Eurographics Conference on Visualization (EuroVis) 2019
M. Gleicher, H. Leitte, and I. Viola
(Guest Editors)

Volume 38 (2019), Number 3

Investigating the Manual View Specification and Visualization by Demonstration Paradigms for Visualization Construction

Author 1 and Author 2

Affiliation

Abstract

Interactivity plays an important role in data visualization. Therefore, understanding how people create visualizations given different interaction paradigms provides empirical evidence to inform interaction design. We present a two-phase study comparing people's visualization construction processes using two visualization tools: one implementing the manual view specification paradigm (Polestar) and another implementing visualization by demonstration (VisExemplar). Findings of our study indicate that the choice of interaction paradigm influences the visualization construction in terms of: 1) the overall effectiveness, 2) how participants phrase their goals, and 3) their perceived control and engagement. Based on our findings, we discuss trade-offs and open challenges with these interaction paradigms.

1. Introduction

Visual representation and interaction are two main components of visualization tools [YKS07, Sed04]. The main focuses of visual representation are mapping data values to graphical representations and rendering them on a display in an effective manner. Interaction provides users the ability to change the parameters of the system to construct and change visual representations and interpret the resulting views [YKS07,DE98]. While the goal of most visualization tools is to accommodate visualization construction, they may use different *interaction paradigms*. In this paper, we define the term interaction paradigm in information visualization to refer to the process of how visualization construction is fostered in a tool.

How the visualization construction process is conducted often depends on the user interface design itself. For example, a commonly used interaction paradigm in most visualization tools is *manual view specification* (MVS) [WQM*17]. Tools implementing MVS often require users to manually specify the desired mappings through GUI operations on collections of visual properties and data attributes that are presented visually on control panels. For instance, to create a scatterplot, users must specify the visualization technique, then select data attributes to map onto the axes, and finally map any additional visual encodings to the desired attributes. While the exact user interface design of different tools may vary [GBTS13], the underlying MVS interaction paradigm is consistent. For example, tools such as Tableau [Tab18] and Polestar choose to implement MVS by dragging and dropping attributes onto shelves to set specifications, while others such as Spotfire [Ahl96] do so by dropdown menus. Regardless of the design choices, the users' responsibilities are still to specify visual properties, so that the system can generate the resulting view.

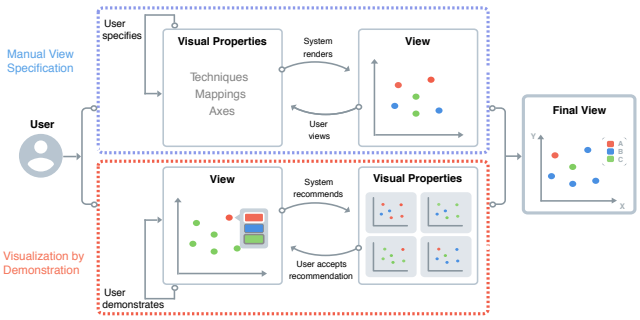

Figure 1: *Visualization construction process using MVS and VbD.*

Saket et al. recently proposed an alternative interaction paradigm for visualization construction called *visualization by demonstration* (VbD) [SKBE17]. This paradigm advocates for a different process of visualization construction. Instead of specifying mappings between data attributes and visual representations directly, VbD lets users demonstrate partial mappings or changes to the output (the visualization). From these given demonstrations, the system interprets user intentions, and recommends potential mappings and specifications. VbD is inspired by previous work that indicated 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 [HCT*14, AEN10, SIM99].

Despite the fact that both interaction paradigms enable users to construct visualizations through an iterative process, there are at least two main differences. The differences that went into design considerations of each of the paradigms could influence how people construct visualizations.

- MVS uses a different process compared to VbD. MVS requires users to specify visualization techniques, mappings, and parameters. In contrast, VbD 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. Thus, while the result of both is a visualization and the corresponding specifications, users follow a different process to get there (see Figure 1).

- MVS 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, in VbD the interface is itself a visual representation. The user can act on the visual representation rather than external interface elements. While the majority of user interaction is on the visual representation, VbD 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: the user can demonstrate intentions by directly manipulating the encodings used in a visual representation.

Many visualization tools implement the MVS paradigm. These tools are successful in easing the processes of visualization construction [GTS10]. They allow users to interactively construct visualizations instead of using programming. However, when new interaction paradigms such as VbD are created, it raises a number of intriguing questions including: *How do interaction paradigms enable different visualization construction processes? How effective are each of these interaction paradigms for specific tasks?* Understanding the differences and trade-offs between various interaction paradigms and how they are used for specific tasks can help designers and developers make informed decisions about interaction design in visualization tools.

In this paper, we take a step towards gaining a better understanding of the trade-offs between these two interaction paradigms. We first conduct a controlled experiment to study the effectiveness (performance time and accuracy) of each paradigm in constructing visualizations. We then conduct 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 paradigm. The main contributions of this paper provide empirical evidence to show: 1) the effectiveness (performance time and accuracy) of each paradigm for different tasks; 2) how the underlying interaction paradigm used in each tool influences the visualization construction process; and 3) trade-offs between the two interaction paradigms.

2. Related Work

There are several visualization process models that explain different steps users follow to construct visualizations [CMS99, CR98, Car99]. One of the well-known models is Card et al's "reference model" [CMS99]. The reference model explains four steps that users often follow to construct visualizations: **Raw Data Transformation**, **Data Table Transformation**, **Visual Properties Specification**, and **View Rendering**. Chi and Riedl [CR98] proposed Data State Model of visualization processes that extends the reference model proposed by Card et al. by allowing for multiple pipelines. Another variation of Card et al.'s reference model was introduced by Carpendale [Car99] that adds **Presentation Transformation** step before the view rendering. Despite the minor differences between these models, most of these models place *visual properties specification* before *view rendering*. This implies the interaction paradigms that realize these models often ask users to assign data attributes to visual properties, then have systems render the specified views.

2.1. Manual View Specification (MVS)

The MVS paradigm also requires users to map data attributes to visual properties prior to rendering the view. Many existing interactive visualization tools implementing this paradigm such as MS Excel, Spotfire [Ahl96], Tableau [Tab18], and Polaris [SH00] require users to specify data mappings before rendering the view. This is how the visualization construction process is enabled at every step. For instance, to create a scatterplot, users must specify the point visualization technique, then select data attributes to map onto the x and y axes. The system then generates a scatterplot with all the specified characteristics.

However, user interface design in visualization tools embodying a specific paradigm can be implemented in various ways. Previously, Grammel et al. [GBTS13] categorized the user interface design in desktop-based visualization tools into six different groups. Among different interface designs identified by Grammel et al., four of them are the most relevant to our work: **template editors**, **shelf configuration**, **visual builder**, and **visual data flow**. Below, we describe each of these designs and discuss while the exact user interface design of many of the existing tools may vary, the underlying MVS paradigm is consistent.

Based on Grammerl et al. **template editor** is a user interface design that enables *"the user selects some data and then picks a predefined visual structure in which to represent it. The distinguishing criteria of this approach are the separation between the initial visualization selection steps and the refinement of the selected visualization"* [GBTS13]. For example tools such as Many Eyes [VWVH*07] and Excel implement the template editors design to enable users to construct visualizations. Template editor design requires users to specify the data attributes before selecting the predefined visual structure.

Tools that implement the **visual builder** design often consists of a palette containing visual element prototypes and empty canvas as an assembly area. The user interface design in such tools is very similar to graphics editor tools such as Sketch or Adobe Photoshop.These tools also require users to specify visualization properties prior to rendering the final view. However, these tools take a different approach from tools such as MS Excel. They first enable users to draw a customized visual glyph and put different visual elements together on the canvas. They then enable users to manually bind the visual glyph to a specific data attribute, effectively creating a mapping between data and customized graphical elements. Different visualization authoring tools such as Data Illustrator [LTW*18], Data-Driven Guides [KSL*17], Charticulator [RLB19], and iVisDesigner [RTY14] implement the visual builder design.

Visualization tools implementing the **data flow** design enable users to construct visualizations by connecting visual components through links. The final graph of connected components represents the dataflow and the final visual output. While this design has a long history [AT95], more recent visualization tools such as iVoLVER [MHN17] and DataMeadow [EST07] implement the data flow design.

Shelf configuration is a commonly used interface design that enables the user to specify the desired mappings through GUI operations on collections of visual properties and data attributes that are presented visually on control panels. Visualization tools might implement this design differently. For example, tools such as Tableau [Tab18] and Polestar [Pol16] choose to implement this design by dragging and dropping attributes onto shelves to set specifications, while others such as Spotfire [Ahl96] do so by dropdown menus. Regardless of the design choices, the users' responsibilities are still to specify visual properties through graphical widgets presented on the interface.

2.2. Visualization by Demonstration (VbD)

In general, demonstration-based interfaces offer users the ability to create visual demonstrations of intended results. They emphasize the ability for people to focus on resulting representations, from which systems infer lower-level specification that create these results.

For example, Saket et al. [SKBE17] presented "visualization by demonstration", which describes a series of design considerations for user interaction in visualization. Their paper discusses how visualization systems can infer lower-level visualization specifications from visual demonstrations provided by users. For example, color mappings can be created by coloring a set of items in a scatterplot, bar charts can be sorted by ordering a subset of bars, etc. They also presented VisExemplar, a visualization tool that implements the VbD paradigm [SKBE17].

The core concept of VbD extends prior work in related computing areas where demonstration-based approaches have been used. In computer programming, programming by demonstration [CH93] enables users to generate code by providing visual demonstrations of intended software functionality. An important aspect of these interfaces is incrementally iterating and improving the state of the system by continuing to demonstrate further changes or directly edit the produced code. Other domains that have successfully used the "by demonstration" paradigm include 3D drawing by demonstration [IH01], data cleaning by demonstration [LWN*09], and interactive database querying by demonstration [Zlo75].

A by-demonstration paradigm has also been used for people to perform a myriad of visual analytic tasks. For example, Kandel et al. showed how data wrangling and cleaning in spreadsheets can be done by demonstrating rules and filters [KPHH11]. Dimension reduction and clustering models can be guided by demonstrating group membership and relative similarity between data points [EFN12, KCPE16]. Additional tasks that have been supported with demonstration-based interfaces include temporal navigation [KC14], adjusting data grouping criteria [SSEW17], and others.

2.3. Sketch-based Visualization Construction

A related line of existing work within the visualization community has examined how sketching or drawing can be used as an approach to allow users to specify their intentions [LKS13, KKW*17, WLJ*12, CMvdP10]. For instance, SketchStory [LKS13] is a system that allows users sketch out an example icon and visualization axis. The system then interprets user interactions and completes the visualization with new visual properties. The sketching-based paradigm relies on digital ink and ink recognition. In addition, it relies on users exemplifying their intentions via drawing. However, there are commonalities with VbD in the sense that both allow users to specify their desired goal instead of manually parameterizing the visual properties. However, in our work we focus on conventional desktop settings, and do not focus on sketch-based interactions.

2.4. Previous Studies of Visualization Construction

Previous studies exist that have investigated the visualization construction process using different tools (e.g., [WPHC16, MHN17, GTS10, Hur14]). For example, Wu et al. [WPHC16] compared the bar chart construction process between MS Excel and a set of physical tiles. Their findings showed that the distribution of time spent and sequence of actions taken to construct visualizations are different depending on the tool. Unlike the study by Wu et al. [WPHC16], we investigate trade-offs of visualization construction in digital visualization tools. In another study, Mendez et al. [MHN17] compared how visualization novices construct visualizations and make design choices using either top-down or bottom-up visualization tools. Their findings indicated trade-offs between the two tools and considerations for designing better interactive visualizations. While we also explore the interactive visualization construction, we focus on investigating the trade-offs between two different interaction paradigms. In particular, we are interested in investigating how each of the two interaction paradigms influence the visualization construction's process.

3. Choice of Visualization Tools

To investigate trade-offs between the two interaction paradigms, we selected two tools that each embody one of the two interaction paradigms examined in this paper. The main experiment used two tools, VisExemplar [SKBE17] and Polestar [Pol16], which satisfied two requirements. One is that each had to clearly embody one of the two interaction paradigms examined in this paper. The other is that to have a fair comparison between these two paradigms, users had to be able to learn and use the system within the duration of the experiment (i.e., the tools should be simple). A video walked the participants through different features and interactions provided by the tool. Although a variety of commercial tools incorporate the MVS paradigm, we decided to use the less complicated visualization tool, Polestar, to control for external factors that might affect the study. The functionality of Polestar is tightly scoped and intentionally limited to control for these potential confounds. Also, Polestar has previously been used as a control condition in previous studies [WMA*16, WQM*17]. Below, we discuss and compare the two tools.

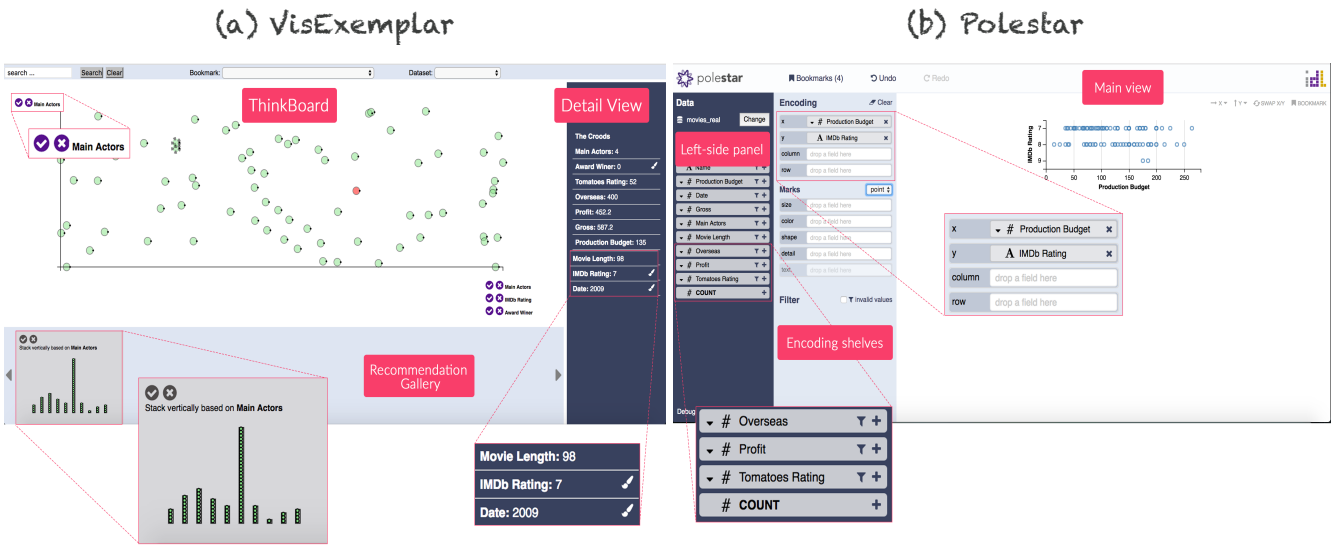

Figure 2: *(A screenshot of VisExemplar and Polestar.)*

VisExemplar[†] (shown in Figure 2-a) 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, VisExemplar enables users to directly manipulate the encodings used in a visual representation (e.g., users stacking data points in the shape of bars to convey their interest in a bar chart). In response to the provided demonstrations, VisExemplar recommends four categories of transformations: (1) change the current visualization technique, (2) define mappings between graphical encodings and data attributes, (3) assign data attributes to axes of a visualization technique, and (4) change the view specifications without changing the underlying technique. By accepting any of the recommended transformation, the system will change the corresponding view.

Polestar[‡] (shown in Figure 2-b) is a visualization tool that implements the MVS paradigm. The 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. The main activity of designing visualizations in Polestar consists of the assignment (through drag-and-drop), of a data attribute onto a shelf to establish a visual encoding. Users can also change properties of the data or the visual encoding variable (e.g., color or sort) via pop-up menus. The Main View of the Polestar shows a created visual representation. After each specification, the system updates the view accordingly.

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 a 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 VisExemplar and the others to work with Polestar. We first asked the participants to watch a tutorial video of the visualization tool. We then asked participants to perform eight trial tasks using the tool (e.g., assign the cylinder attribute to the x axis). Tasks for trial session were designed using the Cars dataset [HV81]. Once they completed the tasks, we asked them to work with the visualization tool for 20 minutes to explore the Cameras dataset [Dat15]. The participants were asked to verbalize their visualization construction process.

We captured two methods that the participants used to phrase their goals while constructing visualizations. In the first method, 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 mark property or axis (e.g., *"I want to assign price to the size of the circles."*) We call this a ***specific*** method. The second method is more abstract. Participants sometimes were unaware of some of the information required for completing their goals. 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"*, or even *"Cameras like these should be larger."* We call this an ***abstract*** method. The key difference between these two methods of phrasing goals/tasks 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 attributes and visual mappings, the specific method was used. Alternatively, when goals are formed on data items and semantic relationships between data items, abstract methods were used.

We initially decided to use the Cars [HV81] and Cameras [Dat15] datasets in our main study. However, it turned out that the partici-

[†] https://github.com/BahadorSaket/VbD
[‡] https://github.com/vega/polestar

pants were not familiar with all data attributes used in the Cameras dataset (or cameras in general), so they tended to not 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 [Dat15] 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 a 20 minute 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, we tested if the way the tasks are phrased affected user performance time and accuracy. We ran a pilot study with four additional participants. We designed 8 trial visualization construction tasks. For half of the tasks we provided all the information participants required to perform them (specific method). For the second half, we explained the tasks in a more abstract way (abstract method). We noticed that the way the tasks are phrased would affect users' performance. For example, in Polestar, participants were faster in performing the tasks that are phrased using the specific method. Thus, we decided to take into account ways that tasks are phrased as one of the factors in our main experiment.

5. Study Design

We conducted a two-phase study. In the first phase, we studied how well each interaction paradigm supports visualization construction in a more controlled setting. We measured the effectiveness (performance time and accuracy) of each paradigm for 16 tasks. In the second phase, we investigated how well visualization construction is supported by each paradigm in a more realistic scenario. 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. We provide all relevant materials (complete list of tasks, datasets, anonymized results, and detailed statistical analysis) for this study in supplemental materials.

Participants and Setting
We recruited 16 participants (9 female and 7 male), between 21 and 32 years old. The participants were undergraduate and graduate science and engineering students. None of them had participated in the pilot. All participants reported to be familiar with reading and creating visualizations using existing tools such as MS Excel (16), D3.js (2), and Tableau (1). During the entire study, participants used a computer with 13 inch screen. Two of the participants took the undergraduate level information visualization course taught in our university. The study took about 1 hour to complete and participants were compensated with a $10 Amazon gift card.

5.1. Phase 1: Controlled Experiment

In this phase, we examined the effectiveness of each interaction paradigm for creating visualizations in a controlled setting. We used a mixed design with the tool as between-subjects factor. 16 subjects participated in our study and were randomly assigned to one of the visualization tools (8 participants per visualization tool). Each participant worked with just one of the visualization tools.

5.1.1. Tasks

To select tasks for our study, we first interacted with both VisExemplar and Polestar exploring different ways in which they support visualization construction. This resulted in a list of 25 visualization construction tasks (e.g., assign a data attribute to size and color of data points). We also reviewed taxonomies of tasks commonly used for interactive visualization construction (e.g., [Shn96, YKS07, RCDD13, DE98]). Considering our experiences with these tools and our knowledge from these taxonomies, we then assigned 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 mark properties:** This category of task requires users to map a data attribute to a mark property.
- **Switching between visualization techniques:** This type of task requires users to change from one visualization technique to a different visualization technique.
- **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 phrasing methods $\times$ 2 trials). Each participant performed all 16 tasks using one of the visualization tools. The phrasings of tasks (abstract and specific) are based on how participants verbalized their goals during the think aloud protocol of the pilot study. Table 1 shows four category of tasks used in our study. For each category, we included an equivalent pairs of tasks that each phrased using a different method (abstract and specific).

5.1.2. Hypotheses

Since the two interaction paradigms have their own characteristics, we expect that each paradigm will have its advantages and disadvantages. Based on our pilot studies and experience with both paradigms, we considered the following hypotheses for our study:

- **H1:** We hypothesize that using Polestar, participants map data attributes to axes and switch from one visualization technique to another significantly faster and more accurately than VisExemplar. However, mapping visualization encodings to data attributes and reconfiguring visualizations would be significantly faster and more accurate using VisExemplar.
- **H2:** We expect Polestar to have better performance (shorter time, higher accuracy) for tasks phrases using the specific method and VisExemplar to have better performance for those phrased abstractly.

5.1.3. Procedure

Training. Before starting the main experiment, participants were briefed about the purpose of the study. At this stage, the participants

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.*

Task Type	Starting Point	Specific Method	Abstract Method
Mapping data attributes to the axes	A visualization (either bar chart or scatterplot). Data attributes assigned to the axes are different from those mentioned in the task.	Change the scatterplot so that it has wheelbase as the x axis.	Change the representation so that circles are horizontally positioned by price.
Mapping data attributes to mark properties	A visualization (either bar chart or scatterplot). mark properties used in the visualization are not mapped to any data attribute.	Change the scatterplot in a way that color is mapped to engine size of the cars.	Change the representation so that cars with the same number of cylinders have the same color.
Switching between visualization techniques	A visualization technique other than the one that participants are supposed to construct. Attributes assigned to the axes are different from those mentioned in the task.	Switch from the given bar chart to a scatterplot where the x axis is price and y axis is wheelbase.	Modify the given representation such that cars are horizontally positioned by horsepower and vertically positioned by engine size.
Reconfiguring the visualization	A bar chart visualization.	Sort the given bar chart in ascending order.	Order the bars from left to right. Shortest bar is on the left and the tallest bar is on the right.

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. 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 phrasing methods $\times$ 1 trial). The participants were not allowed to move to the next training question unless they answered the question correctly. Once comfortable with using the visualization tool, participants were instructed to take a short break and move to the main study.

Main Study. In this phase, each participant performed 16 visualization construction tasks: 4 types of tasks $\times$ 2 phrasing methods $\times$ 2 trials. All tasks were printed on a sheet of paper. Each time the interviewer selected a task randomly and asked the participants to perform the task as fast and accurately as possible. Before performing each task, participants were given a visualization as a starting point. This way we made sure that all the participants performed each task starting from the same visualization. We measured participants performance time and accuracy. To design tasks for this phase, we used the Cars [HV81] dataset. The Cars dataset [HV81] provides details for 407 new cars and trucks for the year 2004. This dataset contains 18 data attributes describing each car.

5.1.4. Data Analysis

To address our first hypothesis (**H1**), we tested how the different tasks were performed using each interaction paradigm in terms of time. 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 paradigms, so we excluded accuracy from our analysis. We first calculated separate mean performance time for all trials. For each participant, we averaged outcome values of trials for each type of task. 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 paradigms (between-subjects factor). The main effect of interaction paradigm indicates which paradigm produces the best performance, regardless of the task. The task $\times$ paradigm interaction indicated whether a particular paradigm 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 phrasing methods of constructing visualizations (within-subjects factor) using two interaction paradigms (between-subjects factor). In particular, we were interested in interaction between the two different phrasing methods for goals and interaction paradigms (Phrasing methods $\times$ interaction paradigm). Investigating the interaction between phrasing methods and interaction paradigms indicated whether a particular paradigm works better with a specific method of phrasing goals (abstract and specific). Thus, we averaged outcome values of trials for each participant.

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.

5.1.5. Results

We found a significant effect of performance time for interaction paradigm ($F(1,14) = 19.6, p < 0.05$) with a slightly large effect size ($\eta_p^2 = 0.63$). Overall, average task completion time across all tasks showed that Polestar was three seconds faster than VisExemplar. Tools such as Polestar that implement MVS are fast and accurate since they enable rapid and formal/exact specification of the visual properties by incorporating a set of consistent user interface elements. We also found a significant interaction between paradigms 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 Polestar compared to VisExemplar. We also found that Polestar was significantly faster than VisExemplar in switching from one visualization to another. However, the participants were significantly faster in mapping data attributes to encodings and reconfiguring visualizations using VisExemplar that implements VbD. Our results partially confirms our first hypothesis (**H1**). See Figure for more details 3.

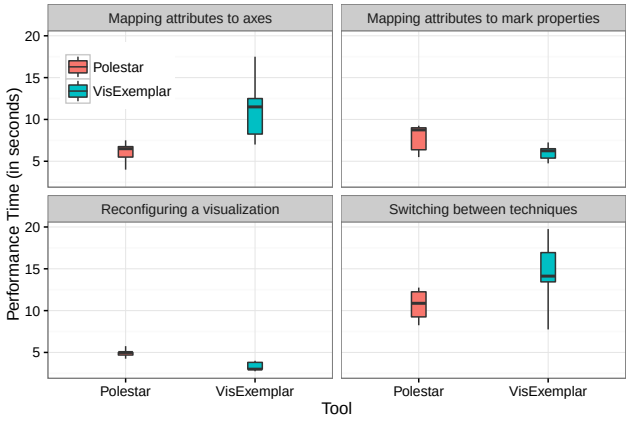

Figure 3: *Participants average performance time for each type of task using Polestar and VisExemplar.*

We also found a significant interaction between paradigms and phrasing methods for performance time ($F(1,14) = 34.5, p < 0.05$) with a large effect size ($\eta_p^2 = 0.71$). The participants performed tasks significantly faster using abstract method than specific method in VisExemplar ($p < 0.001$). Unlike VbD, the participants were significantly faster in performing tasks using specific method than abstract method in Polestar that implements MVS ($p < 0.05$). This confirms our second hypothesis (**H2**). See Figure 4.

5.2. Phase 2: Open-ended Exploration

In this phase, we conducted a think aloud exploratory observational study to understand how the participants use each interaction paradigm to construct visualizations in a more realistic scenario.

5.2.1. Procedure

Main Study. In this phase, the participants were asked to explore the Movies dataset [Dat15] and look for interesting findings about

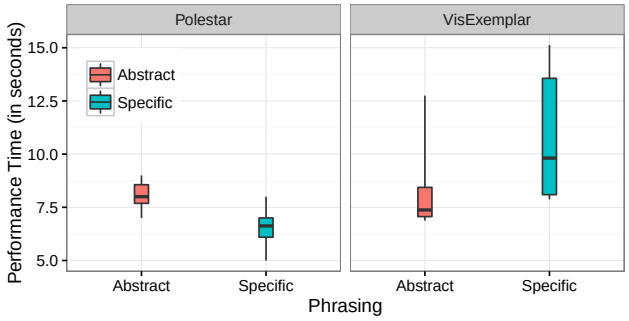

Figure 4: *Participants average performance time for each phrasing method using Polestar and VisExemplar.*

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. 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, sometimes it was necessary to remind the participants that this is a think-aloud study and they need to verbalize their thoughts.

Follow-up Interview. We asked participants what they liked and disliked about the interaction paradigm. This was to allow the participants to convey their feedback and ideas and in order to solicit potentially unexpected insights.

5.2.2. Data Collection and Analysis

To analyze differences between the VbD and MVS conditions, we gathered several types of data. At the beginning of the training session, we used questionnaires to collect participant demographic and background information. During the main study, we took written notes of participants' interaction processes with the tools. We also screen- and audio-recorded the whole study.

To analyze the video and interview material, we followed guidelines provided by Creswell [Cre02, p. 236] for analyzing qualitative data. We first fully transcribed data from the interviews. The coder (first author) then read the transcribed materials to obtain a general sense of the data and thinking about organization of the data. After reading the data, the first author identified the meaningful text segments and assigned a code word or phrase that accurately describes the meaning of the text segment. The coding process was an iterative process with three passes by a single coder in which the coder developed and refined the codes. During the coding phase, we mainly focused on processes of the participants in terms of **usage** (what types of visualization specifications were usually created using each interaction paradigm? What are the patterns in the use of specific functionality?) and **barriers** (when and how difficulties happened while working with each paradigm?) For example, our codes included phrases such as "changing color", "changing size"

and "stacking data points". Finally, we identified and aggregated similar codes that frequently occurred to form themes. Similar to codes we assigned labels to themes. For example, we aggregated the "changing size" and "changing color" codes to create "mapping a data attribute to a visual property" theme. Finally, we identified frequently occurring codes and themes to form higher-level descriptions and to discuss our findings.

5.2.3. Results and Discussion

In this section, we categorize and discuss the findings of our study. We also reflect on the results of our work more broadly with respect to interaction for visualizations.

Interaction Behavior for Visualization Construction Tasks
We divided types of operations the participants performed during the entire visualization construction process into four categories of tasks discussed earlier (Section 5.1).

Mapping Data Attributes to Axes
Participants tended to map more data attributes to axes using Polestar (see Table 2). Four of the eight participants who worked with Polestar stated that the fast speed of the tool in mapping data attributes to axes have contributed to this advantage. For example, one of the participants mentioned *"It is easy to drag and drop data attributes in this tool [Polestar]. I can quickly change axes."* On the other hand, in VisExemplar, five of the participants expressed difficulties in mapping data attributes to axes. To map a data attribute to an axis using VisExemplar, the participants 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. For example, one participant expressed how a large amount of effort was required for him to map a data attribute to an axis: *"You know it is hard to drag the points and track their values, [...] maybe you could somehow highlight the values [data attribute values] while moving the points to decrease users' cognitive load."*

Another challenge that three of the participants encountered while using VisExemplar was the accuracy of the data attributes suggested to be mapped to the axes. After providing visual demonstrations, the system searches for data attributes to recommend for mapping to the axes based on the user interaction. 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 is not among those recognized to be the most related ones by the system. In such cases, users have to provide more demonstrations to help the system to interpret their intentions better. However, providing more demonstrations can be frustrating in some cases. One of the participants mentioned her concern by saying: *"The recommendations on the axes don't always make sense to me. When I have an idea in mind like let me see how these [data attributes] compare, then when I don't see it in the options, I am kinda thrown off because at that point I am kinda doubting whether the way that I am thinking about it is wrong or whether I am doing something wrong with the system."*

Going forward, we envision multiple ways to avoid interaction challenges in tools implementing VbD. One way is to incorporate feedforward [VLvdHC13] and suggested interactivity [BEDF16] to improve the efficiency of tools implementing VbD. For tasks such as "mapping data attributes to the axes" this could help by showing what sequences of operations a user should execute to provide a demonstration. Another way is to incorporate more advanced interactions such as lasso selection to improve users' speed in providing multi-point demonstrations. For example, imagine assigning a data attribute to an axis. Users could do a lasso selection and move multiple points together rather than moving each point individually.

Moreover, while using VisExemplar, participants sometimes were unclear why the system suggested specific recommendations. In such cases, the participants found it difficult to map the recommended options to their interaction with the visualization. Tools implementing VbD suggest potential options based on the given demonstrations. However, there might be cases that the systems do not recommend options expected by the users, and it might not be apparent to users why those recommendations are presented to them. Going forward, we suggest the systems implementing VbD to design methods to explain the reasoning behind recommendations.

Mapping Data Attributes to mark properties
The participants mapped more data attributes to mark properties using VisExemplar (see Table 2). To map a data attribute to size or color using VisExemplar, users could manipulate characteristics of a corresponding encoding in the visual representation. 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. During data analysis, we noted multiple interesting patterns.

In VisExemplar, participants found the process of recommending a subset of appropriate data attributes for mapping to color or size very interesting and helpful. For instance, one participant mentioned that: *"[...] coloring points was fast though. I can color one point and the system suggests a small set of attributes."* On the other hand, one of the participants who used Polestar stated: *"every time I need to skim through attributes on this panel [the panel showing data attributes], pick one, and drag it. It becomes hard to skim through all attributes if we have many of them [data attributes]."* Moreover, 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 regarding VisExemplar, one of the participants mentioned: *"... let's color one and look at recommendations [participant hovered on the recommended attributes to preview the results and explain their findings]."*

We also noted that the participants felt more control over the tool when they were mapping data attributes to mark properties using VisExemplar. One participant expressed his feeling of having control by saying: *"I like that I can color it here [coloring the glyph], I feel like I have control over the circles [data points]"*. This is potentially because VbD advocates for increasing the level of "interaction directness" [BL00] by enabling the users to demonstrate their goals using direct manipulation of graphical encodings used in visual representations. As previous work also indicated the level of interaction directness with the visual representation contributes towards increasing the sense of control and personal agency in the participants [KPV*18]. However, we observed that this level of directness sometimes led the participants to the point that they forget their primary task. For example, one of the participants was so in-

Table 2: *Total and average number of times that participants performed each type of task using VisExemplar and Polestar.*

Task Type		VisExemplar	Polestar
Mapping attributes to axes	Total	53	108
	Avg	7.5	15.4
Mapping attributes to encodings	Total	55	26
	Avg	3.9	1.9
Switching between techniques	Total	12	25
	Avg	1.4	3.5
Reconfiguring a visualization	Total	7	9
	Avg	0.8	1

volved in the process of dragging the points that at one point 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 [PSM07], in that they are designed to be helpful for a particular set of analytic tasks. Advantages and disadvantages in increasing the directness of interaction paradigms then raise a question — What is the right level of interaction directness that should be given to the users of the visualization tools?

Reconfiguring a Visualization

We did not find a large difference in the number of times that the participants reconfigured the visualizations using each tool (see Table 2). In fact, the results of our first phase also indicate that both tools were quite fast in reconfiguring visualizations. However, we noted four of the participants found sorting the bar chart using VisExemplar intuitive and fun. For instance, one of the participants mentioned: *"[...] the sorting was intuitive."* VisExemplar enables users to demonstrate their interests in sorting the bar chart by dragging the shortest/tallest bar to the extreme left/right. The system then recommends sorting the bar chart. We believe interaction directness in tools implementing VbD affects the feeling of engagement and involvement during visualization construction process [Per14, SSEW17]. Another participant 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."* One interesting avenue of research is to investigate the effectiveness of these interaction paradigms 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 [PSM07].

Switching Between Visualization Techniques

The participants switched between visualization techniques more often while using Polestar (see Table 2). The ability to quickly change from one type of technique to another could contribute to this advantage. In particular, seven of the participants found it quite difficult to switch from a scatterplot to a bar chart using VisExemplar. To switch from a scatterplot to a barchart using VisExemplar, the participants had to stack two or more data points vertically. The system then recommended a set of barcharts based on similarity of the data points. Participants found this type of tasks difficult to demonstrate. For example, one participant expressed the difficulty of switching from a scatterplot to a barchart by saying: *"it was a bit awkward and hard for me to stack the points to create a barchart."* Without intuitive and easy visual analogies to demonstrate an intended task or goal, the effectiveness of tools implementing VbD may suffer, and other interaction paradigms such as MVS may be better suited.

No-need-to-think vs. Need-to-think

To construct different visualizations in Polestar, the participants mapped data attributes to different visual properties through GUI operations visually presented via the control panel. While participants experienced fast visualization construction using Polestar, they usually did not reflect much on the meaning and potential impact of their exploration. For example, one of the participants stated: *"I did not need to think of how I want to create visualizations. When I started, I did not have any design in my head. So, I kept creating different designs until I found the one [visualization] that looked interesting."* In such cases participants tried mapping a variety of data attributes to different visual properties until they created a visualization that they liked. Previous work [MHN17] also confirms this notion of "no-need-to-think" when working with Tableau (another tool implementing MVS). This notion enables users to quickly explore a variety of visualization alternatives and explore their data.

In contrast, VisExemplar advocates for the notion of "need-to-think". Users need to think about the visual output or how data mappings should look before starting the demonstration process. With VisExemplar, five participants mentioned that they had to think about how they want their visual outputs to look and only then started providing visual demonstrations of incremental changes to the visual representation. One of the participants stated: *"Here [in VisExemplar] I need to think and imagine the output first. I then need to come up with strategies to show [demonstrate] parts of what I want to the system."* What we can draw attention to here is that unlike Polestar, VisExemplar advocates for the notion of "need-to-think" before specifying visual properties.

The notion of "need-to-think" in tools implementing VbD requires users to put more effort into both thinking about their visualization designs and providing demonstrations to the system, thus making the processes of visualization construction slower. However, this notion might lead to a more thoughtful process since the participants often mentioned the need to plan and/or think prior to engaging in the process of visualization construction. In particular, the notion of "need-to-think" might enable users to think more carefully about marks, visual variables, and their relevance in the design and construction of visualizations.

Combining MVS and VbD for Multi-Paradigm Interfaces

Our findings show that each interaction paradigm has its own advantages and disadvantages. As the two paradigms both have the same end goal or state (a visualization along with the full set of specifications needed to generate the visualization), the opportunity to consider a hybrid, multi-paradigm interface exists. (Figure 5). Based on the results of this study, such an interface can be designed based on task effectiveness. Depending on which tasks and operations an interface wants to support can help determine which of the two paradigms to use for the specific task.

In addition to effectiveness of allocating paradigms by tasks entirely, interfaces like this open the potential to consider executing

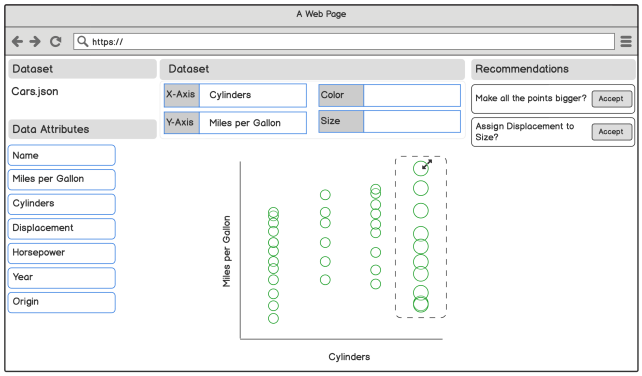

Figure 5: *Combining MVS and VbD into a single interface opens interesting user interface opportunities that can leverage aspects of both paradigms. In this case, the user has constructed a scatterplot using MVS and is demonstrating her interest in assigning a data attribute to the size of data points using VbD.*

tasks using true multi-paradigm interaction. For example, instead of performing a task purely by demonstration, interfaces like these can mix the paradigms within a task. Perhaps users can begin a task by demonstration, then refine the lower-level specifications using MVS. Similarly, perhaps users can specify aspects of their task using MVS first, then demonstrate more high-level concepts visually. The opportunity for multi-paradigm interfaces can help balance the need for people to understand all specifications to perform a task using MVS, and alleviate the need to demonstration tasks if the specifications are known. However, this also raises user interface design challenges, which will need to be explored further.

6. Limitations and Future Work

This study is a first attempt to compare people's visualization construction using MVS and VbD. However, a single study cannot answer all open questions about this process and is necessarily limited by several factors.

Implementation and Design of Visualization Tools

To compare two interaction paradigms, we had to select two digital tools that embody one of the paradigms. Thus, we chose VisExemplar and Polestar because each embodies one of the paradigms and their relative simplicity with regards to the remaining system components. However, user interface design in visualization tools embodying a specific paradigm can be implemented in various ways [GBTS13]. For instance, a tool that embodies MVS could be implemented using the shelf configuration, data flow, or visual builder interface design. IVoLVER [MHN17], for example, follows a data flow-based interface design requiring users to manually draw a visual glyph before binding data to it. While this makes the tool more flexible in terms of customizing the chart, it can also result in the tool being slower than a system like Polestar for specifying standard visualizations. As such, we want to emphasize that the selected tools in this study do not represent all possible interface designs for the MVS and VbD paradigms. Interface design might influence the results of the study. Thus, we encourage future work to consider the effect of tool design when exploring our findings.

Tasks

All tasks used in our study are visualization construction tasks that required participants to specify visualization properties (e.g., map a data attribute to an axis). Testing our research questions using different and more sophisticated types of tasks might reveal nuances of each interaction paradigm that were not tested in this study. As such, our findings should be interpreted in the context of the specified visualization tasks.

Participants' Visualization Expertise

In this study we did not control for participants' expertise. We hypothesize that participants expertise and prior knowledge about visualizations will influence their visualization construction process. For instance, expert users might prefer constructing visualization using the MVS paradigm since they have a better understanding of different visual encodings. In contrast, novice users might prefer working with tools that embody the VbD paradigm because of its freedom of expression and not requiring users to formalize the mappings between the data and visual encodings. However, this remains to be formally studied. Our work can be extended with other studies such as comparing these two paradigms on users with different level of visualization expertise.

7. Conclusions

We present a study comparing visualization construction using two visualization tools: one representing MVS and another representing VbD. Our findings indicate differences in how these interaction paradigms shape people's visualization construction, and people's experience and their perceived control and engagement during visualization construction.

References

[AEN10] ANDREWS C., ENDERT A., NORTH C.: Space to think: Large high-resolution displays for sensemaking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2010), CHI '10, ACM, pp. 55–64. URL: http://doi.acm.org/10.1145/1753326.1753336, doi:10.1145/1753326.1753336. 1

[Ahl96] AHLBERG C.: Spotfire: an information exploration environment. *ACM SIGMOD Record 25*, 4 (1996), 25–29. 1, 2, 3

[AT95] ABRAM G., TREINISH L.: An extended data-flow architecture for data analysis and visualization. In *Proceedings Visualization '95* (Oct 1995), pp. 263–270. 3

[BEDF16] BOY J., EVEILLARD L., DETIENNE F., FEKETE J.-D.: Suggested Interactivity: Seeking Perceived Affordances for Information Visualization. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (Jan. 2016), 639–648. URL: https://hal.inria.fr/hal-01188973, doi:10.1109/TVCG.2015.2467201. 8

[BL00] BEAUDOUIN-LAFON M.: Instrumental interaction: An interaction model for designing post-wimp user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2000), CHI '00, ACM, pp. 446–453. URL: http://doi.acm.org/10.1145/332040.332473, doi:10.1145/332040.332473. 8

[Car99] CARPENDALE M. S. T.: *A Framework for Elastic Presentation Space*. PhD thesis, Burnaby, BC, Canada, Canada, 1999. AAINQ51848. 2

[CH93] CYPHER A., HALBERT D. C.: *Watch what I do: programming by demonstration*. MIT press, 1993. 3

[CMS99] CARD S. K., MACKINLAY J. D., SHNEIDERMAN B. (Eds.): *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999. 2

[CMvdP10] CHAO W. O., MUNZNER T., VAN DE PANNE M.: Poster: Rapid pen-centric authoring of improvisational visualizations with napkinvis. *Posters Compendium InfoVis 2* (2010). 3

[CR98] CHI E. H.-H., RIEDL J.: An operator interaction framework for visualization systems. In *Proceedings of the 1998 IEEE Symposium on Information Visualization* (Washington, DC, USA, 1998), INFOVIS '98, IEEE Computer Society, pp. 63–70. URL: http://dl.acm.org/citation.cfm?id=647341.721078. 2

[Cre02] CRESWELL J. W.: *Educational research: Planning, conducting, and evaluating quantitative*. Prentice Hall Upper Saddle River, NJ, 2002. 7

[Dat15] DATASETS T.: https://public.tableau.com/s/resources, 2015. URL: https://public.tableau.com/s/resources. 4, 5, 7

[DE98] DIX A., ELLIS G.: Starting simple: Adding value to static visualisation through simple interaction. In *Proceedings of the Working Conference on Advanced Visual Interfaces* (New York, NY, USA, 1998), AVI '98, ACM, pp. 124–134. URL: http://doi.acm.org/10.1145/948496.948514, doi:10.1145/948496.948514. 1, 5

[EFN12] ENDERT A., FIAUX P., NORTH C.: Semantic interaction for visual text analytics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2012), CHI '12, ACM, pp. 473–482. URL: http://doi.acm.org/10.1145/2207676.2207741, doi:10.1145/2207676.2207741. 3

[EST07] ELMQVIST N., STASKO J., TSIGAS P.: Datameadow: A visual canvas for analysis of large-scale multivariate data. In *2007 IEEE Symposium on Visual Analytics Science and Technology* (Oct 2007), pp. 187–194. doi:10.1109/VAST.2007.4389013. 3

[GBTS13] GRAMMEL L., BENNETT C., TORY M., STOREY M.-A.: A survey of visualization construction user interfaces. *EuroVis-Short Papers* (2013), 19–23. 1, 2, 10

[GTS10] GRAMMEL L., TORY M., STOREY M.-A.: How information visualization novices construct visualizations. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (Nov 2010), 943–952. doi:10.1109/TVCG.2010.164. 2, 3

[HCT*14] HURON S., CARPENDALE S., THUDT A., TANG A., MAUERER M.: Constructive visualization. In *Proceedings of the 2014 Conference on Designing Interactive Systems* (New York, NY, USA, 2014), DIS '14, ACM, pp. 433–442. URL: http://doi.acm.org/10.1145/2598510.2598566, doi:10.1145/2598510.2598566. 1

[Hur14] HURON S.: Constructive visualization : A token-based paradigm allowing to assemble dynamic visual representation for non-experts. In *Theses. Universite' Paris Sud - Paris XI.* (2014). doi:https://tel.archives-ouvertes.fr/tel-01126892.

[HV81] HENDERSON H. V., VELLEMAN P. F.: Building multiple regression models interactively. *Biometrics* (1981), 391–411. 4, 6

[IH01] IGARASHI T., HUGHES J. F.: A suggestive interface for 3d drawing. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2001), UIST '01, ACM, pp. 173–181. URL: http://doi.acm.org/10.1145/502348.502379, doi:10.1145/502348.502379. 3

[KC14] KONDO B., COLLINS C.: Dimpvis: Exploring time-varying information visualizations by direct manipulation. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (Dec 2014), 2003–2012. 3

[KCPE16] KIM H., CHOO J., PARK H., ENDERT A.: Interaxis: Steering scatterplot axes via observation-level interaction. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (Jan 2016), 131–140. doi:10.1109/TVCG.2015.2467615. 3

[KKW*17] KWON CHUL B., KIM H., WALL E., CHOO J., PARK H., ENDERT A.: Axisketcher: Interactive nonlinear axis mapping of visualizations through user drawings. *IEEE Transactions on Visualization and Computer Graphics 23*, 1 (Jan 2017), 221–230. doi:10.1109/TVCG.2016.2598446. 3

[KPHH11] KANDEL S., PAEPCKE A., HELLERSTEIN J., HEER J.: Wrangler: Interactive visual specification of data transformation scripts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2011), CHI '11, ACM, pp. 3363–3372. URL: http://doi.acm.org/10.1145/1978942.1979444, doi:10.1145/1978942.1979444. 3

[KPV*18] KOYTEK P., PERIN C., VERMEULEN J., ANDRÃL' E., CARPENDALE S.: Mybrush: Brushing and linking with personal agency. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (Jan 2018), 605–615. doi:10.1109/TVCG.2017.2743859. 8

[KSL*17] KIM N. W., SCHWEICKART E., LIU Z., DONTCHEVA M., LI W., POPOVIC J., PFISTER H.: Data-driven guides: Supporting expressive design for information graphics. *IEEE Transactions on Visualization and Computer Graphics 23*, 1 (Jan 2017), 491–500. doi:10.1109/TVCG.2016.2598620. 2

[LKS13] LEE B., KAZI R. H., SMITH G.: Sketchstory: Telling more engaging stories with data through freeform sketching. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (Dec. 2013), 2416–2425. URL: http://dx.doi.org/10.1109/TVCG.2013.191, doi:10.1109/TVCG.2013.191. 3

[LTW*18] LIU Z., THOMPSON J., WILSON A., DONTCHEVA M., DELOREY J., GRIGG S., KERR B., STASKO J.: Data Illustrator: Augmenting Vector Design Tools with Lazy Data Binding for Expressive Visualization Authoring. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2018), CHI '18, ACM, pp. 123:1–123:13. URL: http://doi.acm.org/10.1145/3173574.3173697. 2

[LWN*09] LIN J., WONG J., NICHOLS J., CYPHER A., LAU T. A.: End-user programming of mashups with vegemite. In *Proceedings of the 14th International Conference on Intelligent User Interfaces* (New York, NY, USA, 2009), IUI '09, ACM, pp. 97–106. URL: http://doi.acm.org/10.1145/1502650.1502667, doi:10.1145/1502650.1502667. 3

[MHN17] MÉNDEZ G. G., HINRICHS U., NACENTA M. A.: Bottom-up vs. top-down: Trade-offs in efficiency, understanding, freedom and creativity with infovis tools. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2017), CHI '17, ACM, pp. 841–852. URL: http://doi.acm.org/10.1145/3025453.3025942, doi:10.1145/3025453.3025942. 3, 9, 10

[Per14] PERIN C.: *Direct Manipulation for Information Visualization*. PhD thesis, Paris 11, 2014. 9

[Pol16] POLESTAR: Polestart, http://vega.github.io/polestar/, 2016. URL: http://vega.github.io/polestar/. 3

[PSM07] POUSMAN Z., STASKO J., MATEAS M.: Casual information visualization: Depictions of data in everyday life. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (Nov 2007), 1145–1152. doi:10.1109/TVCG.2007.70541. 9

[RCDD13] REN L., CUI J., DU Y., DAI G.: Multilevel interaction model for hierarchical tasks in information visualization. In *Proceedings of the 6th International Symposium on Visual Information Communication and Interaction* (New York, NY, USA, 2013), VINCI '13, ACM, pp. 11–16. URL: http://doi.acm.org/10.1145/2493102.2493104, doi:10.1145/2493102.2493104. 5

[RLB19] REN D., LEE B., BREHMER M.: Charticulator: Interactive construction of bespoke chart layouts. *IEEE Transactions on Visualization*

and Computer Graphics 25, 1 (Jan 2019), 789–799. doi:10.1109/TVCG.2018.2865158. 2

[RTY14] REN D., TOBIAS H., YUAN X.: ivisdesigner: Expressive interactive design of information visualizations. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (Dec 2014), 2092–2101. doi:10.1109/TVCG.2014.2346291. 2

[Sed04] SEDIG K.: *Need for a Prescriptive Taxonomy of Interaction for Mathematical Cognitive Tools*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 1030–1037. 1

[SH00] STOLTE C., HANRAHAN P.: Polaris: A system for query, analysis and visualization of multi-dimensional relational databases. In *Proceedings of the IEEE Symposium on Information Vizualization 2000* (Washington, DC, USA, 2000), INFOVIS '00, IEEE Computer Society, pp. 5–14. URL: http://dl.acm.org/citation.cfm?id=857190.857686. 2

[Shn96] SHNEIDERMAN B.: The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the 1996 IEEE Symposium on Visual Languages* (Washington, DC, USA, 1996), VL '96, IEEE Computer Society, pp. 336–. URL: http://dl.acm.org/citation.cfm?id=832277.834354. 5

[SIM99] SHIPMAN III F. M., MARSHALL C. C.: Formality considered harmful: Experiences, emerging themes, and directions on the use of formal representations in interactive systems. *Computer Supported Cooperative Work (CSCW) 8*, 4 (1999), 333–352. 1

[SKBE17] SAKET B., KIM H., BROWN E. T., ENDERT A.: Visualization by demonstration: An interaction paradigm for visual data exploration. *IEEE Transactions on Visualization and Computer Graphics 23*, 1 (Jan 2017), 331–340. doi:10.1109/TVCG.2016.2598839. 1, 3

[SSEW17] SARVGHAD A., SAKET B., ENDERT A., WEIBEL N.: Embedded merge & split: Visual adjustment of data grouping. *IEEE Transactions on Visualization and Computer Graphics* (2017). 3, 9

[Tab18] Tableau Software, http://www.tableau.com/, 2018. URL: http://www.tableau.com/. 1, 2, 3

[VLvdHC13] VERMEULEN J., LUYTEN K., VAN DEN HOVEN E., CONINX K.: Crossing the bridge over norman's gulf of execution: Revealing feedforward's true identity. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2013), CHI '13, ACM, pp. 1931–1940. URL: http://doi.acm.org/10.1145/2470654.2466255, doi:10.1145/2470654.2466255. 8

[VWVH*07] VIEGAS F. B., WATTENBERG M., VAN HAM F., KRISS J., MCKEON M.: Manyeyes: a site for visualization at internet scale. *Visualization and Computer Graphics, IEEE Transactions on 13*, 6 (2007), 1121–1128. 2

[WLJ*12] WALNY J., LEE B., JOHNS P., RICHE N. H., CARPENDALE S.: Understanding pen and touch interaction for data exploration on interactive whiteboards. *IEEE Transactions on Visualization and Computer Graphics 18*, 12 (Dec 2012), 2779–2788. doi:10.1109/TVCG.2012.275. 3

[WMA*16] WONGSUPHASAWAT K., MORITZ D., ANAND A., JOCK M., HOWE B., HEER J.: Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (Jan 2016), 649–658. doi:10.1109/TVCG.2015.2467191. 3

[WPHC16] WUN T., PAYNE J., HURON S., CARPENDALE S.: Authoring with microsoft excel and tangible tiles. In *Computer Graphics Forum* (2016). doi:http://dx.doi.org/10.1111/cgf.12887. 3

[WQM*17] WONGSUPHASAWAT K., QU Z., MORITZ D., CHANG R., OUK F., ANAND A., MACKINLAY J., HOWE B., HEER J.: Voyager 2: Augmenting visual analysis with partial view specifications. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2017), CHI '17, ACM, pp. 2648–2659. URL: http://doi.acm.org/10.1145/3025453.3025768, doi:10.1145/3025453.3025768. 1, 3

[YKS07] YI J. S., KANG Y. A., STASKO J.: Toward a deeper understanding of the role of interaction in information visualization. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (Nov 2007), 1224–1231. 1, 5

[Zlo75] ZLOOF M. M.: Query by example. In *AFIPS National Computer Conference* (1975). 3