Methods for Evaluating Visualizations Using Practical Models of Insight, Interaction, and Gaze

by
Steven Richard Gomez
B. A., Dartmouth College, 2007
Sc. M., Brown University, 2011

A dissertation submitted in partial fulfillment of the
requirements for the Degree of Doctor of Philosophy
in the Department of Computer Science at Brown University

Providence, Rhode Island
May 2016

This dissertation by Steven Richard Gomez is accepted in its present form by the Department of Computer Science as satisfying the dissertation requirement for the degree of Doctor of Philosophy.

Date _____ _____

David H. Laidlaw, Director

Recommended to the Graduate Council

Date _____ _____

Jeff Huang, Reader
Brown University

Date _____ _____

Remco Chang, Reader
Tufts University

Approved by the Graduate Council

Date _____ _____

Peter M. Weber
Dean of the Graduate School

Vita

Steven R. Gomez was born and raised in central Vermont. He received the Bachelor of Arts degree *magna cum laude* in 2007 from Dartmouth College in Hanover, New Hampshire. After, he worked as a software engineer at M2S, Inc., in West Lebanon, New Hampshire, where he developed aortic aneurysm visualization software. In 2011, Steve received the Master of Science degree from the Department of Computer Science at Brown University in Providence, Rhode Island, where he studied scientific visualization and human factors under the supervision of David H. Laidlaw. He is the recipient of three Honorable Mention research awards for work published at IEEE InfoVis (poster, 2011), ACM CHI (2012), and IEEE VAST (2015). In 2014, Steve was a Brown University Teaching Fellow and taught "CSCI 0931: Introduction to Computation for the Humanities and Social Sciences" in the Department of Computer Science. In the fall of 2015, he will join the technical staff at MIT Lincoln Laboratory in Lexington, Massachusetts.

Acknowledgments

"All life is an experiment. The more experiments you make the better."
Ralph Waldo Emerson

First, I wish to thank my advisor and mentor, David Laidlaw, for teaching me how to pursue meaningful interdisciplinary research. Our meetings and conversations have been insightful, reassuring, and often very entertaining. I deeply appreciate his trust, patience, and ongoing support. In addition, I thank my dissertation readers, Jeff Huang and Remco Chang, for their guidance and help in putting together this manuscript. I am grateful to James Hays and Steven Sloman for their feedback on my thesis proposal.

So many other people in the Brown CS community have guided me over the years. I thank the technical and administrative staff for keeping me on track. In particular, I thank Lauren Clarke, Genie DeGouveia, and Dawn Reed, who helped me countless times and are genuine superheroes. I thank the CS faculty for fostering a creative, collaborative academic atmosphere. I deeply appreciate the service and support from my fellow graduate students, who listened to my practice talks and never once threw a tomato.

It has been my privilege to collaborate with generous and brilliant people during my dissertation research. In particular, I thank my co-authors: Ryan Cabeen, Jian Chen, Hua Guo, Radu Jianu, and Caroline Ziemkiewicz. I thank all my VRL labmates and CIT officemates over the years for their friendship and for expanding my intellectual interests.

Finally, I thank my family and friends for supporting me unconditionally. For everything: Mom and Dad, Nana, Greg, Brian, Laura, Allison S., Cory C., Dan K., Dan P., Deepak S., Erik M., Irina C., Rebecca M., Sasha B., my step-bunny B., and especially Alyssa.

The research in this dissertation was supported in part by Brown University, NSF award IIS-10-16623, NIH award R01-EB004155, and by Aptima, Inc. All opinions, findings, and conclusions in the dissertation are my own and do not reflect the views of the sponsoring agencies.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Problem Statement

Traditional evaluations for visualizations that focus on benchmark-task performance metrics, like accuracy and speed, often do not help developers understand *why* one visualization outperforms another. Evaluation methods that reveal the cognitive processes of end users during visual analysis are effective for identifying where designs assist or hinder analysis, but existing methods are difficult to use. Novel, practical models of a visualization's insightfulness, how people interact, and how people gaze during analysis can be used in evaluations of visualizations in order to understand how these tools support visual analysis beyond basic task performance metrics in more accessible ways than are currently possible.

Trajectory of this dissertation. The aim of this dissertation is to make it easier for visualization researchers and designers to evaluate the effectiveness of their visualizations and visual analysis systems empirically. To do this, we present three novel evaluation methods that go beyond measuring the time and accuracy of study participants on benchmark tasks. The methods we present reveal empirical evidence of cognitive processes from end users of visualization and are easier to use than traditional approaches to collecting this evidence. For each method, we demonstrate how this cognitive evidence is useful for evaluating visualization designs.

The three methods in this dissertation are presented in order of the coarseness of the cognitive evidence they collect.

- At the highest level, targeting cognitive activities on the order of tens of minutes, we present an insight-based evaluation method that measures analysts' insights during exploratory visualization along with benchmark-task performance in a within-subjects design.
- At the middle level, targeting activities on the order of tens of seconds to minutes, we present a method that helps automate the construction of predictive performance models for visualization tasks using end user interaction logs. A visualization developer does not need cognitive

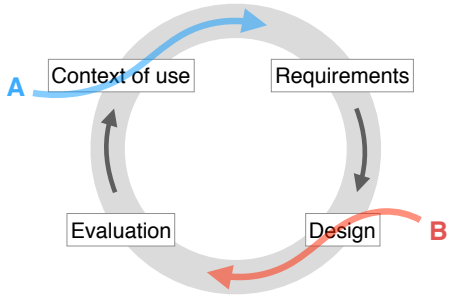


Figure 1.1: Human-centered design cycle (adapted from [75]). Entry points A and B represent alternative trajectories for visualization development that begin with qualitative research (e.g., grounded evaluation [51], pre-design empiricism [11]) or prototyping, respectively.

modeling experience to use the method. Predictions from performance models can be used to evaluate incremental design changes before they are implemented.

- At the lowest level, targeting activities on the order of fractions of seconds, we present a method that lets evaluators estimate where people gaze during visualization analysis tasks, without using an eye tracker.

Together, these methods make it easier for visualization developers to understand how analysts accomplish tasks or discover insights about their data using visualization.

1.2 Background and Motivation

The work described in this dissertation is ultimately concerned with helping people create visual artifacts that are useful cognitive aids for domain scientists and information analysts. More specifically, we aim to help others make insightful graphics, which can be interactive or static, that represent data complex enough to require analysis by a human in order to understand the data. For the remainder of this dissertation, we refer to graphics like these as *visualizations* and applications that incorporate visualizations as the principal visual component as *visualization systems*, unless otherwise qualified. Visualizations are typically created from computer applications that take encoded data and apply algorithms that use the data to create a representation of individually-valued pixels on a digital display. In other words, the software creates an image, or a sequence of image frames, using the data as input. Visualizations can also be drawn by hand and created through physical or chemical processes (e.g., darkroom photography), but in this dissertation we are chiefly concerned with computer-generated visualizations.

Evaluation is critical in the design process

One reason that designing effective visualizations is difficult is that evaluating their effectiveness remains a major challenge [85]. Yet, evaluation is a key part of the human-centered design process (shown in Figure 1.1); it helps people know whether a visualization is useful and ready to be deployed

or whether it ought to be redesigned. Without conclusive evaluation methods, visualizations that are new but not necessarily helpful aids to analysts serve to clutter the space of available tools that analysts must navigate. A worst-case scenario is that an analyst spends time and effort choosing between visualizations only to select an ineffective or misleading design that leads to missed or incorrect conclusions about their data. Since visualizations are ubiquitous data-understanding tools in domains ranging from brain science to intelligence analysis, the stakes for evaluation are high.

Visualizations can be multipurpose and evaluation methods can be multifaceted

Many aspects of visualization contribute to evaluation challenges. First, visualizations can serve different purposes, including narrative or exploratory uses. In turn, evaluations can have different goals and evaluators might not understand the methods that are most appropriate for their intended goal. Lam et al. describe seven scenarios in [74] (shown in Table 1.1) that illustrate the range of methods and questions they address. This and other recent surveys of evaluation approaches, e.g., [52], include references to published visualization studies that serve as exemplars for various evaluation goals and methods. Organizing the existing body of visualization literature based on evaluations is closely related to calls for more benchmark development for visualization [32, 84, 85]: they aim to make it easier to understand new visualization techniques in the context of what has worked or failed in the past by providing reusable methods, tasks, or data as grounds for comparison. In this dissertation, we describe methods that focus on "evaluating user performance" (quantitative user studies) and "evaluating user experience" (qualitative user studies) scenarios from Lam et al.'s taxonomy.

Evaluating how a visualization promotes cognition is difficult with current methods

Another reason visualizations are difficult to evaluate is that a common aim of visualization tools is to promote human understanding of a dataset, which is difficult to observe and quantify. Typically, evaluators ask people to use a visualization to answer predefined questions, then they judge how well a visualization "works" by the accuracy or speed with which people answer. Evaluators select tasks – sometimes with the help of experts in the data domain to ensure they are realistic – and determine ground-truth answers before recruiting participants. However, it can be difficult to select tasks with the same levels of uncertainty or exploration that occur in natural analysis settings. In addition, accuracy and efficiency of responses are somewhat coarse criteria given that many visualization tools aim to promote insights and discovery inside unfamiliar datasets.

In this dissertation, we present methods that provide empirical data about visualization use that reveals evidence of how people interact and their cognitive processes. This data is much more fine grained than basic task-performance metrics like task accuracy, and can therefore lead to more hypotheses and investigation of surprising evaluation results. The methods in this dissertation are practical approaches to using models of *participants' self-reported insights*, their *interaction logs*, and *where they might be looking during analysis tasks* as data for comparing the effectiveness of visualization designs.

Scenario	
UWP	Understanding Environments and Work Practices
VDAR	Evaluating Visual Data Analysis and Reasoning
CTV	Evaluating Communication Through Visualization
CDA	Evaluating Collaborative Data Analysis
UP*	Evaluating User Performance
UE*	Evaluating User Experience
VA	Evaluating Visualization Algorithms

Table 1.1: Seven scenarios for visualization evaluation by Lam et al. [74]. This dissertation focuses on methods that support the UP and UE evaluation scenarios.

Chapters 2–4 describe our new methods and experiments, and provide in-depth background and motivation in the context of specific research questions and hypotheses for these methods.

1.3 Summary of Contributions

This dissertation presents three types of research contributions: (1) novel evaluation methods that visualization researchers and designers can use to assess people's cognitive activities while using visualizations; (2) assessments of the methods applied to case studies; and in some cases (3) findings from the case studies about specific visualization challenges. The novel evaluation methods extend existing evaluation approaches and make them easier to use.

In this section, we give a brief description of the three evaluation methods presented in this dissertation and outline the specific contributions discussed in later chapters.

1.3.1 Contributions in Insight-based Evaluation

We present a novel method for evaluating visualizations using both tasks and exploration, and demonstrate this method in a study of spatiotemporal network designs for a visual analytics system. The method is well suited for studying visual analytics applications in which users perform both targeted data searches and analyses of broader patterns. In such applications, an effective visualization design is one that helps users complete tasks accurately and efficiently, and supports hypothesis generation during open-ended exploration. To evaluate both of these aims in a single study, we developed an approach called layered insight- and task-based evaluation (LITE) that interposes several prompts for observations about the data model between sequences of predefined search tasks. We demonstrate the evaluation method in a user study of four network visualizations for spatiotemporal data in a visual analytics application. Results include findings that might have been difficult to obtain in a single experiment using a different methodology. For example, with one dataset we studied, we found that on average participants were faster on search tasks using a

force-directed layout than using our other designs; at the same time, participants found this design least helpful in understanding the data.

This work is described in Chapter 2 and has been published in [26]. A follow-up study beyond the scope of this dissertation will appear in [34].

Contributions:

- a novel method of evaluating both task performance and insight characteristics of visualizations in a single study using a mixed design;
- a demonstration of the method in a case study of four network-layout designs for spatiotemporal visual analytics;
- guidelines for using the evaluation method in future studies.

1.3.2 Contributions in Performance Modeling

We present Tome, a novel framework that helps developers quantitatively evaluate user interfaces and design iterations by using histories from crowds of end users. Tome collects user-interaction histories via an interface instrumentation library as end users complete tasks; these histories are compiled using the Keystroke-Level Model (KLM) into task completion-time predictions using CogTool. With many histories, Tome can model prevailing strategies for tasks without needing an HCI specialist to describe users' interaction steps. An unimplemented design change can be evaluated by perturbing a Tome task model in CogTool to reflect the change, giving a new performance prediction. We found that predictions for quick (5–60s) query tasks in an instrumented brain-map interface averaged within 10% of measured expert times. Finally, we modified a Tome model to predict closely the speed-up yielded by a proposed interaction before implementing it.

This work is described in Chapter 3 and has been published in [29, 30]. The case study in this chapter is motivated by our work on brain-network visualization applications. Details about these applications, which are beyond the scope of this dissertation, have been published in [28, 33, 35].

Contributions:

- an early implementation of Tome;
- a case study with a brain-circuit visualization that demonstrates the framework's prediction accuracy for task completion times;
- demonstration of usefulness for evaluating new interaction designs. We show that performance predictions for two circuit query tasks average within 10% of expert performance, and we extend one Tome-generated model to evaluate a proposed feature that speeds up one task by 16%.

1.3.3 Contributions in Gaze Modeling with Remote Study Participants

We present the design and evaluation of a method for estimating gazes during the analysis of static visualizations using crowdsourcing. Understanding gaze patterns is helpful for evaluating visualizations and user behaviors, but traditional eye-tracking studies require specialized hardware and local users. To avoid these constraints, we created a method called Fauxvea, which crowdsources visualization tasks on the Web and estimates gaze fixations through cursor interactions without eye-tracking hardware. We ran experiments to evaluate how gaze estimates from our method compare with eye-tracking data. First, we evaluated crowdsourced estimates for three common types of information visualizations and basic visualization tasks using Amazon Mechanical Turk (MTurk). In another, we reproduced findings from a previous eye-tracking study on tree layouts using our method on MTurk. Results from these experiments show that fixation estimates using Fauxvea are qualitatively and quantitatively similar to eye tracking on the same stimulus-task pairs. These findings suggest that crowdsourcing visual analysis tasks with static information visualizations could be a viable alternative to traditional eye-tracking studies for visualization research and design.

This work is described in Chapter 4 and is under review in [27].

Contributions:

- a novel method for crowdsourcing gaze fixation estimates for visualization analysis tasks;
- qualitative and quantitative evaluations of the method that show fixation estimates are comparable to eye-tracking data on basic infovis analysis tasks;
- an evaluation of how well experts can self-assess where others will gaze during visualization analysis tasks; we compare self-assessment to data collected using Fauxvea;
- reproduced findings about visual exploration on tree layouts using the method instead of eye tracking for a more complex graph analysis task.

1.4 Potential Impact and Research Directions

The contributions in this dissertation have the potential to make the visualization-evaluation process easier and more accessible for visualization practitioners, resulting in more effective visualizations. Some methods, like the Fauxvea system in Chapter 4 that uses crowdsourcing to estimate where people gaze during visualization analysis, demonstrate protocols that may be used to evaluate other evidence of thinking and discovery beyond the scope of this work, e.g., biometric signals correlated with insight events or learning.

In Chapter 5, we discuss research directions that are motivated by the challenges and successes we encountered by applying our novel evaluation methods to real case studies. In summary, we see opportunities to:

- incorporate findings from psychology and social science into better controlled experiments involving diverse participants;

- improve human modeling as a tool for evaluation, using machine learning and open datasets of 'cognitive evidence' like eye tracking, cursor traces, etc.;
- integrate evaluation more tightly with the visualization design and development process, so that useful evaluation data can be collected passively, automatically, or systematically at the press of a button.

1.5 Roadmap

The remaining chapters of this dissertation provide details about the research contributions, then relate this work back to the larger context of visualization evaluation and open challenges. Chapters 2–4 of this dissertation describe the contributions in three areas related to visualization evaluation: insight-based evaluation, performance modeling, and gaze modeling. In each of these chapters, we provide a detailed context and motivation for the contribution, a novel evaluation method, and a case study that demonstrates using the method. In these chapters, visualization applications used in the case studies are also described. In Chapter 5, we conclude the dissertation with a summary of discussion questions, limitations of our methods, and open challenges. Below we describe the structure within these chapters.

Chapter 2 begins by introducing the problem of evaluating visualization systems using only tasks or existing insight-based methodologies. Next, it proposes an evaluation method called layered insight- and task-based evaluation (LITE) that combines benchmark tasks and open-ended exploration and show how this differs from related work. To test the method, we identified competing visualization methods for interactive network diagrams that were motivated by the needs of a collaborator studying intelligence analysis systems. These competing designs were implemented, and we designed and conducted a user study using our new evaluation method to determine which design is most appropriate for the collaborator's scenario. We conclude with a discussion of limitations and open challenges for using this method.

Chapter 3 begins by introducing the concept of predictive human-performance modeling and the challenges involved in using performance modeling to evaluate visualization systems. Next, it proposes a technique that makes using one kind of performance modeling (KLM) easier for evaluating visualizations by automating a modeling task that would otherwise require an expert to do by hand. We discuss our implementation of the method, and validate it by performing a user study with a simple brain-network visualization. We find that using the method results in task time predictions that are comparable to measured task times. Finally, we demonstrate how to use the method to evaluate an unimplemented user-interface feature in the brain-network visualization, then conclude with a discussion of open challenges related to automating the modeling process.

Chapter 4 begins by introducing the problem of collecting and using eye-tracking data for visualization evaluation. Next, it proposes a method for estimating gaze during visualization analysis tasks using remote human participants over the Web. The proposed method removes the traditional constraints of needing a hardware eye tracker and local study participants in order to collect gaze

data. The method is described, along with three experiments that demonstrate its usefulness: (1) one that validates gaze estimates compared to eye tracking on basic tasks, (2) one that shows how difficult it is for a human to estimate others' gazes without using the method, and (3) one that shows the method can be used to reproduce some findings from a eye-tracking study of tree visualization layouts. Finally, it concludes with a summary of limitations, open challenges, and conclusions.

Chapter 5 concludes the dissertation. First, it outlines the contributions of the dissertation and relates them back to open challenges in visualization research and evaluation. Next, it discusses the potential impact of this dissertation and identifies application areas where these evaluation methods could be integrated. Finally, it summarizes the thesis statement of this work.

Chapter 2

Insight- and Task-based Evaluation

Discovering insights about data is a high-level cognitive activity that has, in recent years, been integrated visualization evaluation criteria. Existing methods for measuring users' insights, in order to characterize how well a visualization promotes insights, are often difficult to use and one dimensional. This chapter describes a practical method for combining tasks and exploration in a user-study protocol in order to measure both insight characteristics and task-performance metrics for alternative visualization designs. The method helps evaluators assess two aspects of a visualization design in parallel: (1) how well it promotes insight discovery by analysts, and (2) how well it supports routine information retrieval tasks. This is useful because many visual analytics systems serve both exploratory and task-based purposes. This chapter is drawn substantially from [26].

Evaluating visual analytics systems is challenging because users need to know that the system supports both basic information retrieval tasks as well as complex reasoning and exploration. A system that is good for looking up specific data is not always good for building insights and testing hypotheses, and vice versa. At the same time, practical applications frequently demand that the same tool be used for both purposes. Despite visual analytics' focus on reasoning, many studies evaluate tools using task-based protocols that measure only user performance on low-level tasks. By contrast, insight-based methodologies aim to measure how well visualizations promote insight generation, using characteristics like the domain value of observations users make about the data model. However, these methodologies can be difficult to follow, and it is not clear how best to capture insight characteristics alongside users' task performance, as is relevant in visual analytics applications that support both targeted data searches and analysis of broader patterns.

Here we present a method for evaluating visualizations using both tasks and exploration, and demonstrate this method in a study of four spatiotemporal network designs for a visual analytics system. We call the approach *layered insight- and task-based evaluation* (LITE) because it interposes several prompts for observations about the data model between sequences of predefined search tasks. Our evaluation demonstrates the feasibility of a lightweight, within-subjects insight-based evaluation. We reflect on the relationship between users' task performance with a visualization and how well it promotes insights in assessing the best choice among four visualization designs for a spatiotemporal

visual analytics system.

The contributions of this chapter include:

1. a novel method for evaluating both task performance and insight characteristics of visualizations in a single study using a mixed design (Sec. 2.2);
2. a demonstration of the method in a case study of four network-layout designs for a spatiotemporal visual analytics system (Sec. 2.3) and findings about the designs for this application (Sec. 2.5);
3. guidelines for using the evaluation method in future studies (Sec. 2.6).

While our case study focuses on a spatiotemporal visual-analytics application where both exploration and routine search tasks might be performed, the evaluation method can be applied to other visualization domains.

2.1 Related Work

Many evaluation methods have been demonstrated in empirical visualization research. Carpendale reviews evaluation approaches for information visualization [22] and describes challenges outlined in earlier works by Plaisant [85] and others. Another overview of approaches aimed at visual analytics appears in the VisMaster consortium book [67]. The biennial BELIV workshop (Beyond Time and Errors: Novel Evaluation Methods for Visualization) has significantly added to the discussion of challenges in visualization evaluation. The research contributions in its proceedings have focused on developing more effective evaluation methods that avoid the pitfalls of traditional methodologies. Taxonomies of past studies have also been helpful in constructing guidelines for evaluating new visualizations [52, 74, 81].

In the remainder of this section, we describe methods relevant to a combined insight- and task-based evaluation, as well as to evaluations of information layouts for visual analytics.

2.1.1 Task-based Evaluations

Controlled laboratory studies with predefined tasks are commonplace in visualization research. In general, these studies aim to produce measurable outputs that are comparable among participants, design conditions, or other independent variables. Accuracy and response time for tasks are typical measures, with accuracy sometimes being used to filter task executions from the response-time analysis (e.g., [39]). In such studies the objective is to demonstrate differences in task efficiency. The evaluation approach described here collects user efficiency and accuracy measures for tasks selected using a typology covering the basic analysis questions one might ask of a spatiotemporal data model. These tasks represent analysis pieces that could be composed into a larger-scale, exploratory analysis. We acknowledge that there are tradeoffs in the realism of tasks performed in

order to gain precise, quantitative results [74]. Our study uses non-experts rather than professional data analysts, and tasks have been abstracted to remove any dependence on domain knowledge.

2.1.2 Insight-based Evaluations

Unlike task-based evaluation methods, insight-based methodologies are motivated by the realization that the goal of a visualization tool is usually to enhance understanding of the underlying data, not to improve task accuracy and efficiency [20, 77, 84]. Saraiya et al. presented an insight-based approach for evaluating bioinformatics tools [95] and later used it in a longitudinal study where insights were developed over months [96]. Characteristics of insights include the number of distinct data observations, the time needed to reach each insight, the domain value of each insight, breadth-versus-depth labeling, and other characteristics. Quantifying some of these attributes requires domain experts to participate as response coders in the evaluation. Even with this scheme, eliminating all subjectivity from the evaluation is difficult; for instance, the cutoff between a depth insight and a breadth insight might vary depending on the expert coder.

Other studies have applied similar methods to measure insight characteristics between visualization conditions. It is worth noting that insight characteristics have been adapted from those proposed by Saraiya et al. in order to fit the hypotheses of other studies. For instance, O'Brien et al. made an insight-based evaluation of two tools for visualizing genomic rearrangements using a reduced set of insight characteristics [79]: researchers counted the instances of three categories of insights as well as the total number of insights, total "hypothesis-driving" insights, and the insights per minute of analysis. Our method also uses a simplified set of insight characteristics and collects these with a single study protocol alongside task performance.

North et al. found that the results of an insight-based evaluation can both support and contradict findings of studies using benchmark tasks with the same visualizations [78]. It is possible that evaluators who use only one of these methods will miss details visible using the other. We aim to combine the two in a single, practical protocol while minimizing interactions or biases in the results. Our method differs in time scale from longitudinal studies in visualization, such as multidimensional in-depth long-term case studies (MILCS) [101]. Unlike previous insight-based evaluations, the evaluation we present uses non-expert participants. Using non-experts lets us achieve a larger sample size than would otherwise be possible, enabling us to test hypotheses about task performance and quantified insight characteristics more precisely. There are drawbacks in using non-experts; e.g., asking participants for initial analysis questions might be unreliable; however, even if domain experts were used, they would not necessarily have experience with the analysis tools in the study, as in [95]. Furthermore, we expect that combining tasks with exploration provides extra training and motivation for participants. Previous studies [23] and models [88, 89] have demonstrated how predefined tasks enhance exploratory learning of computer interfaces. While the insights themselves are likely to be less deep for non-experts than for domain experts, it is possible to compare insight-promoting characteristics between visualizations using non-experts.

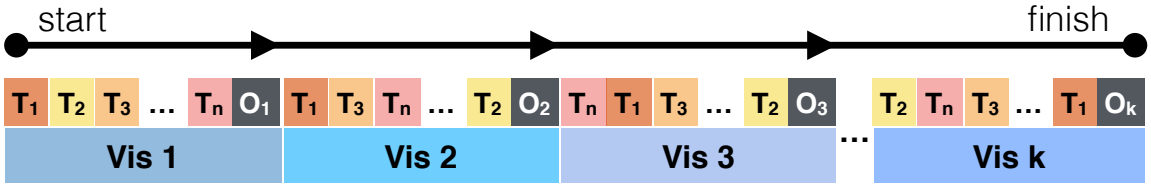


Figure 2.1: Example ordering of k visualization conditions and n task types in LITE. After each block of tasks with a visualization (labeled $T_1 \ldots T_n$), the participant is prompted for exploration and observation about the data (labeled $O_1 \ldots O_k$). Task ordering within a visualization condition is randomized using a balanced Latin square, and visualization orders are randomized between participants using a balanced Latin square. In our case study, $k = 4$ and $n = 4$.

2.1.3 Spatiotemporal Tasks and Visual Designs

An indispensable part of designing a visual analytics tool is considering the set of analytical tasks to be supported. The visualizations evaluated here are grounded in previous work on visual analysis of spatiotemporal data. In [82], Peuquet distinguished three components in spatiotemporal data and queries about those components: space (*where*), time (*when*), and objects (*what*). Users can complete queries when two of the three components are known and the other is the search target. Andrienko et al., drawing on Peuquet's work as well as other task typologies, proposed a typology for visual analytical tasks with the dimensions of search target, search level, and cognitive operation [5]. Others [2, 12, 98] have proposed more general task typologies that also apply to spatiotemporal data.

Many visual analytics designs for spatiotemporal data exist, as reviewed comprehensively in [5]. Notably, maps and timelines, the most common representations for spatial and temporal data, have been combined in previous design studies. Slingsby et al. showed that these representations can be configured as levels of a tree map in order to support different queries [102]. More recently, Andrienko and Andrienko proposed the cartographic map display and time-series display as the two visualization components in their visual analytics framework for spatiotemporal analysis [4].

2.2 Layered Insight- and Task-based Evaluation

We propose combining a lightweight insight-based evaluation adapted from Saraiya et al. [95] with a traditional task-based evaluation. We call this approach **l**ayered **i**nsight- and **t**ask-based **e**valuation, or LITE, because it interposes several prompts for observations about the data model between sequences of predefined search tasks or queries.

2.2.1 Motivation

Two main goals for this method are: 1) to measure the accuracy and efficiency of common tasks alongside insight characteristics without compromising task measurements; and 2) to measure insight characteristics while sidestepping some of the difficulties of performing the insight-based method,

such as:

D1 Users must be intrinsically motivated to look for insights during a session that might be open-ended.

D2 Training new users on visualization interfaces can be challenging. Training can fatigue users and make them try less hard in the actual study [95].

D3 After the user study, coding observations for measurable insight characteristics like domain value is difficult and requires domain experts.

Even when these difficulties are managed in an insight-based evaluation, challenges arise when performing such an evaluation separately from a task-based evaluation so as to collect measures of both task performance and insight generation. If these studies use different participants it can be difficult to draw conclusions about relationships between tasks and exploration. Individual differences or differing sample sizes must be considered.

Performing separate task- and insight-based evaluations back to back creates other challenges. If a full insight-based evaluation is performed before a task-based evaluation, open-ended exploration may fatigue users to the point that they perform poorly on the follow-up study. If a full task-based evaluation is performed before an insight-based evaluation, users may have less motivation to explore the data model: they might satisfice and report only shallow insights in order to finish the study.

2.2.2 Steps

The initial stages in a LITE evaluation are similar to those in previous insight-based methodologies. As a study session proceeds, sets of predefined tasks are interleaved with exploration periods letting participants find and record insights. The steps are:

1. *Background about the dataset* is provided, then participants are prompted for *initial analysis questions*. Alternatively, initial analysis questions can be provided by the evaluators.

2. Participants are then *trained on each task type* for different visualization conditions. Participants are not trained on exploration, as in [95].

3. When the study begins, participants complete *blocks of tasks* with each visualization condition.

4. After each block, participants *explore the data* freely using the visualization and *record insights*. Each exploration period is open-ended. In order to keep participants from skipping these periods, a minimum time requirement may be enforced before they can move to the next visualization and block of tasks. Figure 2.1 shows an example ordering of tasks and visualization conditions in which each participant completes each task type once using each visualization.

5. Finally, a *post-test questionnaire or interview* may be used after all tasks and exploration periods are finished. Subjective feedback about the insightfulness of visualizations may be used to explore findings from insight characteristics measured during exploration periods.

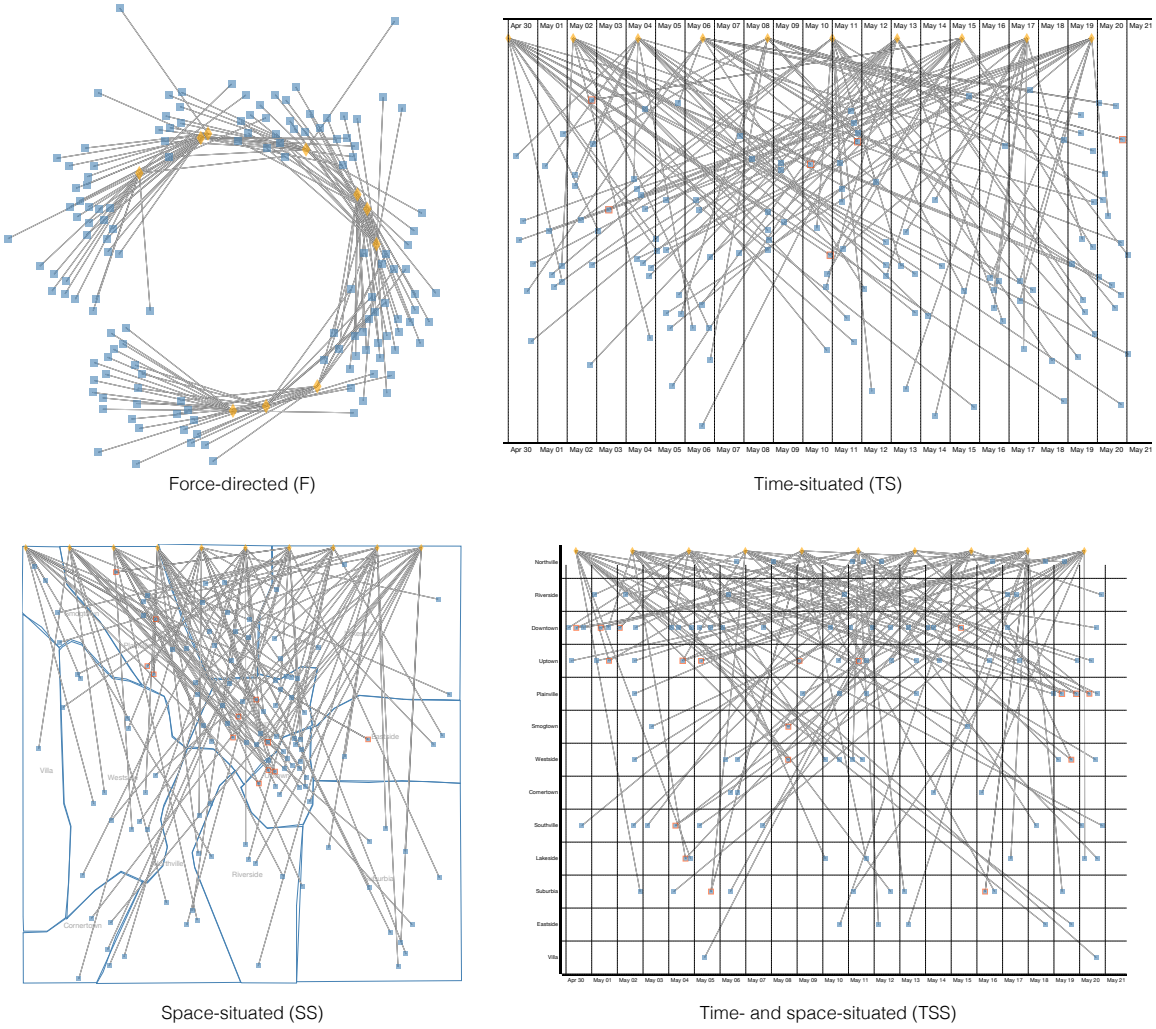


Figure 2.2: Four visualization designs were evaluated using a layered insight- and task-based evaluation: *force-directed* (F), *time-situated* (TS), *space-situated* (SS), and *time- and space-situated* (TSS). These visualizations depict microblog messages and their authors, and the designs differ in how attributes of the nodes, like timestamp and location, are used to lay out the diagram.

The proposed method addresses some difficulties of the traditional insight-based evaluation listed earlier. Study participants in LITE may feel more motivated because the session makes concrete progress through task completions rather than asking for open-ended exploration alone (D1). Tasks may improve participants' confidence with the visualizations and provide extra experience that promotes exploration and insight generation (D2). In our case study, we developed and used a scoring system without domain experts to code the value of insights (D3), but this system is not specific to LITE and could be applied to other insight-based methods.

2.3 Case Study

We evaluated four node-link diagram layout designs for an interactive visual analytics system that uses a graph-based model of real-world entities, like documents and people. We chose node-link diagrams here because of their flexibility in representing arbitrary node and edge types in the model. That said, we expect most nodes to have spatiotemporal attributes that describe when and where events happen. Based on this, we developed designs that differ in how location and time attributes are used to lay out nodes with these attributes in the diagram. Specifically, we looked at ways to project location and time attributes onto the drawing-plane axes. This is conceptually similar to previous work in which generic quantitative attributes are mapped onto axes to guide node placement [18]. In this study, we restricted ourselves to designing a layout for a single display. We considered four distinct node-link diagram layouts for the network model:

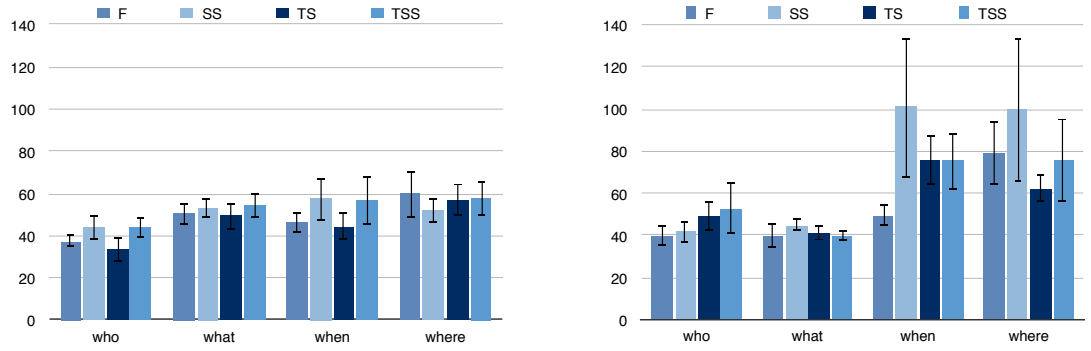
F Force-directed: A force-directed layout plots marks based on a physical simulation and has the effect of reducing visual density in the node-link diagram. Force-directed layouts are widely used and well understood. We consider this a control condition in an evaluation of visualization designs that position nodes using spatial or temporal attributes.

SS Space-situated: The space-situated layout overlays document marks on a map of the city based on documents' geotags. Nodes without geotags are placed at the top of the visualization and distributed evenly.

TS Time-situated: The time-situated layout aligns document marks with a horizontal timeline. The vertical positions of document marks are determined using a force-directed layout to reduce visual density in the diagram. Nodes without timestamps are placed at the top of the visualization and distributed evenly.

TSS Time- and space-situated: The time- and space-situated layout plots document marks according to both geotags and timestamps. Nodes without geotags and timestamps are placed at the top of the visualization and distributed evenly. In TSS the horizontal axis is a timeline, as in TS. In our prototype, the vertical axis is divided into categories corresponding to neighborhoods in the data model. Categories on the vertical axis can be ordered in different ways, for instance from top to bottom based on an ordering of neighborhood locations from northernmost to southernmost. In this case, boundaries between categories could reflect some information about the geographic boundaries between neighborhoods.

Figure 2.2 shows each of these layout designs. All visualizations were prototyped using D3 [10] and JavaScript, and share some visual encodings. The entity type of each node is double-encoded by shape and color. Marks representing documents are blue squares and marks representing people are gold diamonds; these two sorts of marks have roughly the same size in the browser. A detailed description of each node appears in a scrollable tooltip when the user hovers over the node. For documents, this description includes the author, timestamp, location, and content. In general,



(a) Small dataset

(b) Large dataset

Figure 2.3: Response times grouped by task type for each visualization type. Each participant completed each task type with each visualization type. Columns show the mean of total time spent (sec) across participants (n=12 in both (a) and (b) dataset groups) and error bars show ±1 standard error. Response times corresponding to incorrect task answers are not shown.

document content is limited to 140 characters, since documents in our data model are microblog formats like Twitter messages that enforce a content-length limit.

A simple aggregation scheme is built into each prototype so that node marks that would otherwise overlap cannot become inaccessible to the user. When marks of the same entity type overlap, both are removed from the diagram and a single aggregated mark is added. Only marks representing entities of the same type can be aggregated: thus, documents can be aggregated only with other documents. Aggregated marks retain the same entity-type encoding (shape and color) but are distinguished by a red border and increased size. Because multiple marks might overlap, the size of aggregated marks is used to encode the number of individual entities it represents.

We considered several approaches to aggregating nodes in node-link diagrams. A common approach is to aggregate a primary entity node and nodes representing its attributes into a compound node [19, 99]. This approach does not work for our case, however, as the mapping between two types of entities in our data model might be many-to-many. In our prototypes, when a node is aggregated into a different mark, each edge mark connected to that node is replaced by another that is connected to the aggregated node. The underlying data model is not changed by this process. Two nodes connected by an edge cannot be aggregated together.

Hovering over a node mark highlights all edges connected to that entity. For example, hovering over a person node highlights edges to all document nodes connected to that person by an "authored-by" relationship. Hovering over a document highlights the edge to its author node. Highlighting is implemented by restyling edges from transparent gray to opaque red. A selection interaction is also included to allow persistent highlighting during user exploration. Users can toggle selection on node marks by clicking them with the cursor.

2.4 Experimental Design

After a small pilot study, we performed an experiment to evaluate a set of hypotheses about task performance and insight characteristics for participants using four visualization designs. A $2 \times (4 \times 4)$ mixed design was used to examine the independent variables of dataset size between subjects, and visualization design and task type within subjects.

2.4.1 Hypotheses

In general, we expect that layouts that position nodes by projecting their attributes onto the axes will improve task performance and promote insight generation. Below are specific hypotheses about the effect of independent variables on task performance (H1, H2), subjective ratings from participants (H3, H4, H5), and insight characteristics (H6–H10):

H1 For all tasks, participants will be fastest using TSS, which uses both spatial and temporal attributes to lay out nodes. For all tasks, participants will be the slowest using F.

H2 Visualization type will have a significant effect on task accuracy.

H3 Participants will report feeling most confident in their task responses when using TSS and least confident when using F.

H4 Participants will report that TSS is the most helpful visualization type for understanding the data and that F is the least insightful in this way.

H5 Participants will report that TSS is the easiest visualization type to use and that F is the hardest.

H6 Total domain value for observation prompts will be highest for the TSS condition and least for the F condition.

H7 Visualization type will have an effect on the total domain value during observation prompts.

H8 Dataset size will have an effect on both total time and total domain value during observation prompts. Both characteristics will be higher in the large dataset than in the small one.

H9 The order of observation prompts will have an effect on the total domain value during those prompts.

H10 The order of observation prompts will have an effect on the total response time during those prompts.

2.4.2 Visualization Types

The four visualization types in our study are described in Sec. 2.3 and shown in Figure 2.2. In addition to the visualization layouts, the user interface included controls to filter document nodes by publication time and location. Data-filter controls are common in visual analytics applications, and it is important that the test interface match realistic usage scenarios. The time filter is a slider that can be moved on both ends in increments of one day. Node and edge marks related to documents published outside the chosen range are invisible. The location filter contains checkboxes that correspond to all neighborhood locations in the data model and can be toggled to filter marks related to documents published outside selected neighborhoods. This filter also provides "Select all" and "Deselect all" interactions.

2.4.3 Datasets

Dataset size is an important consideration in designing network visualizations. In general, larger data models add complexity and visual density that can expose scalability problems in different designs. For our experiment, two graph-based datasets of different sizes were compiled using data from the 2011 VAST Challenge Mini-Challenge #1 (MC1) [1]. Both are subsets of a synthetic dataset containing timestamped, geotagged microblog messages from residents in a city experiencing a health epidemic.

- Small – includes 10 person nodes and 139 document nodes. There are 139 "authored by" edges that connect documents to their authors. Documents were published from 13 different neighborhoods over a span of 22 days.
- Large – includes 74 person nodes and 999 document nodes. There are 999 "authored by" edges that connect documents to their authors. Documents were published from 13 different neighborhoods, and some lacked a neighborhood-specific location (i.e., location is "Vastopolis", the city name). They were published over a span of 22 days.

Both datasets were created by sampling the Challenges full-size dataset, and both contain evidence of the health epidemic in the microblogs. These 'evidence' microblogs appear in similar proportions in both datasets. We note that, while larger data models are common in real analysis scenarios, we limited the size in order to keep tasks and exploration manageable for non-expert participants during single study sessions.

2.4.4 Tasks

Based on the spatiotemporal network data model, tasks were selected using a simple typology based on *when*, *where*, *what*, and *who* queries. This task typology is similar to ones used in previous studies [5, 98]. We note that in the training and task instructions, the word "tweet" was used as a colloquialism for a microblog message. No data or services from Twitter were used in the study. The four task types are:

- *who + when + where → what*: Given a microblog's author, date, and location, summarize the content in a few words. For example, *"Cara Guthrie published a tweet in Plainville on May 20. Summarize the content of that tweet in a few words."*
- *what + who + when → where*: Given a brief summary of the microblog's content, author, and date, find where it was published. For example, *"Angela Barnett published a tweet about stylish watches on May 5. Where was that tweet published?"*
- *where + what + who → when*: Given a microblog's location, a summary of its content, and its author, find when it was published. For example, *"Bradley Church published a tweet about loss of appetite in Plainville. When was that tweet published?"*
- *when + where + what → who*: Given a microblog's date, location, and a summary of its content, find its author. For example, *"Someone published a tweet about Sham Wow in Uptown on May 11. Who is the author of that tweet?"*

An answer key for all task instances was created in order to score responses as accurate or inaccurate.

Prompts for Exploration and Observation

After each block of tasks, participants were prompted to explore the data using the visualization and record observations relevant to the epidemic in the data model. The instructions are:

> Explore the data using the visualization, then write down your observations about the data below. You should record observations about the data that are relevant to the following questions: "Do you find evidence in the data of an outbreak?"; "If so, when and where do you think it started? And how might the infection be transmitted, and is it contained?" Please number each observation.

These specific questions were taken from the instructions for MC1 [1]; they are the questions MC1 participants were asked to answer by exploring and observing a superset of the data we used. We provided these as replacements for the initial analysis questions asked as part of the insight-based methodology [95].

In response to findings from our pilot study, we added a *minimum time* for the observation prompt before each participant could move ahead to the next block of visualization tasks. During this time, participants could not access the "Next" button. When an onscreen timer showing the amount of time remaining (sec) reached 0, the "Next" button became available. At that point, participants could either continue exploring and making observations about the data or move onto the next block of tasks.

2.4.5 Participants

We recruited 24 participants for the study, 10 men and 14 women. Participants were primarily graduate and undergraduate students whose ages ranged from 19 to 30 years (M=24.4, SD=2.6).

We assigned participants randomly to the small and large dataset groups so that each had 12 people. Participant prior experience with node-link diagrams was similar in both groups. In follow-up questions after the experiment, about half the participants in each group (5 out of 12 in the small dataset and 7 out of 12 in the large dataset) responded that they 'somewhat agree' to 'strongly agree' with the statement "I have experience using visualizations of nodes and edges," using a 7-point Likert scale. The remaining participants responded that they 'somewhat disagree' to 'strongly disagree' with that statement. No participants gave a neutral response.

2.4.6 Protocol

Participants were given background information about the data model and were trained for approximately 20 minutes on the four visualization designs, including the time and location filter controls in the user interface. During this training, participants performed practice trials for each task type. With the informed consent of participants, all tasks and exploration following the training were video-recorded for later analysis.

Each participant in the study performed four blocks of tasks, one per visualization. Each block contained one instance of each of the four task types. Participants performed different task instances between blocks. For each task, responses were recorded and timed for later analysis. At the end of each task block, participants explored the data using the visualization for at least three minutes and recorded insights by typing into an on-screen text field. In total, each participant performed 16 tasks and four observation prompts. This part of the study session lasted 40–60 minutes on average. Figure 2.1 shows an example workflow for this part of the study. Ordering effects for both visualization types and task types are mitigated by counterbalancing. The order of visualization types is chosen between participants using a balanced Latin square, as is the order of task types within each visualization block for each person.

Participants were asked in a post-test questionnaire to report their preferred visualization type for each of the four task types. They were also asked to rate each visualization type for *ease of use*, *confidence* in task responses, and how well the visualization helped them *understand* what is happening in the data model. Ratings were on a 7-point Likert scale from "strongly disagree" to "strongly agree" for statements corresponding to these properties.

Insight Characteristics

Two insight characteristics were measured during each observation prompt in the study: total time spent and total domain value of observations. Total time spent had a lower bound because of the three-minute minimum time before participants could move to the next task block, as described in Sec. 2.4.4.

Scoring Domain Value We developed a simple scoring system to assess the domain value of individual observations. From a four-user pilot study, we identified two main parts of each observation about the data model: a *general claim* about the data (e.g., "It looks like the outbreak started in

Downtown"), and 0 or more specific data points that are *evidence* for the observation (e.g., "John Doe tweeted about feeling sick – from Downtown on April 19"). In the scoring system, each recorded observation has a starting score based on whether or not it makes a new claim that was not previously reported by the user during an earlier observation prompt. Because participants explore the same data model repeatedly, it is important not to double-count observations that were arrived at earlier. For our purposes, a claim is a general hypothesis, question, or remark about the data model that is potentially synthesized from multiple observations.

On top of the starting score, points are added to observations that include specific references to data points in the model as evidence for the claim. The total points awarded during an observation prompt is equal to the sum of scores of individual observations i in the set of observations I:

$$base(i) = \begin{cases} 0 & \text{if } i \text{ makes no new claim} \\ 2 & \text{if } i \text{ makes new claim} \end{cases} \tag{2.1}$$

$$bonus(i) = \begin{cases} +0 & \text{if } i \text{ includes no new, supporting data points} \\ +1 & \text{if 1 new, supporting data point in } i \\ +n & \text{if } n \text{ new, supporting data points in } i \end{cases} \tag{2.2}$$

$$score(i) = base(i) + bonus(i) \tag{2.3}$$

$$total(I) = \sum_{i \in I} score(i) \tag{2.4}$$

In this system, we expect individual observations to range from 2 (e.g., a new claim provided without details) to 5 points (e.g., a new claim with a few supporting data points). Previous insight-based evaluations scored domain values for individual insights in a similar range and also awarded points to insights based on depth [79, 95].

Two of the experimenters independently coded all insights from the experiment using this system. Both coders were doctoral candidates studying visualization and had experience with the datasets and visualization designs. Scores for the total domain value of each observation prompt from both coders were averaged for later analyses.

2.5 Results

All statistical tests described in this section were performed using SPSS. The results include support for some hypotheses from Sec. 2.4.1 but not others: we accept H4, H9, and H10; we find partial support for H3, H5, and H8; and we reject H1, H2, H6, and H7.

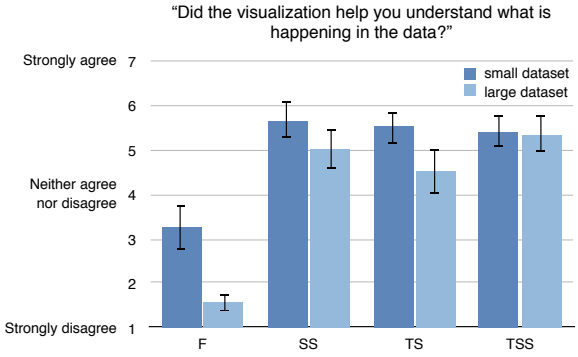


Figure 2.4: Subjective ratings of visualization insightfulness on a 7-point Likert scale collected on the follow-up questionnaire. Columns show the mean response (n=12 in both groups) and error bars show ±1 standard error.

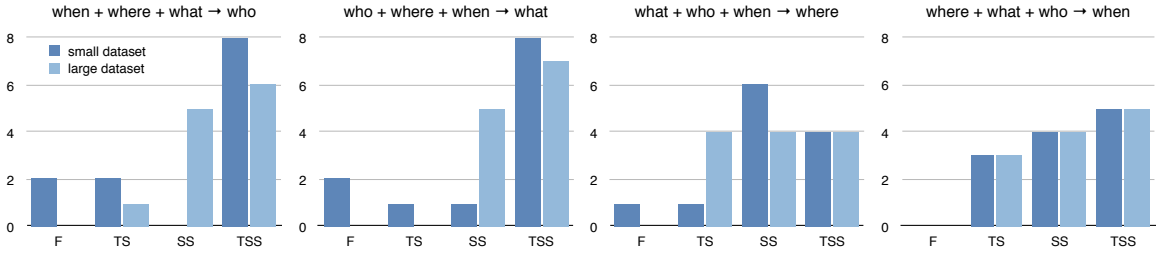


Figure 2.5: Preferences for visualization type based on task type. From left to right, the tasks shown are *who*, *what*, *where*, and *when* queries. Columns show the number of participants (n=12 in both groups) who preferred each visualization for the task.

2.5.1 Task Performance

Overall, participants were very accurate during the study: accuracy across all participants and tasks is 96% and did not differ significantly between visualization types. Therefore, we reject hypothesis H2.

We used a mixed ANOVA to analyze how the response time varied across visualization types and tasks. Average response times for all task and visualization types are shown in Figure 2.3. We performed the ANOVA analysis on the log-transformed time data, as is typical in response-time analysis. Times corresponding to incorrect task answers were replaced with the mean response time for all correct responses under the same condition. Otherwise, the repeated measures analysis would exclude data from correct tasks by participants who gave one or more incorrect answer.

The results showed that task type had a main effect on response time ($p < .001$, $F_{3,50.743} = 13.109$, with Greenhouse-Geisser correction). Pairwise comparisons were made using Bonferroni-corrected p-values by SPSS. These comparisons showed that participants were significantly faster on the *who* task than on the *when* ($p < .001$) and *where* ($p < .001$) tasks. Participants were also significantly faster on the *what* than on the *where* task ($p = .025$).

We did not find support for hypothesis H1 and reject it. In fact, as shown in Figure 2.3, we found that the mean response time using TSS is greater than the mean response time using F for

most task types in both the small and large dataset size conditions. We did not observe a main effect of visualization type on response time (p = .147, $F_{3,66} = 1.848$) or an effect of dataset size on response time (p = .179, $F_{1,22} = 1.931$).

2.5.2 Insight Characteristics

Insight characteristics measured during the study are shown in Figure 2.6 and Figure 2.7. We first analyzed the insight scores together with time spent on each insight task (using a log-transformation on times) with a multivariate mixed ANOVA with visualization type as the within-subject independent variable. The results showed that visualization type did not have a main effect on either the insight value score or the exploration time. We did not find evidence for H6 or H7 and reject both. We found partial support for H8: dataset size had a main effect on time (p = .041, $F_{1,22} = 4.702$), but not on the total domain values of insights ($F_{1,22} = 0.092$, n.s.). There was also an interaction effect between visualization type and dataset size on the total domain values (p = .035, $F_{3,66} = 3.031$) but not on time ($F_{3,66} = 0.347$, n.s.).

We then performed a similar analysis with presentation order of the visualizations as the independent variable. This time we observed a strong main effect of presentation order on both the total domain value scores of insights (p < .001, $F_{3,66} = 7.488$) and the exploration time (p < .001, $F_{3,38.256} = 11.621$, with Greenhouse-Geisser correction). We thus found support for H9 and H10. Participants spent significantly more time on the visualization that was presented first than on the following three (p = .033, p = .011, and p = .002 respectively), and also spent more time on the second visualization than on the last one (p = .02). Participants also had higher insight scores on the first visualization than on the third (p < .001) or the last (p = .005) visualization.

2.5.3 Subjective Ratings

Figure 2.5 shows the numbers of participants who preferred each visualization type for each task type. No participants who interacted with the large dataset preferred the force visualization for any task. TSS was preferred by more participants than any other visualization for both datasets, except on the *where* task. In that task, participants using the small dataset preferred SS more than TSS, and participants using the large dataset preferred TS, SS, and TSS in equal numbers.

We analyzed the subjective Likert-scale ratings of the four visualizations using a multivariate mixed ANOVA. Visualization type had strong main effects on all three measures (*understanding*: p < .001, $F_{3,51} = 18.374$; *ease of use*: p < .001, $F_{3,51} = 9.117$; *confidence*: p < .001, $F_{3,38.955} = 10.386$, with Greenhouse-Geisser correction). Dataset size had a main effect on *understanding* (p = .049, $F_{1,17} = 4.512$) and *ease of use* (p = .014, $F_{1,17} = 7.557$), but not on *confidence* ($F_{1,17} = 0.705$, n.s.).

Pairwise comparisons of the visualization types showed that participants found the force visualization the least useful in helping them understand the dataset; on average F was rated significantly

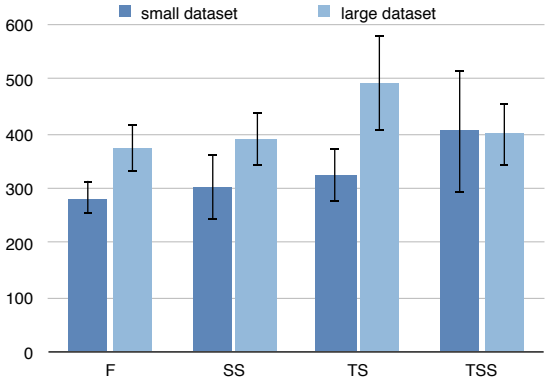
lower than TS, SS, and TSS ($p < .001$ in all cases). TSS was rated as the most helpful for understanding the data, although it was only significantly higher than F. Thus, we find support for H4. F was rated as the most difficult to use (lower than TS, $p = .003$; lower than SS, $p = .004$; lower than TSS, $p = .013$). Participants rated SS easiest (not significantly higher than TS or TSS). Therefore, we found partial support for H5. Participants also felt the least confident with the F (lower than TS, $p = .006$; lower than SS, $p = .001$; lower than TSS, $p = .007$). They were most confident with TS (not significantly higher than SS or TSS). Therefore, we found partial support for H3. Pairwise comparisons for the two dataset sizes showed that participants generally felt that they had a better understanding of the small dataset and also found the visualizations easier to use with the smaller dataset.

2.5.4 What Is the Best Design?

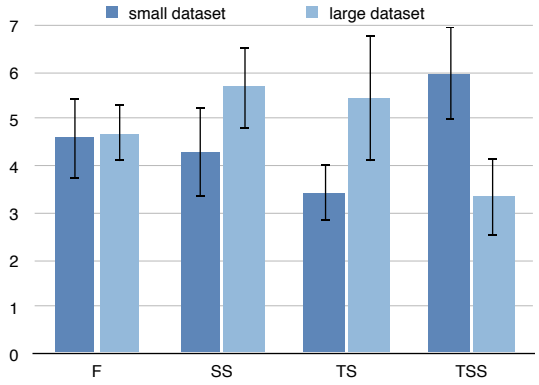
We expected that visualization designs using spatiotemporal attributes of nodes in the layout (SS, TS, and TSS) would have better task performance than F, but this was not the case. A possible explanation is that the process of using node positions along with guide marks on axes (e.g., in TS and TSS) to solve search tasks is less efficient than using the data filters for time and location. In fact, the features of these spatiotemporal layouts might have distracted participants from using filters as much as they did in the F layout. Task-execution videos showed that most participants used filtering often, even with the spatiotemporal layouts, so other factors may be involved. For instance, participants might have taken extra time to verify their answers using guide marks, and tasks in our typology might have been easy enough that this verification step added time without significantly improving accuracy.

The efficiency of filtering might also account for the significant differences in average response time between task types. Overall, participants were faster on *who* and *what* tasks, which gave both location and time components in the task description. In these tasks, participants can use both location and time filters before inspecting any nodes in the visualization. In the other tasks, participants had only enough information to use one filter – location or time – based on the task description.

Looking at task performance alongside user feedback, it is difficult to choose a best layout for the data model studied. The same layout with the fastest overall task performance (F) was also the one that participants felt least confident with overall and found the hardest to use overall. F was rated significantly less helpful in understanding the data than the other types. In such cases, a visual analytics designer must choose a layout by weighing competing objectives for the tool, including efficient task performance and subjective user preferences that might impact adoption rates and indicate insightfulness. When task efficiency is prioritized, F is a good layout choice in a visual analytics system with interactive, spatiotemporal data filtering. If we prioritize user preferences and subjective feedback about usability and insightfulness, SS or TSS might be a better layout.

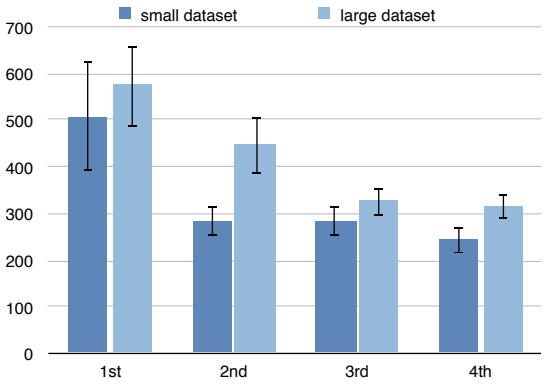


(a) Total time spent per visualization type

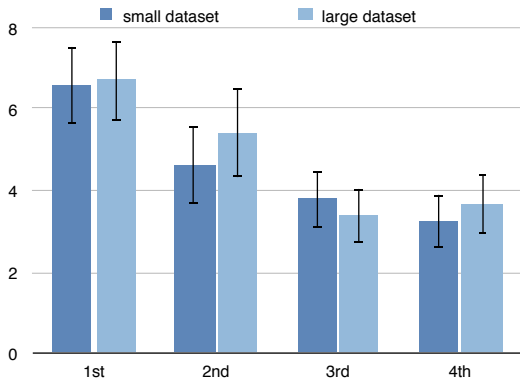


(b) Total domain value of observations per visualization type

Figure 2.6: Insight characteristics organized by the visualization type given to participants, each of whom was prompted for observations once per visualization type. The orderings of visualization types were counterbalanced across participants. Columns show the mean of total time spent (sec) (a) and the mean of total domain value (b) across participants (n=12 in both groups) and error bars show ±1 standard error.



(a) Total time spent per prompt



(b) Total domain value of observations per prompt

Figure 2.7: Insight characteristics organized by the order in which observation prompts were given to participants, each of whom was prompted once per visualization type. The orderings of visualization types were counterbalanced across participants. Columns show the mean of total time spent (sec) (a) and the mean of total domain value (b) across participants (n=12 in both groups) and error bars show ±1 standard error.

2.6 Discussion

Here we discuss what we learned about LITE through our case study and present open challenges and guidelines for using the methodology.

2.6.1 Limitations

We set out to develop a practical visualization evaluation method that combines components of task-based and insight-based evaluations. In doing so, we attempted to explore and mitigate the interactions or biases that North et al. warn about when combining these approaches [78]. Other limitations exist as well.

A practical consideration in most user studies is the time needed to run each participant, and LITE – like insight-based methods – has an open-ended exploration component that makes it difficult to estimate how long a single participant will take. In our case study, sessions lasted from 30 to 90 minutes. This uncertainty must also be considered when designing the tasks and repetitions in the task-based portion of LITE. Conducting a pilot study is a reasonable way to discover whether the task portion is feasible alongside the insight component. LITE studies with many tasks or visualization conditions might be prohibitively lengthy for participants.

A second limitation that follows from splitting time between tasks and exploration is related to the power of the results. The task-based portion of a LITE study design might have fewer trials than a dedicated task-based study design. Therefore, hypotheses could exist about task performance that can be tested in a task-only study but not in a LITE study.

Third, participants in a LITE study alternate between blocks of tasks and exploration, and that context-switching might negatively impact how people perform these activities. On the other hand, it is also possible that these switches keep participants engaged and give them a sense of making concrete progress, as mentioned in Sec. 2.2. Further study is needed to understand how these context switches affect analysis behaviors with visualizations.

Having evaluations of both insight characteristics and task performance is useful for the visual analytics application in our case study; the tool is intended both to promote insights about events and support routine data queries. Other visualizations might be aimed at only one of these purposes, and would be better evaluated using either benchmark tasks or an insight-based evaluation. Evaluators with both aims could opt to run separate studies with those methods, which is more time-consuming than running a single LITE study but might give more powerful results. These tradeoffs should be considered carefully.

2.6.2 Lessons from the Case Study

We encountered a variety of choices and challenges during our study that suggest guidelines for using the method.

Reinforcing Instructions for Different Portions of LITE

Some participants either did not understand the instructions or forgot background information on the data provided during the training period. For instance, one participant commented during her fourth observation prompt that the outbreak “Seems more over the place this time”, even though participants were told that they would explore the same data set multiple times using different

visualizations. This detail could be easy to forget since the network layouts changed between blocks of tasks. Participants who investigated the small dataset made no such observations, possibly because they were able to revisit and recognize microblogs between visualization conditions.

Other participants answered the initial questions given in the prompt directly rather than providing observations about the data that confirm or disconfirm those questions. For instance, some participants began their list of observations like "1. Yes. 2. ...". In a few cases, observations appeared to be numbered according to the three questions in the prompt (i.e., observations specifically for those questions, with no more than three separate observations) rather than being numbered by separate insights about any of the initial questions. We interpreted comments like "yes" as a belief that the corresponding initial question was true.

Guideline *Be explicit about how participants should record insights. Since participants switch between different types of responses during the task and insight portions of the study, these instructions should be reinforced.*

Coder Agreement for Insights

Overall, the two coders were fairly consistent in applying the scoring scheme to assess the domain value of insights for each prompt; their scores were within 2 points of each other for 81 out of 96 prompts, or 84.4% of the time. The coding scores are positively correlated, with Pearson's $r = 0.87$. That said, the coders agreed exactly on a score only in 36 of the 96 prompts, or 37.5% of the time. Evaluating the scoring system in future studies could help improve the scoring rules and coder manual and therefore improve consistency in assessing the domain-value insight characteristic. As far as we know, coder consistency has not been explored in depth in the literature for insight-based evaluations. In some cases, it is unclear whether multiple coders were used to assign domain values, how well they agreed, how coding conflicts were resolved, and what expertise the coders had. We believe that the practice of reporting details of the coding process will generally benefit the development and standardization of insight-based evaluation methodologies.

Guideline *In the results of a LITE or insight-based evaluation, provide information about the process of coding the domain value of insights.*

Reduced Set of Insight Characteristics

Our study measured a subset of insight characteristics adapted from previous studies [79, 95]. Some insight characteristics are difficult to measure using LITE. For instance, we did not measure the time needed to reach each insight, which could be misleading in a within-subjects design that lets participants analyze the same data model over multiple iterations. Instead, the total time for exploration in each visualization condition was used, as in [79].

We also found it difficult to count the number of individual insights without using a think-aloud protocol. Our LITE case study used an on-screen text field that lets participants record observations

in a manner similar to recording task responses. We relied on participants to input observations as a numbered list, but participants had different styles for doing this. One alternative is to guide users in constructing insights and evidence through a user-interface feature. For instance, Jianu and Laidlaw let users click nodes in a protein-signaling visualization to construct visual hypotheses about potential pathways, rather than having them provide unstructured text input [54]. Another possible solution that we did not test formally is using a think-aloud protocol during the insight portions of LITE.

We did not divide insights into categories or label them as breadth versus depth. Instead, the scoring system for domain value distinguishes between claims and supporting evidence. In the datasets used in this study, the range in the types of comments participants made was small and hence we saw no need to impose categories. Distinguishing between insights might be more practical with a dataset that contains more initial questions or in a domain with complicated relationships among data points, like systems biology.

Finally, providing the initial questions about the data model rather than asking participants for their initial questions makes it possible that participants had other unreported insights that seemed irrelevant to the specific initial questions but ultimately showed evidence of insight. Because the participants in our study were non-experts, it is a reasonable assumption that the initial questions encapsulated most of what they were able to analyze and observe. With domain experts as participants, however, there might be questions worth analyzing that would be difficult for us to predict and hard-code into the evaluation. In such cases, starting the evaluation by gathering initial questions from participants makes more sense.

Guideline *Consider the complexity of the data and participant expertise when choosing insight characteristics to measure. With a non-expert study population, provide initial analysis questions rather than requesting them from participants.*

Task and Workflow Considerations

We faced several workflow-related considerations during the design of the case study. First, there is a relationship among the training participants get, the specific tasks they perform, and the types of insights they are likely to report. It is possible that tasks or training direct users toward certain types of exploration activities. We deliberately tried to avoid this scenario in our case study by choosing low-level tasks that were unlikely to lead to insights on their own. An alternative approach used by North et al. is to give more complex tasks that can be classified into the same categories as the insights, in order to directly compare the activities that each visualization supports and promotes [78]. However, in that study, task performance and insights were measured using two separate experiments with different participants, and 'insightful' tasks could significantly interact with exploration and insights in a within-subjects design like LITE.

Second, we recognized that the results in LITE would be impossible to interpret correctly if the order of visualization conditions was not counterbalanced. Because the same data is explored by

each participant repeatedly with different visualizations, an ordering effect on the measured insight characteristics should be expected. In our case study, we found evidence that participants spent more time and reported more valuable observations during the earlier observation prompts than during later ones (see Figure 2.7). Counterbalancing the orderings of visualization conditions, as we did in the case study, can mitigate the effect of order on the results.

Finally, based on our experience in our pilot study, which let participants effectively skip the exploration portion of LITE, we decided to require in our case study a minimum time during each observation prompt. This seemed to motivate participants to explore the data; we did not find that participants sat idly while the clock counted down, or that they ended their exploration as soon as the minimum time was finished. Participants were given as much time as needed to record and explore observations, so this approach does not affect the results as it would in an insight-based study with fixed length. That said, other ways to motivate participants during the insight portion of LITE might be more effective than a time requirement.

Guideline *In LITE, choose low-level tasks that will not steer participants toward insights, and be sure to counterbalance the ordering of visualizations.*

2.7 Concluding Remarks

We present and demonstrate a method for evaluating visualizations called layered insight- and task-based evaluation (LITE) that combines predefined tasks and exploration. The method, which measures both task performance and insight characteristics, was applied in a case study of four different designs for a spatiotemporal network visualization in a visual analytics system. The results of our case study helped us assess which design best fit different objectives for the visual analytics system, including optimizing for task efficiency or promoting insights.

We also identified several guidelines for using LITE based on the study.

- Choose low-level tasks that are components of realistic analysis scenarios but will not steer participants toward insights.
- Counterbalance the ordering of visualizations to mitigate ordering effects in the insight component of LITE.
- Consider the complexity of the data and participant expertise when choosing insight characteristics to measure.
- Report details of the process of coding insights: who are the coders, how well did they agree, and how were disagreements resolved into one score?

Opportunities exist to address the challenges we encountered using LITE. We are interested in better understanding how to run lightweight, insight-based evaluations of visualizations using the non-experts who are often recruited for task-based visualization studies.

Another promising direction is looking for relationships between the types of exploratory interactions an analyst performs with a visualization and the insights they discover. This direction is important because while LITE and other insight-based methods like [95] help quantify the insight-promoting characteristics of a visualization, they do not directly tell us *how* analysts arrive at insights. It would be helpful to answer design questions like, *Do analysts who use Feature X more frequently than others during exploration discover more about the data?* While answering these questions is beyond the scope of this work, the co-authors of the study described in Sec. 2.3 and [26] performed an insight-based evaluation with data from the VAST Challenge 2014. They found correlations between insights characteristics like the *number of generalizations* and *number of facts* discovered with different types of high-level interaction patterns, like *sampling* from a selection of query results or *filtering* large results sets from a dense information display [34]. Hua Guo is the first author of this follow-up work and lead the study. My contributions included characterizing insights and interactions using taxonomies we developed, as well as data analysis. We refer readers of this dissertation to [34] for methods and full results. Further research that incorporates aspects of sensemaking and cognition into visualization research will help us develop design guidelines for promoting insights about data through visualization and interaction.

The work described in this chapter is a step toward comprehensive evaluations that assess multiple objectives for visualization systems in controlled settings, including task performance and how well systems promote insight discovery by their users. Understanding people's insights while they use visualizations provides a context for interpreting task-performance results. For example, we found that the best network layout for tasks based purely on task efficiency was not the most helpful for understanding the data, which could be a bigger priority for developers and the analysis scenarios they aim to support. In the next chapter, lower-level behaviors during visualization use, like how people interact with interface components, are analyzed to evaluate how people perform tasks, how efficient strategies are, and how altering visualization designs can affect users' efficiency at performing tasks. Like modeling insight characteristics, modeling interaction behaviors provides another cognitive context for interpreting task-performance metrics.

Chapter 3

Task Performance Modeling

This chapter focuses on how end users of a visualization interact while performing small-scale tasks on the order of tens of seconds. Observing logs of interactions and how they influence task performance reveals cognitive processes at a finer scale than the insight-based evaluation in Chapter 2. Here we describe a method that collects empirical interaction data and lets evaluators predict how quickly an expert end user can interact with a visualization interface to complete a task. The approach uses interaction logs from end users to construct predictive models semi-automatically, so an evaluator does not need to know how end users typically complete tasks or have modeling expertise to get time predictions. This chapter is drawn substantially from [30].

Quantitative user studies can help visualization developers evaluate new tool designs, but these studies can be difficult to plan and carry out. Collecting and analyzing usage data on each design iteration is often prohibitively expensive. An alternate approach is to construct a predictive model of the tool's utility (e.g., speed or accuracy for an average user) and evaluate interface changes by running the model. Despite the power of cognitive architectures like GOMS [31, 55, 57] and ACT-R [3, 16, 25] in describing these models, constructing them manually is notoriously time-consuming and error-prone [56], and they are therefore not widely used. We test the hypothesis that some difficult steps in model construction – namely, observing how users complete tasks, and then applying the construction steps – can be automated for a simple performance model called the Keystroke-Level Model (KLM) [21]. The KLM predicts the time it takes an expert user to execute necessary keyboard and mouse input, along with cognitive operations (e.g. "mental preparation", including eye movements to look at the display). In this chapter, we use the KLM to predict the time needed to execute visual queries in an interactive brain-circuit diagram.

Our novel framework, Tome, is based on the idea that the interface design and task knowledge needed for KLM use can be extracted from collections of *event-based* interaction histories, which log user-input events and descriptions of application-level events caused by that input (e.g., triggering UI widgets). A related project by John et al. called CogTool [40, 58] runs the KLM from visual *storyboards* and task demonstrations mocked up by a CogTool user. A storyboard is a directed graph that describes how an application transitions between distinct UI states given user input,

and a demonstration describes one specific sequence of user interactions that walk through that storyboard. Tome simplifies that process by automatically compiling task histories gathered by its instrumentation library into storyboards that can be imported by CogTool, which is free and open-source. Tome storyboards are prototypes that can compute a performance prediction right away for the task or can be edited as needed.

Tome aims to improve visualization and UI designers' ability to develop and quantitatively evaluate their tools. We demonstrate this by using Tome with a new brain-circuit visualization that allows neuroscientists to query neuron projections in the rat brain. Our work is in line with the agenda in *Illuminating the Path* [105], addressing the need to "develop tools and techniques to incrementally automate common tasks involved with creating visualizations". Tome reduces performance modeling to a one-time UI instrumentation cost; application users then complete their tasks as usual to gather the model's "knowledge".

The contributions of the work include:

1. a framework, Tome, for logging event-based interaction histories and constructing a canonical *storyboard* for the task executed in those histories (Sec. 3.2). This storyboard can be imported into CogTool, which constructs a KLM prediction for task-completion time, and we created a library for instrumented Java UI components.

2. a case study in brain-circuit visualization that demonstrates Tome's prediction accuracy and usefulness for evaluating new interaction designs (Sec. 3.3). We performed a quantitative evaluation of predicted completion times for quick (5–60 sec.) circuit query tasks. We show that Tome performance predictions average within 10% of expert performance, and extended one model to evaluate a proposed feature that speeds up one task by 16%.

In the remainder of this chapter, we discuss previous usability evaluation tools, history logging, and predictive performance modeling using cognitive architectures. We then describe the components and design of Tome. Finally, we describe a case study we conducted with a Tome-instrumented, interactive brain-circuit visualization and assess our findings.

3.1 Related Work

Tome is related to several projects aimed at quantitatively evaluating interface prototypes. We discuss some of these systems in Sec. 3.1.1; in Sec. 3.1.2, we discuss interaction logging methods related to the way Tome collects histories to construct performance models.

3.1.1 Modeling User Performance

Cognitive architectures for designing models of goals and task workflow have existed for decades, and have proven helpful in optimizing system design. GOMS architectures, which decompose tasks into *goals*, *operators*, *methods*, and *selection* rules, have been widely applied in HCI systems and

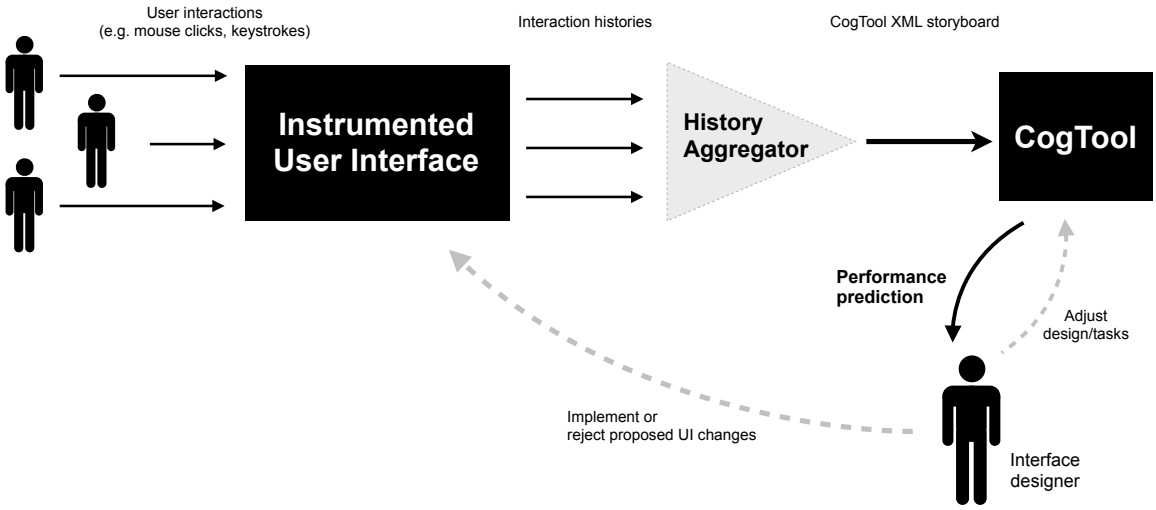


Figure 3.1: The Tome pipeline. Interaction histories are generated when end users complete tasks with the instrumented UI. Histories are aggregated by a program into canonical interaction storyboards for each task; CogTool then produces time predictions from these storyboards. The dotted arrows show actions a UI designer might take having retrieved the performance prediction from CogTool.

practice [57]. Other models based on the ACT-R architecture have been deployed to study tasks like visual search [16, 25] and information search using the Web browser [83]. Gray et al. showed in Project Ernestine that the CPM-GOMS variant, which considers a *critical path* of interactions during task workflow, accurately predicted user time of a proposed telephone operator system design. This example showed that computational models may be more accurate than designer intuition in evaluating interfaces, and motivates our efforts to generate performance models for visualizations that can be used to evaluate or guide design iterations for those tools.

Despite their power, cognitive architectures have not been used widely for visualization evaluation. One system, CAEVA [63], used an ACT-R model to simulate how a user performs analysis tasks with real visualizations. Unlike Tome, it required domain-dependent knowledge in its cognitive model, making it more difficult to deploy.

Several projects have aimed at automating model building [40, 55] or converting between architectures [103] for wider adoption. Recently, a handful of rapid prototyping tools for design evaluation [41, 58] have appeared. Most similar to our work, Hudson *et al.*'s CRITIQUE [50] generates KLM predictions by interacting with a user interface, but that interface must be built with a specialized UI toolkit called subArctic. Our framework also builds a UI model programmatically using a Java instrumentation library that can be used widely with information visualization toolkits like Prefuse [43]. Unlike CRITIQUE, Tome collects many histories to find canonical task executions and make KLM predictions, and does not require a single user to demonstrate task executions. It needs no knowledge a priori of how users complete tasks "in the wild"; it outsources that to end users, then aggregates histories.

Our work makes use of a project by John et al., called CogTool, that lets users diagram interaction storyboards visually; beneath the surface, it runs the KLM to predict user completion time on tasks, given an interface design. Visual storyboards allow for easy editing of the model task, as opposed to editing the KLM operators "by hand". CogTool lets users demonstrate tasks on a mockup, so no coding is needed, but like CRITIQUE it requires one user to construct the mockup and demonstrate task executions. Tome does not require this step. In our work, we create these storyboards programmatically from interaction-event histories. These can be imported directly into CogTool, where a user can edit the model using a visual editor or run it to predict task-completion times.

3.1.2 Interaction Histories

Recent applications of history-based usability evaluation [47] use logging in a couple ways: 1) learning about application usability or utility; 2) building application features that leverage histories for navigation or usability. The CLOTHO system [37] samples operating system state and logs user-initiated events and user annotations; it then computes *predictive variables* that classify applications as "high utility" or not for a given user. In the visualization domain, Heer et al. [44] incorporated graphical interaction histories into the visualization tool Tableau, showed how these histories improved usability – for instance, by displaying 'branches' of edits – and uncovered tool usage patterns. Our tools do not currently let users view or interact with the history as it is being compiled; instead, our goal is to provide post-processing analysis of interactions. Tome's history-logging library is application-independent and straightforward to inject into the source code of a new visualization.

Histories have also been used in environments for semi-automated construction of visualizations. VisTrails [17, 70, 97] lets users query and reuse rendering workflows to build new visualizations. Both VisTrails and Tome create reusable, modifiable artifacts from workflow records. At the same time, they differ in important ways: 1) Tome workflows are user interaction sequences, invisible to application users, and not user-selected imaging modules; 2) we attempt to reduce collections of workflows with similar semantics into a single, representative workflow.

3.2 Design Evaluation by Performance Modeling

Models of human performance with a tool can be used to guide design choices. Our work uses the KLM, which predicts the time an expert user takes to execute necessary keyboard and mouse input and also cognitive operations (e.g., "mental preparation"). Here, an 'expert user' is an application end user who knows the steps necessary to complete a task and can do them as quickly as possible. A prediction should be close to a lower bound on how long it takes to execute the critical interaction path for completing a task.

The Tome framework (shown in Fig. 3.1) gives a performance prediction for a task by: 1) collecting histories of that task as end users execute and label them; 2) determining a *canonical* interaction sequence – or, what end users might reasonably do – that completes the task; 3) compiling

a canonical history into a project for CogTool, which computes time predictions from storyboards of interactions [58].

Collecting Histories

Tome provides an interface instrumentation library based on Java's Swing toolkit that automatically produces interaction histories as end users of applications complete tasks. Library widgets like buttons are meant to be instantiated in place of respective Swing components. A basic logging API can be used to capture other events and build logging widgets. There is a one-time cost of instrumenting an interface, and these applications can be deployed 'as is'. End users can toggle logging on or off by editing a configuration file. Toggling the configuration does not affect regular application functionality, allowing end users to opt out of data collection easily.

Histories are encoded as sequences of widget-triggered interaction events and corresponding screenshots and keyboard or mouse input. In essence, each history gathers the information needed to build a graphical storyboard of the input events that cause GUI state changes throughout the task. Other subtle data is collected; for instance, the on-screen spatial bounds of widgets used are reported to model mouse-targeting times by Fitts' Law [76].

Finding Canonical Interactions

A unique aspect of this work is using many histories to produce a single time prediction for a task. The idea is that for certain types of tasks, the crowd wisdom for how to complete the task can be extracted from a set of real end-user histories.

In our implementation, when a history aggregation program is run, histories are grouped by labels that end users provide after finishing tasks. Within a group, histories that share the same interaction sequence are counted, and the most frequent sequence is treated as the canonical one for the task. This approach filters out noisy task executions (e.g., including accidental mouse or keystroke events) or unpopular strategies without having to interpret the semantics of histories. Furthermore, unlike applying the KLM manually, no modeler must know and express how to complete tasks *a priori*.

Creating a CogTool Project

The program then compiles a single history with the canonical sequence into an XML encoding of a CogTool project that describes a storyboard of the task execution (see Fig. 3.3). Finally, CogTool can open the project and run the model to predict completion time.

The ability to edit these storyboards in CogTool makes our approach more powerful than simply gathering average times from history timestamps; we can compare current UI designs against proposed changes by copying Tome storyboards and perturbing them in the WYSIWYG editor to reflect incremental design changes. This utilizes both CogTool's rapid prototyping ability and Tome's ability to gather baseline models for how end users currently complete tasks. We describe an example design revision in the section titled "Evaluating New UI Features".

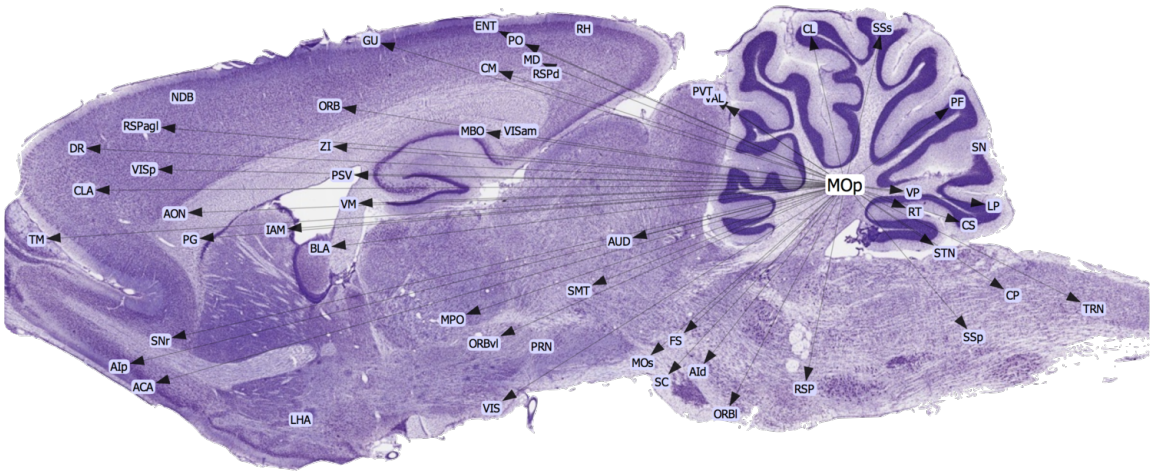


Figure 3.2: Brain diagram.

3.3 Case Study: Interactive Brain Diagrams

We incorporated Tome into the development of a sample interface, which was an interactive visualization prototype of the rat brain circuit (Fig. 3.3 thumbnails). To establish the accuracy of Tome's predictions, we instrumented this interactive diagram to collect task histories, and then compared the KLM predictions with measured task completion times. Furthermore, we modified one model in CogTool to predict the performance improvement given by a proposed feature.

3.3.1 Experimental Design

To collect test histories, eight participants were recruited as application end users and completed two types of tasks with the brain-diagram tool, as described below. All were undergraduate or graduate students in computer science. The participants were split into two groups (A and B) that completed the task types with different brain part queries. Using two groups with different instances of data gives more model predictions to compare, and therefore more confidence in generalizing that these task types can be predicted with the KLM.

With the informed consent of each participant, we recorded participant videos and screen capture for posterior analysis. Participants were trained with the brain node-link diagram for 10-15 minutes and asked to complete the following tasks as quickly and accurately as possible:

T1: *'Nearest neighbor' neural projections.* Given the name of a specific brain part p, *select* the two nearest parts on the map that share a projection (edge) with p.

T2: *Map adjustment.* Given the names of two specific brain parts p_1 and p_2 and a target part t, click and drag both p_1 and p_2 on top of t.

In both tasks, participants were required to interpret the diagram and complete several motor activities using the keyboard and mouse.

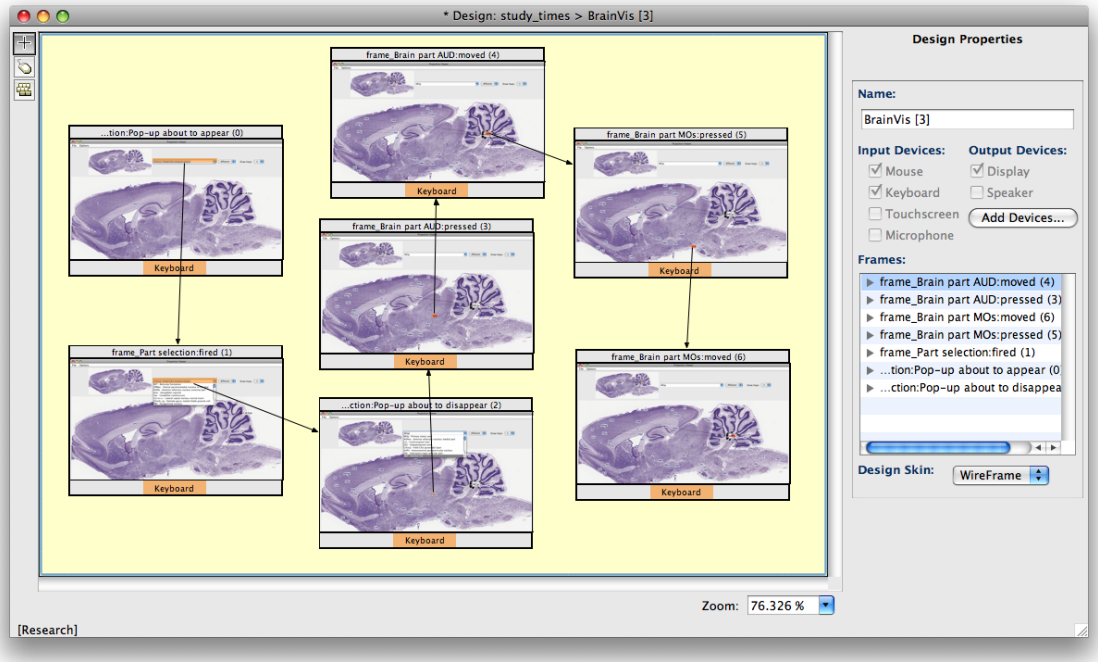
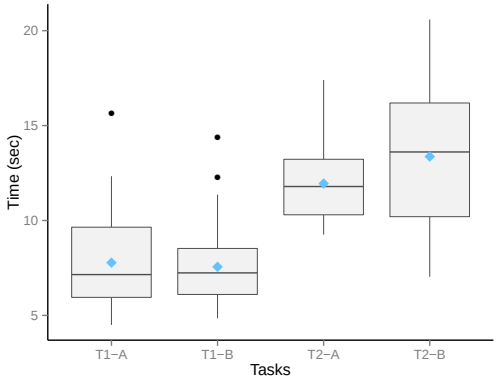


Figure 3.3: The CogTool interface showing a storyboard constructed by TOME during the T2 task. The arrows between frames indicate GUI state transitions caused by user interactions with widgets (located at the base of each arrow).

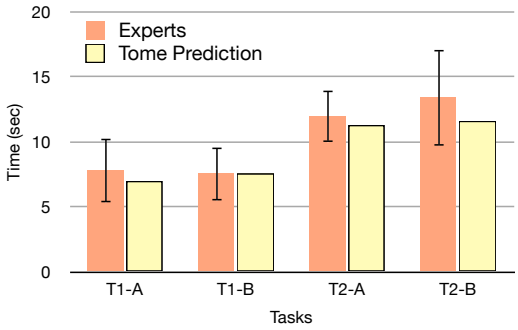
Each participant completed each task 25 times during a session of about an hour. The first five runs in each task tested the subject on all different brain parts so as to increase familiarity with the tool and task. The remaining 20 runs of each task were repeated with the same query in order to estimate the average *expert* completion time (mean from runs 11–20, using a timer) to compare to KLM. Of the 160 total expert runs collected, 5 times were discarded from this mean due to users stopping or encountering technical problems in these trials. Runs 1 through 10 for each task were training data (histories) for TOME to construct the canonical storyboard.

3.3.2 Evaluating New UI Features

After gathering histories and building TOME storyboards, we extended one of these models to evaluate a new feature before implementing it. We used a model created for the T1 task to evaluate an interaction called *radius select* that makes T1 faster. With radius select, a user can select all brain parts within a circular area of interest by choosing a central brain part and a radius on the map; this interaction can thus solve T1 quickly, without individually selecting nearby nodes. We used CogTool to edit the T1 model built by TOME after our experiment (see interface on Fig. 3.3). This amounted to adding one transition triggered by a new mouse action to the previously constructed storyboard. We simulated radius select in CogTool to produce a time prediction for experts.



(a) Box plot showing the distribution of empirical expert times for the study tasks. A blue diamond indicates the mean time for each task. Black dots show outliers identified by the `ggplot2` box-plot analysis.



(b) Expert times compared to predictions. The average prediction error for these models is less than 10%. Error bars show ±1 std. dev.

Figure 3.4: Summary of empirical expert task times compared with TOME predictions for task times.

3.4 Results

Figure 3.4 shows results for completion-time prediction accuracy for the tasks described previously. The worst error was just under 14%, on group B's T2 task. Reviewing the video for this instance showed that one participant repeatedly deviated from the most popular strategy that TOME automatically storyboarded; this participant's significantly slower task executions raised the mean expert time. Performance times over repeated trials became more consistent with experience. For both groups A and B, the standard deviation of all training set times (runs 1-10) was at least 50% higher than in expert trials (runs 11-20) for each task.

We extended the T1-A storyboard to include the *radius select* interaction. The prediction for the T1-A task using this feature was 5.7 seconds, 18% faster than the original prediction (6.9 sec). We implemented this feature and tested it with four participants – two from group B and two new participants – using the previous protocol. One of the 40 expert runs collected was discarded due to a technical problem during the trial. Expert times using radius select are about 16% faster than previous expert times.

3.5 Discussion

Our results show that most models created by TOME fall well within the 20% prediction accuracy claimed by KLM techniques [56]. These numbers might improve further under a study protocol that requires more task experience before counting *expert* runs (for instance, having users complete tasks 100 times each, rather than 20, and binning them 90/10 as *pre-expert*/*expert*). Our protocol was based merely on pilot tests that showed convergence in performance time for these tasks around 10

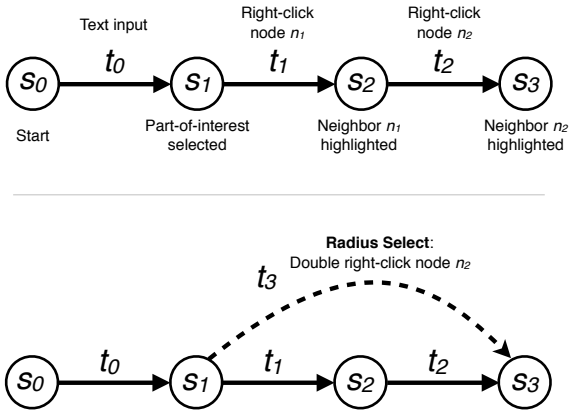


Figure 3.5: Updating a storyboard. States $s_0 \ldots s_3$ and transitions $t_0 \ldots t_3$ express the storyboards for task T1 in the original (top) and modified (bottom) designs. The dotted arrow shows the transition we added in CogTool to predict the performance improvement given by this feature.

iterations.

3.5.1 Instrumentation

Tome decreases the necessary task-modeling expertise of visualization designers at the cost of added attention to code instrumentation. In our opinion, this is a worthwhile tradeoff because it provides *accessibility*, *consistency*, and *incremental improvements* for performance modeling.

Accessibility

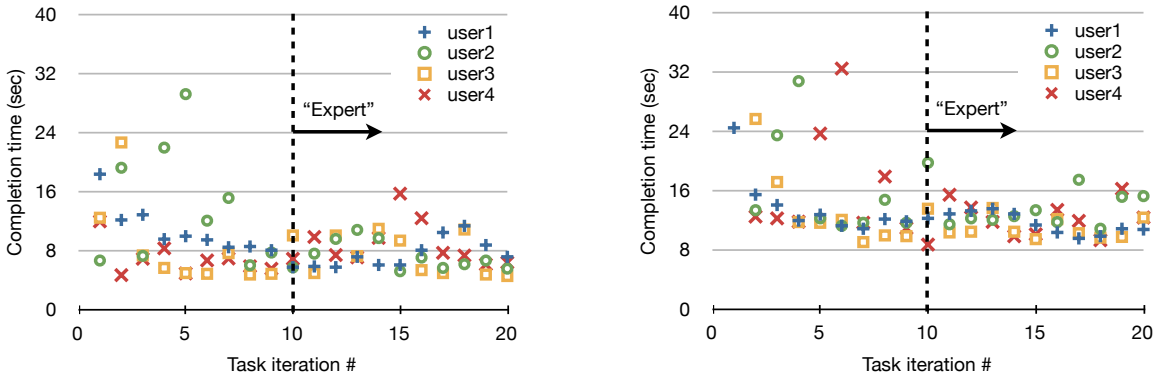
Instrumentation is more straightforward for developers than learning the "ins and outs" of a cognitive modeling architecture. Creating new models can be time-consuming even for experienced cognitive modelers. Tome does this automatically with instrumentation in a process that is invisible to end users.

Consistency

Creating models automatically is more consistent than doing so by hand, as argued by John [56] in support of CogTool. Tome makes model-building consistent *between storyboards* when widget components from the Tome library are used to construct UIs. Since logging can be disabled through a configuration file, developers are free to use these components even if they aren't sure Tome will be used, and its functionality can be turned on after code has been built and deployed.

Incremental Improvements

Even with imperfect instrumentation, the models generated by Tome are likely to capture some correct structure – and have some predictive power – and can be hand-tweaked in CogTool for fine-tuning. The framework itself, as a *tool* that can be deployed and refined, will support a community



(a) *Find nearest neighbors* task (T1)

(b) *Map adjustment* task (T2)

Figure 3.6: Performance times for 20 iterations each of tasks T1 and T2 for users in group A. Completion times begin to flatten out as users gain experience and become more consistent. The fact that completion times converge in these tasks suggests they are meaningful targets for performance model predictions. We thus evaluate Tome's performance by comparing its time predictions to measured completion times in the converged "expert" iterations.

and its applications without hiring modelers in each instance.

3.5.2 Limitations and Open Issues

Exploratory Visualization

Tome is well suited for prototyping models of tasks that require executing steps known ahead of time. Examples include simple visual query tasks, as in our brain-diagram application, data-flow creation or transformation in tools like VisTrails [70, 97] or the telephone operator UI described in Project Ernestine [31]. On the other hand, exploratory visualizations might evoke less routine interactions. For instance, generating hypotheses about a very complex data set may require different tools or approaches from a simple one. Users may follow similar high-level analytical steps in both cases (e.g. "examine the data, identify this irregularity", etc.), but capturing that similar structure could require a higher-level modeling technique than KLM.

Longer Timescales

We have examined tasks that take an average user between 5–60 seconds. These fall around the *rational* and *cognitive* bands of Newell's scale of human actions [3], requiring on the order of minutes to complete. But as described above, many visualization tasks take longer than this. If we include the scope of visual analytics tools that can require days or weeks (or longer!) of data observation and analysis, we anticipate needing modeling tools with coarser precision that can potentially predict for analytical steps beyond the keystroke unit.

Unsupervised Sets of Tomes

As mentioned earlier, each Tome has a user-provided name for the task it completes. This label is important during the reduction scheme that takes many Tomes for the same task and outputs one canonical Tome for storyboarding. Providing that label, however, is an added human step in a pipeline we are trying to automate as much as possible.

We foresee two challenges in removing this label and building storyboards from these unsupervised sets of Tomes. First, it may be difficult to gather enough data to apply clustering or learning methods for different task strategies. Noise due to errors, like extra clicks or button presses, might look like different task strategies that vary by one or two keystrokes. Second, after producing a KLM representation, an evaluator needs to know what task is simulated by the model. That person could inspect the storyboard in CogTool or look at groups of Tomes with TomeVis. Eventually a human must interpret the task model, and this could be time-consuming and error-prone. There are trade-offs between having end-users label tasks or having an overseer sort them afterwards. Another approach would be to have some users label their tasks, and use that smaller set to infer labels for the rest.

Finding Better Canonical Tomes

We described and evaluated an aggregation strategy that selects a Tome with the most frequently observed sequence of interactions by end users. However, this suffers when no clear strategy is used in a majority of histories. A specific case is when all runs contain some noise or variation, as may occur if one asked 20 users to perform a given alpha compositing task in Adobe Photoshop. Few would do it exactly the same way.

There are many possible approaches to overcoming these complications. Aside from "most frequent" selection criteria, we explored a *Longest Common Subsequence* (LCS) approach to computing a minimum necessary interaction sequence for a task, and choosing or generating a new canonical Tome based on the subsequence found. See Bergroth et al. for a overview of LCS algorithms [8]. For Tome, we have to approximate k-LCS, or LCS applied to an arbitrary number of input sequences, a known NP-hard problem. Another difficulty is determining whether a task meets the assumption of using LCS to find viable interaction strategies, i.e., that a necessary and sufficient subsequence of interactions is completed by all end users for the task. If sets of task completions do not share substantial common subsequences, then LCS will find very short or empty "required steps" to complete a task, and will predict completion times that are too short.

Finding Better Storyboards

Tome always computes a storyboard that is a directed path. Directed graphs of keyframes that include cycles are valid in CogTool, but the construction we use is convenient because it maps events from the Tome event record directly to unique keyframes. We can *demonstrate* the task of the canonical Tome by walking through each frame in this storyboard, and CogTool uses this

demonstration to run the KLM.

However, an individual Tome storyboard does not necessarily describe other interaction sequences that are valid in the live application. The fundamental issue is that whatever UI features are not used during tasks are invisible to the storyboarding tool. This doesn't affect the KLM prediction for the task in the canonical Tome, but it means that storyboard is less reusable for modeling other tasks without prior editing. One fruitful direction could be to register multiple path-storyboards automatically from an application, producing a single storyboard that describes multiple interaction scenarios with the UI. This storyboard may include cycles or branching in its frame transitions, and would allow for more tasks to be modeled in a single CogTool project by more accurately describing the state machine of the interface.

3.5.3 Limitations

We evaluated only a small number of end users and tasks in a lab setting. An extensive, longitudinal study of end users completing tasks *in situ* would be more ecologically valid than having participants repeat trials in hour-long sessions.

The main limitation with Tome itself is that only certain kinds of tasks can be modeled with the KLM. Some tasks, like freely exploring a visualization, usually do not have predictable interaction steps that make sense to model with the KLM. Additionally, tasks that can be modeled must be executed in a Tome-instrumented interface. Instrumenting an interface and editing storyboards in CogTool requires time and learning. While our experience suggests that editing a Tome-built storyboard (as in Fig. 3.5) in CogTool is faster than building one from scratch, we did not evaluate the time and difficulty involved.

Automating Tome further could make it easier to use in live settings. An open problem is automatically classifying interaction histories with task labels. What processes are needed to differentiate noisy executions from divergent strategies or different tasks? Currently, end users manually label their task histories, but this bookkeeping may be tedious or error-prone.

3.6 Concluding Remarks

We have described work toward a novel architecture for modeling human task performance from multiple interaction histories. Unlike previous methods, our system does not require an HCI expert to predict and model the steps taken by crowds of end users to complete tasks with an interface. Limitations of this approach include those of the KLM and that end users must label their task histories. Modeling higher-level cognitive processes with minimal human expertise remains an important challenge. Still, our results are encouraging: for quick diagram-query tasks, we demonstrated that Tome generates predictions within the 20% error claimed by KLM [59] and that these models can be used to evaluate iterative designs.

Earlier in this dissertation, we modeled insight-promoting characteristics of a visualization based on how end users discover and record insights during exploratory studies. This chapter focused on

user interactions that typically come before and/or after arriving at an insight. We showed how observations about task-focused interactions can be used to predict an expert's completion time for a task, which is a measure of usability. In the next chapter, we consider even finer-scale user behaviors that occur during and between keystroke-level interactions, and that hint at task-specific regions of interest inside a visualization.

Chapter 4

Crowdsourcing Gaze Estimates for Visualization Analysis Tasks

Eye movements between and during keystroke-level interactions reflect a fine scale of cognitive processes that occur during visualization use. Where people gaze is particularly meaningful during visualization tasks, because areas that are attended can reveal where a visualization design is effectively helping interesting data 'pop out' or helping a person decode task-specific information. In order to make observations about this fine-scale behavior accessible for visualization evaluation, this chapter describes a practical method for estimating where people gaze during visualization analysis tasks. The method is evaluated as an alternative to eye tracking, which captures similar gaze data but requires expensive, specialized hardware and a controlled lab setting. This work is drawn substantially from [27].

The goal of this work is to make it easier to understand where people look in visualizations during analysis tasks. This gaze information is helpful for improving visualization designs. For example, gaze data can reveal whether users attend to guide marks in a visualization. A potential application is verifying that increasing the size of marks or repositioning them draws attention to them, thus helping people interpret the visualization. Finding where people look can also help researchers understand analysis strategies and might improve their ability to identify low-level analysis activities, like finding extrema in a chart [2]. Ultimately, this information could be used to improve the usability of visualization interfaces or choose more effective visual mappings for data visualizations.

We present an evaluation of a crowd-based method for estimating gaze fixations for visualizations. The method builds on an earlier technique called the Restricted Focus Viewer (RFV) [9], an image viewer that simulates movement of the fovea by blurring the image and requiring viewers to deblur regions using the cursor. Essentially, the RFV requires a person to make manual interactions that are easily recorded and correspond to areas of the image she wants to visually decode. We constructed a Web-based version of the RFV called Fauxvea, which has incremental improvements in the design of the focus window and data capture, but most importantly can be accessed

by remote study participants like crowd workers. As a result, the method enables large-scale gaze estimation experiments, and can be used to crowdsource the production of heatmaps showing gaze for visualization stimulus-task pairs.

We demonstrate the method using workers on Amazon Mechanical Turk (MTurk). First, we compared Fauxvea fixation estimates to eye tracking from 18 participants for three common types of information visualization (infovis) charts – scatter plots, bar charts, and node-link diagrams – plus photographs. Second, we compared the Fauxvea estimates with ones predicted by participants with expertise in vision and eye tracking; we show that an individual, even one with experience with vision, cannot predict fixations as well as data from a study using Fauxvea. Third, we reproduced findings from an existing study on tree layouts from Burch et al. [14] that involves a more complex visual analysis task than in the first experiment. In these experiments, we find that gaze locations on the visualizations by online participants are qualitatively and quantitatively similar to gazes from the eye-tracking study.

The contributions of this work are fourfold:

- a novel method for crowdsourcing gaze fixation estimates for visualization analysis tasks (Sec. 4.2);
- qualitative and quantitative evaluations of the method that show fixation estimates are comparable to eye-tracking data on basic infovis analysis tasks (Sec. 4.3);
- an evaluation of how well experts can self-assess where others will gaze during visualization analysis tasks; we compare self-assessment to data collected using Fauxvea (Sec. 4.4);
- reproduced findings about visual exploration on tree layouts using the method instead of eye tracking for a more complex graph analysis task (Sec. 4.5).

Finally, we discuss limitations of the method and present opportunities for developing models of gaze that factor in both visualization stimuli and analysis tasks.

4.1 Related Work

In this section, we describe how our proposed method relates to earlier process-tracking techniques, as well as other approaches for estimating gaze without an eye tracker.

4.1.1 Focus-Window Methods

The idea of restricting visual information to the location of a pointer and tracking its location has existed for decades. An early example is the MOUSELAB system, which was aimed at tracking a study participant's cognitive process during decision tasks involving information on a computer display [60]. In this system, boxes containing information appeared blank until the participant moved the mouse into one, which would reveal the information in that box.

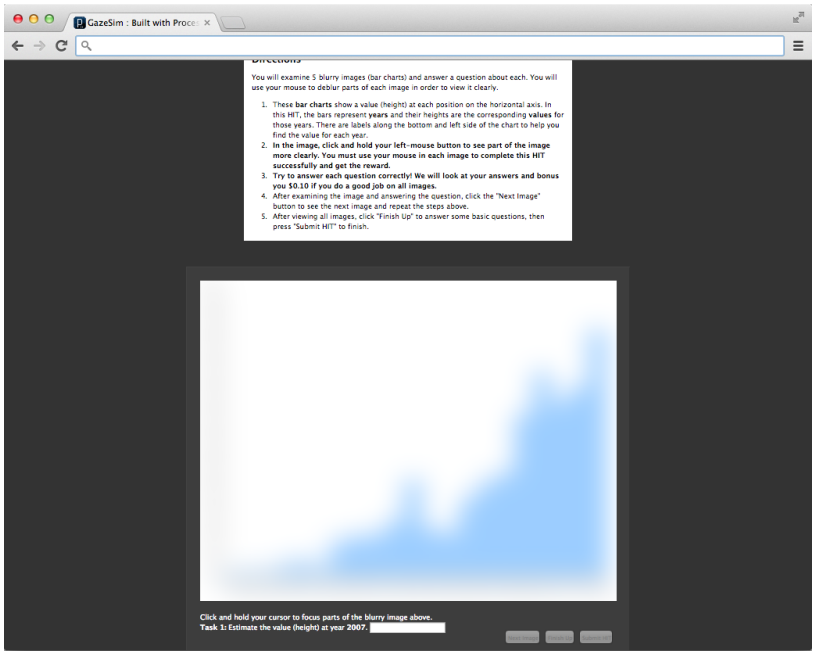
Our method is more closely related to the Restricted Focus Viewer (RFV) [9], an image viewer that requires the user to move the cursor in order to focus regions of the image. Unlike MOUSELAB, the RFV works with images that do not have predefined boundaries of information, so the mouse can be moved to focus any part of the image, and the image outside of the focus window is blurred. Cursor movements can be recorded and replayed as a proxy for actual gaze fixations. Fauxvea adapts this technique for the Web browser, with design changes that make it easier to use. Most significantly, the experiments we performed demonstrate that gaze estimates collected from online crowd workers – even in uncontrolled computing environments – are close to eye tracking for the visualization tasks we studied.

Previous evaluations of the RFV in controlled laboratory experiments have validated the technique and identified some of its limitations, but to the best of our knowledge we are the first to explore its use in estimating gaze during analysis tasks with data visualizations. Blackwell et al. found that when people evaluated causal motions in diagrams of pulley systems, gaze patterns estimated by the RFV were similar to patterns collected from an eye-tracking study on the same stimuli [9]. Bednarik and Tukiainen [7] studied how participants in a controlled eye-tracking study used a Java software debugging environment with the RFV. They found that blurring affected how some users switched gaze between areas on the screen differently compared to eye tracking, but this behavior did not affect task performance; participants were able to extract the same information using the RFV as with a normal image viewer. Stimuli like coding environments or pulley diagrams differ from typical infovis charts in how directly they encode information, so we are motivated to study the focus-window method in this context. We find supporting evidence that the approach works even in realistic visualization scenarios involving moderately complex visual representations and tasks. For instance, as described in Sec. 4.5.3, we found similarities between eye tracking and our crowd-powered RFV (Fauxvea) in how people switched between areas of interest in tree diagrams during a graph analysis task.

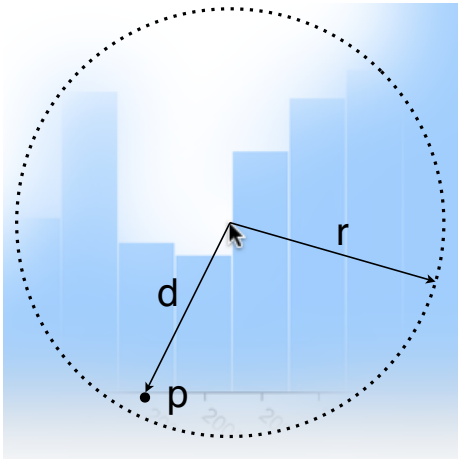
Crowdsourcing might also help users of the RFV to select appropriate parameters for their experiments. Jones and Mewhort [62] found that badly chosen blur levels outside the focus window can affect scan paths. Earlier works have proposed guidelines for setting blur levels [9, 53], but it remains a challenge to apply these. Because picking blur parameters depends on the stimulus-task and not on individual differences, blur levels for stimuli and tasks could be tested rapidly, inexpensively, and at scale in pilot studies on MTurk. In our experiment, we chose a reasonable blur level after rapid testing on MTurk.

4.1.2 Estimating Gaze on the Web

User interactions with a Web browser have been studied to predict a person's gaze, but applications have focused on domains outside of visualization. Much of this work is based on findings about the relationship between gaze and cursor movements (e.g., [24]), which are easy to track in Web applications. Other studies using Web search tasks in lab settings have identified specific types of eye-mouse coordination patterns [90, 91] and demonstrated the predictive power of cursor actions for



(a) Browser interface for Fauxvea



(b) Focus window during fixation

Figure 4.1: (a) Fauxvea interface showing analysis task instructions, the blurred image viewer, and an input field for the task answer. This example shows a bar chart task from Experiment 1. (b) Deblur under the focus window during a Fauxvea fixation. All pixels outside radius r are fully blurred, and pixels inside are blended between the blurred image and the focused one. The blend ratio for each pixel p is proportional to its distance d from the cursor location.

estimating gaze [36]. Huang et al. performed an eye-tracking study relating cursor activity to gaze in search engine results pages (SERPs), then followed up with a large-scale study of cursor tracking that linked cursor movements and results-examination behaviors in SERPs [49]. In [48], Huang et al. identified additional features beyond cursor location, including temporal and task features,

that improve the accuracy of predicting where people gaze. Our method also uses cursor actions to predict gaze, but we make use of deliberate cursor presses and releases rather than hover locations in order to measure start and end times for gaze fixations.

A Web-based system related to a moving focus window is ViewSer, which helped researchers study how remote users examine SERPs without eye-tracking [73]. The interface blurs DOM elements in the page corresponding to search results, and deblurs results when users hover over them with the cursor. One limitation of this method for evaluating visualization analysis is that it can only deblur entire DOM elements. Even if visualization components do correspond with DOM elements, e.g., using D3 [10], the size of the the deblurred component might be large enough that the hovered location does not reflect where the user is gazing at a useful level of precision. With Fauxvea, the deblurring area is based on a simple model of the human fovea. Because the focus region becomes more blurred away from its center, the user must press the cursor near the pixels she wants to see clearly. Therefore, the precision of Fauxvea for estimating gaze is linked to a parameter in the model and is not dependent on the way DOM elements are rendered.

Gaze locations in video frames were crowdsourced using a novel video interface. Rudoy et al. asked workers on Amazon Mechanical Turk ("Turkers") to watch videos then report text codes that randomly appeared on the display in different parts of the image [94]. This allowed researchers to look up an approximate region each Turker was gazing at on a given frame based on the specific code he reported. One limitation of this technique is achieving high spatial resolution of gaze estimates. Codes cannot be so close to one another that a person cannot identify them quickly. Fauxvea has a similar limitation: users might gaze at locations that are within the focus region without bothering to refocus precisely where they are attending. In practice, we find that users like to refocus directly on interesting parts of the focus region.

Webcam-based eye tracking is an alternative to methods like Fauxvea that use interaction data to predict a person's gaze. These systems use computer vision techniques to detect a person's eyes in webcam-recorded video of his or her face during interaction. Hansen and Ji provide a survey of methods for eye detection and tracking [38]. Using eye position, webcam-based eye-tracking systems predict where the user is looking at each moment as a coordinate in the display. We decided against using webcam eye tracking because both technical and social challenges exist for the intended application of crowdsourcing visualization analysis tasks. For example, Turkers' computing environments and capabilities can vary widely and affect webcam eye-tracking performance; by contrast, the technical requirements for end users of Fauxvea (i.e., a modern Web browser) are lower. Overcoming these challenges is an active research problem. In fact, Xu et al. recently demonstrated the feasibility of crowdsourcing saliency during video clips using Turkers and webcam-based eye tracking [106].

4.1.3 Crowdsourcing Visual Analysis Tasks

Recently, crowdsourcing platforms have been used to evaluate visualizations with scalable, non-expert populations. Some notable examples include Heer and Bostock's reproduction of classic

graphical perception results [42], Kong et al.'s study on TreeMap design [69], evaluations by Kosara and Ziemkiewicz of visual metaphors and percentage value reading [72], and Ziemkiewicz et al.'s study of the effects of individual differences on visualization performance [109]. This line of work has provided valuable examples and guidelines for crowdsourcing visualization analysis tasks, but they largely focus on evaluating the speed and accuracy of Turkers' task performance as outputs. Instead, we estimate gaze locations with Fauxvea using additional data from the task execution, e.g., cursor presses that facilitate performing a task.

4.2 Design and Methods

We adapted the RFV into a Web-based application called Fauxvea that estimates gaze fixations during visualization analysis tasks without using eye-tracking hardware. By design, tasks on the interface can be performed in parallel by remote users, or crowdsourced as human intelligence tasks (HITs) on MTurk.

We had two main objectives when designing and building the Fauxvea prototype:

- Collect data that is comparable to eye tracking during analysis of a static visualization.
- Enable scalable experiments with remote users, like crowdsourced participants or remote domain experts who are unavailable for local eye-tracking studies.

4.2.1 Interface Design

Comparable to Eye Tracking

The goal of this work is to make gaze data and metrics more accessible to visualization designers and evaluators. We are mainly interested in the location and duration of fixations – where the eye is focused in the field of view and has the highest visual acuity. If we assume for simplicity the "eye-mind" hypothesis, this data identifies areas of a visualization that a person cognitively processes during an analysis task.

No part of the Fauxvea viewer is focused until the user presses the cursor in the viewport. The time and location of each cursor press are recorded as the start time, end time, and location point of a fixation estimate. This is more precise than determining a fixation based on the speed of a hovering cursor, as in the original RFV. We also considered cursor-dragging as a way to simulate a scan path when the user intends to shift between gaze locations. One challenge in sampling estimated gaze locations from a drag movement is that we must trust that a user is attending to the focused part of the visualization during the drag. Some users may drag from point A to point B as quickly and (un)attentively as a person who mouse-releases at point A, moves the mouse, then presses it at point B. Additional training or other mechanisms could be used to make data collected during dragging more easy to interpret, e.g., as a sequence of fixations or pursuit movements. The prototype described in this chapter does not allow dragging, but future implementations could benefit from more investigation of this interaction.

While the cursor is pressed, image details directly under the cursor are revealed within a focus radius, as shown in Figure 4.1. The blur approximates how details in one's peripheral vision appear when the fovea is fixated elsewhere in the field of view; we use a radius instead of the original RFV's rectangular window with steps of blur. The idea of a focus spotlight is similar to other approaches in foveated imaging and Focus+Context techniques in information visualization, like semantic depth of field [71]. We note that the information loss that occurs in peripheral vision and how it affects visual search are not fully understood. Others have argued that blur is too simple of a model and that other summary statistics may be computed over a pooling region in one's vision [92]. Incorporating different models of lossy visual information into an RFV-like interface is an open challenge beyond the scope of this work.

For the experiments described later, the focus radius is equal to the 1/6 the width of each stimulus, or 133 pixels. For many desktop and laptop computing environments, we expect this radius is a reasonable approximation to the extent of the fovea, which is between 1–2 degrees of the field of view [61]. The Fauxvea focus region does not move if the user drags, forcing the user to release before pressing in a new location. This lets Fauxvea record fixation start and end times. The interface does not support zoom or pan operations, though scrolling in the browser window will not impact the interface. Within the focus radius, each pixel has a color that is a blend of corresponding colors in the original image and blurred image. The blend ratio for each pixel is proportional to its distance from the cursor press location; pixels outside the focus radius are fully blurred.

For the purpose of tracking fixations during visual analysis tasks, the visualization should be blurred enough that the task is impossible to answer correctly without fixating using the cursor. We expect users to fixate in the image using either: 1) previous knowledge of the image type (e.g., where guide marks might exist in a chart), 2) interesting low-resolution details in the blurred image or in the blended focus radius of a previous fixation location. In the Fauxvea prototype, images are blurred as a preprocessing step. For the experiments described later in this chapter, all stimuli are blurred with a Gaussian filter that we selected following a pilot study.

Scalable

Fauxvea is designed to support scalable, online experiments related to visualization analysis. In addition to the image browser, the webpage includes task instructions, controls to navigate between tasks, and an input field for task answers. Cursor interactions and answers to task questions are stored on the client during the task, then sent as a transaction to our database when the task is completed. Full histories of task executions are collected for each user.

4.2.2 Evaluation Methods

We ran three experiments to evaluate the validity of our method as a viable alternative to eye tracking for visualization. First, we collected fixation data using an eye tracker with participants performing analysis tasks on basic infovis charts; we compared these fixations to estimates collected online with our method on the same stimuli and tasks using workers on MTurk. Second, we evaluated how well

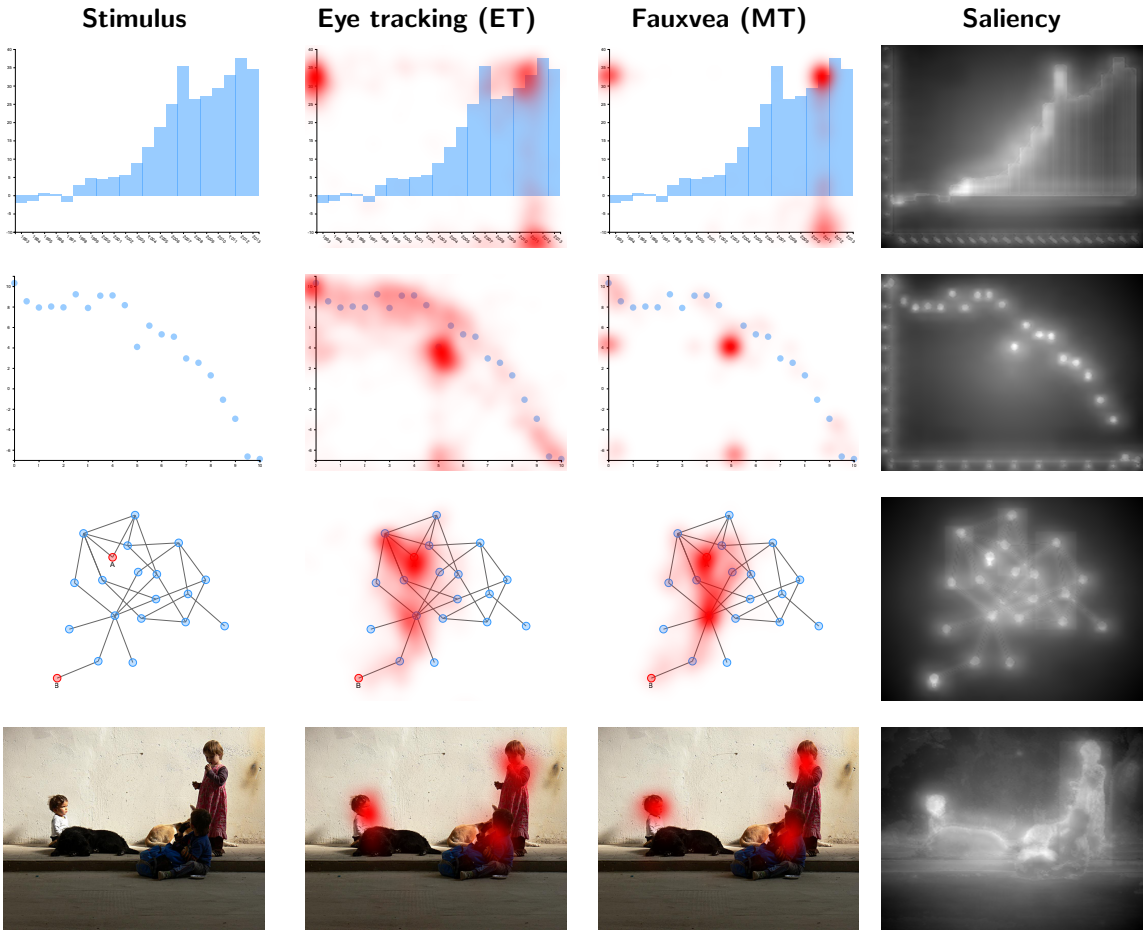


Figure 4.2: Comparison of eye-tracking gazes, Fauxvea gaze estimates from Turkers, and visual saliency maps. Red overlays show maps of fixation locations by 18 eye-tracking participants (middle-left) and between 96–100 Turkers per stimulus type (middle-right). Saliency maps (right) were computed from a visual saliency model [65], but models like these do not account for predefined analysis tasks.

self-assessment works as an alternative to eye tracking or Fauxvea for predicting gaze. Third, we used our method to reproduce findings about visual exploration on tree layouts from an eye-tracking study by Burch et al. [14] to evaluate the method in a realistic scenario with a more complex analysis task.

4.3 Experiment 1

In Experiment 1, we performed in parallel an eye-tracking study and an online study using Fauxvea with workers recruited on MTurk. Both studies asked participants to perform a set of visual analysis tasks for image stimuli. Participants were asked one question per image that required them to inspect the image. In the eye-tracking study, participants viewed the stimuli with a normal image viewer

while the eye tracker collected data. In the MTurk study, Turkers used the Fauxvea interface and pressed the cursor on the interface while inspecting each image to focus the viewer.

We hypothesize that fixation data collected from both studies will be comparable both qualitatively (**H1a, H1b**) and quantitatively (**H2**).

H1a For each stimulus, the two distributions of fixation locations from eye tracking and Fauxvea studies are qualitatively similar.

H1b For each stimulus, the two distributions show patterns that are related to the corresponding analysis task.

H2 Quantitatively, the similarity between the two distributions for each stimulus is significantly higher than the similarity between the eye-tracking distribution and random fixations drawn from a null distribution.

We evaluate **H1a** and **H1b** in Sec. 4.3.6 by generating and interpreting heatmaps of fixation locations using data from each study. We evaluate **H2** in Sec. 4.5.3 by applying a distance function (described later in Sec 4.3.4) that compares two fixation distributions.

4.3.1 Stimuli and Tasks

Three of the most common types of information visualizations were chosen for this experiment: bar charts, scatter plots, and node-link diagrams. A fourth stimulus type, photographs, were also selected from a dataset by Judd et al. [65] and serve in contrast to structured charts in our experiment. We used five images of each type in this experiment, resulting in 20 unique stimuli. All images were scaled to a width of 800 pixels, and the heights ranged from 600-623 pixels.

Each of the visualizations was created programmatically using D3 and Vega. Each bar chart and scatter plot shows 20 samples of a quadratic polynomial with noise added to each value. No axis titles are rendered in the charts. Each node-link diagram showed a graph of 20 nodes with average degree of 3. Networks of this size have been studied in previous eye-tracking experiments [86]. Blurred versions of all stimuli were created using ImageMagick. Additionally, we chose a visual analysis task for each type.

- **Bar charts:** "Estimate the value (height) at year 2008." The domain in each chart represents years from 1993 to 2013, and the year in the task description changed between images.
- **Scatter plots:** "Estimate the (x, y) position of the biggest outlier in this data trend. For example, '(3.5, 14.8)'."
- **Node-link diagrams:** "What is the fewest number of edges to travel between the red marks A and B?" Each image shows a different graph layout and has two randomly selected nodes colored red and labelled A and B.
- **Photos:** "Estimate the average age (years) of all people in the photo." Each photo contains one or more people.

Tasks at this level of complexity have been used in eye-tracking studies involving visualization analysis (e.g., [86]).

4.3.2 Eye-tracking Study (ET)

We recruited 18 participants (14 male, 4 female) for the eye-tracking portion of the experiment. Participants were undergraduate and graduate students, except two who were not students. The eye-tracker used in our study was a contact-free RED 125Hz from Sensory Motor Instruments. The stimuli were displayed on a 1600 x 900 pixel monitor and participants were seated approximately 30 inches from the monitor. In order to faithfully replicate the Fauxvea browser setup, the eye-tracking screen displayed during the study was designed to look like the Fauxvea webpage (see Figure 4.1) but with unblurred stimuli. The unblurred stimuli were shown at the same pixel resolution as used in the browser setup.

After a minimal introduction and eye-tracking calibration, participants were shown all 20 stimuli in succession and were asked to provide verbal responses to the task questions.

4.3.3 MTurk Study (MT)

We created four different HITs on MTurk and recruited 100 Turkers to complete each. Each HIT corresponded to one of four stimulus-task types: bar charts, scatter plots, node-link diagrams, and photographs. In each HIT, participants looked at five images and performed the corresponding visual analysis tasks described earlier. All participants were located within the United States.

Participants were then asked to inspect five visualizations of the same type one at a time before answering the associated question and moving on. The instructions for each HIT briefly described the image type and task. Participants were instructed to press and hold the cursor over the blurred image to reveal details. Based on results from a pilot study with 41 Turkers, we determined that training materials beyond the instructions were not necessary for these tasks. We were cautious not to suggest analysis strategies for completing these tasks. Participants could advance to the next image in the sequence after any amount of time by providing an answer to the question and clicking a button on the webpage. They were not allowed to revisit past images after moving on. Participants were paid $0.15 for completing the HIT. The instructions also told participants they could earn a $0.10 bonus if all answers were good according to an expert reviewer. The goal of the bonus is to incentivize participants to be thorough with the cursor interface in answering the questions. It also provides a quality control mechanism for analyzing Turkers' cursor data.

After the visual analysis tasks, we collected demographic information about participants' age, sex, and cursor device (mouse, trackpad, or other), as well as how often they look at images like these ("never", "sometimes", "often").

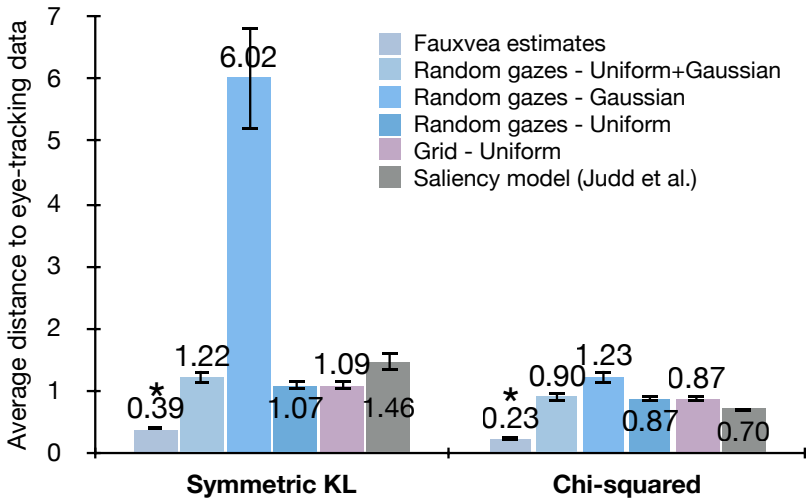


Figure 4.3: Fauxvea estimates are significantly more similar to eye tracking (Experiment 1) than each other baseline is ($p < .001$ for each). Smaller scores indicate more similarity. Error bars show ± 1 standard error.

4.3.4 Comparing Eye Tracking to Fauxvea Estimates

Distance scores were computed between the eye-tracking and crowdsourced gaze data. Low distance scores indicate high similarity between the gaze locations in both data sets. For each image, we considered two sets of points: the union of all cursor press location by Turkers using Fauxvea, and the union of all fixation locations by the eye-tracked participants. For each image, the analysis followed these steps:

1. For both sets, estimate probability density functions for the pixel locations using kernel density estimation (KDE) with a Gaussian kernel. This gives spatially smooth representations of the fixation data.
2. Discretize each smooth representation of the gazes on the original pixel grid. This creates two histograms, H_{ET} and H_{MT}.
3. Compute the distance between H_{ET} and H_{MT}.

This approach is similar to a previous study comparing gaze maps [94].

In this experiment, we tested several distance functions to compare H_{ET} and H_{MT}. In the remainder of this chapter, we report results from the χ^2 goodness-of-fit test and a symmetric version of Kullback-Leibler (KL) divergence. Both are off-the-shelf techniques that have been previously used to quantify differences between gaze sets [94] and between saliency maps and human fixation maps [64, 108]. Other metrics including Earth Mover's Distance (EMD) and Area Under the Curve (AUC) variations have also been used and combined to evaluate saliency models [15] and are applicable to our study; we limited the metrics to χ^2 and KL for simplicity.

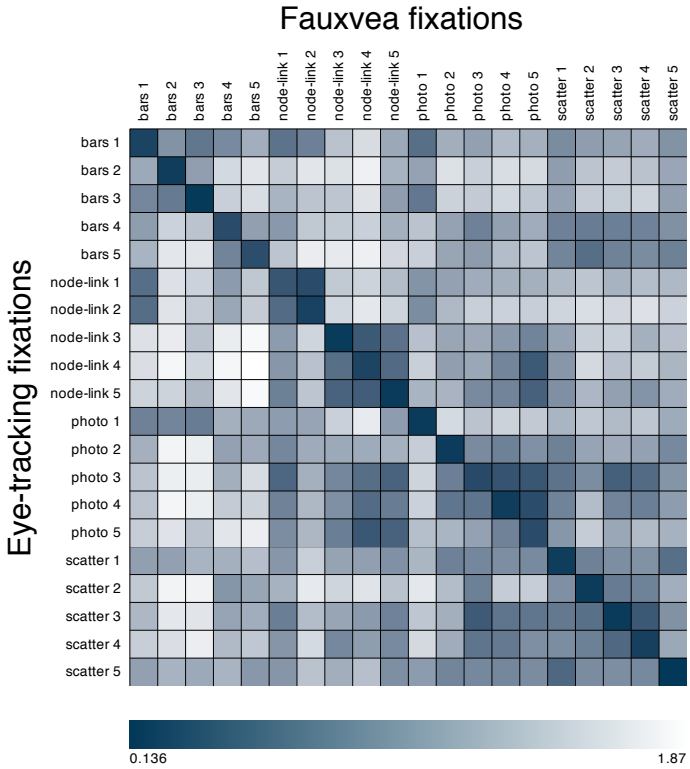


Figure 4.4: Pair-wise χ^2 distances between eye tracking (ET) and Fauxvea gaze estimates on Mechanical Turk (MT) for all 20 stimuli. As a sanity check, we compared each ET dataset to each MT dataset from Experiment 1 and visualized the distance scores in a matrix. We expect that when using a reasonable distance metric, the smallest distances (darkest cells) will appear on the diagonal, where ET and MT are compared for the same stimulus.

4.3.5 Comparing Eye Tracking to Random Gazes

For hypothesis **H1**, we try to reject the null hypothesis that Fauxvea gaze estimates are spatially uncorrelated with actual eye-tracking fixations. In this section, we describe "null" distributions, or baselines, for gazes that we expect to be less similar to ET than MT is. In Sec. 4.6.2, we discuss how building models of gaze during visualization tasks could help us test more realistic null hypotheses.

We expect the distance between a real gaze map and a random gaze map to be significantly larger than the distance between corresponding ET and MT gazes for an visualization. We considered several baseline gaze distributions that we believe are unlikely to be correlated spatially with eye-tracking fixations during visualization tasks:

- *Grid*, where fixations are evenly distributed in the stimulus.
- *Uniform*, where fixations are equally likely in any part of the image. Rudoy et al. compared χ^2 distances between eye tracking and crowdsourced fixations with their method to distances between ET and uniform random fixations [94]. This is a baseline model for saliency ("Chance") in the MIT Saliency Benchmark [15].

Task	Participants			Fixations		Familiarity with the Image Type
	Total	Age	Mouse / Trackpad / Other	Total	Per task	Never / Sometimes / Often
Bars	98	$\mu = 29.1, \sigma = 7.7$	77.6% / 18.3% / 4.1%	4,216	$\mu = 9.9, \sigma = 7.3$	52.0% / 40.8% / 7.2%
Scatter	98	$\mu = 27.5, \sigma = 6.9$	68.4% / 28.6% / 3.0%	7,484	$\mu = 24.9, \sigma = 20.5$	57.1% / 34.7% / 8.2%
Node-link	96	$\mu = 28.2, \sigma = 7.3$	74.0% / 24.0% / 2.0%	4,520	$\mu = 10.2, \sigma = 10.0$	63.5% / 27.1% / 9.4%
Photos	100	$\mu = 29.3, \sigma = 9.7$	73.0% / 24.0% / 3.0%	6,314	$\mu = 14.3, \sigma = 11.7$	6.0% / 43.0% / 51.0%

Table 4.1: Summary of Turkers from Experiment 1. "Fixations" refers to the number of cursor presses that are used to focus on the stimulus. "Total" is the number of fixations for all participants on all five stimuli in each category. "Per task" shows the mean and standard deviation for the number of fixations per user, per stimulus.

- *Centered Gaussian*, where fixations are normally distributed in the center of the image. Judd et al. showed that the center of a photograph is a good a priori estimate of gaze location [65]. This is a baseline model for saliency ("Center") in the MIT Saliency Benchmark [15].
- *Uniform + Centered Gaussian*, which is a combination of the uniform and centered Gaussian distributions.

In addition to the above baselines, we compute outputs from a visual saliency model (*Saliency*) that is task-agnostic and compare these heatmaps to our ET gazes. The motivation for this step is to see how an off-the-shelf saliency detector compares to Fauxvea for predicting gaze during predefined analysis tasks. There are many saliency detectors available that take images as inputs and output smoothed saliency heatmaps; in this experiment, we demonstrate using Judd et al.'s model [65] that is trained using a benchmark set of eye-tracking data, and is therefore transparent for others to use. We report distances from ET to each of these gaze distributions in Sec. 4.3.6.

We computed distances from ET to each baseline.

- For Grid: a set of points were generated forming a $n \times n$ grid on the stimulus, where n is the square root of the number of fixations in the gaze data.
- For Uniform, Centered, or Uniform+Centered: a set of points was sampled from the distribution, using as many fixations as in the gaze data.

The distance between these baseline point data and the gaze data was computed using the algorithm described in Sec. 4.3.4 For the baselines involving a sampling procedure (all but Grid), distances were computed for 100 sampling iterations for each stimulus, then averaged.

For the Saliency baseline, we computed the average distance between ET and the model-generated saliency map for all stimuli. To compute each single distance score for a stimulus, we used the algorithm in Sec. 4.3.4 to get a normalized histogram of the ET gazes, then we normalized the model-generated saliency map as a histogram before applying the distance function.

4.3.6 Results

In this section, we report findings from our comparison of eye tracking and Fauxvea estimates for basic infovis charts and tasks (Sec. 4.3.6).

Image type	Symmetric KL Divergence				χ^2 Distance			
	Uniform+Gaussian	Gaussian	Uniform	Grid	Uniform+Gaussian	Gaussian	Uniform	Grid
Eye tracking								
Scatter	1.09	8.10	0.75	**0.73**	0.83	1.35	0.65	**0.64**
Bars	1.60	9.84	1.08	**1.07**	1.11	1.63	**0.84**	**0.84**
Node-link	**1.20**	2.25	1.52	1.51	**0.90**	0.96	1.13	1.13
Photos	0.98	3.90	**0.96**	1.07	**0.75**	0.98	0.87	0.86

Table 4.2: Distance from eye-tracking data on different visualization types to random gazes from four baseline distributions. For each distance function, bold values show the distribution that most closely fits the image type (smallest distance score). These values suggest which null distribution is the fairest to sample for baseline comparisons against Fauxvea gaze estimates, for each of the four stimuli-task types we evaluated.

Summary statistics for our data collection experiments are shown in Table 4.1. Turkers performed 392 HITs from four different stimuli-task types. Eight Turkers submitted HITs without performing any Fauxvea cursor presses; therefore, their data are not included in our analysis.

In general, we found that fixations are distributed at similar locations between the eye-tracking (ET) and Fauxvea (MT) studies for all infovis stimuli. Heatmaps of fixation locations collected in Experiment 1 are shown in Figure 4.2, along with saliency map generated from the Judd model. Each row shows a sample visualization from our experiment, along with overlays of gaze data collected in ET and MT studies. Red overlays show normalized maps of fixation locations both by eye-tracking participants and Turkers. All stimuli and heatmaps from Experiment 1 are shown in Appendix A.

The similarities in these heatmaps between conditions support **H1a**. In most cases, white spaces in a visualization are not fixated on in either eye tracking or Fauxvea, and the most relevant marks for the analysis task are fixated on most heavily. Evidence supporting **H1b** is clearest in the heatmaps of bar chart and scatter plot, where specific axis labels corresponding to correct task responses (i.e., column heights or (x, y) coordinate values) are fixated on heavily while the others are largely ignored. Heatmaps can also illustrate what visual-analysis strategies are used to complete tasks with less structure that do not use guide marks. For example, it is clear that people primarily fixate on faces in both eye-tracking and Fauxvea results to answer the photograph task *"Estimate the average age of all people in the photo"* and not other context clues (see Figure 4.2 for an example).

We found quantitative evidence that fixation estimates made with Fauxvea are more similar to eye tracking than the baseline estimates we tested. While it is not surprising that Fauxvea performs better than random and task-agnostic gaze estimates, the results confirm a basic requirement for the method and also demonstrate how to quantitatively compare two sets of fixation locations. As we discuss in Sec. 4.6, opportunities exist to use this evaluation approach to compare new models of gaze against each another.

The distance between ET and MT (0.39 using the symmetric KL function, 0.23 using the chi-squared function) is significantly less than the distance between ET and each of the baselines ($p <$ .001 for all paired, two-tail t-tests), which supports **H2**. Figure 4.3 shows the average distance for both symmetric KL divergence and χ^2 distance between all ET and MT data, and the difference between ET data and each of the baselines we considered.

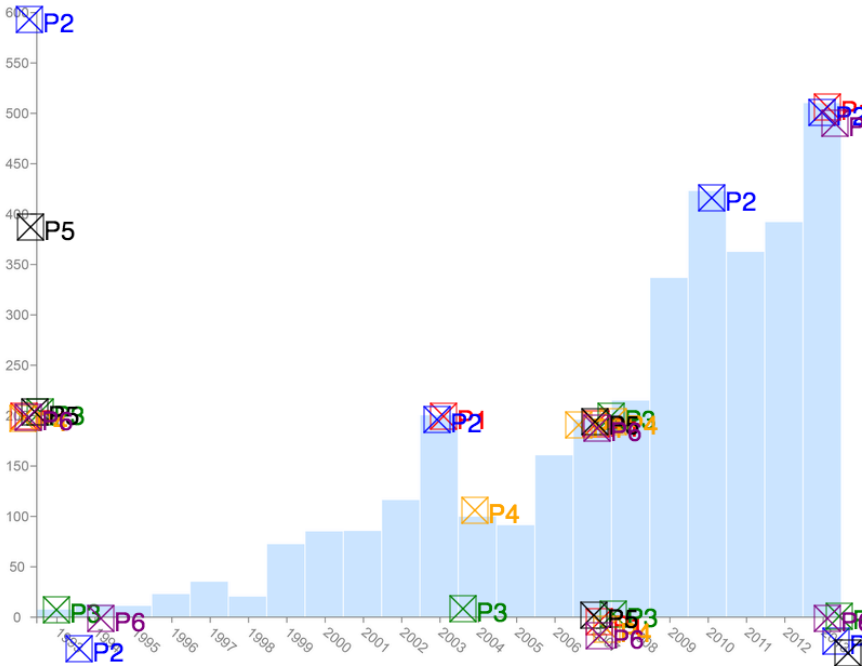


Figure 4.5: Predicted fixation locations for the task "Estimate the value (height) at year 2007" by participants (P1–P6, marked with unique colors) in Experiment 2.

Table 4.2 shows the average distance scores for both metrics between the ET data for the four image-task types and each of four baseline null gaze distributions. These values suggest which null distribution is most fair to sample for random comparisons against Fauxvea gaze estimates, for each of the four stimuli-task types we evaluated. In general, fixations on charts with guide marks near the edge of the image (e.g., bar charts and scatter plots) are most similar to samples from a grid-based or uniform distribution rather than a distribution with higher likelihood near the image center. Marks like axes are critical for decoding information, but where a person attends is usually task-dependent. By contrast, photographers tend to frame the most interesting parts of the image near the center.

4.4 Experiment 2

In this small-scale follow-up experiment, we test whether people with experience and interest in eye tracking are able to reliably predict fixation locations for the tasks and stimuli in Experiment 1. The goal of the experiment is to understand whether it is viable for a person to self-assess where gazes happen during a visualization task instead of running a crowdsourced study with Fauxvea or performing an eye-tracking study.

In Experiment 2, participants were limited to those who had experience or interest in eye tracking, and therefore represent individuals who are most capable and likely to self-assess where others will gaze as an alternative to performing a user study with others. This provides a reasonable

baseline for comparing self-assessment to Fauxvea. Another interesting question is whether non-expert crowdsourced workers like Turkers are also capable of predicting where others gaze. If so, it might be possible for a visualization developer to crowdsource the gaze prediction task to a small number of workers instead of either (1) doing it himself or (2) running a typical Fauxvea user study with a larger sample size, which is more costly. Answering this question is beyond the scope of Experiment 2, but understanding the range of capabilities of remote non-experts is an important challenge for crowd-powered systems that we discuss more in Chapter 5.

4.4.1 Methods

We recruited six participants (four male, two female) who were researchers in human-computer interaction or computer vision at a major research university in the U.S. Each was right-handed, had normal or corrected-to-normal vision, and identified himself as having experience or interest in learning where people look in images. The ages of participants ranged from 25–35 years (M=28.7, SD=4.13). All participants passed an Ishihara test for normal color vision.

In the first task, each participant was seated about 18 inches from a 24" 1920 x 1200 pixel monitor and viewed each of the 20 task-stimulus pairs from Experiment 1. For each task, participants were asked to select five or more locations in the stimulus they believe people performing the task will fixate on. Participants indicated their fixation locations by clicking a cursor at locations inside the image. The tasks were presented in the same order that participants in the eye-tracking condition (ET) in Experiment 1 performed them. We recorded each predicted location.

In the second task, participants were seated at the same display and shown eye tracking (ET) and Fauxvea (MT) gaze heatmaps for each task-stimulus pair, side by side. The ET and MT heatmaps were generated from data collected in Experiment 1. For each of the 20 pairs, each participant was asked to click on the heatmap she believed was generated from real eye-tracking data; the position of the ET heatmap – left or right – was randomly assigned between stimuli. Participants could also select a "Too close to call" button if they could not identify the ET heatmap. We scored how accurate each person was in selecting the ET heatmaps in the set of stimuli.

4.4.2 Results

The results from this study are primarily qualitative for two reasons: 1) recruiting a large sample size of people with experience in vision or eye tracking is difficult; 2) when asked to select fixation locations, most participants selected only the minimum number of locations we requested. Therefore, the data are sparse.

We examined the results from the first task by visualizing all fixations predicted by the expert participants and looking for patterns in how participants chose points across stimuli. These gaze predictions for all stimuli are shown in Appendix B. A filterable visualization of these fixation predictions is available online at: `http://bit.ly/fauxvea-sup`. In general, we found that most participants identified similar key areas in each stimulus, but predictions varied in how people

attend to areas of the stimuli that are visual salient but irrelevant to the particular tasks. Figure 4.5 shows an example where 5 of 6 participants predicted that others would fixate on the top of the column, x-axis guide marks, and y-axis guide marks corresponding to the task; however, there was little consensus on which other bars or guide marks people would attend. One participant (P2), a graduate student who studies eye tracking, did not predict any fixations on the task-specific areas of the bar chart; in a follow-up interview, she indicated that she focused on marking only visually salient regions.

Our main insight is that even with expertise in thinking about where others will gaze in an image, as an individual it is difficult to predict how a population of people will gaze during a visual analysis task. Several participants commented that they made predictions by first solving the task on their own, then reporting where they gazed during that trial; however, this strategy limits the evaluator to only one perspective and is not viable for estimating gazes from a population that might analyze a visualization in different ways.

In the second task, the experts correctly identified the real eye-tracking heatmap with 68.3% accuracy on average. They incorrectly identified the Fauxvea heatmaps as eye tracking 23.3% of the time, and 8.3% of the time they selected "Too close to call". In a follow-up interview, most participants indicated that when heatmaps were noisier it was an indicator of real eye tracking. P5, who had run eye-tracking studies prior to our experiment, commented that adding random noise to the Fauxvea heatmaps could make them look closer to the eye-tracking heatmaps.

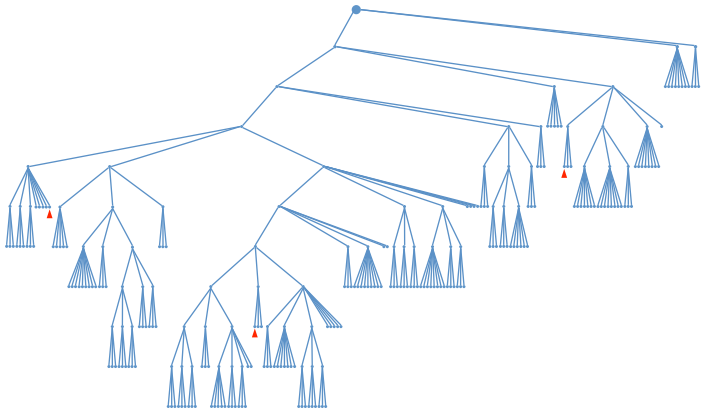
4.5 Experiment 3

We ran a third experiment to see if the Fauxvea method is able to reproduce findings about visual exploration behaviors from an existing eye-tracking study using tree visualizations. This follow-up was aimed at evaluating the external validity of Fauxvea beyond the basic visualization interpretation tasks in Experiment 1. We reproduced the task and three stimuli from an eye-tracking study of traditional, orthogonal, and radial tree layouts [14]. Burch et al. note that these layouts are "frequently used in many application domains, they are easy to implement, and they follow aesthetic criteria for tree drawing".

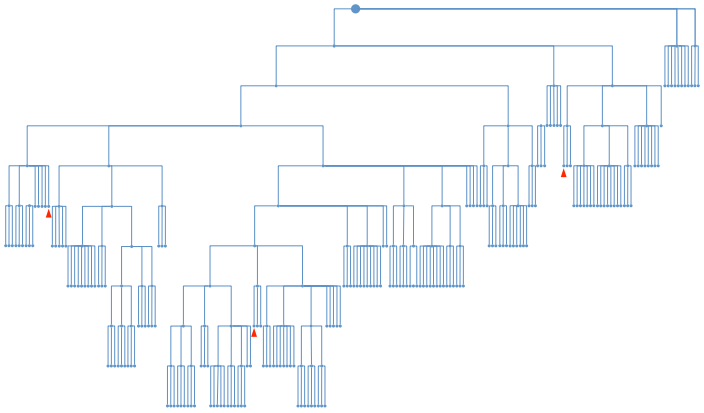
In general, we hypothesize that fixation locations and transition frequencies reported in the original study will be reproduced by running a similar experiment with Fauxvea (**H3**). In addition, we hypothesize that the three findings in the section titled "Analysis of Exploration Behavior" in [14] will be corroborated with data collected from a reproduction of the experiment using Fauxvea instead of an eye tracker (**H4a, H4b, H4c**).

H3 Transition frequencies between predefined areas of interest (AOIs) in the original study will be similar to transition frequencies reproduced using Fauxvea.

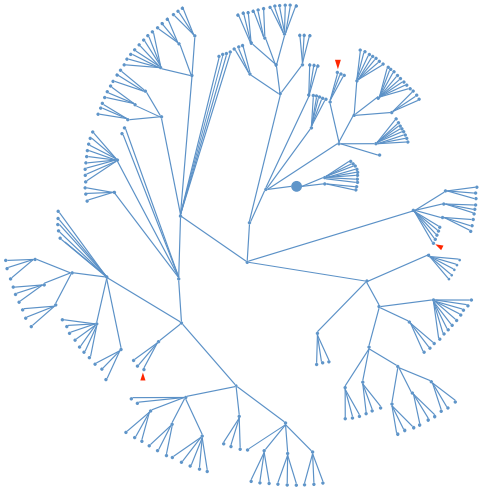
H4a Participants will jump more frequently between leaf nodes that are near each other in the traditional layout compared to the orthogonal and radial layouts.



(a) Traditional layout



(b) Orthogonal layout



(c) Radial layout

Figure 4.6: Three stimuli for Experiment 3. In each diagram, the root node is indicated by a larger circle mark, and the target nodes for the common-ancestor task are indicated by red arrows.

H4b The pixel distance between the marked leaf nodes will affect the transition frequency.

H4c Participants viewing the radial layout will transition back from the root node to AOI 2 more frequently compared to the traditional and orthogonal layouts.

We evaluate **H3** in Sec. 4.5.3 by analyzing the most frequent destination AOI from each source AOI. We evaluate **H4a**, **H4b**, and **H4c** in Sec. 4.5.3 by analyzing specific patterns in the corresponding transition tables.

Testing these hypotheses will help us evaluate whether Fauxvea can reproduce findings about visual exploration visualization without using an eye tracker. We note that we do not compute distance scores, as we did in Experiment 1, because the fixation data from the eye-tracking study are not available. Furthermore, rather than measuring how closely the Fauxvea estimates match the eye-tracking heatmaps, the main goal of this experiment is to corroborate or reject the findings from [14] using a similar analysis.

4.5.1 Stimuli and Tasks

Participants were shown tree diagrams with marks that indicated the root node and three target nodes. Stimuli were composed of three tree layouts: traditional, orthogonal, and radial. The layouts differ in how nodes and edges are aligned. These layouts are shown in Figure 4.6. The participants were asked to find the *least common ancestor* (LCA) of the target nodes in each tree. The instructions included a definition of the LCA written in plain English.

We asked participants to report the coordinates of the LCA for each tree they analyzed, so we added an interaction to the Fauxvea interface that lets users find the coordinates over the cursor location. While interacting with the interface, the user can type the 'Return/Enter' key to place a mark under the cursor and its (x, y) coordinates are displayed on the screen. In this way, participants can quickly find the coordinates of locations in the image without interfering with cursor presses.

4.5.2 MTurk Study

We created three different HITs on MTurk corresponding to the three tree layouts and recruited 85 Turkers to complete each. We restricted each participant to one HIT only because the same underlying graph data is visualized in each HIT. All participants were located within the United States.

In each HIT, participants performed three training tasks using Fauxvea and were shown example trees with the LCA labelled to help them understand the task. For the fourth task, participants completed the task for the test stimulus that was replicated from the Burch et al. study. The test stimulus did not have the LCA labelled. During each HIT, participants could advance to the next image in the sequence after any amount of time by providing and answer to the question and clicking a button on the webpage. They were not allowed to revisit past images after moving on. We expected that each HIT would take 4-5 minutes to complete and paid each Turker who completed

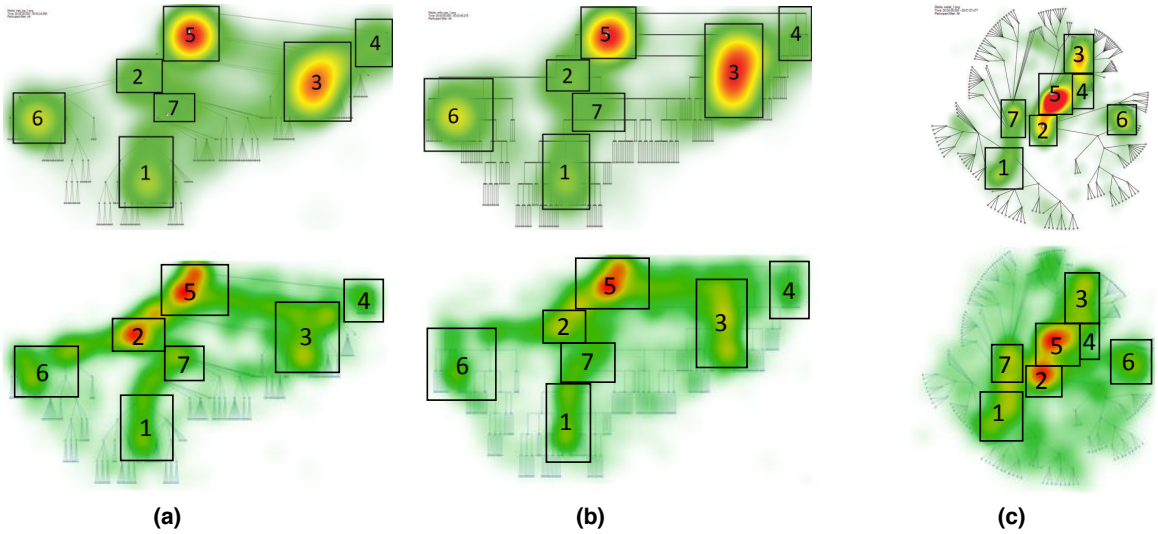


Figure 4.7: Comparison of results from Experiment 3 with the results reported by Burch et al. Data in columns (a), (b), and (c) correspond to traditional, orthogonal, and radial layout conditions. Rows 1 and 2 show the eye-tracking heatmaps from Burch et al. and the gaze estimate heatmaps we collected in Experiment 3, respectively.

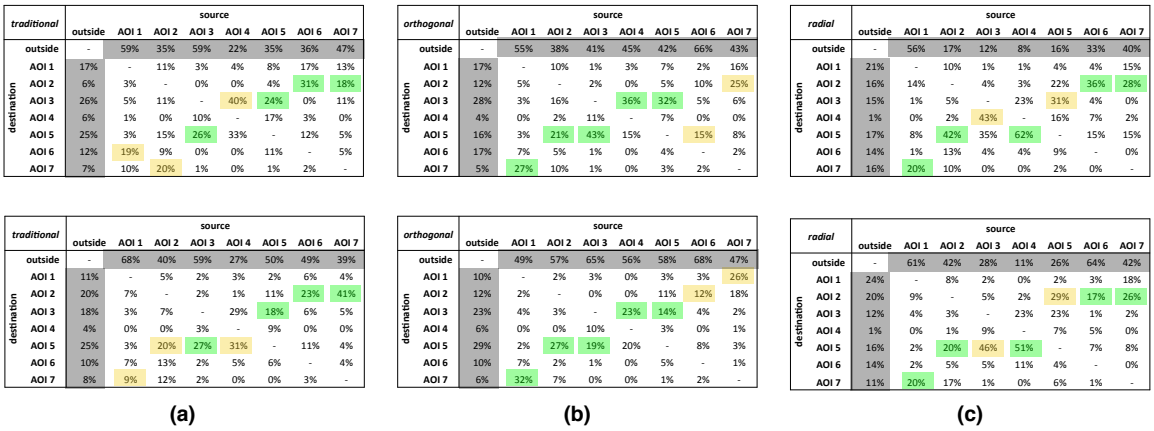
a HIT $0.45. A $0.15 bonus was offered to each Turker who identified the LCA correctly according to an expert reviewer.

After the visual analysis tasks, we collected demographic information about participants' age, sex, and cursor device (mouse, trackpad, or other), as well as how often they look at images like these ("never", "sometimes", "often") and general feedback about the strategy each Turker completed the LCA task.

4.5.3 Results

For each HIT, 85 Turkers performed the LCA task using the Fauxvea interface. Each of the test tasks was deemed accurate if the reported coordinates for the LCA were within 10 pixels of the known answer. Turkers who were accurate were given a $0.15 bonus. The average task accuracy for the HITs differed: Turkers were most accurate with the orthogonal layout (50.6% = 43/85), slightly less accurate with the traditional layout (41.2% = 35/85), and least accurate with the radial layout (20% = 17/85).

Overall, we found that the distributions of fixations from Experiment 3 on the three stimuli were similar to those published in Burch et al. [14]. In Figure 4.7, we show heatmaps of fixations from Experiment 3 alongside those from the original study. The top row shows fixation heatmaps and the AOIs specified in the original study. The second row shows heatmaps we generated from the data collected in Experiment 3. We implemented a heatmap renderer using a rainbow color map to approximate the visualization technique in the earlier work; therefore, some visual differences between these charts may be due to implementation differences.



traditional

destination \ source	outside	AOI 1	AOI 2	AOI 3	AOI 4	AOI 5	AOI 6	AOI 7
outside	-	59%	35%	59%	22%	35%	36%	47%
AOI 1	17%	-	11%	3%	4%	8%	17%	13%
AOI 2	6%	3%	-	0%	0%	4%	31%	18%
AOI 3	26%	5%	11%	-	40%	24%	0%	11%
AOI 4	6%	1%	0%	10%	-	17%	3%	0%
AOI 5	25%	3%	15%	26%	33%	-	12%	5%
AOI 6	12%	19%	9%	0%	0%	11%	-	5%
AOI 7	7%	10%	20%	1%	0%	1%	2%	-

orthogonal

destination \ source	outside	AOI 1	AOI 2	AOI 3	AOI 4	AOI 5	AOI 6	AOI 7
outside	-	55%	38%	41%	45%	42%	66%	43%
AOI 1	17%	-	10%	1%	3%	7%	2%	16%
AOI 2	12%	5%	-	2%	0%	5%	10%	25%
AOI 3	28%	3%	16%	-	36%	32%	5%	6%
AOI 4	4%	0%	2%	11%	-	7%	0%	0%
AOI 5	16%	3%	21%	43%	15%	-	15%	8%
AOI 6	17%	7%	5%	1%	0%	4%	-	2%
AOI 7	5%	27%	10%	1%	0%	3%	2%	-

radial

destination \ source	outside	AOI 1	AOI 2	AOI 3	AOI 4	AOI 5	AOI 6	AOI 7
outside	-	56%	17%	12%	8%	16%	33%	40%
AOI 1	21%	-	10%	1%	1%	4%	4%	15%
AOI 2	16%	14%	-	4%	3%	22%	36%	28%
AOI 3	15%	1%	5%	-	23%	31%	4%	0%
AOI 4	1%	0%	2%	43%	-	16%	7%	2%
AOI 5	17%	8%	42%	35%	62%	-	15%	15%
AOI 6	14%	1%	13%	4%	4%	9%	-	0%
AOI 7	16%	20%	10%	0%	0%	2%	0%	-

traditional

destination \ source	outside	AOI 1	AOI 2	AOI 3	AOI 4	AOI 5	AOI 6	AOI 7
outside	-	68%	40%	59%	27%	50%	49%	39%
AOI 1	11%	-	5%	2%	3%	2%	6%	4%
AOI 2	20%	7%	-	2%	1%	11%	23%	41%
AOI 3	18%	3%	7%	-	29%	18%	6%	5%
AOI 4	4%	0%	0%	3%	-	9%	0%	0%
AOI 5	25%	3%	20%	27%	31%	-	11%	4%
AOI 6	10%	7%	13%	2%	5%	6%	-	4%
AOI 7	8%	9%	12%	2%	0%	0%	3%	-

orthogonal

destination \ source	outside	AOI 1	AOI 2	AOI 3	AOI 4	AOI 5	AOI 6	AOI 7
outside	-	49%	57%	65%	56%	58%	68%	47%
AOI 1	10%	-	2%	3%	0%	3%	3%	26%
AOI 2	12%	2%	-	0%	0%	11%	12%	18%
AOI 3	23%	4%	3%	-	23%	14%	4%	2%
AOI 4	6%	0%	0%	10%	-	3%	0%	1%
AOI 5	29%	2%	27%	19%	20%	-	8%	3%
AOI 6	10%	7%	2%	1%	0%	5%	-	1%
AOI 7	6%	32%	7%	0%	0%	1%	2%	-

radial

destination \ source	outside	AOI 1	AOI 2	AOI 3	AOI 4	AOI 5	AOI 6	AOI 7
outside	-	61%	42%	28%	11%	26%	64%	42%
AOI 1	24%	-	8%	2%	0%	2%	3%	18%
AOI 2	20%	9%	-	5%	2%	29%	17%	26%
AOI 3	12%	4%	3%	-	23%	23%	1%	2%
AOI 4	1%	0%	1%	9%	-	7%	5%	0%
AOI 5	16%	2%	20%	46%	51%	-	7%	8%
AOI 6	14%	2%	5%	5%	11%	4%	-	0%
AOI 7	11%	20%	17%	1%	0%	6%	1%	-

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

Figure 4.8: Comparison of results from Experiment 3 with the results reported by Burch et al. Data in columns (a), (b), and (c) correspond to traditional, orthogonal, and radial layout conditions. Rows 1 and 2 show transition probabilities between AOIs from Burch et al. and the probabilities we found in Experiment 3, respectively. Transition probabilities to or from areas outside any AOI are grayed out. Green cells indicate where the most likely destination AOI from a source is the same in both the eye-tracking results and the Experiment 3 results. Yellow cells indicate where the most likely destination AOI from a source was not the same in both eye-tracking and Experiment 3 results.

We observed several similarities between the heatmaps from the original study and Experiment 3. In all heatmaps, AOI 5 – which contains the root node of the tree – is fixated on heavily. This makes sense because locating the root node is critical for the task of finding the LCA of the target nodes. We also found subtrees and leaf nodes that were essentially ignored in both the original study and in Experiment 3. These include areas that are dense with nodes and edges but are not parts of the visualization that one must attend to find the LCA (e.g., between AOI 1 and AOI 3 in the traditional and orthogonal layouts). This suggests Turkers are able to focus on the task at hand and do not spend effort fixating on areas of the visualization that are irrelevant to the task, even if those areas are comparably visually salient in the blurred viewer. The fact that the heatmaps are similar between studies, and that they show evidence participants fixate on task-specific areas, supports both **H1a** and **H1b**. We did not evaluate **H2** for Experiment 3 because the raw eye-tracking data needed to compute quantitative distance scores were not available.

We also observed some differences between the fixation maps from these experiments. In general, the original eye-tracking fixations appear more focused and less spread out than the Fauxvea fixations. This contrasts our earlier findings in Experiment 1. An exception to this is the set of Fauxvea fixations that occur along edges in the traditional and orthogonal layouts (e.g., between AOI 2 and AOI 6, and between AOI 1 and AOI 7). Another noticeable area where heatmaps differ in the traditional layout is AOI 2, which is fixated on relatively more than AOI 3 in our experiment and less than AOI 3 in the original experiment. In this layout, Turkers were much more likely to transition from AOI 7 to AOI 2 than to any other AOI; in the original eye-tracking study, AOI 2 was also the most frequent transition from AOI 7, but AOI 1 and AOI 3 are other common destinations

(each with > 10% frequency).

All observed transition frequencies, which can be thought of as probabilities, between specific AOIs are shown in Figure 4.8. The top row shows probabilities from the original eye-tracking study, and the second row shows probabilities from the data in our experiment. The similarity in transition tables between experiments suggest that participants explored the AOIs using similar patterns. For 14 out of the 21 source AOIs in the three layouts (67%), the most frequent destination AOI (highlighted in green) was the same in both the original results and our results. This is much better than the 3 or 4 matches (16.7% = 1/6) we expect if the most frequent destination from each AOI were randomly chosen from the remaining six. Therefore, we find support for **H3**. The cells highlighted in yellow show where the most likely destination AOI is different between the experiments. In all but one of these cases, the most likely transition in our experiment was the second most likely in the original experiment.

Examining the transition probabilities, we did not find support for **H4a**. Burch et al. found that the probability from AOI 1 to AOI 6 (and vice versa) was 19% (17%) for the traditional layout, in contrast to 7% (6%) in our study, which is comparable to the orthogonal layout results from both studies. Transition probabilities for these AOIs in the radial layout are comparable between studies.

We found partial support for **H4b**. We re-examined the transitions that supported this hypothesis in the original study. The Fauxvea transition probability from AOI 1 to AOI 3 (and vice versa) is less than the probability from AOI 1 to AOI 6 (and vice versa) in the traditional (3% (2%) compared to 7% (6%)) and orthogonal (4% (3%) compared to 7% (3%)) layouts. The distance between AOIs 1 and 3 is greater than the distance between AOIs 1 and 6. However, for the radial layout, we found that the probability from AOI 1 to AOI 3 (and vice versa) was not necessarily less than the probability from AOI 1 to AOI 6 (and vice versa), as in the eye-tracking study: 4% (2%) compared to 2% (3%). In fact, the distances from AOI 1 to AOI 3 and AOI 6 are not as different in the radial layout and in the traditional and orthogonal ones (see Figure 4.6). We discuss possible explanations for these differences in Sec. 4.6.1.

Finally, we found strong support for **H4c**. In our experiment, the probability from AOI 5 to AOI 2 is 29% in the radial layout but only 11% in both traditional and orthogonal layouts. This is comparable to Burch et al.'s probabilities for this transition: 22% (radial), 4% (traditional), and 5% (orthogonal).

4.6 Discussion

In this section, we discuss how Fauxvea might affect visual exploration behaviors, then we outline opportunities for improving Fauxvea and quantitative comparison methods for gaze estimates.

4.6.1 Visual Exploration Behaviors

We noticed a few differences in how eye-tracking and Fauxvea fixations are distributed spatially for visualization tasks. In Experiment 1, we found that eye-tracking gazes generally occur over wider

regions of the image and appear more spread out than Fauxvea gaze estimates (see Figure 4.2). There are several possible explanations for this behavior:

- People do not look at a singular point of interest for long; instead, their gaze hovers around that point.
- Holding a cursor at a single pixel location over time requires less effort than gazing at one location for the same amount of time.
- The time and effort needed to move and press the cursor is greater than a saccade of the eye.
- For Turkers, there is an opportunity cost to being slow or getting distracted by irrelevant image details. We expected that Turkers would finish the HITs for this experiment as quickly as possible in order to accept new HITs and to maximize their payments on MTurk.
- Eye trackers can have errors due to both calibration and moments when the eye tracker cannot find the eye. Therefore, recorded coordinates might be inexact.

Our findings in Experiment 3, which involves a more complex task, show a different pattern: in some cases, Fauxvea gaze estimates are more spread out than eye-tracking gazes (see the radial layout in Figure 4.7). It is possible that Turkers using the Fauxvea interface had less experience with this task compared to the participants in the eye-tracking study and therefore spend more time exploring the diagram. Turkers also used the interface to fixate on edges in the tree diagrams more than eye-tracking participants, which supports the idea that they focus on tracing paths to complete the task. Eye-tracking participants with the normal viewer, on the other hand, can make saccades between nodes and may rely on peripheral vision to view edges. In both populations, similar hot spots related to the task appear in the heatmaps.

4.6.2 Quantitative Comparisons

In Experiment 1, we compared Fauxvea fixations against eye-tracking fixations using a quantitative approach similar to an earlier evaluation by Rudoy et al. [94]. We evaluated an additional distance function and tested additional baseline distributions of random fixations, plus a grid baseline and a saliency model. We discuss our findings below.

Distance Metrics

We explored two measures of similarity between smoothed representations of gaze locations: a symmetric version of KL divergence and χ^2 distance. Figure 4.4 shows one of these metrics (χ^2) computed between the ET and MT data we collected, for each pair of stimuli-tasks. As a sanity check, we were interested in seeing how values on the diagonal – which are distances between corresponding visualization tasks in the two conditions – compare to distances between unrelated visualization tasks. We note that this type of matrix should not necessarily be symmetric across the diagonal, because the columns (Fauxvea) represent a different modality for which fixations were collected

compared to the rows (eye tracking). The matrix shows that the diagonal is in fact darker than any single row. This is also visual evidence of hypothesis for **H1b**: the fixations from both ET and MT are linked to underlying visualization tasks.

Generating Baseline Gazes

In this work, we evaluated Fauxvea quantitatively by computing how much closer to eye tracking Fauxvea's fixation locations are compared to fixations drawn from null distributions that represent where people might look without regard to the visualization task. Creating realistic computational models of gaze during visualization tasks is an open problem. Task-aware models could replace the need for gaze-estimation methods with humans in the loop, or provide stronger baselines for evaluating new estimation methods.

We used an off-the-shelf, state-of-the-art model of visual saliency [65] and found that the maps it generates from the visualization stimuli in Experiment 1 are not much closer to eye tracking than the other null distributions, like Grid and Centered Gaussian. This is not surprising because people do not necessarily attend to salient regions that are irrelevant to the analysis task they are given. It is possible that people with experience in vision and eye tracking could identify where people will look during tasks, but as we found in Experiment 2, self-assessment of these areas is not consistent even among experts.

4.6.3 Limitations

Imposing on the User

Many challenges remain in the area of estimating gaze patterns without eye trackers. Interactive visualizations are difficult to study with a RFV-based method because the viewer requires cursor interactions that might conflict with underlying interactions. For example, the original method to let users report the LCA in Experiment 3 was to click the node, but this would log an additional fixation. Another issue is that users could accidentally select a node (e.g., on the first cursor press) before focusing the image, resulting in an incorrect response. We designed a keyboard interaction that lets the user find and report the coordinates of a node without extra cursor presses, but for more complex interactions a similar workaround might not be possible. The process of mixing focus interactions and interactions with the underlying visualization might obstruct the user's natural analysis workflow. For these reasons, we have focused on evaluating gaze locations in static visualizations only.

Another limitation of using a RFV-based interface for visualization is that it could discourage users from participating in the evaluation. Bednarik and Tukiainen [7] reported that some users disliked the RFV interface. With Fauxvea, one mitigating factor is the large number of potential participants on MTurk: even if some Turkers decide to avoid Fauxvea HITs after participating once, there are many other workers to recruit. In spite of this, we received positive feedback from Turkers who completed the HITs.

Finally, using any form of eye tracking or gaze estimation to understand a person's cognitive

activities with a visualization depends on the "eye-mind" hypothesis – that what a person gazes indicates her foremost cognitive process [66, 87]. Visualization analysis often requires keeping several pieces of information in mind while solving a task, so it is possible the hypothesis does not hold for some tasks, as Kim et al. found [68]. In this case, it is still valuable to understand which visual information is inspected and needed to answer a task, even if it is difficult to infer deeper cognitive processes of the person. Follow-up interviews or questionnaires could help verify cognitive processes that are apparent in gaze traces.

Supporting Large Images or Longer Tasks

The experiments we performed do not evaluate how the size of static visualizations affects the number of cursor interactions users make, or how task length and cognitive load affect cursor interactions. Large visualizations might require interactions in the browser, like scrolling. Similarly, longer tasks or tasks with more sub-parts might require more effort. We demonstrated a real evaluation scenario in Experiment 3, but studying how more cognitive work and larger image content impacts Fauxvea users would establish broader external validity of the method.

Eye-Tracking Metrics

Our work focused on validating a method that lets others collect gaze-estimate data and gaze-related metrics for visualization use. Poole and Ball summarized three types of eye-movement metrics [87]: fixation-derived, saccade-derived, and scan-path-derived. Of these, our results suggest that Fauxvea fixations can be used to compute fixation-derived metrics. The Fauxvea interface cannot estimate saccades, and more work is needed to validate that Fauxvea scan paths are similar to eye-tracking scan paths.

One must be careful when interpreting metrics involving individual Fauxvea fixations, since Fauxvea fixations might be more coarse-scale than eye-tracking fixations. We found that the duration of a Fauxvea fixation is on average longer than an eye-tracking fixation. One explanation for this is that when a person examines an area of a visualization in a normal image viewer, she might make several fixations near the same area; in Fauxvea, this might be replaced by a single, longer fixation due to the added cost of refocusing, as described in Sec. 4.6.1.

4.6.4 Opportunities

We found several opportunities to build on the Fauxvea method and related approaches. In general, these include collecting and analyzing data at a greater scale, including scan paths and data from remote experts; using data collected by the Fauxvea interface to improve analysis for future users; and using data collected by the Fauxvea interface to develop and train models of gaze without humans in the loop.

Validating Fixation Sequences

Analyzing fixation sequences, or *scan paths*, from Fauxvea compared to eye tracking can validate or disconfirm that Fauxvea can predict temporal aspects of gaze. Using crowdsourced workers for Fauxvea tasks, interacting in uncontrolled computing environments, could have a potentially large impact on how well Fauxvea scan paths and eye-tracking scan paths align temporally. For example, an individual on MTurk could switch to a new browser tab and check his or her email during the middle of a Fauxvea task. Detecting and controlling for these behaviors systematically will require appropriate protocols. It is also possible that unintentional 'time warping' could occur for Turkers who never get distracted but have slow or faulty browser responses.

We have begun to analyze the temporal aspects of fixations from Turkers completing tasks with Fauxvea. In Sec. 4.5, we identified the frequencies at which Turkers transitioned between areas of interest in the visualization, and found these were similar to transitions collected in an eye-tracking study. Finding patterns in sequences of transitions is valuable, but aligning sequences between participants who are faster or slower requires careful treatment. In a preliminary analysis, we clustered Fauxvea and eye-tracking sequences from Experiment 1 using dynamic time warping (DTW) and found some evidence that scan paths from both conditions cluster together on the same task. However, these results were not conclusive because of the small sample size of eye-tracking scan paths. More work is needed to evaluate Fauxvea scan paths quantitatively.

Facilitating Analysis Using Data from Previous Users

Showing visualization end users the data collected from earlier Fauxvea evaluations could facilitate analysis behaviors. For instance, displaying to a domain expert the regions of a visualization that were salient to Turkers could help the expert identify areas that might be overlooked by others, and therefore support quality control during analyses. Another scenario involves training new users of a visualization. Displaying the salient regions or 'search trails' from earlier users could guide a person to understand the important components of an unfamiliar visual representation or task.

Improving Baselines for Models for Gaze with Training Data

We believe the data collected from Fauxvea experiments could be used to improve the baselines mentioned in Sec. 4.6.2 and create computational models of gaze for visualization. In turn, these models could bootstrap the evaluation of gaze-estimation user interfaces that have humans in the loop, like Fauxvea. If a computational model of gaze is created that can generate fixations that are indistinguishable from those produced by Fauxvea or by eye tracking, then Fauxvea and eye tracking will no longer be needed to predict gaze during visualization tasks.

Identifying Features of Focus-Window Movement that Improve Fauxvea-Gaze Prediction Accuracy

Other features of how Turkers use Fauxvea and move the focus window could be used to improve the accuracy of predicting gaze. In our experiments, we predict gaze fixations precisely at the center of the focus window. Individual style and other factors could account for differences between the focus-window location and where a person looks (within or outside that window), as Huang et al. found when comparing gaze to cursor position [48]. Classifying patterns in how people move the focus window, e.g., in a scan-path analysis, could also be useful in predicting analysis styles and performance on future tasks, as Brown et al. found in a map-search task performed by Turkers using zoom and pan controls [13].

Another interesting question is whether certain types of visualizations result in Fauxvea estimates that are close to eye tracking, while other types of visualizations do not. While we did not evaluate this directly, we found that different null distributions of gazes (described in Sec. 4.3.5) were closer or farther from eye tracking depending on the visualization types, as highlighted in Table 4.2. It is possible that visualization type could affect the closeness of Fauxvea estimations and eye-tracking fixations. Anecdotally, we found that visually dense visualization types, like the node-link diagrams used in Experiment 1, appear to have noisier Fauxvea estimates compared to visualizations where task-relevant marks are separated from other distractor marks, e.g., bar charts. An open question is whether the parameters of the focus window in Fauxvea – including blur level, how blur degrades, and the blur radius – could be optimized for the intended visualization type.

Evaluating Gazes from Remote Domain Experts

In addition, we have not yet evaluated Fauxvea using remote visualization-domain experts. Collecting rich empirical evidence about visualization use from remote experts remains an important challenge for visualization researchers [11]. Using Fauxvea, it might be possible for visualization developers to collect gaze estimates from experts that are otherwise inaccessible for traditional eye-tracking lab studies when the developers and experts cannot be co-located. Fauxvea could enable gaze-related visualization studies with larger sample sizes of domain experts than is currently possible.

4.7 Concluding Remarks

We developed and evaluated a crowd-powered method called Fauxvea that estimates gaze fixations for analysis tasks with static information visualizations without using an eye tracker. This work was adapted from an earlier method that had not been previously evaluated with online crowd participants or in the context of information visualization.

We ran three experiments to evaluate the method, including a reproduction of earlier eye-tracking findings about tree visualizations. In Experiment 1, we found quantitative and qualitative evidence

that Fauxvea fixations from many Turkers are similar – and often less noisy – compared to fixations collected in a parallel eye-tracking study with a typical number of participants. In Experiment 2, we found that self-assessment of fixations by individuals can be inconsistent; therefore, using Fauxvea with crowdsourced workers is likely to be a more reliable approximation of eye tracking. In Experiment 3, we found that the way people transition their gaze between AOIs using Fauxvea is similar to eye tracking, but comparing full scan paths remains an open challenge. Our method is a practical alternative to using an eye tracker to find task-specific areas of interest in static visualizations. Creating a robust computational model of gaze for visualization tasks is an open problem, and data collected using our method might be helpful in constructing such a model.

This dissertation has shown that using novel, practical methods that model cognitive activities across a spectrum of high- to low-level processing (insight discovery, keystroke-level interactions, and gaze fixations) can provide empirical data that is useful for evaluating visualizations. In the next chapter, we discuss challenges and research opportunities to improve visualization evaluation methods using human modeling and by integrating these models more tightly into the visualization development cycle.

Chapter 5

Discussion and Conclusion

This dissertation has addressed the need for improved evaluation methods for visualization. Many typical evaluations of visualization designs and visual analysis systems use benchmark tasks that do not necessary reflect real usage scenarios, and therefore sacrifice ecological validity. In addition, experimental designs that involve characterizing visualizations only by the average accuracy and efficiency (response time) of study participants performing benchmark tasks usually reveal only indirect information about the cognitive activities of participants. Since the aim of many visualization tools is to promote the discovery of insights about the visualized data by facilitating reasoning, we focused on developing accessible methods that produce empirical data aligned with evaluating this criterion, including:

1. insights and insight-based metrics;
2. interaction storyboards that summarize how a population of end users completes analysis tasks and that are useful for predictive performance modeling;
3. gaze-location estimates as a person performs a visualization task.

We showed that our methods make it easier for evaluators to collect this data compared to existing approaches.

In the remainder of this chapter, we reiterate the main contributions of our evaluation methods and experiments. Next, we identify open research questions inspired by this work and propose opportunities to build upon these contributions. Finally, we conclude with a brief summary of the dissertation and its potential impact.

5.1 Summary of Primary Contributions

The main contributions of this dissertation are novel methods for evaluating visualizations, as well as case studies that demonstrate these methods with realistic visualization applications. The methods

were described make it easier to understand cognitive activities during user studies with visualizations. Minor contributions include specific findings from the case studies described in Chapters 2 and 4 regarding layout designs for node-link diagrams.

The LITE method presented in Chapter 2 lets evaluators assess a visualization system's ability both to promote exploration and support routine search tasks in a dataset. The protocol is within-subjects and, across visualization design conditions, it interleaves blocks of routine tasks with open-ended exploration periods, during which the insight-promoting characteristics of the system can be measured.

The Tome framework presented in Chapter 3 lets evaluators construct predictive human-performance models from interaction logs. It removes the need for an evaluator to apply the Keystroke-Level Model (KLM) by hand, which can be error-prone and time-consuming [56]. Instead, Tome uses an instrumentation library for UI widgets and can aggregate collections of interaction logs semi-automatically into project files for CogTool – a visual environment that simulates task executions using the KLM. We also demonstrated in a case study of brain-network diagrams that Tome is useful in predicting whether a proposed UI change would reduce task completion times.

The Fauxvea method described in Chapter 4 lets evaluators estimate where people fix their gaze on a static visualization during an analysis task using crowdsourced workers. It builds on the idea of a "restricted focus viewer" that estimates where people fixate in an image based on where they position a cursor-centered focus window over an otherwise blurred image. The main benefit of our approach is that it is more accessible to evaluators than performing a lab-based study with a hardware eye tracker, while providing similar empirical data. Our contributions include qualitative and quantitative evaluations of the gaze estimate data that show it is:

1. comparable to eye-tracking data for basic infovis tasks;
2. a better estimate of where people will look than predictions made by individuals with eye-tracking expertise;
3. viable for reproducing findings about visual attention on static visualizations that were originally obtained using eye tracking.

Together, these methods go beyond evaluations of visualizations that consider only average accuracy and efficiency of people performing benchmarks tasks, which is typical in visualization research. They make it easier to use alternative evaluations that depend on modeling expertise (Tome) or eye-tracking hardware (Fauxvea), and augment typical task-based evaluation with aspects of insight-based evaluation (LITE).

5.2 Research Opportunities and Directions

1. Effect of individual differences on visualization analysis performance

One benefit of the insight- and task-based evaluation we describe in Chapter 2 is that it gathers insight characteristics about different visualization conditions using the same study participants,

within-subjects. A challenge in between-subjects studies is that *how* one explores data might differ between individuals; breaking a study population into smaller groups for different conditions could result in differences in insight characteristics due to the groupings, not the underlying condition. Several studies have shown that individual differences between participants affect analysis behaviors or outcomes during interaction with computer information displays. Earlier in this dissertation, we described work by Huang et al. that found individual styles of using the cursor affected the alignment of a person's cursor position and gaze on search-engine results pages [48]. Personality traits that shape how a person views the world can influence the way they interact with information systems. For example, a person's sense of internal or external control over things, called locus of control [93], was shown to be correlated with her task performance using visual representations that depict a containment metaphor (e.g., treemaps [100]) [109]. In some cases, evidence of individual differences can be observed through interaction choices or strategies used during analysis. Brown et al. demonstrated this by inferring participants' personality traits using off-the-shelf machine learning techniques applied to interaction logs from a map analysis task [13].

Mapping out how individual differences affect interaction and analysis could significantly impact experimental design for visualization research. It is possible that controlling for individual differences in evaluations of visualizations could lead to findings about human visual-analysis performance that would otherwise be statistically insignificant in an uncontrolled population. Furthermore, controlling for individual differences is critical for protecting the generalizability of findings; participants in visualization and HCI research experiments at leading institutions may not be representative of wider populations [46].

Understanding individual differences will also be important for designing user interfaces for visualization systems that are "cognitively optimized" for the analyst. For instance, after assessing an analyst's personality, ability, or work style, features of the application that are tailored to the individual might be enabled. A simple example of this tailoring in the perceptual domain is a visualization that first applies a color vision test to the analyst, then chooses a color palette for the visualization that sidesteps any abnormalities found by the test.

2. Insight-based evaluation with remote expert participants

We presented methods that let evaluators capture evidence of cognitive activities that is useful in characterizing whether a visualization design or system is more effective than an alternative. Insight-based methods are particularly well-suited for quantifying the "aha!" moments of discovery that visualization tools enable, but these methods rely on an evaluator to code self-reported insights from participants; this is difficult in laboratory settings, and more so in remote settings where it is difficult for an evaluator to capture and record self-reported insights and interaction data, or ask targeted questions in a follow-up interview.

At the same time, enabling remote study participants in insight-based evaluations could have a major impact on visualization research. Existing evaluations with exploratory components, including ours (see Chapter 2) and previous studies [79, 95, 96], have included local participants with less

expertise than full-fledged domain experts in order to achieve reasonable sample sizes. Making these studies web-accessible could make it much easier to recruit a large population of experts from around the world – the same population of consumers for the tools being studied.

New software tools and experimental protocols could make insight-based evaluations easier to execute with remote users. First, there is a lack of integrated tools that record and synchronize data typically collected in insight-based evaluations, including: (1) screen-capture video, (2) video recording of participants, (3) audio recording for participant utterances, and (4) low-level interaction logs. Many tools to capture these data streams already exist, but there are few standards for encoding and synchronizing these data. For example, developing a standard for interaction logs that is application-agnostic will let evaluators compare visualization systems more easily. Taxonomies that describe top-level interactions and types of abstract analytical activities have been used as descriptive frameworks and to simplify data analyses [2, 45, 107]. We explore this in a follow-up study [34] we performed with the visual analytics application described in Chapter 2.

It is possible that a web-based platform for insight-based studies will present unexpected challenges for visualization and HCI researchers. For example, in Chapter 2 we found that converting the think-aloud protocol for reporting insights to one using web forms seemed to affect when participants felt "done" with exploration, despite similar instructions. One explanation is that a participant's motivation to continue open-ended exploration may be lessened when a proctor is absent. However, we believe the ability to recruit domain experts who may be intrinsically motivated mitigates this concern.

3. Predictive human performance modeling from interaction logs

Modeling human performance on visualization tasks remains an important challenge. The potential impact of evaluating human performance without having to run a user study with people is high: studies involving people are often costly and difficult to control, as we mentioned earlier in this chapter. With human models – which Bonnie John calls "cognitive crash dummies" [58] – evaluations of user interfaces and visualization tools can be run on demand, encouraging more rapid iterations through the design process. In contrast to simple predictive models like the KLM, which only models expert task executions on known interaction sequences, we believe cognitive crash dummies in the future need real interaction data from end users to simulate analysis behaviors for visualization.

Constructing task representations from unstructured interaction data is a step in this direction. We showed in Chapter 3 that *supervised* interaction logs, which are manually categorized by task, can be aggregated into storyboards representing expert task executions and simulated using the KLM. Earlier we referred to this as semi-automated because it requires a human to group logs manually based on the tasks attempted during the corresponding interactions. This is only reasonable when end users are given predefined tasks to complete (as in our case study with the brain-network diagram). Looking at many interaction logs "in the wild" and labeling the tasks is extremely difficult, even when logs formatted consistently and a set of supported tasks is known a priori. Therefore we need computational tools that are able to look at collections of logs and with minimal

human interaction answer the following:

- On a given log, when does one analysis task begin and end, and another starts?
- Is a sequence of interactions focused on a particular task, or just exploratory?
- Based on a collection of logs, what are the analysis tasks supported by the visualization?
- For an analysis task, what does a "good" execution look like, and what are noisy or error-prone executions?
- For an individual user, is she learning how to use the visualization and having better analysis outcomes over time?

4. Incorporating automated and semi-automated evaluation into the visualization design cycle

Our work is aimed at making effective evaluations of visualization tools easier so that the design process as a whole works better, resulting in more effective visualization artifacts. We have talked about automation and semi-automation of evaluations in Chapters 3 and 4. With more automated tools in development, an open question is: *What is an effective way to integrate automated evaluation into the design process?*

One direction is to leverage the fact that many visualization developers already use computational tools to track revisions in source code. Version control systems (e.g., Git, Mercurial, Subversion, CVS) are ubiquitous and can track code changes and, in effect, design changes for user interfaces and visualizations. These systems also support version tagging and branching, and hooks for running scripts on a server on predefined trigger events, like committing code changes. In some cases, developers use these hooks for running automated suits of unit tests, which determine whether units of functionality in an application are working correctly on test input.

We imagine adapting the model of unit testing on regular intervals or major design versions for visualization evaluation. In a "human unit testing" framework for visualization, an evaluator can run tests at the press of a button and later receive test results based on the performance of crowdsourced humans who are recruited on demand to use the visualization. Swearngin et al. applied a similar UI regression-testing approach using CogTool and the KLM to generate test cases and simulate tasks [104]. However, as we discussed in Chapter 3, the KLM can only predict expert completion times for tasks, and therefore cannot perform regression testing on features like task accuracy. Recently, Okoe and Jianu tested a more flexible system for getting feedback on graph layouts using crowdsourcing [80]. Building the software framework for launching crowdsourced task is feasible; in fact, the MTurk tasks described in Chapter 4 were posted by running a script we developed, and we experimented with installing this script to run when code changes were pushed to the Git repository with a specific log message. However, there are open research questions:

- What are the limitations of non-expert workers on various visualization analysis tasks, and should workers be screened before participating?

- What is the best way to aggregate the results of many workers completing an analysis task with a visualization, given that there may be a range of abilities?
- When re-running a specific protocol of human tests on design iterations of a single visualization, how do we control for learning effects that might occur if the workers participate multiple times?
- When re-running a specific protocol of human tests on two distinct visualizations that support similar analyses, how do we control for learning effects that might occur if the workers participate in both tests?

5.3 Visualization Evaluation in the Future

Advances in modeling human interactions and reasoning with information systems will have a great impact on how data visualizations are used and evaluated. For example, models of human perception and cognition could become so realistic they effectively replace people in user studies of visualization tools. "Cognitive crash dummies," as Bonnie John calls them, will not only enable studies that are more efficient than current user-study protocols that involve human participants, but they could be used to simulate visualization tasks on *descriptions* of visualization systems rather than working implementations of the systems. In other words, these models will inform the design of systems earlier in the design cycle (see Fig. 1.1) than what is typical today and reduce engineering costs as a result.

It is possible these advances will parallel the development of new machine-learning techniques and artificial intelligence systems, and that many scenarios in which we use currently visualization (e.g., interpreting MRI images for medical diagnostics) will be reliably answered by computers with minimal human decision making. Incidentally, as the design and evaluation of visualizations becomes easier and more effective by integrating human modeling capabilities, data visualization – as a technology that leverages human perceptual bandwidth – may be utilized less in favor of automated analysis systems that also benefit from advances in these capabilities.

The experiments in this dissertation demonstrate progress toward modeling some cognitive activities that occur during visualization analysis, from high-level insight discovery to low-level gaze movement during tasks. We have shown these simple models are helpful for evaluating visualization designs. However, integrating human modeling into visualization is still an early research direction, and building general purpose "cognitive crash dummies" that can help characterize differences in notorious visualization examples, like Anscombe's quartet [6], is a very distant goal. The importance of effective visualization tools is growing as the scale of datasets grows in critical domains like brain science, systems biology, and intelligence analysis. Designing these tools with require novel evaluation methods that are practical to use and incorporate aspects of analysts' cognition.

5.4 Summary

This dissertation investigated ways of collecting and using empirical data from visualization users about their cognitive processes for the purposes of evaluating visualizations and visualization systems. The methods were presented on a trajectory from coarse to fine-scale cognitive-motor behaviors: from modeling measurable insights after an analysts' tens-of-minutes-long exploration, to modeling keystroke-level interactions for search tasks on the order of tens of seconds to complete, to modeling gaze fixations on the order of fractions of a second without using an eye tracker.

The methods in this dissertation make it easier for visualization researchers and designers to evaluate the effectiveness of their visualizations empirically beyond basic accuracy and efficiency on benchmark tasks. These methods incorporate practical approaches for using insights, interactions, and gaze estimates as part of the evaluation criteria for visualizations. These methods were developed alongside new visualizations for brain-network data and spatiotemporal intelligence data, like geotagged microblogs. Ultimately we are interested in improving data analysis outcomes by designing better visualization tools. Evaluation is a critical part of that design process and will continue to be as data-driven applications become more prominent.

There are many opportunities to build on this work in visualization evaluation. We identified research directions that include:

1. clarifying how individual differences affect visual analysis behaviors and outcomes;
2. incorporating remote participants in visualization evaluations, including insight-based studies;
3. classification and clustering problems related to modeling tasks from unsupervised interaction data;
4. integrating evaluation more tightly with the visualization design and development process, so that useful evaluation data can be collected passively, automatically, or systematically at the press of a button.

Furthermore, we expect innovative research toward these directions will be applicable to areas of human-computer interaction and user-interface design outside of data visualization.

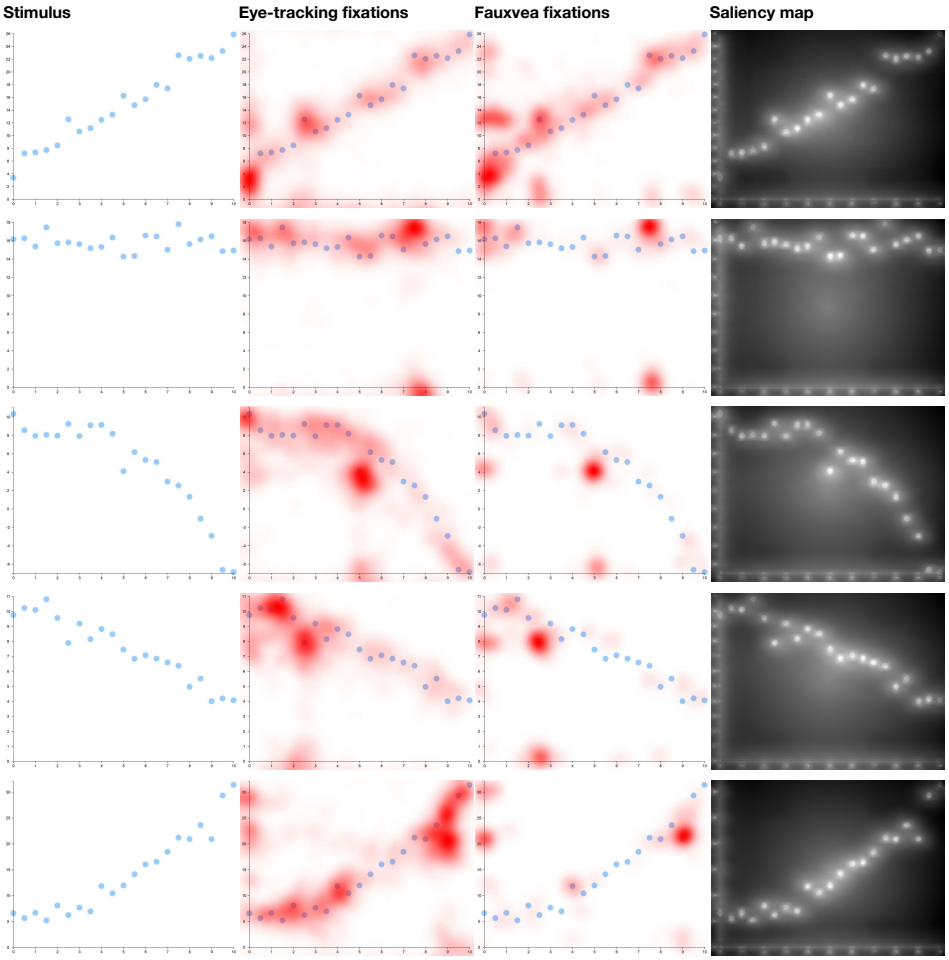
Appendix A

Gaze Location Estimates

This appendix includes visualization stimuli and tasks with corresponding eye-tracking heatmaps, crowdsourced gaze heatmaps, and saliency maps. See Section 4.3 for more details about this experiment.

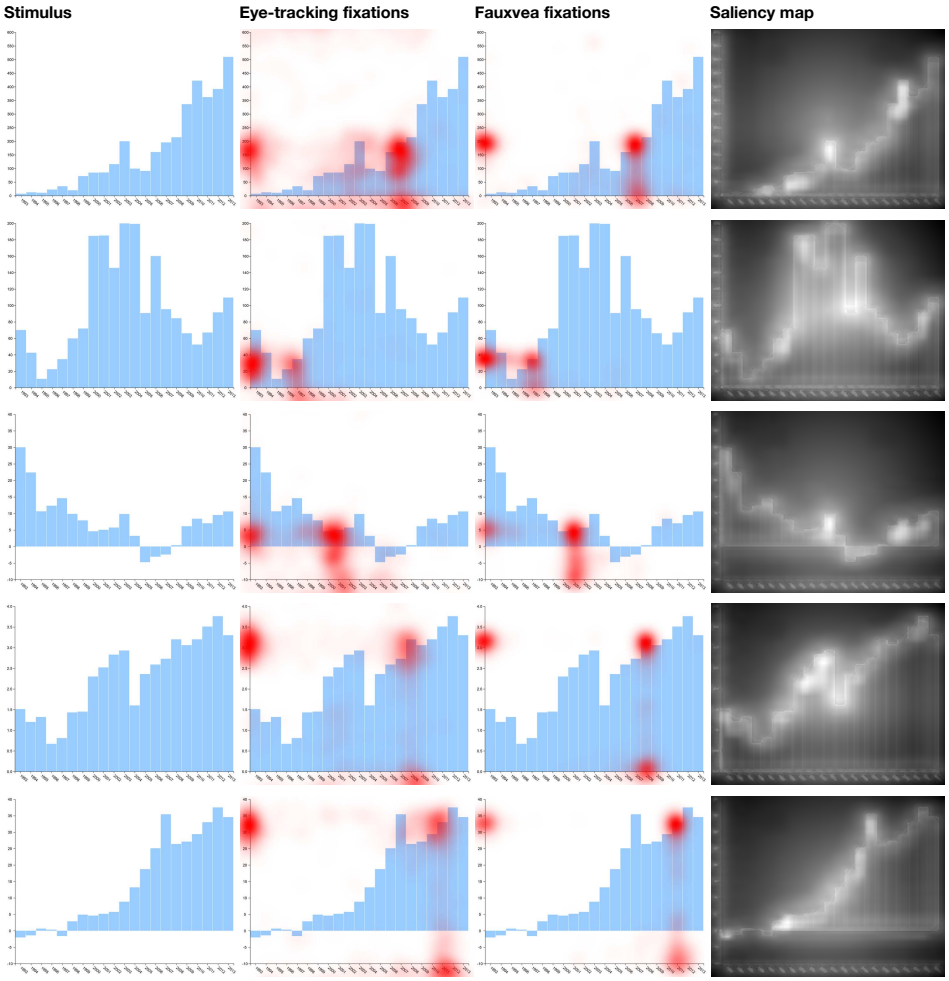
Scatter plots

Task: Report the (x, y) location of the point in the data that is the biggest outlier to the trend.



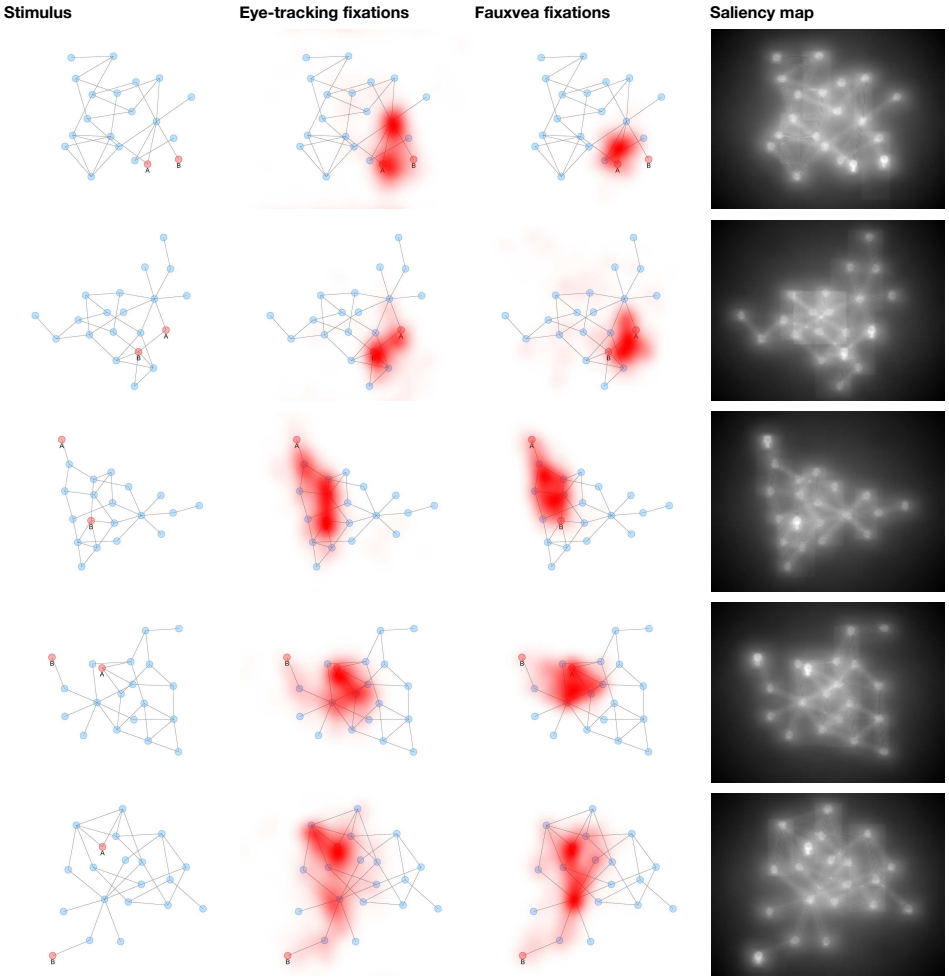
Bar charts

Task: Given a specific value in the domain, report the value (height) of the corresponding column.



Node-link diagrams

Task: Report the minimum number of edges needed to travel from highlighted node A to highlighted node B.



Photographs

Five photos were sampled from a large dataset of Flickr images that are freely available under a Creative Commons license.

Task: Report the average age in years of all people in the photograph.

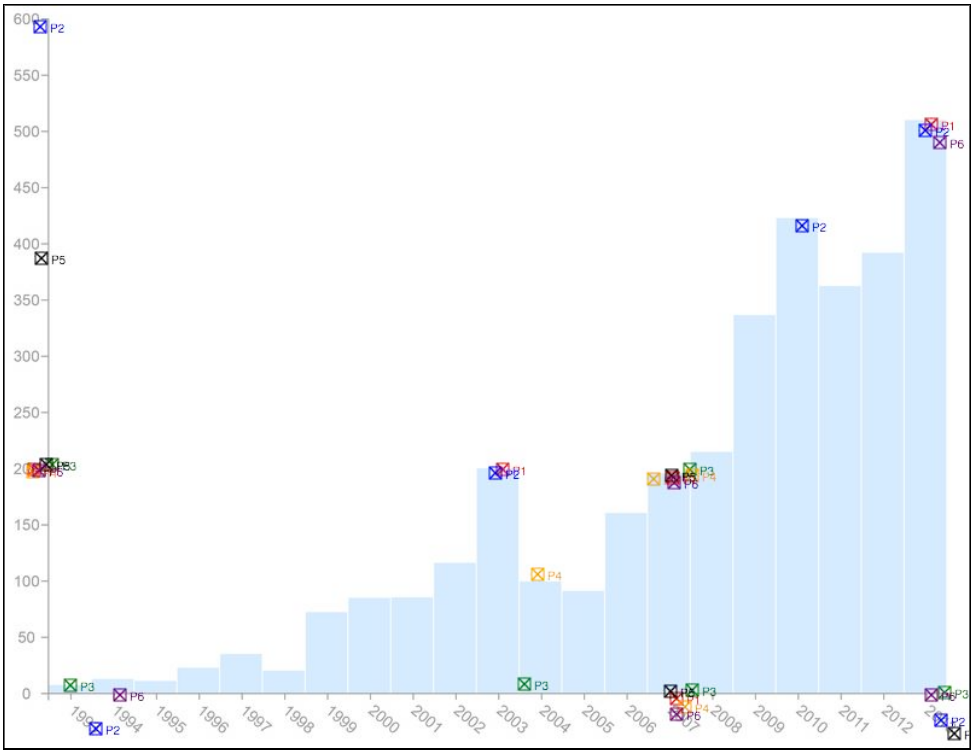
Appendix B

Expert Gaze Location Predictions

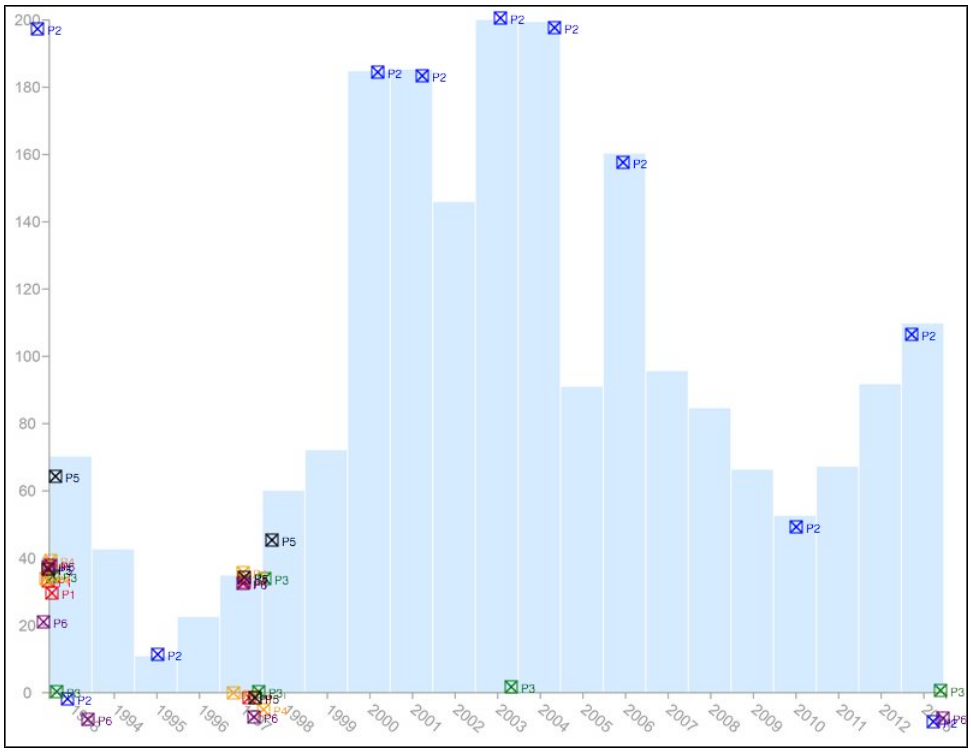
This appendix includes visualization stimuli and gaze location predictions made by individuals (labeled as P1–P6). See Section 4.4 for more details about this experiment.

Bar charts

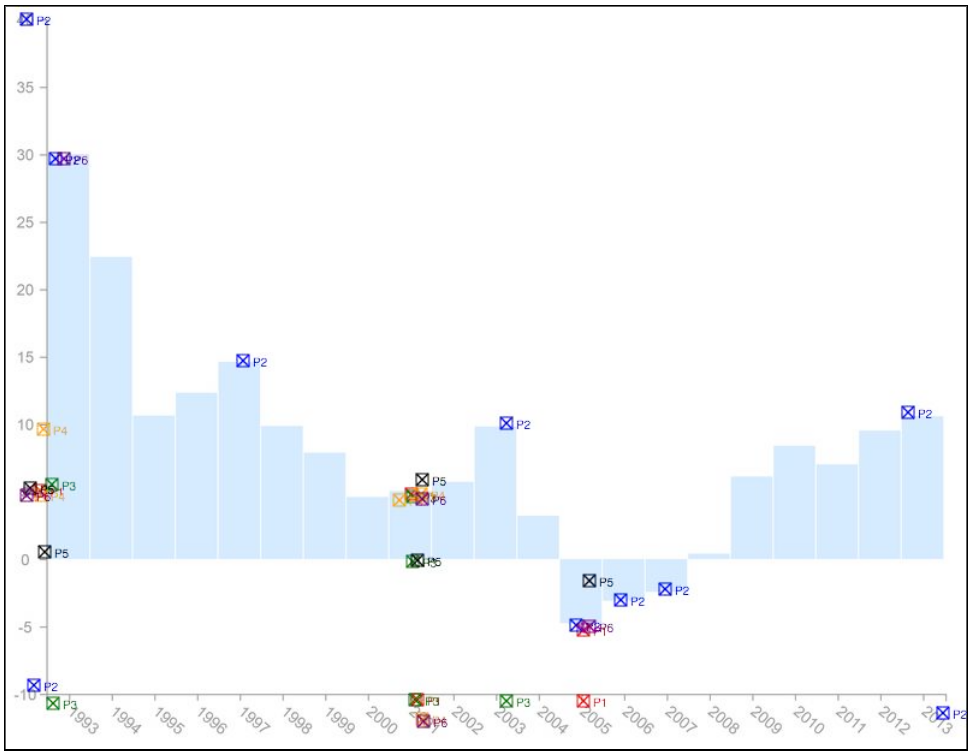
Task 1: Estimate the value (height) at year 2007.



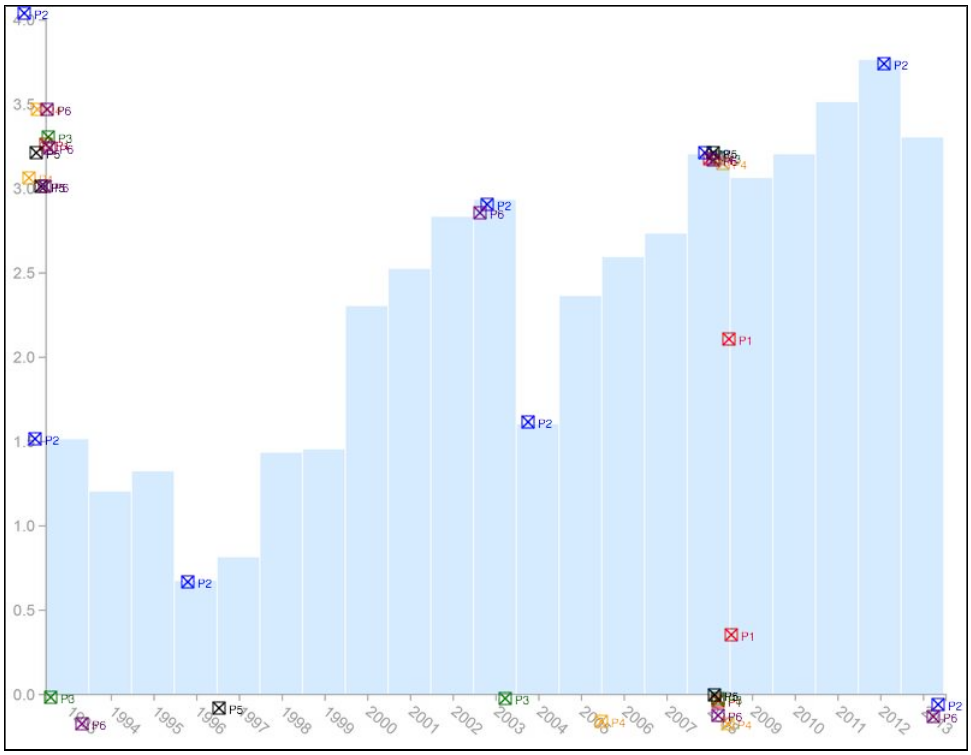
Task 2: Estimate the value (height) at year 1997.



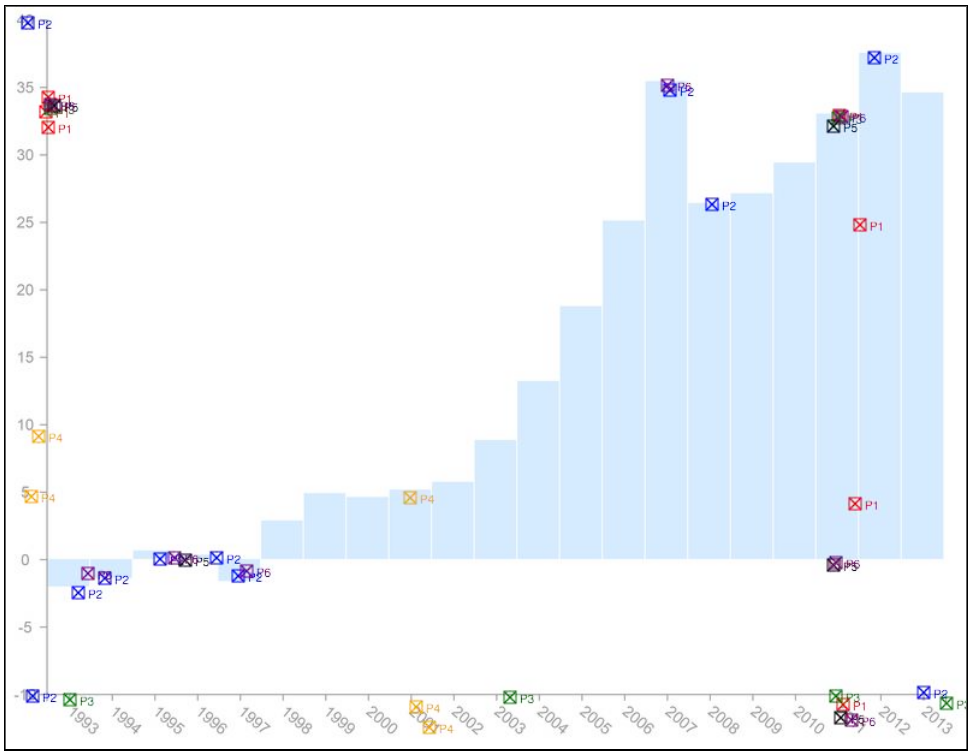
Task 3: Estimate the value (height) at year 2001.



Task 4: Estimate the value (height) at year 2008.

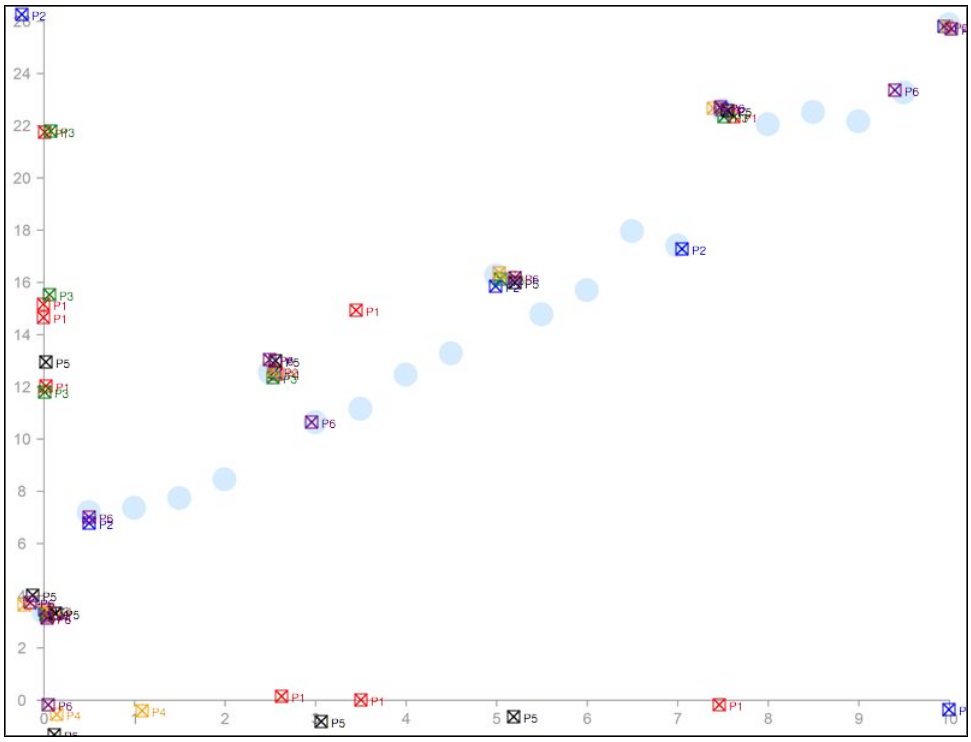


Task 5: Estimate the value (height) at year 2011.

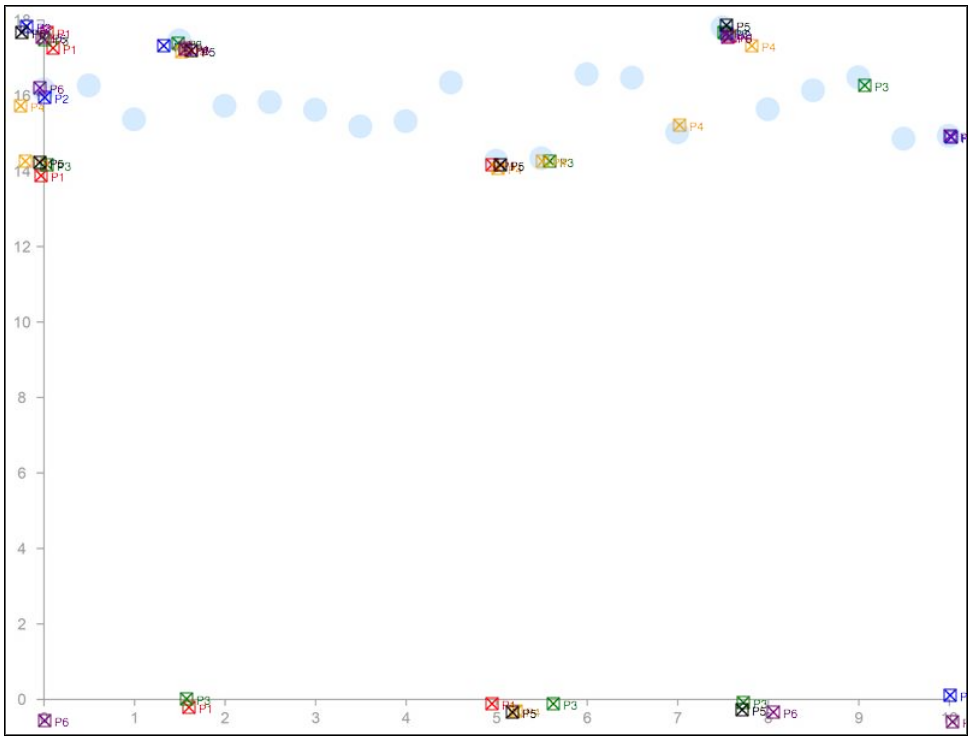


Scatter plots

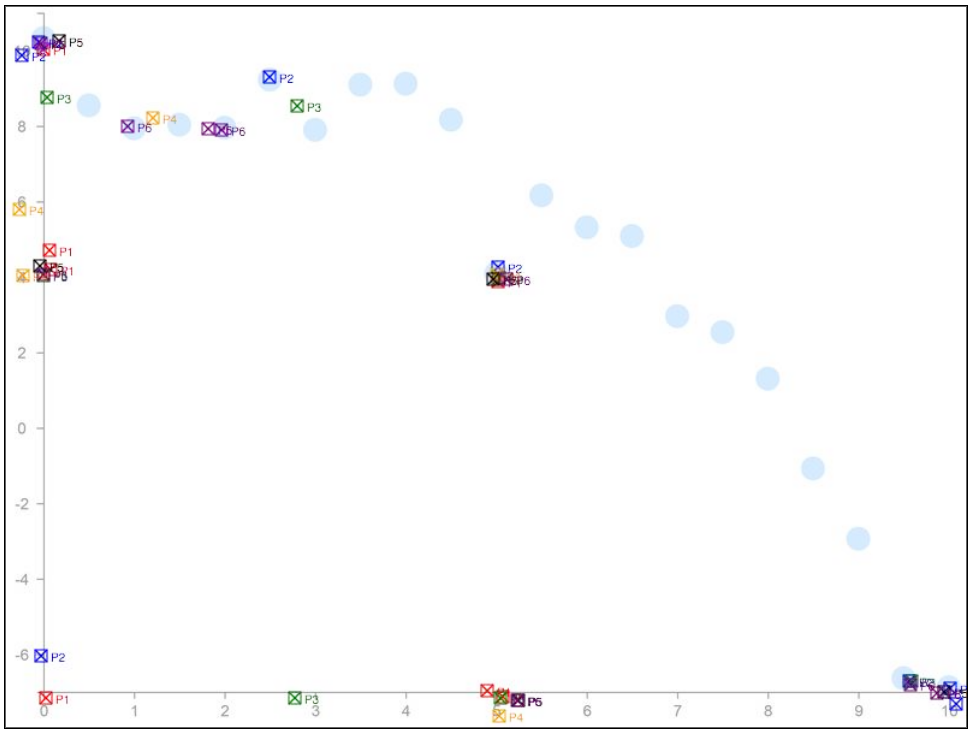
Task 6: Estimate the (x, y) position of the biggest outlier in this data trend. For example, "(3.5, 14.8)".



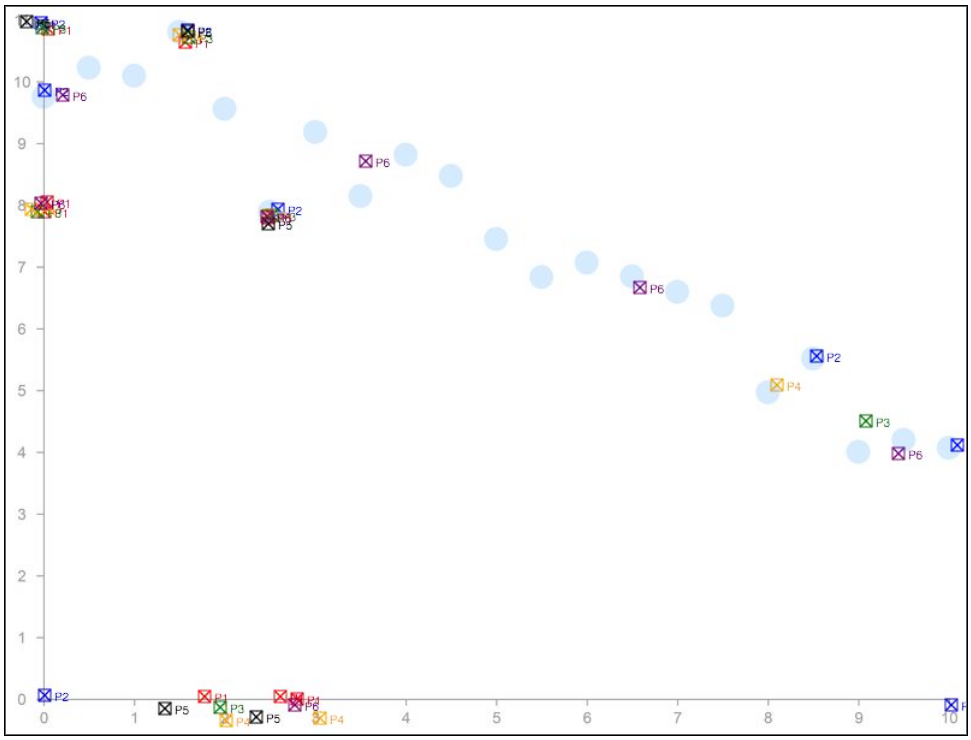
Task 7: Estimate the (x, y) position of the biggest outlier in this data trend. For example, "(3.5, 14.8)".



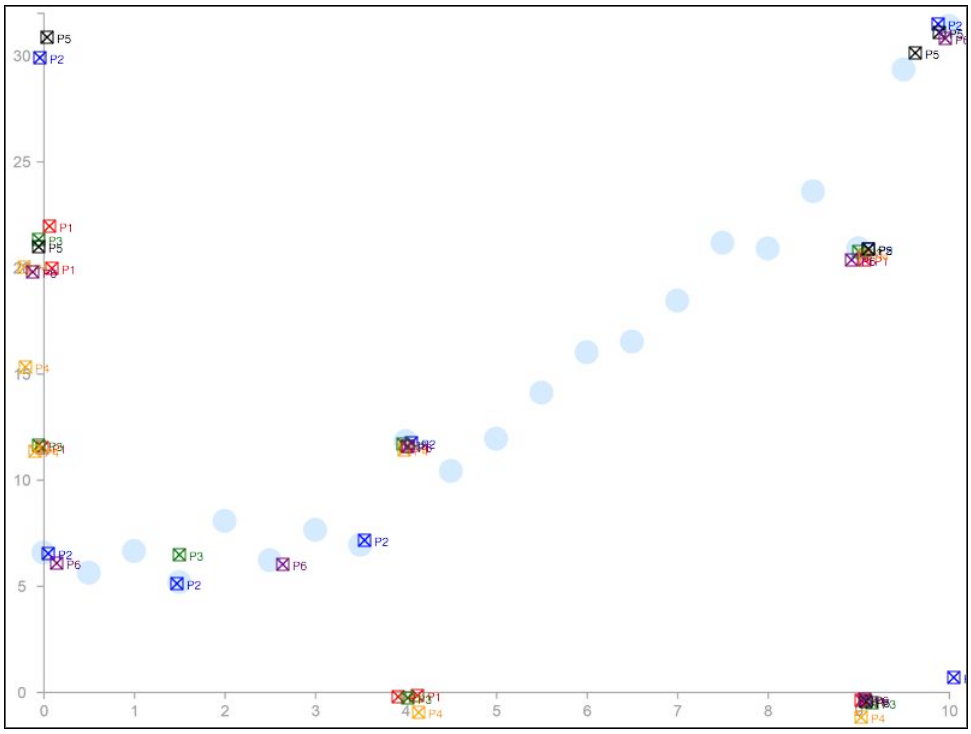
Task 8: Estimate the (x, y) position of the biggest outlier in this data trend. For example, "(3.5, 14.8)".



Task 9: Estimate the (x, y) position of the biggest outlier in this data trend. For example, "(3.5, 14.8)".

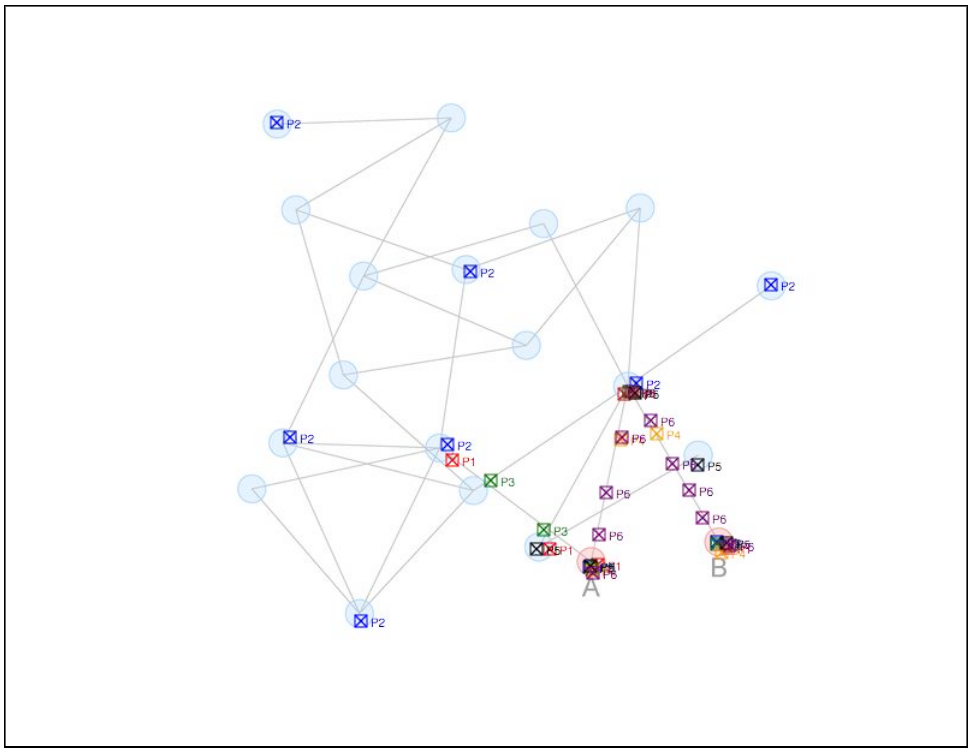


Task 10: Estimate the (x, y) position of the biggest outlier in this data trend. For example, "(3.5, 14.8)".

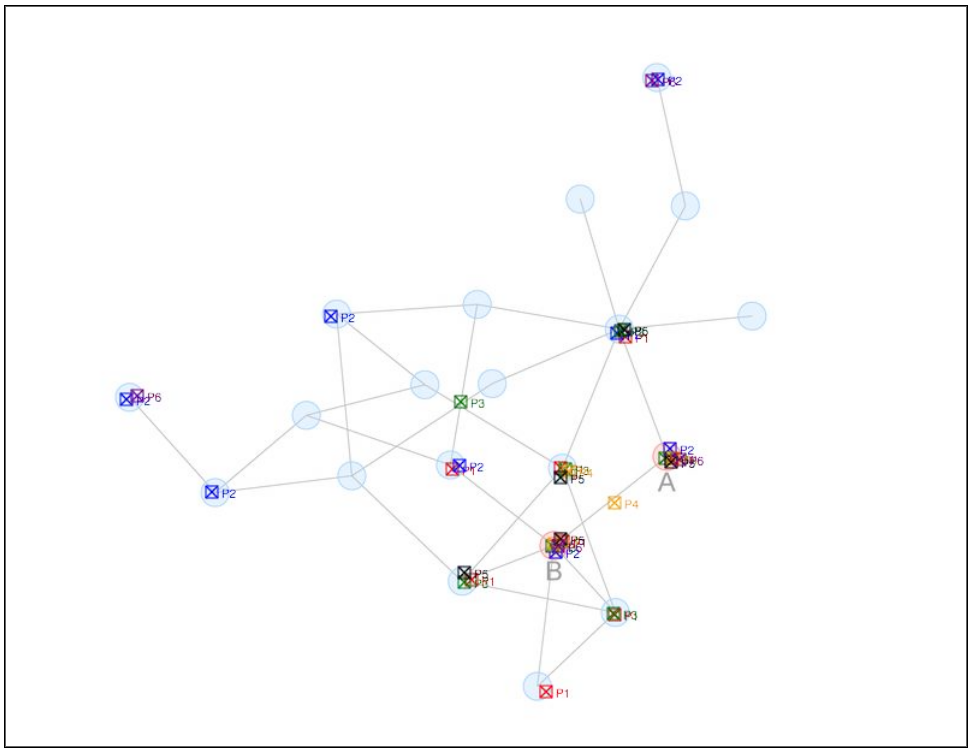


Node-link diagrams

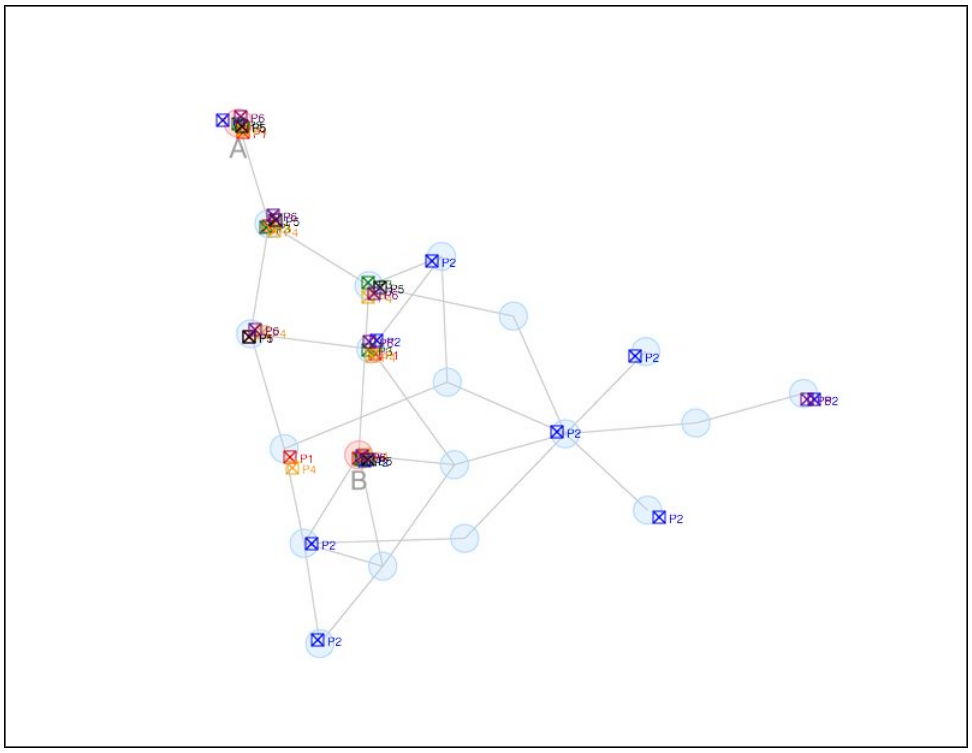
Task 11: What is the fewest number of edges to travel between the red marks A and B?



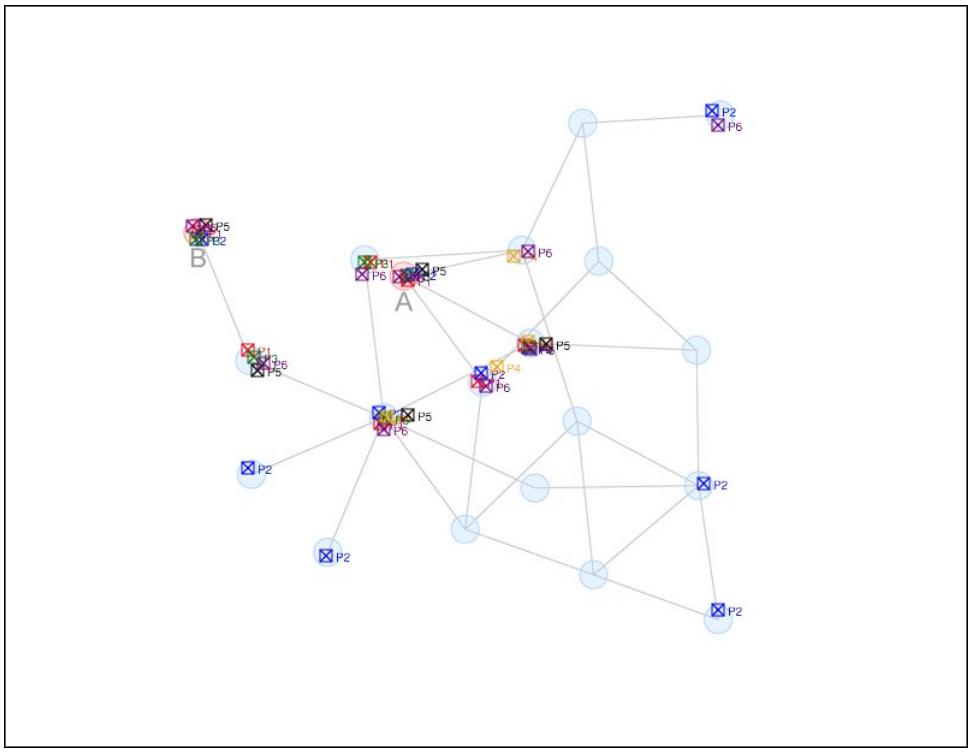
Task 12: What is the fewest number of edges to travel between the red marks A and B?



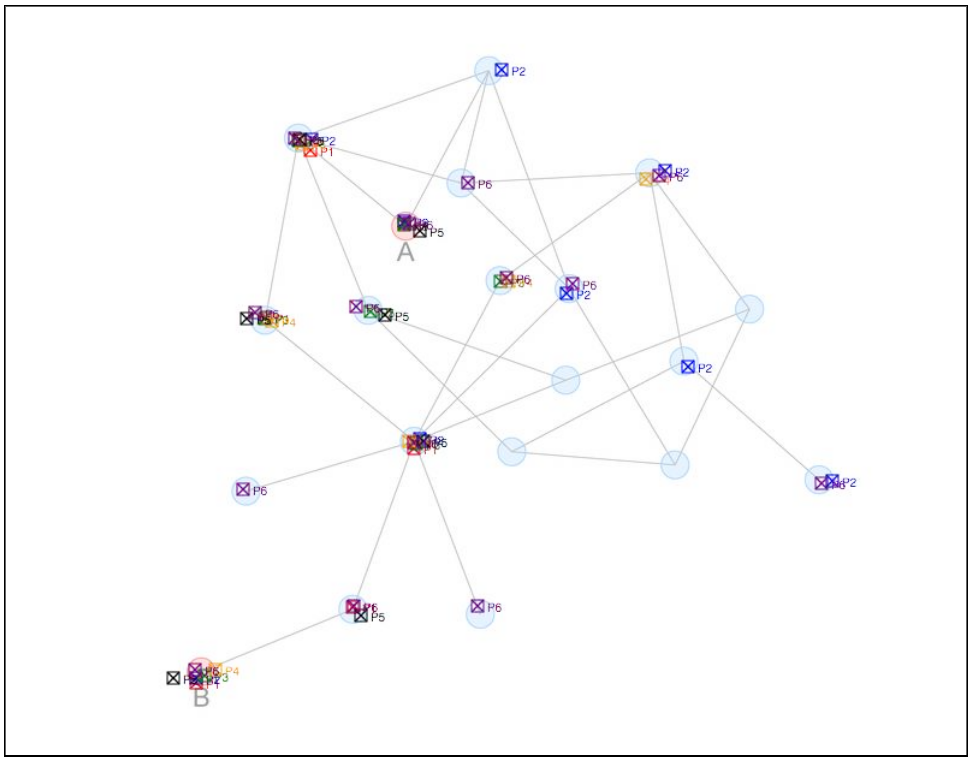
Task 13: What is the fewest number of edges to travel between the red marks A and B?



Task 14: What is the fewest number of edges to travel between the red marks A and B?

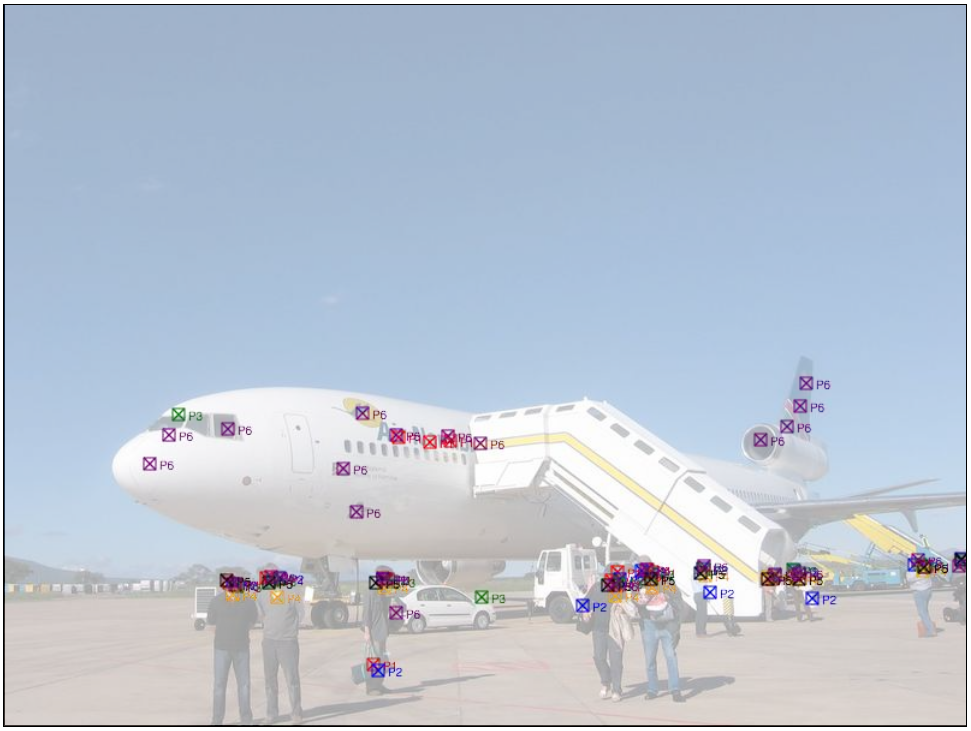


Task 15: What is the fewest number of edges to travel between the red marks A and B?



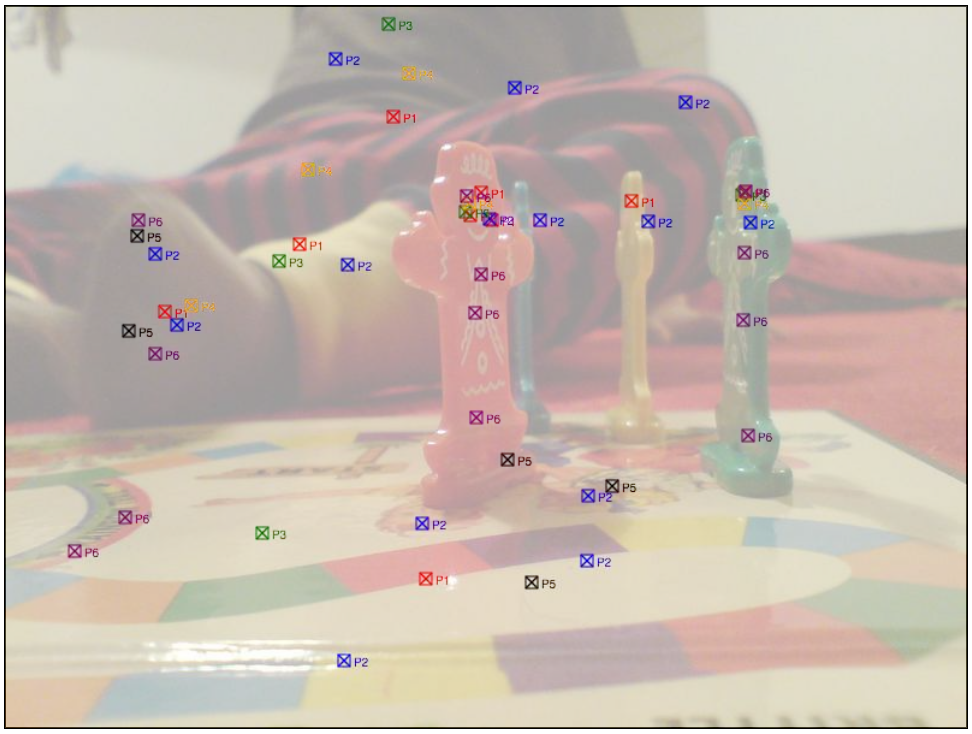
Photographs

Task 16: Estimate the average age (years) of all people in the photo.



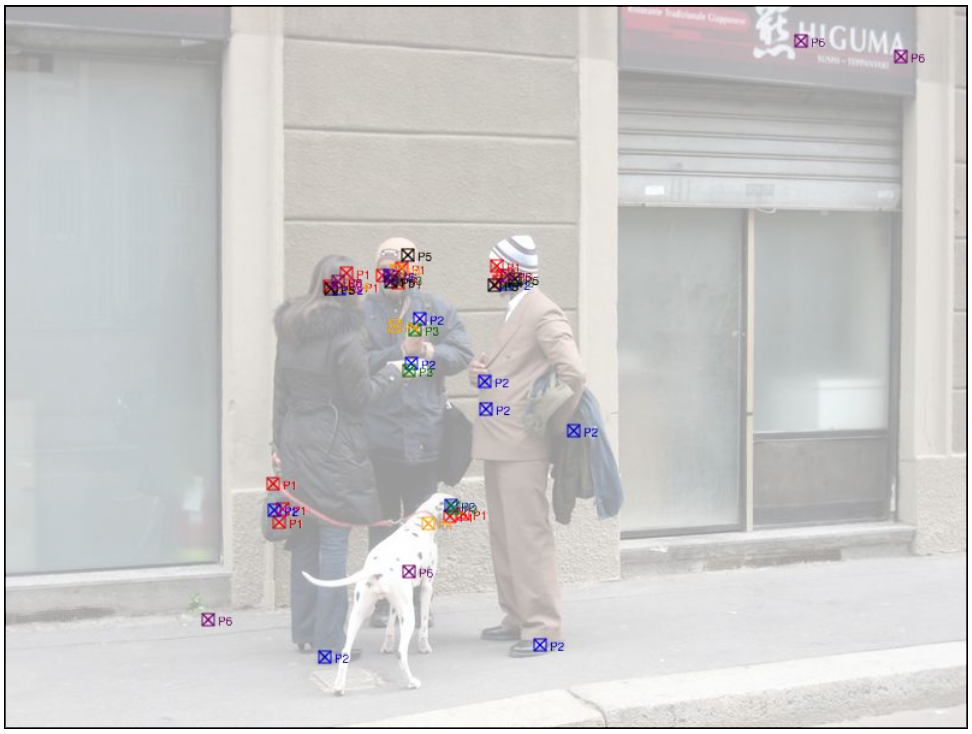
Task 17: Estimate the average age (years) of all people in the photo.

Task 18: Estimate the average age (years) of all people in the photo.



Task 19: Estimate the average age (years) of all people in the photo.

Task 20: Estimate the average age (years) of all people in the photo.



Bibliography

[1] IEEE VAST Challenge 2011. `http://hcil.cs.umd.edu/localphp/hcil/vast11/`. Accessed: 2014-03-22.

[2] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*, pages 111–117, Oct 2005.

[3] J. R. Anderson. Spanning seven orders of magnitude: a challenge for cognitive modeling. *Cognitive Science*, 26:85–112, 2002.

[4] Natalia Andrienko and Gennady Andrienko. A visual analytics framework for spatio-temporal analysis and modelling. *Data Min. Knowl. Discov.*, 27(1):55–83, July 2013.

[5] Natalia Andrienko, Gennady Andrienko, and Peter Gatalsky. Exploratory spatio-temporal visualization: an analytical review. *Journal of Visual Languages & Computing*, 14(6):503–541, 2003.

[6] F. J. Anscombe. Graphs in statistical analysis. *The American Statistician*, 27(1):17–21, 1973.

[7] Roman Bednarik and Markku Tukiainen. Validating the restricted focus viewer: A study using eye-movement tracking. *Behavior Research Methods*, 39(2):274–282, 2007.

[8] L. Bergroth, H. Hakonen, and T. Raita. A survey of longest common subsequence algorithms. In *Proceedings of the 7th International Symposium on String Processing and Information Retrieval (SPIRE)*, pages 39–48, 2000.

[9] Alan F. Blackwell, Anthony R. Jansen, and Kim Marriott. Restricted focus viewer: A tool for tracking visual attention. In *Proceedings of the First International Conference on Theory and Application of Diagrams*, Diagrams '00, pages 162–177, London, UK, UK, 2000. Springer-Verlag.

[10] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D3: Data-driven documents. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2301–2309, Dec 2011.

[11] Matthew Brehmer, Sheelagh Carpendale, Bongshin Lee, and Melanie Tory. Pre-design empiricism for information visualization: Scenarios, methods, and challenges. In *Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization*, pages 147–151. ACM, 2014.

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

[13] E.T. Brown, A. Ottley, H. Zhao, Quan 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, Dec 2014.

[14] Michael Burch, Natalia Konevtsova, Julian Heinrich, Markus Hoeferlin, and Daniel Weiskopf. Evaluation of traditional, orthogonal, and radial tree diagrams by an eye tracking study. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2440–2448, December 2011.

[15] Zoya Bylinskii, Tilke Judd, Ali Borji, Laurent Itti, Frédo Durand, Aude Oliva, and Antonio Torralba. Mit saliency benchmark.

[16] Michael D. Byrne. ACT-R/PM and menu selection: applying a cognitive architecture to HCI. *International Journal of Human-Computer Studies*, 55(1):41–84, 2001.

[17] Steven P. Callahan, Juliana Freire, Emanuele Santos, Carlos E. Scheidegger, Cláudio T. Silva, and Huy T. Vo. Vistrails: visualization meets data management. In *ACM SIGMOD*, pages 745–747, 2006.

[18] Mike Cammarano, Xin Dong, Bryan Chan, Jeff Klingner, Justin Talbot, Alon Halevy, and Pat Hanrahan. Visualization of heterogeneous data. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1200–1207, 2007.

[19] Nan Cao, David H Gotz, and Jimeng Sun. Multi-faceted visualization of rich text corpora, August 31 2010. US Patent App. 12/872,794.

[20] S. Card, J.D. Mackinlay, and B. Shneiderman. *Readings in Information Visualization – Using Vision to Think*. Morgan Kaufmann, San Francisco, CA, USA, 1999.

[21] S. K. Card, T. P. Moran, and A. Newell. The keystroke-level model for user performance time with interactive systems. *Comm. of the ACM*, 23(7):396–410, 1980.

[22] Sheelagh Carpendale. Evaluating information visualizations. In Andreas Kerren, John T. Stasko, Jean-Daniel Fekete, and Chris North, editors, *Information Visualization*, pages 19–45. Springer-Verlag, Berlin, Heidelberg, 2008.

[23] Davida Charney, Lynne Reder, and Gail Kusbit. Goal setting and procedure selection in acquiring computer skills: A comparison of tutorials, problem solving, and learner exploration. *Cognition and Instruction*, 7(4):323–342, 1990.

[24] Mon Chu Chen, John R Anderson, and Myeong Ho Sohn. What can a mouse cursor tell us more?: correlation of eye/mouse movements on web browsing. In *CHI '01 Extended Abstracts on Human Factors in Computing Systems*, pages 281–282. ACM, 2001.

[25] Michael D. Fleetwood and Michael D. Byrne. Modeling the visual search of displays: a revised ACT-R model of icon search based on eye-tracking data. *Hum.-Comput. Interact.*, 21:153–197, 2008.

[26] Steven R. Gomez, Hua Guo, Caroline Ziemkiewicz, and David H. Laidlaw. An insight- and task-based methodology for evaluating spatiotemporal visual analytics. In *IEEE VAST*, VAST '14, 2014.

[27] Steven R. Gomez, Radu Jianu, Ryan Cabeen, Hua Guo, and David H. Laidlaw. Fauxvea: Crowdsourcing gaze estimates for visualization analysis tasks. *Visualization and Computer Graphics, IEEE Transactions on*, in review (2015).

[28] Steven R. Gomez, Radu Jianu, and David H. Laidlaw. A fiducial-based tangible user interface for white matter tractography. In *Advances in Visual Computing*, pages 373–381. Springer, 2010.

[29] Steven R. Gomez and David H. Laidlaw. Modeling human performance from visualization interaction histories. In *Proceedings of IEEE InfoVis (Posters)*. IEEE, 2011.

[30] Steven R. Gomez and David H. Laidlaw. Modeling task performance for a crowd of users from interaction histories. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2465–2468, New York, NY, USA, 2012. ACM.

[31] Wayne D. Gray, Bonnie E. John, and Michael E. Atwood. Project Ernestine: validating a GOMS analysis for predicting and explaining real-world task performance. *Hum.-Comput. Interact.*, 8(3):237–309, 1993.

[32] Georges G. Grinstein, Patrick E. Hoffman, and Ronald M. Pickett. Information visualization in data mining and knowledge discovery. chapter Benchmark Development for the Evaluation of Visualization for Data Mining, pages 129–176. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.

[33] Hua Guo, Steven R. Gomez, Mark J. Schnitzer, and David H. Laidlaw. Visualization to facilitate structured exploration of published findings in rat brain connectivity. In *Proceedings of IEEE InfoVis (Posters)*. IEEE, 2013.

[34] Hua Guo, Steven R. Gomez, Caroline Ziemkiewicz, and David H. Laidlaw. A case study using visualization interaction logs and insight metrics to understand how analysts arrive at insights. In *IEEE VAST*, VAST '15, 2015.

[35] Hua Guo, Arthur Yidi, Steven R. Gomez, Mark J. Schnitzer, David Badre, and David H. Laidlaw. Toward a visual interface for brain connectivity analysis. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, pages 1761–1766, New York, NY, USA, 2013. ACM.

[36] Qi Guo and Eugene Agichtein. Towards predicting web searcher gaze position from mouse movements. In *CHI'10 Extended Abstracts on Human Factors in Computing Systems*, pages 3601–3606. ACM, 2010.

[37] Joshua Hailpern, Nicholas Jitkoff, Joseph Subida, and Karrie Karahalios. The CLOTHO project: predicting application utility. In *ACM DIS*, DIS '10, pages 330–339, New York, NY, 2010. ACM.

[38] D.W. Hansen and Qiang Ji. In the eye of the beholder: A survey of models for eyes and gaze. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(3):478–500, March 2010.

[39] Steve Haroz and David Whitney. How capacity limits of attention influence information visualization effectiveness. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2402–2410, Dec 2012.

[40] Brett N. Harris, Bonnie E. John, and Jonathan Brezin. Human performance modeling for all: importing UI prototypes into CogTool. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, pages 3481–3486, 2010.

[41] Björn Hartmann, Sean Follmer, Antonio Ricciardi, Timothy Cardenas, and Scott R. Klemmer. d.note: revising user interfaces through change tracking, annotations, and alternatives. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 493–502, 2010.

[42] Jeffrey Heer and Michael Bostock. Crowdsourcing graphical perception: using mechanical turk to assess visualization design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 203–212, New York, NY, USA, 2010. ACM.

[43] Jeffrey Heer, Stuart K. Card, and James A. Landay. Prefuse: a toolkit for interactive information visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 421–430, 2005.

[44] Jeffrey Heer, Jock Mackinlay, Chris Stolte, and Maneesh Agrawala. Graphical histories for visualization: supporting analysis, communication, and evaluation. *Visualization and Computer Graphics, IEEE Transactions on*, 14:1189–1196, November 2008.

[45] Jeffrey Heer and Ben Shneiderman. Interactive dynamics for visual analysis. *Commun. ACM*, 55(4):45–54, April 2012.

[46] Joseph Henrich, Steven J Heine, and Ara Norenzayan. The weirdest people in the world? *Behavioral and brain sciences*, 33(2-3):61–83, 2010.

[47] David M. Hilbert and David F. Redmiles. Extracting usability information from user interface events. *ACM Comput. Surv.*, 32:384–421, December 2000.

[48] Jeff Huang, Ryen White, and Georg Buscher. User see, user point: Gaze and cursor alignment in web search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 1341–1350, New York, NY, USA, 2012. ACM.

[49] Jeff Huang, Ryen W. White, and Susan Dumais. No clicks, no problem: using cursor movements to understand and improve search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 1225–1234, New York, NY, USA, 2011. ACM.

[50] Scott E. Hudson, Bonnie E. John, Keith Knudsen, and Michael D. Byrne. A tool for creating predictive performance models from user interface demonstrations. In *ACM UIST*, pages 93–102, 1999.

[51] Petra Isenberg, Torre Zuk, Christopher Collins, and Sheelagh Carpendale. Grounded evaluation of information visualizations. In *Proceedings of the 2008 Workshop on BEyond Time and Errors: Novel evaLuation Methods for Information Visualization*, BELIV '08, pages 6:1–6:8, New York, NY, USA, 2008. ACM.

[52] Tobias Isenberg, Petra Isenberg, Jian Chen, Michael Sedlmair, and Torsten Moller. A systematic review on the practice of evaluating visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2818–2827, 2013.

[53] Anthony R. Jansen, Alan R. Blackwell, and Kim Marriott. A tool for tracking visual attention: The restricted focus viewer. *Behavior Research Methods, Instruments, and Computers*, 35(1):57–69, 2003.

[54] Radu Jianu and David Laidlaw. An evaluation of how small user interface changes can improve scientists' analytic strategies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2953–2962, New York, NY, USA, 2012. ACM.

[55] Bonnie John, Alonso Vera, Michael Matessa, Michael Freed, and Roger Remington. Automating CPM-GOMS. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 147–154, 2002.

[56] Bonnie E. John. Reducing the variability between novice modelers: results of a tool for human performance modeling produced through human-centered design. In *BRIMS*, pages 95–102, 2010.

[57] Bonnie E. John and David E. Kieras. The GOMS family of user interface analysis techniques: comparison and contrast. *ACM Trans. Comput.-Hum. Interact.*, 3:320–351, December 1996.

[58] Bonnie E. John, Konstantine Prevas, Dario D. Salvucci, and Ken Koedinger. Predictive human performance modeling made easy. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 455–462, 2004.

[59] Bonnie E. John and Shunsuke Suzuki. Toward cognitive modeling for predicting usability. In *HCI International*, pages 267–276, 2009.

[60] Eric J. Johnson, John W. Payne, David A. Schkade, and James R. Bettman. Monitoring information processing and decisions: The Mouselab system. Technical Report 89-4, Office of Naval Research, 1989.

[61] Jeff Johnson et al. *Designing with the mind in mind: Simple guide to understanding user interface design rules.* Morgan Kaufmann, 2010.

[62] Michael N. Jones and D. J. K. Mewhort. Tracking attention with the focus-window technique: The information filter must be calibrated. *Behavior Research Methods, Instruments, & Computers*, 36(2):270–276, 2004.

[63] Octavio Juarez-Espinosa. CAEVA: Cognitive Architecture to Evaluate Visualization Applications. In *Proceedings of the Seventh International Conference on Information Visualization*, Washington, DC, 2003. IEEE Computer Society.

[64] Tilke Judd, Frédo Durand, and Antonio Torralba. A benchmark of computational models of saliency to predict human fixations. In *MIT Technical Report*, 2012.

[65] Tilke Judd, Krista Ehinger, Frédo Durand, and Antonio Torralba. Learning to predict where humans look. In *IEEE International Conference on Computer Vision (ICCV)*, 2009.

[66] Marcel Adam Just and Patricia A. Carpenter. Eye fixations and cognitive processes. *Cognitive Psychology*, 8:441–480, 1976.

[67] Daniel Keim, Jörn Kohlhammer, Geoffrey Ellis, and Florian Mansmann, editors. *Mastering the Information Age: Solving Problems with Visual Analytics.* Eurographics Association, Goslar, Germany, 2010.

[68] Sung-Hee Kim, Zhihua Dong, Hanjun Xian, B. Upatising, and Ji Soo Yi. Does an eye tracker tell the truth about visualizations?: Findings while investigating visualizations for decision making. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2421–2430, Dec 2012.

[69] Nicholas Kong, Jeffrey Heer, and Maneesh Agrawala. Perceptual guidelines for creating rectangular treemaps. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):990–998, 2010.

[70] David Koop. Viscomplete: Automating suggestions for visualization pipelines. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1691–1698, 2008.

[71] Robert Kosara, Silvia Miksch, and Helwig Hauser. Semantic depth of field. In *Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01)*, Washington, DC, USA, 2001. IEEE Computer Society.

[72] Robert Kosara and Caroline Ziemkiewicz. Do mechanical turks dream of square pie charts? In *Proceedings of the 3rd BELIV'10 Workshop: BEyond time and errors: novel evaLuation methods for Information Visualization*, pages 63–70. ACM, 2010.

[73] Dmitry Lagun and Eugene Agichtein. Viewser: enabling large-scale remote user studies of web search examination and interaction. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR '11, pages 365–374, New York, NY, USA, 2011. ACM.

[74] H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale. Empirical studies in information visualization: Seven scenarios. *Visualization and Computer Graphics, IEEE Transactions on*, 18(9):1520–1536, Sept 2012.

[75] David Lloyd and Jason Dykes. Human-centered approaches in geovisualization design: Investigating multiple methods through a long-term case study. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2498–2507, 2011.

[76] I. Scott MacKenzie and William Buxton. Extending Fitts' law to two-dimensional tasks. In *ACM CHI*, pages 219–226, 1992.

[77] C. North. Toward measuring visualization insight. *Computer Graphics and Applications, IEEE*, 26(3):6–9, May 2006.

[78] Chris North, Purvi Saraiya, and Karen Duca. A comparison of benchmark task and insight evaluation methods for information visualization. *Information Visualization*, 10(3):162–181, July 2011.

[79] T.M. O'Brien, A.M. Ritz, B.J. Raphael, and D.H. Laidlaw. Gremlin: An interactive visualization model for analyzing genomic rearrangements. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):918–926, Nov 2010.

[80] Mershack Okoe and Radu Jianu. Graphunit: Evaluating interactive graph visualizations using crowdsourcing. In *Computer Graphics Forum*, volume 34, pages 451–460. Wiley Online Library, 2015.

[81] A. Perer and B. Shneiderman. Integrating statistics and visualization for exploratory power: From long-term case studies to design guidelines. *Computer Graphics and Applications, IEEE*, 29(3):39–51, May 2009.

[82] Donna J. Peuquet. It's about time: a conceptual framework for represetation of temporal dynamics in geographic information systems. *Annals of the Association of American Geographers*, 84(3):441–461, 1994.

[83] Peter Pirolli and Wai-Tat Fu. SNIF-ACT: a model of information foraging on the world wide web. In *Proceedings of the 9th International Conference on User Modeling*, UM'03, pages 45–54, Berlin, Heidelberg, 2003. Springer-Verlag.

[84] C. Plaisant, J.-D. Fekete, and G. Grinstein. Promoting insight-based evaluation of visualizations: from contest to benchmark repository. *Visualization and Computer Graphics, IEEE Transactions on*, 14(1):120–134, Jan 2008.

[85] Catherine Plaisant. The challenge of information visualization evaluation. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '04, pages 109–116, New York, NY, USA, 2004. ACM.

[86] Mathias Pohl, Markus Schmitt, and Stephan Diehl. Comparing the readability of graph layouts using eyetracking and task-oriented analysis. In *Proceedings of the Eurographics Conference on Computational Aesthetics*, pages 49–56, Aire-la-Ville, Switzerland, Switzerland, 2009. Eurographics Association.

[87] Alex Poole and Linden J. Ball. Eye tracking in human-computer interaction and usability research: Current status and future prospects. In Claude Ghaoui, editor, *Encyclopedia of Human-Computer Interaction*. Idea Group, 2005.

[88] John Rieman. A field study of exploratory learning strategies. *ACM Trans. Comput.-Hum. Interact.*, 3(3):189–218, September 1996.

[89] John Rieman, Richard M. Young, and Andrew Howes. A dual-space model of iteratively deepening exploratory learning. *Int. J. Hum.-Comput. Stud.*, 44(6):743–775, June 1996.

[90] Kerry Rodden and Xin Fu. Exploring how mouse movements relate to eye movements on web search results pages. In *Web Information Seeking and Interaction Workshop*, 2006.

[91] Kerry Rodden, Xin Fu, Anne Aula, and Ian Spiro. Eye-mouse coordination patterns on web search results pages. In *CHI' 08 Extended Abstracts on Human Factors in Computing Systems*, pages 2997–3002. ACM, 2008.

[92] Ruth Rosenholtz, Jie Huang, Alvin Raj, Benjamin J. Balas, and Livia Ilie. A summary-statistic representation in peripheral vision explains visual search. *Journal of Vision*, 12(4):1–17, 2012.

[93] Julian B Rotter. Generalized expectancies for internal versus external control of reinforcement. *Psychological monographs: General and applied*, 80(1):1, 1966.

[94] Dmitry Rudoy, Dan B. Goldman, Eli Shechtman, and Lihi Zelnik-Manor. Crowdsourcing gaze data collection. In *Conference on Collective Intelligence (CI)*, 2012.

[95] P. Saraiya, C. North, and K. Duca. An insight-based methodology for evaluating bioinformatics visualizations. *Visualization and Computer Graphics, IEEE Transactions on*, 11(4):443–456, July 2005.

[96] P. Saraiya, C. North, V. Lam, and K.A. Duca. An insight-based longitudinal study of visual analytics. *Visualization and Computer Graphics, IEEE Transactions on*, 12(6):1511–1522, Nov 2006.

[97] Carlos Scheidegger, Huy Vo, David Koop, Juliana Freire, and Claudio Silva. Querying and creating visualizations by analogy. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1560–1567, 2007.

[98] H.-J. Schulz, T. Nocke, M. Heitzler, and H. Schumann. A design space of visualization tasks. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2366–2375, Dec 2013.

[99] Zeqian Shen, Kwan-Liu Ma, and Tina Eliassi-Rad. Visual analysis of large heterogeneous social networks by semantic and structural abstraction. *Visualization and Computer Graphics, IEEE Transactions on*, 12(6):1427–1439, 2006.

[100] Ben Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transactions on Graphics*, 11(1):92–99, January 1992.

[101] Ben Shneiderman and Catherine 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*, BELIV '06, pages 1–7, New York, NY, USA, 2006. ACM.

[102] A. Slingsby, J. Dykes, and J. Wood. Configuring hierarchical layouts to address research questions. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):977–984, Nov 2009.

[103] Robert St. Amant and Frank E. Ritter. Automated GOMS–to–ACT-R model generation. In *Proceedings of the Sixth International Conference on Cognitive Modeling (ICCM)*, pages 28–34, 2004.

[104] Amanda Swearngin, Myra B. Cohen, Bonnie E. John, and Rachel K. E. Bellamy. Human performance regression testing. In *Proceedings of the 2013 International Conference on Software Engineering*, ICSE '13, pages 152–161, Piscataway, NJ, USA, 2013. IEEE Press.

[105] James J. Thomas and Kristin A. Cook. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. National Visualization and Analytics Center, 2005.

[106] Pingmei Xu, Krista A. Ehinger, Yinda Zhang, Adam Finkelstein, Sanjeev R. Kulkarni, and Jianxiong Xiao. Turkergaze: Crowdsourcing saliency with webcam based eye tracking. *CoRR*, abs/1504.06755, 2015.

[107] Ji Soo Yi, Youn ah Kang, John T Stasko, and Julie A Jacko. Toward a deeper understanding of the role of interaction in information visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1224–1231, 2007.

[108] Lingyun Zhang, Matthew H Tong, Tim K Marks, Honghao Shan, and Garrison W Cottrell. Sun: A bayesian framework for saliency using natural statistics. *Journal of Vision*, 8(7):32, 2008.

[109] C. Ziemkiewicz, A. Ottley, R. J. Crouser, A. R. Yauilla, S. L. Su, W. Ribarsky, and R. Chang. How visualization layout relates to locus of control and other personality factors. *Visualization and Computer Graphics, IEEE Transactions on*, 19(7):1109–1121, 2013.