Research Statement

Bahador Saket

April 1, 2017

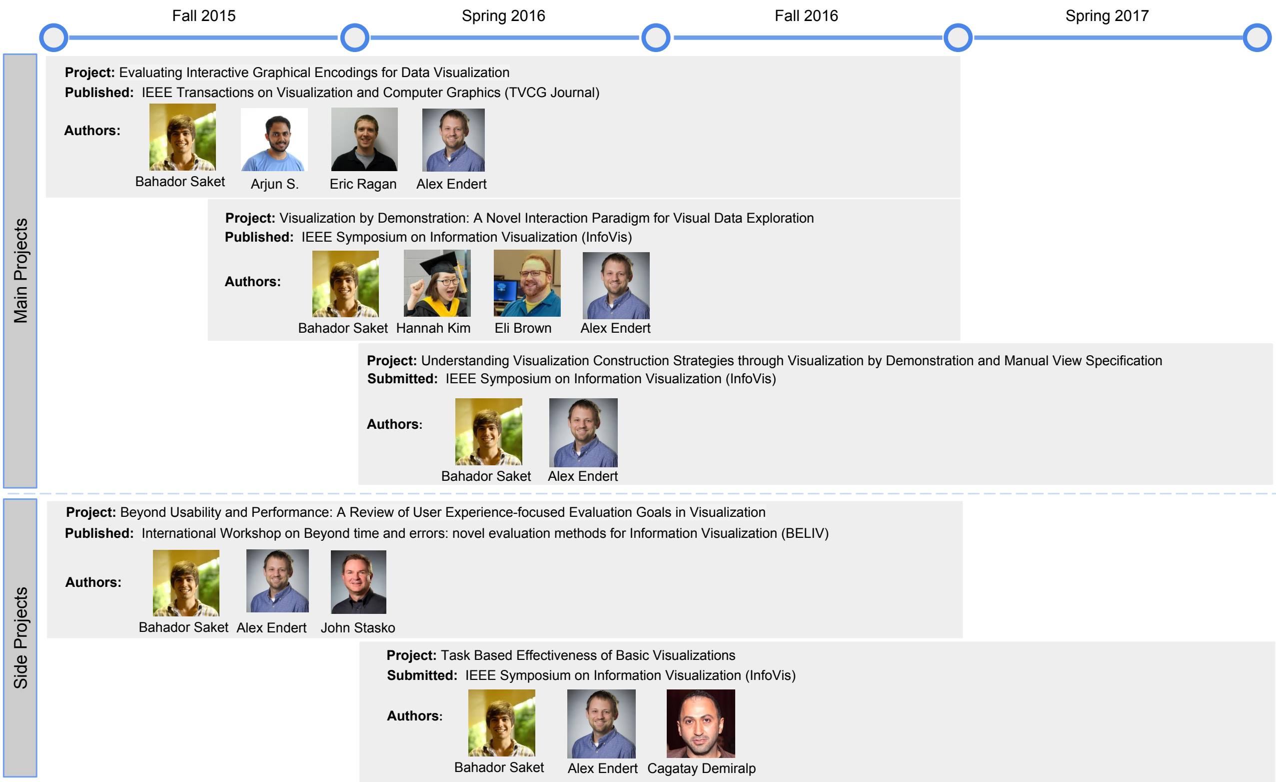


Figure 1: Overview of the selected research projects that I have worked on over the past two years.

1 Introduction to my research

Data visualization provides an opportunity for broad impact as its use has gone mainstream — from business intelligence to data-driven journalism, society has embraced visualization to record, analyze, and communicate data. Visualization researchers and practitioners continue to develop a wide range of interactive visualization tools which allow users to explore and make sense of their data. These visualization tools often consist of two main components: visual representation and interaction [10]. The fundamental focus of visual representation is mapping from data values to graphical representations and how the graphical representations are rendered on a display. The main focus of interaction is to enable users to manipulate and interpret the representation. At a high level my research focuses on the interaction component of the visualization tools.

The primary interaction technique of majority of the existing visualization tools is manual view specification [9]. Manual view specification incorporates users interaction with interface elements such as menus, sliders, and dialog boxes that act as mediators between users and the visual representation. Visualization tools incorporating manual view specification include interface elements and visual representations of data *(e.g., bar charts)* in two visually-separate panels. In order to interact with the visualization tools, users normally interact with these elements in one panel and observe the resulting changes to the visualization in another view. We observed three main challenges in user interaction with visualization tools incorporating the manual view specification technique. Below, I describe each of these challenges in visualization tools implementing manual view specification.

- **Challenge 1:** Visualization tools implementing manual view specification need users to constantly switch their attention between the interface elements and the visual features of interest while interacting with the tools [3, 6]. For example, while manipulating the slider on the control panel users should constantly observe the changes on the visual output.
- **Challenge 2:** Visualization tools implementing manual view specification need users to manually specify visualization characteristics that require time and effort of users [7, 8, 9]. For example, prior to constructing a visualization users need to manually specify the data attributes related to the task at the hand, specify a visualization technique/mark type to represent the data, and assign data attributes to visual encodings.
- **Challenge 3:** Visualization tools implementing manual view specification require users to have some amount of fundamental knowledge about the data, the domain, and of visualization techniques [7, 9]. For example, for visualizing a dataset using Microsoft Excel, users required to upload or enter their data. They then need to select data attributes that they are interested in visualizing. Users finally need to select a suitable visualization techniques.

My goal during my PhD is to develop novel interaction techniques for information visualization tools to overcome some of these challenges (Challenge 1 - 3).

2 What have I done so far ?

Towards the goal of overcoming these challenges in visualization tools (Challenge 1 - 3), I have been working on several projects over the past two years [4, 5, 6, 7]. Below, I will describe some of these projects.

Embedded Interaction for Data Visualization

Visualization tools incorporating manual view specification require users to interact with interface elements in one view and observe the resulting changes to the visualization in another view; see Figure 4-a. As we discussed, one of the main challenges in incorporating manual view specification technique to visualization tools is that users need to constantly switch their attention between the interface elements and the visual features of interest while interacting with the tools **(Challenge 1)** [3]. To overcome this challenge, there has been an increasing trend in embedding user interaction into the visual representations (e.g., [1, 3, 2]); see Figure 4-b. For example, DimpVis is a recent system that allows users to directly interact with the length, angle and position of the visual representations, as a means for temporal navigation [3]. In DimpVis, users can adjust the height of a bar to see its value at different moments in time. For instance, to check if at any point in time the value associated with a bar is half its current value, the user can drag the bar vertically downwards to compare its values at different points in time. As more systems leverage graphical encodings in the visual representations not only to represent data visually but also to serve as the method for user interaction, this motivated the need to understand the effectiveness of interaction with these graphical encodings. In the first project [6], we studied the effectiveness of 12 elementary graphical encodings when serving as the method for interaction. Overall, the results of our study indicate that people can interact accurately with some of the graphical encodings such as length and poisition when serving as the method for interaction [6].

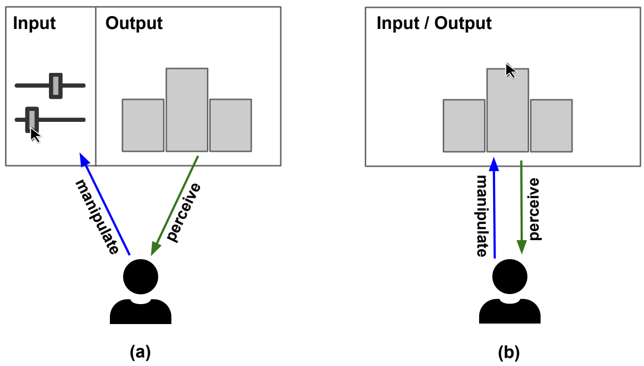


Figure 2: Two different forms of interaction in visualization tools. In order to interact with the visualization, users are required to either manipulate the interface elements in a separate panel (a) or directly manipulate the graphical encodings used in a visual representation (b).

Visualization by Demonstration

Based on our previous work [6] and inspired by earlier work (e.g., [1, 3]), we next designed a novel interaction method for data visualization called visualization by demonstration [7]. This technique advocates for balancing user and system responsibilities for data visualization **(Challenge 2)** and and decreasing fundamental knowledge required for visual data exploration **(Challenge 3)**. Instead of specifying which visualization technique, mappings, and parameters to generate and update a visualization, visualization by demonstration allows users to provide visual demonstrations of incremental changes to the visual representation from which transformations are recommended. Using these demonstrations, the system estimates the intentions and generates potential transformations *(e.g., a bar chart, mapping color to a data attribute)*. This iterative process allows users to visually explore their data without requiring direct visualization specification. That is, the goal is to balance the responsibility of data exploration between the user and the system — users provide visual demonstrations, while the system generates visualizations and defines the visualization mappings and parameters.

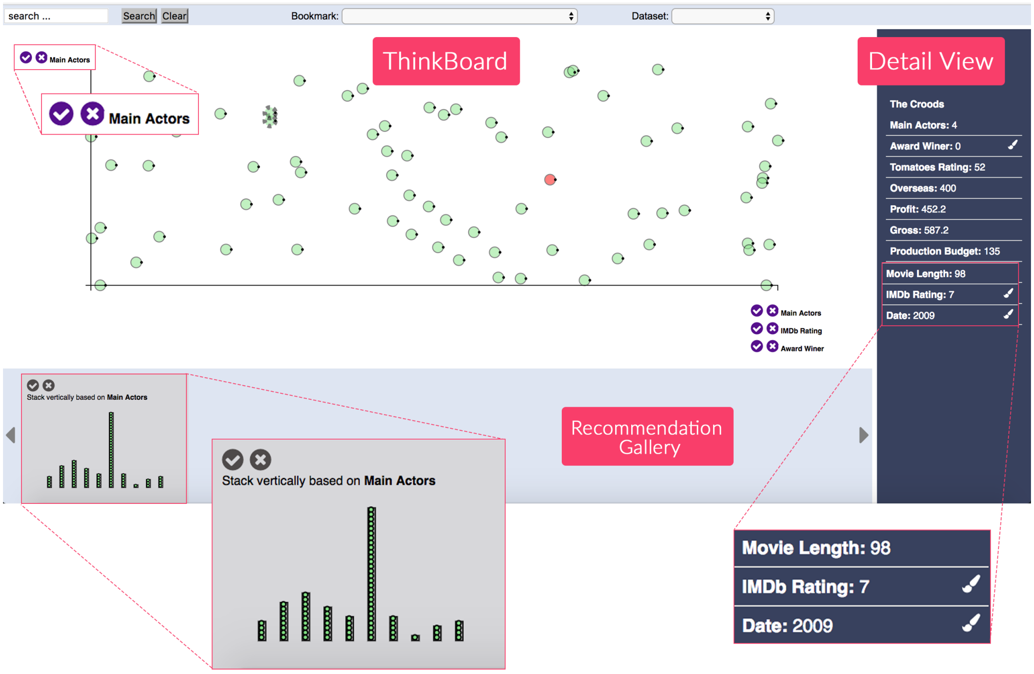


Figure 3: The VisExemplar user interface consists of a ThinkBoard, Recommendation Gallery, and a Detail View panel. ThinkBoard shows each data point as a circle. The Recommendation Gallery shows visualization technique transformations. The Detail View shows data details, and also recommended data mapping transformations.

To show the feasibility of the visualization by demonstration concept, we implemented VisExemplar; see Figure 3. VisExemplar is a mixed-initiative data exploration prototype that allows users to explore their data using visualization by demonstration. VisExemplar allows users to provide visual demonstrations of incremental changes to the visualization by directly manipulating the visual representation *(e.g., moving one data point on the top of another, changing the color of a select set of point, etc.)*. VisExemplar uses a recommendation engine that generates transformations in response to the given demonstrations using a set of intent functions. To help users understand the results of different recommended transformations before they commit, VisExemplar contributes novel methods for presenting recommended transformations.

Evaluation of the Visualization by Demonstration Technique

There are two main differences between manual view specification technique and visualization by demonstration. First, manual view specification uses a different model of the information visualization process compared to visualization by demonstration [6]. Second, these two interaction techniques use different interaction metaphors [6]. While we understand the differences in design considerations that went into manual view specification and visualization by demonstration techniques, it remains unclear how these differences affect the process of visualization construction and data exploration, and the trade-offs between the two techniques for specific user tasks. Understanding these trade-offs informs the visualization community, in particular, those whose goal is to design new interaction techniques or adapt existing techniques in their visualization tools. To explore these questions, we designed a two-phase study. In the third project, our goal was to investigate which processes people follow while exploring their data, which common patterns appear, and which barriers people encounter using each interaction technique. Thus, we conducted a two-phase study comparing people's visualization construction and data exploration process using two visualization tools: one implementing the manual view specification technique (Polestar) and another implementing the visualization by demonstration technique (VisExamplar). Findings of our study indicate that these two techniques influence: 1) the effectiveness of visualization process, 2) strategies used for constructing visualizations and exploring the data, and 3) feeling of control and engagement during the visualization process [6].

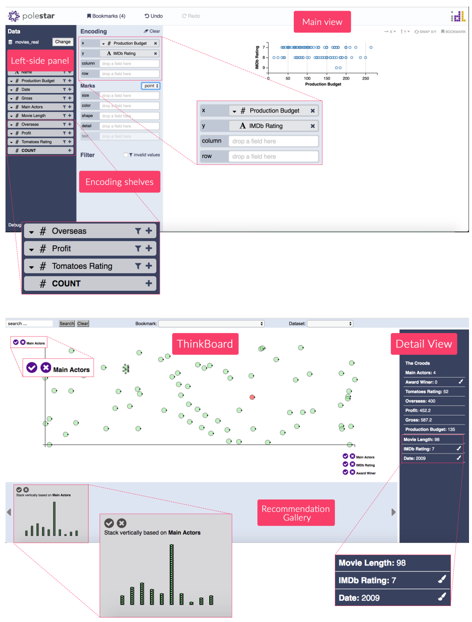


Figure 4: Two visualization tools implementing each interaction technique. VisExamplar (Bottom) incorporates visualization by demonstration and Polestar (Top) incorporates manual view specification.

3 What is next?

Generalizing visualization by demonstration
We developed VisExemplar to show the feasibility of visualization by demonstration. The current version of VisExamplar supports two types of visualization techniques (bar chart and scatterplot) and direct manipulation of three graphical encodings (position, size, and color). It supports interactions which are recognized to be meaningful by previous work. I view the current version of VisExemplar as the first step towards exploring the visualization by demonstration paradigm. Multiple avenues for future work lie in improving the VisExamplar interface, as well as enriching the visualization by demonstration idea space. We envision expanding VisExamplar by including other visualization techniques *(e.g., barchart and linecharts)* and graphical encodings *(e.g., angle and volume)*, and working towards a generalizable interaction framework for visualization. Generalizing visualization by demonstration requires support for providing demonstrations to imply more sophisticated analytic operations and visualization techniques. For example, how can users indicate their interest in data grouping or aggregation? This requires demonstrations that trigger analytic operations on the data, and show the results visually. For example, users could draw regions around specific data points to demonstrate their interest in executing a clustering algorithm.

Incorporating both manual view specification and visualization by demonstration in one system
Our user study comparing visualization by demonstration and manual view specification showed that each of the interaction techniques has their own advantages and disadvantages in visualization construction [4]. One of the main findings of our study is that people follow different strategies to construct visualizations [4]. When participants knew the exact information required for constructing or refining a visualization (specific strategy), manual view specification technique performed better. In contrast, when participants were unaware of some of the information required for completing constructing or refining a visualization (abstract strategy), visualization by demonstration was more effective. An important avenue for continued research is the design and evaluation of systems that support both visualization construction strategies (abstract and specific strategies). One possible solution is to say we design two types of visualization tools. One type adapts the manual view specification technique to better support specific strategy and another incorporates the visualization by demonstration technique to better supports abstract strategy. However, majority of users who apply a combination of both specific and abstract strategies might not benefit from any of the two visualization tools. We can envision a system that combines both manual view specification and visualization by demonstration techniques. This way users users would benefit the most from using visualization tools in their data exploration process regardless of the strategy they use to construct visualizations.

References

[1] Alex Endert, Patrick Fiaux, and Chris North. "Semantic interaction for visual text analytics". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2012, pp. 473–482.

[2] Hannah Kim et al. "InterAxis: Steering Scatterplot Axes". In: *IEEE Visual Analytics Science and Technology (VAST)* (2015).

[3] Brittany Kondo and Christopher M Collins. "Dimpvis: Exploring time-varying information visualizations by direct manipulation". In: *Visualization and Computer Graphics, IEEE Transactions on* 20.12 (2014), pp. 2003–2012.

[4] Bahador Saket and Alex Endert. "Understanding Visualization Construction Strategies through Visualization by Demonstration and Manual View Specification". In: *IEEE Transactions on Visualization & Computer Graphics* 1 (2017), **Under Review**.

[5] Bahador Saket, Alex Endert, and Demiralp Çağatay. "Task Based Effectiveness of Basic Visualizations". In: *IEEE Transactions on Visualization and Computer Graphics* (2017), **Under Review**.

[6] Bahador Saket et al. "Evaluating Interactive Graphical Encodings for Data Visualization". In: *IEEE Transactions on Visualization and Computer Graphics* (2017), pp. 1–14.

[7] Bahador Saket et al. "Visualization by Demonstration: An Interaction Paradigm for Visual Data Exploration". In: *IEEE Transactions on Visualization & Computer Graphics* 1 (2017), pp. 331–340.

[8] Kanit Wongsuphasawat et al. "Voyager 2: Augmenting Visual Analysis with Partial View Specifications". In: *ACM Human Factors in Computing Systems (CHI)*. 2017.

[9] Kanit Wongsuphasawat et al. "Voyager: Exploratory Analysis via Faceted Browsing of Visualization Recommendations". In: *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)* (2015).

[10] Ji Soo Yi et al. "Toward a deeper understanding of the role of interaction in information visualization". In: *Visualization and Computer Graphics, IEEE Transactions on* 13.6 (2007), pp. 1224–1231.

Bahador Saket

Education

Fall 2015 **Georgia Institute of Technology**, Atlanta, Georgia.
Ph.D in Human-computer Interaction
Advisor: Dr. Alex Endert

2014 – 2015 **University of Arizona**, Tucson, Arizona.
Transferred to Georgia Tech
Advisor: Dr. Stephen Kobourov

2009 – 2013 **Multimedia University**, Cyberjaya, Selangor.
BSc in Computer Science

Professional and Research Experience

2015 - 2016 **Georgia Tech**, Graduate Research Assistant.
Advisor: Dr. Alex Endert

Project: Designing a Visual Data Exploration Technique for Novices

We first conducted a series of exploratory studies to understand how novices construct visualizations. We then designed a novel interaction paradigm for visual data exploration called Visualization by Demonstration. We are currently testing the effectiveness of the Visualization by Demonstration paradigm.

Papers: [InfoVis 2016]

Summer 2014 **Microsoft Research**, Research Intern.
Mentor: Dr. Darren Edge & Dr. Koji Yatani

Project: Designing a Mobile Application for Timing Support

We first applied HCI-Q methodology to elicit presenters feedback and concerns during presentations. We then developed and evaluated a mobile application allowing presenters to set and follow timing targets for sections of slides. We implemented both an experimental program and a practical application using PowerPoint add-in and Windows Phone. Finally, we conducted to studies to measure the effectiveness of our proposed approach.

Papers: [MobileHCI 2014 — Recipient of Honorable Mention Award]

Spring 2012 **National University of Singapore (NUS)**, Research Intern.
Mentor: Dr. Shengdong Zhao

Project: Designing a Vibration-based Notification Interface for Mobile Phones

We first conducted an experiment to understand how mobile phone users perceive the urgency of ten simple vibration alerts. We then developed an Android application that can assign different levels of urgency to different contacts' incoming calls or messages. Finally, we conducted two studies to measure the effectiveness of the designed application.

Papers: [CSCW 2013]

Publications

B. Saket, A. Srinivasan, E. Ragan, A. Endert “Evaluating Interactive Graphical Encodings for Data Visualization”. **(TVCG Journal)**, 2017.

B. Saket, H. Kim, E. T. Brown, A. Endert “Visualization by Demonstration: An Interaction Paradigm for Visual Data Exploration”. **(InfoVis)**, 2016.

B. Saket, C. Scheidegger, S. Kobourov. “Comparing Visualizations From An Enjoyment Perspective”. **(EuroVis)**, 2015.

B. Saket, C. Scheidegger, S. Kobourov. “Map-based Visualizations Increase Long-Term Recall of Data”. **(EuroVis)**, 2015.

B. Saket, C. Scheidegger, S. Kobourov. “Towards Understanding Enjoyment and Flow in Information Visualization”. **(EuroVis)**, 2015.

B. Saket, P. Simonetto, S. Kobourov, K. Borner. “Node, Node-Link, and Node-Link-Group Diagrams: An Evaluation”. **(InfoVis)**, 2014.

S. Kobourov, S. Pupyrev, **B. Saket**. “Are Crossings Important for Drawing Large Graphs?”. **(GD)**, 2014.

B. Saket, P. Simonetto, S. Kobourov. “Group-Level Graph Visualization Taxonomy”. **(EuroVis)**, 2014.

B. Saket, S. Yang, H. Z. Tan, K. Yatani, D. Edge. “TalkZones: Section-based Time Support for Presentations”. **(MobileHCI)**, 2014. Recipient of Honourable Mention Award

B. Saket, C. Prasojo, S. Zhao. “Designing an Effective Vibration-Based Notification Interface for Mobile Phones”. **(CSCW)**, 2013.

B. Saket, T. Y. Lim, F. Behrang. "Toward Emotional Design: An Exploratory Study of iPhone 4". **(AHFE)**, 2012.

Poster and Workshop

H. Saadati, **B. Saket**, N. Memon. “Detecting Malicious Logins in Enterprise Networks Using Visualization”. *13th IEEE Symposium on Visualization for Cyber Security (VizSec), 2016.*

B. Saket, A. Endert, J. Stasko. "A Review of User Experience-focused Evaluation Goals in Visualization". **(BELIV)**, 2016.

F. Tavakolizadeh, J. Gu, B. Saket. "Traceband: Locating Missing Items by Visual Remembrance". **(UIST)**, 2014.

Invited Talks

2016 **IBM Research**, *Balancing User and System Responsibilities for Data Visualization*, USA.

2015 **Dagstuhl Seminar**, *Evaluation in the Crowd: Crowdsourcing and Human-Centred Experiments*, Germany.

Professional Activities

2017 **Organizing Committee**, *Immersive Analytics Workshop (will be held in IEEEVIS 2017 at Phoenix).*

2016 **Program Committee**, *KDD Workshop on Interactive Data Exploration and Analytics.*

2016 **Reviewer**, *IEEE VIS / ACM CHI / IEEE VAST Challenge.*

2015 **Reviewer**, *IEEE VIS / ACM CHI / IEEE VAST Challenge.*

2014 **Reviewer**, *IEEE VIS / ACM CHI / ACM TEI.*

References

Available upon request

List of Courses

Breadth Component Areas	Class
Human Computer Interaction	CS 6750 Human Computer Interaction CS 7450 Information Visualization
Machine Learning	CSE 6242 Data & Visual Analytics
Intelligent Systems	CS 6795 Introduction to Cognitive Science
Algorithm	CS 7510 Graph Algorithms

Teaching Assistant Experience

CS 3750	User Interface Design

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2017.2680452, IEEE Transactions on Visualization and Computer Graphics

Evaluating Interactive Graphical Encodings for Data Visualization

Bahador Saket, Arjun Srinivasan, Eric D. Ragan, Alex Endert

Abstract—User interfaces for data visualization often consist of two main components: control panels for user interaction and visual representation. A recent trend in visualization is directly embedding user interaction into the visual representations. For example, instead of using control panels to adjust visualization parameters, users can directly adjust basic graphical encodings *(e.g., changing distances between points in a scatterplot)* to perform similar parameterizations. However, enabling embedded interactions for data visualization requires a strong understanding of how user interactions influence the ability to accurately control and perceive graphical encodings. In this paper, we study the effectiveness of these graphical encodings when serving as the method for interaction. Our user study includes 12 *interactive graphical encodings*. We discuss the results in terms of task performance and interaction effectiveness metrics.

Index Terms—Information visualization, user interaction, graphical encodings, graphical perception

1 Introduction

INTERACTIVITY is a central component of visual data analysis. Traditionally, many data visualization systems have included interactive widgets *(e.g., drop-down menus)* and visual representations of data *(e.g., bar charts)* in two visually-separate panels. In order to interact with the system, users normally interact with these widgets in one panel and observe the resulting changes to the visualization in another view *(e.g., [41])*; see Figure 1-a.

More recently, rather than requiring interaction through external widgets, there has been an increasing trend of allowing users to directly interact with graphical encodings used in visual representations themselves *(e.g., [7], [13], [19], [48])*; see Figure 1-b. In this paper, we refer to this form of interaction as "*embedded interaction*". We define *embedded interaction* for visualization as a form of interaction that incorporates one or more *interactive graphical encodings* into a visual metaphor. We describe *interactive graphical encodings* as elementary encodings where the visual structure used to show the data value can be directly changed. For example, imagine a bar chart that enables users to directly change the height of bars. In this case, the visual metaphor (bar chart) adapts embedded interaction through interactive graphical encodings (height of the bars in a bar chart). Embedded interaction is used in various visualization techniques. The interaction design of these techniques requires users to directly scale the graphical encoding to perform higher level tasks, such as model steering and data querying.

Model steering is a method of interactively exploring data in visual analytic tools [13], [47]. Visual analytic tools often pass data through statistical models *(e.g., principal component analysis)* and visualize the computed structure of the dataset for the user. Thus, to explore different aspects of the data, users are required to interact with parameters of the model used for computing the structure. Several projects from the visual analytics community have adopted embedded interactions as a means of steering the parameters of

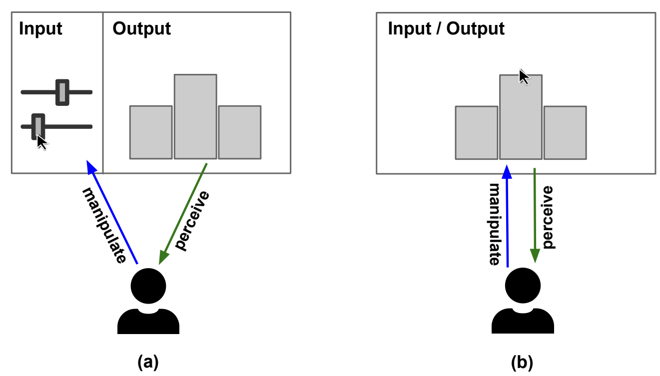


Fig. 1. Two different forms of interaction in many visualization systems. In order to interact with the visualization, users are required to either manipulate the external components in a separate panel (a) or directly manipulate the visual elements (b).

Fig. 2. In the Visualization by Demonstration paradigm [38], a user directly interacts with a point by making its size larger to demonstrate the interest in generating a visualization in which this point, and points like this, are larger. In response, the system extract data attributes that can be mapped to size and suggest them.

- *Bahador Saket, Arjun Srinivasan, and Alex Endert are with Georgia Tech. E-mail: {saket, arjun010, endert}@gatech.edu.*
- *Eric D. Ragan is with Texas A&M University. E-mail: eragan@tamu.edu.*

Manuscript received April 19, 2005; revised August 26, 2015.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2017.2680452, IEEE Transactions on Visualization and Computer Graphics

underlying models used in visualization tools *(e.g., [7], [13], [14], [19])*. For instance, InterAxis allows users to directly interact with the length of a bar in a bar chart to adjust the relative weight of data attributes in the system [19]. In InterAxis, attribute weights are shown using bar lengths next to the data attribute names. To adjust the weight assigned to an attribute, users adjust the length of the bar. For example, if the user wanted to indicate that the attribute "Price" was twice as important as its current value, the user would need to increase the length of the bar accordingly. This triggers a change in the underlying model used to compute the new axis for the scatterplot. AxiSketcher is another tool that allows users to revise nonlinear axes of scatterplot by direct interaction with graphical encodings [25]. Similarly, some systems allow users to adjust the distance between data items *(e.g., documents and glyphs)* to steer distance and similarity functions [7], [13], [14]. In each of these techniques, adjustment of the interactive graphical encodings implies an intent to change the result of a computation, rather than changing the data value directly.

Embedded interactions have also been used for data querying, as well as changing the parameters of visualizations for exploration. For example, DimpVis is a recent system that allows users to directly interact with the length, angle and position of the visual representations, as a means for temporal navigation [21]. In DimpVis, users can adjust the height of a bar to see its value at different moments in time. For instance, to check if at any point in time the value associated with a bar is half its current value, the user can drag the bar vertically downwards to compare its values at different points in time. Saket et al. [38] also introduced *Visualization by Demonstration*, in which users can directly interact with graphical encodings to provide visual demonstrations of incremental changes to the visual representation. For example, the user makes the size of a data point two times larger to demonstrate interest in generating a visualization in which this point and similar points are classified together and shown larger than other data points. In response, the system solves for data attributes that can be mapped to size and suggests the attributes. See Figure 2 for more details. Kondo et al. [22] also proposed Glidgets, a method that adapts embedded interaction for exploring and querying changes of elements in dynamic graphs. In general, this form of embedded interaction adjusts specific parameters of the data transformations and visual mappings to help users to explore their data.

The appeal of embedded interactions can be attributed to several factors. First, users do not need to shift their attention from the visual features of interest when interacting [21]. Secondly, users can make intuitive and direct visual adjustments without needing to understand the potentially complex system parameters being controlled [14]. Additionally, embedded interaction simplifies the visualization interface by obviating the need for additional control panels or widgets [26]

As more systems leverage graphical encodings in the visual representations not only to represent data visually but also to serve as the method for user interaction, this motivates the need to understand the effectiveness of interaction with these graphical encodings. While previous studies *(e.g., [8], [17], [27])* have contributed towards an understanding of perception of different static graphical encodings, the field lacks the knowledge of how different graphical encodings can serve as the basis for user interaction. Enabling embedded interactions for data visualizations requires a strong understanding of how direct interaction influences the ability to accurately control and perceive graphical encodings.

In this paper, we present a study of the effectiveness of 12 different interactive graphical encodings for magnitude production tasks [5], [51]. We conducted a within-subjects study in which participants performed magnitude production tasks *(e.g., change the value of the interactive graphical encoding to x% of its current value)*. Our results indicate that some interactive graphical encodings (e.g., *position*) are more effective than others *(e.g., shading/texture)* in terms of task completion time and accuracy. Finally, we analyzed users' interaction logs generated during each trial to gain a deeper understanding about the interaction cycles performed by each user. Since interactive graphical encodings foster a tight coupling between perception and manipulation, the interaction logs reveal insights about effectiveness beyond frequently-used completion time and error metrics.

The primary contributions of this paper are:

- A better understanding of interactive graphical encodings based on user interaction metrics (target re-entry and movement direction changes) proposed in previous work [30].
- Using interactive magnitude production to measure the effectiveness of 12 different interactive encodings and rank them based on task completion time and accuracy.

2 Background

Due to the rich body of research currently investigating embedded visual-centric interaction (also known as post-WIMP or post direct manipulation [26]), a wide variety of interaction techniques have been developed for—or have been applied to show—embedded interaction with graphical encodings. Our work builds on a strong research foundation of the perception of visual data encodings in the field of information visualization. Below, we discuss some of the most relevant studies on graphical perception and user interaction.

2.1 Psychophysics and Graphical Perception

Psychophysics is a research area that focuses on measuring the relationship between perceived and actual properties of an object [5], [51]. Most relevant to our study are the common psychophysics evaluation methods of magnitude estimation and magnitude production.

2.1.1 Magnitude Estimation

Magnitude estimation has been used in several studies to measure perception of different graphical encodings and how perceptual judgments impact the utility of visualizations [8], [17], [42], [44]. Estimating the proportion of part to whole of an object is the task usually used in this method to measure a user's visual perception of an object.

Previous work has used magnitude estimation to study the ability of viewers to accurately perceive the data values encoded using graphical encodings. Following previous researchers *(e.g., [8], [17])*, we use the term *graphical perception* to refer to this ability of accurately interpreting data values from visualizations. Simkin and Hastie [42] found that people perform different mental comparisons given specific visualizations. For example, individual bars in bar charts were often read by comparing a single bar to the height of all the bars. In contrast, individual slices in a pie chart were compared to other individual slices. Spence and Lewandowsky [44] also studied the graphical perception of bar charts, tables and pie charts for proportional comparison tasks. Their findings indicate that

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2017.2680452, IEEE Transactions on Visualization and Computer Graphics

when participants were asked to make comparison of combinations of proportions, the pie charts outperformed bar charts. Their results also show that for tasks were participants were asked to retrieve the exact value of proportions, tables outperform pie charts and bar charts. More recently, Skau and Kosara [43] assessed graphical perception of pie and donut charts in which data is encoded in three ways: arc length, center angle, and segment area. Their study indicated that angle was the least important visual cue for both pie and donut charts. In another study, Kosara and Skau [23] assessed several pie chart variations that are frequently used in Infographics including exploded pie charts, pies with larger slices, elliptical pies, and square pies. Their results indicated that people are less accurate at perceiving charts that distort the shape.

One of the most relevant studies for our research is that by Cleveland and McGill [8]. The study tested the graphical perception of 10 elementary graphical encodings (see Figure 3). They asked participants to visually compare values of two marks *(e.g., two bars of different lengths)* and estimate what percentage the smaller value was of the larger. They used the results to rank the graphical encodings; one elementary graphical encoding is taken to be more accurate than another if it leads to human judgments that are closer to the actual encoded values. Heer and Bostock [17] conducted a similar study to evaluate graphical perception. Their crowdsourced results validated the previously established graphical encoding rankings, and the authors discussed similar design guidelines for future work. Our study tests perception of graphical encodings similar to the studies by Cleveland and McGill [8] and Heer and Bostock [17]; however, rather than magnitude estimation with static images, our study requires interactive magnitude adjustment, which is of particular importance for embedded interaction.

Our work differs from previous work that used magnitude estimation mainly because we use magnitude production tasks in our study. In particular, we are interested in understanding the effectiveness of user interaction with the encodings rather than the how well we perceive their encoded values. Interactive adjustment of graphical encoding is different from perception alone. User interaction involves continuous manipulation and perception. One of the theories which describes this cycle is Norman's Action Model [33]. Execution is defined as taking an action to change something and evaluation is defined as perceiving the changes made. As Norman mentions, most interactions will not be satisfied by single manipulation and perception. There must be numerous sequences. For instance, a user might manipulate a length of a bar and perceive the value a few times before deciding on the final value.

Another main difference between our work and previous studies [9], [17] is that our use of the magnitude production tasks allows us to collect user interaction logs. Analyzing these logs helped measure the effectiveness of different interactive graphical encodings based on metrics that describe user interaction behaviors.

2.1.2 Magnitude Production

Magnitude production method requires a user to change the intensity of a graphical encoding in proportion to a reference point. The reference point can be the graphical encoding's initial value or the value of another element on the display. For example, adjusting the length of a bar to 10% of its current value would be an example of a magnitude production task.

Bezerianos and Isenberg [5] studied perception of three different graphical encodings (angle, area, and length) on wall-sized displays using a magnitude production task. Their study

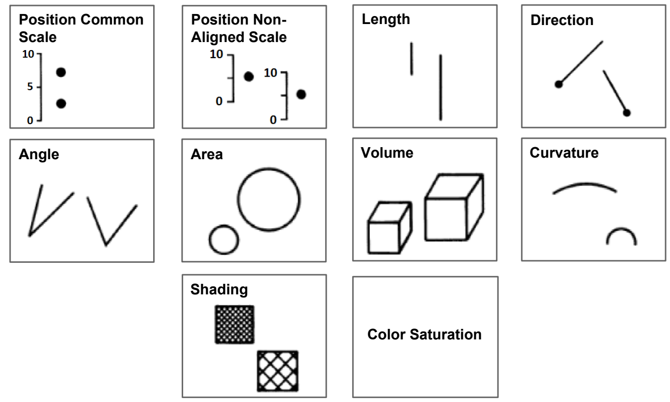


Fig. 3. Elementary graphical encodings studied by Cleveland and McGill [8] (these images were recreated and adapted based on [8]).

used wall-sized displays, and they asked participants to decrease the magnitude of a graphical encoding to match the magnitude of another graphical encoding at a distant region in the display. Participants changed the magnitude of the encodings using the UP and DOWN arrow keys of a keyboard. Their results showed participants' perception was mostly affected when they were close to the display. We similarly use a magnitude production task in our study to assess user interaction with 12 different interactive graphical encodings. However, we are interested in understanding user interaction with the graphical encodings where interactions are directly on the encodings.

2.2 Target Acquisition

Fitts' law [15] is one of the models of human movement that predicts the time required to quickly move to a target area when a target has a given size and distance. Variations of the law have been proposed to extend Fitts' law to two-dimensional tasks [28], [29]. These studies tested the performance of Fitts' law by requiring participants to perform target acquisition tasks in which participants had to move the pointer to the specified target on the screen.

For our study of embedded interaction, we considered using Fitts' law to model user interaction time with the graphical encodings. However, with embedded interaction, the target size may be determined by a function of the initial value of the graphical encoding rather than being not explicitly shown. Thus, interaction with graphical encodings could not be modeled using Fitts' law since there are no constant visual target dimensions (i.e., width).

2.3 User Interaction

In discussion of interactive graphics, Becker et al. [3] described direct manipulation and immediate change as the two core properties. In addition to direct input, Spence even included the notion of *passive interaction*, through which the user's mental model on the data set is changed or enhanced rather than the system or visual content being changed [36]. While finding a single agreed-upon definition of *interaction* is difficult, more specific *interaction techniques* can be less challenging to express and are more tangible concepts than the more nebulous concept of *interaction* itself [55]. Yi et al. [55] explain interaction techniques in information visualization as a set of tools that allow users to manipulate and interpret the data representations.

Information visualization is one domain that can directly benefit from interactive graphical encodings. For instance, Elmqvist

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2017.2680452, IEEE Transactions on Visualization and Computer Graphics

et al. discuss how good interaction design can foster effective "flow" through an interface [11]. Willett et al. [53] discuss how visual cues can be leveraged as part of interface controls to enhance user interaction. Additionally, as complexity and size of data sets expand, interactivity of information visualizations becomes increasingly important. For instance, Heer et al. [16] show how advanced interaction techniques for selection can help in constructing generalized queries. Many visualization tasks cannot be completed using static images alone. Interaction techniques in information visualization consist of a set of tools that allow users to manipulate and interpret the data representations [55]. The manipulation and interpretation occurs in a frequently iterating cycle previously described by Norman [33]. He describes the cycle with steps that include evaluating the state of a system, planning for an intended change, and executing actions intended to make that change happen. In the case of interactive data visualization systems, interaction techniques currently fall into two interaction designs: *graphical widgets* [41], [54] and *embedded interaction* [12], [21].

Data visualization systems usually contain graphical widgets and visual representations in two visually-separate panels. The control panel affords direct manipulation on graphical widgets, from which updated visualizations are shown. For example, a common interaction technique for filtering in many visualization systems is either selecting ranges via sliders or choosing particular values via check boxes in one panel and observing the resulting changes on the visualization in another panel [1], [45]. Historically, interaction with common widgets *(e.g., sliders, check-boxes, radio-buttons, and drop-down lists)* has been the norm for tasks like passing input parameters and filtering.

The concept of embedded interaction was first introduced by Andries van Dam [50]. The goal of his study was to make interfaces as invisible as possible and tighten the gap between a user's intent and the execution of that intent. More recently, there has been a trend towards using embedded interaction as a replacement for (or an addition to) old user interfaces in information visualization [26]. Lee et al. [26] reflected on advantages of embedded interaction techniques (as one of the interaction methods adapted to post-WIMP interfaces [2]) over the WIMP (Windows, Icons, Menus, and Pointers) techniques in information visualization. Rzeszotarski et al. [37] proposed Kinetica as an approach for multivariate data visualization on tablets. Kinetica applies embedded interaction techniques to accommodate the process of data exploration on multivariate data visualization. Results of their study indicate that embedded interaction helps users to explore multiple dimensions at once and to make more descriptive findings about their data set. As another example, Kondo and Collins [21] presented DimpVis, an interaction technique for effective visual exploration of time in information visualizations through embedded interaction. Another example of embedded interactions in information visualization is interactive map legends [35].

For visual analytics, many systems use complex statistical models that make user interaction more difficult [18]. In order to simplify user interaction in visual analytics systems, different studies applied embedded interaction. Endert et al. [12], [13], [14] have shown how similar approaches can be used to steer and train user and data models based on user interactions directly in the visualization. For example, changing the relative spatial distance between data items *(e.g., documents, images, or glyphs)* can be used to steer distance and similarity functions to re-arrange the spatial layout, retrieve additional data, and other analytic models [7], [13], [14], [48].

2.4 Formulating Embedded Interaction

We define *Embedded Interaction* as a form of interaction that allows users to directly manipulate the graphical encodings used in a visual representation. Interfaces using embedded interaction do not rely solely on additional graphical widgets (e.g., menus and check boxes) to specify commands. In the literature, the concept of embedded interaction is sometimes defined using different terminology. For example, Endert et al. defined it as *observation-level interaction* [14] and *semantic interaction* [13], and Kondo and Collins called it *object-centric interaction* [21].

Embedded Interaction is inspired by direct manipulation [40], which supports performing direct and iterative interactions on representation rather than through complex and abstract syntax. To describe embedded interaction we use the instrumental interaction model [2] that defines three properties (*degree of indirection, integration and compatibility*) to operationalize design and evaluation of interaction paradigms.

Overall, embedded interaction uses interactive encodings that have a low degree of indirection and high degree of compatibility. These encodings have low spatial indirection because the interaction instruments (handles) are superimposed on top of the graphical encodings themselves, so the distance between the instruments and the objects of interest is low. They also have low temporal indirection because manipulation of the instruments and changes to the encodings happen in real-time. Degree of compatibility of the interactive graphical encodings is high since the interaction instruments follow the movements of the cursor *(e.g., dragging handles)*. The degree of integration could vary depending on the design of the graphical encodings (how many degrees of freedom are used in construction and manipulation of the graphical encoding), and the input device used *(e.g., mouse, multitouch, etc.)*.

3 EXPERIMENT

We conducted a user study to achieve a better understanding of the issues raised in the previous section *(e.g., how users interact with graphical encodings and which are more effective for embedded user interaction)*. We studied interaction effectiveness (performance accuracy and time) for 12 interactive graphical encodings.

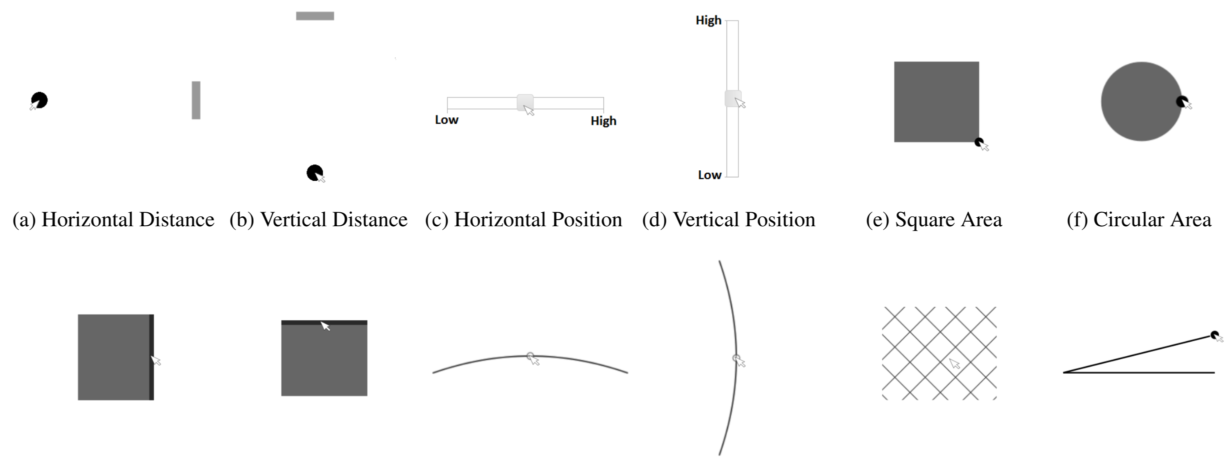
In an attempt to support more familiar and natural methods of user interaction, we chose to run the study as an online experiment so participants could use the setups and environments familiar to them *(e.g., their own machines with their own familiar input configuration)*. Previous work [17], [34] has validated the use of web experiments for user studies despite their limitations of experimental control.

3.1 Interactive Graphical Encodings

To study interactive graphical encodings, we first selected seven common elementary graphical encodings (following previous work [8], [17]) used to construct many visualizations today: *distance, position, length, angle, curvature, shading*, and *area*. We then developed 12 interactive versions of these graphical encodings by taking horizontal and vertical orientations into account for *distance, position, length* and *curvature*; see Figure 4. This section describes the types of interactive graphical encodings used in the experiment.

Distance (Horizontal and Vertical). This interactive graphical encoding contains a rectangle (a reference position) and a small circle as the controller (see Figures 4-a and 4-b). Participants could

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2017.2680452, IEEE Transactions on Visualization and Computer Graphics



(a) Horizontal Distance (b) Vertical Distance (c) Horizontal Position (d) Vertical Position (e) Square Area (f) Circular Area

(g) Horizontal Length (h) Vertical Length (i) Horizontal Curve (j) Vertical Curve (k) Shading (l) Angle

Fig. 4. The 12 interactive graphical encodings assessed in this study, designed based on seven common elementary graphical encodings used in data visualization: *distance, position, length, angle, curvature, shading*, and *area*. Interactive graphical encodings are elementary graphical encodings that can be directly manipulated or adjusted.

adjust the *distance* between the circle and the reference rectangle by dragging the circle with a mouse along a single dimension. This encoding is common in visualization systems that allow users to adjust the distance between visual elements where similar elements are spatially close to one another *(e.g., [13])*. For our analysis, we calculated the error of participants' responses by comparing the distance (in pixels) in the user's response to the expected response.

Position (Horizontal and Vertical). This interactive graphical encoding presents a horizontal or vertical slider to the participants (see Figures 4-c and 4-d). Variations of sliders are commonly used for filtering in different visualization systems. To interact, participants moved the *position* of the box at the center of the slider by dragging it with a mouse. While the *Position* and *distance* encodings are similar, we note a key difference between the two: the *position* encoding presents users with explicit low and high points, and it includes a visible one-dimensional scale in the background (the slider's scale). The primary reason for including both encodings was to see whether adding an explicit movement boundary (low and high points along with the background scale) affects user performance. For our analysis, to compute the error of participants' responses, we compared the user's position of the slider box on the scale versus the expected position.

Area (Rectangular and Circular). This interactive graphical encoding came in two variations: square and circle. Participants adjusted the *area* of the shape by dragging a small handle (tiny black circle) on the perimeter of the object; see Figures 4-e and 4-f. One of the applications of *area* manipulation is *rectangular brushing*, in which users select a subset of the data items by drawing a rectangle with an input device (examples can be found in the D3.js visualization library [6]). For our analysis, we compared the area of the user's object versus the expected area to compute the error of participants' responses.

Length (Horizontal and Vertical). This interactive graphical encoding involves re-sizing the *length* of a rectangle (see Figures 4-g and 4-h). Participants adjusted the *length* by dragging the right or top edge of the rectangle with a mouse cursor. Directly manipulating the length of a bar has been used as a method for filtering data *(e.g., [19])*. For our analysis, we compared the horizontal length (or height) of the rectangle (in pixels) versus the expected length to compute the error of participants' responses.

Curvature (Horizontal and Vertical). The implementation of this interactive graphical encoding is comprised of a curved line with a small circular handle at its center. Participants adjusted the *curvature* of the line by dragging the handle along a single dimension (horizontally or vertically); see Figures 4-i and 4-j. For our analysis, we compared the horizontal or vertical distance (in pixels) between the circle and the line segment between the end points of the curve versus the expected distance to compute error. Similar to Cleveland and McGill's experiments [9], we used the horizontal or vertical distance between the circle (mid-point of the curve) and the line segment connecting the end points of the curve as our measurement metric. For our analysis, we compared this value to the expected distance to compute error.

Shading. This interactive graphical encoding contains a rectangular area with cross-hatched shading (see Figure 4-k). Participants adjusted the *density* of the hatch pattern by dragging the mouse cursor up or down. This interaction was selected for consistency with the other interactive graphical encodings. *Shading* is often similar to color saturation for graphical perception [31], and these encodings are commonly used in many different types of visualizations, including infographics, choropleths, and heatmaps. For our analysis, we compared the number of cross-hatched rectangles in the object versus the expected number of cross-hatched rectangles in the object to compute the error in participants' responses.

Angle. This interactive graphical encoding contains two line segments that meet at an angle with a handle (a small black circle) at the end of one of the line segments (see Figure 4-l). Participants could adjust the inner *angle* between two lines by dragging the handle with a mouse. Angular representations are common in pie charts, and interactive angles could also be used in other forms of visualizations, as graphical perception of static angles has been shown to be fairly accurate [8], [17]. For our analysis of interaction accuracy with the angular encodings, we compared the inner angle (in degrees) between the line segments versus the expected inner angles to compute the error of responses.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2017.2680452, IEEE Transactions on Visualization and Computer Graphics

3.2 Hypotheses

Based on earlier work [4], [8], [17] and our own experiences, we considered the following hypotheses for our study:

- **H1:** We expected accuracy and interaction time to be different among different interactive graphical encodings. More specifically, we expected accuracy to be better and interaction time to be faster for *distance, position*, and *length* compared to *area* and *shading*. Prior research shows people can perceive *length* and *position* more accurately than *area, curvature*, and *shading* in static visualizations [8], [17]. We also expected that *curve* and *angle* would fall somewhere in the middle of the ranking for both accuracy and interaction time.
- **H2:** We hypothesized that accuracy of horizontal interactive graphical encodings would be higher than for vertical orientations. Research by Benner [4] found that humans are better at estimating *position, distance*, and *length* of objects that are oriented horizontally, as compared to those with vertical orientation. Thus, we decided to include both horizontal and vertical orientation for each interactive graphical encoding when applicable for the graphical encoding type (that is, some types did not have natural horizontal and vertical variations).
- **H3:** We hypothesized that when interacting with a graphical encoding, patterns of interaction behavior would correspond to different degrees of accuracy. This hypothesis is based on the idea that users would adjust values more frequently when having more uncertainty or difficulty in graphical perception. A high number of directional changes might indicate an inability to estimate the represented value of the interactive graphical encoding. To capture such interaction behavior, we adapted a metric called *movement direction change* (MDC), which was introduced in previous work as a means of studying pointing interactions [30]. We explain the MDC metric in the "Interaction Effectiveness Results" section.

3.3 Participants

The study was conducted online by invitation to students at a single university. Of the 46 participants who began the study, 35 completed the study (22 male, 13 female). Ages ranged from $18-34$ years. Participants were mostly undergraduate and graduate students in science and engineering programs, and they were familiar with plots and computers. The participants were provided with the URL and could participate in the study using any device. Participants who completed the study were compensated with a $5 Starbucks gift card. In addition, the three participants with the most accurate and fastest responses were given a $25 gift card.

We also collected logs containing users' operating systems and input devices. Participants used different operating systems (20 Mac OS, 11 Windows, and 4 Linux users) to participate in our experiment. Moreover, 18 of the participants used a mouse and the rest used a trackpad to adjust the interactive graphical encodings.

3.4 Task

Each interactive graphical encoding was accompanied by instructions that required the participant to adjust the interactive graphical encoding to a target value. A target value is a certain percentage that we asked each participant to adjust the interactive encoding to. For example, for the *length* encoding, we asked participants to adjust the length to 150% of its current value. Participants could adjust the graphical encodings' values by directly manipulating them, as described previously.

In a pilot study, participants reported sometimes losing track of the starting value for the question while performing a task. To address this feedback, we made sure the interface for the experiment always showed the initial value as a reference point while users interacted with encodings. Since the order of encodings and target values was randomized, this reference point helped users to keep track of the initial position for the given encoding. The initial value was shown as a semi-transparent reference point for all the graphical encodings except shading (see Figure 5). For shading, we showed two shadings side by side, where the right side always showed the initial value, and the the left side was the one that the participants could interact with.

Our task resembles a magnitude production task [51] (as described in Section 2.1). This task is motivated by the fact that while users manipulate a visual element on the interface *(e.g., position of a knob on a slider)* they constantly compare its current value to a reference point [10]. In our study, the reference point is the reference value *(i.e., the starting value encoded)*.

3.5 Training Procedure

At the beginning of the study, participants were briefed about the purpose of the study and their rights. They then were instructed how to complete the experiment.

In order to familiarize the participants with the graphical encodings, interactions, and questions, participants first completed 12 practice trials (one trial per interactive graphical encoding). Each trial included the task description *(e.g., make the inner angle between the two lines 200% of its current value.)* and the interaction instructions *(e.g., drag the black circle to move the line)*; see Figure 5-Left. To provide feedback after completing each trial, participants were shown a visual comparison between their response and the correct answer for each trial; see Figure 5-Right. Thus, the task description and training showed the participants how to perceive and manipulate each encoding.

3.6 Experimental Procedure

Participants performed seven trials for each of the 12 versions of interactive graphical encodings, and each trial had a different target value (25%, 50%, 75%, 125%, 150%, 175%, and 200%). Participants performed 84 tasks (12 interactive graphical encodings $\times$ 7 trials) with randomized task order. Current value (starting point) of all interactive graphical encodings was 100%.

After completing the practice trials, participants began the main experiment with the 84 randomized trials. For each question, we logged interaction time and the changes in accuracy made every millisecond. Interaction time started as soon as participants started interacting with an interactive graphical encoding. A screenshot of the experiment's interface is shown in Figure 5-Left.

4 Task Performance Results

In this section, we first describe the methods used to analyze the data collected from the experiment. We then provide an overview of our results, with more detailed quantitative results listed in Figure 6. The collected data has 2940 answers (84 trials $\times$ 35 participants). We measured both interaction time and accuracy for each trial. Interaction time was measured by computing the total time each participant spent interacting with a primitive. Accuracy percentage was measured by subtracting the percentage of response error from 100, where the response error is:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2017.2680452, IEEE Transactions on Visualization and Computer Graphics

$$Error = \frac{|Response\,Value - Expected\,Value|}{Expected\,Value} \times 100$$

To account for data quality from online data collection, outlier handling was performed to account for trials where participants were likely to have disruptions or mistakes that were greater than would be expected with a usual attempt. For instance, trials having very long completion times were excluded because users likely did not spend the entire duration performing the single task in such cases. We excluded 268 (9%) of the collected responses as outliers based on interquartile range (IQR), where an outcome was considered an outlier if it was more than 1.5 times the size of the IQR away from either the lower or upper quartiles. The outlier distribution of the 9% of trials was spread across encoding type (2.1% shading, 1.9% area, 1.6% curvature, 1.1% length, 0.9% position, 0.7% distance, and 0.7% angle). To some extent, more outliers were associated with encodings with lower performance, but the variation was not extreme. We applied the outlier removal procedure for each encoding separately.

4.1 Task Performance: Data Analysis

To address our first two hypotheses, we needed to test how the different interactive graphical encodings (**H1**) and differences in adjustment orientation (horizontal or vertical, as described in **H2**) affected the performance outcomes of interaction time and interaction accuracy. We provide all relevant materials for this study online [1]: software for running the experiment, anonymized results, and statistical test results.

To analyze the differences among the various interactive graphical encodings, we first calculated separate mean performance values for all trials. That is, for each participant, we averaged outcome values of trials for each interactive graphical encodings. To test effects due to orientation, performance outcomes for each level (horizontal and vertical) were averaged for the trials of each interactive graphical encoding with the appropriate orientation. Adjustment orientation was only varied for four graphical encoding types (*distance*, *position*, *length*, and *curve*).

To test the combined effects of interactive graphical encodings and adjustment orientation, we would ideally turn to a two-way factorial analysis of variance (ANOVA). However, because adjustment orientation was only variable for a subset of the graphical encodings, a factorial analysis was not appropriate for the unbalanced design. As an alternative, we conducted a one-way repeated-measures ANOVA to test for differences among the various interactive graphical encodings, and a separate two-way repeated-measures ANOVA to test for interactions between interactive graphical encodings and adjustment orientation for the subset of encodings that had horizontal and vertical versions.

Before testing, we checked that the collected data met the assumptions of appropriate statistical tests. The assumption of normality was satisfied for parametric testing, but Mauchly's Test of Sphericity indicated that the assumption of sphericity had been violated for both accuracy and speed. To address this issue, we report test results with corrected degrees of freedom using Greenhouse-Geisser estimates for $\varepsilon < 0.75$ and otherwise with Huynh-Feldt correction.

1. http://va.gatech.edu/encodings/

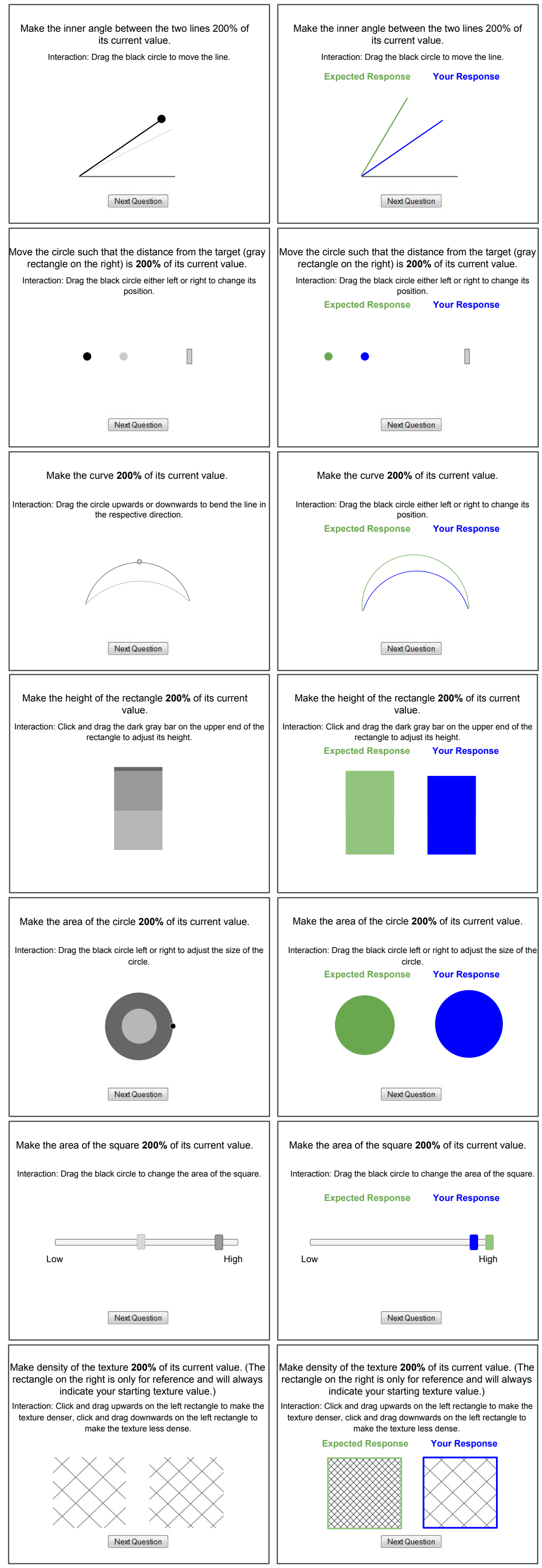
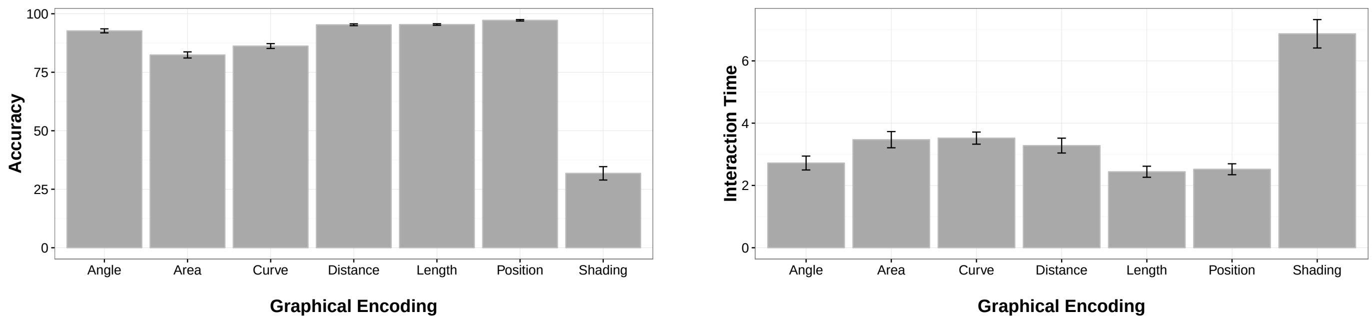


Fig. 5. Each row shows two screenshots from trials in the training phase. The left side shows the initial representation with instructions (all a 200% increase adjustment in this image), and the right image shows the interface after each trial during the training session, where participants were shown a visual comparison between their response and the correct answer.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2017.2680452, IEEE Transactions on Visualization and Computer Graphics



Test of Within-Subjects Effects	
Interactive Graphical Encodings ($F_{(1.7,58.5)} = 401.5, p < 0.001$, $\eta_p^2 = 0.92$)	
Pairwise Comparisons (ranked from most accurate to least)	
p values are corrected using Bonferroni correction.	
Graphical Encodings	
Position **vs.** Angle, Area, length, Curve, Distance & Shading	($p < .01$)
Length **vs.** Angle, Area, Curve and Shading	($p < .01$)
Distance **vs.** Angle, Area, Curve & Shading	($p < .01$)
Angle **vs.** Area, Curve & Shading	($p < .01$)
Curve **vs.** Shading	($p < .01$)
Area **vs.** Shading	($p < .01$)

(a)

Test of Within-Subjects Effects	
Interactive Graphical Encodings ($F_{(2.9,100.0)} = 95.2, p < 0.001$, $\eta_p^2 = 0.73$)	
Pairwise Comparisons (ranked from fastest to slowest)	
p values are corrected using Bonferroni correction.	
Graphical Encodings	
Length **vs.** Area, Curve, Distance & Shading	($p < .01$)
Position **vs.** Area, Curve, Distance & Shading	($p < .01$)
Angle **vs.** Area, Curve, Distance & Shading	($p < .01$)
Distance**vs.** Shading	($p < .01$)
Curve **vs.** Shading	($p < .01$)
Area **vs.** Shading	($p < .01$)

(b)

Fig. 6. Performance results for different interactive graphical encodings along with statistical test results. Mean accuracy is shown in (a), and mean interaction time is shown in (b). Error bars represent standard error.

4.2 Task Performance: Results Overview

In this section, we organize the results of the statistical tests by independent variables and interactions.

Interactive Graphical Encodings. We found significant main effects for both accuracy and time for encodings, and we followed up with Bonferroni-corrected posthoc comparisons; see Figure 6.

Figure 6-a shows accuracy by interactive graphical encoding type. *Position* has the best and *shading* has the worst accuracy. Accuracy of *position* was significantly better than all other interactive graphical encodings. However, Figure 6-a shows that practical advantages are notably small for *position* over *length* and *distance*, even though standardized effect sizes are high (Cohen's $d = 0.84$ between *position* and *length*, and $d = 0.91$ between *position* and *distance*). Pairwise comparisons did not detect significant differences among *length, distance*, and *angle*. In other words, *length, distance* and *angle* were interpreted with similar accuracy. We also found that *shading* was significantly less accurate than all other encodings. Moreover, *area* and *curve* fall somewhere in the middle in terms of the accuracy ranking.

Participants had the fastest interaction times using *length*, *position*, and *angle*, respectively. Although results of pairwise comparisons did not show significant difference among the three interactive graphical encodings, they were significantly faster than *area, curve, distance*, and *shading*. *Curve, distance*, and *area* were in the middle in terms of time. Results indicate that ranking of the interactive graphical encodings by accuracy is slightly different from the ranking based on interaction time. Rankings of the encodings for both accuracy and interaction time are shown in Table 1. *Position*, *length* and *angle* are among the best and *shading* is the worst in term of both accuracy and interaction time. More details are shown in Figure 6.

Adjustment Orientation. The tests failed to detect significant main effects of adjustment orientation for either accuracy ($F_{(1,34)} = 0.7, p > 0.05$) or interaction time ($F_{(1,34)} = 6.6, p > 0.05$); therefore, the results do not serve as evidence for interaction performance being influenced by horizontal or vertical orientation.

Interactive Graphical Encodings $\times$ Adjustment Orientation. There was a significant interaction between graphical encodings and adjustment orientation for both accuracy ($F_{(1.7,58.5)} = 4.7, p < 0.5$) and interaction time ($F_{(2.5,87.6)} = 17, p < 0.05$). While participants had more accurate interactions for the vertical versions of *length*, *curve*, and *position*, accuracy was lower for the vertical *distance*. In terms of time, participants were faster with vertical *position* and *distance* than the horizontal versions. This was opposite for *length* and *curve*; participants had a slower interaction with vertical *length* and *curve* than their horizontal versions.

4.3 Task Performance: Discussion

Table 1 shows rankings of the interactive graphical encodings based on the different metrics assessed in this paper alongside rankings of graphical encodings provided by Cleveland and McGill [8]. In each column, interactive graphical encodings are ranked from best to worse according to performance in each metric. For example, *position* has the best and *shading* has the worst accuracy in our study. Unlike the study by Cleveland and McGill [8], we did not include some graphical encodings such as *volume*, *color* and *direction*. Using *volume* is not recommended in many visualizations due to confusion that this type of graphical encoding might cause [49]. Similar to previous work [17], we excluded *color* mainly because we lacked control over participants' display configurations in the online study.

In our ranking, accuracy of *curve* was not significantly different from *area*. Note that this was a different result as the ranking provided in previous work (see Table 1), which found *area* to be more accurate than *curvature*. While average accuracy of *curve* was higher than *area* in our ranking, the pairwise comparison did not indicate a significant difference between their accuracy. Additional testing would be required to determine the ordering or equivalence between these two encodings. As previous work [17] discusses, the study by Cleveland and McGill did not find a significant difference between *length* and *angle* encodings (as psychophysical theory would predict [8], [52]). However, the results of our study found

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2017.2680452, IEEE Transactions on Visualization and Computer Graphics

TABLE 1
Ranking of the interactive graphical encodings based on completion accuracy and interaction time. Rows indicate significant differences between encodings.

Our Study		Cleveland & McGill [8]
Time	Accuracy	Accuracy
Length, Position, **Angle**	Position	Position
Distance, **Curve**, **Area**	Length, Distance	Length, Direction, Angle
Shading	**Angle**	Area
	Curve, **Area**	Curve, Volume
	Shading	Shading, Color

a significant difference between these two encodings in terms of accuracy.

4.4 Bias Analysis

We conducted chi-squared tests to check whether user interactions with different encodings were biased towards overestimation or underestimation (see Table 2). For each encoding, we ran separate tests for trials asking for increasing values and for those requiring decreasing values. For these tests, we excluded responses with exact accuracy for the level of precision in data collection.

The results show that there are significant effects in responses being biased towards either over or underestimation—particularly for responses where participants were asked to decrease an encoding's value, where significant effects were detected for all encodings. When participants were asked to increase the values, significant response biases were observed for 3 out of the 7 encodings (*shading*, *curve*, *area*)—the encodings having lowest overall accuracy. For example, *area* is the only encoding with an underestimation bias for increasing the value. This could be explained by the fact that increasing the value in *area* is not a linear increase but a squared increase. Among all encodings, shading has the highest skew towards under or over estimation, which is likely related to the ineffectiveness of the encoding. For responses where participants were asked to decrease the value of the *shading* encoding, all the responses underestimated the expected value. While these results indicate that bias is important when exploring the effectiveness of interactive graphical encodings, further studies will be needed to fully understand what causes these biases.

5 Interaction Effectiveness Results

We used line charts to visualize the collected interaction logs for each interactive graphical encodings; see Figure 9. The red lines show the target value that participants were trying to match with the interaction. The small dark dots indicate the final value for each participant at the end of the trial.

We only include logs for tasks with the target value of 200% in the paper, but log visualizations for all tasks are provided online [2]. In Figure 9, we scaled the horizontal axis to 10 seconds and the vertical axis to 300% for all interactive graphical encodings for the sake of readability and comparability. In addition, we note that outlier trials were not included in the log charts, as outliers were removed as described in the previous section.

2. http://va.gatech.edu/encodings/

TABLE 2
Percentages of overestimated and underestimated responses when increasing or decreasing values using different encodings. Chi-squared tests compared frequencies of overestimated and underestimated responses to test for directional response bias. Significant differences are indicated by star ($\star$).

Encoding	Direction	Over	Under	Chi-squared Test
Angle	Decrease	60.6%	39.4%	$\chi^2 = 14.4$, $p < 0.001$ $\star$
	Increase	53.6%	46.4%	$\chi^2 = 3.7$, $p = 0.07$
Area	Decrease	32.4%	64.8%	$\chi^2 = 36.1$, $p < 0.05$ $\star$
	Increase	35.7%	62.5%	$\chi^2 = 24.3$, $p < 0.05$ $\star$
Curve	Decrease	32.4%	57.6%	$\chi^2 = 13.3$, $p < 0.001$ $\star$
	Increase	56.1%	38.6%	$\chi^2 = 16.3$, $p < 0.001$ $\star$
Distance	Decrease	56.2%	41.8%	$\chi^2 = 10.7$, $p < 0.05$ $\star$
	Increase	48.0%	51.0%	$\chi^2 = 2.4$, $p = 0.11$
Length	Decrease	40.5%	51.4%	$\chi^2 = 9.2$, $p < 0.05$ $\star$
	Increase	43.9%	49.6%	$\chi^2 = 3.7$, $p = 0.06$
Position	Decrease	56.0%	42.5%	$\chi^2 = 5.1$, $p < 0.05$ $\star$
	Increase	43.2%	40.0%	$\chi^2 = 0.2$, $p = 0.59$
Shading	Decrease	0.0%	100%	$\chi^2 = 105$, $p < 0.001$ $\star$
	Increase	80.7%	19.3%	$\chi^2 = 52.8$, $p < 0.001$ $\star$

5.1 Interaction Behavior: Data Analysis

To analyze interaction behavior, we considered *target re-entry* (TRE) and *movement direction change* (MDC). While we briefly describe these metrics and discuss their meaning for our study, MacKenzie et al. [30] explain the metrics in more detail. Table 8 shows the means and standard deviations of the interaction behavior metrics (TRE and MDC), interaction time and accuracy for all interactive graphical encodings. We averaged the horizontal and vertical adjustments.

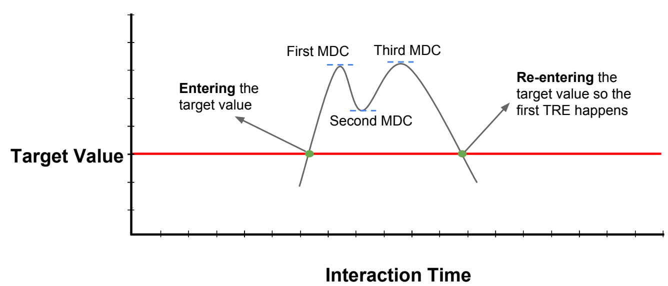


Fig. 7. This indicates part of a line chart used to visualize the interaction log for a particular user. This part of the interaction log enters the target value, leaves, and re-enters once. In this case, there is one target re-entry (TRE), and three movement direction changes (MDC).

Target Re-entry. During an interaction, if a user enters the target value, leaves, and then re-enters, this is an instance of TRE; see Figure 7.

Movement Direction Change. As it is shown in Figure 7, an instance of MDC occurs when a user changes the direction of the interaction. Figure 7 shows value selection over time with respect to the target value.

In order to get the final TRE and MDC values for each *interactive graphical encoding*, we divided the number of times each behavior happened by the total number of participants. We excluded outliers from this analysis.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2017.2680452, IEEE Transactions on Visualization and Computer Graphics

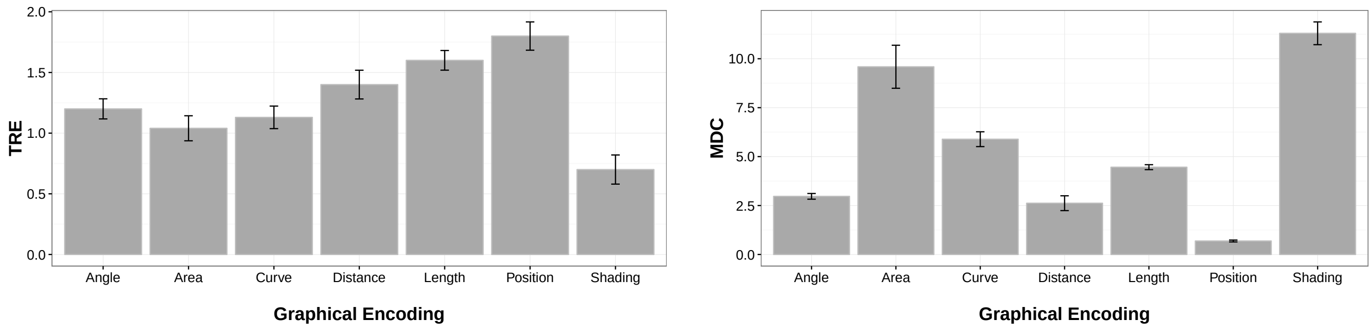


Fig. 8. Means and standard deviations of TRE and MDC for each *interactive graphical encoding*. The units in this table are "mean count per participant" for TRE and MDC. Error bars represent standard error.

5.2 Interaction Behavior: Results Overview

Our analysis of interaction behaviors revealed that, overall, the encodings with high accuracy (*distance*, *position*, and *length*) have smoother interaction patterns compared to *shading*, *area*, and *curve*. For the encodings with high accuracy, participants started by making large changes early, and then they made small changes while they were getting closer to the correct value; see Figure 9.

The charts for curves (Figure 9-h, i) demonstrate a surprising degree of consistency of error, with the vertical positions of dots showing many participants adjusting the value to 160% instead of 200%. This could suggest underestimation of quantitative representations with curves. Another finding is how participants adjusted values while working with shading (see Figure 9-l). Many participants ended with their final values lower rather than higher than the starting point, which suggests that their interpretation of the direction corresponding to "increasing" the value was probably the opposite of the implementation (and the version shown in the practice/instructions). This analysis also reveals a probable reason why accuracy was so poor with shading. This provides information about how inconsistent people can be in mapping shading to quantitative values, and it could suggest different groups of interpretation (such as participants moving in one direction or another). It is important to note that even if adjusting the calculation of accuracy to account for supposed alternative targets, the accuracy would still be extremely poor, and the rankings would remain unchanged.

To determine whether new accuracy metrics are related to completion time and accuracy, we first calculated the correlation between TRE, MDC, completion accuracy, and interaction time. Our results indicate that there is a strong negative correlation between accuracy and MDC (Pearson's $r_{(7)} = -0.76, p < 0.05$), which means the higher the accuracy, the lower the number of directional changes in users interaction. This confirms hypothesis H3. We also found a strong positive correlation between accuracy and target re-entry (Pearson's $r_{(7)} = 0.78, p < 0.05$). This means the higher the accuracy, the more times the users pass and re-enter the expected value. A possible explanation for this might be that for encodings that exhibit a high bias (Section 4.4), there are fewer target re-entries because participants form a mental target that is below/above the target value.

Finally, we found that each of the behavior metrics were strongly correlated with interaction time (Pearson's $r_{(7)} = 0.85, p < 0.05$). This means the longer the interaction time, the higher the number of movement direction changes.

We summarize the findings of this section as following:

- **More** movement direction changes result in **lower** accuracy and **longer** interaction time.
- **More** target re-entries result in a **higher** accuracy.

6 Discussion

Designers might find ranking of one metric more important than another depending on their requirements. As an example, one might argue that the accuracy of an interactive graphical encoding plays a more important role than interaction time. Depending on the application of the visualization, designers might take into account one or several of these rankings while designing an interactive visualization. While we do not claim that making design decisions based on completion time and accuracy metrics is wrong, we emphasize that looking at metrics computed based on user behavior during the interaction cycle *(e.g., TRE, MDC)* can be helpful as well. Comparing interactive graphical encodings based on several metrics might help designers have a more holistic view of how well embedded interactions might work with certain encodings.

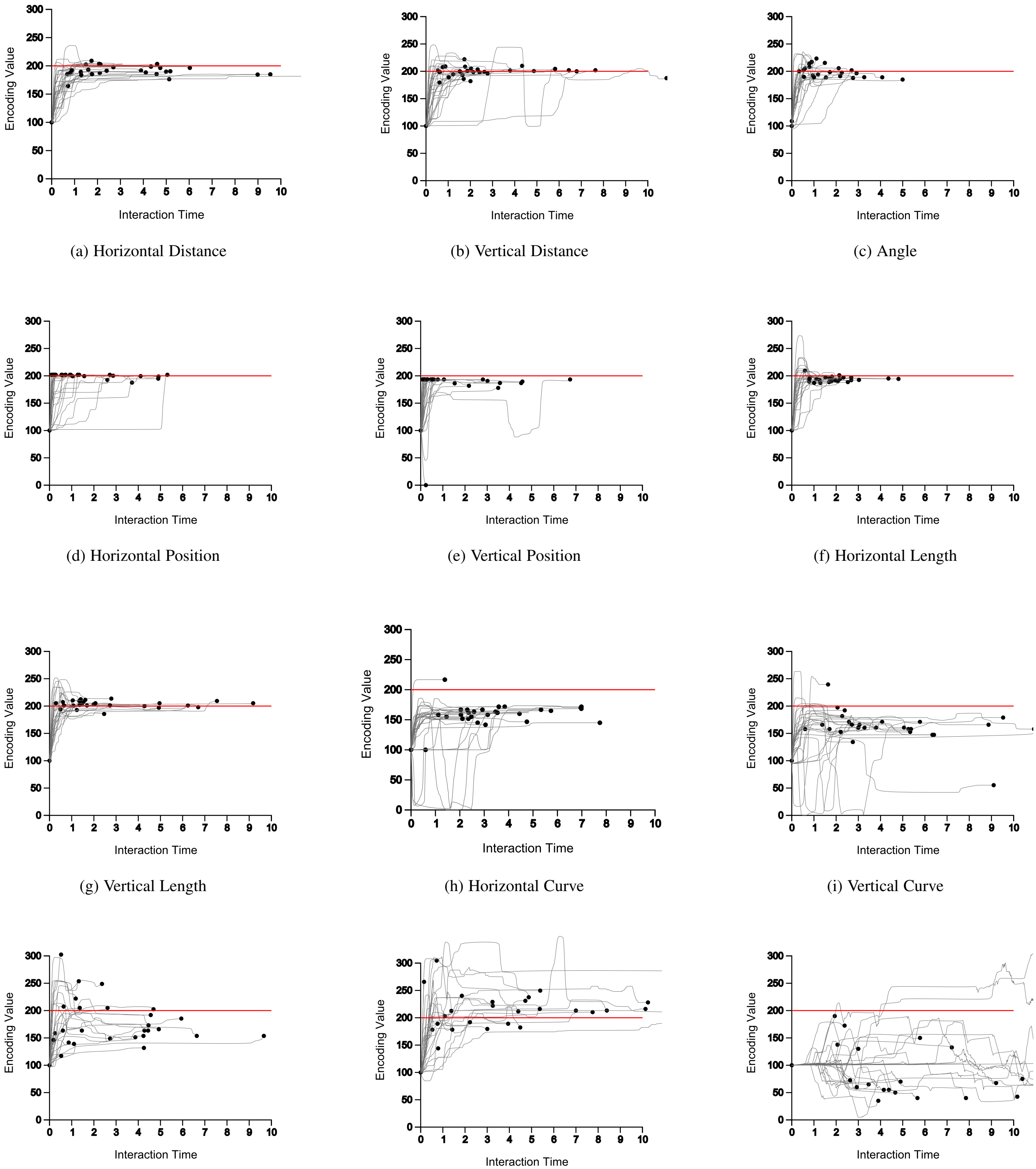
6.1 Incorporating the Interactive Graphical Encodings

If the decision is made to adapt the interactive graphical encodings in a visualization system, we suggest the following guidelines.

Making encodings interactive requires careful design considerations. Not every encoding used in a given visualization needs to be interactive. In cases where the chosen visual representation requires the use of an encoding with low performance, perhaps the use of traditional control panels for interaction is the better design decision. For example, visual representations that use *shading* or area as the primary method to encode data may be augmented with control panels to control the filtering or querying rather than embedded interaction *(e.g., geospatial choropleth maps)*. Instead, visual representations that use effective encodings lend themselves better to incorporating interactivity directly on the encoding.

Provide additional feedback if accuracy is important. Providing additional feedback might be helpful to improve the performance of specific encodings. For example, during embedded interaction with *shading*, interaction performance might be improved by also showing exact values via textual overlay. Additionally, we could highlight the aspects of the encodings that contribute to the value change. For example, for *angular* encodings, we could highlight the angle subtended or the height between the two arcs. Similarly, for *area* encodings, we could highlight the width and height of the square to show the squared value. While we did not test the effectiveness of such potential design improvements in our study, these considerations could be of interest for future design and evaluation efforts.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2017.2680452, IEEE Transactions on Visualization and Computer Graphics



(a) Horizontal Distance (b) Vertical Distance (c) Angle

(d) Horizontal Position (e) Vertical Position (f) Horizontal Length

(g) Vertical Length (h) Horizontal Curve (i) Vertical Curve

(j) Circlular Area (k) Square Area (l) Shading

Fig. 9. Representations of interaction logs for 12 interactive graphical encodings assessed in this study. The X axis (interaction time) is per second and the Y axis (encoding values) is based on percentage. All participants were asked to manipulate each interactive graphical encoding to 200% of its current value. All current values are shown as 100%. The small dark circles show the final point of each interaction log. Each line in the charts represents one interaction log for a participant who completed the task using the specific encoding. The same set of participants interacted with all encodings.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2017.2680452, IEEE Transactions on Visualization and Computer Graphics

6.2 Applications of Our Findings

In information visualization and visual analytics, the results of this study can be applied to inform the design of interactive legends [24], [35]. Interactive legends are controls that allow users to select or filter data by directly interacting with the graphical encodings used on the legends [35]. With the knowledge gained from this study, we suggest using the graphical encodings that have high accuracy *(e.g., length)* while designing interactive legends. Alternatively, legends using encodings with lower accuracy can provide additional feedback to users (e.g. textual values) to improve the accuracy of interaction. Another approach could be to resort to more conventional user interface widgets to perform tasks like filtering.

Another set of applications that could leverage the results of our study are graphical editing tools (e.g., Adobe Photoshop and Illustrator) and visualization authoring tools (e.g., Lyra [39], Data-driven Guides [20]). Our findings can assist design decisions about where interactions must be enabled on the graphical encodings versus where additional widgets may be required. For example, to allow users to create a rectangle with a specific texture, these tools could let users adjust the dimensions of the rectangle using embedded interaction and provide additional widgets on a separate control panel.

6.3 Interaction Combines Perception and Manipulation

Although the methodology used in this study is different from that by Cleveland and McGill [8] due to our use of interactive magnitude adjustment, our ranking of the interactive graphical encodings produced a similar ranking. At a high level, our ranking follows that of the prior studies, with the exception of our results indicating a significant difference between length and angle (in terms of accuracy). An explanation for this similarity may be that manipulation and perception are not mutually exclusive, and input from perception continually influences interaction. Thus, the performance of interaction with an encoding might be connected to the perception of the encoding itself. If an encoding supports sheer perception well, it would also support interactivity well.

One possible follow-up research direction includes quantifying the distribution of how much of an effect both perception and manipulation have while interacting with a graphical encoding. To do so, the study design would need to directly control for, and decouple, perception from interaction. For example, this might involve shielding the participants' line of sight for the encoding they are asked to manipulate. However, this seems to be at odds with the design guidelines of embedded interaction, where users directly interact with handles superimposed on the graphical encodings Thus, performing a study where perception is intentionally excluded may limit the applicability of the results to informing the design of embedded interaction for visualization. However, the results of such a controlled study would reveal knowledge about the perception has on interaction.

6.4 Indirection, Compatibility, and Integration

The graphical encodings used in our study have different degrees of compatibility, indirection, and integration [2]. *Position*, *length*, *angle* and *distance* have low degrees of integration and indirection, and high degrees of compatibility. Thus, these encodings are more efficient than others encodings that have higher degrees of indirection and integration, and lower degrees of compatibility.

The differences in degrees of compatibility, indirection, and integration among various encodings may affect their performance. In particular, having a higher degree of indirection and lower degree of compatibility might decrease the performance of an encoding. One interesting avenue for continued research could be the investigation of effects of the parameters of this Instrumental Interaction framework proposed by Beaudouin-Lafon [2] on the performance of the encodings.

6.5 Confidence Initiation

We found interesting patterns by visualizing interaction logs *(e.g., Figure 9)*. In some log visualizations, participants started making changes with high variation at first, then they considerably reduced the variation of the changes as they narrowed down on their final values. We can consider the interaction bahaviors and confidence initiation findings with respect to Fitts' Law, which is often used when describing the tradeoff between speed and accuracy during target selection [28]. Fitts' Law can describe how multiple successive movements *(e.g., fluctuations in increasing and decreasing value adjustments)* are likely to be more common before an expected termination point is known or expected. In our scenario, the adjustment "noise" will significantly diminish as the user approaches confidence of the intended value. For the interpretation of the interaction behaviors in our study, we refer to the point where participants started making changes with small variations as the *confidence initiation* point. In other words, *confidence initiation* is the point when coarse adjustments end, and participants are close enough to the target value for finer adjustments. There are several noticeable findings here:

- Overall, for the interactive graphical encodings with higher overall accuracy, participants came to the confidence initiation point faster than with the other encodings.
- It took participants longer to come to the confidence initiation point using *curve*. Refer to Figures 9-h and i. Interestingly, participants' interaction logs for both horizontal and vertical *curve* ended by converging somewhere below the real target value (the red line). This suggests a mismatch between perceived and actual values represented by the encoding.
- Interaction paths for *area* and *shading* either do not converge (e.g., *area*) or the variation in their changes does not decrease (e.g., *shading*). This could mean that participants never reached a point where they felt confident about the changes they were making. In other words, they did not know whether the changes they made were correct. Another possibility is that they felt the need to test a wide range of options with the interactive graphical encoding before quickly deciding on the final setting.
- In the first half of a second, participants made changes with high variation using all interactive graphical encodings except *shading*. Looking at Figures 9-l, it seems that participants did not make many changes at the beginning of their interaction with *shading*. This delay in interaction with *shading* might be because participants spent that period of time thinking of a correct way to map degree of shading/texture density to a quantitative value.

7 Limitations

Our results should be interpreted in the context of the specified encodings, adjustment orientations, target values, and tasks. We wanted to first gain a basic understanding of the rankings for simple interactive graphical encodings to see if and how they are different from the graphical perception results from prior studies [8], [17].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2017.2680452, IEEE Transactions on Visualization and Computer Graphics

7.1 Lack of Control for Physical Devices

Since our study was online, we did not have control over users' physical devices. This decision was intentional so that participants could use input devices that they were familiar and comfortable with, but it also allows the possibility of effects due to system differences. We did record participants' operating system types and input devices, and we tested for effects using t tests. The results did not indicate a statistically significant effects due to mouse and trackpad for either accuracy ($t_{(33)} = 0.08$, $p = 0.93$, $power = 0.72$) or interaction time ($t_{(33)} = 0.49$, $p = 0.06$, $power = 0.38$). The near-significant trend in time due to interaction device reinforces the need to study the effect of interaction device in future studies.

We also did not find a significant effect of operating system for either performance time ($F_{(2,32)} = 3.02$, $p = 0.07$, $power = 0.95$) or accuracy ($F_{(2,32)} = 0.69$, $p = 0.51$, $power = 0.93$). Unfortunately, our collected logs did not contain information about participants' browser types and screen sizes; we suggest that future interaction-related online experiments take these two factors into account.

7.2 Limited Training

To perform the tasks in this study, participants had to first estimate a percentage of change needed and then adjust the graphical encodings accordingly. However, estimating the amount of changes required for some encodings *(e.g., curvature, shading)* might be harder and more ambiguous than others *(e.g., length, distance)*.

The ambiguity of the tasks might have been lowered if participants had been trained prior to the primary trials. Our study included an instructional phase in which participants were required to perform a set of trial tasks, and they were given feedback after completing each trial. The system showed them their accuracy (visually and percentage) compared to the correct response. However, we did not enforce or control participant accuracy before continuing to the main trials. For instance, an alternative approach would have been to have participants perform trials until they achieve a given success rate with each encoding. Since we did not do this, it could potentially explain the low accuracy for some encodings such as shading, area, and curvature.

8 Future Work

Another interesting factor for further study could involve consideration for different user methods for judging graphical representations. In previous work, Talbot et al. [46] indicated that people might use different approximation methods to make judgments of a graphical encoding. More specifically, they found that people use either inner angle or height approximations when making slope judgments. During the trial session of our study, the task description and training showed participants how to perceive and manipulate each encoding. However, we did not explicitly control the approximation methods participants used to make judgments of individual graphical encodings in our study. It could be interesting for future work to investigate which approximation methods people use to perceive each of the encodings. In future studies, it might be interesting, for instance, to use eye-tracking during participant trials to contribute more insight about where participants look when making value adjustments, and that might help us to better understand how participants are perceiving values.

Matejka et al. [32] recently studied the effects of slider appearance to understand trade-offs between bias, accuracy, and speed-of-use. Their findings suggest providing dynamic feedback on the slider handle if a task requires precision. As part of future work, it would be interesting to explore how the appearance of a representation affects the interaction with corresponding encodings.

Another research avenue could be exploring the study of different types of input devices and mechanisms *(e.g., touch and multi-touch instead of mouse and trackpad)*. Different input devices involve different physical motions. Though we did not detect evidence of effects due to input device in our study, the study was not designed to focus on this issue. Studying additional interactions or more complex interaction types could also involve different types of physical movements or sequences of multiple movements. Studying such interactions could further the knowledge of interactive graphical encodings and broaden the understanding of embedded interactions for more complex scenarios.

9 Conclusion

We studied the effectiveness of interacting with 12 elementary graphical encodings for basic value-adjustment tasks, and compared our ranking of the interactive graphical encodings from Cleveland and McGill [8]. In general, our ranking follows that of the prior studies, with the exception of our study observing a significant difference between length and angle in terms of accuracy. By studying interaction behavior, our results contribute the finding that users achieve confidence during interaction more quickly when adjusting encodings that exhibit higher overall accuracy. We discuss these results in the greater context of the role of user interaction for visualization. Through our research, we strive to motivate data visualization designers to incorporate such interactive graphical encodings into their interaction design in concert with direct manipulation and dynamic querying techniques.

References

[1] C. Ahlberg. Spotfire: an information exploration environment. *ACM SIGMOD Record*, 25(4):25–29, 1996.

[2] M. Beaudouin-Lafon. Instrumental interaction: an interaction model for designing post-wimp user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 446–453. ACM, 2000.

[3] R. A. Becker, W. S. Cleveland, and A. R. Wilks. Dynamic graphics for data analysis. *Statistical Science*, pages 355–383, 1987.

[4] P. Benner. *Interpretive phenomenology: Embodiment, caring, and ethics in health and illness*. Sage publications, 1994.

[5] A. Bezerianos and P. Isenberg. Perception of visual variables on tiled wall-sized displays for information visualization applications. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2516–2525, 2012.

[6] M. Bostock, V. Ogievetsky, and J. Heer. D^3 data-driven documents. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2301–2309, 2011.

[7] E. T. Brown, J. Liu, C. E. Brodley, and R. Chang. Dis-function: Learning distance functions interactively. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, pages 83–92. IEEE, 2012.

[8] W. S. Cleveland and R. McGill. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American statistical association*, 79(387):531–554, 1984.

[9] W. S. Cleveland and R. McGill. Graphical perception and graphical methods for analyzing scientific data. *Science*, 229(4716):828–833, 1985.

[10] A. Dix. *Human-computer interaction*. Springer, 2009.

[11] N. Elmqvist, A. V. Moere, H.-C. Jetter, D. Cernea, H. Reiterer, and T. Jankun-Kelly. Fluid interaction for information visualization. *Information Visualization*, 10(4):327–340, 2011.

[12] A. Endert, L. Bradel, and C. North. Beyond control panels: Direct manipulation for visual analytics. *Computer Graphics and Applications, IEEE*, 33(4):6–13, 2013.

[13] A. Endert, P. Fiaux, and C. North. Semantic interaction for visual text analytics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 473–482. ACM, 2012.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2017.2680452, IEEE Transactions on Visualization and Computer Graphics

[14] A. Endert, C. Han, D. Maiti, L. House, S. Leman, and C. North. Observation-level interaction with statistical models for visual analytics. In *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*, pages 121–130. IEEE, 2011.
[15] P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology*, 47(6):381, 1954.
[16] J. Heer, M. Agrawala, and W. Willett. Generalized selection via interactive query relaxation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 959–968. ACM, 2008.
[17] J. Heer and M. Bostock. Crowdsourcing graphical perception: using mechanical turk to assess visualization design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 203–212. ACM, 2010.
[18] A. Kerren, J. Stasko, J.-D. Fekete, C. North, D. Keim, G. Andrienko, C. Görg, J. Kohlhammer, and G. Melançon. Visual Analytics: Definition, Process, and Challenges. In *Information Visualization*, volume 4950 of *Lecture Notes in Computer Science*, pages 154–175. Springer Berlin / Heidelberg, 2008.
[19] H. Kim, J. Choo, H. Park, and A. Endert. Interaxis: Steering scatterplot axes. *IEEE Visual Analytics Science and Technology (VAST)*, 2015.
[20] N. W. Kim, E. Schweickart, Z. Liu, M. Dontcheva, W. Li, J. Popovic, and H. Pfister. Data-driven guides: Supporting expressive design for information graphics. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):491–500, 2017.
[21] B. Kondo and C. M. Collins. Dimpvis: Exploring time-varying information visualizations by direct manipulation. *Visualization and Computer Graphics, IEEE Transactions on*, 20(12):2003–2012, 2014.
[22] B. Kondo, H. Mehta, and C. Collins. Glidgets: Interactive glyphs for exploring dynamic graphs, 2014. Best Poster Award.
[23] R. Kosara and D. Skau. Judgment error in pie chart variations. In *Proceedings of the Eurographics/IEEE VGTC Symposium on Visualization*, pages 91–95. Wiley Online Library, 2016.
[24] B. Kwon, W. Javed, N. Elmqvist, and J.-S. Yi. Direct manipulation through surrogate objects. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*, pages 627–636, 2011.
[25] B. C. Kwon, H. Kim, E. Wall, J. Choo, H. Park, and A. Endert. Axisketcher: Interactive nonlinear axis mapping of visualizations through user drawings. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):221–230, 2017.
[26] B. Lee, P. Isenberg, N. H. Riche, and S. Carpendale. Beyond mouse and keyboard: Expanding design considerations for information visualization interactions. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2689–2698, 2012.
[27] S. Lewandowsky and I. Spence. Discriminating strata in scatterplots. *Journal of the American Statistical Association*, 84(407):682–688, 1989.
[28] I. S. MacKenzie. Fitts' law as a research and design tool in human-computer interaction. *Human-computer interaction*, 7(1):91–139, 1992.
[29] I. S. MacKenzie and W. Buxton. Extending fitts' law to two-dimensional tasks. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 219–226. ACM, 1992.
[30] I. S. MacKenzie, T. Kauppinen, and M. Silfverberg. Accuracy measures for evaluating computer pointing devices. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 9–16. ACM, 2001.
[31] J. Mackinlay. Automating the design of graphical presentations of relational information. *Acm Transactions On Graphics (Tog)*, 5(2):110–141, 1986.
[32] J. Matejka, M. Glueck, T. Grossman, and G. Fitzmaurice. The effect of visual appearance on the performance of continuous sliders and visual analogue scales. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 5421–5432. ACM, 2016.
[33] D. A. Norman. *Things that make us smart: Defending human attributes in the age of the machine*. Basic Books, 1993.
[34] U.-D. Reips. Standards for internet-based experimenting. *Experimental psychology*, 49(4):243–256, 2002.
[35] N. H. Riche, B. Lee, and C. Plaisant. Understanding interactive legends: a comparative evaluation with standard widgets. In *Computer graphics forum*, volume 29, pages 1193–1202. Wiley Online Library, 2010.
[36] S. Robert. Information visualization-design for interaction. *UK: Pearson Educated Limited*, 2007.
[37] J. M. Rzeszotarski and A. Kittur. Kinetica: Naturalistic multi-touch data visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 897–906. ACM, 2014.
[38] B. Saket, H. Kim, E. T. Brown, and A. Endert. Visualization by demonstration: An interaction paradigm for visual data exploration. *IEEE Transactions on Visualization and Computer Graphics*, PP(99):1–1, 2016.
[39] A. Satyanarayan and J. Heer. Lyra: An interactive visualization design environment. In *Computer Graphics Forum*, volume 33, pages 351–360. Wiley Online Library, 2014.
[40] B. Shneiderman. 1.1 direct manipulation: a step beyond programming languages. *Sparks of innovation in human-computer interaction*, 17:1993, 1993.
[41] B. Shneiderman. Dynamic queries for visual information seeking. *Software, IEEE*, 11(6):70–77, 1994.
[42] D. Simkin and R. Hastie. An information-processing analysis of graph perception. *Journal of the American Statistical Association*, 82(398):454–465, 1987.
[43] D. Skau and R. Kosara. Arcs, angles, or areas: Individual data encodings in pie and donut charts. In *Computer Graphics Forum*, volume 35, pages 121–130. Wiley Online Library, 2016.
[44] I. Spence and S. Lewandowsky. Displaying proportions and percentages. *Applied Cognitive Psychology*, 5(1):61–77, 1991.
[45] R. Spence and L. Tweedie. The attribute explorer: information synthesis via exploration. *Interacting with Computers*, 11(2):137–146, 1998.
[46] J. Talbot, J. Gerth, and P. Hanrahan. An empirical model of slope ratio comparisons. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2613–2620, 2012.
[47] J. J. Thomas and K. A. Cook. Illuminating the path, 2005.
[48] R. S. Torres, C. G. Silva, C. B. Medeiros, and H. V. Rocha. Visual structures for image browsing. In *Proceedings of the twelfth international conference on Information and knowledge management*, pages 49–55. ACM, 2003.
[49] E. R. Tufte and P. Graves-Morris. *The visual display of quantitative information*, volume 2. Graphics press Cheshire, CT, 1983.
[50] A. Van Dam. Post-wimp user interfaces. *Communications of the ACM*, 40(2):63–67, 1997.
[51] M. Wagner. *The geometries of visual space*. Psychology Press, 2006.
[52] C. Ware. *Information visualization: perception for design*. Elsevier, 2012.
[53] W. Willett, J. Heer, and M. Agrawala. Scented widgets: Improving navigation cues with embedded visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1129–1136, 2007.
[54] C. Williamson and B. Shneiderman. The dynamic homefinder: Evaluating dynamic queries in a real-estate information exploration system. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 338–346. ACM, 1992.
[55] J. S. Yi, Y. ah Kang, J. T. Stasko, and J. 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.

Bahador Saket is currently a Ph.D. student at Georgia Institute of Technology. His research areas of interest include human-computer interaction and information visualization.

Arjun Srinivasan is currently a Ph.D. student at Georgia Institute of Technology. His research areas of interest include Visual Analytics, InfoVis, and Natural Language Processing.

Eric D. Ragan is an Assistant Professor in the Department of Visualization at Texas A&M University. He directs the Interactive Data and Immersive Environments (INDIE) lab, and he is a faculty lead of the VIZ Human-Computer Interaction lab (HCI-VIZ). His research interests include HCI, information visualization, visual analytics, and virtual reality.

Alex Endert is an Assistant Professor in the School of Interactive Computing at Georgia Institute of Technology. He directs the Visual Analytics Lab, where him and his students explore novel user interaction techniques for visual analytics.

Visualization by Demonstration: An Interaction Paradigm for Visual Data Exploration

Bahador Saket, Hannah Kim, Eli T. Brown, Alex Endert

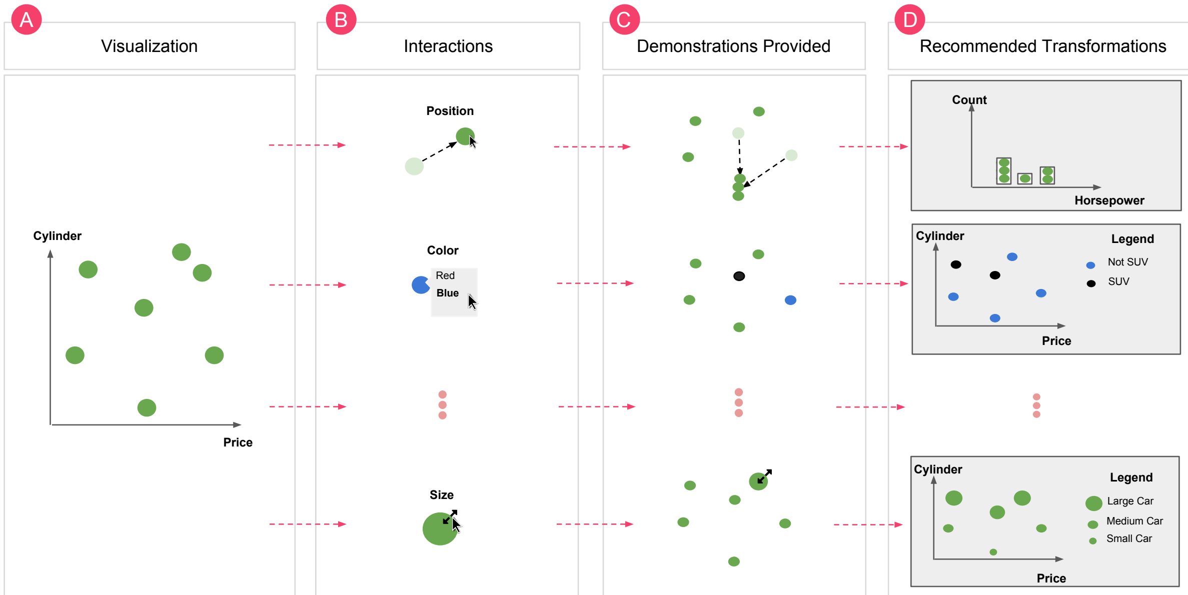


Figure 1: An overview of the Visualization by Demonstration paradigm (illustrated using a car dataset). A) Interactive visualization displayed. B) Users can manipulate spatial and graphical encodings directly *(e.g., coloring data points)*. C) Users provide visual demonstrations of incremental changes to a visualization. D) Using these demonstrations, the system estimates the intended results and recommends possible transformations.

Abstract—Although data visualization tools continue to improve, during the data exploration process many of them require users to manually specify visualization techniques, mappings, and parameters. In response, we present the *Visualization by Demonstration* paradigm, a novel interaction method for visual data exploration. A system which adopts this paradigm allows users to provide visual demonstrations of incremental changes to the visual representation. The system then recommends potential transformations *(Visual Representation, Data Mapping, Axes, and View Specification transformations)* from the given demonstrations. The user and the system continue to collaborate, incrementally producing more demonstrations and refining the transformations, until the most effective possible visualization is created. As a proof of concept, we present VisExemplar, a mixed-initiative prototype that allows users to explore their data by recommending appropriate transformations in response to the given demonstrations.

Index Terms—Visualization by Demonstration, Visualization Tools, Visual Data Exploration

1 Introduction

Visualization researchers and practitioners continue to develop a wide range of interactive visualizations which allow users to explore and make sense of their data. One widely-used interaction paradigm for these visualizations is direct manipulation through control panels and other graphical widgets [28]. This includes controls and selections such as view specification, filtering, assigning data attributes to visual encodings such as color or size, and others. Such methods require users to specify their visualization techniques *(e.g., selecting a scatterplot)*, mappings *(e.g., assigning size to a data attribute)*, and parameters *(e.g., assigning data attributes to axes)* in order to generate and modify visualizations to explore their data. Alternatively, more recent work in visualization recommender systems has given users the ability to select data attributes of interest, from which visualizations are generated [33, 4]. The recommendations are generated based on the computed data characteristics and the user input about which data attributes are of interest.

There also exist prior studies that show the effectiveness of letting people create spatial representations of data points manually, without the need to formalize the mappings between the data and the spatial constructs created. For example, Andrew et al. [2] found that people use the spatial environment to create layouts of information which have meaning to the person, without requiring users to specify the formal definition of the layout. For example, people create clusters of similar data points by moving similar data points closer to each other. Further, during such spatial exploration processes, some of the spatial arrangements exhibit characteristics similar to formal visualization techniques [15, 16]. For example, participants stacked data points in the shape of bars to count the number of specific items (similar to

- *Bahador Saket, Hannah Kim, and Alex Endert are with Georgia Institute of Technology. E-mail: {saket, hannahkim, endert@gatech.edu}.*
- *Eli T Brown is with DePaul University E-mail ebrown80@cdm.depaul.edu.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

the well-known bar chart visualization technique). These studies show that users are effective at providing visual demonstrations of visualization techniques, mappings, and parameters.

The demonstration-based paradigm for human-computer collaboration has seen use in a variety of contexts. In computer programming, programming by demonstration [8] allows users to generate code by providing demonstrations of some intended result, often done visually. The user and the system continue to collaborate, using further user demonstrations to incrementally improve the computer-generated code. Wrangler [18] is also an instance of "by demonstration" systems. Wrangler allows users to provide demonstrations of expected results on tabular data by directly showing results in the table view *(e.g., selecting a substring of a column to generate the transformation for creating a new column)*. Other application domains that exhibit the "by demonstration" approach include query by demonstration [35], data cleaning by demonstration [21], 3D drawing by demonstration [17], and more. In this work, we show that this interaction paradigm can be extended to visual data exploration and information visualization.

We present the *Visualization by Demonstration* paradigm. The paradigm advocates for decreasing the level of formalism and fundamental knowledge required for visual data exploration. Instead of specifying which visualization technique, mappings, and parameters to generate and update a visualization, Visualization by Demonstration allows users to provide visual demonstrations of incremental changes to the visual representation from which transformations are recommended. Using these demonstrations, the system estimates the intentions and generates potential transformations *(e.g., a bar chart, mapping color to a data attribute)*. This iterative process allows users to visually explore their data without requiring direct visualization specification. That is, the goal is to balance the responsibility of data exploration between the user and the system — users provide visual demonstrations, while the system generates visualizations and defines the visualization mappings and parameters (see Figure 1).

The contribution of Visualization by Demonstration over existing work is twofold. First, the concept of Visualization by Demonstration does not require users to specify visualization techniques ahead of time. Instead, the paradigm will extract the visualization technique from the given demonstrations. Second, the paradigm also extracts visualization mappings and parameters that match given demonstrations. For example, if users stack data points vertically, Visualization by Demonstration suggests a bar chart with appropriate axes.

To show the feasibility of the Visualization by Demonstration concept, we implemented VisExemplar. VisExemplar is a mixed-initiative data exploration prototype that allows users to explore their data using Visualization by Demonstration. VisExemplar allows users to provide visual demonstrations of incremental changes to the visualization by directly manipulating the visual representation *(e.g., moving one data point on the top of another, changing the color of a select set of point, etc.)*. VisExemplar uses a recommendation engine that generates transformations in response to the given demonstrations using a set of intent functions. To help users understand the results of different recommended transformations before they commit, VisExemplar contributes novel methods for presenting recommended transformations.

The remainder of the paper is organized as follows. Section 2 first discusses differences between interaction methods used in existing visualization tools and the Visualization by Demonstration paradigm. It then describes some of the studies that inspired the Visualization by Demonstration paradigm. In the light of previous work, Section 3 describes Visualization by Demonstration in more detail. Section 4 provides a usage scenario and the system design of VisExemplar. In Section 5 we discuss the potential value of Visualization by Demonstration and possible research avenues for continued research.

2 Related Work

2.1 Visual Data Exploration

Visual representations are one of the fundamental components of any visualization tool [34]. A central component of visual representations

Figure 2: Constructing a visualization with tokens. Figure from [16] used with permission.

is the mappings from data values to graphical representations [7]. Visual representations are constructed using a combination of different visual encodings *(e.g., length, position, size, color, etc.)* [6]. Thus, interactivity in visualization tools is often designed to change one or more mappings between the data and the visual encodings. A popular method for visualizing data is using pre-existing interactive visualization tools *(e.g., SpotFire [29] and Tableau [30])*. Such visualization systems allow users to specify direct mappings between their data and the visual representation without requiring users to have programming skills. However, such systems require some amount of fundamental knowledge about the data, the domain, and of visualization techniques. For instance, to create a scatterplot, users must specify the technique, and then which of the data attributes to map onto the x and y axes.

There also exist mixed-initiative systems [14] which aim to balance the responsibility of data visualization between the user and the system. One category of such systems is visualization recommendation tools. Many visualization recommendation systems have been developed to assist users to visualize their data *(e.g., [4, 12, 22, 23, 33])*. They suggest alternative views (visual encodings, data attributes, and data transformations) based on user-specified data of interest and computed characteristics about the data. Voyager [33] is mixed-initiative system that couples faceted browsing with visualization recommendation to support visual data exploration based on user-specified data attributes of interest. VizAssist [4] generates visualizations by requiring users to specify both desired data attributes and tasks.

Instead of requiring users to specify the visualization technique, mappings, and parameters a priori (or specify data attributes of interest), Visualization by Demonstration extracts this information from created visual demonstrations. Visualization by Demonstration can be used independently or to augment existing visualization tools to increase their functionality in concert with direct manipulation controls.

2.2 Flexibility of Spatial Data Organization

Andrews et al. [2] showed how space was used by their participants as both an external memory aid in which the spatial constructs carried meaning. The participants formed spatial constructs *(e.g., groups, lists, clusters)* as a means to organize the information, as well as structure their analytic process. Various studies also used spatial environments as a thinking medium to allow users to construct their visualizations incrementally [27, 31, 32]. For example, Walny et al. [32] showed how pen and touch can be used to construct visualizations on interactive whiteboards. Similarly, SketchStory [20] demonstrates how users can draw visualization characteristics (e.g., axes of a scatterplot) to generate visualizations. Moreover, Schroeder et al. [26] introduced a sketching technique that enable graphic designers and artists to construct multivariate time-varying visualizations by painting on a digital data canvas, sketching data glyphs, and blending together multiple layers of animated 2D graphics. Satyanarayan and Heer [25] presented Lyra, a direct manipulation environment that allows users to create customized visualizations by without requiring coding (e.g., dragging and dropping data attributes directly onto visual glyphs). Similarly Ren et al. [24] presented iVisDesigner, an interactive environment that allows users to design visualizations interactively, without the need for coding.

Huron et al. [15] proposed a method called "Constructive Visualization", which advocates for allowing people to create visualizations by manually moving, adding, and removing physical tokens. In their study, each token represents a basic data unit. Thus, constructing a visualization means assembling these tokens to encode the data in a meaningful way (see Figure 2). They found people use the spatial environment to construct visualizations and explore their data, where many spatial arrangements exhibit characteristics similar to formal visualization techniques. For example, people stacked data points in the shape of bars to count the number of specific items (similar to the well-known bar chart visualization technique). Inspired by these studies, the direct manipulation of spatial encodings of a visual representation is one of the methods that people can use to provide demonstrations in the Visualization by Demonstration paradigm.

Prior work also exists which allows users to directly adjust the position of data points (e.g., documents), interpret this feedback via a dimensionality reduction model to generate a new spatialization that better reflect the users understanding of the high-dimensional data [5, 9, 10]. DimpVis is one of the recent systems which applies embedded interaction as a substitution to other options *(e.g., time slider)* for querying and exploring time-varying information visualizations. The system allows users to directly manipulate the data points in visualizations to perform temporal navigation of the dataset [19].

Visualization by Demonstration advocates for a similar form of direct visual and graphical manipulation of data points. People are given the ability to reposition and re-encode visual demonstrations of data, from which the mappings are computed.

3 Visualization by Demonstration

The systems which make use of the Visualization by Demonstration paradigm allow users to provide visual demonstrations of incremental changes to the visualization as a method for user interaction. These demonstrations could be provided by direct manipulation of the spatial and graphical encodings used in a visualization. The systems then recommend potential transformations from the given demonstration. In this section we first discuss two different methods that the Visualization by Demonstration paradigm supports for providing demonstrations. We then discuss different classes of transformations supported by this paradigm. Finally, we present design guidelines that should be considered by systems adopting Visualization by Demonstration.

3.1 Methods for Providing Demonstrations

Building on the strong research foundation of previous work [2, 15, 16, 32], one of the methods by which the systems adopting Visualization by Demonstration might allow users to provide visual demonstrations is by directly adjusting the spatial layouts of data points *(e.g., users stacking data points in the shape of bars to convey their interest in a bar chart or placing two data points in desired positions along the x or y axis to demonstrate the attribute to map to the axis).* In addition to direct manipulation of spatial encodings, the systems adopting this paradigm might allow users to demonstrate desired graphical encodings applied to data attributes by adjusting the graphical encodings used in a visualization *(e.g., changing the color or size of data points in a scatterplot, length or color of bar in a bar chart, and others).*

3.2 Recommending Potential Transformations

Visualization by Demonstration suggests possible transformations that can be applied based on the provided demonstrations. For each demonstration, the system checks how the current state of the visualization, parameters, and mappings should be transformed to create meaningful visual representations that match the demonstration. The system then recommends these possible transformations. We categorize these transformations into four main categories, described below.

Visualization Representation Transformations change the current visualization technique to a different visualization technique *(e.g., transforming from a scatterplot to a bar chart).* To convey interest in transforming to a new visualization technique, users can manipulate the spatial encoding to create a spatial layout similar to the intended visualization technique. For example, users can stack two or more data points vertically or horizontally in a scatterplot to demonstrate their interest of switching to a vertical or horizontal bar chart.

Data Mapping Transformations define mappings between graphical encodings and data attributes *(e.g., mapping color to an attribute).* To convey interest in assigning a graphical encoding to a data attribute, users can manipulate the corresponding graphical encoding in the visual representations. For example, users could color one or more data points red to convey their interest in mapping color to a data attribute.

Axes Transformations assign data attributes to axes of a visualization technique *(e.g., assigning an attribute to the x axis of a scatterplot).* To assign new data attributes to axes, users can manipulate the corresponding graphical encoding or spatial encoding in the visual representations. For example, in a scatterplot, users could move one or more data points to a positions along an axis to demonstrate their interest in having a scatterplot in which the manipulated data point is close to the current coordinates, thus changing the attribute assigned to the axis. Alternatively, in a bar chart, users could change the length of one of more bars to demonstrate mapping a new attribute to the axis.

View Specification Transformations change the view specifications without changing the underlying technique *(e.g., aggregation, average, sorting).* For example, users could convey their interest in sorting a bar chart by dragging the longest bar in the current bar chart to the most left or right side of the axis.

3.3 Design Guidelines

We identify the following design guidelines for applications that make use of Visualization by Demonstration. These were refined through our experiences through several design iterations.

G I: Support direct manipulation of visual representations to foster visual data exploration. Inspired by previous studies [15, 32, 20], Visualization by Demonstration should provide an environment in which users provide demonstrations by manipulating the spatial and graphical encodings of data directly in the visual representations.

G II: Balance human and machine workload in the visual data exploration process. Allowing users to construct demonstrations by manipulating the spatial and graphical encodings increases the directness of their interactions. However, manually manipulating all the data points in a visualization is time consuming. To balance human and computer's effort in this process, this paradigm advocates for a mixed-initiative approach to human-computer collaboration during the visual data exploration process – users provide visual demonstrations, and the system provides recommended transformations.

G III: Enable user interactions to drive recommended transformations. Systems adopting this paradigm should allow users to specify their intentions or desired aspects of the data by directly manipulating the data points in a visual representation. In aggregate, these interactions create visual demonstrations which serve as the primary units by which users communicate their intended changes to the system.

G IV: Enhance interpretability of recommendations. Recommended transformations should be presented in way that allow users to extract why specific transformations were shown and what would be the resulting outcome if they accept any of the transformations. This requires recommended transformations to be presented in a right position in the interface, and provide users with the rationale behind why they were recommended.

4 VisExemplar

To demonstrate the application of Visualization by Demonstration, we developed VisExemplar (see Figures 3 and 4). All components of the VisExemplar are implemented using JavaScript and the visualization modules are built using the D3 toolkit [3]. Datasets in comma-separated values format are supported. The implemented system is available at `http://bitbucket.org/bahadorsaket9/vbd`.

4.1 Usage Scenario

In this section, we motivate the design of our system and illustrate the functionality via a usage scenario. We indicate how someone can utilize VisExemplar to examine data about cars. The car dataset [13] provides specifications on new cars and trucks for the year 2004. The dataset contains 122 data points, with 18 data attributes describing each car. This dataset is used throughout the motivating examples in this paper.

Figure 3: The VisExemplar user interface consists of a ThinkBoard, Recommendation Gallery, and a Detail View panel. ThinkBoard shows each data point as a circle. The Recommendation Gallery shows visualization technique transformations. The Detail View shows data details, and also recommended data mapping transformations.

Assume Amy wants to buy a car, and wants to make a data-driven decision based on this dataset. Amy needs to find a single car to purchase that best meets her needs and preferences, and decides to do so using VisExemplar. She has limited knowledge in constructing visualizations (as well as little domain expertise about cars), but would like to use visualization to help her make her purchase.

Upon loading the data, VisExemplar shows each car as a green circle on the ThinkBoard (ThinkBoard is shown in Figure 3). Amy can interact with the cars *(e.g., move, resize, or recolor)* on the ThinkBoard, or search for a specific car or manufacturer that she is familiar with by using the search box. Two of Amy's friends drive a Toyota Prius and a Honda Civic Hybrid. She likes both of them, but is not as familiar with the attributes that define the cars, so she simply starts by searching for both. See Figure 4-(a) for more details.

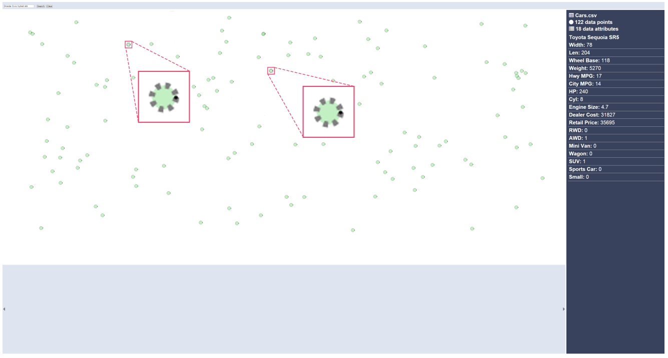
She decides to put the cars that she is interested in close to each other somewhere on the ThinkBoard. She drags the two cars close to each other (see Figure 4-(b)). Upon this interaction, the system recognizes that there are possible visual representations in which these two cars are close to each other. In this case, there exist $x-y$ axes pairs that would result in a scatterplot consistent with the two example data points placed in close proximity of each other. The system recommends placing different options for the x and y axis *(e.g., length of the cars (Len) for x axis and weight of the cars for y axis)*. See Figure 4-(b) for more details. Amy looks at different attribute options to assign to the axes. Due to her interest in a med-sized sedan, she decides to select *length of the cars (Len)* as the x axis to see how the lengths of the cars compare across the dataset.

After assigning *Len* to the x axis, VisExemplar produces the scatterplot shown in Figure 4-(c). Amy notices that both cars she initially dragged close together have a length of 175. She decides to hone her search of a car to other vehicles that have roughly this size by coloring several cars with a length of about 175 red by right clicking on them and picking the red color (as shown in Figure 4-(d)). The system automatically extracts data attributes that can be mapped to color *(e.g., cylinder (Cyl), as well as others)*. Data attributes which can be assigned to color are indicated by a brush icon ( ) next to the data attributes in the detail panel (see Figure 4-(d)). In this case, Amy notices that the system recommended assigning color to the number of cylinders (Cyl). Intruiged, she decides to accept this mapping by double-clicking on the brush icon. Figure 4-(e) shows the resulting view, where the color mapping is shown in the legend on the Detail View panel.

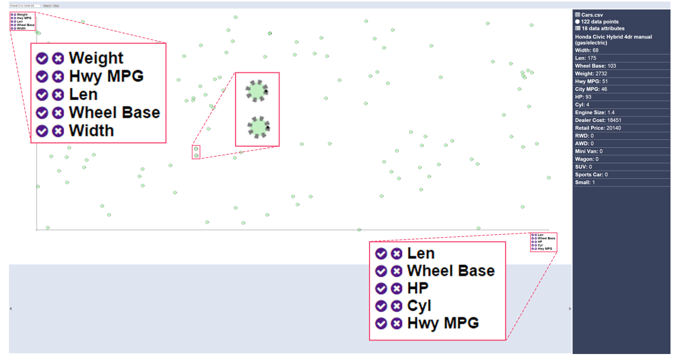
Amy notices that many of the cars with a length of 175 are 4 cylinder cars (shown in red), and asks herself "how many of the cars have a length of 175, compared to the lengths of other cars?". She stacks a few cars with the length of 175 vertically on the top of each other to group these items together and count them (Figure 4-(f)). Based on her example, VisExemplar recommends a selection of bar charts. The recommendations are based on the attributes that the stacked cars have in common. Each proposed bar chart has one data attribute in the x axis and the corresponding count of cars in the y axis. The bars are drawn as a box containing the counted cars. Amy explores different recommended bar charts by scrolling the Recommendation Gallery, and chooses one showing length as an axis labelled "Stack vertically based on **Len**." (Figure 4-(f)).

At this point, Amy has a visualization where the x axis of the bar chart assigned to Len and the y axis as the number of cars (Figure 4-(g)). By looking at the y axis of the bar chart Amy notices that among all 122 cars only 34 of the cars have a length between 175 and 180. Among these, 13 are 4 cylinder cars (colored red). Amy hovers over these to get more details, and finds two additional cars (Toyota Corolla CE and Honda Civic EX) which have the characteristics that she has found interesting up to this point.

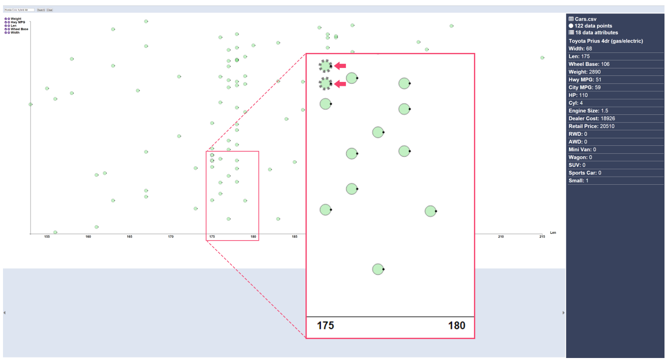
She wants to switch back to a scatterplot to see additional attributes on the axis to make a more detailed comparison. She drags these four cars out of the bars, demonstrating the intent to switch to a scatterplot. The system will again compute the $x-y$ axes pairs that would result in a scatterplot given the locations of the dragged cars. Amy starts exploring different recommended scatterplots by scrolling the Recommendation Gallery. Suddenly the label **Retail Price** in one of the thumbnails grabs her attention (Figure 4-(h)). She picks that recommendation as she realizes she has ignored price up to this point. The visualization shows that all four cars are roughly the same in price, and decides to schedule a test drive for each of them as she feels confident in her choices, and has learned a bit more about attributes of cars that define her subjective preference for cars.



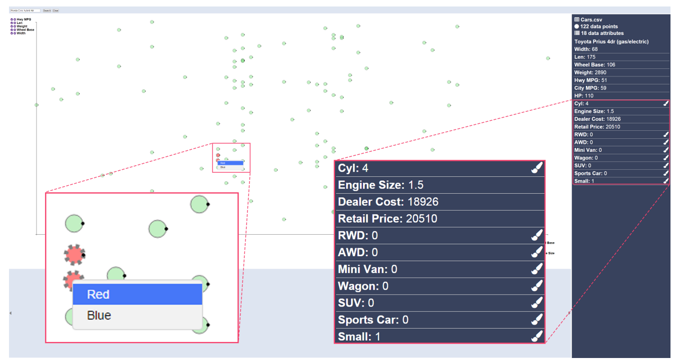
(a) Searched for two cars on the search box. Border line of the selected cars changed to dashed line.



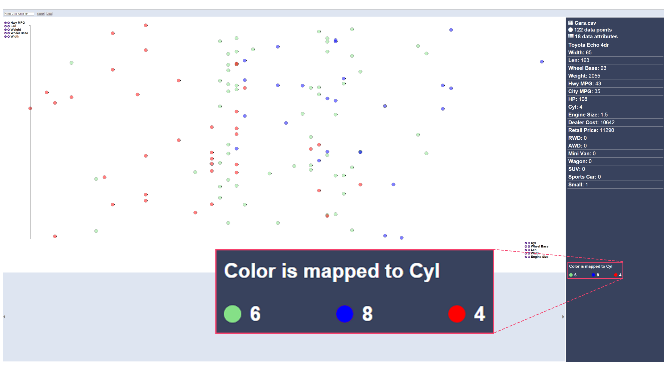
(b) Dragged two cars close to each other. The system recommended options for assigning to the x and y axis.



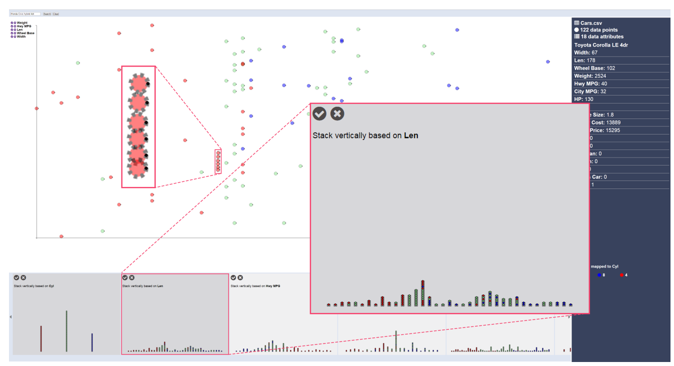
(c) Assigned Len to the x axis.



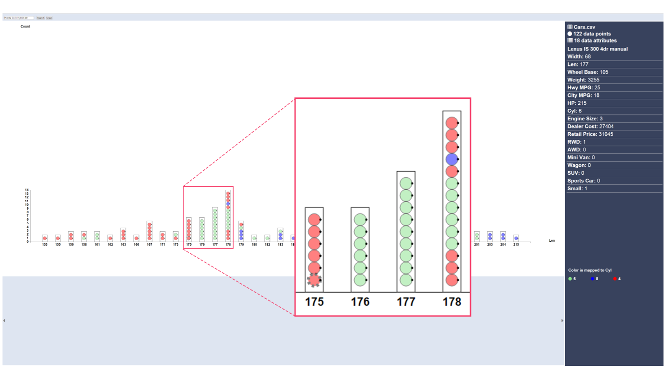
(d) Colored two cars. The system recommended data attributes that can be mapped to color.



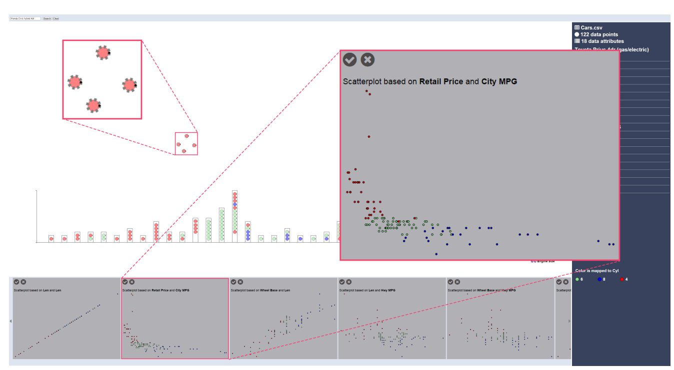
(e) Color mapping is shown. Color is mapped to Cylinder (Cyl).



(f) Stacked a few of the cars with the length of 175 vertically on the top of each other. The system recommends different bar charts.



(g) Selected a bar chart where x axis assigned to Len and the y axis as the number of cars.



(h) Selected a scatterplot where $x-y$ axes indicate Retail Price and City MPG.

Figure 4: A series of screenshots showing the usage of Visualization by Demonstration in VisExemplar.

4.2 The VisExemplar Interface

Figure 3 shows VisExemplar's interface, consist of a ThinkBoard, a Recommendation Gallery, and a Detail View panel. ThinkBoard is a thinking medium for users and allows them to construct their demonstrations through direct manipulation of the visual representation. Moreover, possible *Axes and View Specification transformations* be shown on the ThinkBoard. *Visual Representation transformations* will be presented in the Recommendation Gallery. The primary goal of the Detail View panel is to show details of selected data points. By hovering on a data point on the interaction board, the Detail View panel will show detail information related to that data point. See Figure 3 for more details. In addition, *Data Mapping transformations* will be shown as small icons on this panel.

VisExemplar realizes the four Visualization by Demonstration design guidelines presented in Section 3 as follows. VisExemplar provides an environment similar to a spatial workspace in which users provide demonstrations by manipulating the spatial and graphical encodings used in visual representations (**G I**). In addition, to balance human and computer effort during data exploration process, VisExemplar suggests variety of possible relevant transformations in the form of *Visual Representations transformations*, *Data Mapping transformations*, *Axes transformations*, and *View Specification transformations*. Depending on the transformation type, they will be shown in different forms and locations on the interface (**G II**). VisExemplar allows users to provide demonstrations by directly manipulating the data points in a visual representation (**G III**). Finally VisExemplar uses different methods for showing recommendations to the user. First, visual representation transformations are shown as thumbnails below the ThinkBoard. These are ordered and colored based on their computed relevance to the visual demonstration. Data mapping transformations are shown as icons in the detail panel, and axes transformations on the axes. This helps users browse the possible space of transformations and interpret their result (**G IV**).

4.3 Transformations Supported in VisExemplar

VisExemplar currently supports four categories of transformations.

Visual Representation Transformation. VisExemplar currently supports transformations from bar charts to a scatterplots and vice versa. Recommended *Visual Representation Transformations* will be shown on the Recommendation Gallery. Each recommended transformation is shown as a thumbnail in the gallery. Users can explore different transformations by scrolling through gallery. Each thumbnail consists of a textual explanation describing what the visualization in the thumbnail is showing *(e.g., Stack vertically based on Cylinder)* and a visualization which gives an overview of the transformation. We decided to show this type of recommendation as thumbnails because during the design process we found it difficult to imagine the resulting changes from one visual representation to another without seeing the resulting view. Relevance of this type of transformations is dually encoded by color and position. By default, we use a light gray color as background for recommended transformations in the gallery. We show the relevance of the recommended transformations by adjusting the darkness of the background color, the lighter the background color the lower the relevance. In addition, the recommended transformations are ordered left to right based on relevant (left being highest). Figures 4-(f) and 4-(h) indicate examples of *Visual Representation transformations* in VisExemplar.

Data Mapping Transformation: The current version of VisExemplar supports color and size encodings. These types of transformations are shown as small icons on Detail View panel corresponding to the attribute which is being recommended to map to the visual demonstration. We decided to show this type of transformation on the Detail View panel since each icon is located beside corresponding data attributes. For those recommended data attributes that can be assigned to color, a small brushing icon (🖌) will appear near the data attributes on the detail panel. Similarly, the system recommends data attributes to be mapped to size by showing a small expand icon (⤢) beside the appropriate data attributes on the Detail View panel. The background color of the data attributes on hover shows the relevance. The lighter

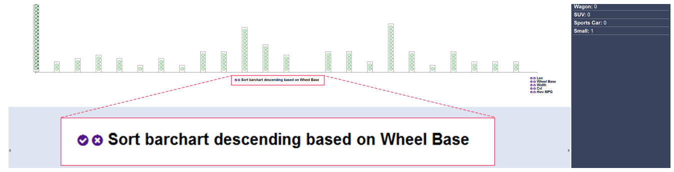


Figure 5: An example of a view specification transformation.

the background color the lower the relevance. Hovering on the recommended data attributes will also show a preview of resulting changes. A user can apply any of the recommended transformations by double clicking on the suggested data attribute. Figure 4-(d) shows examples of *Data Mapping transformations* in VisExemplar.

Axes Transformation. We show *Axes transformations* directly on the corresponding axes in the ThinkBoard. In the early stages of our design process, we noticed that it is easier to understand the meaning of these type of transformations when they are located close to the corresponding axis. For this type of transformation, the position of the data attributes beside each axis show their relevance. The higher the data attribute the higher the relevance. Figure 4-(b) shows examples of *Axes transformations* in VisExemplar.

View Specification Transformation. This type of transformation is shown on the ThinkBoard below the visualization technique. One of these transformations that VisExemplar currently supports is sorting the bar chart in ascending or descending order (See Figure 5).

4.4 Recommendation Engine

When a user performs an interaction with the visual representation and generates a visual demonstration, the recommendation engine of our system accepts the interaction as input, and produces the recommendations as outpout for transformations mentioned in Section 4.3. VisExemplar allows direct manipulation of three encodings of data points including position, color, and size (see Figure 6-(A)). Direct manipulation of each encoding will invoke a series of intent functions related to that specific encoding (see Figure 6-(C)). Based on the demonstrations provided, the intent functions determine which transformations are most relevant. VisExemplar contains seven intent functions. All related intent functions are checked against every interaction. For example, by directly re-positioning data points in a scatterplot to new x coordinates, one of the intent functions which will be invoked is the *assigning X axis* function. Considering the points that have been moved, the system then recommends potential data attributes for the *x* axis that would result in a scatterplot where the moved data points would be as close as possible to the new x coordinates (see Figure 6).

As a result of each interaction, the recommendation engine will update the recommendation table (see Figure 6-(D)). The recommendation table consists of a set of potential transformations. Each row of the table represents a potential transformation. Each transformation consists of a name *(e.g., xAxis_Cylinder)*, relevance, and location on the interface. The table will be created only once and will be updated after each interaction. The relevance value for each transformation indicates the number of times the transformation is generated. The relevance value is normalized to a range of $[0,1]$ and the table is updated to contain the normalized relevance value for each transformation. The system only shows transformations with relevance above 0.3. Upon accepting a transformation the changes are applied and relevance values of all recommendations reset. The transformation type column dictates where each transformation is shown in the interface.

The recommendation engine then passes the recommendation table to the interface. The interface will update the visualization based on the given recommendation table; See Figure 6-(E).

4.4.1 Intent Functions

Depending on the interaction, any of seven currently-supported intent functions might be invoked (see Figure 6-(C)). For example, changing the position of a data point could invoke functions 1, 2, 3 or functions 4 or 5, depending on current state of the visualization (scatterplot or bar chart). If the current visualization is a scatterplot, then resizing a data

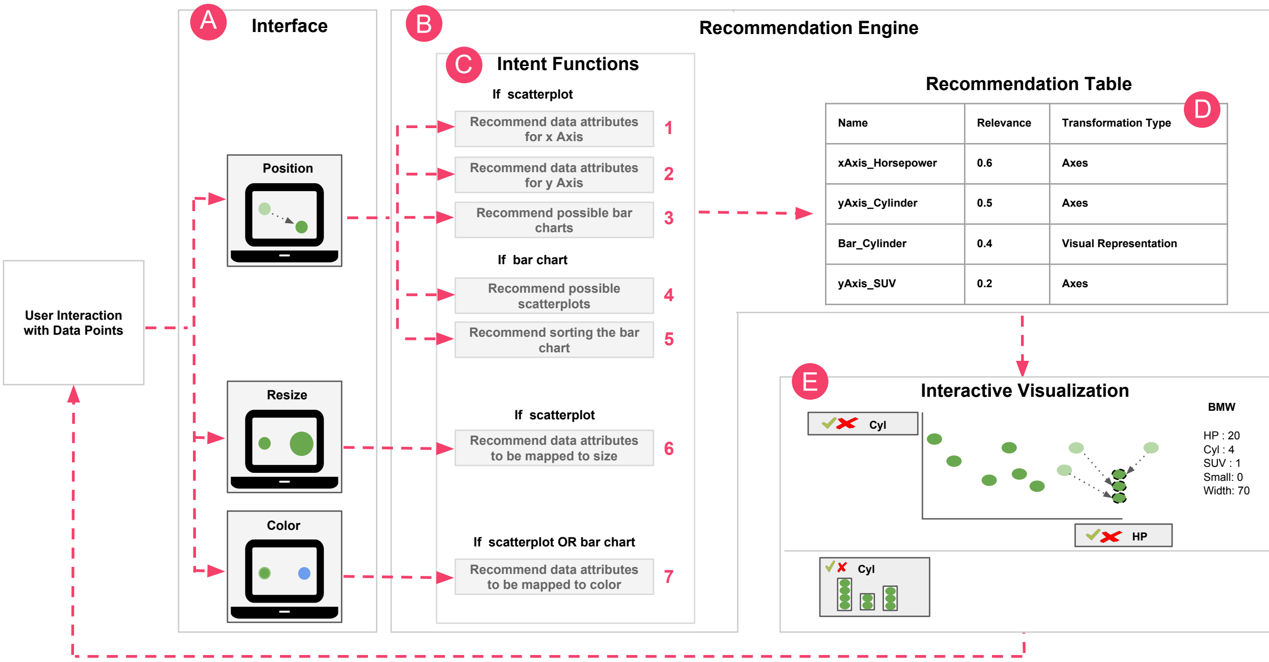


Figure 6: VisExemplar's Low-level Architecture. A) Recommendation engine takes user interactions as input. B) A series of intent functions drive the recommendation table. C) Direct manipulation of each encoding will invoke a series of intent functions related to that specific encoding. D) Recommendation Table will be updated after each interaction and stores a ranked list of potential transformations. E) The updated recommendation table feeds the recommendations in the user interface.

point invokes intent function 6. Recoloring a data point will invoke intent function 7 regardless of the current state of visualization. Below we explain how each of these functions work. Full support of Visualization by Demonstration will require additional intent functions and our system design supports this extensibility.

The notations used in this section are summarized in Table 1. We refer to the data generally in normalized form, i.e. scaled into the interval $[0,1]$ by attribute:

$$\tilde{d_{ij}} = \frac{d_{ij} - \min(\mathbf{d}_{\cdot j})}{\max(\mathbf{d}_{\cdot j}) - \min(\mathbf{d}_{\cdot j})}, \tag{1}$$

where $\min(\mathbf{v})$ and $\max(\mathbf{v})$ indicate the smallest and largest element respectively of vector $\mathbf{v}$.

Position: Depending on the type of the current visualization and user interactions, upon changing the positions of data points, intent functions 1, 2, or 3 (for a scatterplot), or functions 4 or 5 (for a bar chart) are triggered. In this section, we describe how each of these five functions will be triggered after moving the data points.

If the current visualization is a scatterplot, upon the movement of the point, the system either recommends changing the Axes (changing the attributes assigned to x or y-axis) or changing the visual representation to a bar chart.

Intent Function 1 (Figure 6-(C)-1): After a position-changing interaction, the system searches for data attributes to assign to the axes in a scatterplot based on the positions of the moved data points. For example, in Figure 7, the user starts with a scatterplot whose x-axis represents miles-per-gallon (MPG). When the user moves a data point (black arrow in Figure 7 (a)), the x coordinates of the scatterplot no longer map to MPG. Rather, their position is better aligned with the length attribute (Figure 7 (b)). In this case, the system recommends assigning length to the x-axis.

In detail, we linearly normalize coordinate vectors $\mathbf{x}$ and $\mathbf{y}$ into $\tilde{x}$ and $\tilde{y}$ in the same way as Eq. 1, so that they are in the range of $[0,1]$. As a result, coordinate values and data attributes values are in the same scale. Then, we find an data attribute that minimizes the sum of squared differences between normalized coordinate values and data

Table 1: Notation used in this paper.

NOTATION	DESCRIPTION
n	The number of data points
m	The number of data attributes
$\mathbf{d}_i$	The i-th data point
d_{ij}	The j-th attribute value of the i-th data point
$\mathbf{d}_{\cdot j}$	The j-th attribute vector
$\mathbf{x}$, $\mathbf{y}$	The vector of plotted coordinates in the x (or y) axis
x_i, y_i	The plotted value of the i-th data point in $\mathbf{x}$ or $\mathbf{y}$
c_i	The plotted color of the i-th data point
a_i	The plotted area of the i-th data point
$\tilde{v}$	The normalized value of vector v into the interval $[0,1]$
$\lvert S \rvert$	The number of elements in a set S

attribute values of data points. In other words, the system recommends attribute j^* to be assigned to x-axis such that

$$j^* = argmin_j \left\| \tilde{\mathbf{d}}_{\cdot j} - \tilde{\mathbf{x}} \right\|_2^2.$$

Intent Function 2 (Figure 6-(C)-2): In the same way, the system also recommends data attribute(s) for the y-axis of a scatterplot.

Intent Function 3 (Figure 6-(C)-3): If a position-changing interaction results in more than three data points lined up in a row, the system then checks if the three data points are within a specified distance from each other. If so, the system detects user interest for transforming the visualization into a bar chart. In other words, the system interprets the visual demonstration of "stacked data points" as a user's interest in transition to a bar chart. The function then computes common data attributes shared by the aligned data points. Finally, it updates the recommendation table. For instance, if a user places four data points that represent SUVs with six cylinders on top of each other, the system recommends a bar chart by car type *(e.g., SUVs, sports cars, etc)* and a bar chart by the number of cylinders.

If the current visualization is a bar chart, users can move the data points inside each bar or move a bar itself within the bar chart. Note that each bar is shown as a visible box which contain a set of corresponding data points.

Intent Function 4 (Figure 6-(C)-4): If the user drags a data point out of a bar in a bar chart, the system interprets it as a demonstration of changing the visual representation to a scatterplot. After the user drags out two or more data points, the system searches for data attributes that will be assigned to axes of a new scatterplot (using a similar method to the Intent Functions 1 and 2). Based on the current x and y coordinates of the moved points, the system recommends potential x and y axes so that the new scatterplot representations of the moved points would be similar to the current user-defined positions. The recommendations with previews will be shown in the Recommendation Gallery.

Intent Function 5 (Figure 6-(C)-5): Users can drag and drop any of the bars shown in a bar chart. If the user drags the longest bar in the bar chart to the left most side of the bar chart the system recommends sorting the bar chart descending. If the user drags the longest bar in the bar chart to the right most side of the bar chart the system recommends sorting the bar chart in the ascending order.

Resizing: Users can resize data points any time during the data exploration process. Users can adjust the size by dragging a small handle (tiny black circle) on the perimeter of the data point.

Intent Function 6 (Figure 6-(C)-6): When a user resizes a data point in a scatterplot, the system interprets it as the user's interest in encoding a data attribute to the demonstrated sizes of data points. In order to provide enough information to the system for recommending a mapping from a data attribute to data point sizes, the user has to resize two or more data points. The system shows recommended transformations that are above a developer-defined threshold by showing a expand icon (⤢) beside those attributes on the Detail View. Specifically, we first normalize the sizes of data points the user has adjusted in a way that reflects the fact that the minimum is nonzero and that we are seeking changes in the same direction (bigger or smaller) by setting the default drawing value to 0.5. We calculate a scaled value $\tilde{a}_i\ \forall i \in S$ (the set of modified points) as follows:

$$\tilde{a}_i = \begin{cases} \frac{1}{2}\left(1 + \frac{a_i - a_0}{\max(a) - a_0}\right) & \text{if } a_i > a_0 \\ \frac{1}{2}\left(\frac{a_i - \min(a)}{a_0 - \min(a)}\right) & \text{if } a_i < a_0 \end{cases},$$

where a_i is the current plotted size of the i-th data point, a_0 is the default plotting size, and the max and min functions return the predetermined maximum and minimum drawing sizes for points in the visualization. The system recommends attribute j^* for size encoding such that

$$j^* = argmin_j \sum_{i \in S} (\tilde{d}_{ij} - \tilde{a}_i)^2,$$

where S is the set of resized data points.

Recoloring: Users can recolor a data point by right clicking on it and picking a color from the pop-up menu. VisExemplar currently supports three colors: red, blue, and green (default).

Intent Function 7 (Figure 6-(C)-7): We now describe the intent function triggered by changing the colors of data points. For coloring interactions, we consider categorical attributes only and ignore numerical attributes. We define an attribute as categorical if the number of unique values present is fewer than ten and also fewer than half of the number of data points. That is, attribute j is categorical if and only if

$$|\text{unique}(\mathbf{d}_{\cdot j})| \leq \min(10, \frac{n}{2})$$

If the user changes the color of one data point, the system makes a recommendation for each categorical attribute, suggesting applying the same recoloring to all other points sharing the value. For example, if the user changes the color of an AWD sedan with 6 cylinders to red, the system recommends three options: coloring all AWD vehicles red, coloring all sedans red, or coloring all 6-cylinder cars red.

When a user colors two or more data points, two conditions are checked to find the appropriate mapping. The first checks for positive correspondence between a data attribute and the assigned color. The condition is satisfied for an attribute if all points given the same color also have the same value for that attribute. The second condition tests

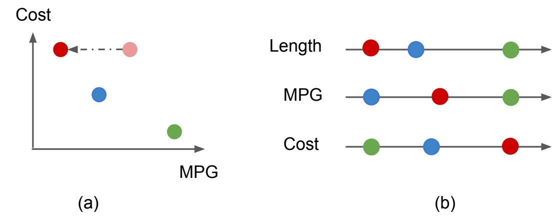


Figure 7: Position-changing interaction. (a) A scatterplot with MPG as x-axis and cost as y-axis. A user moves a red data point. As a result, the system recommends assigning length attribute to x-axis. (b) A visual representation of data distributions for potential data attributes.

that whenever two points have different colors, they have different attribute values. Given a set of indices of k modified points $i_1, \ldots, i_k$, the two conditions on attribute j are, for all pairs $(k, p) : k, p \in i_1, \ldots, i_k$ and $k \neq p$:

$$\text{Condition I: If } c_k = c_p \text{, then } d_{kj} = d_{pj}$$

$$\text{Condition II: If } c_k \neq c_p \text{, then } d_{kj} \neq d_{pj}$$

If all the user-colored data points have the same color, the system checks every categorical data attribute to see if its attribute values of the colored data points are the same using Condition I. The system then recommends all data attributes that meet Condition I. For example, suppose the user colors an AWD sedan with 6 cylinders and a non-AWD sedan with 6 cylinders red. Since both cars are 6-cylinder sedans, two attributes, car-body-type (which includes sedan) and number-of-cylinders, satisfy Condition I. The system recommends two options: coloring all sedans red or coloring all 6-cylinder cars red.

If the data points are colored with two or more colors, the system uses both conditions to evaluate attribute mappings. The two or more colors specify not only the mapping of color to an attribute, but the assignment of values of that attribute to one specific color. First, Condition I is applied across each subset of the re-colored points that have been assigned the same color. This discovers the data attributes shared by each colored group of the modified points. Second, Condition II is applied with the attributes revealed by Condition I to find which attributes can account for the differences across color groups. For example, suppose the user re-colors three data points: two representing cars with attribute values given by the tuples (AWD, sedan, 6 cylinders) and (FWD, sedan, 6 cylinders) are colored red; the third, (AWD, sedan, 4 cylinders), is colored blue. In the red group, both body-type *(e.g., sedan)* and number-of-cylinders satisfy Condition I. However, when Condition II is checked, we see that only the cylinder attribute satisfies both conditions. The system recommends mapping the cylinders attribute to color by coloring 6-cylinder cars red and 4-cylinder cars blue. A brush icon (🖌) beside each of the candidate attributes on the Detail View shows the recommendation to the user.

5 Discussion and Future Work

5.1 Interoperability with Direct Manipulation

There is an inherent tradeoff between the flexibility of Visualization by Demonstration and the loss of formality and expressiveness. A given demonstration could imply multiple transformations and multiple demonstrations might imply the same transformation. Of course this many to many relationship between given demonstrations and transformations raises technical challenges in translating given demonstrations into meaningful transformations. Early on during the process of providing demonstrations, mapping the given demonstrations to possible transformations might be more ambiguous but as users continue completing their demonstrations and providing more evidence and training data to the system, the number of possible transformations would decrease (and ideally become more accurate with respect to the user's goals). This is similar to interactive machine learning [11], in which more examples lead to more accurate decisions. For instance, by coloring a single data point blue, there may be many recommended transformations because the system has less accuracy about the user intentions. However, coloring a few more data points would increase the accuracy of the recommended transformations.

Visualization by Demonstration can be used independently or augment the interaction design of existing visualization tools. For example, the interaction design of SpotFire [1] can be enhanced by Visualization by Demonstration. In this case, expert users could create a bar chart by specifying the data attributes for $x-y$ axes and a visualization technique from the control panel. In addition, non-expert users could benefit from the Visualization by Demonstration approach by providing visual demonstrations to the visualization incrementally and letting the system compute possible transformations.

An important avenue for continued research is conducting a study to compare Visualization by Demonstration with interaction paradigms applied in other existing visualization tools. This requires a separate in-depth study utilizing both qualitative and quantitative techniques to measure the impact of Visualization by Demonstration compared to other interaction methods, using various usability *(e.g., time and error)* and user experience *(e.g., engagement)* metrics. We anticipate that using the Visualization by Demonstration paradigm will increase user engagement, but this remains to be formally studied.

5.2 Generalizing Visualization by Demonstration

We developed VisExemplar to show the feasibility of Visualization by Demonstration. The current version of VisExamplar supports two types of visualization techniques (bar chart and scatterplot) and direct manipulation of three graphical encodings (position, size, and color). It supports interactions which are recognized to be meaningful by previous work. We view the current version of VisExemplar as the first step towards exploring the Visualization by Demonstration paradigm. Multiple avenues for future work lie in improving the VisExamplar interface, as well as enriching the Visualization by Demonstration idea space. We envision expanding VisExamplar by including other visualization techniques *(e.g., linecharts)* and graphical encodings *(e.g., angle and volume)*, and working towards a generalizable interaction framework for visualization.

Generalizing Visualization by Demonstration requires support for providing demonstrations to imply more sophisticated *analytic operations* and *visualization techniques*. For example, how can users indicate their interest in data grouping or aggregation? This requires demonstrations that trigger analytic operations on the data, and show the results visually. For example, users could draw regions around specific data points to demonstrate their interest in executing a clustering algorithm. Additionally, computing standard error to support error bars in bar charts can be demonstrated by drawing the error bars directly. Using demonstration techniques as a medium for performing analytic operations can make complex computation more accessible to a broader set of users.

Further, how can users indicate their interest in transforming to visualization techniques that use a different graphical encoding to encode data points *(e.g., while data points in a scatterplot are encoded using $x-y$ position in a Cartesian space, parallel coordinates encode these data points using lines across multiple parallel axis representing data attributes)*. Implying this type of transition requires new strategies for providing demonstrations. For example, users could connect two data points in a scatterplot using a line to demonstrate their interest in switching to a line chart. Additionally, users could draw vertical parallel lines on a scatterplot to show their interest in switching to a parallel coordinates. We believe that Visualization by Demonstration can generalize to many visualization techniques, although additional forms of interaction may be needed (e.g., sketching and visual authoring).

Generalizing Visualization by Demonstration should be properly evaluated. There exist multiple methods to demonstrate specific transformations. Some are likely more easy to use than others. One research direction might be to observe different strategies people use to demonstrate their interest in more sophisticated operations, extract the common strategies used, and adapt them to expand the Visualization by Demonstration paradigm.

5.3 Consistency of Visual Mappings

Not all the visual mappings used in a visualization technique can be maintained throughout the exploration process. Similar to the existing tools, when transforming from one visualization technique to another, we are required to reset some of the encodings to their initial values. For example, if size is mapped to a data attribute in a scatterplot and then the user changes the technique to a bar chart, the system resets size because it is inaccurate to have circles with different sizes where the y axis indicates the number of data items. However, we can envision creating hybrid visualization techniques and transformations in the future that allow automated swapping of data mappings to other valid visual encodings when switching techniques.

5.4 Recommendation Presentation and Timing

Although exploring different ways and timings of representing transformations is not our main focus in this work, during the design process of VisExemplar, we examined different ways of presenting recommended transformations.

Similar to many existing tools *(e.g., [4, 12, 22, 23, 33])*, we first decided to show all the recommended transformations as small thumbnails in the Recommendation Gallery. We tried this method for *Visual Representation transformations* and it worked well. However, by adding other types of transformations to the Gallery, we faced two challenges. First, we found exploring the Gallery consisting of different types of transformations shown in one place confusing. Interpreting what the recommended transformation changes from the current view to the recommended view was not clear. The second challenge was the large number of potential recommendations shown in the Gallery made it less readable. We thought it might be a better idea if we located different types of transformations in places which are still meaningful to users and easy to understand. We tried different places and finally decided to put the *Axes transformations* and *View Specification transformations* on the ThinkBoard, and *Data Mapping* transformations in the Detail view. One possible follow-up research direction includes exploring methods of presenting recommendations in visualizations and evaluating their effectiveness.

In the current version of VisExemplar, the recommended transformations will be updated in the interface whenever the recommendation table in the recommendation engine gets updated. However, one interesting research direction includes understanding the impacts of interruptions caused by incoming recommendations and investigating methods for minimizing the interruption caused by incoming recommendations. For example, one way could include presenting recommendations upon pressing a specific button on the interface. Alternatively, the system could observe the cadence of user interaction with the system and propose recommendations at a less active time.

6 Conclusion

This paper introduces Visualization by Demonstration, a user interaction paradigm for visual data exploration. The paradigm advocates for mapping visual demonstrations provided by users to meaningful visual transformations and recommending them to users. Users are able to provide demonstrations of intended changes to an existing visualization, and the system computes the appropriate transformations to accommodate the desired change as closely as possible. In order to demonstrate the technical feasibility of this paradigm, we developed a prototype called VisExemplar. VisExemplar allows users to explore their data via Visualization by Demonstration. After users provide visual demonstrations, the system recommends possible transformations based on the given demonstrations. VisExemplar computes these recommendations through a set of intent functions used to compute a best fit from the provided visual demonstrations and resulting transformations. While this paper has taken initial steps to introduce the concept of Visualization by Demonstration, there exist several exciting avenues for continued research.

7 Acknowledgements

We would like to thank the reviewers and GT Visualization Lab members for their feedback.

REFERENCES

[1] C. Ahlberg. Spotfire: an information exploration environment. *ACM SIGMOD Record*, 25(4):25–29, 1996.

[2] C. Andrews, A. Endert, and C. North. Space to think: large high-resolution displays for sensemaking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 55–64. ACM, 2010.

[3] M. Bostock, V. Ogievetsky, and J. Heer. D^3 data-driven documents. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2301–2309, 2011.

[4] F. Bouali, A. Guettala, and G. Venturini. Vizassist: an interactive user assistant for visual data mining. *The Visual Computer*, pages 1–17, 2015.

[5] E. T. Brown, J. Liu, C. E. Brodley, and R. Chang. Dis-function: Learning distance functions interactively. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, pages 83–92. IEEE, 2012.

[6] W. S. Cleveland and R. McGill. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American statistical association*, 79(387):531–554, 1984.

[7] W. S. Cleveland and R. McGill. Graphical perception and graphical methods for analyzing scientific data. *Science*, 229(4716):828–833, 1985.

[8] A. Cypher and D. C. Halbert. *Watch what I do: programming by demonstration*. MIT press, 1993.

[9] A. Endert, P. Fiaux, and C. North. Semantic interaction for visual text analytics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 473–482. ACM, 2012.

[10] A. Endert, C. Han, D. Maiti, L. House, S. Leman, and C. North. Observation-level interaction with statistical models for visual analytics. In *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*, pages 121–130. IEEE, 2011.

[11] J. A. Fails and D. R. Olsen Jr. Interactive machine learning. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 39–45. ACM, 2003.

[12] D. Gotz and Z. Wen. Behavior-driven visualization recommendation. In *Proceedings of the 14th international conference on Intelligent user interfaces*, pages 315–324. ACM, 2009.

[13] H. V. Henderson and P. F. Velleman. Building multiple regression models interactively. *Biometrics*, pages 391–411, 1981.

[14] E. Horvitz. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 159–166. ACM, 1999.

[15] S. Huron, S. Carpendale, A. Thudt, A. Tang, and M. Mauerer. Constructive visualization. In *Proceedings of the 2014 conference on Designing interactive systems*, pages 433–442. ACM, 2014.

[16] S. Huron, Y. Jansen, and S. Carpendale. Constructing visual representations: Investigating the use of tangible tokens. *Visualization and Computer Graphics, IEEE Transactions on*, 20(12):2102–2111, 2014.

[17] T. Igarashi and J. F. Hughes. A suggestive interface for 3D drawing. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology*, UIST '01, pages 173–181, New York, NY, USA, 2001. ACM.

[18] S. Kandel, A. Paepcke, J. Hellerstein, and J. Heer. Wrangler: Interactive visual specification of data transformation scripts. In *ACM Human Factors in Computing Systems (CHI)*, 2011.

[19] B. Kondo and C. M. Collins. Dimpvis: Exploring time-varying information visualizations by direct manipulation. *Visualization and Computer Graphics, IEEE Transactions on*, 20(12):2003–2012, 2014.

[20] B. Lee, R. H. Kazi, and G. Smith. Sketchstory: Telling more engaging stories with data through freeform sketching. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2416–2425, 2013.

[21] J. Lin, J. Wong, J. Nichols, A. Cypher, and T. A. Lau. End-user programming of mashups with vegemite. In *Proceedings of the 14th international conference on Intelligent user interfaces*, pages 97–106. ACM, 2009.

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

[23] D. B. Perry, B. Howe, A. M. Key, and C. Aragon. Vizdeck: Streamlining exploratory visual analytics of scientific data. 2013.

[24] D. Ren, T. Höllerer, and X. Yuan. iVisDesigner: Expressive interactive design of information visualizations. *IEEE transactions on visualization and computer graphics*, 20(12):2092–2101, 2014.

[25] A. Satyanarayan and J. Heer. Lyra: An interactive visualization design environment. *Computer Graphics Forum (Proc. EuroVis)*, 2014.

[26] D. Schroeder and D. F. Keefe. Visualization-by-sketching: An artist's interface for creating multivariate time-varying data visualizations. *Visualization and Computer Graphics, IEEE Transactions on*, 22(1):877–885, 2016.

[27] F. M. Shipman III and C. C. Marshall. Formality considered harmful: Experiences, emerging themes, and directions on the use of formal representations in interactive systems. *Computer Supported Cooperative Work (CSCW)*, 8(4):333–352, 1999.

[28] B. Shneiderman. Dynamic queries for visual information seeking. *Software, IEEE*, 11(6):70–77, 1994.

[29] SpotFire. http://www.spotfire.com, 2015.

[30] Tableau. Tableau software, http://www.tableau.com/, 2015.

[31] J. Walny, S. Carpendale, N. H. Riche, G. Venolia, and P. Fawcett. Visual thinking in action: Visualizations as used on whiteboards. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2508–2517, 2011.

[32] J. Walny, B. Lee, P. Johns, N. H. Riche, and S. Carpendale. Understanding pen and touch interaction for data exploration on interactive whiteboards. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2779–2788, 2012.

[33] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2015.

[34] J. S. Yi, Y. ah Kang, J. T. Stasko, and J. 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.

[35] M. M. Zloof. Query by example. In *Proceedings of the May 19-22, 1975, national computer conference and exposition*, pages 431–438. ACM, 1975.

Understanding Visualization Construction Strategies through Visualization by Demonstration and Manual View Specification

Bahador Saket and Alex Endert

Abstract— Visualization tools consist of two main components: visual representation and interaction. While visual representation has received the majority of attention from the information visualization community, less work has been done to investigate interaction. The dominant interaction technique used in many visualization tools is manual view specification. This method enables users to interact with a visualization tool through manipulation of interface elements (e.g., drop-down menus and check boxes) that act as mediators between users and the visual representation. Recently, we proposed visualization by demonstration, a novel interaction technique for visual data exploration and visualization construction. While we understand differences in design considerations went into each of these techniques, it remains unclear how these differences affect the process of visualization construction and data exploration, and what are the trade-offs between the two techniques. We present a two-phase study comparing people's visualization construction and data exploration process using two visualization tools: one implementing the manual view specification technique (Polestar) and another implementing the visualization by demonstration technique (VisExamplar). Findings of our study indicate that these two techniques influence: 1) the effectiveness of visualization process, 2) strategies used for constructing visualizations and exploring the data, and 3) feeling of control and engagement during the visualization process. Based on our findings, we discuss some of the trade-offs and open challenges in incorporating these techniques. In addition, we derive recommendations on how to incorporate these interaction techniques in visualization tools.

Index Terms—Manual View Specification, Visualization by Demonstration, Visualization Tools, Visual Data Exploration

1 Introduction

Visual representation and interaction are the two main components of information visualization tools [64]. The fundamental focus of visual representation is mapping from data values to graphical representations and how the graphical representations are rendered on a display. Yi et al., [64] defined interaction as a feature that can be added to visualization tools to enable users to manipulate and interpret the representation.

The visual representation component has received the majority of attention in information visualization research. A large body of prior research has designed novel techniques for mapping data to graphical representations (e.g., [6, 10, 17, 27, 54, 59]). Another body of work has studied how various properties of representation impact the effectiveness of visualizations (e.g., [7, 8, 20, 21, 31, 33, 36, 52, 57, 58]). While techniques proposed and guidelines provided by these earlier studies have tremendous impact on data visualization today, relatively less work has been done to understand and improve interaction in information visualization [39].

The primary interaction technique of majority of the existing visualization tools is *manual view specification*. Manual view specification incorporates users interaction with interface elements such as menus, sliders, and dialog boxes that act as mediators between users and the visual representation. This method requires users to interact with such interface elements to specify their visualization techniques, mappings, and parameters. For example, users may need to go through a drop-down menu to specify the visualization technique, or to change an encoding. Many of the existing visualization tools (e.g., [53, 55, 56, 60]) also adapt this form of interaction. A few studies have investigated interaction in desktop-based visualization tools (e.g., [15, 30, 34, 38, 43, 47, 51]). However, the information visualization community seems to have settled on the manual view specification technique, and there is less work in interaction design for visualization construction.

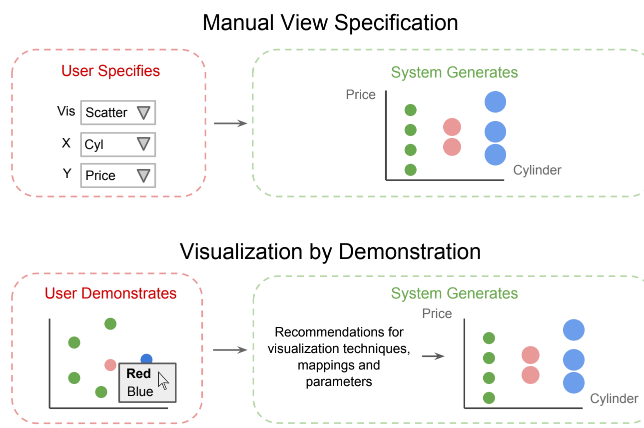


Figure 1: Two different information visualization process models. Manual view specification requires users to specify visualization techniques, mappings, and parameters. Then the system creates a visualization and renders it on the display. In contrast, visualization by demonstration requires users to demonstrate their intended changes on a visualization first. The system then recommends potential techniques, mappings, and parameters based on the given demonstrations.

We recently proposed *Visualization by Demonstration* [46], an interaction technique for visual data exploration. Visualization by demonstration allows users to provide visual demonstrations of incremental changes to the visual representation. From the given demonstrations, the system adapting visualization by demonstration then recommends potential transformations. For example, users demonstrate their interest in mapping size to a data attribute by resizing a single data point. In response, the system recommends data attributes that can be mapped to size. Visualization by demonstration advocates for balancing human and system responsibilities for visualization construction and data exploration by shifting some of the burden of specification from users to algorithms.

There are at least two main differences between design considerations went into each of these interaction techniques that could influence the ways people construct visualizations and explore their data.

- *Bahador Saket and Alex Endert are with Georgia Institute of Technology. E-mail: {saket, endert@gatech.edu}.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

First, manual view specification uses a different model of the information visualization process compared to visualization by demonstration. Manual view specification requires users to formally specify visualization technique, mappings, and parameters. In contrast, visualization by demonstration requires users to provide visual demonstrations of incremental changes to the visualization. It then recommends potential visualization techniques, mappings and parameters from the given demonstrations (see Figure 1).

Second, these two interaction techniques use different interaction metaphors. The manual view specification technique uses a conversation metaphor [25], in which it introduces interface elements such as menus and dialog boxes that act as mediators between users and the visual representation. In this case, the interface is an intermediary between the users and the visual representation. In contrast, visualization by demonstration mainly relies on a model-world metaphor [25], in which the interface is itself a visual representation and the user can act on the visual representation rather than external interface elements. While the majority of user interaction is on the visual representation, visualization by demonstration still makes use of some interface elements for accepting or rejecting the recommendations. In this case, there are fewer intermediary elements between the user and visual representation because the user can directly manipulate the graphical encodings used in a visual representation (see Figure 2).

While we understand the differences in design considerations that went into each of these techniques, it remains unclear how these differences affect the process of visualization construction and data exploration, and the trade-offs between the two techniques for specific user tasks. Understanding these trade-offs informs the visualization community, in particular, those whose goal is to design new interaction techniques or adapt existing techniques in their visualization tools. To explore these questions, we designed a two-phase study. We first conducted a controlled experiment to study the effectiveness (performance time and accuracy) of each interaction technique in constructing visualizations. We then conducted an exploratory study to investigate which processes people follow while exploring their data, which common patterns appear, and which barriers people encounter using each interaction technique. Our study resulted in several findings. In particular, we found that:

- People often take two different strategies to construct visualizations using the interaction techniques: **specific** and **abstract** strategies. In the specific strategy the participants know the exact information needed for creating or refining a visualization. In the abstract strategy, the participants were unaware of some of the information required for creating or refining a visualization. We also found that each of the interaction techniques supports one of the strategies better than another.

- Each interaction technique is more effective for different categories of tasks. For example, manual view specification was significantly faster when the participants were asked to map data attributes to axes or switch between the visualization techniques. However, visualization by demonstration was faster in mapping data attributes to visual encodings.

- Participants who used the manual view specification technique always took the same linear path throughout their visualization construction process *(define a goal → identify data attributes → specify axes → specify marks → specify visual encodings)*. In contrast, we could not find a generalizable pattern that all participants follow while using visualization by demonstration technique.

The remainder of the paper is organized as follows. We first discuss related work on interaction in desktop-based visualization systems. We then explain procedures and main findings of our pilot studies. In the light of our pilot studies, we then explain our research questions and details of our two-phase study. In Discussion section we discuss our findings, trade-offs, and open challenges in incorporating each of these interaction techniques. We also provide possible research avenues for continued research.

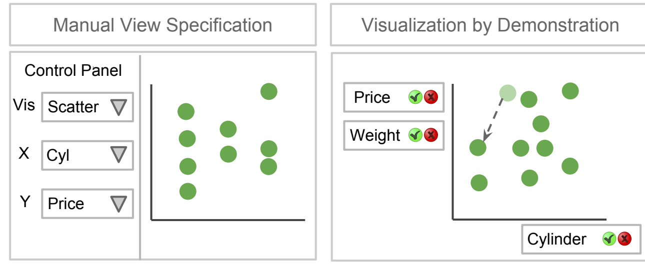


Figure 2: In manual view specification, users specify visualization techniques, mappings, and parameters through manipulation of the mediators (e.g., check boxes or drop-down menus). In visualization by demonstration, users demonstrate part of the intended specifications by directly manipulating the visual representation. The system then recommends potential visualization techniques, mappings, and parameters based on the given demonstrations.

2 RELATED WORK

Understanding Interaction: Yi et al. [64] define interaction in information visualization as a set of features that provide users with the ability to directly or indirectly manipulate the visualization specifications. They recognize interaction as one of the main components of the visualization tools [64]. For example, a drop-down menu for assigning a data attribute to an axis of a barchart is an interaction technique since it enables users to manipulate the visualization specifications. Pike et al. [39] discussed interaction challenges raised in the visual analytics research and the relationship between interaction and cognition. A few other studies have mainly focused on the interaction component of information visualization (e.g., [5, 14, 30, 34, 38, 64]), but these are relatively uncommon when compared to previous work investigating the visual representation component.

Interaction in Existing Visualization Tools: While many visualization tools produce similar outcomes with different user interfaces, the primary interaction technique of a majority of the existing visualization tools is manual view specification (e.g., [40, 53, 55, 56]). Manual view specification requires users to specify visualization's components and characteristics through a set of interface elements that act as mediators between users and the visual representation. For example, specifying the data attributes on the axes using drop-down menus in the user interface. While the main interaction technique in these tools is manual view specification, interaction designs in them might be slightly different. For example, some of these tools need users to specify visualization characteristics through cursor operations (drag and drop), while others allow visualization specification through menus and dialogs.

Another set of tools are visualization tools such as Lyra [48], iVisDesigner [42], and Data-Driven Guides [29] that enable visualization designers to construct more customized visualizations. The primary interaction technique in many of the existing visualization authoring tools is also manual view specification. For example, Lyra [48] requires users to specify the axes of a visualization by dragging and dropping the data attributes on the axes or specify the visualization characteristics through user interface elements presented in a control panel.

Visualization recommendation tools are another category of systems that aim to balance the responsibility of visualization specifications between the user and the system. Different visualization recommendation systems have been developed to assist users to construct visualizations (e.g., [18, 37, 62]). While these tools are designed to facilitate breadth-oriented exploration, the primary interaction technique in these tools is still manual view specification. For example, Voyager [62] is a visualization recommendation tool that recommends visualization based on user-specified data attributes of interest. Voyager requires users to specify the data attributes of interest through a control panel.

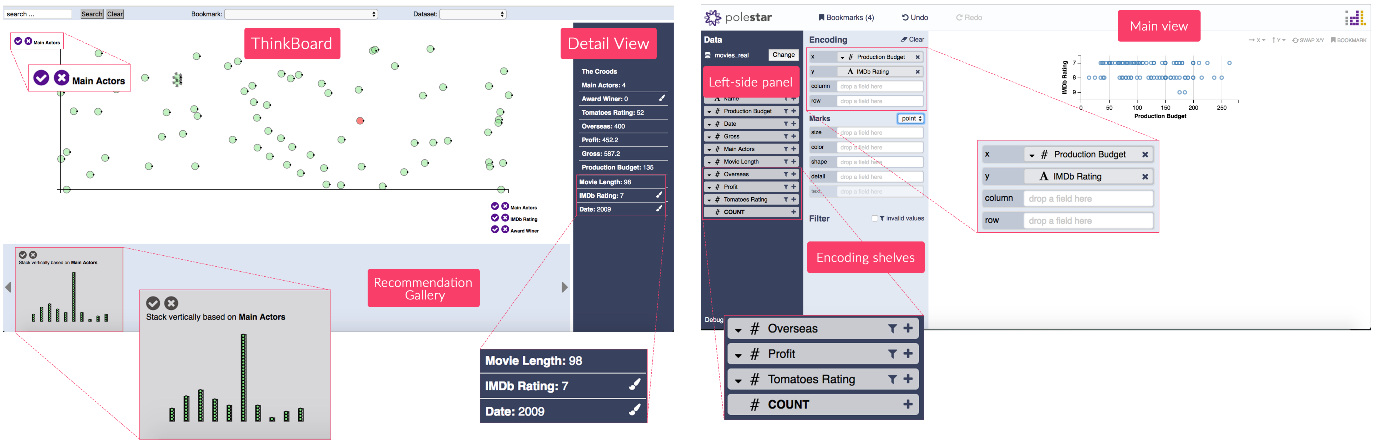


Figure 3: A scaled-down screenshot of VisExamplar (left) and Polestar (right).

Other Interaction Techniques for Visualization Tools: Prior work has investigated new techniques for interacting with visualizations on touch-screen devices (e.g., [2, 13, 26, 35, 44, 45, 61]). Several projects from the visual analytics community have designed novel techniques to interact with parameters of the model used for computing the structure of visualizations (e.g., [4, 15, 16, 28, 32]). For example, AxiSketcher is a tool that allows users to revise nonlinear axes of scatterplot by direct interaction with graphical encodings [32].

Inspired by direct manipulation [50], Kondo and Collins proposed DimpVis [30] interaction technique for temporal navigation. DimpVis enables users to directly manipulate length, angle and position of the glyphs in the visual representations, as a means for temporal navigation. For instance, to check if at any point in time the value associated with a bar is half its current value, the user can drag the bar vertically downwards to compare its values at different points in time. In a controlled experiment that they conducted, they found that the participants preferred using DimpVis over manual view specification. Their findings also showed that the technique used in DimpVis was significantly faster than manual view specification technique for temporal navigation tasks. Inspired by previous work [1, 23, 24], we recently introduced visualization by demonstration [46], in which users can directly manipulate graphical encodings to provide visual demonstrations of incremental changes to the visual representation. Visualization by demonstration then advocates for recommending potential transformations based on the given demonstrations.

While DimpVis is designed for querying temporal navigation, manual view specification and visualization by demonstration are suited for more holistic data exploration and visualization construction. Our goal in this study is to investigate the trade-offs between the interaction techniques for a more holistic visualization construction and data exploration process. We also did not include interaction techniques designed for touch-based devices because we found it difficult to control for external factors (different input devices and interfaces) that might affect the study. Additional studies are required to investigate interaction techniques that use input devices other than mouse and keyboard, and interfaces other than desktop devices.

3 Visualization Tools

The main experiment of this study entailed the use of two visualization tools, VisExamplar [46] and Polestar [40], which satisfied two requirements. One is that each had to clearly embody one of the two interaction techniques examined in this paper. The other is that to have a fair comparison between these two techniques, users had to be able to learn and use the system within the duration of the experiment. As each tool adopted only one of the two interaction techniques, we compared VisExamplar, which incorporates visualization by demonstration and Polestar, which incorporates manual view specification. We previously developed VisExamplar to show the feasibility of the visualization by demonstration paradigm [46]. For this study we extended the previous implementation by adding some additional features and improving some of the underlying algorithms. Although a variety of commercial tools incorporate the manual view specification technique, we decided to use the less complicated visualization tool, Polestar, to control for external factors that might affect the study. Also, Polestar has previously been used in other studies to contrast recommendation browsing technique with manual view specification [62, 63].

VisExamplar: VisExamplar[1] tool (shown in Figure 3) consists of two main components: demonstrations provided by users to show their intended actions and transformations that are recommended by the system in response to the given demonstrations. To provide demonstrations, VisExamplar supports two methods. One is that it allows users to directly adjust the spatial layouts of data points (e.g., users stacking data points in the shape of bars to convey their interest in a barchart) and the other, which allows users to provide demonstrations by adjusting the graphical encodings used in a visualization (e.g., users changing the size of data points in a scatterplot). In response to the provided demonstrations, VisExamplar recommends four categories of transformations (change the current visualization technique, define mappings between graphical encodings and data attributes, assign data attributes to axes of a visualization technique, change the view specifications without changing the underlying technique). By accepting any of the recommended transformation the system will change the corresponding view.

The VisExemplar user interface consists of a ThinkBoard, Recommendation Gallery, and a Detail View panel. Users can construct their demonstrations through direct manipulation of the visual representation on the ThinkBoard. Some of the recommendations (e.g., recommending data attributes to be mapped to axes) might also be shown on the ThinkBoard. Other recommendations will be presented in the Recommendation Gallery. The primary goal of the Detail View panel is to show data attribute types and values for a selected data point. For more details about the VisExamplar please refer to main work which describes this system [46].

Polestar: Polestar[2] (shown in Figure 3) is a visualization tool that implements the manual view specification paradigm. Polestar is Tableau-style user interface for visual analysis, building on top of Vega-Lite [49]. Polestar user interface consist of a Left-side panel, Encoding Shelves, and the Main View. The Left-side panel presents the data schema, listing all data attributes in the dataset. Encoding Shelves located next to the data schema and represent different encoding channels. Users can drag and drop a data attribute onto a shelf to establish a visual encoding. They can also change properties of the data (e.g., data types, data transformations) or the visual encoding variable (e.g., color or sort) via pop-up menus. Main View of the Polestar shows a created visual representation. Upon user interaction, Polestar updates the visual representation shown on the main view.

[1] `https://github.com/BahadorSaket/VbD`
[2] `https://github.com/vega/polestar`

4 MAIN FINDINGS OF OUR PILOT STUDIES

Our study design in this paper was shaped by two pilot studies. Below, we describe goals, procedure, and main findings of both pilot studies in more details.

Our goal in the first pilot study was to observe patterns that people use to construct visualizations. In addition, we wanted to capture potential flaws in our main study design and understand if the datasets planned to be used are appropriate enough to support 20 minutes data exploration task during our main experiment. For the first pilot study, we recruited six participants. We randomly assigned three of the participants to work with VisExamplar and the others to work with Polestar. We first trained the participants for few minutes. We then asked participants to perform eight trial tasks using the tool. Tasks for trial session were designed using the Cars dataset [22]. Once they completed the tasks, we asked them to work with the visualizations tool for 20 minutes to explore the Cameras dataset [12]. The participants were asked to verbalize their data exploration process.

We captured two strategies that the participants used to construct or refine a visualization. In the first strategy, the participants knew the exact information required for constructing or refining a visualization. For example, they knew which data attribute should be mapped to which visual encoding or axis. We call this a ***specific*** strategy. The second strategy is more abstract. Users sometimes were unaware of some of the information required for completing constructing or refining a visualization. For example, the participants sometimes tried to express that they want to map an attribute to the size by saying, *"I want to make expensive cameras bigger."* We call this an ***abstract*** strategy. The key difference between these two strategies is how well the users can articulate their goals based on the data attributes and visualization characteristics. When goals are formed in the way of data attributes and visual mappings, the specific strategy was used. Alternatively, when goals are formed on data items and semantic relationships between data items, abstract strategies were used.

We initially decided to use the Cars [22] and Cameras [12] datasets in our main study. However, it turned out that the participants were not familiar with all data attributes used in the Cameras dataset (or cameras in general), so they tended not to explore for as long and make more impetuous decisions and not examining attributes that they were not familiar with. We instead decided to use the Movies dataset [12] that provides details for 335 movies released from 2007 to 2012, and contains 12 data attributes. We selected the Cars and Movies dataset for our experiment based on two considerations. First, the datasets contained enough data attributes to support 20 minutes data exploration task. Second, the participants were unfamiliar with the content of the datasets but familiar with the meaning of the data attributes used in the dataset (e.g., participants knew the meaning of IMDb rating, profit, gross, genre, and etc.).

In the second pilot study, our goal was to check if visualization construction strategies used in the questions will affect people performance time and accuracy. We carried out with four additional participants. We designed 8 trial visualization construction tasks. For half of the questions we provided all the information participants required to perform the tasks (specific strategy). For the second half, we explained the tasks in a more abstract way (abstract strategy). We noticed that the way that the questions are asked would affect users' performance time and accuracy. Thus, we decided to take into account visualization construction strategy as one of the factors in our main experiment.

5 RESEARCH QUESTIONS

The goal of our study is to investigate trade-offs between the two interaction techniques implemented in these visualization tools rather than the tools themselves. Therefore, we carefully designed the tasks to focus on basic interactive tasks, rather than higher-level tasks which rely on the holistic design and usability of the two visualization tools. Based on our prior knowledge about design differences between the two interaction techniques (different models of information visualization process and interaction metaphors used in each of them) and observations during the pilot study, our research questions are:

- **How do the differences between two interaction techniques affect the visualization construction process?**
- **How well each of the visualization construction strategies is supported by each interaction technique?**

6 RESEARCH METHODOLOGY

To address these research questions, we conducted a two-phase study. In the first phase, we studied how well each interaction technique supports visualization construction in a more controlled setting. We measured the effectiveness (performance time and accuracy) of each interaction technique for four different types of visualization construction tasks (assigning data attributes to axes of a visualization, mapping visual encodings to data attributes, switching from one visualization technique to another, and reconfiguring the visualization). Each of the tasks were asked using either specific or abstract strategies. In the second phase, we investigated how well visualization construction is supported by each technique using a more realistic setting. We conducted a think-aloud exploratory observational study in a laboratory setting where participants were asked to use a visualization tool to explore a dataset. Participants were asked to walk the investigator through their exploration process.

6.1 Participants

To recruit participants, we announced our study in a visualization class, posted to student mailing lists, and put up flyers on bulletin boards across campus. We recruited 16 participants (7 male, 9 female) aged 21 - 32 years. The participants were undergraduate and graduate science and engineering students. All participants had some experience creating visualizations using different visualization tools such as Microsoft Excel. A statistical power analysis was performed for sample size estimation, based on data from pilot studies. The effect size in this study was 0.6, considered to be medium using Cohen's criteria [9]. With an $alpha = .05$ and $power = 0.80$, the projected sample size needed with this effect size is approximately N= 14.

6.2 Setting and Apparatus

The study was conducted in a usability lab. During the entire study participants used a computer with 2.7 GHz Intel Core i5 processor and 13 inch screen with 1680×1050 pixel resolution. Participants interacted with a mouse to complete the tasks. The study took about 1 hour to complete and participants were compensated with a $10 Amazon gift card.

6.3 Phase 1: Controlled Experiment

In this phase, we examined the effectiveness of each interaction technique for creating visualizations using four different types of tasks in a controlled setting. We used a between-subjects designs. 16 subjects participated in our study were randomly assigned to one of the visualization tools (8 participants per visualization tool). Each participant worked with just one of the visualization tools.

Tasks

To select tasks for construing visualizations, we first considered user interactions with VisExamplar and Polestar. We interacted with both tools for a week exploring different ways in which they support visualization construction and reconfiguration. This resulted in a list of 45 tasks. We then encapsulated these tasks into four categories according to the type of changes they make to a visualization.

- **Mapping data attributes to the axes:** This category of task requires users to assign data attributes to either one or both axes of a visualization.
- **Mapping data attributes to visual encodings:** This category of task requires users to map a data attribute to a visual encoding.
- **Switching between visualization techniques:** This type of task requires users to change from one visualization technique to a different visualization technique.

Table 1: The table provided examples of the tasks used in the first phase of our study. For each category we provided two phrasings (abstract and specific). The phrasings are based on how participants verbalized their goals during the think aloud protocol of the pilot study.

TYPES OF OPERATIONS	SPECIFIC STRATEGY	ABSTRACT STRATEGY
Mapping data attributes to the axes	`Starting point:` A visualization (either bar chart or scatterplot). Data attributes assigned to the axes are different from those mentioned in the task. `Task Example:` Change the scatterplot so that it has wheelbase as the x axis.	`Starting point:` A visualization (either bar chart or scatterplot). Data attributes assigned to the axes are different from those mentioned in the task. `Task Example:` Change the representation so that circles are horizontally positioned by price.
Mapping data attributes to visual encodings	`Starting point:` A visualization (either bar chart or scatterplot). Visual encodings (e.g., color or size) used in the visualization are not mapped to any data attribute. `Task Example:` Change the scatterplot in a way that color is mapped to engine size of the cars.	`Starting point:` A visualization (either bar chart or scatterplot). Visual encodings (e.g., color or size) used in the visualization are not mapped to any data attribute `Task Example:` Change the representation so that cars with the same number of cylinders have the same color.
Switching between visualization techniques	`Starting point:` A visualization technique other than the one that participants are supposed to construct. Attributes assigned to the axes of the visualization are different from those mentioned in the task. `Task Example:` Switch from the given bar chart to a scatterplot where the x axis is price and y axis is wheelbase.	`Starting point:` A visualization technique other than the one that participants are supposed to construct. Attributes assigned to the axes of the visualization are different from those mentioned in the task. `Task Example:` Modify the given representation such that cars are horizontally positioned by horsepower and vertically positioned by engine size.
Reconfiguring the visualization	`Starting point:` A bar chart visualization. `Task Example:` Sort the given bar chart in ascending order.	`Starting point:` A bar chart visualization. `Task Example:` Order the bars from left to right. Shortest bar is on the left and the tallest bar is on the right.

- **Reconfiguring a visualization:** This category of task requires users to change the view specification of a visualization without changing the underlying technique and mappings.

For this phase of our study, we designed 16 tasks for participants to perform (4 categories of tasks $\times$ 2 strategies $\times$ 2 trials). Each participant performed all 16 tasks using one of the visualization tools. For the tasks in each category, we derived two alternative phrasings (abstract and specific). For two of the trials in each category we provided exact information required for completing them (specific strategy), for another two trials we were more abstract (abstract strategy). The phrasings of trials (abstract and specific) are based on how participants verbalized their goals during the think aloud protocol of the pilot study. Table 1 shows examples of these tasks for different categories. In this table we only provided one of the two trials used for each category. The List of all 16 tasks is provided in our supplementary materials [3].

Hypotheses

Since the two interaction techniques have their own characteristics, we expect that each technique will have its advantages and disadvantages. Based on informal studies we conducted previously during the development process of VisExamplar [46], pilot studies, and our own experiences with both interaction techniques, we considered the following hypotheses for our study:

- **H1:** We hypothesize that using manual view specification, participants map data attributes to axes and switch from one visualization technique to another significantly faster and more accurate than the visualization by demonstration technique. However, mapping visualization encodings to data attributes and reconfiguring visualizations would be significantly faster and more accurate using visualization by demonstration technique.

- **H2:** We expect the manual view specification technique to support the specific strategy of visualization construction better (significantly faster and more accurate) because it allows participants to map their mental operations to the corresponding interface operations. However, we hypothesize the visualization by demonstration technique supports the abstract strategy better because it allows users to convey their mental picture of the visualization to the system by directly manipulating the visual representation.

[3] `https://github.com/gtvalab/VbD-Study`

Training

Before starting the main experiment, the participants were briefed about the purpose of the study, and their rights. At this stage, the participants were also asked to answer to some demographic questions (e.g., age, sex, and prior experience in creating visualizations). Each participant was asked to work with one of the visualization tools. We first walked the participants through the training session to familiarize them with the study. As our participants had no prior experience using these particular tools, we reduced their initial learning time by offering a brief introduction to the tool they would use. To prevent inconsistencies in the training session, we asked participants to watch a tutorial video of the visualization tool (we created the video prior to the study). This enabled all the participants to go through the same instruction process during the training. The video walked the participants through different features and interactions provided by the tool. The participants were allowed to watch the video as many times as they want. After watching the video, we asked participants to work with each tool for 10 minutes. In addition, we encouraged the participants to ask as many questions as they want during this stage. We then asked participants to perform 8 training tasks (4 types of tasks $\times$ 2 levels of explicitness $\times$ 1 trial). The participants were not allowed to move to the next training question unless they answered the question correctly. We did not record the time and accuracy during training session. Once comfortable with using the visualization tool, users were instructed to take a short break and move to the main study.

Procedure

In this phase, participants performed 16 visualization construction tasks: 4 types of tasks $\times$ 2 strategies $\times$ 2 trial. All tasks were printed on a sheet of paper. Each time interviewer selected a task randomly and asked the participants to perform the task as fast and accurately as possible. Before performing each task, participant were given a visualization as a starting point. This way we made sure that all the participants performed each task starting form the same visualization. We measured participants performance time and accuracy. To design tasks for this phase, we used the Cars [22] dataset. The Cars dataset [22] provides details for 407 new cars and trucks for the year 2004. This dataset contains 18 data attributes describing each car.

Data Analysis

To address our first hypothesis (**H1**), we needed to test how the different tasks were performed using each interaction technique in terms of time. To analyze the differences among the various types of tasks, we first calculated separate mean performance time for all trials. That is, for each participant, we averaged outcome values of trials for type of task. We initially planned to take into account both performance time and accuracy in our analysis. However, the participants performed all the tasks correctly using both techniques, so we excluded accuracy from our analysis. We then conducted a mixed analysis of variance (ANOVA) to test for differences among the four types of tasks (within-subjects factor) using two interaction techniques (between-subjects factor). The main effect of interaction technique indicates which technique produces the best performance, regardless of the task. The task $\times$ technique interaction indicated whether a particular interaction technique works better with a particular task.

To address our second hypothesis (**H2**), we conducted the second mixed ANOVA to test for differences among the two different strategies of constructing visualizations (within-subjects factor) using two interaction techniques (between-subjects factor). In particular, we were interested in interaction between the two different visualization construction strategies and interaction techniques (visualization construction strategy $\times$ interaction technique). Investigating the interaction between visualization construction strategies and techniques indicated whether a particular interaction technique works better with a specific strategy (abstract and specific). That is, for each participant, we averaged outcome values of trials.

Before testing, we checked that the collected data met the assumptions of appropriate statistical tests. The assumption of normality was satisfied for parametric testing, but Mauchly's Test of Sphericity indicated that the assumption of sphericity had been violated for time. To address this issue, we report test results with corrected degrees of freedom using Greenhouse-Geisser estimates for $\varepsilon < 0.75$ and otherwise with Huynh-Feldt correction. We provide all relevant materials for this study online [4].

Results

We found a significant effect of performance time for interaction technique ($F(1,14) = 19.6, p < 0.05$) with a slightly large effect size ($\eta_p^2 = 0.63$). Overall, manual view specification was three seconds faster than visualization by demonstration across all tasks. We also found a significant interaction between techniques and tasks for performance time ($F(1,14) = 16.8, p < 0.05$) with a slightly large effect size ($\eta_p^2 = 0.56$). Our results show that the participants mapped data attribute to axes significantly faster using manual view specification compared to visualization by demonstration. We also found that manual view specification was significantly faster than visualization by demonstration in switching from one visualization to another. However, the participants were significantly faster in mapping data attributes to encodings and reconfiguring visualizations using visualization by demonstration. Our results partially confirms our first hypothesis (**H1**). The performance time results are summarized in Table 2.

We also found a significant interaction between techniques and visualization construction strategies for performance time ($F(1,14) = 34.5, p < 0.05$) with a large effect size ($\eta_p^2 = 0.71$). The participants preformed tasks significantly faster using abstract strategy ($M = 8.1$ $[6.9, 9.3]$) than specific strategy ($M = 10.2$ $[9.1, 12.5]$) using the visualization by demonstration technique ($p < 0.001$). However, this was the opposite in the manual view specification technique. The participants were significantly faster in performing tasks using specific strategy ($M = 6.5$ $[4.8, 8.2]$) than abstract strategy ($M = 8.0$ $[6.8, 9.1]$) in manual view specification technique ($p < 0.05$). This confirms our second hypothesis (**H2**). We defer the discussion of the results for this phase to the later section along with findings of our observational study.

[4] `https://github.com/gtvalab/VbD-Study`

Table 2: Means of completion times (M) in seconds for Visualization by Demonstration (VbD) and Manual View Specification (MVS). The standard deviation (STD) is indicated as well. Significant differences in completion time are indicated by $\star$. Significantly faster results are highlighted in bold. All p-values are Bonferroni-corrected.

Task Type	VbD	MVS	p-val
Mapping data attribute to axes	M=11.9 STD=3.1	**M=6.1** **STD=1.2**	$< .001 \star$
Mapping data attributes to encodings	**M=5.7** **STD=1.0**	M=7.59 STD=1.5	$< .05 \star$
Switching between techniques	M=16.7 STD=7.6	**M=10.6** **STD=1.7**	$< .05 \star$
Reconfiguring a visualization	M=3.3 STD=0.6	M=4.3 STD=0.4	$= .06$

6.4 Phase 2: Free Exploration

In this phase, we conducted a think aloud exploratory observational study to understand how the participants use each interaction technique to construct visualizations in a more realistic setting. We then interviewed the participant to elicit difficulties they encountered using each interaction technique. During this phase, we recorded the participants audio and the screen of the computer they worked with.

Procedure

After completing the tasks in the first phase, we asked the participants to take a five minutes break. The participants were then asked to explore the Movies dataset [12] and look for interesting findings about the data. In particular, the participants were told to imagine their employer asked them to analyze the dataset using the visualization tool for 20 minutes and report their findings about the data. The participants were instructed to verbalize analytical questions they have about the data, the tasks they perform to answer those questions, and their answers to those questions in a think-aloud manner. In addition, we instructed them to come up with data-driven findings rather than making preconceived assumptions about the data. The participants were not allowed to ask any question during this phase. We tried to avoid interrupting the participants as much as possible during their data exploration process. However, we sometimes needed to remind the participants that this is a think-aloud study and encouraged them to verbalize their thoughts.

Follow-up Interview

This phase of our study concluded with a follow-up interview, in which we asked participants about what they liked and disliked about the interaction technique. This was to allow the participants to convey their feedback and ideas and in order to solicit potentially unexpected insights. We asked participant to explain what they liked or disliked about the interaction technique implemented in the visualization tool they used. We instructed the participants to give feedback on the interaction technique incorporated in the tool rather than the user interface.

Data Analysis

We used the qualitative data analysis approach proposed by Creswell [11] to analyze the video material collected in our study. The material was then coded by a single coder (the first author) in several passes. In each pass, the coder developed, refined, and consolidated the codes. The videos were coded using the Boris[5] qualitative data analysis software. We coded processes of the participants in terms of **usage** (what types of visualization specifications were usually created using each interaction technique?) and **barriers** (when and how difficulties happened while working with each technique?). We identified frequently occurring codes in our data and used them to form the higher-level descriptions, a technique known as focused coding. Coding, interpretation, verification, and comparison were ongoing and iterative throughout the course of the research.

Results

The video analyses revealed types of specifications the participants created during the entire visualization construction process. We divided these specifications into four categories: mapping data attribute to axes, mapping data attributes to encodings, switching from a visualization technique to another, and reconfiguring a visualization.

Mapping Data Attributes to Axes: Participants tended to map more data attributes to axes using manual view specification (see Table 3). Fast speed of the manual view specification technique in mapping data attributes to axes might have contributed to this advantage. Some of the participants who used visualization by demonstration found it difficult to demonstrate their interest in mapping a data attribute to an axis. To do this, they had to position a few data points relative to their data attribute values. The system then recommended potential data attributes to be assigned to the axes. Some of the participants expressed difficulties in mapping data attributes to axes using the visualization by demonstration technique. For example, one participant expressed how a large amount of effort was required for him to map a data attribute to an axis: *"I need to drag a point, keep track of its value, and compare its position with other points."* Later during the follow-up interview, the same participant mentioned: *"you know it is hard to drag the points and track their values but I like that the system gives me this flexibility and control ... maybe you could somehow highlight the values [data attribute values] while moving the points to decrease users cognitive load."*

Another challenge that a few of the participants encountered while using visualization by demonstration was the accuracy of the data attributes suggested to be mapped to the axes. After a position-changing interaction, the system searches for data attributes to recommend for map to the axes based on the positions of the moved data points. The recommendation engine prioritizes potential suggestions and shows those above a certain threshold. However, there might be cases where a user's expected data attribute was not among those suggestions that the system recognized to be the most related ones. In such cases, users had to provide more demonstrations but the providing more demonstrations frustrated them. One of the participants mentioned her concern by saying: *"The recommendations on the axes don't always make sense to me, so when I have an idea in mind like let me see how these [data attributes] compare and when I don't see it in the options, I am kinda thrown off because at that point I kinda doubting whether the way that I am thinking about it is wrong or I am doing something wrong with the system."*

Mapping Data Attributes to Encodings: The participants mapped more data attributes to visual encodings using visualization by demonstration technique (see Table 3). To map a visual encoding to a data attribute using visualization by demonstration, users could directly manipulate characteristics (e.g., size) of a corresponding encoding in the visual representations. For example, users could color one or more data points red to convey their interest in mapping this specific color to a data attribute. The system then recommends a set of data attributes that can be mapped to color. In the early stages of our data analysis, we noticed participants' interest in mapping data attributes to encodings using the visualization by demonstration approach. We then had to go through the videos and interviews to capture reasons behind this interest. We noted two interesting findings in participants' exploration process.

[5] http://www.boris.unito.it/

Table 3: This shows the total and average number of times that participants created each type of visualization specification using Visualization by Demonstration (VbD) and Manual View Specification (MVS).

Visualization Specification Type		VbD	MVS
Mapping data attributes to axes	Total	53	108
	Avg	7.5	15.4
Mapping data attributes to visual encodings	Total	55	26
	Avg	3.9	1.9
Switching between visualization techniques	Total	12	25
	Avg	1.4	3.5
Reconfiguring a visualization	Total	7	9
	Avg	0.8	1

First, participants found mapping data attributes to encodings fast using visualization by demonstration. In such cases, participants had a clear idea of which data attribute they want to map to color or size. They found visualization by demonstration fast because unlike manual view specification, they did not have to look for a specific data attribute among all data attributes and assign it to an encoding. Instead, using visualization by demonstration they could color or resize data point(s) directly, the system then would recommend a subset of data attributes that they could assign to color or size. For instance, one participant mentioned that: *"I really liked how simple and intuitive it was to change color and size."* Another participant express this by saying: *"... coloring points was fast though ... I can color one point and the system suggests me attributes."*. The results of our quantitative analysis in the first phase (Phase 1) also indicated that the visualization by demonstration technique is significantly faster than manual view specification for mapping a data attribute to an encoding.

Second, we saw an interesting pattern emerge when the participants did not intend to map any specific data attribute to an encoding but wanted to explore different mapping options by hovering on the recommended data attributes. For example, one of the participants mentioned: *"... lets color one and to look at recommendations [recommended data attributes]"* This is interesting because if the participants wanted to do this using manual view specification they had to assign a data attribute to an encoding everytime to see different options. However, this was the opposite in visualization by demonstration. The participants could color or resize a data point to get data attributes recommended to them by the system. They then would hover on those recommended attributes to preview the results and explain their findings. We noted that participants really appreciated the capability of the system in recommending the data attributes and enabling them to quickly preview the mappings. For example, one participant who had experience using other visualization tools mentioned: *"in other tools that I used for 7450 [InfoVis class taught at Georgia Tech] we had to assign an attribute to color for each attribute separately but here [using visualization by demonstration] is easier because I can color one circle [data point] and the system gives me subset of attributes and I can hover on them to see [preview] the results."*

Third, the participants felt more control over the tool when they were mapping data attributes to visual encodings using visualization by demonstration. This is potentially because the technique enables them to directly manipulate characteristics (e.g., size) of encodings. This way the interaction between the users and the system is very direct and there is no intermediary between the users and the visual representation. One participant expressed his feeling of having control by saying: *"I like that I can color it here [directly changing the color of the encoding], I feel like I have control over the circles [data points]"*.

Switching Between Visualization Techniques: The participants switched between visualization techniques more often while using the manual view specification technique (see Table 3). The ability to quickly change from one visualization technique to another could contribute to this advantage. In particular, the participants found it quite difficult to switch from a scatterplot to a barchart using the visualization by demonstration technique. To switch from a scatterplot to barchart using visualization by demonstration, participant had to stack two or more data points vertically. The system then recommended a set of barcharts based on similarity of the data points. The participants found stacking data points difficult. One participant expressed the difficulty of switching from a scatterplot to a barchart by saying: *"At the beginning it was a bit hard and awkward for me to stack the points to create a barchart but after a while it became natural."*

Reconfiguring a Visualization: We did not find a large difference in the number of times that the participants reconfigured the visualizations using each interaction technique. In fact, as the results of our first phase also indicates both interaction techniques were quite fast in reconfiguring visualizations. However, we noted several participants found sorting the bar chart using visualization by demonstration paradigm intuitive and fun. For instance, one of the participants mentioned: *"... the sorting was intuitive ..."* Another participants said: *"interactions like sorting are fun and natural. Have you ever thought to test your tool on high school students? I think they will like it a lot because they can move things around and play with it while they are learning."* Visualization by Demonstration allows user to sort the barchart by dragging the shortest/tallest bar to extreme left/right. For example, users could drag the shortest bar to the extreme left to demonstrate they are interested in sorting the bar chart. The system then would recommend sorting the bar chart in ascending order. Direct interaction of the participants with the bars might contribute in making this form of interaction intuitive and fun.

Flow of the Visualization Construction Process

In manual view specification, all participants followed a linear visualization construction and data exploration process. As we observed in video analysis, the linearity in visualization construction process was as follow: 1) participants created a goal/task to perform, 2) identified the data attributes related to the task at the hand, 3) selected a visualization technique/mark type to represent the data, 4) mapped data attributes to visual encodings, 5) engaged in interacting with data items by hovering on them to get details. This linear process was repeated multiple times by all participants during the data exploration process observed in Phase 2 of our study.

In visualization by demonstration, participants also first created goals/tasks and then identified the data attributes relevant to their task. However, they then took different paths to construct visualizations and explore their data. Since visualization by demonstration shows a visual representation at the beginning, participants can take completely different paths to construct visualizations and explore their data. Some of the participants first mapped data attributes to visual encodings (e.g., assigned genre of movies to color). Other participants who did not have an idea in mind about how to represent data attributes started their process by reading and comparing data points values for specific data attributes, they then gradually started visualization specification. That is, we could not find a single generalizable pattern that all participants follow while using this technique.

7 Discussion

In this section, we reflect on the results of our work more broadly with respect to interaction for visualizations. We discuss several aspects of our work, implications of the study findings, provide a series of guidelines on improving interaction techniques for visualizations.

Support Both Specific and Abstract Strategies

As we discussed earlier, the participants in our study used two different strategies to construct visualizations: specific and abstract. A few of the participants always used the specific strategy, a few of them always used abstract strategy, and majority of the participants used combination of both specific and abstract strategies for visual construction and data exploration.

Our results in the first phase showed that the participants were significantly faster in performing tasks using specific strategies ($M = 6.5$ $[4.8, 8.2]$) than abstract strategies ($M = 8.0$ $[6.8, 9.1]$) using manual view specification technique ($p < 0.05$). This means that this form of interaction is more effective for cases where users are aware of exact information required for constructing or refining a visualization. On the other hand, the participants were significantly faster using the abstract strategy ($M = 8.1$ $[6.9, 9.3]$) than the specific strategy ($M = 10.2$ $[9.1, 12.5]$) using the visualization by demonstration technique ($p < 0.001$). This means that this technique can support cases where participants are less aware of some of the information required for constructing or refining a visualization.

Previous work indicated that visualization novices have less knowledge about visualization techniques, mappings, and parameters [19, 23]. As Grammel et al. [19] discussed, the major barrier faced with many novices while constructing visualizations and exploring their data was that they often had partial information about the details of the intended specifications. For example, novices mentioned that they want *"a mapping so that more sales relate to a larger circle"* instead of saying *map sales to size of the circles.* We hypothesize that the type of strategy that users apply to construct visualization depends on their level of visualization expertise. In particular, we believe that visualization experts with a good knowledge of visualization domain would prefer specific strategy over abstract strategy, but this remains to be formally studied. However, even visualization experts may prefer to start with a by-demonstration approach then refine the visualization through manual view specification.

An important avenue for continued research is the design and evaluation of systems that support both visualization construction strategies. One possible solution is to say we design two types of visualization tools. One type adapts the manual view specification technique to better support specific strategy and another incorporates the visualization by demonstration technique to better supports abstract strategy. However, majority of users who apply a combination of both specific and abstract strategies might not benefit from any of the two visualization tools. We can envision a system that combines both manual view specification and visualization by demonstration techniques. This way users users would benefit the most from using visualization tools in their data exploration process regardless of the strategy they use to construct visualizations.

How to Improve Recommendations?

One of the main goals of visualization by demonstration is to balance the human and system responsibilities for visual data exploration by shifting some of the burden of specifications from users to algorithms. To that extent, the system recommends transformations in response to the demonstrations provided by users. To better understand the impact of the recommended transformations on the participants' visualization construction and data exploration process, we observed user interaction with different types of recommendations.

Recommendation Reasoning: While using visualization by demonstration, there were cases where the participants were unclear why the system proposed specific recommendations. In such cases, the participants found it difficult to map the recommended options to their interaction with the visualization. For example, one of the participants mentioned: *"The recommendations on the axes don't always make sense to me, so when I have an idea in mind like let me see how these [data attributes] compare and when I don't see it in the options, I am kinda thrown off because at that point I kinda doubting whether the way that I am thinking about it is wrong or I am doing something wrong with the system."* Another participant stated: *"If the system could explain me why it is suggesting me these barcharts, I would then adjust my actions to get more related suggestions."* All recommendation systems suggest potential options to users based on a specific set of criteria. There might be cases that the systems do not recommend options expected by the users and it might not be clear to users why those recommendations are presented to them. Going forward, we suggest the systems adapting visualization by demonstration approach to design methods that explain the reasoning behind recommendations. This is similar to the way that Netflix recommends movies. Netflix explains it recommends movies because they are trending, because they are recently added to the collection of movies, or because these movies are similar to a specific movie was watched by the user.

Recommendation Timing: When observing the participants using the visualization by demonstration technique, we noticed that they sometimes found incoming recommendations interrupting. For example, one of the participants mentioned that *"is there a way to tell the system to do not update the recommendations after each interaction? something like a button to stop the incoming ones [recommendations]."* In the current version of VisExamplar, the recommendations will be updated in the interface whenever the recommendation table in the recommendation engine gets updated. We suggest the systems which plan to make use of the visualization by demonstration technique consider investigating methods for minimizing the interruption caused by incoming recommendations. We can envision two strategies to overcome the timing problem. First, systems present recommendations upon pressing a specific button on the interface. Second, systems could observe the cadence of user interaction with the system and make recommendations at a less active time.

Trade-offs and Open Challenges

In this part we discuss trades-off between the two interaction techniques and open challenges in designing interaction techniques for information visualization. These challenges are based on the collected data and our observations of the participants' interaction with both tools. In addition, we discuss some of the avenues for continued research on interaction in information visualization.

Interaction Complexity: Visualization by demonstration requires users to provide visual demonstrations of incremental changes to the visual representation. We noticed that the participants encountered two types of complexities while providing visual demonstrations. First, providing some demonstrations require time and accuracy. For example, to switch from a scatterplot to a barchart, participants need to stack a few data points vertically so that data points overlap. This takes participants time to demonstrate because they need to drag at least two data points and stack them. The participants also need to pay attention to make sure that the stacked data points are overlapping. Second, providing some demonstrations is cognitively complex. For example, mapping data attribute to axes requires participants to simultaneously keep track of a data point's value and its position relative to another data point.

We did not observe such complexity in manual view specification technique mainly because this technique enables rapid and formal/exact specification of visualization technique, mappings and parameter by incorporating a set of limited and consistent user interface elements. For example, in a system such as Polestar, majority of visualization specification can be completed using simple drag and drop interaction (e.g., users can drag a data attribute to shelves to assign data attribute to axes or visual encodings).

We noticed that participants are not interested in encountering either type of complexities while providing demonstrations. Going forward, providing additional feedback might be helpful to improve the performance of specific demonstrations. For example, during dragging data points, interaction performance might be improved by also showing exact values via textual overlay. While we did not test the effectiveness of such potential design improvements, these considerations could be of interest for future design and evaluation efforts.

Control and Engagement: The level of interaction directness [3] in manual view specification is less than visualization by demonstration. Unlike manual view specification, visualization by demonstration advocates for direct interaction with the components used in a visualization. This level of directness is effective since it gives the participants the feeling of being in control of all the elements in the user interface. In fact, as we discussed earlier some of the participants even applauded this level of control in the visualization by demonstration. However, we observed that this level of directness sometimes lead the participants to the point that they forget their primary task. For example, one of the participants was so involved in the process of dragging the points that at some points he asked: *"I forgot what I was going to do."* This could potentially go against the goal of traditional visualization tools that maintain a functionalist perspective, in that they are designed to be helpful for a particular set of analytic tasks. Advantages and disadvantages in increasing the directness of interaction techniques then raise a question — What is the right level of interaction directness that should be given to the users of the visualization tools?

Results of our observations and interviews indicate that the high-level of interaction directness in visualization by demonstration resulted in a more engaging and involved visualization construction and data exploration experience. We noticed that some of the participants found it more engaging and fun to be able to directly manipulate the graphical encodings used in a visual representation. One participant mentioned: *"... interactions like sorting are fun and natural. Have you ever thought to test your tool on high school students? I think they will like it a lot because they can move things around and play with it while they are learning."* Another participant stated that *"I think putting [stacking] circles [data points] on the top of each other takes time but it is fun because it is like building blocks puzzle."* One interesting avenue of research is to investigate the effectiveness of these interaction techniques on visualization tools that are designed for different categories of users. For example, while user engagement and involvement might not be the primary goal of visualization tools that are designed to support a particular set of analytic tasks, it might be important for visualization tools that are designed for educational purposes or casual information visualization tools [41].

8 Limitations and Future Work

Our findings should be interpreted in the context of the specified visualization tools, datasets, and tasks. Due to practical limitations of conducting the study (e.g., length and complexity of the experiment), we did not test for all possible types of visualization tools, tasks, or datasets. We tested the two different visualization tools with their own idiosyncrasies and workflows. We chose these tools because each implements one of the interaction techniques, and their relative simplicity with regards to the remaining system components. Our goal was to try and isolate the difference between the two tools as the interaction technique implemented, and minimize the effect of other system design choices and features. While the affect of tool design and implementation has to be considered when extrapolating our findings, we believe that our observations can be linked to the interaction techniques incorporated in the tools rather than specifics of the tools tested in our study. Our participants also highlighted the advantages and disadvantages of underlying interaction techniques rather than a specific implementation of each tool. That being said, additional studies are required to test our research questions taking into account additional visualization tools, tasks and datasets.

9 Conclusions

We present a two-phase study comparing people's visualization process using two visualization tools: one promoting the manual view specification technique (Polestar) and another promoting the visualization by demonstration technique (VisExamplar). Findings of our study indicate that people often take two specific and abstract strategies to construct visualizations. We also found that each interaction technique is more effective for different categories of tasks. We learned that while in manual view specification participants often followed a linear visualization process, they took different paths while constructing visualization using visualization by demonstration. We discuss some of the trade-offs and open challenges in incorporating these interaction techniques and compile recommendations on how to incorporate these interaction techniques in visualization tools.

References

[1] C. Andrews, A. Endert, and C. North. Space to think: large high-resolution displays for sensemaking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 55–64. ACM, 2010.

[2] D. Baur, B. Lee, and S. Carpendale. Touchwave: kinetic multi-touch manipulation for hierarchical stacked graphs. In *Proceedings of the*

2012 ACM international conference on Interactive tabletops and surfaces, pages 255–264. ACM, 2012.

[3] M. Beaudouin-Lafon. Instrumental interaction: an interaction model for designing post-wimp user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 446–453. ACM, 2000.

[4] E. T. Brown, J. Liu, C. E. Brodley, and R. Chang. Dis-function: Learning distance functions interactively. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, pages 83–92. IEEE, 2012.

[5] A. Buja, J. A. McDonald, J. Michalak, and W. Stuetzle. Interactive data visualization using focusing and linking. In *Visualization, 1991. Visualization'91, Proceedings., IEEE Conference on*, pages 156–163. IEEE, 1991.

[6] S. K. Card and D. Nation. Degree-of-interest trees: A component of an attention-reactive user interface. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, pages 231–245. ACM, 2002.

[7] W. S. Cleveland. *Visualizing Data*. Hobart Press, 1993.

[8] W. S. Cleveland and R. McGill. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American statistical association*, 79(387):531–554, 1984.

[9] J. Cohen. Statistical power analysis. *Current directions in psychological science*, 1(3):98–101, 1992.

[10] C. Collins, G. Penn, and S. Carpendale. Bubble sets: Revealing set relations with isocontours over existing visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1009–1016, 2009.

[11] J. W. Creswell. *Educational research: Planning, conducting, and evaluating quantitative*. Prentice Hall Upper Saddle River, NJ, 2002.

[12] T. Datasets. https://public.tableau.com/s/resources, 2015.

[13] S. M. Drucker, D. Fisher, R. Sadana, J. Herron, et al. Touchviz: a case study comparing two interfaces for data analytics on tablets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2301–2310. ACM, 2013.

[14] N. Elmqvist, A. V. Moere, H.-C. Jetter, D. Cernea, H. Reiterer, and T. Jankun-Kelly. Fluid interaction for information visualization. *Information Visualization*, page 1473871611413180, 2011.

[15] A. Endert, P. Fiaux, and C. North. Semantic interaction for visual text analytics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 473–482. ACM, 2012.

[16] A. Endert, C. Han, D. Maiti, L. House, S. Leman, and C. North. Observation-level interaction with statistical models for visual analytics. In *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*, pages 121–130. IEEE, 2011.

[17] E. R. Gansner, Y. Hu, and S. Kobourov. Gmap: Visualizing graphs and clusters as maps. In *2010 IEEE Pacific Visualization Symposium (PacificVis)*, pages 201–208. IEEE, 2010.

[18] D. Gotz and Z. Wen. Behavior-driven visualization recommendation. In *Proceedings of the 14th international conference on Intelligent user interfaces*, pages 315–324. ACM, 2009.

[19] L. Grammel, M. Tory, and M.-A. Storey. How information visualization novices construct visualizations. *IEEE transactions on visualization and computer graphics*, 16(6):943–952, 2010.

[20] J. Heer and M. Agrawala. Multi-scale banking to 45 degrees. *IEEE Trans. Visualization & Comp. Graphics*, 12:701–708, 2006.

[21] J. Heer, N. Kong, and M. Agrawala. Sizing the horizon: The effects of chart size and layering on the graphical perception of time series visualizations. In *ACM Human Factors in Computing Systems (CHI)*, 2009.

[22] H. V. Henderson and P. F. Velleman. Building multiple regression models interactively. *Biometrics*, pages 391–411, 1981.

[23] S. Huron, S. Carpendale, A. Thudt, A. Tang, and M. Mauerer. Constructive visualization. In *Proceedings of the 2014 conference on Designing interactive systems*, pages 433–442. ACM, 2014.

[24] S. Huron, Y. Jansen, and S. Carpendale. Constructing visual representations: Investigating the use of tangible tokens. *Visualization and Computer Graphics, IEEE Transactions on*, 20(12):2102–2111, 2014.

[25] E. L. Hutchins, J. D. Hollan, and D. A. Norman. Direct manipulation interfaces. *Human–Computer Interaction*, 1(4):311–338, 1985.

[26] P. Isenberg, T. Isenberg, T. Hesselmann, B. Lee, U. Von Zadow, and A. Tang. Data visualization on interactive surfaces: A research agenda. *IEEE Computer Graphics and Applications*, 33(2):16–24, 2013.

[27] B. Johnson and B. Shneiderman. Tree-maps: A space-filling approach to the visualization of hierarchical information structures. In *Visualization, 1991. Visualization'91, Proceedings., IEEE Conference on*, pages 284–291. IEEE, 1991.

[28] H. Kim, J. Choo, H. Park, and A. Endert. Interaxis: Steering scatterplot axes. *IEEE Visual Analytics Science and Technology (VAST)*, 2015.

[29] N. W. Kim, E. Schweickart, Z. Liu, M. Dontcheva, W. Li, J. Popovic, and H. Pfister. Data-driven guides: Supporting expressive design for information graphics. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):491–500, 2017.

[30] B. Kondo and C. M. Collins. Dimpvis: Exploring time-varying information visualizations by direct manipulation. *Visualization and Computer Graphics, IEEE Transactions on*, 20(12):2003–2012, 2014.

[31] N. Kong, J. Heer, and M. Agrawala. Perceptual guidelines for creating rectangular treemaps. *IEEE Trans. Visualization & Comp. Graphics*, 16(6):990–998, 2010.

[32] B. C. Kwon, H. Kim, E. Wall, J. Choo, H. Park, and A. Endert. Axisketcher: Interactive nonlinear axis mapping of visualizations through user drawings. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):221–230, Jan 2017.

[33] H. Lam, T. Munzner, and R. Kincaid. Overview use in multiple visual information resolution interfaces. *IEEE Trans. Visualization & Comp. Graphics*, 13(6):1278–1285, 2007.

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

[35] B. Lee, G. Smith, N. H. Riche, A. Karlson, and S. Carpendale. Sketchinsight: Natural data exploration on interactive whiteboards leveraging pen and touch interaction. In *Visualization Symposium (PacificVis), 2015 IEEE Pacific*, pages 199–206. IEEE, 2015.

[36] S. Lewandowsky and I. Spence. Discriminating strata in scatterplots. *Journal of American Statistical Association*, 84(407):682–688, 1989.

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

[38] C. Perin. *Direct Manipulation for Information Visualization*. Theses, Université Paris Sud - Paris XI, Nov. 2014.

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

[40] PoleStar. Polestart ,http://vega.github.io/polestar/, 2016.

[41] Z. Pousman, J. Stasko, and M. Mateas. Casual information visualization: Depictions of data in everyday life. *IEEE transactions on visualization and computer graphics*, 13(6), 2007.

[42] D. Ren, T. Höllerer, and X. Yuan. iVisDesigner: Expressive interactive design of information visualizations. *IEEE transactions on visualization and computer graphics*, 20(12):2092–2101, 2014.

[43] N. H. Riche, B. Lee, and C. Plaisant. Understanding interactive legends: a comparative evaluation with standard widgets. In *Computer graphics forum*, volume 29, pages 1193–1202. Wiley Online Library, 2010.

[44] J. M. Rzeszotarski and A. Kittur. Kinetica: Naturalistic multi-touch data visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 897–906. ACM, 2014.

[45] R. Sadana and J. Stasko. Designing multiple coordinated visualizations for tablets. In *Computer Graphics Forum*, volume 35, pages 261–270. Wiley Online Library, 2016.

[46] B. Saket, H. Kim, E. T. Brown, and A. Endert. Visualization by demonstration: An interaction paradigm for visual data exploration. *IEEE Transactions on Visualization and Computer Graphics*, PP(99):1–1, 2016.

[47] B. Saket, A. Srinivasan, E. D. Ragan, and A. Endert. Evaluating interactive graphical encodings for data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 2017.

[48] A. Satyanarayan and J. Heer. Lyra: An interactive visualization design environment. *Computer Graphics Forum (Proc. EuroVis)*, 2014.

[49] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer. Vega-lite: A grammar of interactive graphics. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2017.

[50] B. Shneiderman. 1.1 direct manipulation: astep beyond programming languages. *Sparks of innovation in human-computer interaction*, page 17, 1993.

[51] B. Shneiderman. Dynamic queries for visual information seeking. *Software, IEEE*, 11(6):70–77, 1994.

[52] D. Simkin and R. Hastie. An information-processing analysis of graph perception. *Journal of the American Statistical Association*, 82(398):454–465, 1987.

[53] SpotFire. http://www.spotfire.com, 2016.

[54] J. Stasko and E. Zhang. Focus+ context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. In *Information Visualization, 2000. InfoVis 2000. IEEE Symposium on*, pages 57–65. IEEE, 2000.

[55] C. Stolte and P. Hanrahan. Polaris: A system for query, analysis and visualization of multi-dimensional relational databases. In *INFOVIS*, pages 5–14, 2000.

[56] Tableau. Tableau software, http://www.tableau.com/, 2016.

[57] J. Talbot, J. Gerth, and P. Hanrahan. Arc length-based aspect ratio selection. *IEEE Trans. Visualization & Comp. Graphics*, 2011.

[58] J. Talbot, S. Lin, and P. Hanrahan. An extension of Wilkinson's algorithm for positioning tick labels on axes. *IEEE Trans. Visualization & Comp. Graphics*, 2010.

[59] F. B. Viégas and M. Wattenberg. Timelines tag clouds and the case for vernacular visualization. *interactions*, 15(4):49–52, 2008.

[60] H. Wickham. *ggplot2: elegant graphics for data analysis*. Springer, 2016.

[61] W. Willett, Q. Lan, and P. Isenberg. Eliciting multi-touch selection gestures for interactive data graphics. In *Short-Paper Proceedings of the European Conference on Visualization (EuroVis)*. Eurographics, 2014.

[62] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2015.

[63] K. Wongsuphasawat, Z. Qu, D. Moritz, R. Chang, F. Ouk, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager 2: Augmenting visual analysis with partial view specifications. In *ACM Human Factors in Computing Systems (CHI)*, 2017.

[64] J. S. Yi, Y. ah Kang, J. T. Stasko, and J. 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.

Task Based Effectiveness of Basic Visualizations

Bahador Saket, Alex Endert, Çağatay Demiralp

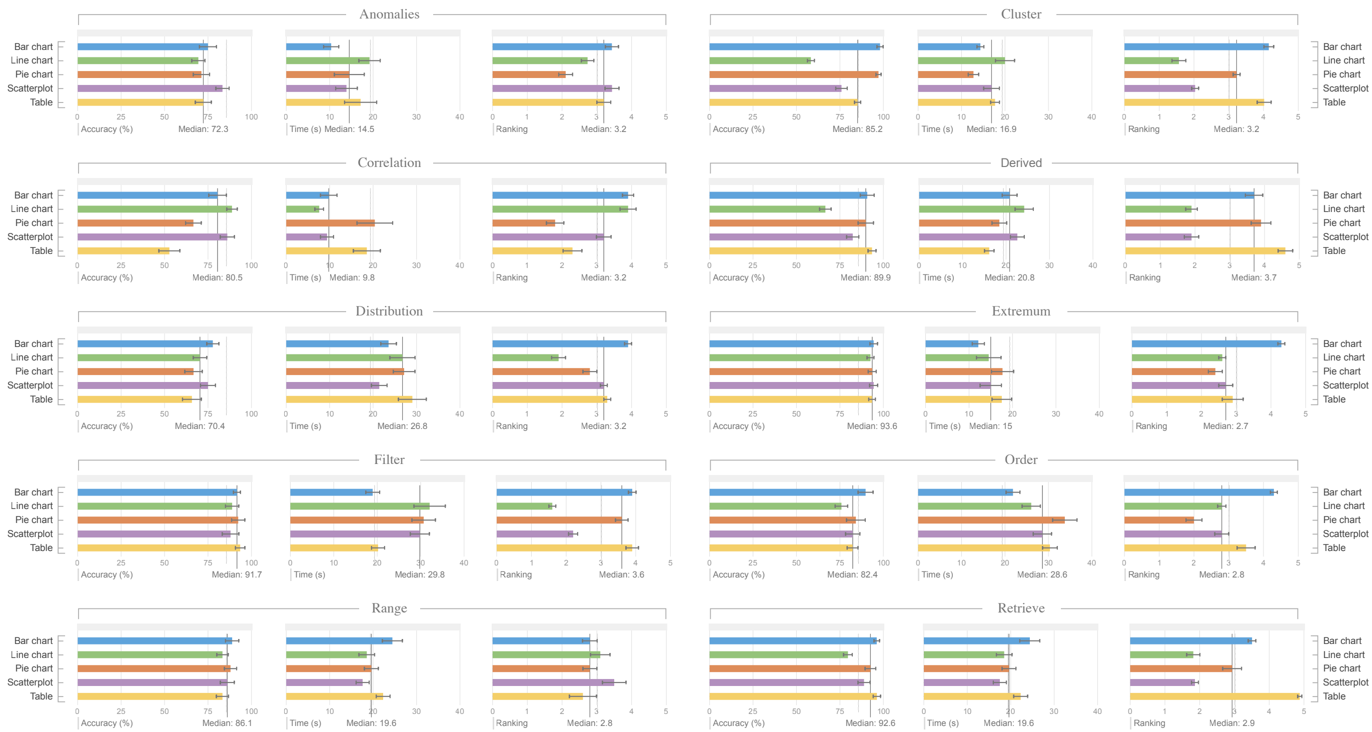


Figure 1: This figure shows performance results for different tasks. Performance results for each task are shown using three sub-charts. Mean accuracy results are shown on the left (mean accuracy is measured in percentage), mean time results are shown in the middle, and user preferences/rankings are shown at the right (1 shows least preferred and 5 shows the most preferred). Medians for each task is indicated using the vertical line. Dashed lines show the median across all tasks. Error bars represent standard error.

Abstract—Visualizations of tabular data are widely used; understanding their effectiveness in different task and data contexts is fundamental to scaling their impact. However, little is known about how basic tabular data visualizations perform across varying data analysis tasks. In this paper, we report results from a crowdsourced experiment to evaluate the effectiveness of five visualization types—Table, Line Chart, Bar Chart, Scatterplot, and Pie Chart—across ten common data analysis tasks using two real world datasets. We found the effectiveness of these visualization types significantly varies across task, suggesting that visualization design would benefit from considering context-dependent effectiveness. Based on our findings, we derive recommendations on which visualizations to choose based on different task.

1 Introduction

The demand for data visualization has significantly grown in recent years with the increasing availability of digitized data across everyday domains [44]. Visualizations aim to enhance understanding of underlying data by leveraging human visual perception, evolved for fast pattern detection and recognition. Understanding the effectiveness of a given visualization in achieving this goal is a fundamental pursuit in data visualization research and has important implications in practice.

- *Bahador Saket and Alex Endert are with Georgia Institute of Technology. E-mail: {saket, endert@gatech.edu}.*
- *Çağatay Demiralp is with IBM Research. E-mail: {cagatay.demiralp@us.ibm.com}*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

A large body of research has studied how various properties of data visualizations, from low-level visual encoding choices to Gestalt characteristics, impact the effectiveness of visualizations (e.g., [10, 12, 14, 28, 31, 36, 40, 43, 56, 58, 59, 63, 64, 66]). Previous work also has evaluated visualization types for their effectiveness [15, 16, 20, 24, 27, 35, 60]. Guidelines and insights derived from these earlier studies have significant influence on data visualization today. However, these studies were conducted under conditions that were inconsistent across studies, with varying sample sizes, a limited number of tasks, and using different datasets. Research indicates, however, the effectiveness of a visualization depends on several factors including task/question at the hand [1], and data attributes and datasets visualized [8, 53]. For example, while one chart might be suitable for answering a specific type of question (e.g., to check whether there is a correlation between two data attributes), it might not be appropriate for other types (e.g., to find a data point with the highest value). Yet, we know little about how some of the basic data visualizations perform across different visual analysis tasks.

In this paper we conduct a crowdsourced study to evaluate the effectiveness of five basic visualization types (Table, Line Chart, Bar Chart, Scatterplot, and Pie Chart) across 10 different visual analysis tasks [1] and two different datasets (Cars and Movies). Our results indicate that the effectiveness of these visualization types often significantly varies across tasks. For example, while pie charts are one of the most effective visualizations for finding the extremum value, they are less effective for finding correlation between two data attributes. We also asked participants to rank five different visualization types in the order of their preference for performing each task. We found a positive correlation between accuracy and user preference, indicating people have a preference for visualizations that allow them to accurately complete a task.

There is a renewed interest [5, 67, 68] in visualization recommendation systems that aim to shift some of the burden of visualization design and exploration decisions from users to algorithms. We discuss how empirical perception in general and our results in particular can be applied in improving visualization recommendation systems moving forward. Based on our analysis of the user task performances and visualization preferences, we also provide several recommendations on which visualization types to use under different tasks.

2 RELATED WORK

We draw from earlier work studying the impact of visual encoding decisions on decoding of the data presented in visualizations and the general effectiveness of visualizations for different design choices and task purposes. Below, we discuss some of the most relevant studies.

2.1 Graphical Perception

Data representation is a main component of information visualizations. The fundamental focus of data representation is mapping from data values to graphical representations [12, 14]. Visualization designers use elementary graphical units called visual encodings to map data to graphical representation [12]. Consider a case in which we visualize two numerical values using two bars with different lengths. Here, length is the primary encoding variable used to map the data values.

A good deal of prior work has studied how different choices of visual encodings influence visualization effectiveness. Bertin recognized that different visual variables have different effectiveness levels (or capacities) for encoding types of data [3]. Through human-subject experiments, researchers have investigated the effects of visual encoding on the ability to read and make judgments about data represented in visualizations [13, 29, 36, 43, 52, 56, 58, 59, 66]. For example, Skau and Kosara [58] recently studied the effectiveness of pie and donut charts in which data is encoded in three ways: arc length, center angle, and segment area. Their results showed that angle was the least important visual cue for both pie and donut charts. Consequently, prior research has provided rankings of visual variables by user performance for nominal, ordinal or numerical data [13, 43, 45, 46, 55]. Researchers have also investigated how design parameters beyond visual encoding variables such as aspect ratio [10, 28, 63], size [11, 31, 40], chart variation [37, 65], and axis labeling [64] impact the effectiveness of visualizations.

2.2 Effectiveness of Visualizations

Although prior research has proposed models of visualization comprehension [25, 38, 49, 57], little is known about how visual encoding or design parameters interact with each other or different data and task contexts in forming the overall performance of a given visualization. Earlier work [15–17, 20, 24, 27, 35, 58, 60] has also studied the effectiveness of visualization types with their common design configurations for a select number of tasks.

Eells [20] investigated effectiveness of proportional comparison (percentage estimation) task in divided (stacked) bar charts and pie charts. Eells asked participants to estimate the proportions in pie charts and bar charts. He found pie charts to be as fast as and more accurate than bar charts for proportional comparison task. He also found that as the number of components increases, divided bar charts become less accurate but pie charts become more (maximum five components were considered). In a follow up study with a different setting, Croxton and Stryker [16] also tested the effectiveness of divided bar charts and pie charts using a proportional comparison task. They also found pie charts to be more accurate than divided bar charts in most cases, but contrary to Eells' study, not all.

Study	Visualizations	Task
Eells [20]	Bar chart, Pie chart	Proportional comparison
Croxton et al. [16]	Bar chart, Pie chart	Proportional comparison
Spence et al. [60]	Table, Bar chart, Pie chart	Proportional comparison
Zacks et al. [69]	Bar chart, Line chart	Comparing Values Finding Trends
Harrison et al. [27]	Scatterplot, Radar, donut, stacked charts (bar, line, and area), parallel coordinates	Finding Correlation

Table 1: Settings of most related studies that previously measured **the effectiveness of visualizations** for different tasks.

Spence et al. [60] studied the effectiveness of bar charts, tables and pie charts. They found that when participants were asked to compare combinations of proportions, the pie charts outperformed bar charts. Their results also show that for tasks where participants were asked to retrieve the exact value of proportions, tables outperform pie charts and bar charts. In another study comparing the effectiveness of bar charts and line charts, Zacks and Tversky [69] indicated that when participants were shown these two types of visualizations and asked to describe the data, they constantly used bar charts to reference the compare values (e.g., A is 10% greater than B). Whereas with line charts, participants described trends.

Harrison et al. [27] measured the effectiveness of different visualizations for explaining correlation, finding that parallel coordinates and scatterplots are best at showing correlation. They also found that stacked bar charts outperform stacked area and stacked line. In a follow up study, Kay and Heer reanalyzed [34] the data collected by Harrison et al. [27]. The top ranking visualization remained the same.

While these independent studies provide helpful generic guidelines, they were conducted under different conditions, varying sample sizes, datasets, and for a disperse set of tasks. In fact, several of these studies used manually created visualizations in their experiments without using actual datasets [16, 20, 60, 69] or created visualizations using artificial datasets [27]. Also, these earlier studies have conducted experiments typically using atomic generic tasks such as comparison of data values (e.g., [16, 69]) or estimation of proportions (e.g., [20, 58, 60]), see Table 1. However, many higher level visual analysis tasks (e.g., filtering, finding clusters) require integration of results from multiple atomic tasks, limiting the applicability of earlier findings [1, 2]. Inconsistency in experimental settings and limited atomic tasks used in previous work encourages studying the effectiveness of visualization types for larger spectrum of data analysis tasks in a more consistence setting.

2.3 Importance of Tasks and Datasets

While performing analytical activities, users usually have a set of tasks/goals to perform [1, 6]. These tasks range from broader, "high-level" goals to more specific, "low-level" inquiries. Several studies have investigated spectrum of possible tasks that can be performed using different visualizations and summarized them as task taxonomies [1, 7, 41, 54]. Earlier research has also applied these taxonomies to evaluate different visualizations [19, 39, 41]. Previous work evidence that the effectiveness of visualizations might change with different task types [19, 33, 51]. Prior research also advocates the use of different datasets and data attribute types in evaluating the effectiveness of visualizations [8, 53].

To the best of our knowledge, the present work is the first systematic study to evaluate the effectiveness of the most common visualization types across a spectrum of tasks relevant to visual analysis using different datasets and data types.

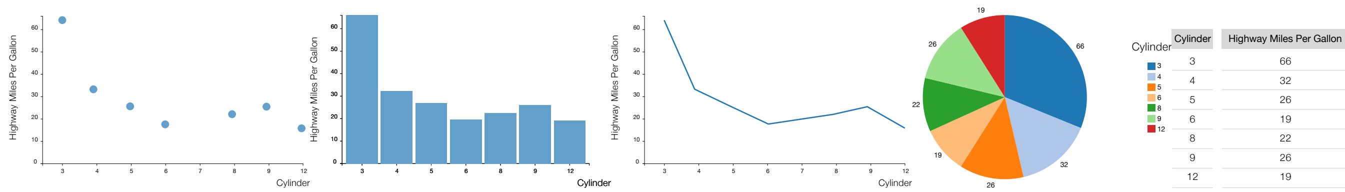


Figure 2: Five different types of visualizations used in this study. In this case, each visualization shows average highway miles per gallon (a numerical data attribute) for cars with different number of cylinders (an ordinal data attribute).

3 Research Goals and Experiment

Our goal is to better inform visualization design by understanding and quantifying how the effectiveness of tabular data visualizations changes with typical data analysis tasks. In particular, we seek to answer the following questions:

- **Q1:** How well are different tasks supported by each visualization? For example, what tasks can be performed well using a scatterplot?
- **Q2:** Is there a correlation between user preferences and their performance time and accuracy when using different visualizations for each task?

To address these questions, we conducted a crowdsourced experiment to measure accuracy, time, and user preferences in performing 10 visual analysis tasks using five visualization types. We discuss the details of the experimental design along with the rationale behind our choices regarding the design variables in the following sections.

3.1 Visualization Types used in Our Study

When deciding which visualization types to include in our experiment, we balanced the familiarity of the visualizations considered with the comprehensiveness of the experiment. On the one hand, we would like to have more generalizable results, which suggested considering a broad set of visualization techniques in our experiment. At the same time, we would like our study to have the members of general public as our participants: this would suggest to include a set of visualization techniques which are understandable by all participants.

Previous work studied which visualization types non-experts are familiar with. Borner et al. [4] assessed understanding of the members of general public with different visualization techniques. They showed 20 different visualization techniques to 273 participants and asked participants to name and explain how to read different techniques. Their results indicate that most of the participants had a difficult time naming and interpreting many of the techniques except those that they have seen often in books, at work, on the Internet, and in the news (e.g., bar charts, pie charts, line charts, tables, geographical maps, and scatterplots). Lee et al. [42] also show that bar charts, line charts, scatterplots, and pie charts are frequently shown up in news outlet and visualization tools.

We extracted a list of visualization techniques supported by some of the well-known visualization tools (e.g., Microsoft Excel, Tableau, Spotfire, QlickView, Adobe Analytics, IBM Watson Analytics). While all these tools support a variety of techniques, few of techniques are supported by all of them. Building on previous work [4] and investigations on visualization techniques supported by different visualization tools, we decided to include five well-recognized visualization techniques in our study. In this study, we include Bar Chart, Line Chart, Scatterplot, Table, and Pie Chart (see Figure 2).

3.2 Datasets used in Our Study

To create visualizations for our experiment, we selected datasets that the participants were unfamiliar with the content of the datasets but familiar with the meaning of the data attributes used in the dataset. This is particularly important since we did not want user performance to be affected by unfamiliarity of participants with the meaning of the data attributes.

We first selected five different datasets including Cereals [50], Cars [32], Movies [18], Summer Olympics Medalists [18], and University Professors [50]. We then printed a part of each dataset on paper and showed them to six participants (4 male, 2 female). We asked participants *"Please look at data attributes used in each of these datasets. Which datasets do you feel contain data attributes that you are more familiar with?"* Cars and Movies datasets were the ones that the majority (five out of six participants) of participants selected. The Cars dataset [32] provides details for 407 new cars and trucks for the year 2004. This dataset contains 18 data attributes describing each car. The Movies dataset [18] provides details for 335 movies released from 2007 to 2012, and contains 13 data attributes.

Both datasets include data attributes of Nominal, Ordinal, and Numerical types. We define Nominal data attribute type as categorically discrete data such as types of cars (e.g., Sedan, SUV, Wagon). Ordinal is defined as quantities within a specific range that have a natural ordering such as rating of movies (the number of unique data values ranged from 6 to 12). We define Numerical as continuous numerical data such as Profit values of movies.

3.3 Tasks

We selected the tasks for our study based on three considerations. First, tasks should be drawn from those commonly encountered while analyzing tabular data. Second, the tasks should be present in existing task taxonomies and often used in other studies to evaluate visualizations. Third, the tasks should be performed in a reasonable amount of time.

Previously, Amar et al. [1] proposed a set of ten low-level analysis tasks that describe users' activities while using visualization tools to understand their data. First, these tasks are real world tasks because users came up with them while exploring five different datasets with different visualization tools. Second, different studies used these tasks to evaluate effectiveness of visualizations. Finally, these task are can be performed in a reasonable amount of time. With this in mind, we used the low-level taxonomy by Amar et al. [1], described below.

Find Anomalies. We asked participants to identify any anomalies within a given set of data points with respect to a given relationship or expectation. We crafted these anomalies manually so that, once noticed, it would be straightforward to verify that the observed value was inconsistent with what would normally be present in the data (e.g., movies with zero or negative length would be considered abnormal). For example, *which genre of movies appear to have abnormal length?*

Find Clusters. For a given a set of data points, we asked participants to find clusters of similar data attribute values. For example, *how many different genres are shown in the chart below?*

Find Correlation. For a given set of two data attributes, we asked participants to determine if there is a correlation between them. To verify the responses to correlate tasks, we computed Pearson's correlation coefficient (r) to ensure that there was an strong correlation ($r \leq -0.7$ or $r \geq 0.7$) between the two data attributes. For example, *is there a strong correlation between average budget and movie rating?*

Compute Derived Value. For a given set of data points, we asked participants to compute an aggregate value of those data points. For example, *what is the sum of the budget for the action and the sci-fi movies?*

Characterize Distribution. For a given set of data points and an attribute of interest, we asked participants to identify the distribution of that attribute values over the set. For example, *what percentage of the movie genres have a average gross value higher than 10 million?*

Find Extremum. For this task, we asked participants to find data points having an extreme value of an data attribute. For example, *what is the car with highest cylinders?*

Filter. For given concrete conditions on data attribute values, we asked participants to find data points satisfying those conditions. For example, *which car types have city miles per gallon ranging from 25 to 56?*

Order. For a given set of data points, we asked participants to rank them according to a specific ordinal metric. For example, *which of the following options contains the correct sequence of movie genres, if you were to put them in order from largest average gross value to lowest?*

Determine Range. For a given set of data points and an attribute of interest, we asked participants to find the span of values within the set. For example, *what is the range of car prices?*

Retrieve Value. For this task, we asked participants to identify values of attributes for given data points. For example, *what is the value of horsepower for the cars?*

3.4 Visualization Design

To create Scatterplots, Bar Charts, and Line Charts, we used the same length, font size, and color to draw their $x-y$ axes. In addition, all the visual elements (e.g., bars in a bar chart) used in the three charts had the same blue color (see Figure 2-a, b, c).

Unlike other visualizations, pie charts do not have any axis to read the values from. That is, to create Pie Charts we had to make design decisions on how to show values of two data attributes used to generate them. The main design decision that we had to make for Pie Charts was whether to include legends. Instead of having legends, we could potentially add labels on the top of slices of Pie Charts. We tried to put the labels on the top of slices but this caused visual clutter, particularly in cases where the labels were long. Additionally, using legends for Pie Charts is a common practice in majority of commercial visualization dashboards [61, 62]. We decided to not show any value on the top of the slices of Pie Charts, instead showing the values of one data attribute using a legend and another one beside the slices (see Figure 2-d).

For Tables, we separated different rows of the table using light gray lines. We used a darker background color to make the labels (two data attributes used for creating the table) distinguishable (see Figure 2-e).

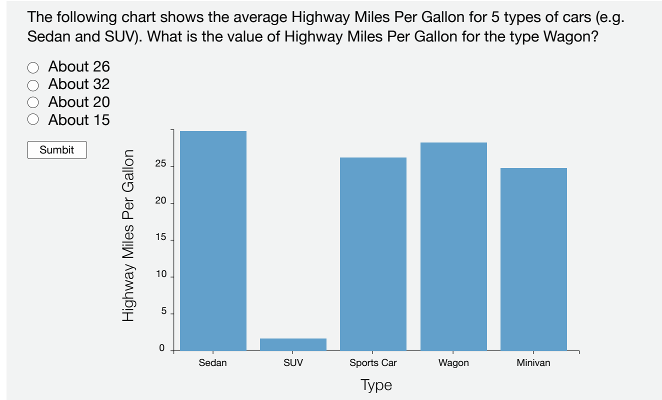
4 User Experiment

In this section, we explain the details of the experiment. We make all the relevant materials for our analysis publicly available[1], including the web application used to run the experiment along with its source code, the anonymized data collected from the participants, and the statistical test results.

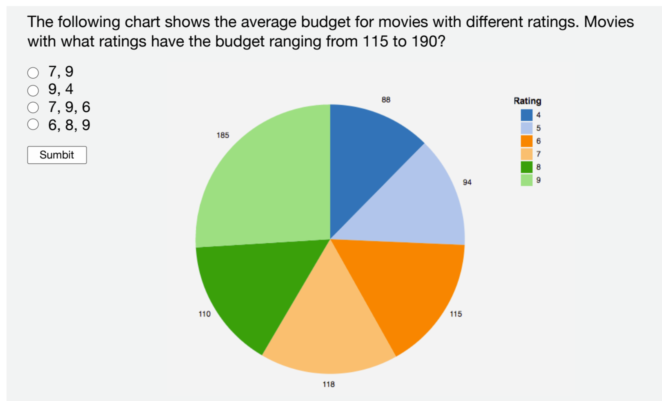
4.1 Experimental Platform & Participants

We conducted our experiment by posting it as a job, Human Intelligence Task (HIT), on Amazon's Mechanical Turk (MTurk). Earlier work demonstrates the viability of graphical perception experiments run on MTurk by reproducing in-lab study results [30]. To be able to participate in our study, MTurk workers (who perform tasks posted on MTurk), had to have an approval rate of 95% and at least 100 approved HITs as a quality check. We implemented our experiment as a web application hosted on a server external to MTurk. Participants accessed the experiment through a URL link posted on the MTurk site. Each worker could participate in our study only once. The study took

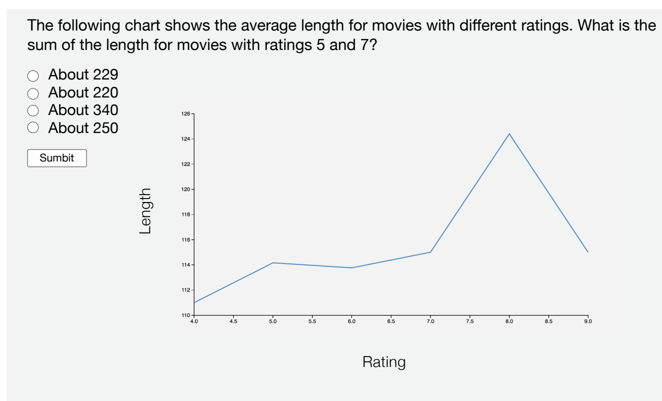
[1]https://github.com/gtvalab/ChartsEffectiveness



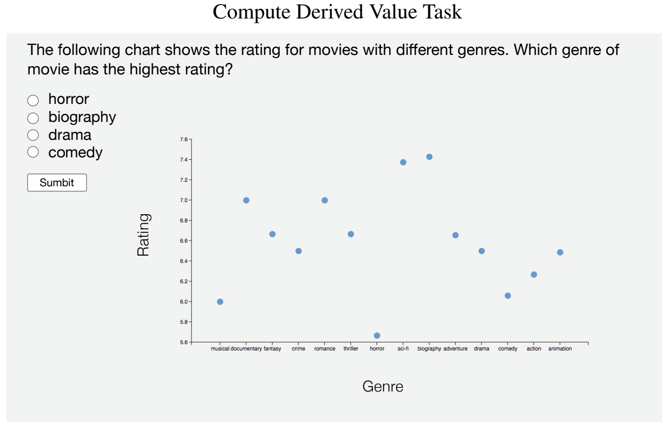
Retrieve Value Task



Determine Range Task



Compute Derived Value Task



Find Extremum

Figure 3: Screenshots of four of the trials used in this experiment. Each of the trials asks users to perform a specific task.

about 40 minutes to complete and we compensated the workers who participated $4.

In order to determine the minimum number of participants needed for our study, we first conducted a pilot study with 50 participants on Amazon's Mechanical Turk. Based on the data collected from our pilot study, we conducted a statistical power analysis to ensure that our experiment included enough participants to reliably detect meaningful performance differences across independent variables of the experiment. Our power analysis based on the results of the pilot study indicated that at least 160 participants would be required to detect a large effect.

After determining the number of subjects required to participate in our study, we recruited 203 workers to participate in our study. Among the 203 who participated in our study 180 of them (105 Male, 75 Female) completed the study. The age of our workers ranged from 25–40 years. All workers participated in our experiment were based in the United States and have used visualizations before. 107 of the participants had experience creating visualizations using Microsoft Excel. Five of the participants also had experience in creating visualizations using Tableau software.

4.2 Procedure

Training. Before starting the main experiment, participants were briefed about the purpose of the study and their rights. At this stage, the participants were also asked to answer to some demographic questions (e.g., age, sex, and prior experience in creating visualizations). Participants were then asked to perform 5 trial questions as quickly and accurately as possible. During this session, after answering each question participants received feedback that showed the correctness of their answers. To prevent the participants from skipping the training questions, participants were not able to move to the next training question unless they answered the question correctly.

Main Experiment. During the main experiment 180 participants were randomly assigned to 10 tasks (18 participants per task). So, each participant performed questions designed for one type of task. For each type of task, we had 30 questions (5 $Visualizations \times$ 2 $Datasets \times$ 3 $Trials$). As recommended by previous work [47], we also designed two additional questions to detect if a participant answered the questions randomly. These two questions were straightforward and designed to make sure that participant read the questions. Questions were presented in a random order to prevent participants from extrapolating new judgments from previous ones. Screenshots of the questions for the main experiment are shown in Figure 3.

Follow Up Questions. After completing the main experiment, the participants were asked to perform 6 additional ranking questions (3 $Trials \times$ 2 $Datasets$). In each ranking question the participants were asked to rank the five different visualizations in the order of their preference for performing this task. A screenshot of the ranking question is shown in Figure 4. Before finishing the experiment, we asked participants to *"Please enter the criteria you used for ranking the charts along with any other additional comments you have about the experiment in general"*. This was to allow the participants to convey their feedback and in order to solicit potentially unexpected insights.

4.3 Data Analysis

To address our research questions (**Q1, Q2**), we needed to test how the different visualization techniques affected the user performance time, accuracy, and preferences for different tasks.

To analyze the differences among the various visualizations, we first calculated separate mean performance values for all questions. That is, we averaged outcome values of questions for each visualization and task. Before testing, we checked that the collected data met the assumptions of appropriate statistical tests. The assumption of normality was not satisfied for performance time. However, the normality was satisfied for log transformed of time values. So, we treated log-transformed values as our time measurements.

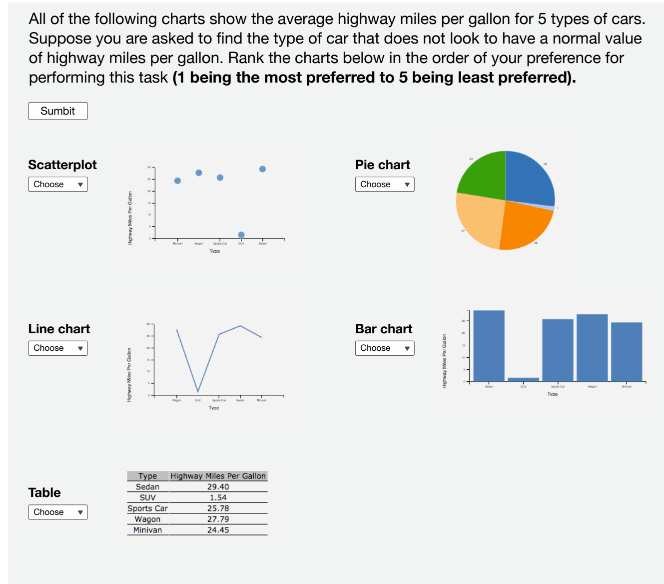


Figure 4: A screenshot of the ranking question. Participants could not choose the same ranking for multiple visualizations.

We conducted repeated-measures analysis of variance (ANOVA) for each task independently to test for differences among the various visualizations, datasets, and their interactions with one another. While Visualization had significant effects on both accuracy and time, Dataset had no significant effect on accuracy or time.

5 Summary of Results

We first give an overview of our analysis of the results and then discuss them in detail for each task. Throughout the following sections, accuracy refers to values in percentages (%) and time refers to values in seconds.

Results, aggregated over tasks and datasets, show that Bar Chart is the fastest and the most accurate visualization type. This result is inline with prior work on graphical perception showing that people can decode values encoded with length faster than other encodings such as angle or volume [12,57,65]. Conversely, Line Chart has the lowest aggregate accuracy and speed. However, Line Chart is significantly more accurate than other charts for Correlation and Distribution tasks. This finding concurs with with earlier research reporting the effectiveness of line charts for trend finding tasks (e.g., [69]). Nonetheless, the overall low performance of Line Chart is surprising and, for some tasks, can be attributed to the fact that the axes values ("ticks") were drawn at intervals. This makes it difficult to precisely identify the value for a specific data point.

While Pie Chart is comparably as accurate and fast as Bar Chart and Table for Retrieve, Range, Order, Filter, Extremum, Derived and Cluster tasks, it is less accurate for Correlation, Anomalies and Distribution tasks. Pie Chart is the fastest visualization for performing Cluster task. High performance of Pie Chart for these tasks can be attributed to its relative effectiveness in conveying part-whole relations and facilitating proportional judgments, particularly when the number of data points visualized is small [20,60]. Pie Chart may have been further helped by having colored slices with text labels showing the data values.

Overall, Scatterplot performs reasonably well in terms of both accuracy and time. For the majority of tasks Scatterplot is among the most effective top three visualizations, and it was never the least accurate or slowest visualization for any of the tasks. One reason for this could be that people are very accurate and fast in perceiving "position" [12].

Bar Chart and Table are the two visualization types highly preferred by participants across most of the tasks. Bar Chart is always among two top-performing visualizations for almost all tasks, so this makes sense that people prefer using Bar Chart over other visualizations.

Surprisingly, while performing some of the tasks (e.g., Distribution, Anomalies) using Table is relatively slow and less accurate, participants still prefer Table for performing these tasks. Familiarity of people with tables and easiness in understanding tables could have helped people to prefer using tables over other visualizations. To determine whether performance time and accuracy are related to user preferences, we calculated the correlation between performance time, accuracy, and user preference. We found a positive correlation between accuracy and user preference (Pearson's $r_{(5)} = 0.68, p < 0.05$), indicating people have a preference for visualizations that allow them to accurately complete a task. We also found a weak negative correlation between performance time and user preferences (Pearson's $r_{(5)} = -0.43, p < 0.05$).

6 RESULTS

We provide detailed analysis of the results, breaking them down by performance time (see Figure 5), accuracy (see Figure 6), and user preferences (see Figure 7) for all tasks. Each figure shows the effectiveness of different visualizations for all tasks.

7 DISCUSSION

In this section, we reflect on the results of our work more broadly with respect to information visualization. We discuss several aspects of our work, implications of the study findings, and provide a series of guidelines on choosing visualizations based on different tasks.

7.1 No One Size Fits All

Depending on the task at the hand, various visualizations perform differently (**Q1**). That is, we do not advocate generalizing the performance of a specific visualization on a particular task to every task. For example, throughout the history of the graphical perception research, pie charts have been a subject of passionate arguments [16, 20, 60] for and against their use. Although the current common wisdom among visualization researchers is to avoid them, pie charts continue to be popular in everyday visualizations. Results of our study present a more nuanced view of pie charts. We found that pie charts can be as effective as other visualizations for task types such as Cluster, Extremum, Filter, Retrieve, and Range. On the other hand, our results suggest that pie charts perform poorly in Correlation and Distribution tasks.

7.2 User Preferences

How do user preferences relate to user performance (**Q2**)? Our results show user preferences correlate with user accuracy and speed in completing tasks. Before completing the study, we asked participants to explain the criteria they used for ranking the visualizations. Some of participants explicitly mentioned perceived accuracy of the charts as one of the factors that influenced their decision while ranking visualizations. For example, one of the participants stated: *"Just by how accurate I felt my own answer was, and how easy it was to derive the answer from the graphs."*

Neither accuracy nor speed appear to be the only criteria by which participants describe their individual rankings. Additionally, perceived accuracy does not always match with task accuracy. We noticed that for some task types such as Distribution and Cluster, preference for using tables and bar charts is significantly higher than other visualizations, even though these two visualizations are not the most effective ones for these type of tasks. Interestingly, some of the participants took into account their familiarity with visualizations as one of the factors for preferring some visualization over others. For example, one of the participants mentioned: *"I just went with the ones I felt were familiar to me."* Another participant also stated: *"I deal with bars and tables everyday. I know how to read them."*

We anticipate that users' familiarity with visualizations has positive correlation with their preference in using those visualizations, but this remains to be formally studied.

7.3 Which Visualization Type to Use?

Based on our findings in the study, we derive the following guidelines on choosing visualizations under various task contexts.

G1. For all tasks, bar charts and scatterplots are good defaults. Not only do these two visualizations have a high performance across all tasks, but they also have the lowest variation of performance across different data attributes. This indicates that bar charts are very robust across various tasks and data attributes.

G2. Avoid line charts for tasks that require readers to precisely identify the value of a specific data point. The low performance of line charts for some tasks such as Derived and Cluster might be attributed to the fact that the axes values (i.e., the "ticks") were drawn at uniform intervals. This makes it difficult to precisely identify the value of a specific data point.

G3. Use scatterplots and bar charts for finding anomalies. Results of our study indicate that scatterplots and bar charts have high accuracy, speed, and are highly preferred by users for this type of task.

G4. Use tables for retrieving values. Tables are fastest and the most accurate visualization types for retrieving values task. Also, user preference in using tables is significantly higher than other visualizations.

G5. Use pie charts and bar charts for finding clusters. Our results show that these two visualizations performed well for this type of task.

G6. Use line charts and scatterplots for finding correlations. We found that line charts and scatterplots have high performance for finding correlations.

G7. Use bar charts for finding extremums. Based on our results bar charts are the fastest, most accurate, and highly preferred visualization for finding extremum tasks. Previous work also suggests using bar charts for finding maximum/minimum values [22].

G8. Use bar charts for ordering tasks. Considering speed, accuracy, and user preference, bar charts have a higher performance compared to other visualizations for ordering tasks.

8 APPLICATIONS OF OUR FINDINGS

Our findings from the current study inform the ongoing design and development of Foresight [9] at IBM. Foresight is a visual analysis recommendation system for rapidly discovering visual insights from large high-dimensional datasets. Foresight starts the data exploration process by automatically recommending visualizations, and then gives the user increasing control over the exploration process as familiarity with the data increases.

An important avenue for continued research is creating a recommendation engine that suggest visualizations based on user-specified tasks. One relevant application area of such a recommendation engine can be natural language interfaces for data visualization [23]. In such interfaces people tend to specify tasks as a part of their queries (e.g., "Is there a correlation between price and width of cars in this dataset?"). The engine can be used to leverage this knowledge to suggest more effective visualizations for the given query. Another category of systems that might benefit from such an engine are behavior-driven recommendation systems [26]. Such systems often extract analytical tasks from user interactions. These extracted tasks can be used as input to the engine for these systems to suggest more effective visualizations. All these applications would also benefit from models trained on the results of the current study (or any other empirical performance data) that can predict the task performance of "unseen" visualizations.

9 LIMITATIONS AND FUTURE WORK

Our experimental results should be interpreted in the context of the specified visualizations tasks, and datasets. Due to practical limitations of conducting the study (e.g., length and complexity of the experiment), we did not test for all possible types of visualizations or tasks. While our findings should be interpreted in the context of the specified settings and conditions, we tested the most common visualization techniques

Find Anomalies *Visualizations* ($F_{(4,68)} = 0.48, p < 0.001, \eta_p^2 = 0.27$)

Visualizations ranking from fastest (left) to slowest (right)
Bar Chart → Scatterplot → Pie Chart → Table → Line Chart

Pairwise Comparisons (p values are corrected using Bonferroni correction)
Bar Chart **vs.** Line Chart, Table ($p < .05$)

Main Findings. We found a significant main effect of Visualization on time. Bar Chart was the fastest visualization for performing this type of tasks. Posthoc comparisons indicate that Bar Chart was significantly faster than Line Chart and Table. This might be because people can decode values encoded with length faster than other encodings such as angle or distance [12, 57, 65].

Find Correlation *Visualizations* ($F_{(1,479.7)} = 42.3, p < 0.001, \eta_p^2 = 0.7$)

Visualizations ranking from fastest (left) to slowest (right)
Line Chart → Scatterplot → Bar Chart → Table → Pie Chart

Pairwise Comparisons (p values are corrected using Bonferroni correction)
Line Chart **vs.** Pie Chart, Table ($p < .001$)
Scatterplot **vs.** Pie Chart, Table ($p < .01$)
Bar Chart **vs.** Table ($p < .05$)
Bar Chart **vs.** Pie Chart ($p < .01$)

Main Findings. There is a significant effect of Visualization on time. We found that Line Chart, Bar Chart and Scatterplot were significantly faster than Pie Chart and Table. In fact, our results validates the findings of the previous work that showed the effectiveness of Scatterplots and Line charts for Correlation tasks [27, 48].

Order *Visualizations* ($F_{(3.3,0.6)} = 9.3, p < 0.001, \eta_p^2 = 0.35$)

Visualizations ranking from fastest (left) to slowest (right)
Bar Chart → Pie Chart → Table → Scatterplot → Line Chart

Pairwise Comparisons (p values are corrected using Bonferroni correction)
Bar Chart **vs.** Pie Chart ($p < .01$)
Bar Chart **vs.** Table ($p < .001$)
Line Chart **vs.** Table ($p < .05$)
Scatterplot **vs.** Pie Chart ($p < .05$)

Main Findings. There is also a significant main effect of Visualization on Time. Bar Chart is significantly faster than Pie Chart and Table. Line Chart is also significantly faster than Table for Order tasks. We also found that Scatterplot is significantly faster than Pie Chart. We did not find a significant difference among Line Chart, Scatterplot, and Bar Chart in terms of time. High performance of Line Chart, Scatterplot, and Bar Chart could be the due to their usage of length and position as primary graphical encodings. Length and position are fastest encodings to perceive [12, 14].

Characterize Distribution *Visualizations* ($F_{(4,68)} = 5.6, p < 0.01, \eta_p^2 = 0.25$)

Visualizations ranking from fastest (left) to slowest (right)
Scatterplot → Bar Chart → Line Chart → Pie Chart → Table

Pairwise Comparisons (p values are corrected using Bonferroni correction)
Scatterplot **vs.** Table ($p < .01$)
Scatter **vs.** Pie Chart ($p < .05$)
Bar Chart **vs.** Pie Chart, Table ($p < .05$)

Main Findings. We found a significant effect of Visualization on time, and our results indicate that Scatterplot and Bar Chart are significantly faster than Pie Chart and Table for Distribution tasks. There is not a significant difference between Line Chart, Bar Chart and Scatterplot. Previous work also showed the fast speed of Scatterplot for correlation tasks [27, 34].

Filter *Visualizations* ($F_{3.6,61.5)} = 19.4, p < 0.001, \eta_p^2 = 0.53$)

Visualizations ranking from fastest (left) to slowest (right)
Bar Chart→ Table → Pie Chart → Scatterplot → Line Chart

Pairwise Comparisons (p values are corrected using Bonferroni correction)
Bar Chart **vs.** Pie Chart, Scatterplot, Line Chart ($p < .001$)
Table **vs.** Pie Chart, Scatterplot, Line Chart ($p < .001$)

Main Findings. Results indicate that Bar Chart and Table are significantly faster than other visualizations. We did not find a significant difference among Scatterplot, Line Chart, and Pie Chart for time.

Find Clusters *Visualizations* ($F_{(3.9,67.9)} = 6.9, p < 0.001, \eta_p^2 = 0.29$)

Visualizations ranking from fastest (left) to slowest (right)
Pie Chart → Bar Chart → Scatterplot → Table → Line Chart

Pairwise Comparisons (p values are corrected using Bonferroni correction)
Pie Chart **vs.** Line Chart ($p < .01$)
Pie Chart **vs.** Table ($p < .05$)

Main Findings. We found a significant effect of Visualization on time. Pie Chart was the fastest visualization for performing Cluster tasks. While Pie Chart was significantly faster than Table and line Chart, there was no significant difference between Pie Chart and the other two visualizations. We believe that uniquely coloring different slices of pie charts improved the performance of Pie Chart for thhis type of tasks.

Compute Derived Value *Visualizations* ($F_{(3.2,0.4)} = 9.6, p < 0.001, \eta_p^2 = 0.36$)

Visualizations ranking from fastest (left) to slowest (right)
Table → Pie Chart → Bar Chart → Scatterplot → Line Chart

Pairwise Comparisons (p values are corrected using Bonferroni correction)
Table **vs.** Bar Chart ($p < .01$)
Table **vs.** Line Chart ($p < .001$)
Table **vs.** Scatterplot ($p < .01$)

Main Findings. Table had the fastest speed for Derived tasks. There was, however, no significant difference in time among Table and Pie Chart. High effectiveness of Table might be because the exact values for each data point is shown in tables. So it might be the case that less cognitive work is required to aggregate the values when the exact values are shown.

Retrieve Value *Visualizations* ($F_{(3.0,52.1)} = 4.34, p < 0.05, \eta_p^2 = 0.26$)

Visualizations ranking from fastest (left) to slowest (right)
Table → Bar Chart → Pie Chart → Scatterplot → Line Chart

Pairwise Comparisons (p values are corrected using Bonferroni correction)
Table **vs.** Scatterplot ($p < .05$)
Table **vs.** Line Chart ($p < .01$)

Main Findings. We found a significant main effect of Visualization on time. Table is the fastest visualization for performing this type of tasks. Table, Bar Chart, and Pie Chart are the fastest visualizations for performing Retrieve tasks. Successful performance time of Retrieve tasks highly depends on readers ability to rapidly identify the value for a certain data point. As Ehrenberg [21] points out, tables are well-suited for retrieving the numerical value of a data point when a relatively small number of data points are displayed.

Find Extremum *Visualizations* ($F_{(4,0.4)} = 10.4, p < 0.001, \eta_p^2 = 0.38$)

Visualizations ranking from fastest (left) to slowest (right)
Bar Chart → Scatterplot → Line Chart → Pie Chart → Table

Pairwise Comparisons (p values are corrected using Bonferroni correction)
Bar Chart **vs.** Pie Chart, Table ($p < .01$)
Line Chart **vs.** Pie Chart, Table ($p < .05$)
Scatterplot **vs.** Pie Chart, Table ($p < .05$)

Main Findings. We found a significant effect of Visualization on time. In terms of time, Bar Chart is the fastest and Pie Chart is the slowest visualization for this type of tasks. In fact, Bar Chart, Line Chart, and Scatterplot are significantly faster than Table and Pie Chart. Previous work also recommends using Bar Chart in cases where readers are looking for a maximum or minimum values [22].

Determine Range

Visualizations ranking from fastest (left) to slowest (right)
Scatterplot → Line Chart → Pie Chart → Table → Bar Chart

Pairwise Comparisons (p values are corrected using Bonferroni correction)
NA

Main Findings. We did not found a significant main effect of Visualization on accuracy. In other words, there was not a significant difference between any two visualizations for Determine Range task.

Figure 5: Performance time results for different tasks along with statistical test results. For each task, we ranked visualizations from the fastest (left) to the slowest (right) based on the average completion time.

Find Anomalies *Visualizations* ($F_{(3.4,4915.1)} = 3.03, p < 0.05, \eta_p^2 = 0.15$)

Visualizations ranking from most accurate (left) to least (right)
Scatterplot → Bar Chart → Table → Pie Chart → Line Chart

Pairwise Comparisons (p values are corrected using Bonferroni correction)
Scatterplot **vs.** Line Chart ($p < .05$)

Main Findings. We found a significant effect of Visualization on accuracy. Overall, Line Chart, Table, and Pie Chart had lower accuracy compared to Bar Chart and Scatterplot. Results of Bonferroni-corrected post-hoc comparisons showed that Line Chart was significantly less accurate than Scatterplot.

Find Clusters *Visualizations* $F_{(2.6,45065.1)} = 60.7, p < 0.01, \eta_p^2 = 0.78$)

Visualizations ranking from most accurate (left) to least (right)
Bar Chart → Pie Chart → Table → Scatterplot → Line Chart

Pairwise Comparisons (p values are corrected using Bonferroni correction)
Bar Chart **vs.** Table, Scatterplot, Line Chart ($p < .01$)
Pie Chart **vs.** Table, Scatterplot, Line Chart ($p < .01$)

Main Findings. Results indicate that there was a significant effect of Visualization on accuracy. Results of Bonferroni-corrected posthoc comparisons show that Pie Chart and Bar Chart were significantly more accurate than other visualizations.

Find Correlation *Visualizations* ($F_{(2.5,20528.2)} = 12.1, p < 0.001, \eta_p^2 = 0.41$)

Visualizations ranking from most accurate (left) to least (right)
Line Chart → Scatterplot → Bar Chart → Pie Chart → Table

Pairwise Comparisons (p values are corrected using Bonferroni correction)
Line Chart **vs.** Bar Chart, Pie Chart, Table ($p < .01$)
Scatterplot **vs.** Bar Chart, Pie Chart, Table ($p < .01$)
Bar Chart **vs.** Pie Chart, Table ($p < .01$)

Main Findings. We found a significant main effect of Visualization on accuracy. Pairwise comparison show that Line Chart and Scatterplot were significantly more accurate than other charts. Bar Chart was also significantly more accurate than Pie Chart and Table.

Compute Derived Value *Visualizations* ($F_{(2.7,18234.2)} = 16.2, p < 0.001, \eta_p^2 = 0.49$)

Visualizations ranking from most accurate (left) to least (right)
Table → Bar Chart → Pie Chart → Scatterplot → Line Chart

Pairwise Comparisons (p values are corrected using Bonferroni correction)
Table, Bar Chart, Pie Chart, Scatterplot **vs.** Line Chart ($p < .01$)

Main Findings. We found a significant main effect of Visualization on accuracy. Table, Bar Chart, and Pie Chart had the highest accuracy for this type of task. Accuracy of Line Chart was significantly lower than rest of the four chart types. On the other hand, there was no significant difference among Bar Chart, Scatterplot, Pie Chart, and Table.

Order *Visualizations* ($F_{(4,.03)} = 2.6, p < 0.05, \eta_p^2 = 0.17$)

Visualizations ranking from most accurate (left) to least (right)
Bar Chart → Pie Chart → Table → Scatterplot → Line Chart

Pairwise Comparisons (p values are corrected using Bonferroni correction)
Bar Chart **vs.** Line Chart ($p < .05$)

Main Findings. We found a significant main effect of Visualization on accuracy. Bar Chart has the highest accuracy for Order tasks. In addition, Bar Chart is significantly more accurate than Line Chart. We did not find a significant difference among Bar Chart, Pie Chart, Scatterplot, and Table.

Retrieve Value *Visualizations* ($F_{(2.9,7114.1)} = 7.7, p < 0.001, \eta_p^2 = 0.32$)

Visualizations ranking from most accurate (left) to least (right)
Table → Bar Chart → Pie Chart → Scatterplot → Line Chart

Pairwise Comparisons (p values are corrected using Bonferroni correction)
Table, Bar Chart, Pie Chart **vs.** Line Chart ($p < .01$)

Main Findings. We found a significant main effect of Visualization on accuracy. Overall, Bar Chart, Table and Pie Chart were significantly more accurate than Line Chart. The difference between accuracy in Scatterplot and Line Chart was not significant.

Figure 6: Performance accuracy results for different tasks along with statistical test results. For each task, we ranked visualizations from the most accurate (left) to the least accurate (right) based on the average accuracy. We did not include the results for tasks that we could not find the main effect of Visualization on their accuracy. This tasks include characterized distribution, find extremum, filter, and determine range.

incorporated in various visualization dashboards [4, 42], analytical tasks used different studies [1,2] and real world datasets used in various studies [18, 50]. That being said, additional studies are required to test our research questions taking into account additional visualization techniques, tasks and datasets.

For Anomalies task, the variation of data attributes used to generate visualizations is not large. The main reason is because we hand crafted these anomalies so that it would be straightforward to verify that the observed value was inconsistent with what would normally be present in the data (e.g., movies with zero or negative length would be considered abnormal). However, verifying abnormal data points in some of these data attributes might not be that straightforward (e.g., it is possible to have negative profit or gross values in a data set). Thus, we created anomalies for only a subset of the attributes. We encourage future work to take into a larger set of data attributes for this type of task.

10 Conclusion

In this work, we report the results of a study that gathers user performance and preference for performing ten common data analysis tasks using five basic visualization types, Table, Line Chart, Bar Chart, Scatterplot, and Pie Chart. We use data sampled from two real world datasets to further support the ecological validity of results. We find that the effectiveness of the visualization types considered significantly changes from one task to another. We compile our findings into a set of recommendations to inform data visualization in practice.

Prior work on graphical perception has served visualization practitioners well by providing guidelines and heuristics on visual encoding and design choices. However, no measure of effectiveness is free from consideration of task and data. To further increase the impact of graphical perception research in visualization practice, we must tightly couple perceptual considerations into everyday visualization tools. This starts with systematically gathering empirical evidence on how visualizations perform across contexts and building models based on the evidence. Our work contributes towards the former.

To facilitate future research by the community, we have made all relevant materials including anonymized data collected from our participants, data analysis, and source code of our experiment publicly available at `https://github.com/gtvalab/ChartsEffectiveness`.

References

[1] 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. IEEE, 2005.

[2] R. Amar and J. Stasko. A knowledge task-based framework for design and evaluation of information visualizations. In *Procs. InfoVis*, pages 143–150, 2004.

[3] J. Bertin. *Semiology of graphics*. University of Wisconsin Press, 1983.

[4] K. Börner, A. Maltese, R. N. Balliet, and J. Heimlich. Investigating aspects of data visualization literacy using 20 information visualizations and 273 science museum visitors. *Information Visualization*, page 1473871615594652, 2015.

Find Anomalies *Visualizations* ($F_{(3.1,45.56)} = 5.9, p < 0.05, \eta_p^2 = 0.26$)

Visualizations ranking from most preferred (left) to least (right)
Bar Chart → Scatterplot → Table → Line Chart → Pie Chart

Pairwise Comparisons (p values are corrected using Bonferroni correction)
Bar Chart, Scatterplot **vs.** Pie Chart, Line Chart ($p < .05$)

Main Findings. We found a significant effect of Visualization on user preference. For the Anomalies task type, results of pairwise comparisons show that user preference in performing Anomalies tasks using Bar Chart and Scatterplot were significantly higher than Pie Chart and Line Chart.

Find Clusters *Visualizations* ($F_{(2.9,188.56)} = 30.2, p < 0.001, \eta_p^2 = 0.64$)

Visualizations ranking from most preferred (left) to least (right)
Bar Chart → Table → Pie Chart → Scatterplot → Line Chart

Pairwise Comparisons (p values are corrected using Bonferroni correction)
Bar Chart, Table **vs.** Pie Chart, Scatterplot, Line Chart ($p < .01$)

Main Findings. Results indicate a significant main effect of Visualization on user preferences. User preferences in using Bar Chart and Table were significantly higher than other visualizations. While user preferences in using Bar Chart can be explained by its high accuracy and speed, it is surprising that Table was also highly preferred by users for Cluster tasks. Table had the lowest performance time and its accuracy was significantly lower than Pie Chart and Bar Chart.

Find Correlation *Visualizations* ($F_{(3.6,75.2)} = 13.6, p < 0.001, \eta_p^2 = 0.44$)

Visualizations ranking from most preferred (left) to least (right)
Line Chart → Scatterplot → Bar Chart → Pie Chart → Table

Pairwise Comparisons (p values are corrected using Bonferroni correction)
Line Chart, Scatterplot **vs.** Pie Chart, Table ($p < .01$)
Bar Chart **vs.** Pie Chart, Table ($p < .05$)

Main Findings. We found a significant main effect of Visualization on user preference. User preference in performing Correlations tasks using Bar Chart, Line Chart and Scatterplot were significantly higher than that of Pie Chart and Table. Positive correlation between accuracy and user preference can be seen here again.

Compute Derived Value *Visualizations* ($F_{(3.1,187.8)} = 35.3, p < 0.001, \eta_p^2 = 0.67$)

Visualizations ranking from most preferred (left) to least (right)
Table → Bar Chart → Pie Chart → Scatterplot → Line Chart

Pairwise Comparisons (p values are corrected using Bonferroni correction)
Table, Bar Chart, Pie Chart, Scatterplot **vs.** Line Chart ($p < .01$)

Main Findings. We found a significant main effect of Visualization on user preference. Participants preference for using Table, Pie Chart, and Bar Chart is significantly higher than Scatterplot and Line Chart.

Order *Visualizations* ($F_{(3.0,103.3)} = 11.8, p < 0.001, \eta_p^2 = 0.52$)

Visualizations ranking from most preferred (left) to least (right)
Bar Chart → Table → Scatterplot → Line Chart → Pie Chart

Pairwise Comparisons (p values are corrected using Bonferroni correction)
Bar Chart **vs.** Table, Scatterplot, Line Chart, Pie Chart ($p < .05$)

Main Findings. We found a significant main effect of Visualization on user preference. For Order tasks, users preferred Bar Chart significantly more than other visualizations. Moreover, our results indicate that user preference in using Pie Chart is significantly lower than other visualizations. There was not a significant different in user preference for Line Chart and Scatterplot.

Retrieve Value *Visualizations* ($F_{(1.5,417.2)} = 47.1, p < 0.001, \eta_p^2 = 0.73$)

Visualizations ranking from most preferred (left) to least (right)
Table → Bar Chart → Pie Chart → Scatterplot → Line Chart

Pairwise Comparisons (p values are corrected using Bonferroni correction)
Table **vs.** Bar Chart, Pie Chart, Scatterplot, Line Chart ($p < .001$)
Bar Chart **vs.** Pie Chart, Scatterplot, Line Chart ($p < .01$)

Main Findings. We found a significant main effect of Visualization on user preference. User preference for performing Retrieve tasks using Table is significantly higher than other visualizations. After Table, Bar Chart is the second most visualization type highly preferred by users to perform this type of tasks. Moreover, user preference in using Bar Chart is significantly higher than Pie Chart, Scatterplot, and Line Chart. Line Chart and Scatterplot got the lowest user preference.

Characterize Distribution *Visualizations* ($F_{(2.5,20528.2)} = 12.1, p < 0.001, \eta_p^2 = 0.41$)

Visualizations ranking from most preferred (left) to least (right)
Bar Chart → Table → Scatterplot → Pie Chart → Line Chart

Pairwise Comparisons (p values are corrected using Bonferroni correction)
Bar Chart, Table, Scatterplot **vs.** Pie Chart, Line Chart ($p < .01$)

Main Findings. We found a significant effect of Visualization on user preference, and our results indicate that participants preferred Bar Chart, Scatterplot, and Table significantly more than Pie Chart and Line Chart. It is surprising that even though Table was not faster than the other four visualizations, participants highly preferred using it.

Find Extremum *Visualizations* ($F_{(2.8,89.4)} = 8.2, p < 0.001, \eta_p^2 = 0.61$)

Visualizations ranking from most preferred (left) to least (right)
Bar Chart → Table → Scatterplot → Line Chart → Pie Chart

Pairwise Comparisons (p values are corrected using Bonferroni correction)
Bar Chart **vs.** Table, Scatterplot, Line Chart, Pie Chart ($p < .001$)

Main Findings. There is a significant main effect of Visualization on user preference. For Extremum tasks, participant preference in using bar charts is significantly higher than all other visualizations.

Filter *Visualizations* ($F_{(2.2,210.5)} = 42.2, p < 0.001, \eta_p^2 = 0.72$)

Visualizations ranking from most preferred (left) to least (right)
Table → Bar Chart → Pie Chart → Scatterplot → Line Chart

Pairwise Comparisons (p values are corrected using Bonferroni correction)
Bar Chart, Table **vs.** Scatterplot, Line Chart ($p < .01$)
Pie Chart **vs.** Scatterplot, Line Chart ($p < .05$)

Main Findings. We found a significant main effect of Visualization on user preference. In particular, participant preference towards using Table, Bar Chart, and Pie Chart is significantly higher than Line Chart and Scatterplot for Filter tasks.

Determine Range

Visualizations ranking from most preferred (left) to least (right)
Scatterplot → Line Chart → Bar Chart → Pie Chart → Table

Pairwise Comparisons (p values are corrected using Bonferroni correction)
NA

Main Findings. We did not find a significant main effect of Visualization on user preference for Range tasks. This indicates that user preference was not significantly different among all visualizations.

Figure 7: User preference results for different tasks along with statistical test results. For each task, we ranked visualizations from the most preferred (left) to the least preferred (right) based on the average preference rating.

[5] F. Bouali, A. Guettala, and G. Venturini. Vizassist: an interactive user assistant for visual data mining. *The Visual Computer*, pages 1–17, 2015.
[6] M. Brehmer and T. Munzner. A multi-level typology of abstract visualization tasks. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2376–2385, 2013.
[7] S. Card, J. Mackinlay, and B. Shneiderman. Information visualization. *Human-computer interaction: design issues, solutions, and applications*, 181, 2009.
[8] S. Carpendale. Evaluating information visualizations. In A. Kerren, J. T. Stasko, J.-D. Fekete, and C. North, editors, *Information Visualization*, pages 19–45. Springer-Verlag, Berlin, Heidelberg, 2008.
[9] Çağatay Demiralp, P. J.Haas, S. Parthasarathy, and T. Pedapati. Foresight: Recommending visual insights. Under Review at International Conference on Very Large Data Bases (VLDB), 2017.
[10] W. S. Cleveland. *Visualizing Data*. Hobart Press, 1993.
[11] W. S. Cleveland, P. Diaconis, and R. McGill. Variables on scatterplots look more highly correlated when the scales are increased. Technical report, DTIC Document, 1982.
[12] W. S. Cleveland and R. McGill. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American statistical association*, 79(387):531–554, 1984.
[13] W. S. Cleveland and R. McGill. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American Statistical Association*, 79(387):531–554, 1984.
[14] W. S. Cleveland and R. McGill. Graphical perception and graphical methods for analyzing scientific data. *Science*, 229(4716):828–833, 1985.
[15] M. Correll and M. Gleicher. Error bars considered harmful: Exploring alternate encodings for mean and error. *IEEE transactions on visualization and computer graphics*, 20(12):2142–2151, 2014.
[16] F. E. Croxton and R. E. Stryker. Bar charts versus circle diagrams. *Journal of the American Statistical Association*, 22(160):473–482, 1927.
[17] M. Dambacher, P. Haffke, D. Groß, and R. Hübner. Graphs versus numbers: How information format affects risk aversion in gambling. *Judgment and Decision Making*, 11(3):223, 2016.
[18] T. Datasets. https://public.tableau.com/s/resources, 2015.
[19] S. M. Drucker, D. Fisher, R. Sadana, J. Herron, and m. schraefel. Touchviz: A case study comparing two interfaces for data analytics on tablets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 2301–2310, New York, NY, USA, 2013. ACM.
[20] W. C. Eells. The relative merits of circles and bars for representing component parts. *Journal of the American Statistical Association*, 21(154):119–132, 1926.
[21] A. E. Ehrenberg. *Data Reduction: Analysing and interpreting statistical data*. John Wiley and Sons, London, 1975.
[22] S. Few. *Information dashboard design*. O'Reilly, 2006.
[23] T. Gao, M. Dontcheva, E. Adar, Z. Liu, and K. G. Karahalios. Datatone: Managing ambiguity in natural language interfaces for data visualization. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, UIST '15, pages 489–500, New York, NY, USA, 2015. ACM.
[24] R. Garcia-Retamero and M. Galesic. Who proficts from visual aids: Overcoming challenges in people's understanding of risks. *Social science & medicine*, 70(7):1019–1025, 2010.
[25] D. J. Gillan and R. Lewis. A componential model of human interaction with graphs: 1. linear regression modeling. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 36(3):419–440, 1994.
[26] D. Gotz and Z. Wen. Behavior-driven visualization recommendation. In *Proceedings of the 14th International Conference on Intelligent User Interfaces*, IUI '09, pages 315–324, New York, NY, USA, 2009. ACM.
[27] L. Harrison, F. Yang, S. Franconeri, and R. Chang. Ranking visualizations of correlation using weber's law. *Visualization and Computer Graphics, IEEE Transactions on*, 20(12):1943–1952, 2014.
[28] J. Heer and M. Agrawala. Multi-scale banking to 45 degrees. *IEEE Trans. Visualization & Comp. Graphics*, 12:701–708, 2006.
[29] J. Heer and M. Bostock. Crowdsourcing graphical perception: Using mechanical turk to assess visualization design. In *ACM Human Factors in Computing Systems (CHI)*, 2010.
[30] J. Heer and M. Bostock. Crowdsourcing graphical perception: using mechanical turk to assess visualization design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 203–212. ACM, 2010.
[31] J. Heer, N. Kong, and M. Agrawala. Sizing the horizon: The effects of chart size and layering on the graphical perception of time series visualizations. In *ACM Human Factors in Computing Systems (CHI)*, 2009.
[32] H. V. Henderson and P. F. Velleman. Building multiple regression models interactively. *Biometrics*, pages 391–411, 1981.
[33] H. Ibrekk and M. G. Morgan. Graphical communication of uncertain quantities to nontechnical people. *Risk analysis*, 7(4):519–529, 1987.
[34] M. Kay and J. Heer. Beyond weber's law: A second look at ranking visualizations of correlation. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2016.
[35] M. Kay, T. Kola, J. R. Hullman, and S. A. Munson. When (ish) is my bus? user-centered visualizations of uncertainty in everyday, mobile predictive systems. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 5092–5103. ACM, 2016.
[36] N. Kong, J. Heer, and M. Agrawala. Perceptual guidelines for creating rectangular treemaps. *IEEE Trans. Visualization & Comp. Graphics*, 16(6):990–998, 2010.
[37] R. Kosara and D. Skau. Judgment error in pie chart variations. In *Proceedings of the Eurographics/IEEE VGTC Symposium on Visualization*, pages 91–95. Wiley Online Library, 2016.
[38] S. M. Kosslyn. Understanding charts and graphs. *Applied cognitive psychology*, 3(3):185–225, 1989.
[39] B. C. Kwon and B. Lee. A comparative evaluation on online learning approaches using parallel coordinate visualization. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 993–997, New York, NY, USA, 2016. ACM.
[40] H. Lam, T. Munzner, and R. Kincaid. Overview use in multiple visual information resolution interfaces. *IEEE Trans. Visualization & Comp. Graphics*, 13(6):1278–1285, 2007.
[41] B. Lee, C. Plaisant, C. S. Parr, J.-D. Fekete, and N. Henry. Task taxonomy for graph visualization. In *Procs. BELIV*, pages 1–5. ACM, 2006.
[42] S. Lee, S. H. Kim, and B. C. Kwon. Vlat: Development of a visualization literacy assessment test. *IEEE Transactions on Visualization and Computer Graphics*, PP(99):1–1, 2016.
[43] S. Lewandowsky and I. Spence. Discriminating strata in scatterplots. *Journal of American Statistical Association*, 84(407):682–688, 1989.
[44] P. Lyman and H. Varian. How much information 2003?, 2004.
[45] A. MacEachren. *How Maps Work: Representation, Visualization, and Design*. Guilford Press, 1995.
[46] J. D. Mackinlay. Automating the design of graphical presentations of relational information. *ACM Trans. Graph.*, 5(2):110–141, 1986.
[47] J. S. Olson and W. A. Kellogg. *Ways of Knowing in HCI*. Springer, 2014.
[48] A. V. Pandey, J. Krause, C. Felix, J. Boy, and E. Bertini. Towards understanding human similarity perception in the analysis of large sets of scatter plots. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 3659–3669, New York, NY, USA, 2016. ACM.
[49] S. Pinker. A theory of graph comprehension. *Artificial intelligence and the future of testing*, pages 73–126, 1990.
[50] U. M. L. Repository. https://archive.ics.uci.edu/ml/datasets.html, 2016.
[51] B. Saket, P. Simonetto, S. Kobourov, and K. Borner. Node, node-link, and node-link-group diagrams: An evaluation. *Visualization and Computer Graphics, IEEE Transactions on*, 20(12):2231–2240, 2014.
[52] B. Saket, A. Srinivasan, E. D. Ragan, and A. Endert. Evaluating interactive graphical encodings for data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 2017.
[53] B. S. Santos. Evaluating visualization techniques and tools: What are the main issues. In *the 2008 AVI Workshop on Beyond Time and Errors: Novel Evaluation Methods For information Visualization (BELIV'08)*, 2008.
[54] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pages 336–343. IEEE, 1996.
[55] B. Shortridge. Stimulus processing models from psychology: can we use them in cartography? *The American Cartographer*, 9:155–167, 1982.
[56] D. Simkin and R. Hastie. An information-processing analysis of graph perception. *Journal of American Statistical Association*, 82(398):454–465, 1987.
[57] D. Simkin and R. Hastie. An information-processing analysis of graph perception. *Journal of the American Statistical Association*, 82(398):454–465, 1987.
[58] D. Skau and R. Kosara. Arcs, angles, or areas: Individual data encodings in pie and donut charts. In *Computer Graphics Forum*, volume 35, pages 121–130. Wiley Online Library, 2016.
[59] I. Spence and S. Lewandowsky. Displaying proportions and percentages. *Applied Cognitive Psychology*, 5:61–77, 1991.

[60] I. Spence and S. Lewandowsky. Displaying proportions and percentages. *Applied Cognitive Psychology*, 5(1):61–77, 1991.
[61] SpotFire. http://www.spotfire.com, 2016.
[62] Tableau. Tableau software, http://www.tableau.com/, 2016.
[63] J. Talbot, J. Gerth, and P. Hanrahan. Arc length-based aspect ratio selection. *IEEE Trans. Visualization & Comp. Graphics*, 2011.
[64] J. Talbot, S. Lin, and P. Hanrahan. An extension of Wilkinson's algorithm for positioning tick labels on axes. *IEEE Trans. Visualization & Comp. Graphics*, 2010.
[65] J. Talbot, V. Setlur, and A. Anand. Four experiments on the perception of bar charts. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2152–2160, Dec 2014.
[66] L. Tremmel. The visual separability of plotting symbols in scatterplots. *Journal of Computational and Graphical Statistics*, 4(2):101–112, 1995.
[67] M. Vartak, S. Madden, A. Parameswaran, and N. Polyzotis. Seedb: automatically generating query visualizations. *Proceedings of the VLDB Endowment*, 7(13):1581–1584, 2014.
[68] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2015.
[69] J. Zacks and B. Tversky. Bars and lines: A study of graphic communication. *Memory & Cognition*, 27(6):1073–1079, 1999.

Beyond Usability and Performance: A Review of User Experience-focused Evaluations in Visualization

Bahador Saket
Georgia Institute of Technology
saket@gatech.edu

Alex Endert
Georgia Institute of Technology
endert@gatech.edu

John Stasko
Georgia Institute of Technology
stasko@cc.gatech.edu

ABSTRACT

Traditionally, studies of data visualization techniques and systems have evaluated visualizations with respect to *usability goals* such as effectiveness and efficiency. These studies assess performance-related metrics such as time and correctness of participants completing analytic tasks. Alternatively, several studies in InfoVis recently have evaluated visualizations by investigating *user experience goals* such as memorability, engagement, enjoyment and fun. These studies employ somewhat different evaluation methodologies to assess these other goals. The growing number of these studies, their alternative methodologies, and disagreements concerning their importance have motivated us to more carefully examine them. In this article, we review this growing collection of visualization evaluations that examine user experience goals and we discuss multiple issues regarding the studies including questions about their motivation and utility. Our aim is to provide a resource for future work that plans to evaluate visualizations using these goals.

Keywords

Data visualization, user experience goals, usability goals, memorability, enjoyment, engagement.

1. INTRODUCTION

Evaluating what has been built is at the heart of a design process [42]. Evaluating a design allows one to ensure that it is appropriate and meets user requirements. For many years the notion of a "successful" design has centered on a system meeting all of its *usability goals* [41, 42]. Usability refers to the effectiveness, efficiency, safety, utility, and learnability of a design [17]. These principles have been at the core of evaluation research within the field of human-computer interaction for decades.

Today, however, not only designers but also cognitive scientists acknowledge that other important design goals must be considered. These new goals go beyond traditional usability-driven objectives and address issues related to a person's experience when using a system. In a world full of choices where the short attention of a person becomes a prime resource, it is essential that designers create not just usable products, but also memorable, fun, enjoyable, and engaging experiences [41]. Multiple HCI researchers have emphasized the importance of emotion, enjoyment and fun, memorability, and engagement in their work [5, 19, 37, 55, 56]. As Rogers et al. advocate, designs should meet both *usability goals* and *user experience goals* [54]. User experience goals complement traditional usability goals to provide more holistic evaluations. In his recent book *Emotional Design*, Don Norman states [41]:

> *" we scientists now understand how important emotion is to everyday life, how valuable. Sure, utility and usability are important, but without fun and pleasure, joy and excitement, and yes, anxiety and anger, fear and rage, our lives would be incomplete."*

Like that done in HCI, the field of data visualization also has embraced the importance of usability. However, the primary design goal for visualization is to effectively communicate a thorough understanding of the data it represents. This utility of a visualization does include usability goals, but ultimately revolves around the visualization's ability to help people better understand data. We will use the term "performance" to represent this ultimate purpose of a visualization. The vast majority of visualization user studies and evaluations thus have focused upon performance-related characteristics, and they have assessed metrics such as task completion-time and accuracy, which can be objectively measured.

The BELIV Workshop was created to explore new evaluation methods that go beyond these well-established metrics such as time and errors. While it has largely succeeded in doing so, we would argue that the vast majority of research presented at the workshop has focused on performance-related metrics such as "insight" or has proposed new evaluation methodologies such as more observational, qualitative case studies. Those metrics and methods still relate to a visualization's ability to help people understand data better and answer questions about that data. Evaluation studies of this type still focus on assessing performance goals.

As discussed above, the HCI community has now embraced the importance of design goals beyond the usability of a system. Recently, a similar effort to encompass and evaluate design goals beyond a system's prime purpose has emerged in the field of data visualization as well. In visualization, multiple projects have evaluated aspects of a visualization beyond its utility and performance-related characteristics. Several recent papers study enjoyment, fun, memorability, and engagement of visualizations. Brehmer and Munzner [12] recognize enjoyment as one of the three reasons that one might use visualizations. Haroz et al. [24] argue that a primary goal of some visualizations (particularly news articles) is to engage people and make them pause and look. Several studies also emphasize the importance of memorability in visualizations [9, 8, 52]. These types of studies that move beyond performance-related objectives are not universally accepted, however. Some people question whether they

BELIV '16 Baltimore, Maryland USA

DOI: 10.475/123_4

deserve a place in the visualization research agenda and believe they only distract from the most important cognitive measures [21].

Historically, the general interest in and adoption of usability goals can be explained at least in small part by how well-understood and well-defined these metrics are to measure (e.g., performance time and accuracy). This is especially true in contrast to user experience goals that appear to be more difficult to define, much less to measure [50]. Despite this challenge, a number of studies recently have begun to examine and evaluate visualizations with respect to user experience goals. However, the methods these studies applied vary significantly, employing qualitative methods, quantitative methods, or some combination of both. Unfortunately, no existing work summarizes and classifies the methodologies used by these studies to assess user experience goals.

In this paper, we review a collection of academic publications and online articles that evaluate visualizations based on user experience goals. Each of them contribute methods, metrics, and/or comment on the application of user experience goals in data visualization. More specifically, we concentrate on memorability and recall, engagement, and enjoyment and fun (defined in Table 1). In reviewing the existing literature in this area, we specifically explain the methodology used by each study to measure these metrics.

2. USER EXPERIENCE GOALS

Undoubtedly, the vast majority of people would agree that a visualization's ability to convey understanding of the data it represents is absolutely paramount. However, researchers have now begun to explore other applications and benefits of visualizations as well. The designers of visualizations often have different goals for their creations. Accordingly, how one evaluates a visualization should depend on those corresponding goals.

In this section we review user studies that have focused on goals and metrics other than typical performance measures related to knowledge about the underlying data. We examine three primary user experience goals: memorability and recall, engagement, and enjoyment and fun.

Our methodology for collecting articles reviewed in this work began by searching for academic publications that evaluated visualizations via these user experience goals. We looked in well-known visualization venues such as the IEEE InfoVis, PacificVis, EuroGraphics EuroVis, and ACM CHI Conferences, as well as the ACM BELIV Workshop. Additionally, we searched in the *Information Visualization* and *IEEE TVCG* journals. Within each relevant article we found, we examined the references to discover other potentially relevant articles in these and other venues.

While reviewing these articles we performed two phases of coding. In the first phase, we categorized each study based on the user experience goals (memorability, enjoyment, and engagement). In the second phase, we coded memorability studies based on what types of memory (e.g., short-term, working, and long-term) they evaluated. In both phases, we used the terms reported in each article to categorize the evaluation and report on the type(s) of memory being examined. That is, we categorize each memorability study by the type of memory it examines, according to the paper's authors. Note that in some cases different authors apply these terms for types of memory in different ways.

2.1 Memorability and Recall

Memorability has been defined as a capability of maintaining and retrieving information [13]. Multiple studies have examined the memorability of visualizations, a person's ability to remember and recall information about the visualization. These studies all have employed an implicit assumption that being able to remember a visualization is good. Presumably, this is because the visualization has made a lasting impact and facilitates a person revisiting or discussing the relevant data at a later time. Different studies have focused on the memorability of different aspects of the visualization, however, such as its message, the underlying data, or its visual imagery. Table 2 shows a summary of the methods these studies used to assess memorability. We also discuss each in more detail below.

Table 1: Definitions of the three user experience goals discussed in this paper.

GOALS	DEFINITION
Memorability	Memorability is a capability of maintaining and retrieving information [13].
Engagement	Emotional, cognitive and behavioural connection that exists, at any point in time and possibly over time, between a user and a resource [2].
Enjoyment	Feeling that causes a person to experience pleasure [16]. Pleasure is recognized with occurrent happiness and excitement, which can be explained in terms of belief, desire, and thought.

2.1.1 *Memorability of Embellished Visualizations*

The work by Bateman et al. [5] was one of the first studies that aimed to measure memorability of visualizations. The goal of the study was to evaluate the impact of embellishments on visualization memorability and comprehension. The researchers conducted a multi-phase controlled experiment to test comprehension and recall of visualizations using embellished and plain versions. In the first phase of their experiment, participants were shown a series of visualizations and asked to perform four tasks (e.g., subject, values, trend, value message). In the second phase (recall phase), the researchers measured both short-term (5 minutes after the first phase) and long-term (about a week after the first phase) recall of the visualizations. Participants first were asked to remember as many visualizations as possible. They then were asked to describe the charts as completely as possible. The researchers found that the participants were better able to recall the embellished visualizations, and participants' accuracy in describing the embellished visualizations was no worse than for plain ones.

Li et al. [34] reported a replication of the Bateman et al. study, limiting their selection to charts of data sets with 10 or more observations. The researchers found that the presence of a time limit decreased the accuracy of participants in describing and recalling visualizations, while the type of visualization significantly affected short-term recall.

Borgo et al. [7] also conducted an experiment to examine the effects of visual embellishments in visualization processes in relation to several fundamental aspects of perception and cognition including both working memory and long-term memorability.The researchers showed 18 visualizations (6 horizontal bar charts, 6 vertical bar charts and 6 bubble charts) to participants. Some of the visualizations were embellished and some were not. First participants were shown each visualization for 9 seconds. Participants next were shown a gray masking screen for 5 seconds and then were asked a question regarding the visualization that just was shown. Study results indicate that when embellishments are grouped with a numerical representation, they have the most beneficial influence on working memory tasks. The researchers also investigated the effect of embellished charts on long-term memory. Participants were each shown a visualization for 9 seconds, similar as done in

Table 2: Different methods used to measure memorability in visualizations. Each row of the table describes a study that has measured memorability. Columns indicate WHAT each paper investigates in the experiment, HOW it was evaluated, WHEN it was evaluated during the experiment, HOW MANY subjects have participated in the study, and WHICH VISUALIZATIONS were used in the study.

REFERENCES	WHAT?	HOW?	WHEN?	HOW MANY?	WHICH VISUALIZATIONS?
Bateman et al. [5]	Immediate memory Long-term memory	**Immediate Memory:** Asked participants to recall as many visualizations as possible. **Long-term Memory:** Asked participants to recall as many visualizations as possible.	**Immediate Memory:** Five minutes after performing tasks using those visualizations. **Long-term Memory:** Two to three weeks after performing tasks using those visualizations.	20 participants	Bar chart, line chart, and pie chart
Li et al. [34]	Short-term memory Long-term memory	**Short-term Memory:** Asked participants to recall as many visualizations as possible. **Long-term Memory:** Asked participants to recall as many visualizations as possible.	**Short-term Memory:** Five minutes after performing tasks using those visualizations. **Long-term Memory:** Four days after performing tasks using those visualizations.	15 participants	Bar chart, line chart, and pie chart
Saket et al. [52]	Short-term memory Long-term memory	**Short-term Memory:** Asked participants to recall data shown in the visualizations. **Long-term Memory:** Asked participants to recall data shown in the visualizations.	**Short-term Memory:** Right after performing the tasks using those visualizations. **Long-term Memory:** Four days after performing the tasks using those visualizations.	40 participants	Node-Link and Node-Link-Group (map) visualizations
Borgo et al. [7]	Sensory memory Working memory	**Sensory Memory:** Asked participants to answer questions about visualizations. **Working Memory:** Asked participants to answer questions about visualizations.	**Sensory Memory:** Each visualization was shown for 9 seconds (5 seconds break between each two visualizations). **Working Memory:** Each visualization was shown for 9 seconds (30 seconds break between each two visualizations).	35 participants	Bar chart and Bubble chart, and line chart
Haroz et al. [24]	Working memory	**Working Memory:** Asked participants to recall data shown in the visualizations.	**Working Memory:** Each visualization was shown to participants for 1.5 seconds.	22 participants	Different variation of bar charts
Borkin et al. [9]	Not specified	Asked participants to press a key if they saw a visualization for a second time.	Visualizations were shown for a few seconds with 1.5 seconds gap.	261 participants	Almost all types of well-known visualizations
Borkin et al. [8]	Not specified	**Recognition Phase:** Asked participants to press a key if they saw a visualization for a second time. **Recall Phase:** Asked participants to describe visualizations as detail as possible.	**Recognition Phase:** After showing each visualization for 10 seconds. **Recall Phase:** After showing all visualizations (each visualization was shown for 10 seconds).	33 participants	Almost all types of well-known visualizations
Marriott et al. [38]	Short-term memory	**Short-term Memory:** Asked participants to redraw the visualizations.	**Short-term Memory:** After seeing and performing tasks using visualizations.	25 participants	Graph visualizations

the working memory section. Next, a gray masking screen appeared for 30 seconds. The researchers found that visual embellishment enhances information retention in terms of both accuracy and time required for short-term memory recall. Since they concentrated on "visual perception and cognitive speed-focused tasks" that leverage cognitive abilities, they used analytical tasks that forced attention to switch from one task to another.

Borkin et al. [9] conducted an online study with Amazon's Mechanical Turk that used thousands of real world visualizations. Their goal was to reduce the biases caused by the limited number of participants and target visualizations. Each target visualization was coded by the research team for different characteristics (visualization type, number of colors used, data-ink ratio, etc.). Their experiment was set up as a game on Amazon Mechanical Turk, where participants were presented with series of visualizations (each visualization was shown to participants for 1 second, with a 1.4 seconds gap between consecutive visualizations). Participants had to press a key if they saw a visualization for the second time in the sequence. The study found that attributes such as color and the inclusion of human-recognizable objects improve memorability.

Haroz et al. [24] tested the effect of pictographical representations on working memory (memory for briefly glanced information), as well as memory under load (memory is more crowded). In order to test the impact of pictographical representations on working memory, they ran a study where participants were shown visualizations with different values for 1.5 seconds. After 1.5 seconds, participants were shown the visualization again but this time values in the visualization were invisible. Participants were asked to recall each of the values in the visualization they just saw. In the second phase of the study, the researchers tested how pictographs impact memory when memory is more crowded. The procedure for this experiment was similar to the first phase (working memory) but used a "1-back design". Participants were asked to recall visualizations, but this time they were always tested on the visualization before the one that they just saw, introducing the need to store two charts at all times. The researchers found that ISOTYPE visualizations were recalled more accurately than those having plain charts.

More recently, Borkin et al. [8] conducted a more comprehensive study to move beyond memorability and investigate how visualizations are recognized and recalled. Their experiment contained three phases: *encoding*, *recognition*, and *recall*. In the encoding phase, participants were shown different visualizations (each for 10 seconds) while being eye tracked. The goal of this phase was to "allow participants more time to explore visualizations and to facilitate the collection of eye movements." In the recognition phase, participants were shown exactly the same visualizations as in the first phase of the experiment, as well as different unseen visualizations. Each visualization appeared for 2 seconds. Participants had to indicate which visualizations they remembered from the first phase of the experiment. Finally, in the recall phase of the experiment, participants were shown the visualizations they successfully recognized in the prior phase, and they were asked to describe the visualization in as much detail as possible. From this study, four general design guidelines emerge. First, visualizations that are memorable at a glance also have memorable content. Second, titles and text are important elements for recalling a visualization's message. Third, pictorial images do not decrease the memorability or understanding of a visualization. Fourth, redundancy increases recall and understanding of visualizations.

2.1.2 Memorability of Network Visualizations

Network diagrams, in particular, have served as the focus of multiple memorability studies [29, 38, 52]. As a part of an experiment comparing the effectiveness of four different visualization techniques, Jianu et al. [29] asked participants to complete several different tasks including one memorability related task. Their results suggest that Map-like network visualizations are more memorable than three other techniques. In a study aiming to investigate cognitive impact of different layout features (e.g, symmetry and alignment) on the memorability of graphs, participants were shown different graphs and were asked to study, remember and redraw them [38]. Their findings show that visual features such as symmetry, collinearity and orthogonality of network visualizations have an impact on memorability.

In another memorability focused study, Saket et al. [52] assessed the memorability of displayed network data, minutes and days after interaction with two different types of visualizations (node-link diagrams and map-based visualizations). The researchers argued that, unlike previous work, their study sought to measure the memorability of data shown in a visualization and not just the visualization appearance itself. They ran a three-phase experiment. In the first phase of the study, participants were asked to examine the visualization without a preset time limit and without performing specific tasks. In the second phase, participants were asked to perform different tasks using the visualization as fast and as accurately as possible. Finally, after a predetermined period of time depending on whether the subject was in the immediate or long-term treatment condition, participants were asked to recall the underlying data presented using visualizations. Study results found that participants recalled data in map-like visualizations more accurately, but not any faster, both in terms of remembering graphs and actual content.

In general, results of all these studies examining memorability have shown that embellishing visualizations appears to enhance their memorability. Almost all these studies additionally found that visualization embellishment did not come at the cost of performance. Since the majority of these studies evaluated the memorability of visual encodings used in visualizations and not the underlying data, whether embellishment enhances the memorability of the underlying data remains an open problem. In spite of these findings, the importance of the memorability of a visualization is not a universally agreed upon point. In fact, it has been the subject of heated debate recently. We will return to a discussion of such issues later in this article.

2.2 Engagement

User engagement has been defined as the *"emotional, cognitive and behavioural connection that exists, at any point in time and possibly over time, between a user and a resource* [2]". This broad definition of user engagement emphasizes its holistic character and different methods that can be applied to measure it. Previous work in visualizations refers to user engagement as users' interest in putting effort to investigate and explore visualizations and gain more insights [24, 10]. This definition of engagement supports the argument that an engaging activity attracts user attention to the activity, excluding other things and people [43]. The amount of time a person spends on an activity has been shown to be an effective indicator of the level of his/her cognitive involvement in that activity [44]. That is, the more engaged someone is in an activity, the more likely they underestimate the passage of time [3]. Read et al. [49] also indicated that people will remember engaging experiences and want to repeat them. This aspect of user engagement refers to the possibility of remembering an experience and the desire to repeat it.

Engagement factors are known as a medium for attracting users' interest in designs [41]. We speculate that this also applies to visualizations. Heer et al. [26] advocated for engaging new audiences for visualizations. Mahyar et al. [35] also suggested moving toward engaging visualizations.

Today, many visualizations are deployed on the Web where metrics such as visits and dwell time are considered important. Such metrics often drive commercial rewards and monetary gains. Fundamentally, a more engaging visualization that is viewed more often will be perceived as being more "successful" and impactful.

Table 3 summarizes different studies that have examined engagement as a primary metric, and we discuss each of these studies further below.

Haroz et al. [24] conducted a study to measure the engagement of different pictographical representations. More specifically, they wanted to know *"Will an ISOTYPE visualization be better at capturing attention than a simple bar chart?"* In the engagement phase of their experiment, participants were shown a 3x3 grid of items (3 bar charts, 3 stacked pictograph charts, and 3 pieces of text). Each item contained a short title above a small, blurred thumbnail. The thumbnail was either a visualization or a set of sentences about the topic. Participants were given two minutes and asked to look through the thumbnails. They could click whichever item they were interested in to view the information at full screen resolution. Clicking again would return them to the main grid. The researchers measured engagement of visualization by computing the total time participants spent working with each type of visualization. Study results showed that participants were more engaged with ISOTYPE-style visualizations.

Previous work also showed how product reaction card methods can be used to evaluate user experience with different visualization types [6]. Reaction cards provide "a way for users to tell the story of their experience, choosing the words that have meaning to them as triggers to express their feelings - negative or positive - about their experience" [4]. Suggesting the use of product reaction card method to evaluate user experience, Merčun evaluated four different visualization techniques (Indented list, Radial tree, Circlepack, Sunburst) to demonstrate how the results of this method could be analysed and used for comparing different designs [39]. Merčun first recruited 120 participants and asked each of them to work with three randomly counterbalanced visualization techniques. After completing 10 tasks

Table 3: Different methods used to measure engagement in visualizations. Each row of the table describes a study that has measured engagement. HOW **engagement was evaluated,** WHEN **it was evaluated during the experiment,** HOW MANY **subjects have participated in the study, and** WHICH VISUALIZATIONS **were used in the study.**

REFERENCES	HOW?	WHEN?	HOW MANY?	WHICH VISUALIZATIONS?
Haroz et al. [24]	Measured total time spent working with charts.	While participants were interacting with the charts.	10 participants	Different variation of bar charts
Boy et al. [10]	Measured total time spent working with charts. In addition, recorded interaction logs (e.g., number of clicks) with charts.	While participants were interacting with the charts.	Not included	Variety of interactive visualizations
Merčun [4]	Product reaction card method.	While participants were interacting with the charts.	120 participants	Indented list, radial tree, Circlepack, and Sunburst
Saket et al. [51]	Total time spent looking at a chart.	While working with the charts.	32 participants	Node-Link and Node-Link-Group (map) visualizations

using each visualization, participants were asked to choose from a set of cards (adjectives) those that best reflect their experience with the visualization. Finally, participants were asked to explain more about the visualization they used by applying the selected adjectives. The study found that participants were more engaged with sunburst and circlepack visualizations than other ones. In addition, the study shows that product reaction card method can be used to evaluate aspects such as engagement, ease of use, appeal, efficiency, and usefulness of visualizations.

Boy et al. [10] investigated the effect of using initial narrative visualization techniques and storytelling on user engagement with exploratory information visualizations. They ran three field study experiments. For each experiment, they created two different exploratory visualization webpages. One webpage contained *"an introductory narrative component, which told a short 'story' about the topic and context of the data provided initial insights and unanswered questions, and introduced the different visual encodings"*; and another webpage that did not. The researchers sought to understand whether augmenting such a visualization with an introductory 'story' can help engage users in exploration. In order to measure user engagement, they examined different user interaction logs such as the amount of time participants spent to explore, number of meaningful interactions, etc. Their results indicated that combining exploratory visualizations with introductory 'stories' does not enhance user-engagement.

Saket et al. [51] measured people's engagement with two different visualizations of the same relational data–node-link and map-based visualizations–by measuring the total amount of time participants spent looking at visualizations. Their main insight for this way of measurement is that *"if users are surreptitiously given the option to experience either of two visualizations, they will spend longer with the one they enjoy better."* In the first phase of their study, they invited each participant to a room where posters of two different types of visualizations were placed on two walls. The interviewer then told each participant: *"I need to bring some equipment before running the experiment. Please stay in the room for few minutes. I will be back soon and we'll start the experiment."* The interviewer then left the room and came back after a few minutes. During this time a camera was recording participants' interactions with the posters. The study findings show that the participants spent on average more time looking at the map visualizations but the difference was not significant.

Mahyar et al. [35] emphasize the lack of a taxonomy for measuring engagement in visualization, one that evaluates engagement based on the level of a user's understanding of a visualization. The researchers proposed a five level taxonomy for engagement. Each level of the taxonomy is correlated to the level of cognitive task user required to perform.

Expose: knowing how to read a data point
Involve: interacting with the visualization and manipulating the data
Analyze: analyzing the data, finding trends, and outliers
Synthesize: being able to form and evaluate a hypothesis
Decide: being able to draw final decisions based on evaluations of different hypotheses

These few prior studies of engagement used metrics such as a count of user interactions with a visualization and time spent viewing and/or working with a visualization. The studies were different enough so that no one common take-away message emerged from them. Although their methodologies can begin to help people measure engagement broadly, the metrics they assessed fall short of evaluating deeper levels of engagement. Additionally, it remains an open question of how to extract factors that make one visualization more engaging than another.

2.3 Enjoyment and Fun

Davis defines enjoyment as a feeling that causes a person to experience pleasure [16]. Pleasure is recognized with occurrent happiness and excitement, which can be explained in terms of belief, desire, and thought. Furthermore, enjoyable and fun experiences make people smile [41, 55]. Enjoyment has been carefully studied in psychology and HCI. For instance, Brandtzæg et al. [11] suggested that when designing for enjoyment, one should consider demands, allow a high degree of decision latitude, and provide socially rewarding activity. One of the best known models for understanding and measuring enjoyment in psychology is the flow model of Csikszentmihalyi [15]. In a series of experiments in different countries, people were asked to describe when and how they achieved the highest level of enjoyment when performing some activity. Csikszentmihalyi notes,

> *"Regardless of culture, social class, gender or age, the respondents described enjoyment in very much the same way. What they did to experience enjoyment varied dramatically – the elderly Koreans liked to meditate, the teenage Japanese liked to swarm around in motorcycle gangs – but they described how it felt when they enjoyed themselves in almost identical terms"*.

He then suggests several factors (challenge, focus, clarity, feedback, control and immersion) that encompass the experience of enjoyment. His flow model has been applied by researchers in other fields to create new models [23, 56, 57] and to assess enjoyment [45,

Table 4: Different methods used to measure enjoyment in visualizations. Each row of the table describes a study that has measured enjoyment. HOW **enjoyment was evaluated,** WHEN **it was evaluated during the experiment,** HOW MANY **subjects have participated in the study, and** WHICH VISUALIZATIONS **were used in the study.**

REFERENCES	HOW?	WHEN?	HOW MANY?	WHICH VISUALIZATIONS?
Bateman et al. [5]	Self Report (asked participants to rate enjoyability of each type of chart)	After performing all tasks using each type of charts.	20 participants	Bar chart, line chart, and pie chart
Li et al. [34]	Self Report (asked participants to rate enjoyability of each type of chart)	After performing all tasks using each type of charts.	15 participants	Bar chart, line chart, and pie chart
Saket et al. [51]	Applied Flow Model in visualizations [50] and asked open-ended questions.	After performing all tasks using each type of charts.	32 participants	Node-Link and Node-Link-Group (map) visualizations

46, 48]. Malone explored the topic of enjoyable and fun interfaces in the early studies of games [36]. He summarized the design heuristics for enjoyable and fun interfaces with these criteria: challenge, curiosity, and fantasy. Later, Shneiderman commented on enjoyable and fun experiences. He stated, *"They [enjoyable and fun experiences] are a break from the ordinary and bring satisfying feelings of pleasure for body and mind."* [55]

While enjoyment and fun have been discussed and studied extensively in psychology [11, 15, 48] and HCI [23, 37, 55, 56], the enjoyment and fun aspects of visualizations have not been explored significantly, even though enjoyment is often given as a reason to consume visualizations [12]. Moreover, previous work indicated that positive mental states appear linked to better problem-solving performance in general [22] and in information visualization in particular [25, 28]. In other words, if people are happier, then they will be more effective performing tasks using visualizations.

Fundamentally, people gravitate toward activities that are fun and enjoyable. As discussed above regarding engagement, metrics such as page visits and time on task are often viewed as measures of success, especially on the web. Online visualizations such as the Baby Name Wizard [1] have drawn tremendous interest and attention. The creators of the system speculated that factors such as enjoyment and engagement helped foster that interest [58].

Although enjoyment and engagement are actually two different concepts [59], some studies in visualization have used the terms interchangeably. We view them in this context as related, but not quite the same. Enjoyable activities typically are strongly engaging too, but it is clearly possible to be deeply engaged in an activity without the activity necessarily being fun and enjoyable. Table 4 reviews studies examining enoyment issues, and we describe each of the studies in further detail below.

Earlier, we described a study by Bateman et al. [5] to evaluate the comprehension and recall of charts using an embellished version and a plain version. As a part of this study, the researchers also asked participants to rate the enjoyability of each type of chart. Their results suggest that embellished charts are more enjoyable than plain ones. Li et al. [34] recently reported a replication, limiting their selection to those charts that consist of data sets with 10 or more observations. They also found that embellished charts are more enjoyable than plain ones.

Saket et al. [50] subsequently argued that although the study by Bateman et al. identified which visualization type is more enjoyable, it did not identify *what* makes one type of visualization more enjoyable than another. The researchers then suggested a model for measuring enjoyment and flow in visualizations. Their proposed model is built based on the Flow model of Csikszentimihayli [15] and Munzner's nested model [40] of visualization evaluation. More specifically, the model suggests that enjoyment in information visualization encompasses six different elements, to measure enjoyment evaluations must control as many of these elements as possible.

Challenge: Challenges in a visualization should match the skills of the user who is working with the visualization.

Focus: the user should be able to have a complete attention on the task at the hand.

Clarity: the user must understand exactly what the task's goals are.

Feedback: Visualization provides immediate feedback about the progress with he task at the hand.

Control: the user feels a complete control over the visualization itself and interaction techniques available to him/her.

Immersion: the user loses his/her sense of self and become "lost" in the process of working with a visualization.

In a subsequent study, Saket et al. measured the enjoyment and fun of two different node-link and node-link-group visualizations using their enjoyment model [51]. The study determined that participants found map-like visualizations more enjoyable than node-link visualizations.

In this small set of studies examining enjoyment of visualizations, the researchers used self-reporting methods such as Likert scale questions and interview questions to gain subjective information about participants' perceptions of a visualization's enjoyability. The common take-away was that embellished visualizations or ones with more pictorial representations tended to be more enjoyable. This is very initial research, however, and further work needs to be done to identify the particular aspects of a visualization that might make it more enjoyable and fun.

3. DISCUSSION

3.1 Debating the Merits

The importance of factors outside of the traditional usability and performance objectives for data visualization research is not a universally agreed upon view. Some researchers and practitioners have questioned whether issues such as memorability and enjoyment are important enough to merit further study.

In one notable example, Borkin et al. [9] published a study in 2013 that evaluated the memorability of a variety of real world visualizations. Visualization practitioner Stephen Few commented on this study in his blog, arguing that it simply assessed whether participants could remember a visualization's design, not its content, and that it did not reveal anything about what makes a visualization actually memorable [20]. His criticisms centered on a belief that visualizations cannot be read and understood in one second.

In 2015, Borkin et al. [8] conducted a more comprehensive study to move beyond memorability and investigate how visualizations are recognized and recalled. However, very soon after that in a newsletter entitled "Information Visualization Research as Pseudo-Science", Few again critiqued the previous work on memorability and more specifically this most recent study [21]. Few stated:

> *"Visualizations don't need to be designed for memorability – they need to be designed for comprehension. For most visualizations, the comprehension that they provide need only last until the decision that it informs is made. Usually, that is only a matter of seconds."*

Following Few's review, Ben Jones [30], a data visualization practitioner, responded to Few's newsletter. Jones stated:

> *"This statement [mentioned above by Stephen Few] helped me understand why Few and I disagree about memorability: we disagree about how data visualizations are used by groups of people. Simply put, I don't believe data visualizations are usually followed by decisions only a matter of seconds later. That may be how a robot or a computer algorithm would approach decision-making, but it's just not how groups of humans in organizations go about it."*

Later, in his own blog, Robert Kosara also responded to Few's article and advocated for investigating memorability in visualizations [31]. From his point of view, many cases exist where people would benefit from their audience remembering the visualizations (e.g., visualizations showing health related information).

We surmise that all visualization researchers consider a visualization's ability to accurately and effectively convey an understanding of the data it represents to be paramount. Debates such as those described above also make it clear that the importance of user experience-related goals is not as clear or universally agreed upon. Just how important is it for a visualization to be memorable, engaging, and/or fun? Moreover, do those attributes come at a corresponding cost? Research questions such as these addressing the value and merits of different characteristics arise in addition to more basic questions on how to simply measure and evaluate them. This article primarily has focused on the methodological issues involved in user experience metrics, but we felt it important to at least discuss the value issues and raise fundamental questions about the metrics' importance in the field as a whole.

3.2 Evaluation Methodology Challenges

As is evident from the studies reviewed earlier, a variety of evaluation methods have been employed to assess user experience metrics. One key differentiating factor appears to be whether subjective, often self-reported, measures are employed or more objective measures are gathered.

Memorability seems a little different than the other two prime goals we profiled. It easily lends itself to an objective measurement—Can a person remember a visualization or not? But even this question is not so simple. Many of the memorability studies tested whether a person could recall the imagery associated with a visualization, but not anything about the data being represented and its "story" or message. Aren't those things ultimately what the designer of a visualization wants their audience to take away, not just the visualization's appearance and style?

To assess engagement, many studies measured the time a person spent viewing or interacting with a visualization, which is clearly one form of engagement. But how about stronger immersion and a visualization's ability to foster people wanting to know more about the data and its domain? Facilitating deep(er) interest is another type of engagement that appears to be more difficult to objectively assess.

Finally, the sense of enjoyment and fun a visualization generates is typically measured through self-reported subjective methods such as interviews and Likert-style questions. Self-reported data often is questioned, however. Perhaps more objective measures such as counting smiles or laughs from people could be explored. It also seems clear that many forms of acknowledgment on social media such as page links, visits, likes, etc., can be correlated with stronger senses of engagement and enjoyment. These last metrics, e.g., counting likes, also highlight methodological issues of whether evaluations are carried out in controlled experimental settings or if the evaluation occurs in the field in a more natural setting and environment. Earlier work by Carpendale [14] and Lam et al. [33] highlight these and many more methodological issues that are equally important for user experience evaluation as for usability and performance evaluation.

3.3 Moving Forward

The previous subsections highlight some of the complex issues and challenges that are resident for this topic. Others surely exist as well, some relating back to the central questions about the importance of these experience-driven metrics. For example, a visualization can be memorable for many different reasons, including one such as being exceptionally bad. It is important not to group together all evaluations and metrics into very coarse categorizations. Surely, designers do not want their visualizations to be highly memorable because of their poor quality. We envision that user experience metrics should be assessed in coordination with other more traditional performance-driven measures in order to identify worthwhile and beneficial experience characteristics.

Another limitation of most of the studies we reviewed in this article is that they involve static visualizations. The importance of interaction in visualization is clear [60, 47, 18] and thus to focus only on static visuals simply appears to be too limiting. In fact, the interactive aspects of some visualizations (e.g., the earlier mentioned Baby Names Wizard) often strongly contribute to their high level of engagement and enjoyability. Of course, static visuals are easier to assess in controlled experiments—an interactive visualization allows varied exploration paths and hence comparisons between participants may be less reliable.

One interesting avenue for continued research is to understand the role that user interaction plays in making visualizations more engaging or enjoyable. Does adding a certain set of user interactions (e.g., zooming and panning, or linking and brushing) make a visualization more engaging? Does adding affordances to visualizations to help explain their functionality make them more appealing to users? Can one measure the contribution of user interaction versus visual representation to enjoyment and engagement? Are touch-based visualizations (e.g., tablet-based visualizations) more enjoyable and engaging than visualizations on desktops? We anticipate that user interaction plays a crucial role in making a visualization more engaging or enjoyable, but this remains to be formally studied.

Another future research direction is to investigate the impact of display size in which visualizations are shown on user experience. Do visualizations on large displays increase user engagement and enjoyment?

Additionally, it is absolutely crucial to consider the ultimate purpose or objective of a visualization when evaluating its user experience attributes. It is clear that many of the criticisms launched against assessing user experience metrics, such as those by Few [20, 21], are posed in light of visualizations used for analytical, exploratory purposes. This application is often considered first when reviewing visualizations. Only recently has the value of using visualizations for presentational and narrative purposes arisen in the visualization research community [53, 27, 32], even if this application is common outside research settings. We suspect that the user experience goals and metrics reviewed in this article play a more important role on visualizations used for narrative and storytelling purposes. Engagement, fun, and memorability are all well-accepted, desirable characteristics of successful presentations. Talented speakers deliver presentations high in these characteristics.

Furthermore, the entire context of use of a visualization definitely plays a strong role in the importance of these experience-related goals. For business- and work-related use where people interact with a visualization daily, a visualization's ability to effectively represent data and its information density likely will be more important than memorability and fun. Conversely, for entertainment contexts such as pass-by browsing of websites, billboards, or commercials, where catching a person's attention is vital, then user experience characteristics rise in importance.

Finally, the three main user experience goals reviewed in this article clearly are not the only experience-related goals that researchers might examine. Other characteristics of visualizations, such as their ability to be entertaining, provocative, surprising, motivating, and perhaps even exciting, all could be important toward a particular visualization being considered "successful." We chose to examine the three attributes discussed in this article because user studies exploring those attributes already had been published. Little, if any, research has explored other experience-related characteristics, so this is definitely an area for future opportunities and new work.

4. CONCLUSION

This paper discusses goals, metrics, and methods for evaluating data visualization techniques beyond traditional usability goals, such as efficiency and effectiveness. Performance-focused metrics such as time and accuracy are applicable in specific tasks and usage scenarios. Alternatively, methods focused on user experience goals are less frequently found in the visualization literature. This paper describes three categories of user experience goals: memorability, engagement, and enjoyment (or fun). Our discussion summarizes the study methodologies and metrics used to evaluate visualizations based on these three user experience goals. We summarize each of these by observing trends in current approaches, as well as highlight potential challenges and opportunities going forward. Our aim is to encourage visualization researchers to consider these user experience focused evaluation methodologies, where appropriate. Additionally, the specific challenges presented in this paper can lead to important research in evaluation methodologies for data visualization.

5. REFERENCES

[1] Baby Name Wizard - Name Voyager. http://www.babynamewizard.com/voyager.

[2] S. Attfield, G. Kazai, M. Lalmas, and B. Piwowarski. Towards a science of user engagement (position paper). In *WSDM workshop on user modelling for Web applications*, Feb. 2011.

[3] D. Baldauf, E. Burgard, and M. Wittmann. Time perception as a workload measure in simulated car driving. *Applied ergonomics*, 40(5):929–935, 2009.

[4] C. M. Barnum and L. A. Palmer. More than a feeling: understanding the desirability factor in user experience. In *CHI'10 Extended Abstracts on Human Factors in Computing Systems*, pages 4703–4716. ACM, 2010.

[5] S. Bateman, R. L. Mandryk, C. Gutwin, A. Genest, D. McDine, and C. Brooks. Useful junk?: The effects of visual embellishment on comprehension and memorability of charts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 2573–2582, New York, NY, USA, 2010. ACM.

[6] J. Benedek and T. Miner. Measuring desirability: New methods for evaluating desirability in a usability lab setting. In *Proceedings of Usability Professionals Association*, pages 8–12, July 2002.

[7] R. Borgo, A. Abdul-Rahman, F. Mohamed, P. W. Grant, I. Reppa, L. Floridi, and M. Chen. An empirical study on using visual embellishments in visualization. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2759–2768, 2012.

[8] M. Borkin, Z. Bylinskii, N. Kim, C. Bainbridge, C. Yeh, D. Borkin, H. Pfister, and A. Oliva. Beyond memorability: Visualization recognition and recall. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):519–528, Jan 2016.

[9] M. Borkin, A. Vo, Z. Bylinskii, P. Isola, S. Sunkavalli, A. Oliva, H. Pfister, et al. What makes a visualization memorable? *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2306–2315, 2013.

[10] J. Boy, F. Detienne, and J.-D. Fekete. Storytelling in information visualizations: Does it engage users to explore data? In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 1449–1458, New York, NY, USA, 2015. ACM.

[11] P. B. Brandtzæg, A. Følstad, and J. Heim. Enjoyment: lessons from karasek. In *Funology*, pages 55–65. Springer, 2005.

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

[13] J. Brown, V. Lewis, and A. Monk. Memorability, word frequency and negative recognition. *The Quarterly Journal of Experimental Psychology*, 29(3):461–473, 1977.

[14] S. Carpendale. Evaluating information visualizations. In A. Kerren, J. T. Stasko, J.-D. Fekete, and C. North, editors, *Information Visualization*, pages 19–45. Springer-Verlag, Berlin, Heidelberg, 2008.

[15] M. Csikszentmihalyi. *Flow: The Psychology of Optimal Experience*. Harper Perennia, New York, 1990.

[16] W. A. Davis. A causal theory of enjoyment. *Mind*, 91(362):240–256, 1982.

[17] A. Dix. *Human-computer interaction*. Springer, 2009.

[18] N. Elmqvist, A. Vande Moere, H.-C. Jetter, D. Cernea, H. Reiterer, and T. J. Jankun-Kelly. Fluid interaction for information visualization. *Information Visualization*, 10(4):327–340, Oct. 2011.

[19] X. Feng, S. Chan, J. Brzezinski, and C. Nair. Measuring enjoyment of computer game play. In *Proceedings of AMCIS 2008*, 2008.

[20] S. Few. *Review of the Research Study "What Makes a Visualization Memorable?"*, 2013 (accessed February 3, 2016). https://www.perceptualedge.com/blog/?p=1770.

[21] S. Few. *Information Visualization Research as Pseudo-Science*, 2015 (accessed February 3, 2016). https://www.perceptualedge.com/articles/visual_business_intelligence/infovis_research_as_pseudo-science.pdf.

[22] B. Fredrickson. What good are positive emotions? *Review of General Psychology*, 3:300–3019, 1998.

[23] F.-L. Fu, R.-C. Su, and S.-C. Yu. Egameflow: A scale to measure learners' enjoyment of e-learning games. *Computers & Education*, 52(1):101–112, Jan. 2009.

[24] S. Haroz, R. Kosara, and S. L. Franconeri. Isotype visualization: Working memory, performance, and engagement with pictographs. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 1191–1200, New York, NY, USA, 2015. ACM.

[25] L. Harrison, D. Skau, S. Franconeri, A. Lu, and R. Chang. Influencing visual judgment through affective priming. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 2949–2958, New York, NY, USA, 2013. ACM.

[26] J. Heer, F. van Ham, S. Carpendale, C. Weaver, and P. Isenberg. Creation and collaboration: Engaging new audiences for information visualization. In A. Kerren, J. T. Stasko, J.-D. Fekete, and C. North, editors, *Information Visualization*, pages 92–133. Springer-Verlag, Berlin, Heidelberg, 2008.

[27] J. Hullman and N. Diakopoulos. Visualization rhetoric: Framing effects in narrative visualization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2231–2240, Dec. 2011.

[28] A. Isen. Positive affect facilitates creative problem solving. *Journal of Personality and Social Psychology*, 52(6):1122–1131, 1987.

[29] R. Jianu, A. Rusu, Y. Hu, and D. Taggart. How to display group information on node-link diagrams: an evaluation. *IEEE Transactions on Visualization and Computer Graphics*, 20(11):1530–1541, 2014.

[30] B. Jones. *When Memorability Matters: Another Practitioner's View*, 2015 (accessed February 3, 2016). http://dataremixed.com/2015/12/when-memorability-matters-another-practitioners-view/.

[31] R. Kosara. *Memorability, Science, and The Value of Thinking Outside the Box*, 2015 (accessed February 3, 2016). https://eagereyes.org/blog/2015/memorability-science-and-the-value-of-thinking-outside-the-box.

[32] R. Kosara and J. Mackinlay. Storytelling: The Next Step for Visualization. *Computer*, 46(5):44–50, May 2013.

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

[34] H. Li and N. Moacdieh. Is "chart junk" useful? an extended examination of visual embellishment. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 58, pages 1516–1520. SAGE Publications, 2014.

[35] N. Mahyar, S.-H. Kim, and B. C. Kwon. Towards a taxonomy for evaluating user engagement in information visualization. *Workshop on Personal Visualization: Exploring Everyday Life*, 2015.

[36] T. W. Malone. Toward a theory of intrinsically motivating instruction. *Cognitive science*, 5(4):333–369, 1981.

[37] T. W. Malone. Heuristics for designing enjoyable user interfaces: Lessons from computer games. In *Proceedings of the 1982 Conference on Human Factors in Computing Systems, CHI '82*, pages 63–68, New York, NY, USA, 1982. ACM.

[38] K. Marriott, H. Purchase, M. Wybrow, and C. Goncu. Memorability of visual features in network diagrams. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2477–2485, 2012.

[39] T. Merčun. Evaluation of information visualization techniques: Analysing user experience with reaction cards. In *Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization*, BELIV '14, pages 103–109, New York, NY, USA, 2014. ACM.

[40] T. Munzner. A nested model for visualization design and validation. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):921–928, 2009.

[41] D. A. Norman. *Emotional design: Why we love (or hate) everyday things*. Basic books, 2004.

[42] D. A. Norman. *The design of everyday things: Revised and expanded edition*. Basic books, 2013.

[43] H. L. O'Brien and E. G. Toms. What is user engagement? a conceptual framework for defining user engagement with technology. *Journal of the American Society for Information Science and Technology*, 59(6):938–955, 2008.

[44] H. L. O'Brien and E. G. Toms. The development and evaluation of a survey to measure user engagement. *Journal of the American Society for Information Science and Technology*, 61(1):50–69, 2010.

[45] F. Pachet and A. R. Addessi. When children reflect on their own playing style: Experiments with continuator and children. *Comput. Entertain.*, 2(1):14–14, 2004.

[46] R. Parncutt and G. E. McPherson. *The Science & Psychology of Music Performance: Creative Strategies for Teaching and Learning Book*. Oxford University Press, 2009.

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

[48] K. Rathunde and M. Csikszentmihalyi. Middle school students's motivation and quality of experience: A comparison of montessori and traditional school environments. *American Journal of Education*, 111:341–371, 2005.

[49] J. C. Read, S. MacFarlane, and C. Casey. Endurability, engagement and expectations: Measuring children's fun. In *Interaction design and children*, volume 2, pages 1–23. Shaker Publishing Eindhoven, 2002.

[50] B. Saket, C. Scheidegger, and S. Kobourov. Towards understanding enjoyment and flow in information visualization (short paper). In *EuroVis 2015 Conference*, 2015.

[51] B. Saket, C. Scheidegger, and S. G. Kobourov. Comparing node-link and node-link-group visualizations from an enjoyment perspective. *Computer Graphics Forum*, 35(3):41–50, 2016.

[52] B. Saket, C. Scheidegger, S. G. Kobourov, and K. Börner. Map-based visualizations increase recall accuracy of data. *Computer Graphics Forum*, 34(3):441–450, 2015.

[53] E. Segel and J. Heer. Narrative Visualization: Telling Stories with Data. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1139 –1148, Dec. 2010.

[54] H. Sharp, Y. Rogers, and J. Preece. *Interaction Design: Beyond Human Computer Interaction*. John Wiley & Sons, 2007.

[55] B. Shneiderman. Designing for fun: how can we design user interfaces to be more fun? *interactions*, 11(5):48–50, 2004.

[56] P. Sweetser and P. Wyeth. Gameflow: A model for evaluating player enjoyment in games. *Computers in Entertainment*, 3(3),

July 2005.

[57] M. Vass, J. M. Carroll, and C. A. Shaffer. Supporting creativity in problem solving environments. In *Proceedings of the 4th Conference on Creativity & Cognition*, pages 31–37, New York, NY, USA, 2002. ACM.

[58] M. Wattenberg and J. Kriss. Designing for social data analysis. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):549–557, July 2006.

[59] L. Xie, A. N. Antle, and N. Motamedi. Are tangibles more fun?: Comparing children's enjoyment and engagement using physical, graphical and tangible user interfaces. In *Proceedings of the 2nd international conference on Tangible and embedded interaction*, pages 191–198. ACM, 2008.

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