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

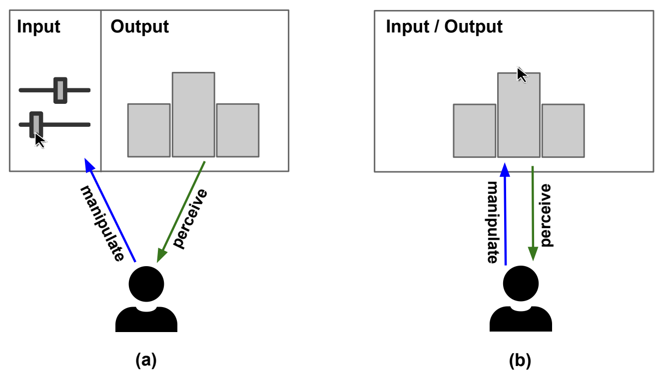


Fig. 1. Two different forms of interaction in many visualization systems. In order to interact with the visualization, users are required to either manipulate the external components in a separate panel (a) or directly manipulate the visual elements (b).

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 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.

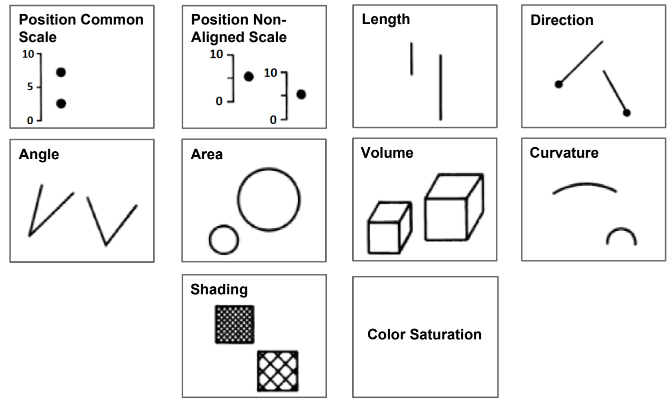


Fig. 3. Elementary graphical encodings studied by Cleveland and McGill [8] (these images were recreated and adapted based on [8]).

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.

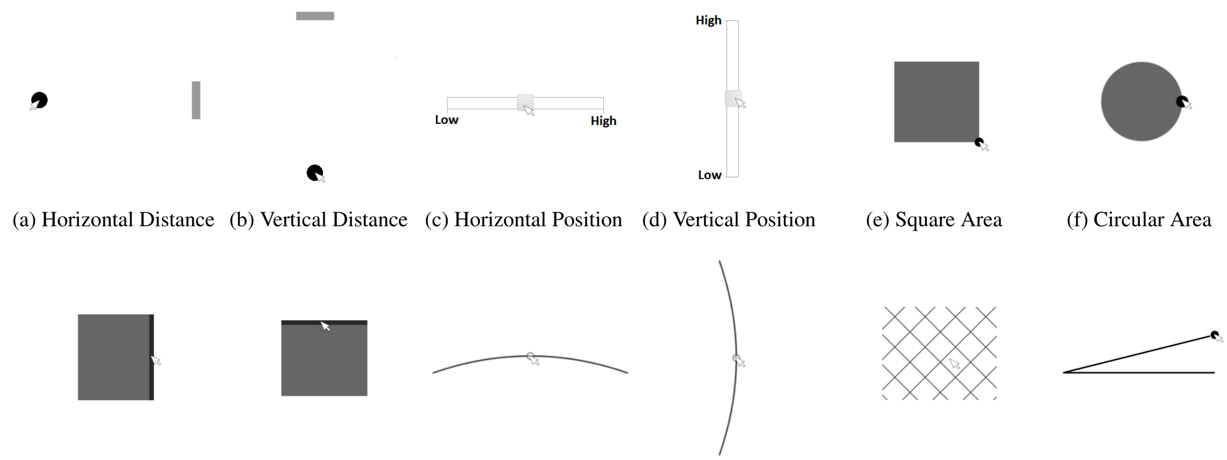
In an attempt to support more familiar and natural methods of user interaction, we chose to run the study as an online experiment so participants could use the setups and environments familiar to them *(e.g., their own machines with their own familiar input configuration)*. Previous work [17], [34] has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

$$Error = \frac{|ResponseValue - ExpectedValue|}{ExpectedValue} \times 100$$

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/

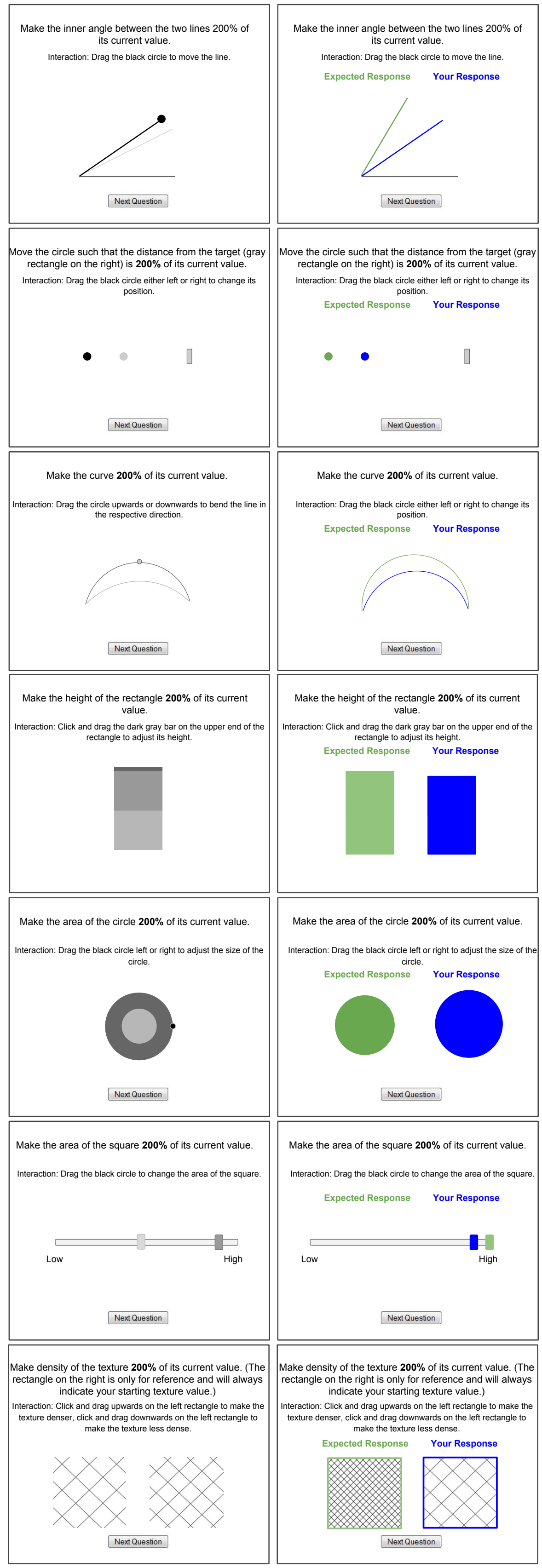
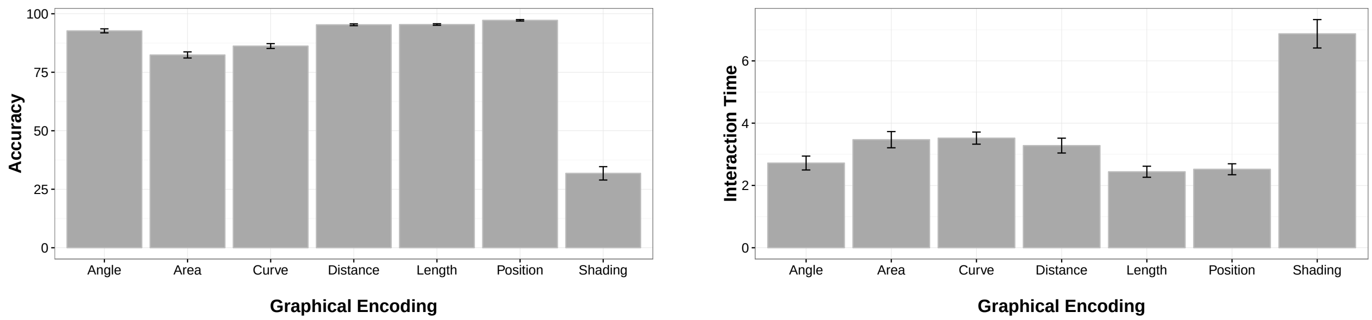


Fig. 5. Each row shows two screenshots from trials in the training phase. The left side shows the initial representation with instructions (all a 200% increase adjustment in this image), and the right image shows the interface after each trial during the training session, where participants were shown 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 ($F_{(1.7,58.5)} = 401.5, p < 0.001$, η_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 ($F_{(2.9,100.0)} = 95.2, p < 0.001$, η_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 (⋆).

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

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.

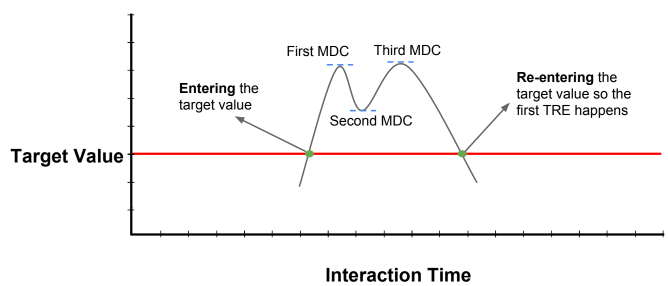


Fig. 7. This indicates part of a line chart used to visualize the interaction log for a particular user. This part of the interaction log enters the target value, leaves, and re-enters once. In this case, there is one target re-entry (TRE), and three movement direction changes (MDC).

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

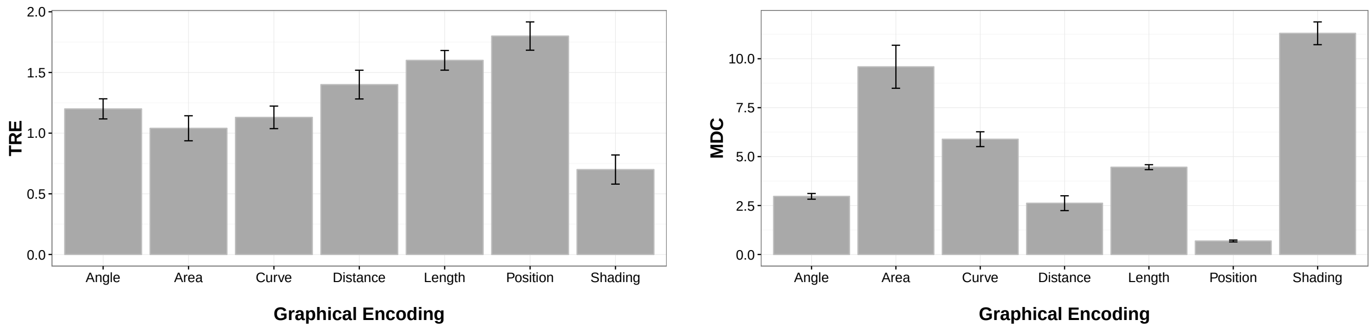


Fig. 8. Means and standard deviations of TRE and MDC for each *interactive graphical encoding*. The units in this table are "mean count per participant" for TRE and MDC. Error bars represent standard error.

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.

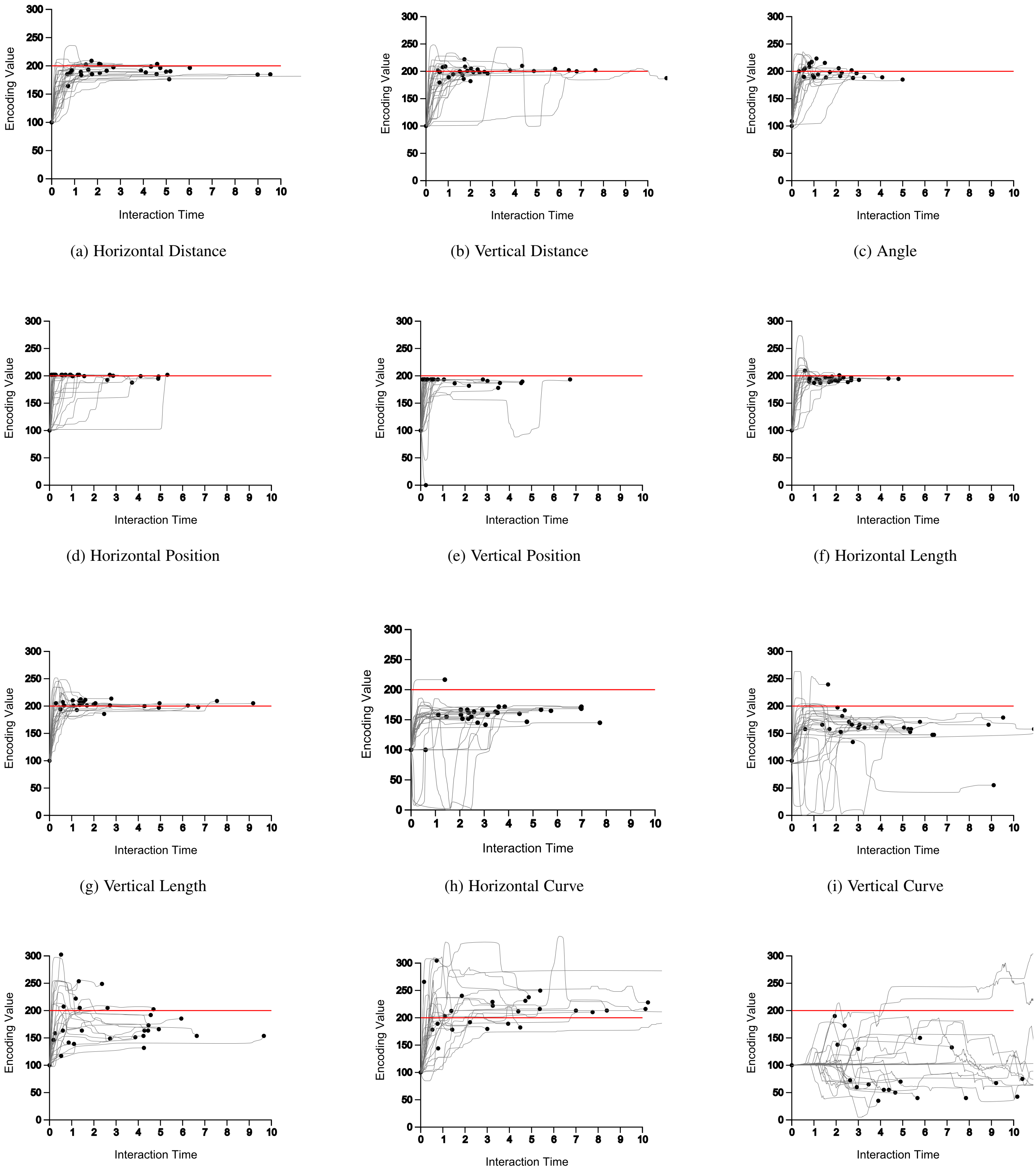
6.1 Incorporating the Interactive Graphical Encodings

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.