also depend on the state of other visualizations. This can be tracked by either logging the computed state as well as updates for each visualization or the initial state combined with linkage information. The deriving challenge for logging is to decide how to record this state, thus balancing between amount of stored data and how easily logs can be restored.

Furthermore, as the setup is also extended by a new modality, the spatial arrangement of devices, it is necessary to log the device location (position and orientation). The required values can be distributed (i.e., provided by the devices) or centralized (i.e., provided by an external tracking system) and be relative or absolute. Whereas relative values require to store pairwise relations to be able to reconstruct the device locations, the absolute values require calculation steps during restoring. In both cases the provided values can be noisy and high-frequency, resulting in a large amount of data to be logged. For instance, logging 3DoF (4 Byte per float value) with 60 Hz tracking frequency already results in 720 Byte per second per device that can significantly increase further depending on the logging format (e.g., XML syntax).

It is important to be aware of the fact, that logs cannot be equivalently replayed in such a setup as devices would have to be moved automatically. Furthermore, since the combinations also utilize device properties (e.g., size), the system state cannot be transferred to other setups with different devices. Although it is possible to virtually simulate the devices and their content on a larger display, the tangible characteristic of the interaction concept would be lost.

4 Spatial Navigation with a Mobile Device (3D)

Instead of arranging mobile devices on a 2D surface, we also investigated the combination of wall-sized high-resolution displays with spatially tracked mobile devices for graph exploration in a 3D space [1]. Our concepts focus on supporting selection, presenting additional information, or applying lens functions (Fig. 3). By tracking the device's position in space, the system can associate the user's actions with the device to individual graph elements presented on the display wall.

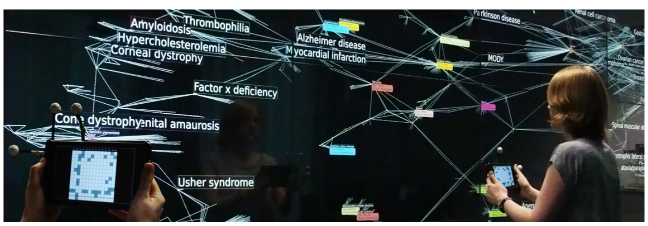

Figure 3: When using mobile devices for focus views in front of large wall-sized displays [1], all log events must retain their relation to the wall-sized visualization.

Similar to the example in 2D space, the mobile device's location (3D position and orientation) is essential for the logging process. However, the state of the current (part of the) visualization on the mobile device largely depends on the relative position to the visualization on the large display as they function as focus and context. Hence, to create a meaningful log, captured visualization states on the individual displays need to be fundamentally interwined and cannot be separated. Furthermore, touch interaction on the mobile device may manipulate and locally affect the visualization on the large display wall. Similar, interactions with the mobile device (both the simple presence/position and mobile device gestures) will have to be logged in relation to elements on the display wall. Complexity increases when multiple people (and devices) move in front of the display.

While individual touch events on both displays and the tracking data stream from the mobile device can be replayed from the logs, the impressions, the user's individual view on both the mobile device and the display wall, and the situation setup cannot be restored without active reenactment.

5 Body-centric Physical Navigation

In BodyLenses [2], we explore the design space and usage of body-centric movements for interactive visualization lenses. In the application example of a graph explorer, we used the body position and shape (tracked by a Kinect) in front of a display wall to apply lenses onto a graph visualization (Fig. 4a). These lenses can be further configured through touch interaction on the display wall (Fig. 4b).

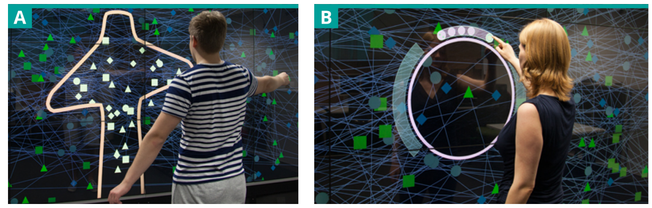

Figure 4: Body-centric interactions may require logging of individual body parts. These interactions can be used to influence tools on large display visualizations, e.g., interactive lenses [2].

Body-centric interaction may require tracking of not only one single position but the positions of multiple body parts (e.g., hands, head, arms, legs). The sum of these joints are skeletons already extracted from the Kinect video data stream. While event-based gesture recognition can be handled similar to mouse or touch-based interpretation, the adaptation of visualization or interactive tools like lenses requires continuous position data. This tremendously increases the size of the logged data and requires additional thought of which frequency and granularity is required for logging. Due to the highly personal shapes, restoring the data from the logs is even more complex and replay is nearly impossible.

6 Discussion & Conclusion

As illustrated by our examples, one major challenge when logging interactive visualizations in novel display environments is to handle the various input streams. The synchronization of these individual streams may be already difficult, and even more so when no central server organizes communication. Besides the need of synchronization, each stream can be high-frequent with multiple DoF and thus result in large log files. Therefore, it is important to consider filter mechanisms that are able to remove noise as well as unnecessary data (e.g., unchanged positions). An additional way to reduce the data amount is using delta encoding, i.e., only the changes are being logged. However, these can increase the complexity when restoring or analyzing the logs.

Since in our projects we incorporate physical interactions (e.g., spatial position or arrangement), providing a history or a replay functionality based on the logged data may not be possible. This is a crucial drawback of replaying, as the interaction order as well as the specific arrangement, position, or field of view of users during the interaction steps might affect the number, type, and quality of insights. Furthermore, in some cases it may not even be possible to internally log all user interactions (e.g., point of view, conversations) requiring extra video recording. As stated before, a session could be replayed in a pure virtual way, probably even in VR environments, but would still lack the important immersion during interaction.

All in all, novel interaction techniques come along with new challenges for logging in visualizations, especially regarding handling of input streams and providing the possibility of replays. Besides the used modalities in our example projects there exists many more (e.g., pen, gaze) that, however, face the same challenges. In the future, modern technologies such as augmented/virtual reality could even require additional considerations when enabling everywhere or immersive information visualizations. At the same time, logging these novel interactions and analyzing them afterwards could enhance our understanding of how people read visualizations on their own or discuss them during collaboration. Of course it is also an interesting challenge to find appropriate visualizations to support gaining these insights.

References

[1] U. Kister, K. Klamka, and R. Dachselt. Supporting graph exploration tasks on display walls using spatially-aware mobile devices. In *Proc. InfoVis '16*. IEEE, 2016.

[2] U. Kister, P. Reipschläger, F. Matulic, and R. Dachselt. Bodylenses – embodied magic lenses and personal territories for wall displays. In *Proc. ITS '15*, pages 117–126. ACM, 2015.

[3] R. Langner, T. Horak, and R. Dachselt. Towards combining mobile devices for visual data exploration. In *Proc. InfoVis '16*. IEEE, 2016.

[4] R. Langner, U. Kister, C. Engel, T. Horak, and R. Dachselt. Towards a fluid interaction concept bringing star plots to interactive displays. In *Proc. ITS '15*, pages 349–354. ACM, 2015.

[5] B. Lee, P. Isenberg, N. H. Riche, and S. Carpendale. Beyond mouse and keyboard: Expanding design considerations for information visualization interactions. *IEEE Trans. Vis. Comput. Graphics*, 18(12):2689–2698, 2012.

Leveraging Interaction History for Intelligent Configuration of Multiple Coordinated Views in Visualization Tools

Andrew Pachuilo* Texas A&M University

Eric D. Ragan† Texas A&M University

John R. Goodall‡ Oak Ridge National Laboratory

Abstract

Visualization tools can take advantage of multiple coordinated views to support analysis of large, multidimensional data sets. Effective design of such views and layouts can be challenging, but understanding users' analysis strategies can inform design improvements. We outline an approach for intelligent design configuration of visualization tools with multiple coordinated views, and we discuss a proposed software framework to support the approach. The proposed software framework could capture and learn from user interaction data to automate new compositions of views and widgets. Such a framework could reduce the time needed for meta analysis of the visualization use and lead to more effective visualization design.

Index Terms: H.5.2 [Information interfaces and presentation (e.g., HCI)]: User-centered design;

1 Introduction

Analysis of large, multidimensional data sets is challenging due to the need to inspect and interpret different attributes or data types concurrently. To address this challenge, visualization tools often provide multiple views that allow analysts different perspectives of the data [7]. With the application of common methods such as small-multiple views, focus-plus-context viewing, and brushing and linking, multiple views can be highly effective for allowing analysts to inspect different properties of complex data sets.

In practical scenarios, analysts tend to rely more heavily on reduced subsets of all available data, and different views might be preferred for different people or certain strategies. From a design standpoint, predicting what data attributes an analyst will find most useful can be challenging, and with high enough dimensionality, it is impractical to show all dimensions. Filtering data based on certain attributes can help speed up analysis by reducing the amount of data visualized at a time, but having enough widgets to adjust all possible data types or attributes would be unwieldy or impossible due to limitations in screen space. Additionally, certain data attributes might be understood more easily when viewed together, so it would be beneficial to keep such views in close proximity to make analysis more efficient.

With so many considerations, determining an appropriate configuration of views and widgets is a non-trivial task for the design of visual analysis tools. Researchers have proposed recommender systems to help select appropriate views based on properties of the data and common visualization guidelines (e.g., [4, 8]). However, for longer analysis sessions, it is important to consider user preferences and strategies. We propose an approach for improving the design and layout of multiple coordinated views by analyzing user interaction patterns collected through system logs. Once enough logs are collected they will be automatically processed and data mined to gain more insight about users and interpreted by an intelligent configuration process to create a new design. The recommended designs can be rolled out for use and additional log collection for iterative learning and design improvement. In our current work, we are considering this problem and approach for cyber security visualization tools.

*e-mail: apachuilo@tamu.edu

†e-mail:eragan@tamu.edu

‡e-mail:jgoodall@ornl.gov

2 Cyber Security Visualization Scenario

While our approach is not necessarily limited to any particular domain, we discuss our research of intelligent view configuration in the context of cyber security analysis. In our research, we work with cyber security analysts tasked with investigating and identifying of suspicious network activity. Cyber analysts must routinely monitor and sift through a large collection of data with numerous fields. Designing visualizations for such tasks can be difficult due to the exploratory nature of the task, the high volume of data, and the continuous streaming of incoming data. In addition to the raw data, cyber tools also often incorporate alerts from signature-based detection systems or analytic techniques (e.g., anomaly detection) to help flag potentially interesting or suspicious items. Consider a cyber security system that collects intrusion detection alerts from network activity across the globe. An analyst may wish to filter the data to only view intrusion alerts related to a specific source country over a specific port. To further simplify the task, they may opt to limit viewing to only those alerts flagged as the highest priority level by the system. As part of the analysis, the analysts might also be looking for recurring relationships over a certain period of time to better protect current systems, and these relationships could be easily spotted via certain filtered views of the data.

With such complex data and layered analysis goals, multiple views can be of assistance in helping analysts to make sense of different types of information together. At the same time, multiple widgets can help analysts to filter data, choose preferred visualization methods, or request additional data types for inspection. Previous tools have taken this approach. For example, the *Time-based Network traffic Visualizer* by Goodall et al. [2] providing a focused view on the packet level in the context of a network traffic view. In another example, Noel et al. [5] use multiple views to show network attack graphs, matrix representations, vulnerability details, and user annotations.

In our own work, we are also designing tools with multiple coordinated views. Figure 1 shows a screenshot of our cyber analysis prototype. The figure shows a collection of coordinated bar charts, histograms, and a map view to provide a composite view of cyber alerts. The views are interactive and double as widgets to filter or link selected attributes in other views. Having many multiple coordinated views and widgets together can be useful, but a designer's a priori view layout may not be optimal. Using this tool, we discuss a method for capturing and studying analysis patterns in order to improve the composition of views and the effectiveness of the visualization.

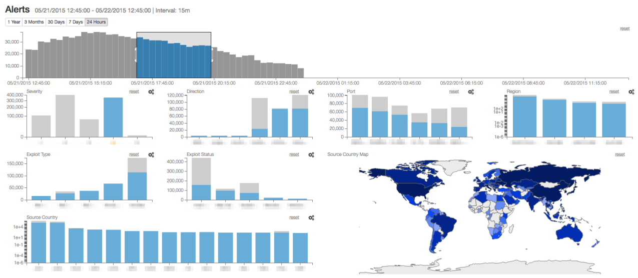

Figure 1: A cyber analysis prototype with multiple coordinated views.

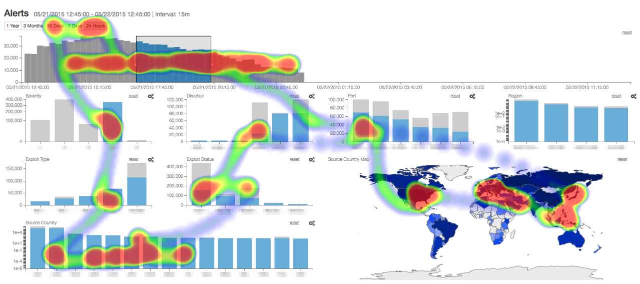

Figure 2: Interaction patterns (as shown here by a heatmap overlay) can reveal preferred views.

3 Intelligent Configuration of Multiple Views

To address the challenges in designing multi-view visualization tools, we discuss a general approach for improving the process of configuring views by capturing and data mining user interaction logs. Using this method, we can learn various strategies users take during analysis via their interactions and adopt them into a new configuration that better fits their needs. The methods requires a highly composable software framework that supports both logging of various interaction types and flexible configuration of the tool's visualizations and view layout.

The most useful interaction data to log will depend on the purpose for using the history data [6]. For our current cyber analysis scenario, we are most interested in eye tracking, mouse movements, keystrokes, and visualization meta-data (e.g., view state and data properties associated with interactions). Mouse and keyboard input demonstrates basic interaction history, eye tracking data provides a record of informational attention that may be independent of system input, and widget meta-data is important for matching input and viewing data with data state to learn analysis strategies.

Once a substantial amount of data has been captured, it can be mined to learn user priorities, usage patterns about how multiple widgets are used together, and which views are most beneficial for a given task. Additionally, interaction patterns could inform predictive models about probable sequences of actions. This information could then be interpreted to compose a new design that better streamlines user mouse movements, better groups certain widget or views, removes less useful items, and adds new views. Once the new design has been created, it will be immediately rolled out so that more information can be collected about the new design to provide richer insight for designing future iterations.

Implementation of such an approach will require a software framework that automates the processes of collecting interaction data, learning interaction patterns, and configuring views and widgets. The first step will be to create a framework for web-based analysis tasks for online testing to collect enough interaction logs for data mining. After the first iteration of collection, the data will undergo exploratory data mining with various methods to find what works best at understanding user strategies, movement, and the data set. Possible methods include using quality metrics [1] to better understand user strategies, exploring various machine learning algorithms for creating predictive models, and exploring methods for generating adaptive configurations for the layout [3] to better streamline interaction with the visualization.

This information can be used to create an intelligent system to interpret interaction data with consideration for specific users and particular data sets to create new layouts. The system would then roll out the new designs for use to gain additional interaction data for future iterations. With the generation of multiple layouts, it could also be valuable to find a way to map certain analysis strategies to different layouts so that particular layouts could be better optimized for certain tasks.

The automated framework should also include support for visualization tools to aide in the meta analysis of the framework itself. Determining what visualization designs would be most beneficial for meta analysis of interaction history would require additional research alongside exploring the various metrics and machine learning methods to use for data mining. These tools would help with understanding usage patterns for deciding possible metrics and machine learning approaches as a means to test the new designs for the framework to automatically generate. Figure 2 shows an example of a heatmap visualization tool that could help compare different layout configurations and visualization designs to help assess the effectiveness of the automation framework. Overlaying two heatmaps—one for eye movement and one for mouse interaction—for a certain time period of usage could also help researchers understand if a sensible decision was made during rearrangement of widgets. Another research goal for the development of the framework is to find a way to incorporate widget and view metadata into these tools to help make sense of the framework's recommended outcomes when considering user strategies.

Such a software framework that automates the configuration of multiple views could aide in the creation of more effective visualization design and reduce the amount of time spent manually doing meta analysis of the visualization. In future work, we plan on implementing and testing the proposed approach for creating an appropriate configuration for cyber analysis tools and multidimensional data.

Acknowledgements

This material is based on work supported by NSF 1565725. This manuscript has been authored, in part, by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. This research used resources of the Center for Computational Sciences at Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DEAC0500OR22725.

References

[1] E. Bertini, A. Tatu, and D. Keim. Quality metrics in high-dimensional data visualization: An overview and systematization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2203–2212, Dec 2011.

[2] J. R. Goodall, W. G. Lutters, P. Rheingans, and A. Komlodi. Preserving the big picture: visual network traffic analysis with tnv. In *IEEE Workshop on Visualization for Computer Security, 2005. (VizSEC 05).*, pages 47–54, Oct 2005.

[3] P. Langley. Machine learning for adaptive user interfaces. In *Annual Conference on Artificial Intelligence*, pages 53–62. Springer, 1997.

[4] J. Mackinlay, P. Hanrahan, and C. Stolte. Show me: Automatic presentation for visual analysis. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1137–1144, 2007.

[5] S. Noel, M. Jacobs, P. Kalapa, and S. Jajodia. Multiple coordinated views for network attack graphs. In *IEEE Workshop on Visualization for Computer Security, 2005. (VizSEC 05).*, pages 99–106, Oct 2005.

[6] E. D. Ragan, A. Endert, J. Sanyal, and J. Chen. Characterizing provenance in visualization and data analysis: An organizational framework of provenance types and purposes. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):31–40, Jan 2016.

[7] M. Q. Wang Baldonado, A. Woodruff, and A. Kuchinsky. Guidelines for using multiple views in information visualization. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '00, pages 110–119, New York, NY, USA, 2000. ACM.

[8] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE transactions on visualization and computer graphics*, 22(1):649–658, 2016.

Requirements for Visual Interaction Analysis Systems

Yi Han*
Georgia Institute of Technology

Gregory D. Abowd†
Georgia Institute of Technology

John Stasko‡
Georgia Institute of Technology

Abstract

Designers who deploy visualization applications usually want to assess how those applications are being used in the field. A promising and scalable method for understanding such use is to collect event logs of people's interactions with the applications. The challenge is how to then analyze the interactions in the logs in order to discover insights. Researchers have used visual analytics to support this analyst-driven process with some success. However, we found that existing visual interaction analysis systems are limited in their flexibility, scalability, and generalizability to fully support this challenging task. In this article we identify the primary tasks of interaction log analysis, discuss the main units of analysis, and derive a set of system requirements to inform the design of future visual interaction analysis systems.

Index Terms: H.5.2 [Information Systems]: Information Interfaces and Presentation—User Interfaces

1 Introduction

Designers who deploy visualization applications often seek to assess how those applications are being used in the field. By examining application usage, its designer can begin to understand the usability and utility of the application, learn about its users, and understand usage patterns/analysis methods. One promising source of such information is an interaction log. Modern visualization applications routinely incorporate a multitude of interactions to support the flexible, exploratory analysis processes of their users. While it has become easier to collect interaction logs, it is still challenging to effectively analyze them.

We consider the analysis of visualization application interaction logs to be highly exploratory, and thus closely aligned to other types of sense-making activities that involve repeated cycles of foraging and gathering information, and then reflecting on those findings to generate new schemas and hypotheses about what is actually occurring [20]. An analyst, who typically is the designer of the visualization application, frequently might explore interaction logs without explicit questions or goals in mind, or the goals may change over time. The detailed, subjective, and open-ended nature of the analysis process may be overwhelming, especially to those who rarely conduct this type of analysis.

We believe that these analysis needs suggest a visual analytics solution. Visual analytics is particularly suitable for interaction log analysis because it effectively combines automated computational analysis with human exploration and guidance, especially when applied to large collections of data [15]. Furthermore, a visual analytics approach is helpful when analysis goals are dynamic and possibly imprecise. Unfortunately, existing visual interaction analysis systems are not flexible, scalable, and generalizable enough to support this need. We speculate that this is a reason why interaction data from design studies are seldom extensively analyzed in the visualization literature, despite such data's prominent role in the analysis process.

The objective of this work is to establish a set of requirements for visual interaction analysis systems. We make the following contributions. (1) We identify an analyst's tasks and discuss how these tasks can be accomplished by analyzing interaction logs. (2) We identify and present three analysis units that are essential for the process. (3) For these tasks and analysis units, we derive a set of requirements for visual analytics on interaction logs that emphasize flexibility, scalability, and generalizability.

2 Related Work

Researchers have been using visualizations for interaction log analysis on systems other than visualization applications in a variety of domains for about two decades. The visualizations typically have been static representations of users' interaction patterns using heat maps for general UI interactions [10], trees for web navigations [5, 19, 27, 31], graphs for social network interactions [1, 23], or line and bar charts for online video interactions [6, 16]. While useful, at times such visualizations can be limiting because the view cannot be modified or transformed. Therefore, researchers recently have turned to interactive visualization systems to more dynamically explore interaction data. These interactive visualizations include interconnected bar charts and graphs for web search behaviors [17], timelines for web interactions [18], stacked area charts for online video interactions [25], icicle trees for social network and web interactions [24, 29], connected matrices [32], and even visual clusters of sequences [28]. Although these visualization techniques have been applied to interactions on *non-visualization applications*, we can learn from those experiences and examine which techniques may apply well to interactions on *visualization applications*.

With respect to the use of visualization to analyze interactions on visualization applications, researchers have used static visualizations such as state transition graphs [22], colored bars [14], and graphs and scatterplots [12]. Some used interactive visualizations as well. For example, Jeong et al. [13] created two interactive visualization systems, one for exploring interaction data on a timeline and one in treemaps. Blascheck et al. [2] also designed an interactive visualization system to analyze interactions on a text visualization application. They used a line chart as the primary visualization to show interactions with think aloud and eye movement data. A particular strength of their work is in the computational analysis of the data – The system can automatically identify similarities between users and help analysts find usage patterns. Such computational methods are particularly useful when the amount and variety of interactions are large.

These prior systems depicting visualization application interactions have limitations, however. First, most of these projects were research prototypes that visualize a relatively small-scale interaction dataset [2, 7, 13, 14, 22]. For example, Blascheck et al. [2] only studied 16 participants, each using the visual analysis application once, in a lab. In a real-world deployment, the amount of log data can be significantly larger than what these systems can support. Heer et al. [12] seemed to have visualized a realistic real-world dataset but one of the applied visualization techniques, the behavior graph, did not seem to scale well. A second limitation of these projects was that, during analysis, they each organized interactions into a single, subjectively-determined set of cat-

*e-mail:yihan@gatech.edu
†e-mail:abowd@gatech.edu
‡e-mail:stasko@cc.gatech.edu

egories [7, 12, 13, 14, 22]. These categorizations were often necessary to reduce the complexity of the data for meaningful patterns to emerge. However, most of these projects only selected and used one type of categorization. For example, Reda et al. [22] categorized interactions by whether the interactions significantly changed the visualization layout. In realistic analysis tasks, the categorizations should depend upon and iteratively change with an analyst's goal. Methods for flexibly supporting this goal are largely missing in past work. A third limitation is that many of these visualizations were designed specifically for analyzing interactions from one particular visualization application in one specific study [7, 12, 13]. Some data dimensions were hard-coded into the analysis systems, making results difficult to generalize. For example, Jeong et al.'s systems [13] specifically defined and laid out visualizations based on their data dimensions, such as the views of the analyzed visualization application. This design limits the generalizability of their systems. We believe that to fully address these limitations, one needs to re-examine interaction analysis tasks fundamentally.

3 Interaction Analysis Tasks

Based on our review of past research and our own experiences, we have identified the following key tasks or goals of an analyst when seeking to understand how his/her visualization application is being used.

Assess Usability

An analyst seeks to understand how easy it is to use and learn a visualization application. This information could be gleaned from interactions in many ways. For example, features that are easy to use are likely to be performed more frequently, assuming they are important to the application. If a feature is not employed much but was expected to be frequently used, the feature may not be designed well.

Diving deeper, the learning curve of an application could be assessed from interactions. For example, one hypothesis is that when a user is less familiar with an application, his/her interactions would be more diverse and random, showing an experimental usage of features. But as the user discovers which features are more useful from experience, his/her interactions would become more focused on those features and use them in a consistent manner. By assessing the change trajectory, an analyst could infer the learning curve of his/her visualization application. If an analyst seeks to assess which features are more challenging to learn, he/she may inspect how long on average it takes for a user to go from first encountering a feature to using it efficiently.

An easy-to-use application is often efficient to use. In non-visualization applications, fewer mouse clicks and shorter mouse movement distances in usage sessions are indicators of efficient UI design. However, in visualization applications, when visual exploration is the task and broader understanding of data is the goal, these traditional usability indicators may need to be interpreted differently. A visualization application that encourages extensive interactions may help an application user more easily explore the data.

Assess Utility

Card et al. claimed that "the purpose visualization is insight, not pictures" [4]. A good metric to assess the utility of a visualization application is to examine whether its users are able to find insights. A typical method for determining this answer is to interview users. But can this information be acquired from interaction logs instead as interview data are more difficult to collect? Gotz and Zhou defined a set of interaction types as "Insight Actions" in their work because some interactions could be connected to insights [9]. For example, they called Bookmark and Annotate events "Visual Insight Actions". These interactions could indicate an insight was discovered because finding an insight is one reason for bookmarking and annotating views in visualizations. But not all bookmarks and annotations are indications of insights. For example, an annotation could be used to add missing information to the data. Therefore, to differentiate insight-indicating bookmarks and annotations, an analyst would need to manually examine the bookmarked or annotated content to determine which ones are actually insights. The downside is that this analysis would require a significant effort from the analyst and sometimes the bookmarked or annotated contents are difficult to interpret as they are generated by someone else. As a result, visualization designers who plan to use visualization interactions to infer insights might want to explicitly prompt their users to tag insights in bookmarks and annotations to allow automatic classification of insights from interaction logs. After insights are determined, a simple count of them might be sufficient to approximate the utility of a visualization application.

Learn About Users

An analyst seeks to learn more about a visualization application's users from usage behaviors. For example, which people are "expert" users and which ones need some extra help? For a specific feature, which users are able to properly employ it? Any one feature could be implemented with multiple UI interactions. For example, zooming into a view could be implemented by clicking a Zoom-In button or selecting a Zoom-In menu item. Which method is preferred by users for their day-to-day tasks? When using an application for different types of analysis tasks or occasions, how do the users' behaviors differ? At an abstract level, an analyst seeks to find groups of users or sessions that exhibit certain/varying behaviors (e.g., different keyword search [2]) or examine the behaviors of users or sessions under certain circumstances (e.g., different display sizes [22]). The key to this analysis is to map the user information to the varying interaction patterns in the data. This information is helpful to an analyst for understanding user differences in visualization usages.

Understand Usage Patterns/Analysis Methods

An analyst seeks to explore the variety of ways a visualization application was used. Specifically, frequent ways of using the application, which form usage patterns, are of particular interest. Some of these patterns are expected by the analyst. For example, an analyst of interaction logs from a visual text analysis application would expect its users to extensively read the text documents. But what other usage patterns might there be? Kang et al. found several more specific usage patterns after studying the usages from Jigsaw [14]. For example, some users start from scanning all the documents first to filter out irrelevant ones, then read those remaining documents. Some users repeatedly search the document set with different keywords and read the documents in the search result. These usage patterns might not have been expected by the analyst. Therefore, finding the relative portions of these usage patterns helps the analyst gain a deeper understanding of the varying ways his/her application was operated in actual usage scenarios.

Some usage patterns of a visualization application may indicate that a visual analysis method (VAM) is taken. A VAM, which is sometimes called a visual analysis strategy [7, 14, 22], is a methodological and semantically meaningful way of operating a visualization application. Many researchers look for VAMs in their applications to learn about such semantically meaningful usage patterns. One example of a VAM is the Visual Information-seeking Mantra [26]. This VAM is indicated by a set of usage patterns that start from an "overview", followed by a mixture of "zooming" and "filtering", and then show "details" on demand. Shneiderman identified this VAM as being widely used in a variety of visualization applications. Other VAMs may only occur when certain types of data or visualization techniques are employed. For example, Kang et al. identified a set of VAMs (strategies) for analyzing text documents with a specific set of text visualization techniques [14]. These VAMs are less generalizable but are more contextually relevant to the application. Finding the set of VAMs employed in a visualization application is useful for understanding users' reasoning processes behind the usage patterns [7].

4 ANALYSIS UNITS

Within the analysis tasks above, we found that an analyst needs to find frequency distributions of not just individual interaction events but also groups of events in categories or sequences. Therefore, identifying these analysis units are vital to the analysis. But how should these events, categories, and sequences be defined and identified? Once they are determined, identifying their frequency distribution over time or any other contextual information should be relatively easy. In this section, we discuss these analysis units.

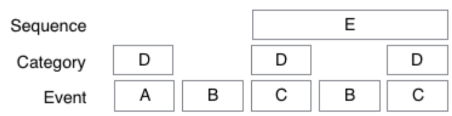

Figure 1: Interaction events. Individual events (ABC) could be categorized (A→D, C→D) or grouped by their sequences (DBD→E).

Event
We assume that individual interaction events are the basic unit logged. As shown in Figure 1, suppose a string of five events (ABCBC) of three types (A, B, and C) are logged in a usage session. Example events may include "clicking the Zoom-In button" or "scrolling the view." Events sometimes include the visualized data as a parameter. Such events are generally defined in the code that produces the log. Individual events can be analyzed directly but typically they are first filtered, categorized, and grouped into more semantically meaningful units to the analyst.

Category
Because a visualization application may have a large number of interactions, an analyst typically organizes them into a smaller set of categories that are more semantically meaningful and suitable to his/her analysis goal. For example, some researchers [11, 21] organized events into the intent-based interaction categories defined by Yi et al. [30]. Other researchers used other classification criteria, such as the view an event occurs in [14] or the significance of a layout change [22]. A categorization is illustrated in the second row of Figure 1 where events A and C are both classified into category D.

Two analyst-driven steps are required in the categorization process. An analyst not only needs to determine which categories to use for the analysis but also how the events should be mapped to the categories. First, the categories to be used should largely depend on the analysis goal. For example, if an analyst wishes to study the uses of different views in a visualization application, the analyst could categorize each interaction event by its view [14]. Second, determining the category that each event should be mapped to requires the analyst's subjective determination. Some determinations are easy, such as whether the interaction is within a specific view. Other determinations are more difficult, such as determining the interaction intent category from Yi et al.'s interaction taxonomy [30]. The more difficult categorizations typically require a semantic interpretation of the events and thus a potentially significant effort from the analyst. This labor requirement can become quite significant as reclassifications may frequently be needed when goals change over the course of analysis. As a result, it is important for an interaction analysis system to be able to flexibly and efficiently support this analyst-driven categorization process.

Sequence
An analyst typically examines sequences of interactions in order to identify longer and higher-level usage patterns (e.g., DBD→E in Figure 1). For example, to determine whether a specific system feature that requires multiple interactions is being used as expected, an analyst may need to look for a specific set of interaction sequences. Frequently occurring interaction sequences are considered particularly informative because they often represent useful or conventional ways of using an application. An interaction sequence may include both consecutive and non-consecutive interactions. For example, a sequence of "Inspect" actions is considered a "Scan pattern" in Gotz and Wen's work [8]. Conversely, an interesting sequence may be made up of non-consecutive interactions. For example, suppose an analyst wishes to determine whether people are following the Visual Information-Seeking Mantra [26] when they use an application. To do so, the analyst does not need to find overview, zoom and filter, and details on demand interactions necessarily occurring back-to-back. The interactions simply need to occur non-consecutively in the proper sequence for the analyst to determine if the mantra has been followed. Therefore, being able to identify sequences that may not only include events occurring back to back increases the flexibility in identifying longer and higher-level usage patterns.

5 SYSTEM REQUIREMENTS

Using the tasks and analysis units as a basis, we identify a set of requirements for a visual interaction analysis system to assist with interaction log analysis, specifically for supporting flexibility, scalability, and generalizability.

Support Event Organization (Flexibility)
An interaction analysis system needs to be able to support the flexibile oganization of events. Different analysts and situations may call for different organizations of interaction events because of their varying needs. For example, one analyst may be looking for underused application functions, another may be optimizing the interface, and a third may simply be seeking to understand the visual analysis methods people employ using the application. The same analyst may even change analysis focus "on the fly" as new discoveries about application use are made. Below we describe three specific requirements for flexibily organizating the logged interaction events that we believe an ideal interaction analysis system should provide.

a. Select events of interest
A system may log all interaction events that occur, but only a subset of the logged interaction events may be relevant for a given analysis goal. An analyst needs to be able to flexibly select these relevant events for further exploration. Otherwise, when many irrelevant events are kept in the analysis, the "noise-level" of the data may hide otherwise meaningful patterns. How to determine which events are relevant during an analysis session is very likely a subjective judgment of the analyst based on knowledge of the analysis goal and the interaction data.

b. Define analysis perspective
Analysts may approach log events with a wide variety of goals. Thus, it should be possible for analysts to flexibly define different analysis perspectives for classifying the events based on these different goals. An analysis perspective provides the means to differentiate events that can be based on different criteria. It essentially defines a set of categories for organizing events.

For example, a perspective for understanding visualization application commands and operations that are performed would simply be a set of categories of those commands and operations. One could consider this a relatively low-level perspective. An alternative perspective for understanding which and when application views are used would include a category for each view in the application. Kang et al. [14] employed this perspective when performing an analysis of usage of the Jigsaw visual analytics system. They identified Jigsaw interactions by the views in which they occurred in order to analyze overall strategies taken by each user. Yet another perspective might include a set of categories defined by user intent, that is, what was the intent of the person using the visualization application when performing an interface operation. Pohl et al. [21] used the interaction intent framework introduced by Yi et al. [30] to categorize user interactions with two visualization systems, VisuExplore and Gravi++, in order to explore and compare user strategies when using those systems. Guo et al. [11] also used those intent-based categories plus a new category called "retrieve" as "high-level actions" in their interaction analysis of a visualization application. As a final example analysis perspective, an analyst may be interested in the degree that application usage follows a well-known analysis method, such as Shneiderman's "Overview first, zoom and filter, then details on demand [26]." For this perspective, the set of categories would be those four activities defined in the mantra.

c. Categorize events

Once an analysis perspective is defined, an analyst next assigns the events to categories. For some perspectives, multiple interaction events can be considered "similar" and placed into the same category. For example, interface events such as clicking on buttons, spinning the mouse wheel, or choosing menu commands, when they are relevant to zooming in or out of the visualization, can all be placed into a "zoom" category. This ability to flexibly categorize interaction events is important for log analysis.

Provide Configurable Visualizations (Flexibility)

As discussed in the related work, many researchers have adopted different styles of interactive visualizations for use in their analysis processes. From these efforts, we learned that effective visualizations not only need to provide visual overviews of the interaction data, but also interactive features to support functions such as querying and filtering patterns on demand. Providing an overview of event sequences may be challenging due to size of the interaction logs, however. Moreover, which interactive features to include and how they should be designed in order to give analysts investigative flexibility and strength is still an important research question.

Include Automated Computational Assistance (Scalability)

Usage patterns at a large scale may be difficult to manually identify. Automated computational data analysis is essential to help analysts discover patterns from large interaction logs. For example, it is relatively easy and significantly faster to algorithmically identify and quantify frequent/infrequent events, categories, and sequences. After the computation, the output of these algorithms can be visualized to provide an analyst a summarized view of the information in a large scale interaction dataset. For example, frequent interaction sequences are commonly automatically extracted and visualized to identify higher-level usage patterns [2, 3, 11, 21]. This use of computational analysis significantly increases a person's ability to analyze the interaction data.

Apply to Any Visualization Application (Generalizability)

As mentioned earlier, a number of interaction analysis systems have been built for analyzing *particular* visualization applications. When focusing on only one visualization application, it is easier to custom-design an analysis approach and system with a very specific set of tuned views. However, when designing an analysis system that could be applied to any visualization application's interaction logs, the analysis system needs to be highly configurable to support varying log formats, interaction types, and analysis needs.

6 Conclusion

We presented a set of tasks, analysis units, and system requirements for designing future visual interaction analysis systems. We hope these requirements can bring about discussions on this challenging research topic.

References

[1] F. Benevenuto, T. Rodrigues, M. Cha, and V. Almeida. Characterizing user navigation and interactions in online social networks. *Information Sciences*, 195:1–24, July 2012.

[2] T. Blascheck, M. John, K. Kurzhals, S. Koch, and T. Ertl. VA2: A Visual Analytics Approach for Evaluating Visual Analytics Applications. *IEEE TVCG*, 22(1):61–70, Jan. 2016.

[3] E. Brown, A. Ottley, H. Zhao, Q. Lin, R. Souvenir, A. Endert, and R. Chang. Finding Waldo: Learning about Users from their Interactions. *IEEE TVCG*, 20(12):1663–1672, Dec. 2014.

[4] S. K. Card, J. Mackinlay, and B. Shneiderman, editors. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, 1999.

[5] S. K. Card, P. Pirolli, M. Van Der Wege, J. B. Morrison, R. W. Reeder, P. K. Schraedley, and J. Boshart. Information Scent As a Driver of Web Behavior Graphs: Results of a Protocol Analysis Method for Web Usability. In *Proc. of ACM CHI '01*, pages 498–505, 2001.

[6] K. Chorianopoulos. Collective intelligence within web video. *Human-centric Computing and Information Sciences*, 3(1):1–16, June 2013.

[7] W. Dou, D. H. Jeong, F. Stukes, W. Ribarsky, H. Lipford, and R. Chang. Recovering Reasoning Processes from User Interactions. *IEEE Computer Graphics & Applications*, 29(3):52–61, May 2009.

[8] D. Gotz and Z. Wen. Behavior-driven Visualization Recommendation. In *Proc. of ACM IUI '09*, pages 315–324, 2009.

[9] D. Gotz and M. Zhou. Characterizing user's visual analytic activity for insight provenance. In *Proc. of IEEE VAST '08*, pages 123–130, Oct. 2008.

[10] M. Gray, A. Badre, and M. Guzdial. Visualizing usability log data. In *Proc. of IEEE Symposium on Information Visualization '96*, pages 93–98, 130, Oct. 1996.

[11] H. Guo, S. Gomez, C. Ziemkiewicz, and D. Laidlaw. A Case Study Using Visualization Interaction Logs and Insight. *IEEE TVCG*, 22(1):51–60, 2015.

[12] J. Heer, J. Mackinlay, C. Stolte, and M. Agrawala. Graphical Histories for Visualization: Supporting Analysis, Communication, and Evaluation. *IEEE TVCG*, 14(6):1189–1196, Nov. 2008.

[13] D. H. Jeong, W. Dou, H. Lipford, F. Stukes, R. Chang, and W. Ribarsky. Evaluating the relationship between user interaction and financial visual analysis. In *Proc. of IEEE VAST '08*, pages 83–90, Oct. 2008.

[14] Y.-a. Kang, C. Gorg, and J. Stasko. How Can Visual Analytics Assist Investigative Analysis? Design Implications from an Evaluation. *IEEE TVCG*, 17(5):570–583, May 2011.

[15] D. Keim, G. Andrienko, J.-D. Fekete, C. Görg, J. Kohlhammer, and G. Melançon. Information visualization. chapter Visual Analytics: Definition, Process, and Challenges, pages 154–175. Springer-Verlag, Berlin, Heidelberg, 2008.

[16] J. Kim, P. J. Guo, D. T. Seaton, P. Mitros, K. Z. Gajos, and R. C. Miller. Understanding In-video Dropouts and Interaction Peaks Inonline Lecture Videos. In *Proc. of the ACM Conference on Learning @ Scale*, pages 31–40, 2014.

[17] H. Lam, D. Russell, D. Tang, and T. Munzner. Session Viewer: Visual Exploratory Analysis of Web Session Logs. In *Proc. of IEEE VAST '07*, pages 147–154, Oct. 2007.

[18] P. Nguyen, K. Xu, A. Wheat, B. Wong, S. Attfield, and B. Fields. SensePath: Understanding the Sensemaking Process Through Analytic Provenance. *IEEE TVCG*, 22(1):41–50, Jan. 2016.

[19] S. Pachidi, M. Spruit, and I. van de Weerd. Understanding users behavior with software operation data mining. *Computers in Human Behavior*, 30:583–594, Jan. 2014.

[20] P. Pirolli and S. Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proc. of International Conference on Intelligence Analysis*, volume 5, pages 2–5, May 2005.

[21] M. Pohl, S. Wiltner, S. Miksch, W. Aigner, and A. Rind. Analysing Interactivity in Information Visualisation. *KI - Knstliche Intelligenz*, 26(2):151–159, May 2012.

[22] K. Reda, A. E. Johnson, J. Leigh, and M. E. Papka. Evaluating User Behavior and Strategy During Visual Exploration. In *Proc. of ACM BELIV '14*, pages 41–45, 2014.

[23] F. Schneider, A. Feldmann, B. Krishnamurthy, and W. Willinger. Understanding Online Social Network Usage from a Network Perspective. In *Proc. of ACM SIGCOMM '09*, pages 35–48, 2009.

[24] Z. Shen, J. Wei, N. Sundaresan, and K.-L. Ma. Visual analysis of massive web session data. In *Proc. of IEEE LDAV '12*, pages 65–72, Oct. 2012.

[25] C. Shi, S. Fu, Q. Chen, and H. Qu. VisMOOC: Visualizing video clickstream data from massive open online courses. In *Proc. of IEEE VAST '14*, pages 277–278, Oct. 2014.

[26] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Proc. of IEEE Symposium on Visual Languages '96*, pages 336–343, Sept. 1996.

[27] M. Spiliopoulou and C. Pohle. Data Mining for Measuring and Improving the Success of Web Sites. *Data Mining and Knowledge Discovery*, 5(1-2):85–114, Jan. 2001.

[28] J. Wei, Z. Shen, N. Sundaresan, and K.-L. Ma. Visual cluster exploration of web clickstream data. In *Proc. of IEEE VAST '12*, pages 3–12, Oct. 2012.

[29] K. Wongsuphasawat and J. Lin. Using visualizations to monitor changes and harvest insights from a global-scale logging infrastructure at Twitter. In *Proc. of IEEE VAST '14*, pages 113–122, Oct. 2014.

[30] J. S. Yi, Y. a. Kang, J. Stasko, and J. Jacko. Toward a Deeper Understanding of the Role of Interaction in Information Visualization. *IEEE TVCG*, 13(6):1224–1231, Nov. 2007.

[31] O. Zaïane, M. Xin, and J. Han. Discovering Web access patterns and trends by applying OLAP and data mining technology on Web logs. In *Proc. of IEEE International Forum on Research and Technology Advances in Digital Libraries '98*, pages 19–29, Apr. 1998.

[32] J. Zhao, Z. Liu, M. Dontcheva, A. Hertzmann, and A. Wilson. MatrixWave: Visual Comparison of Event Sequence Data. In *Proc. of ACM CHI '15*, pages 259–268, 2015.

Logging Interactions to Learn About Visual Data Coverage

Tanja Blascheck*, Steffen Koch*, and Thomas Ertl*

*University of Stuttgart, Germany

ABSTRACT

Evaluating interactive visualizations requires examining the complex interaction techniques used in the visualization, but also makes it necessary to investigate which part of the data participants are exploring. In this paper, we discuss how logging of these complex interaction techniques may help people to understand the data explored in order to improve a visualization technique. By answering questions about how much, which part, or how often a part of the data was inspected, we can infer valuable information about the usefulness and effectiveness of an interactive visualization technique. This visual data coverage allows us to also make inferences about traceability and accountability of a visualization technique. We present experiences we made during our studies, discuss challenges, and point out future directions of this work.

1 INTRODUCTION

To increase insight generation based on interaction logs in the context of modern, interactive visualizations, it is valuable to also log the extent and kind of data surveyed with continuous interactions. This information about *visual data coverage* brings new potential to the *evaluation* of a visualization and its interaction design, *tracking* which portion of data a user inspected visually at a specific level of detail and suggestions on what to investigate next, as well as *accountability* of data analysis and insights. Therefore, we focus on these three topics and in particular on closely related questions regarding data coverage during visual analysis: how much of the data was explored, how often was the same data item inspected, which parts of the data were examined at which level of detail, and was the right data investigated?

In order to analyze visual data coverage, logging of this information is required. There is extensive work on analyzing interactions [18, 23, 4] and many approaches exist on how to categorize interactions [4, 23] or interaction costs [14]. We discuss the benefits and limitations of logging visual data coverage when using complex interaction techniques in context of evaluation, traceability, and accountability. We illustrate our experiences, insights, and challenges when collecting information about the data explored with a focus-and-context lens and share some preliminary ideas about how to analyze this data.

2 EVALUATION

Interactive visualizations are developed to explore data to find insights in the represented data. When developing an interactive visualization method, a designer makes design decisions. Evaluating a novel visualization technique requires finding evidence if users can find insights in the data while applying a technique. Thus, the goal of an evaluation is to analyze if the design decisions made were the right ones and can be understood by users to generate insights from the data.

Analyzing a visualization with complex interaction techniques, like focus-and-context techniques, requires looking at both how

*e-mail: firstname.lastname@vis.uni-stuttgart.de

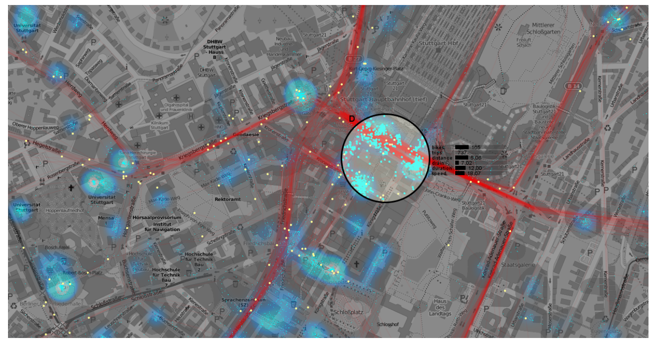

Figure 1: TrajectoryLenses is a Visual Analytics system allowing a user to explore trajectories displayed on a map using a focus-and-context lens. The heatmap shown depicts the end points of movement trajectories to guide the user in exploring the data.

users interact with the technique and how much data is explored visually at which level of detail. Recording the visual data coverage during a study, for example, by logging each data item investigated can be used to answer our questions. If the amount of data (how much data has been seen) is low, this can indicate issues with the efficiency of chosen interaction and visualization techniques. For example, some parts of the data might not be accessible with a certain visualization technique. Which part or which data items participants inspected can give insights about accessibility of the data. If an attention guiding method is used, for example, a heatmap indicating dense regions that are in particular interesting for close inspection, analyzing which part of the data was examined might also give insights about how well this attention guiding worked. If the heatmap led users to explore interesting parts of the data, this information can be inferred from visual data coverage. If a count is recorded each time a participant explored a data item in detail, an analyst can see which data items got more attention and which received less. This may indicate that some parts of the data were more interesting, more in focus, or more available. Another important question is whether or not a user examined the right data. Depending on the task, some part of the data may be more valuable or even necessary to complete a task successfully. This requires defining a ground truth from data items that need to be investigated for each task if this is to be assessed during evaluation. With this information it is at least possible to determine whether a user explored data important for solving a task or not.

Despite this, only taking into account the visual data coverage might not be enough to make clear if participants actually perceived or understood all the data inspected during exploration. So far, we have only looked at the visual data coverage logged during a study. For example, a high visual data coverage or a thorough investigation of one data item might be the result of the logging mechanism. Thus, just looking at this metric might not be enough and other parameters or data sources have to be combined to make better inferences. Eye tracking could be one means to investigate if participants actually explored a specific data item for a sufficient amount of time, or a user's attention passed over a visual detail without realizing its importance. Interaction logs have been correlated with other data sources such eye tracking [2, 3, 1, 19] or think-aloud

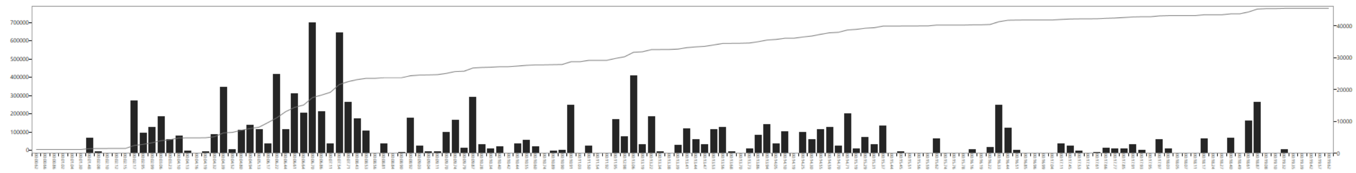

Figure 2: Visual data coverage graph showing the amount of data a user has visually explored over time (bar chart) as well as aggregated over the complete task duration (line chart). Note that the graph shows two different granularities: the overall visual data coverage (about 7,000,000 data items) and the visual data coverage of one time step (maximum about 42,000 data items).

protocols [7, 16, 10]. However, these approaches do not consider the actual data and visual level of detail inspected.

3 Traceability

Taking our idea one step further, visual data coverage analysis can also be used for tracing user interactions and guiding the user. Analyzing the traceability of the insight process has been proposed by Gotz and Zhou [9]. Their approach focuses on low-level versus high-level actions, where low-level actions are basic interactions. They do not consider the data that a user has investigated. Based on our questions, we can ponder on how visual data coverage information can be used in applications that continuously log user interaction. If the recorded data information is available immediately, it can be analyzed to guide the user to new data items or parts of the data that was not inspected visually yet. The information about how much of the data a user has explored can be used to make the user aware that s/he has only examined a small part or already investigated all available data. The part of the data a user has explored can be used to guide a user to parts of the data not analyzed yet. Additionally, if some ground truth is available for a task, a user can be guided to inspect the right data when solving a task.

For example, a visualization as shown in Figure 3 is useful to let users understand which parts of the data they explored in detail and which might deserve additional attention. The time spent for inspecting particular data items may be plausible if the interactive visualization is well designed and helped to steer users attention to exactly those details in the data that are interesting in the context of the task to be solved. With often underspecified tasks and explorative approaches, as supported by visual analytics solutions, this perfect match might not be guaranteed. Tracing the level of detail, the frequency in which data items were inspected as well as the quantity of explored data items and reflecting this to users, might help to prevent severe errors in assessing a situation. In the given example, the bar graph shows that some elements have not been investigated at all, yet. Extending this visualization to show all data items might help the user to see which parts were not inspected. Adding brushing methods and linking the visualization with the visual analytics tool that was used for exploration, for example, by clicking on an item to highlight the corresponding elements, could be a valuable means to reduce analysis faults from overlooking important information.

4 Accountability

Accountability has been discussed so far in different context. It has been used to describe 'truthfulness' of visualization approaches [21] which is inevitably hampered by data uncertainties and the impossibility of reflecting the full complexity of the real world. Reducing this problem certainly includes a critical reflection of a users analysis process which can be supported by improving traceability. The term accountability is also used with respect to security and privacy, for example, by Butin [6] and Weitzner [22] who suggest to make business transactions verifiable and dishonest or even illegal use of information transparent. While these aspects are not directly covered by our discussion, it would be interesting to indicate them if information foraging is supported through interactive visual interfaces. Many other semantic meanings of accountability exist in the context of HCI and other research fields [8]. Accountability is of special importance in decision-making situations like visual analytics accountability [15, 20]. If an expert using a tool is misled by the data and comes to a wrong decision, the question of who is responsible becomes an issue. Logging interaction data and visual data coverage might be one option to overcome such an issue. It may become clear from the logging data that an expert has not inspected the data thoroughly enough. For example, how much data an expert investigated can indicate if the expert has examined all data or just parts. If we can infer which parts of the data s/he explored it may demonstrate if just obvious parts, or all parts of the data were inspected. Knowing which data must be explored at least to make valid inferences can again help to know if the right part of data was examined. However, all of this information may also help to investigate if an expert was not able to make a valid decision because s/he was misled by the visualization or could not get all appropriate data necessary.

5 Example of Evaluation using Visual Data Coverage

To show how visual data coverage information can be used when evaluating an interactive visualization, we give an example from a recent study we conducted. In this study, we collected interaction data as well as data that a participant inspected while using a focus-and-context lens. The interaction data was collected by instrumenting the analyzed system. We analyzed the visual analytics system called TrajectoryLenses [13] consisting of a focus-and-context lens which can be used to explore trajectory data displayed on a map (cf. Figure 1). We recorded the data with a frequency of 60 Hz to achieve a sufficient sampling rate of the data and to be able to synchronize it with eye movement data we recorded as well. In our case, we collected an ID, timestamp, and the position of the lens, all in addition to the data that was currently depicted underneath the lens. We achieved this through a hit test with the data items' positions in order to save its ID. We saved all IDs as a string attached to the current interaction. Following this, we calculated how many data items were investigated at each time step and accumulated the data for each unique data item that was examined over time into an overall visual data coverage amount. Additionally, we calculated how often a participant explored each data item by counting how often the ID was recorded.

Since we do not focus on analyzing and visualizing interaction logs, many of the proposed methods are only partially useful. Typically, a timeline is used which depicts interactions either as thumbnails [11, 9] or as color-coded glyphs [7, 12]. Transition matrices or transition charts depicting the transitions between different interaction categories [17, 10, 5] are a different approach, however, we are more interested in how to represent the visual data coverage, rather than individual interactions. Thus, we present two ideas on how to depict this data.

Figure 2 shows the visual data coverage graph for one participant and data visualized at a specific level of detail. On the x-axes we depict the time and on the two y-axis we depict the overall data amount as well as the visual data coverage for each time step indi-

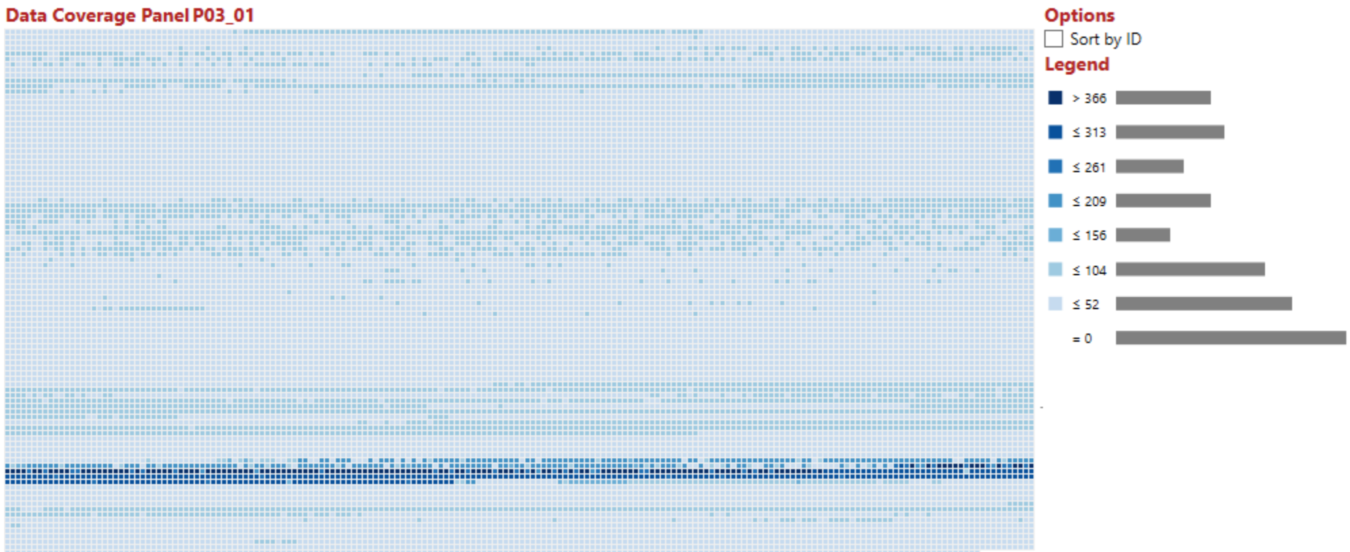

Figure 3: The visual data coverage of one participant where each rectangle represents a data item. The data items are sorted based on the time they were examined for the first time. The color corresponds to the number of times a data item was inspected, the darker the more it was explored. In the legend a bar chart depicts how may data items belong to each range.

vidually. The line in the chart represents the overall amount of data a participant investigated over time. The bar charts indicate the amount of detail data explored interactively over time during the study. We can see that at the beginning of the study, this participant examined a lot of data indicated by the bars on the left of Figure 2. The number of overall data items the participant inspected in detail was 774,141 data items and the maximum number of data items investigated at one time step was 41,070. This provides us with information about how much of the data this participant analyzed.

Additionally, we created a simple chart for showing how often each data item was inspected. Figure 3 shows the data from the same participant. The color of each rectangle represents how often a data item was explored with the lens-based technique using a binning technique. We can see that a few data items have been examined thoroughly, as the dark blue rectangles at the bottom of Figure 3 show. The bar chart on the right next to the ranges indicate how many data points have been investigated in each range. In this case, there is also a large amount of data that has not been inspected in detail (see Figure 3; lowest bar next to = 0). If this part of the data, which a participant did not explore, contains valuable information, a hypothesis could be that the visualization system was not developed appropriately to guide the user. However, if the system was developed trying to guide the user to parts of the data which are of high interest, having a large amount of data not being examined, can be a positive result as well if a user did not inspect unnecessary information. With this chart, we can infer which part of the data and how often each data item was investigated.

6 Conclusion

In this paper we have explored the idea of analyzing the visual data coverage while using an interactive visualization. We have shown how analyzing the amount and parts of the data represented can give valuable insights into how well a visualization technique was designed, and what we can infer from this data regarding traceability and accountability. We indicated which insights can be gained from logging information as well as the visually represented data as an effect of logged interactions. Despite this, there are still many open issues on how to track and analyze this kind of data. We believe that analyzing a user's visual data coverage may help to guide to exploring parts of a visualization more closely in the future. Traceability may help in real time to inspect important and interesting data more appropriately, and from the interaction logs we can get insights with respect to accountability if questionable decisions are made based on visual analysis.

References

[1] F. Beck, T. Blascheck, T. Ertl, and D. Weiskopf. Exploring word-sized graphics for visualizing eye tracking data within transcribed experiment recordings. In *ETVIS*, 2015.

[2] T. Blascheck, M. John, S. Koch, L. Bruder, and T. Ertl. Triangulating user behavior using eye movement, interaction, and think aloud data. In *Proceedings of the Symposium on Eye Tracking Research & Applications*, pages 175–182. ACM, 2016.

[3] T. Blascheck, M. John, K. Kurzhals, S. Koch, and T. Ertl. VA^2: A visual analytics approach for evaluating visual analytics applications. *IEEE Transactions on Visualization and Computer Graphics*, 22(01):61–70, 2016.

[4] M. Brehmer and T. Munzner. A multi-level typology of abstract visualization tasks. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2376–2385, 2013.

[5] E. Brown, A. Ottley, H. Zhao, Q. Lin, R. Souvenir, A. Endert, and R. Chang. Finding waldo: Learning about users from thier interactions. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1663–1672, 2014.

[6] D. Butin, M. Chicote, and D. L. Mtayer. Log design for accountability. In *Security and Privacy Workshops (SPW)*, pages 1–7. IEEE, 2013.

[7] W. Dou, D. H. Jeong, F. Stukes, W. Ribarsky, H. Lipford, and R. Chang. Recovering reasoning processes from user interactions. *IEEE Transactions on Computer Graphics and Applications*, 29(3):52–61, 2009.

[8] S. Eriksén. Designing for accountability. In *Proceedings of the Second Nordic Conference on Human-computer Interaction*, pages 177–186. ACM, 2002.

[9] D. Gotz and M. X. Zhou. Characterizing users' visual analytic activity for insight provenance. *Information Visualization*, 8(1):42–55, 2009.

[10] H. Guo, S. Gomez, C. Ziemkiewicz, and D. Laidlaw. A case study using visualization interaction logs and insight metrics to understand how analysts arrive at insights. *IEEE Transactions on Visualization and Computer Graphics*, 22(01):51–60, 2016.

[11] J. Heer, J. Mackinlay, C. Stolte, and M. Agrawala. Graphical histories for visualization: Supporting analysis, communication, and evalua-

tion. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1189–1196, 2008.

[12] M. R. Jakobsen and K. Hornbæk. Fisheyes in the field: Using method triangulation to study the adoption and use of a source code visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1579–1588. ACM, 2009.

[13] R. Krüger, D. Thom, M. Wörner, H. Bosch, and T. Ertl. Trajectorylenses - a set-based filtering and exploration technique for long-term trajectory data. *Computer Graphics Forum*, 2013(3):451–460, 2013.

[14] H. Lam. A framework of interaction costs in information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1149–1156, 2008.

[15] V. L. Lemieux and T. Dang. *Building Accountability for Decision-Making into Cognitive Systems*, pages 575–586. Springer, 2013.

[16] H. Lipford, F. Stukes, W. Dou, M. Hawkins, and R. Chang. Helping users recall their reasoning process. In *IEEE Symposium on Visual Analytics Science and Technology*, pages 187–194. IEEE, 2010.

[17] Z. Liu and J. Heer. The effect of interactive latency on exploratory visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2122–2131, 2014.

[18] W. Pike, J. Stasko, R. Chang, and T. O'Connell. The science of interaction. *Information Visualization*, 8(4):263–274, 2009.

[19] M. Pohl and F. Scholz. *How to Investigate Interaction with Information Visualisation: An Overview of Methodologies*, pages 17–29. Springer, 2014.

[20] A. C. Robinson and C. Weaver. Re-visualization: Interactive visualization of the process of visual analysis. In *Workshop on Visualization, Analytics & Spatial Decision Support at the GIScience conference*, 2006.

[21] T. R. Wallace and C. van den Heuvel. Truth and accountability in geographic and historical visualizations. *The Cartographic Journal*, 42(2):173–181, 2005.

[22] D. J. Weitzner, H. Abelson, T. Berners-Lee, J. Feigenbaum, J. Hendler, and G. J. Sussman. Information accountability. *Commun. ACM*, 51(6):82–87, 2008.

[23] J. S. Yi, Y. ah Kang, J. Stasko, and J. Jacko. Toward a deeper understanding of the role of interaction in information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1224–1231, 2007.

Results and Challenges in Visualizing Analytic Provenance of Text Analysis Tasks Using Interaction Logs

Rhema Linder* Alyssa M. Peña† Sampath Jayarathna‡ Eric D. Ragan§

Texas A&M University

ABSTRACT

After data analysis, recalling and communicating the steps and rationale followed during the analysis can be difficult. This paper explores the use of interaction logs to generate summaries of an analyst's interest based on interactions with specific data items in a text analysis scenario. Our approach uses data-interaction events as a proxy for user interest in and experience of information. Logging can produce verbose logs that detail all available readable content, so the discussed approach uses topic modeling (LDA) over different time segments to summarize the verbose information and generate visualizations of the history of user interest. Our preliminary results motivate a discussion on potential benefits and challenges of using interaction data to generate provenance visualizations for text analytics.

Index Terms: H.5.2. [Information Interfaces and Presentation (e.g., HCI)—User Interfaces]: Graphical user interfaces (GUI), Interaction styles (e.g., commands, menus, forms, direct manipulation)

1 INTRODUCTION

Complex and open-ended data analysis tasks require exploration of data over extended periods of time. A strong understanding of the data often involves the identification of connections among entities or patterns across data attributes. To assist with the inherently complex analyses, analysts often use a variety of visualization and analytics tools to facilitate the process. However, in addition to understanding the data itself, real-world analysis scenarios also require understanding of the analysis *processes* used in the investigation. For example, after performing an intelligence analysis task, a team of analysts must explain how they arrived at a conclusion about a terrorist attack. This would require citing sources and explaining the evidence supporting their hypotheses.

Due to the importance of understanding the history of the analysis process, many visualization and data analysis tools aim to capture *analytic provenance*, which refers to the history of steps taken and changes made throughout the duration of the analysis [9, 14, 20, 21]. Numerous types of provenance information (e.g., the history of data, visualizations, interactions, insights, or rationale) are considered to be important for visual analysis [21].

Interaction logs can be highly effective for understanding the history of data analysis [7]. However, in order for practical use of interaction data to understand analytic provenance, a clear and efficient means of interpreting that information is needed. In this paper, we describe methods that use interaction data from text-analysis activities as proxy for thought processes. We use interactions as means of approximating importance and implicit interest in content, and we apply topic modeling [4] to summarize the information that has been encountered and interacted with.

We show example visualizations generated from the interactions of a proof-of-concept study where we recorded logs data form an open-ended text analytics task. The results reveal research opportunities for finding interactions that best represent user interest and analyzing history in meaningful time segments. Our preliminary results motivate a discussion on potential benefits, challenges, and research space for future provenance visualizations of text analytics processes.

2 RELATED WORK

The concept of analytic provenance includes a variety of types of information about the history of data analysis. Researchers have previously discussed interpretations, definitions, and potential uses for use of provenance information for the purposes of visualization and data analysis (e.g., [9, 11, 20]). In a recent review of visualization literature, Ragan et al. [21] organized different perspectives and interpretations of *types* of provenance information and the *purposes* for its use.

Many previous projects support provenance visualization, and every project focuses on different types and purposes of provenance. Here, we describe a few of examples. For instance, the previously mentioned *VisTrails* uses a tree view to visually represent the sequence of actions and changes during a scientific workflow [3]. Other systems also adopt tree-style views to represent history (e.g., [13, 8]).

Some provenance systems aim to provide an overview of topic coverage during analysis. For example, Sarvghad and Tory [25] demonstrated the use of radial, treemap, and sequence-flow diagrams to help users understand data coverage from previous analyses. Text analytics systems infer and show relationships among documents and topics. *CzSaw* [16] and Jigsaw [26] are two examples that do so through various types of visualizations. Our approach differs in that we capture and represent the history of information encountered in the analysis process, not a complete assessment of data coverage. Prior research considers the use of additional annotation interaction to help clarify the process with user-provided notes and input (e.g., [9, 13]), but we seek an approach that does not require additional input from users.

Researchers have shown how processing and visualizing interaction logs can aid both researchers in inferring strategies and analysts in recalling insight. For example, Gotz and Zhou [11] described how the use of common actions could be used to infer the history of meaningful behavior and rationale that lead to insights during analysis, and Dou et al. [7] studied the feasibility and effectiveness of interpreting user interaction logs to understand an analyst's rationale. Lipford et al. [17] found evidence that the interaction visualization can improve recall of certain insights and rationale from the analysis. Also looking at a type of visualization created through interaction alone, a study by Ragan, Goodall, and Tung [23] found that the visual state of the workspace at the end of an analysis was enough to significantly improve memory of the process. Taking a different approach, Brown et al. [5] demonstrated how analysis

*e-mail: rhema@tamu.edu
†e-mail: mupena17@tamu.edu
‡e-mail: sampath@tamu.edu
§e-mail: eragan@tamu.edu

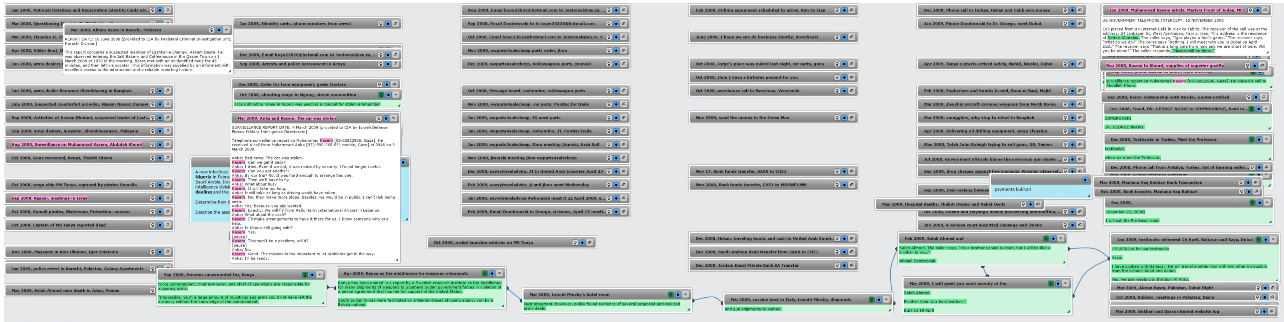

Figure 1: A screenshot of the text exploration tool that shows search results (pink highlights) along with a string of user-highlighted text (dark green highlights), reduced-to-highlight boxes (light green), and documents connected with linking lines. The participant arranged a chain of documents along the bottom and right of the workspace.

Figure 2: A close up of top right of the larger screenshot of the document explorer tool used for the data collection study.

of interaction data was able to determine information about users' strategies as well learn about the users themselves. Like our work, these prior projects took advantage of normal interactions without requiring additional input.

Prior projects have shown strong correlations among user interest and implicit indicators based on interaction [1]. Reading, organizing, and spending time on a document correlated to later ratings of relevance. By weighting the information, Bae et al. created a system for aiding text analysis tasks by automatically annotating multiple user interests [2].

Our approach uses interaction data to summarize analysis themes over time, focusing on textual analysis. For textual data, topic models have drawn interest and been utilized in a wide range of research including the humanities and social sciences [19], large scale social media studies [15], and analyzing political speeches [10]. Topic modeling is an appealing method for text analysis because it organizes words into coherent themes. We investigate topic modeling (Latent Dirichlet Analysis [4]) as a means to make sense of analyst encountered information over time.

3 Method

3.1 Collecting Provenance Data

To design and test our method for generating provenance summaries, we conducted a study where participants performed a text analysis task (while providing think-aloud verbal updates), and we recorded interaction logs. For the analysis scenario, we selected a task with sufficient complexity and scope to allow the exploration of various topics and hypotheses. To this end, we chose a text analysis scenario from the IEEE VAST 2010 Challenge Mini Challenge #1 [12].

To analyze the data, participants used a document exploration application (as in [22] and [23]) where text documents could be viewed, searched, and spatially manipulated to support organization. In this application, documents were placed in a 2D space and could be manipulated, similar to prior tools [26, 27]. The exploration tool (Figures 2 and 1) logged various actions performed by each user (see Table 1).

Interactions Captured

Search	Search the data set for a word or phrase
Reduce to Highlight	Reduce the visible text in a document to only the highlighted content
Highlight Text	Highlight text in a document
Connect Document	Create a new connection line linking two document or note windows
Collapse Document	Minimize the document window to only show the its title bar
Open Document	Expand a collapsed document window to show its full text
Move Document	Drag a document window to a new location
Mouse Enter	The mouse position moves over a document window

Table 1: Types of interactions logged by the document exploration tool.

We recruited six participants for the study, five males and one female. Ages ranged 18 to 30 years old, and all had low to moderate experience in data analysis or visualization. As the data was about weapons dealing, participants were asked to explore the documents to report on the connections and plans involving illegal weapons trade. We recorded log data for their actions and recorded video.

3.2 Automatic Provenance Summary Generation

We describe our preliminary methods for generating provenance visualizations automatically using interaction logs. As a case study, we used information and logs from the open-ended text analytics task described in the previous section.

Our method is summarized in five steps: (1) Capture interaction events during a period of analysis. (2) Generate a sequence of text by using the interaction events to establish the encountered text over time. (3) Process the text data using standard information retrieval techniques [18]. (4) Segment the corpus of texts by periods of time and generate topic models per segment. (5) Visualize topics over time to facilitate easier interpretation and pattern recognition.

The first step of our summary generation process is to *Capture Interaction Events* from user interaction with the analysis tool to

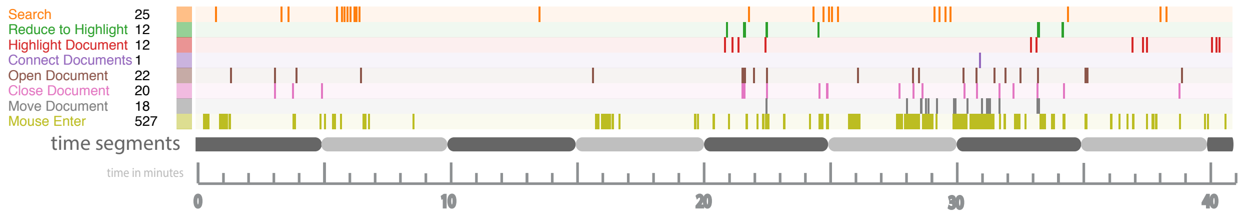

Figure 3: This visualization depicts the interaction history from a participants' document explorer study session. Analysis time is stretched across the horizontal axis. Each thin colored line represents a single action at a given time. The number on the left of each row shows the total number of actions. Beneath the timeline, we show how a five-minute breakdown produced the segments used in the provenance visualization.

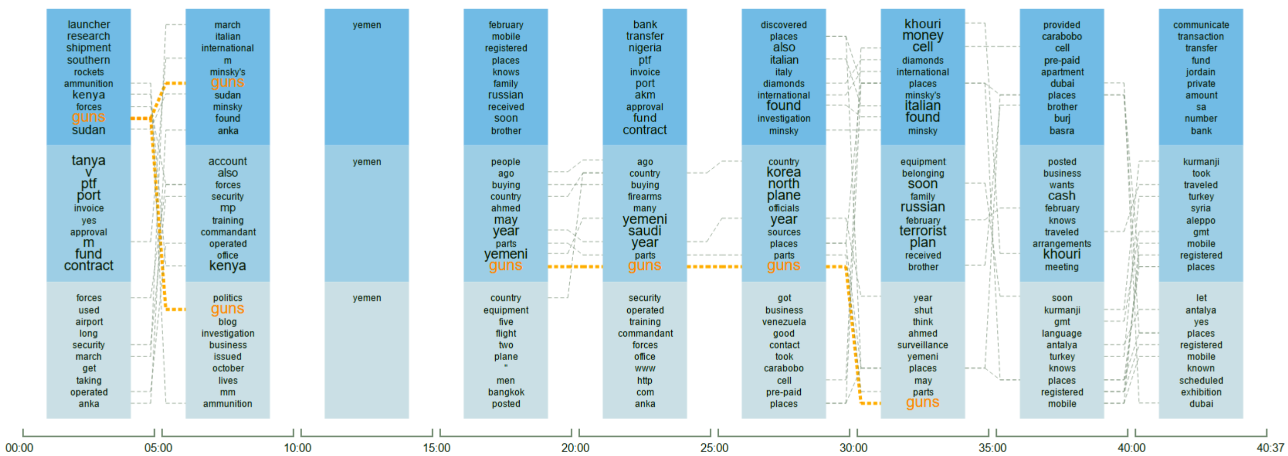

Figure 4: A parallel word cloud design shows the topic segments with the terms in each topic. In this figure, each blue column shows three topics in a time segment. The orange words have been moused-over in order to show all occurrences of the term across all segments. Note that the third column is missing interactions except a single search. This is consistent with our observations of the participant during the study.

create a history of their actions. This history in and of itself explains how the interface was used, but it provides nothing about the content and context of user actions. Ideally, it would be possible to capture and represent the analytic process without requiring additional input or description from analysts.

The second step is to *Generate a Sequence of Text*. Since most interactions involve interacting with a specific document, it was possible we could associate each interaction with a text. Thus, each interaction even can provide a sequence of text documents that the user considered during the analysis. We used these sequences of text as the basis for topic modeling to infer the main stages of analysis over time.

The third step is to *Process the Text* after capturing the interaction history to make it more suitable for topic modeling. We use standard information retrieval methods [18] for tokenization and removing stop words. Additionally, we create another list of stop words of common, less-useful terms from entries (e.g., *report*).

Segmentation and Topic Modeling occurs the data is processed. In the end, our goal is to create summaries of the stages and themes of analysis over time. Each record from interaction logs includes associated text, a type, and timestamp data. Simply printing these records would be too verbose for practical interpretation.

To simplify these records, we first use a *segmenter* to break the history down into discrete time segments the summarize with topic modeling. In the results presented here, we break down the 35–40 minute analysis session data into five-minute segments. Once segmented, we use a topic modeler (LDA) to generate a set of topics for each time period. Per the case study, we show three topics and found 15 iterations of LDA to be sufficient. The final results are serialized and saved to provide data for the last step: visualization.

The final step is *Visualization* (see Figure 4). We used *parallel word clouds* [6, 10] as the base representation to visualize the topics over time. In our visualization, the topics are shown in lists of words embedded in blue columns. Each column represents a time segment of user interaction history. Columns show their beginning and end times and their topic model summaries. Within each column, the three topics are sorted based on coherence, as calculated in the Gensim framework [24] and are distinguished with different shades of blue. For a single topic within a column, we show a vertical list of terms ordered by the probabilities of in the model. To highlight important words within topics, individual terms are scaled based on TF–IDF (term frequencyinverse document frequency) scores to decrease the importance of common words words and help representative words remain prominent.

In addition, the design includes linking lines that connect any word to the same word in adjacent time segments. When the user brushes or hovers the cursor over a word, it changes color (orange) and increases for all instances of the word. Linking lines will also be highlighted in orange for increased salience. A "brushed" word slowly transitions back to its default style when the cursor is moved off of the word, causing the word's highlight to persist briefly. This helps a viewer reveal patterns of clusters of repeated words.

4 Discussion and Conclusion

4.1 Preliminary Results

Our preliminary results are promising. The automatically generated snapshots of analyst interest over time, by our observations, capture meaningful topics. The visualization presents themes based on interactions where users open, looked at, and manipulated content. These topics are connected together to show the flow of topics over time, and for our case study and test data, the effectiveness is clear. While some segment's topic summaries make more sense than other, they are on the whole meaningful.

Our observations are that different participants used different strategies: breadth-first, depth-first, and cyclical processes where a particular theme is repeatedly revisited. We found that exploring with brushing and linking can reveal participants general strategies. For instance, one participant said, "*I feel like I'm doing a depth first search*", and the summaries showed this with many connections from one time segment to the next, rather than sudden changes in topics and interest.

4.2 Implicit Interest from Different Interaction Types

This research raises the importance of discovering the most salient interaction types during analysis. For example, looking at a timeline of interactions (Figure 3), note that some types of interactions are performed more than others. Including and excluding data based on the type of interaction may impact the effectiveness of summaries using the discussed approach. For example, *Mouse Over* events occur many times, and it is likely that participants who used *Mouse Over* were interested in the content of the documents they interacted with. However, an analyst may accidentally *Mouse Over* a document they have no interest in when moving their mouse across the screen. At the same time, we observed that many *Mouse Over* events are meaningful. Participants would move the mouse back and forth between documents, weighing their information and planning what to look for next. On the other hand, *Open* events occur far less frequently, but might be more meaningful than *Mouse Over* events. Performing the *Open Document* action represented committed actions—usually occurring only when a participant thought they were likely to find important material. We find many interaction types could be thought of as having little to much meaning and occur at different frequencies. It is likely that studies and observations are needed for determining which events are meaningful and frequent enough.

4.3 Temporal Segmentation Schemes

Another research opportunity is to create methods for segmenting interaction events over time. Our example (as seen in Figures 3 and 4) shows the topic history with five-minute segments. While this is straightforward to implement and understand, it comes with significant drawbacks. In some cases, adjacent columns in the visualization were too similar, adding limited additional information. For example, one participant stopped interacting with the system for about six minutes in the middle of the analysis, opting to read and plan but not move the mouse or click on documents. Using static five-minute segments, this created a gap where only a single search term was captured (see Figure 4, third column from the left).

Better segmentation schemes are needed. After looking at the timelines from different participants, we noticed that searches usually occur in bursts. A single burst of search interaction usually meant that they tried a few queries until the results seemed promising. With these observations in mind, we see an opportunity to create different segmentation schemes that take advantage of a combination of (1) the degree of change in content and (2) implicit time boundaries based on interaction types (e.g., *Open* or *Search*).

4.4 Text-Associated Interactions as a Proxy to Thought Processes

This work has begun to explore automated methods for summarizing encountered information during text analysis. While a tremendous amount of text can be generated from interaction logs, reviewing all logs would be impractical. Instead, we believe a summarization approach such as ours can serve as a proxy for understanding analytic provenance and analysts' interests.

While an ideal provenance summary of an analysis would include the internal thought of analysts, this is technically impossible to capture completely without disrupting the analysis. However, associating interaction events with text can provide a view of repeatedly encountered information, which influences the thoughts of analysts. Also, for the purpose of task resumption, we expect that showing analysts provenance visualizations will help them remember their rationals and insights. For researchers looking to understanding others' pathways to insight, we expect that these visualizations will provide views that help them see strategies, such as bread-first or depth-first approaches.

4.5 Conclusion

We investigated how topic modeling can be used to automatically generate provenance visualizations from interaction logs. To explore the approach, we collected data from participants who performed a text analytics task. We recorded their interactions and created visual summaries for preliminary evaluation of our method. While far from perfect, the method created surprisingly representative text summaries. We found the visualizations were useful for understanding user interest over time, recurring interest of topics, and aspects of analyst's strategies. The results could be improved by addressing the research opportunities of (1) finding which interaction events are most representative of user interest and (2) developing interaction timeline segmentation schemes. Future work should evaluate the efficacy of using this and similar methods for understanding thought processes in text analysis tasks.

Acknowledgements

This material is based on work supported by NSF 1565725.

References

[1] R. Badi, S. Bae, J. M. Moore, K. Meintanis, A. Zacchi, H. Hsieh, F. Shipman, and C. C. Marshall. Recognizing user interest and document value from reading and organizing activities in document triage. In *Proceedings of the 11th International Conference on Intelligent User Interfaces*, IUI '06, pages 218–225, New York, NY, USA, 2006. ACM. 2

[2] S. Bae, D. Kim, K. Meintanis, J. M. Moore, A. Zacchi, F. Shipman, H. Hsieh, and C. C. Marshall. Supporting document triage via annotation-based multi-application visualizations. In *Proceedings of the 10th Annual Joint Conference on Digital Libraries*, JCDL '10, pages 177–186, New York, NY, USA, 2010. ACM. 2

[3] L. Bavoil, S. P. Callahan, P. J. Crossno, J. Freire, C. E. Scheidegger, C. T. Silva, and H. T. Vo. Vistrails: Enabling interactive multiple-view visualizations. In *IEEE Conference on Visualization*, pages 135–142, 2005. 2

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003. 1, 2

[5] E. T. Brown, A. Ottley, H. Zhao, Q. Lin, R. Souvenir, A. Endert, and R. Chang. Finding waldo: Learning about users from their interactions. *Visualization and Computer Graphics, IEEE Transactions on*, 20(12):1663–1672, 2014. 2

[6] C. Collins, F. B. Viegas, and M. Wattenberg. Parallel tag clouds to explore and analyze faceted text corpora. In *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on*, pages 91–98. IEEE, 2009. 3.2

[7] W. Dou, D. H. Jeong, F. Stukes, W. Ribarsky, H. R. Lipford, and R. Chang. Recovering reasoning processes from user interactions. *IEEE Computer Graphics and Applications*, (3):52–61, 2009. 1, 2

[8] C. Dunne, N. Henry Riche, B. Lee, R. Metoyer, and G. Robertson. Graphtrail: Analyzing large multivariate, heterogeneous networks while supporting exploration history. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1663–1672. ACM, 2012. 2

[9] J. Freire, D. Koop, E. Santos, and C. T. Silva. Provenance for computational tasks: A survey. *Computing in Science & Engineering*, 10(3):11–21, 2008. 1, 2, 2

[10] S. Gad, W. Javed, S. Ghani, N. Elmqvist, T. Ewing, K. N. Hampton, and N. Ramakrishnan. Themedelta: dynamic segmentations over temporal topic models. *Visualization and Computer Graphics, IEEE Transactions on*, 21(5):672–685, 2015. 2, 3.2

[11] D. Gotz and M. X. Zhou. Characterizing users' visual analytic activity for insight provenance. *Information Visualization*, 8(1):42–55, 2009. 2, 2

[12] G. Grinstein, S. Konecni, J. Scholtz, M. Whiting, and C. Plaisant. VAST 2010 challenge: arms dealings and pandemics. In *IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 263–264. IEEE, 2010. 3.1

[13] D. P. Groth and K. Streefkerk. Provenance and annotation for visual exploration systems. *Visualization and Computer Graphics, IEEE Transactions on*, 12(6):1500–1510, 2006. 2

[14] Z. Hensley, J. Sanyal, and J. New. Provenance in sensor data management. *Communications of the ACM*, 57(2):55–62, 2014. 1

[15] L. Hong and B. D. Davison. Empirical study of topic modeling in twitter. In *Proceedings of the first workshop on social media analytics*, pages 80–88. ACM, 2010. 2

[16] N. Kadivar, V. Chen, D. Dunsmuir, E. Lee, C. Qian, J. Dill, C. Shaw, and R. Woodbury. Capturing and supporting the analysis process. In *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on*, pages 131–138. IEEE, 2009. 2

[17] H. R. Lipford, F. Stukes, W. Dou, M. E. Hawkins, and R. Chang. Helping users recall their reasoning process. In *Visual Analytics Science and Technology (VAST), 2010 IEEE Symposium on*, pages 187–194. IEEE, 2010. 2

[18] C. D. Manning, P. Raghavan, H. Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge University Press, 2008. 3.2, 3.2

[19] J. W. Mohr and P. Bogdanov. Topic models: What they are and why they matter. *Poetics*, 41(6):545–569, 2013. 2

[20] C. North, R. Chang, A. Endert, W. Dou, R. May, B. Pike, and G. Fink. Analytic provenance: process+ interaction+ insight. In *ACM CHI Extended Abstracts on Human Factors in Computing Systems*, pages 33–36, 2011. 1, 2

[21] E. D. Ragan, A. Endert, J. Sanyal, and J. Chen. Characterizing provenance in visualization and data analysis: An organizational framework of provenance types and purposes. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):31–40, 2016. 1, 2

[22] E. D. Ragan and J. R. Goodall. Evaluation methodology for comparing memory and communication of analytic processes in visual analytics. In *Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization*, pages 27–34. ACM, 2014. 3.1

[23] E. D. Ragan, J. R. Goodall, and A. Tung. Evaluating how level of detail of visual history affects process memory. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 2711–2720. ACM, 2015. 2, 3.1

[24] R. Řehůřek and P. Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. 3.2

[25] A. Sarvghad and M. Tory. Exploiting analysis history to support collaborative data analysis. In *Proceedings of the 41st Graphics Interface Conference*, pages 123–130. Canadian Information Processing Society, 2015. 2

[26] J. Stasko, C. Görg, and Z. Liu. Jigsaw: supporting investigative analysis through interactive visualization. *Information visualization*, 7(2):118–132, 2008. 2, 3.1

[27] J. A. Wise, J. J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow. Visualizing the non-visual: spatial analysis and interaction with information from text documents. In *Information Visualization, 1995. Proceedings.*, pages 51–58. IEEE, 1995. 3.1

SpatialVis: Visualization of Spatial Gesture Interaction Logs

Erik Paluka and Christopher Collins

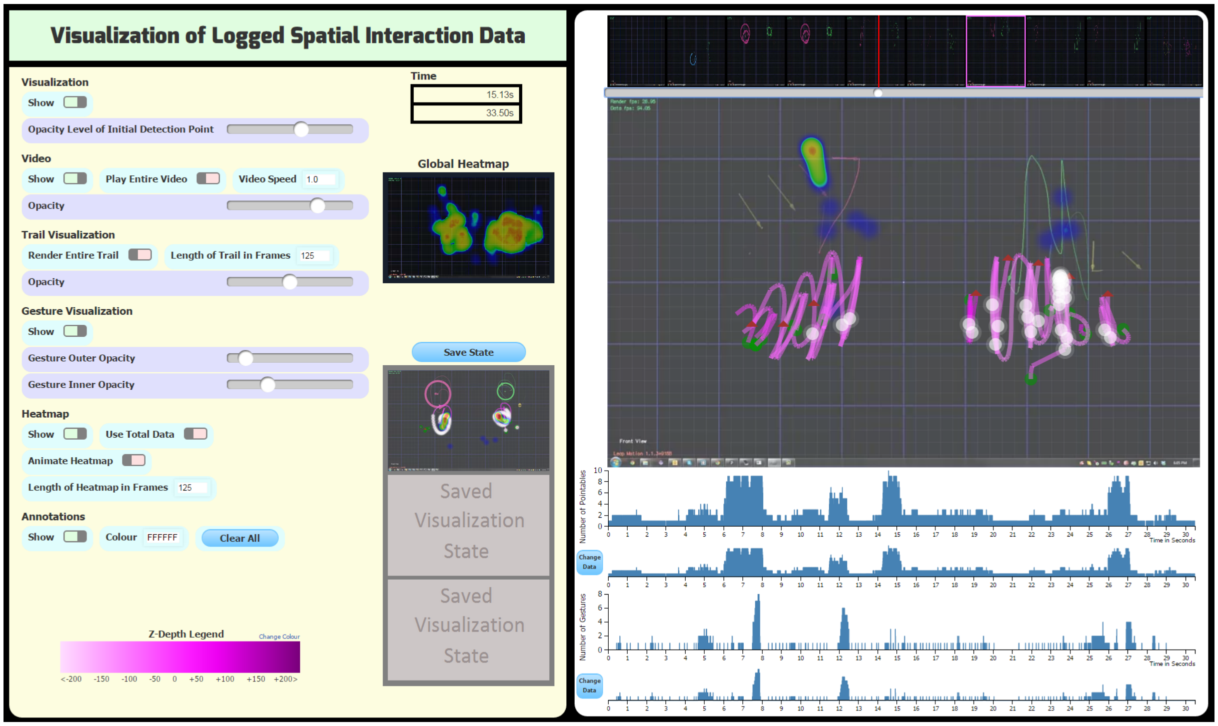

Figure 1: SpatialVis: Web application for visualizing logged spatial interaction data.

Abstract

This paper presents SpatialVis, a system for logging spatial gesture interactions and a visualization interface to analyze those logs. SpatialVis overlays a gesture log visualization atop screen recordings of interaction sessions to allow for replay of experimental trials. We discuss the challenges of logging spatial interactions and recommendations for best practices.

Keywords: interaction design, visualization, spatial gestures

1 Introduction

Spatial interaction devices, which enable the control of traditional pointers as well as the performance of single and multi-hand mid-air gestures to interact with computers, are becoming commonplace. Precise spatial gesture hardware such as the Leap Motion device allow for the design of gestures for interacting with visualizations, such as selection, zoom, filter, and other basic information visualization interactions [5]. They offer new capabilities to create visualization systems for use in environments where touch screens and mouse interaction are inappropriate, for example in sterile environments.

It is also possible to adapt existing systems for use with spatial gestures. An easy way to integrate spatial interaction on a desktop computer is to use spatial gestures to control the pointer. Although, this causes challenges since standard desktop graphical user interfaces (GUI) are designed for precise input devices such as the mouse. A typical virtual object's small display and interaction spaces reflect this, which can lead to problems selecting items as well as other fundamental tasks. To mitigate this problem, a designer can integrate concepts from techniques that facilitate target acquisition (e.g. Bubble Cursor [2]) into the mid-air selection gesture.

When an information space is larger than the display, it is typical for interfaces to only support interacting with content that is rendered within its viewport. To support interacting with off-screen content, our previous work [4] explored the design and evaluation of several spatial off-screen exploration techniques that make use of the interaction space around the display. These include *Paper Distortion*, *Dynamic Distortion*, *Dynamic Peephole Inset*, *Spatial Panning*, and *Point2Pan* (see Figure 2).

When implementing new interaction techniques for spatial interaction, as we did with Off Screen Desktop, it is beneficial to test the gestures with people who are not accustomed to them. To gather data for analysis, one can video record and/or observe people as they use the gesture in conjunction with a GUI, as well as administer post-questionnaires and interviews. Log data from the spatial gesture sensors can also be gathered. The problem with this is that, other than the video and observational data, logging techniques produce high frequency 3D data logs in text form. Long log files do not harness the full power of the human visual system; therefore making the analysis difficult.

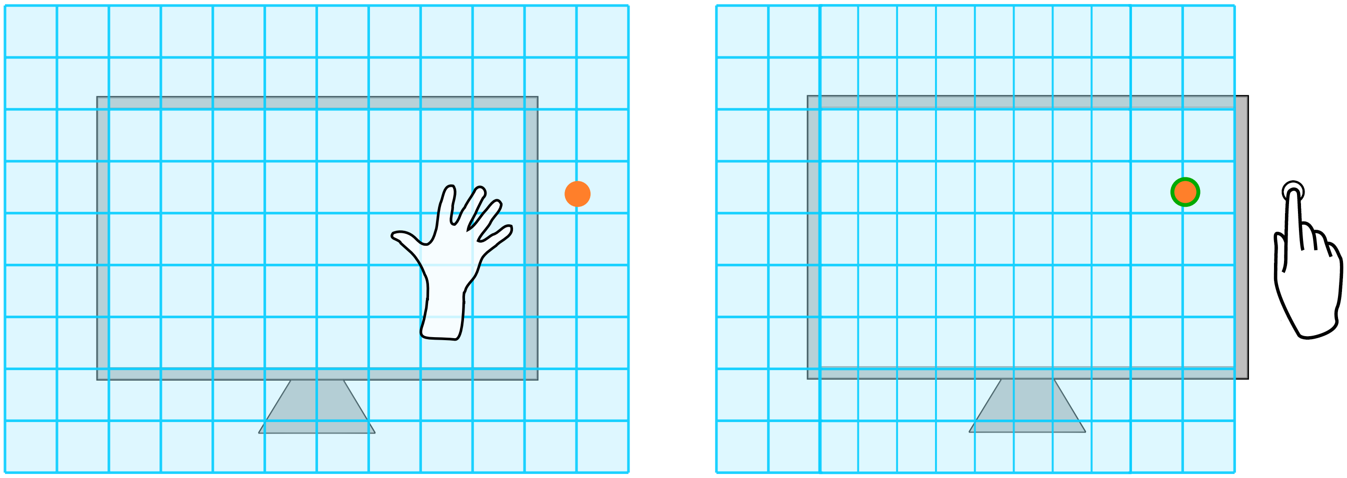

Figure 2: Off Screen Desktop geometrically transforms the visual presentation of the information space without affecting the interaction space to bring off-screen content into the viewport. Since the interaction space remains unchanged, users are able to directly manipulate off-screen content that has been brought onto the display by performing a spatial selection (e.g. tap or grab) in its original off-screen location beside the display. In the above figure, the Dynamic Distortion technique is being employed to transform the on-screen information to create room for off-screen content.

To mitigate these problems and help designers build better spatial user interfaces, as well as help us study our off-screen interaction system, we developed a web-based application that visualizes logged spatial interaction data. By first uploading a log file and an associated video screen capture of the display, an investigator can employ its features to analyze the 3D interactions and their effects on the graphical user interface. Our system is not meant to replace any other method, but to fit within the investigative process to gain further insight into the related phenomena.

We implemented the application using JavaScript, HTML, and the D3 visualization toolkit [1]. Our prototype supports the spatial interaction data types provided by the Leap Motion controller (see **??**) and assumes that the controller's interaction space is in front and centred with the display. We also created a modified version of the application to be able to handle interaction spaces at the sides of the display. We did this to visualize data gathered from the study of our off-screen interaction system in order to gain further insight into participant usage patterns.

2 SpatialVis

To use our system to analyze a spatial GUI, the application being tested must automatically log all associated spatial interaction data. A video of this interface must also be recorded, using screen capture software, with a video length equal to the period of time spent interacting with the interface. This allows log data to be mapped to the user interface events that occur in the video. When complete, the designer or investigator can upload the video and log files to our web application, which will then display the video on the right side of the interface with a heatmap and a path visualization overlaid on top of it. The system also includes a timeline situated above the video, graphs of the interaction data underneath the video, and a global heatmap and different controls to the left (see Figure 1). When the video is played, the overlaid visualizations display spatial interaction data that is temporally related to the video's current frame.

Going back to the spatial target acquisition example from above, the analyst can use our system in conjunction with observational notes or a video recording of the person performing the gestures. For example, this would allow one to view what data the motion sensing hardware is producing and if that matches up with the gesture that the person is trying to perform. If this analysis was done with logged data that was not visualized, the investigator would have to look through hundreds or thousands of lines of text and would be very tedious.

2.1 Video Timeline

The timeline is created by dividing the video into ten equally sized sections and using images from the beginning of each video segment to represent each section (see F in Figure 3). When a user hovers over one of the sections, its border changes colour (see purple box at the top of Figure 1) and they are then able to select the section to seek the video to the start of it. If a section is selected, then the heatmap will update to show data only from this section's time range. If the video is then played, it will stop playing at the end of the selected section unless a widget on the left side of the interface is toggled. The timeline also contains a slider widget for seeking, while hovering over its handle will cause the play, pause and restart video controls to appear beside it.

2.2 Spatial Interaction Graphs

The graphs below the video show spatial interaction information over the length of the video. Their size match the width of the timeline to allow a person to match the current time of the video (slider's handle and vertical line above) with the graph data, as well as to provide awareness of video's current time value. The graphs are also enabled with the brushing and linking [3] techniques. Therefore, if one discovers a time range with an interesting data pattern, the visual complexity of the interface can be reduced to allow the analyst to concentrate on this subset of data. This is accomplished by selecting the data or time range of interest, which will then cause the rest of the data to be filtered out (see B in Figure 3). This brushing will then be reflected in the other graph, as well as in the heatmap and path visualization that are overlaid on top of the video. The video is also seeked to the beginning of the time range associated with the brushed data and if played, will stop at the end of this range. If the user is interested in analyzing other spatial interaction data types, they can change the data visualized in each graph from a range of options including pointables (tools and fingers), tools, fingers, hands, all gestures, as well as each individual gesture type.

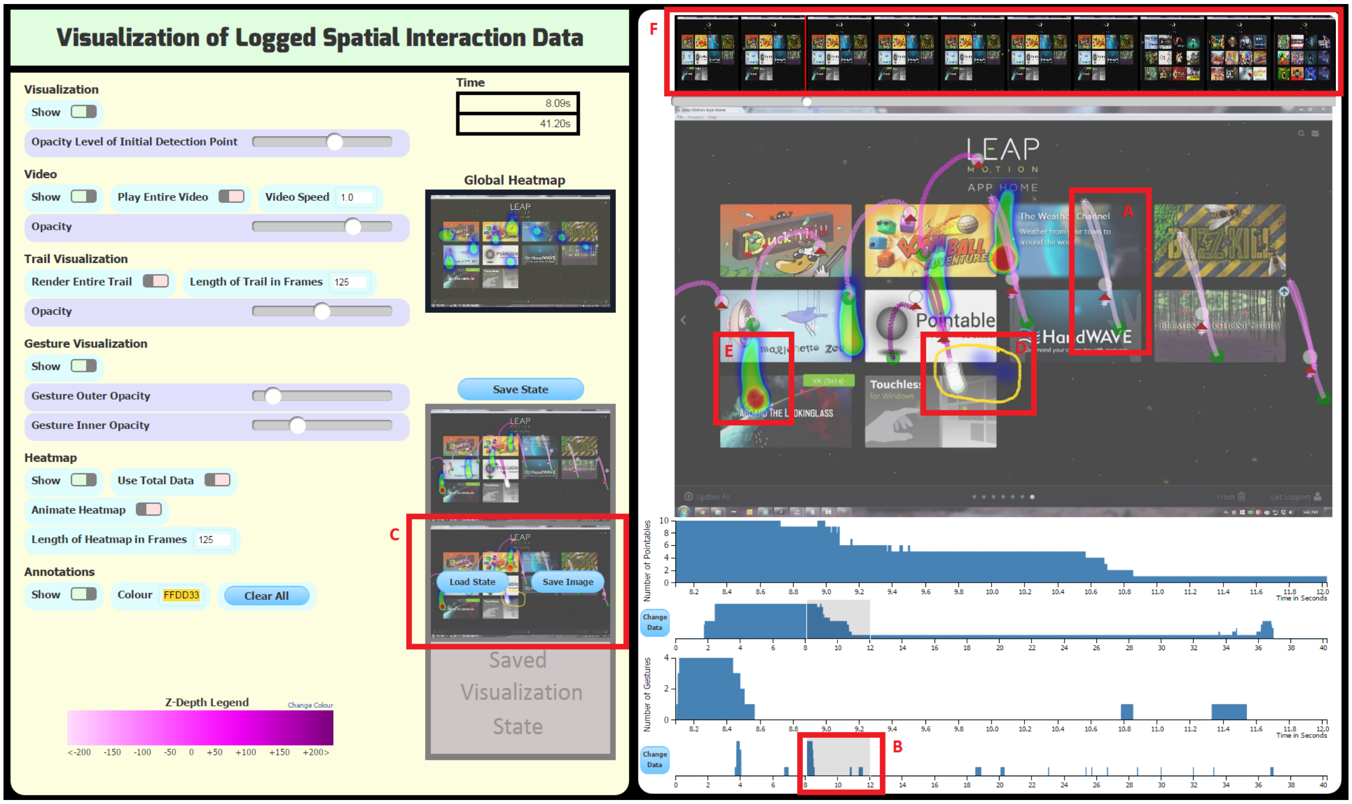

Figure 3: SpatialVis being used by an analyst. (A) Visualizing portion of spatial interaction data.(B) Brushing to show data only associated with 8 to 12 seconds into the video. (C) Saved visualization state. (D) User annotation. (E) Heatmap of data associated with 8 to 12 seconds into the video. (F) Video timeline.

2.3 Video Visualizations

We employed different visualization techniques to visualize each spatial interaction's location with respect to the user interface contained in the video. This was accomplished by overlaying them on top of the video using an orthographic projection mapping. We used a static heatmap to visualize the frequency of gestures that were performed at different locations. Data is selected to be visualized in the heatmap if its associated frame is within a non-sliding window. When the user first loads the required data into the application, the window is the size of the entire video; therefore all of the gesture data is initially visualized. If the video is played or seeked, then the window's starting frame is set to the seeked location or the beginning of the video segment being played. The window's ending frame is then calculated by adding a user-changeable value, contained in a widget, to the starting frame. Although, if the timeline sections or graphs are used to seek the video instead of the time slider, then the window's ending frame is set to either the timeline section's last frame or the last frame associated with the selected graph data. The interface also contains some other widgets that allow the user to set the window's ending frame to always be the last frame in the video, as well as to animate the heatmap over time using the data contained in its window.

We also visualized the path of each pointable (finger or tool) using a semi-transparent path. The pointable's Z-depth is encoded using colour with either a monochromatic or dichromatic divergent colour scheme. The path contains green semi-transparent circles to visualize the location of each pointable when it was first detected by the spatial interaction sensor. To visualize a pointer's spatial location in the current frame, a red semi-transparent triangle is attached to the end of the path. We also affixed a white semi-transparent circle to the path for visualizing the spatial location of different gestures, such as swipe, screen tap and key tap. For example, the white circles in Figure 1 show the location of discrete screen tap gestures. The visualization is dynamic since it displays data from a temporal sliding window that starts in the past and ends at the video's current frame. As the video plays, new data is visualized when it enters the sliding window and old data that is no longer inside the sliding window is removed. This aids the analysis process since interaction data would quickly disappear if only the current frame's data was visualized. The path visualization's sliding window is automatically set to a low value, but the user has the ability to change it, such as when one wants to visualize entire paths of all pointers.

In addition to the aforementioned visualizations, our system allows the analyst to create their own visual markings by providing video annotation abilities (see D in Figure 3), which can then be used to label the location of interesting events, for example.

2.4 Global Heatmap, Controls & Visualization States

Context is important for analysis, therefore we included a global context view of the gesture data with the use of a miniaturized image of the video that is overlaid with a heatmap that visualizes gesture data from the entire video. To further facilitate the analysis process, we also provide the ability to save and load different visualization states (see C in Figure 3). The video frame with the overlaid visualizations and user annotations that are associated with a saved visualization state can then be downloaded as an image for sharing and offline analysis. The interface also contains widgets on the left side to allow the investigator to show or hide the heatmap, path visualization, user annotations or the video itself. Opacity lev-

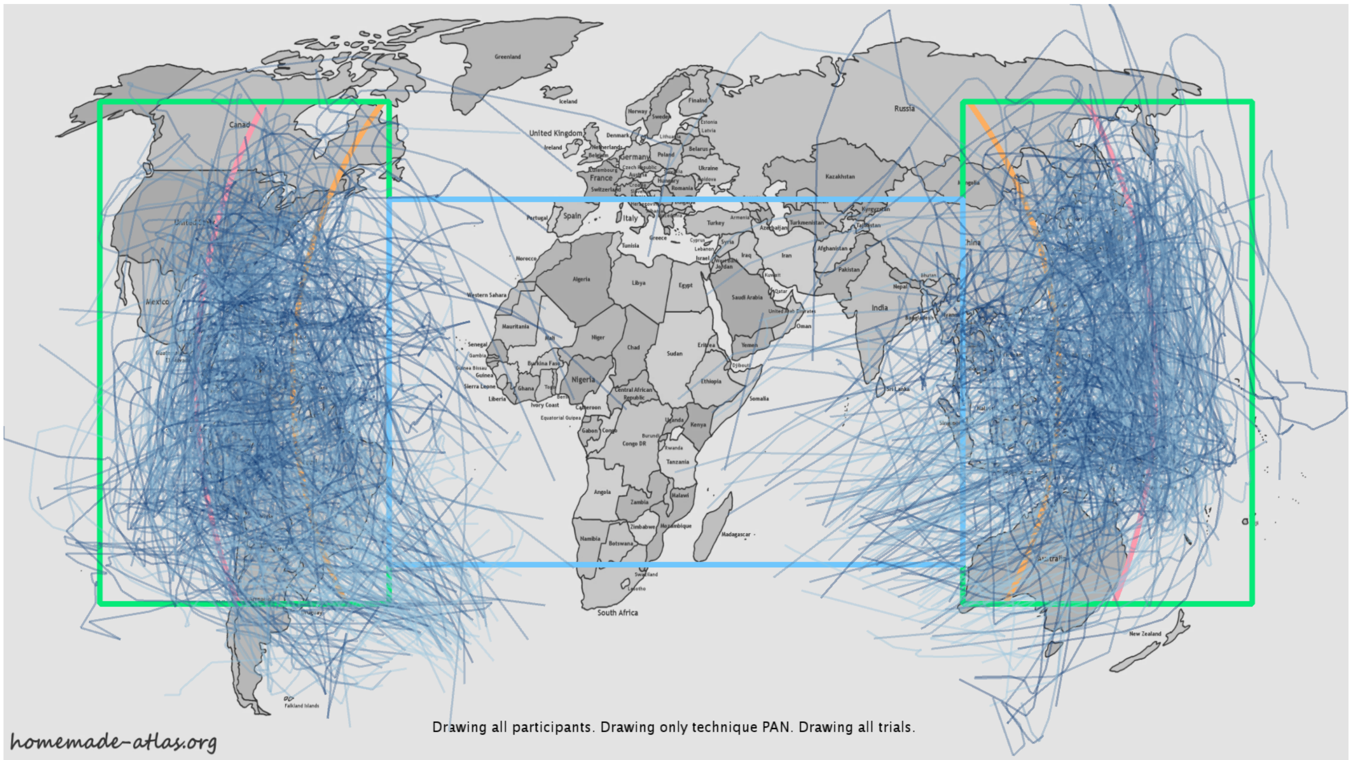

Figure 4: Path visualization showing the movement of all participants hands while searching the off-screen space with the Direct Spatial Panning technique.

els associated with the video and each component of the path visualization can be modified as well. The interface also contains widgets to allow the video's playback speed and the colour of the annotation brush to be changed.

3 Challenges

During the development and use of SpatialVis to support our research on the Off Screen Desktop project, we encountered several challenges which point to future research opportunities.

Color Scheme First of all, the color scheme of the overlay was difficult to design in a way that provided depth information as well as being discernible from the underlying application screen capture. Our solution to this is to allow the screen capture video to be toggled, but this is not ideal. In addition, in many cases the specific finger used in a gesture is important, and our visualization does not reflect this information. This could be encoded with color in the visualization, but additional colors would exacerbate the challenge of the visualization palette interacting with the screen capture.

2D projection The use of a 2D visualization (with depth of interaction encoded as color) to visualize a 3D interaction space results in some difficulty interpreting the resulting views. In particular, a lot of over-plotting can occur for even brief interaction logs. Our workaround for this is to limit the length of the spatial gesture interaction trails using the sliding window, but another alternative to explore would be the provision of a 3D reconstruction view of the spatial interactions.

Coupling Logging and Application

Interaction logging can be general (focused on the input device) or specific to an application (focused on high-level application events). In the case of a general application like SpatialVis, the interactions are logged by a process monitoring the input stream, and also recording the screen. The resulting log files from a variety of applications can be loaded into the same log analysis system. The advantage of this approach is that the logging and analysis system is generic and reusable. The disadvantage is that the logging is separated from the interaction events generated by the software which is being tested. For example, if a gesture is used to generate a selection event on a visualization application, or a specific data element receives a lot of interaction attention, SpatialVis has no knowledge of this. The analyst would have to derive this insight. If the spatial and interaction logging were both embedded in the application, potentially more useful details about how gestures trigger interface events would be available, at the cost of losing generality.

A potential compromise to this problem would be to create a standard logging format which application developers could use to output interaction event logs (including low level events such as button press and high level events such as filtering a view). These logs would be the same whether touch, mouse, or a spatial interface was used. Then these could be interleaved with the SpatialVis logs and screen recordings to create a unified analysis system without requiring integration of spatial interaction logging into the test application itself.

Scalability SpatialVis logs were useful in analyzing the behavior of individual experimental participants in detail. However, the visualization is not scalable to multiple participants as it quickly becomes too cluttered. Also, unless the screen capture is static, it does not make sense to overlay multiple screen captures. To see the trends for multiple participants in a repeated trial experiment, we had to create a simplified visualization we called PathVis (see Figure 4) which shows the spatial position of the pointer in the information space over time, overlaid for multiple participants. The spatial depth information as well as multiple fingers were removed from this view.

4 Conclusion

Spatial interaction is an emerging modality for many types of applications, including information visualization. We have presented an initial visualization application, available online at (URL to come for final version), for analyzing experimental trials of interactions with visualization applications using spatial gestures.

References

[1] M. Bostock, V. Ogievetsky, and J. Heer. D3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, Dec. 2011.

[2] T. Grossman and R. Balakrishnan. The Bubble Cursor: Enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '05, pages 281–290, New York, NY, USA, 2005. ACM.

[3] D. A. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, Jan. 2002.

[4] E. Paluka. Spatial peripheral interaction techniques for viewing and manipulating off-screen digital content. Master's thesis, University of Ontario Institute of Technology, Oshawa, Ontario, Canada, 2015.

[5] J. S. Yi, Y. a. Kang, J. Stasko, and J. Jacko. Toward a deeper understanding of the role of interaction in information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1224–1231, Nov. 2007.