Data-Driven Guides: Supporting Expressive Design for Information Graphics

Nam Wook Kim, Eston Schweickart, Zhicheng Liu, Mira Dontcheva, Wilmot Li, Jovan Popovic, and Hanspeter Pfister

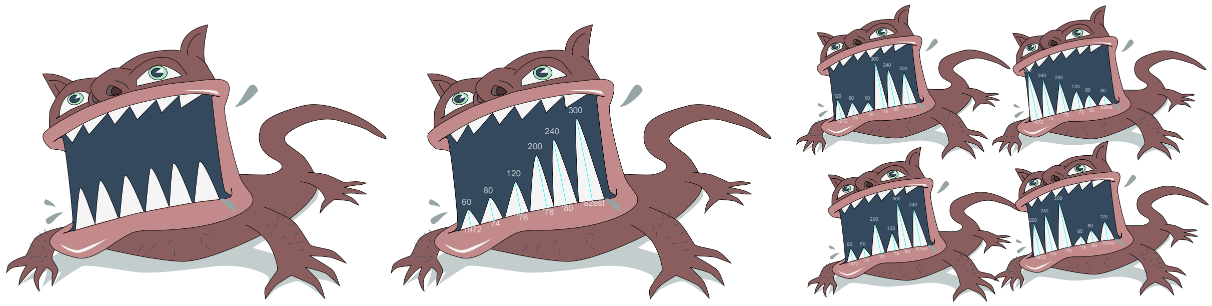


Fig. 1: Nigel Holmes' *Monstrous Costs* chart, recreated by importing a monster graphic (left) and retargeting the teeth of the monster with DDG (middle). Taking advantage of the data-binding capability of DDG, small multiples are easily created by copying the chart and changing the data for each cloned chart (right).

Abstract—In recent years, there is a growing need for communicating complex data in an accessible graphical form. Existing visualization creation tools support automatic visual encoding, but lack flexibility for creating custom design; on the other hand, freeform illustration tools require manual visual encoding, making the design process time-consuming and error-prone. In this paper, we present *Data-Driven Guides* (DDG), a technique for designing expressive information graphics in a graphic design environment. Instead of being confined by predefined templates or marks, designers can generate guides from data and use the guides to draw, place and measure custom shapes. We provide guides to encode data using three fundamental visual encoding channels: length, area, and position. Users can combine more than one guide to construct complex visual structures and map these structures to data. When underlying data is changed, we use a deformation technique to transform custom shapes using the guides as the backbone of the shapes. Our evaluation shows that data-driven guides allow users to create expressive and more accurate custom data-driven graphics.

Index Terms—Information graphics, visualization, design tools, 2D graphics.

1 Introduction

With the increased quantity and improved accessibility of data, people from a variety of backgrounds, including journalist, bloggers, and designers, seek to effectively communicate messages found from complex data in an accessible graphical form. Unlike traditional visualizations (e.g. bar charts or scatterplots) that focus on data exploration and analysis, communicative visualizations put more emphasis on presentation [31]. Commonly referred to as *infographics*, these visualizations are often embellished with unique representations to convey a story or specific message. When creating such custom information graphics, designers must consider various factors including not only perceptual effectiveness, but also aesthetics, memorability, and engagement [37, 7, 17]. While embellishments in visualization design have traditionally been considered harmful, thoughtfully crafted custom visualizations can be highly engaging and get the messages across more effectively.

In recent years, many visualization creation tools have been developed to meet the growing demand for visually communicating data [14]. To make visualization construction easier, most existing tools automate the visual encoding process. For instance, chart templates ease the burden of manually encoding data by providing predefined palettes of chart types. However, they do not allow users to create novel custom charts except changing a small number of style parameters such as colors or fonts. More sophisticated tools improve upon the template-based approach by enabling a wide range of specifications for data graphics including marks, scales, and layouts. While these tools make complex visual encoding easy, they tend to limit the design space or enforce a rigid order of operations in order to achieve desired effects. As noted in the comprehensive user study by Bigelow et al. [3], the lack of flexibility in existing visualization creation tools reduces their applicability to designers. For this reason, designers still rely on freeform illustration tools such as Adobe Illustrator[1] to create custom visualizations, which currently do not provide visualization-specific abstractions. This results in time-consuming and error-prone manual visual encoding that prevents designers from exploring diverse design variations.

In this research, we addresses the question of how to reduce the gap between easy-to-use visualization creation and flexible graphic design tools. Informed by Bigelow et al. [3] who studied how designers work with data, we focus on the less explored area of how designers manually encode data into custom graphics in a graphic design environ-

- *Nam Wook Kim and Hanspeter Pfister are with the John A. Paulson School of Engineering and Applied Sciences, Harvard University, Email:namwkim, pfister@seas.harvard.edu*
- *Eston Schweickart is with the Computer Science department at Cornell University, Email: ers273@cornell.edu*
- *Zhicheng Liu, Mira Dontcheva, Wilmot Li, and Jovan Popovic are with Adobe Research, Email:leoli@adobe.com, mirad@adobe.com, wilmotli@adobe.com, jovan@adobe.com*

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

[1] `http://www.adobe.com/Illustrator`

ment. To identify the challenges designers face in creating custom visualizations of data, we conducted semi-structured interviews with infographic designers. Findings reveal that tool flexibility is important in infographic design for various reasons including designing custom marks and adding annotations. Designers employ various tricks and hacks to work around the lack of data-driven abstractions in graphic design tools. More specifically, two common issues that emerged from the manual encoding practice are 1) the laborious task of placing and measuring graphics based on data using guides such as rulers or grids, and 2) the absence of data binding in hand-crafted design or externally created charts. We conclude that there are unique opportunities to improve current graphic design tools by providing support for data-driven design, instead of trying to make current visualization creation tools more flexible by adding more parameters.

To alleviate the issues in the existing infographic practice, we contribute *Data-Driven Guides*(DDG), a technique for designing expressive data-driven graphics. Instead of being confined by predefined templates or marks, designers can generate guides from data and use the guides to accurately place and measure custom shapes. We provide guides to encode three main visual channels: length, area, and position following the principles of information encoding [2, 36]. Users can combine more than one guide to construct a variety of visual structures that represent data. When the underlying data is changed, we use a 2D deformation technique to transform the user-defined shapes based on the guides.

To demonstrate how DDG can be integrated into a designer's flexible workflow, we implement DDG in the context of a web-based vector drawing tool to support infographic design. To evaluate DDG's ability to support the flexible, expressive, and accurate design of custom data graphics, we demonstrate its use to create diverse example graphics that are difficult to manually construct or that were inaccurately created with existing tools. We also conducted a user study to evaluate the usability of DDG. Participants describe the interaction model for DDG as intuitive and straightforward, suggesting DDG was more useful for data-driven drawing compared to traditional guides including rulers and grids. Their feedback confirms that DDG would improve their design practice of creating custom information graphics.

2 Related Work

2.1 Visualization Design Tools & Environments

For the last few decades, there has been considerable effort to create easy-to-use interfaces for data visualization. Grammel et al. provide a survey on various types of visualization construction tools [14]. Among them, chart templates are most widely used in many applications including spreadsheets, presentation software, graphic design tools, and online services (e.g., Many Eyes [49], RAW [2], Plotly [3]). They facilitate the quick and easy construction of charts, though users are limited to predefined chart types and are only allowed to change a small number of configuration parameters.

More advanced tools enable more expressive design of data graphics by exposing low-level specification parameters such as scales and marks. Some of these tools [45, 41] are based on formal graphical specifications such as the grammar of graphics [54] or a declarative model [19, 10]. Tableau [4] follows a similar formulation, and has been designed to support rapid exploratory data analysis rather than custom visualization design. On the other hand, Lyra [41], built on the Vega grammar [5], provides a more accessible interface to customize visual encodings. Spritzer et al. [44] use CSS-like stylesheets to touch up the look of nodes, edges, and presentation aids in node-link diagrams to enhance their communicative power.

Some tools attempt to reduce the gulf of execution by appropriating direct manipulation techniques [26, 40, 33] or demonstrational interfaces [38]. Other tools allow users to interactively construct multiple, coordinated visualizations [39, 52]. Although existing tools have enabled the design of highly customized visualizations without programming, they are still confined to preset scales, layouts, and marks. The ability for designers to directly manipulate visual structures on the canvas is very limited compared to freeform drawing tools.

On the other hand, there are a variety of programming toolkits that enable a high degree of control. They provide flexibility in creating custom representations with the help of the expressive power of the underlying programming languages. Some are based on a formal specification approach [8, 53], and others provide composable operators for different visual encodings [20] or extensible visualization widgets [12]. However, the flexibility of programming languages comes with a steep learning curve for people who do not have programming expertise, including many designers.

Other generic tools can also be used to create custom visualizations. Visual programming approach allows users to construct the underlying data flow and visual structure of a visualization using node-link style interfaces [13, 30]. However, they separate the construction interface from the canvas area (i.e., poor 'closeness of mapping' between the problem domain and the tool [15]). Constraint-based drawing tools enable users to create a visualization purely based on geometric constraints on the properties of graphical primitives including position, rotation, and scaling. They are not optimized for visualization construction (e.g., repetition of a mark)[6] or require procedural thinking to define loops to generate visual marks parametrically [7].

Despite the wealth of tools available for creating visualizations, Bigelow et al. [3] found in their study that designers, who are primary producers of many popular visualizations in recent years, do not use most, if any, of those tools. They found that manual visual encoding using freeform illustration tools is tolerated in order to maintain flexibility and richness in the design process. Similarly, previous studies investigated potential benefits (e.g., familiarity and expressivity) of manual visual mapping in physical design environments where a visualization is constructed through hand-drawn sketching [51, 50] and manipulation of tangible materials [25, 29, 46, 24, 55].

Inspired by these works, we attempt to take advantage of the benefits of manual encoding to support expressive design of infographics. Instead of developing another visualization creation tool that enforces a rigid order of operations and prevents users from engaging in manual encoding, we provide guides generated from data that help designers draw their own visual marks and layouts in a flexible graphic design environment.

2.2 Benefits of Visual Embellishments

A common belief in the visualization community with regards to visualization design is that visual representations should maximize the data-ink ratio and avoid unnecessary decoration as much as possible [48]. Most visualization systems today are based on these principles that inform perceptually effective visual encodings of data [11, 18, 47]. It is only recently that researchers have started exploring other aspects of visualization design such as memorability [7, 6], aesthetics [37], and engagement [17, 9]. These metrics focus on communication and presentation rather than data exploration and analysis. Recent studies looked at the benefits of embellishments on comprehension and recall [1, 16, 23, 5]. Although embellishments can have negative impacts on visual search time [5] or certain analytic tasks [43], it is now generally understood that embellishment is not equivalent to chart junk. *Judiciously* embellished visual representations can help communicate the context of data that makes it easier to remember and recall. As a result, there has been active development of presentation-oriented visualization techniques [31, 44] that are beginning to find applications in visual storytelling [33, 32].

As noted by Bigelow et al. [3], designers are likely to continue to use freeform graphic design tools for the sake of flexibility, but these tools do not currently provide well-defined data-driven abstractions. Our goal in this research is to provide appropriate tools to alleviate error-prone manual operations required in the design of engaging and memorable custom infographics.

[2] http://raw.densitydesign.org
[3] https://plot.ly/
[4] https://public.tableau.com
[5] https://vega.github.io/vega/
[6] http://aprt.us
[7] https://vimeo.com/66085662

2.3 Deformation for Vector Graphics

Deformation is a well-studied topic in the area of computer graphics and has found many applications in various fields including computer-aided design, fabrication, and computer animation. It can be used to transform raster images, geometric models, and vector graphics in a more flexible way compared to object-level affine transformations. The most common technique used for shape deformation is linear blend skinning [28]. It applies a weighted sum of affine transformations to each point on the object, where weights are often chosen manually by users. Recently, Jacobson et al. [27] developed bounded biharmonic weights that can reduce tedious manual weight painting and allow interactive deformation through convenient handles such as points, bones, and cages. Liu et al. [34] extended this work to allow for deforming vector graphics (e.g., Bézier splines).

In this work, we augment the deformation techniques listed above to transform custom shapes based on data using guides as the backbone of the shapes. This approach is similar in intent (though not in implementation) to the skeletal strokes by Hsu et al. [22]. While creating a data visualization usually involves simple linear transformations applied to conventional marks such as rectangles and circles, flexible shape deformation allows for more expressive design.

3 Infographic Design Practice

In order to understand the design practice employed by infographic designers, we conducted semi-structured interviews with six design professionals; we also analyzed examples of infographics from online tutorials, books, and videos on hand-crafting infographics. The interviewees include two professional designers (P1, P2) hired through UpWork [8], three student designers (P3, P4, P5) enrolled in a master's program in information design and visualization, and a visualization researcher (P6) who has experience in infographic design. Through the interviews and analysis of existing design practices we hoped to understand the infographic design process, why designers still rely on graphic design tools, and what difficulties they face in creating infographics. Each interview session took about 45-60 minutes. The interviewees were asked to walk through their design processes as well as to elaborate on how they manually encode data into graphics using their own examples. While our findings in general are in line with the the work by Bigelow et al. [3] we focused particularly on the challenges in manual visual encoding.

Lack of flexibility or data-encoding support in existing tools. All the interviewees emphasized the importance of tool flexibility in order to cope with many intricate factors in infographic design. This was even true for those who have experience in programming or advanced visualization creation tools. They particularly valued the ability to make design decisions on their own (P2, P4), and disliked the rigid and automatic design process enforced by visualization creation tools (e.g., Tableau's shelf configuration) (P3, P5). They said that these tools are not necessarily designed with communication in mind, making it difficult to customize graphics through means such as adding annotations and embellishments or designing new visual marks and layouts (P1, P3, P6). For example, two of them complained that their clients requested a specific aspect ratio or corporate theme, which is impossible to handle using existing visualization creation tools (P1, P3). These findings conform with the previous research [3].

However, an interviewee (P5) also suggested that graphic design tools are also limited because of their lack of support for drawing data-driven graphics. In order to work around the limitation, designers often employ various manual tricks and hacks (e.g., using brushes to create arc bars [9]). Interviewees acknowledged that they made mistakes due to this manual encoding process, often generating inaccurate representations of data (P2, P3, P4) such as placing or scaling a visual mark a few pixels off from its actual data value. Interestingly, they did not consider this as an important issue, instead emphasizing the communication of the overall message.

[8]`https://www.upwork.com`

[9]`https://youtu.be/VMgN6niy3ns`

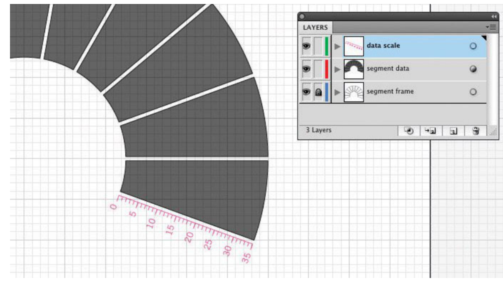


Fig. 2: Excerpt from an online tutorial[10] explaining how to measure a custom visual mark using a hand-crafted scale.

Tedious manual encoding required in creating custom design. To design visual representations of data, designers employ diverse design methods. In the case of using a built-in charting tool (e.g., the graph tool in Adobe Illustrator) or importing external charts, the degree to which designers perform customizations is marginal (e.g., changing basic style properties such as colors, sizes, or fonts). On the other hand, when designing unique visual marks or layouts, design methods are more involved and require laborious manual encoding. The purpose of such methods all comes down to how to place and measure custom shapes based on data. We found that designers typically rely on traditional guides such as rulers or grids, or even create their own scales (Fig. 2). For example, an interviewee (P5) explained the use of a bar chart as a measurement tool by juxtaposing the bar chart with custom shapes. Other interviewees also showed other cases such as measuring areas using the AutoCad software (P1) and finding the locations of visual marks on a map using a GIS software (P2). Interviewees, particularly those who have experience in programming (P3, P6), also said that it would be best to use external tools and environments for handling large data or more complex charts such as graphs or networks.

Absence of data binding in custom or imported charts. When designers use unconventional methods to craft custom visual encodings, data binding is usually not supported. Even though the final design outcomes are static printed media, the lack of data binding interferes with the overall design process. For example, as noted in [3], a complete redesign is often caused by the designers' wrong assumptions about the data behavior and because they start sketching without using actual data. Our interviewees also mentioned that they occasionally use wrong data or need to update existing data sets with new ones, showing a potential benefit of data binding (P1, P5, P6). In a related example, designers use the graph tool in Adobe Illustrator to create a chart. To be able to freely manipulate the geometric primitives, they ungroup the chart elements. However, the ungrouping operation results in a loss of data binding. An interviewee said that, for this reason, she usually adheres to the given chart even if she wants to further customize its design such as deleting its axis lines (P5); we later found that this was the case for many designers [11]. Another interviewee also expressed frustration about the absence of data binding and the need for externally created and imported charts (P6).

4 Design goals

Our findings from the interviews imply that graphic design tools are flexible but currently lack appropriate support for data-driven design. To address this challenge, we decide to augment the current design experience already common to designers, rather than develop a completely new visualization design tool. Based on our interviews and analysis of how infographics are usually created we have identified the following design goals for an infographic design tool. These design goals provide concrete guidelines to improve the process of constructing custom data graphics within the context of designers' exist-

[10]`http://www.digitalartsonline.co.uk/tutorials/adobe-illustrator/design-magazine-infographic/#9`

[11]`https://www.smashingmagazine.com/2010/09/creating-graphs-with-adobe-illustrator/`

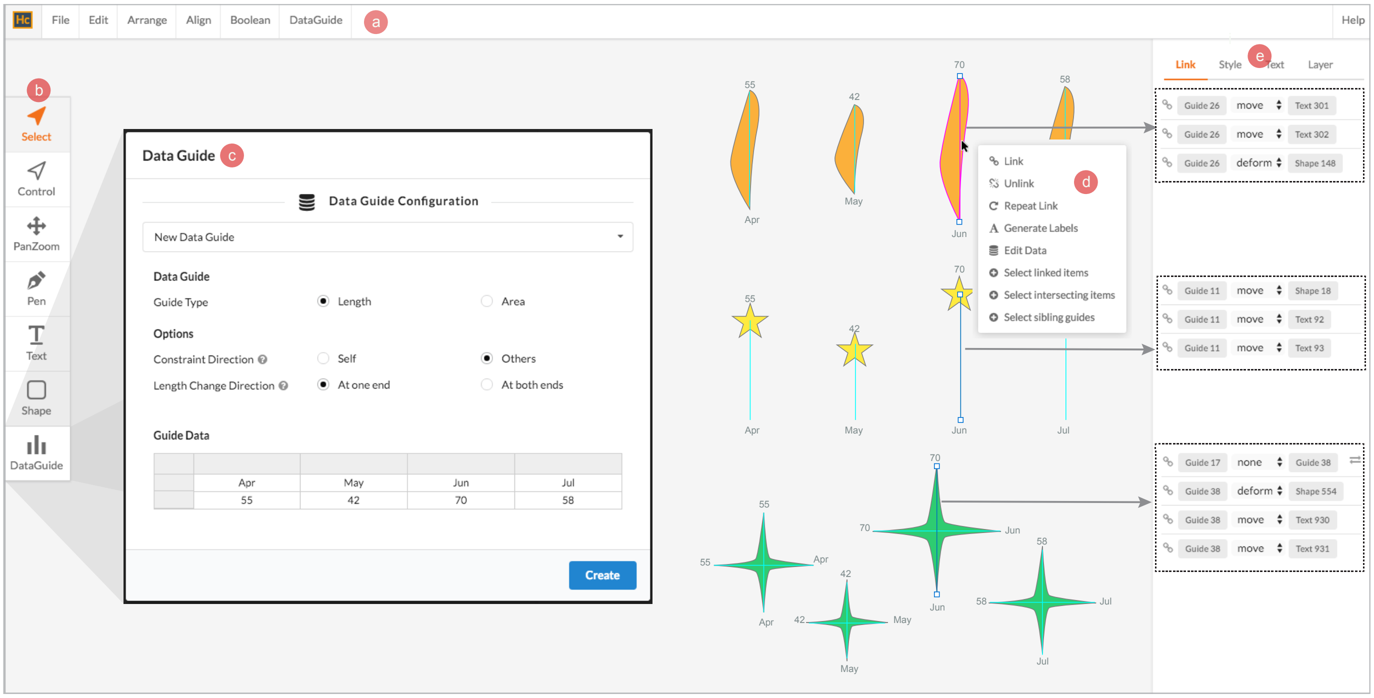


Fig. 3: Overview of the vector drawing tool in which DDG is implemented: (a) top menu bar providing conventional features such as undo, redo, and SVG import & export operations. The menu item for DDG is the same as the context menu. (b) left toolbar providing the DDG tool as well as other selection and drawing tools. (c) DDG tool panel for specifying a dataset and options to create data guides. It is also used for updating the guides. (d) context menu for executing DDG related commands such as linking objects, repeating shapes, and generating labels. (e) side panel for adjusting the various properties of selected objects such as the behavior or direction of the links as well as object styles including color or opacity. The side panel currently shows link configurations for three selected data guides.

ing design practice.

1. Maintain flexibility in the design process. Our interviewees listed the lack of freedom as one of the major factors that discourages them from using existing visualization creation tools. Augmenting current graphic design tools would be beneficial, since designers are already familiar with these tools and their flexibility in dealing with intricate design considerations. Instead of enforcing a rigid order of operations to create visualizations, the tool should relax constraints on the sequence of infographic construction enabling diverse workflows. Designers often create infographics through a top-down graphical process in which they design the overall appearance before plotting the real data [3]. The tool should be flexible enough to allow for the custom design of marks and layouts, satisfying the designers' need to express their creativity and to design novel infographics from the ground up.

2. Provide methods for accurate data-driven drawing. The designers in our interviews raised concerns around the tedious and error-prone process of manually encoding data into custom graphics. Their existing design practice demonstrated that the current support from graphic design tools such as rulers or grids is not sufficient. Providing advanced guides can be a potential solution as interviewees reported that they relied on custom scales or traditional charts as data guides. The advanced guides can be driven by data and designed to place and measure custom shapes along any dimension in contrast to existing orthogonal axes, which work best with conventional marks such as rectangles, lines, or circles. Embedding data-driven drawing capabilities into graphic design tools can significantly reduce the need for manual and error-prone data encodings.

3. Support persistent data binding for freeform graphics. A common challenge designers face is the absence of data binding support in custom charts and imported charts. Therefore, it would make sense that graphic design tools support persistent data bindings for imported charts [4] or provide ways to bind data to freeform graphics. In order to support the top-down design process, the ability to place data on existing graphics can be a possible solution instead of generating visualizations directly from data in a non-visual, symbolic manner via a GUI [3]. The flexible data binding support will increase the reusability of custom charts and also help designers avoid tedious rework when data is changed, allowing them to explore different design variations more efficiently.

5 Data-Driven Guides

We now introduce DDG, a technique for designing expressive custom infographics based on data. We draw inspiration from existing design practices in areas such as architectural or user interface design, where a guide is used as a reference (e.g., a ruler and grid) for precise drawing or alignment. We explain DDG in the context of a vector drawing tool we built to provide a flexible design environment. The technique can be implemented in other graphic design tools such as Adobe Illustrator. The term DDG can refer to both the technique we are introducing and the actual guides. To disambiguate, we use "DDG" to refer to the technique and "data guides" when we talk about the actual guides in the rest of this paper.

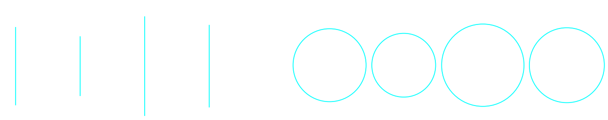


Fig. 4: Length and area guides that can also be used as a position guide.

We follow the theoretical frameworks of visual encodings [2, 36] that describe the most effective channels to encode information. We use data guides to size the primary visual variables of length and area, which in turn are represented as line- and circle-shaped guides, respectively (Fig. 4). The resulting guides can be used to encode positions as well (See balloons in Fig. 12f). In addition, more than one guide can be combined to create more expressive visual structures (Fig. 12f-i). The visual variables—length, position, and area—are popularly used in infographic design [7], though area encodings are frequently misused by designers using inaccurate scales (e.g., using the diameter of circles to match data instead of the area [12]). Other visual variables such as color or angle are not as effective as length, area, and position for encoding quantitative values, and are left for future work.

5.1 Providing Flexible and Familiar Interactions

DDG is designed to be fluidly integrated in a flexible graphic design environment, favored by infographic designers. To this end, the visual appearance and interaction model of DDG follow those of regular guides available in existing graphic design tools. The data guides always appear on top of other objects, have particular fill and stroke styles (e.g., no fill color or stroke width), and are not printed in the final design. Similar to regular guides, data guides do not impose a specific design workflow, meaning that they can be used at any stage of the design process (e.g., both top-down and bottom-design workflows).

A main difference from regular guides is that data guides are driven by data. A group of data guides can be created from a tabular dataset consisting of a series of numerical values and their category names; depending on the encoding type, the length (line) or area (circle) of a data guide represents a data value in the dataset. The relative sizes of data guides within the parent group are preserved in order to be in sync with the underlying dataset. For example, if a user scales a data guide, the other guides in the same group (i.e., its siblings) are proportionally scaled (Fig. 5).

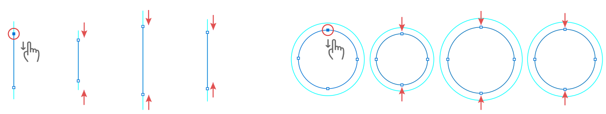


Fig. 5: Changing a data guide will affect its siblings in order to preserve the relative size differences within the same group (blue colors represent new guide states and red colors represent the direction of user manipulation, e.g., grabbing and moving an anchor point.).

To provide familiar interactions used in graphic design tools, DDG supports free manipulation (i.e.,move, rotate, scale etc) to create a custom layout. Users can also change the underlying dataset by manipulating data guides directly on the canvas. For instance, when a data guide is copied and pasted, a new data guide is created within the same group, which also adds a new data value to the dataset; likewise, if the whole group of guides is copied and pasted, a whole new dataset is created (Fig. 1 right). Also, combining different groups of guides has the effect of combining different datasets graphically.

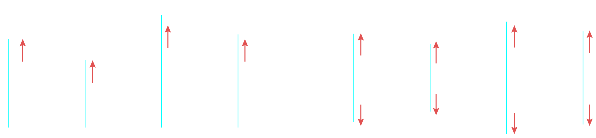


Fig. 6: Users can specify the direction in which a length guide change its length: either endpoint (left) or both endpoints (right) of the line segment.

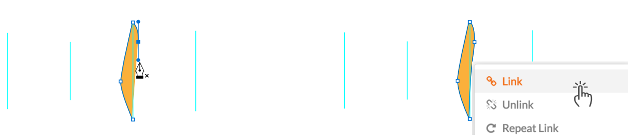


Fig. 7: Drawing a shape directly on top of data guides will link the shape to the guide (left). Otherwise, users can explicitly link the shape with the guides (right).

[12] `http://www.coolinfographics.com/blog/2014/8/29/false-visualizations-sizing-circles-in-infographics.html`

5.2 Data-Driven Drawing with DDG

A data guide serves as a ruler backed up by data to minimize designer's effort to manually place and measure graphics; its size and shape indicate where a data value lies on the canvas. Users can draw custom shapes from scratch directly on top of data guides (Fig. 7 left). The overall drawing experience is closer to drawing with a pen and ruler (i.e., bottom-up design process). Alternatively users can use data guides to repurpose existing artworks by matching the artworks to the size of the guides (Fig. 1); this workflow is the top-down, graphical process of placing data on existing graphics. A data guide supports snapping to its anchor points and path segments for precision.

To keep track of the correspondence between guides and shapes, DDG introduces the notion of linking. To reduce the effort of explicitly creating links, we automatically create the links in certain cases. For example, when a shape's anchor point is placed directly on a guide (e.g., drawing a shape on top of DDG), or if a guide is adjusted so that its own anchor point is placed on a shape (e.g., retargeting existing shape), then the shape is automatically linked to the guide. Data labels are generated only if requested (Fig. 10), and automatically linked to related guides on creation.

To ensure flexibility in measuring custom shapes, a length guide can be a curved line, if necessary, by adding additional anchor points along its path and adjusting the handles of the anchor points (Fig. 12e). The length guide has one additional option indicating the direction in which it increases or decreases when its data value is changed (*Length change direction* option in Fig. 3c); i.e., both endpoints or either endpoint of the line segment (Fig. 6). An area guide always remains a circle shape for accurate perception of the area; a squared rectangle would provide a comparable area perception, although the perception judgement may be impaired by varying orientations [11].

Users can combine multiple data guides in order to construct more expressive structures (Fig. 12c, f-i). This has the same effect as combining different visual variables, in our case length, area and position. To help constructing a visual structure, we provide two simple layout functions including linear and radial layouts (,) in addition to conventional alignment functions (e.g., align to left, distribute vertically etc).

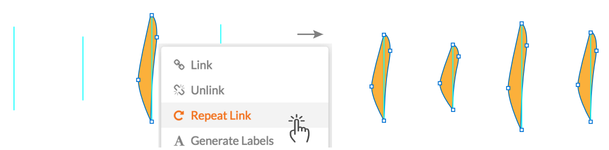


Fig. 8: Selecting and repeating a shape will duplicate the shape over the sibling guides of its linked guides.

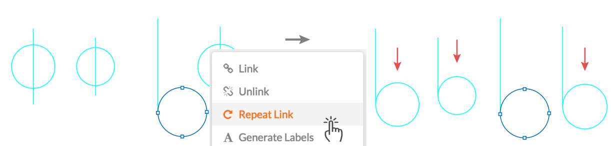


Fig. 9: Selecting and repeating a guide (area) will reposition its sibling guides based on its position relative to the linked guide (length).

To further assist in drawing a data-driven graphic, we provide a number of visualization-specific features. The repeat feature in DDG allows an associated shape to be repeated over its sibling guides in the same group (Fig. 8, 9). If a pair of guides from two groups is supposed to encode a single shape, we only repeat the shape once (e.g., a pregnant women figure in which her belly and height is associated with area and length guides respectively, in 12i). Our repeat command is optional and unobtrusive, meaning users can use a different mark for each guide. They can also customize each shape after executing the repeat command.

In addition, creating a visualization often involves the generation of many repeated visual elements in a small area on the canvas. In order to assist in making edits to the visualization, we provide a number of

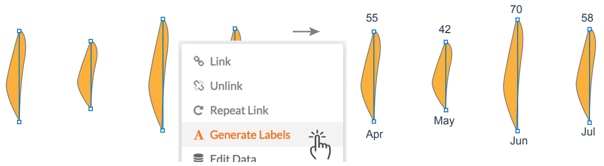


Fig. 10: Labels are generated only when requested and are automatically linked to data guides when created.

selection methods. 1) selecting objects associated with data guides , 2) selecting sibling guides within the same group, and 3) selecting intersecting objects. These selection methods operate based on the current selection and can be progressively applied to expand the current selection. This selection mechanism resembles D3 [8]'s selectors or the magic wand tools in graphic design applications that select objects of the same color, stroke weight, or opacity.

5.3 Supporting Data Binding with Custom Graphics

DDG is basically an intermediate layer for associating data with any objects including shapes, texts, or guides. When data is changed, related guides will consequently change their forms, which in turn transforms the objects that are linked to the guides (Fig. 11).

To address the challenge of binding data to freeform shapes, we employ a deformation technique for vector graphics [27, 34] as it enables a shape to change its form along any dimension (i.e., using the data guide as the backbone of the shape). The deformation technique applies a linear combination of affine transformations to each anchor point on the shape in order to adapt to the new state of the associated guide. Occasionally, we would find that the change of a backbone guide would cause associated shapes to extend beyond the end of the backbone, or not extend far enough. This is a particularly important scenario to avoid in DDG, as it can lead to misleading data graphics. If we identify points on a shape that should exactly follow the end of its backbone (i.e., the end-points of a length guide), we set a stronger weight for a transformation handle at the end of the backbone.

We also provide a translation-only behavior in case the user wants to use a guide to position custom graphics. In this case, the shape is not deformed but translated along the direction in which the associated guide is changed (e.g., along the path of a length guide or the direction from the center to the perimeter of an area guide). The translation behavior is the only option for the links to data guides and text labels as they are not deformable.

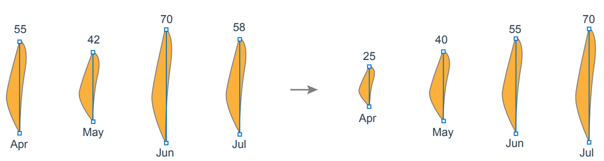


Fig. 11: Updating data will transform shapes using the guides as the backbones of the shapes.

To enable more expressive design, we attempt to relax the constraints on how links are constructed. More than one shape can be linked to a data guide (e.g., a composite balloon shape is associated with an area guide in Fig. 12g), and likewise multiple guides can be associated with a single shape (e.g., a cloud has four area guides attached in Fig. 12h). In the latter case, all the guides are used to transform the shape. In addition, data guides can be linked to themselves, allowing more complex structures (Fig. 12c, f-i). For example, a length guide can be used to position an area guide (in Fig. 12g, i) or a group of guides (in Fig. 12f, h).

When a new guide is added or an existing guide is removed (i.e., data cardinality change), we need to rearrange data guides accordingly. To achieve this, we heuristically estimate the categorical scale and orientation of the guides. This problem is difficult because the layout of the guides is not fixed, and it becomes more complicated when the visual structure of an infographic combines multiple groups of guides. DDG currently works in limited cases (e.g., a single group with linear and radial layouts), and needs to be improved in the future.

6 Interface Design and System Implementation

Figure 3 shows our vector drawing tool that combines all the features of DDG in a unified application. The overall interface is similar to other graphic design tools. In the left toolbar, basic drawing and selection tools are supported along with the *DataGuide* tool with which users can create and update data guides (Figure 3c). Using the *Select* tool, users can manipulate objects including guides (i.e., moving, rotating, and scaling). Initially, guides generated from the same dataset belong to a group. To select an individual data guide, the user must double-click the group first. Using the *Control* tool, users can select anchor points, handles, as well as individual paths. They can use the *Pen* tool to draw custom shapes, made of spline curves, with the help of the data guides. Users can link the shapes with the data guides through the context menu or top toolbar, in which they can also enable other features such as repeat, label generation, and selection (Figure 3a, d).

In the right panel (Figure 3e), users can change various properties of selected objects such as fill color, stroke width, or text size through *Style* and *Text* tabs. When a data guide is selected, a linear or radial layout configuration option is activated on the *Style* tab. Users can also inspect the underlying scene graph through the *Layer* tab. On the *Link* tab, users can inspect all the links associated with selected objects (Figure 3e). They can also change the desired behavior of the link (i.e., deformation or translation).

In the menu bar, we provide common drawing tool features such as aligning, arranging, and grouping objects, undo/redo operations, and other features designed for data guides. Guide-specific features are also available through the context menu. The tool has the ability to import SVGs and export the canvas area, which is useful for repurposing existing artworks using data guides as well as leveraging other full-fledged design tools to draw more complex shapes. When the canvas area is exported, our tool-specific properties such as links between guides and objects are preserved in the exported SVG file.

The tool is web-based and heavily depends on Paper.js [13], a vector graphics framework that runs on top of the HTML5 Canvas. The tool runs on a Python web server that interfaces with our deformation framework written in C++. The deformation framework accepts three inputs that include a list of shapes to be deformed, a list of guides before a deformation, and a list of guides after deformation. We plan to make the tool and all our code open-source available upon publication of this paper.

7 Evaluation

7.1 Example Infographics

To evaluate the expressivity of DDG, we created a diverse set of example data-driven infographics. The examples include simple basic charts as well as more expressive data graphics that are difficult to create using existing tools or with programming; they are only a subset of infographics that users can create with DDG. We also found that some existing infographics that were not created with DDG contained inaccurate data mapping when compared to DDG generated from the same datasets.

As DDG does not enforce predefined palettes, users have the freedom to create custom designs. They can freely manipulate the guides and design their own visual marks to create different styles of charts as shown in Fig. 12. For example, they can create visual marks from a single stroke (Fig. 12b), using the repeat command (Fig. 12d), or by drawing different marks for individual guides (Fig. 12a). They also can create a layout using simple alignment and distribution features (e.g., aligned to bottom in 12c) that are typically available in any graphic design tool or using the layout functions (e.g., radial layout in Fig. 12d).

In addition to changing the basic styles of charts, users can create further embellished data graphics. We recreated two Nigel Holmes'

[13] http://paperjs.org/

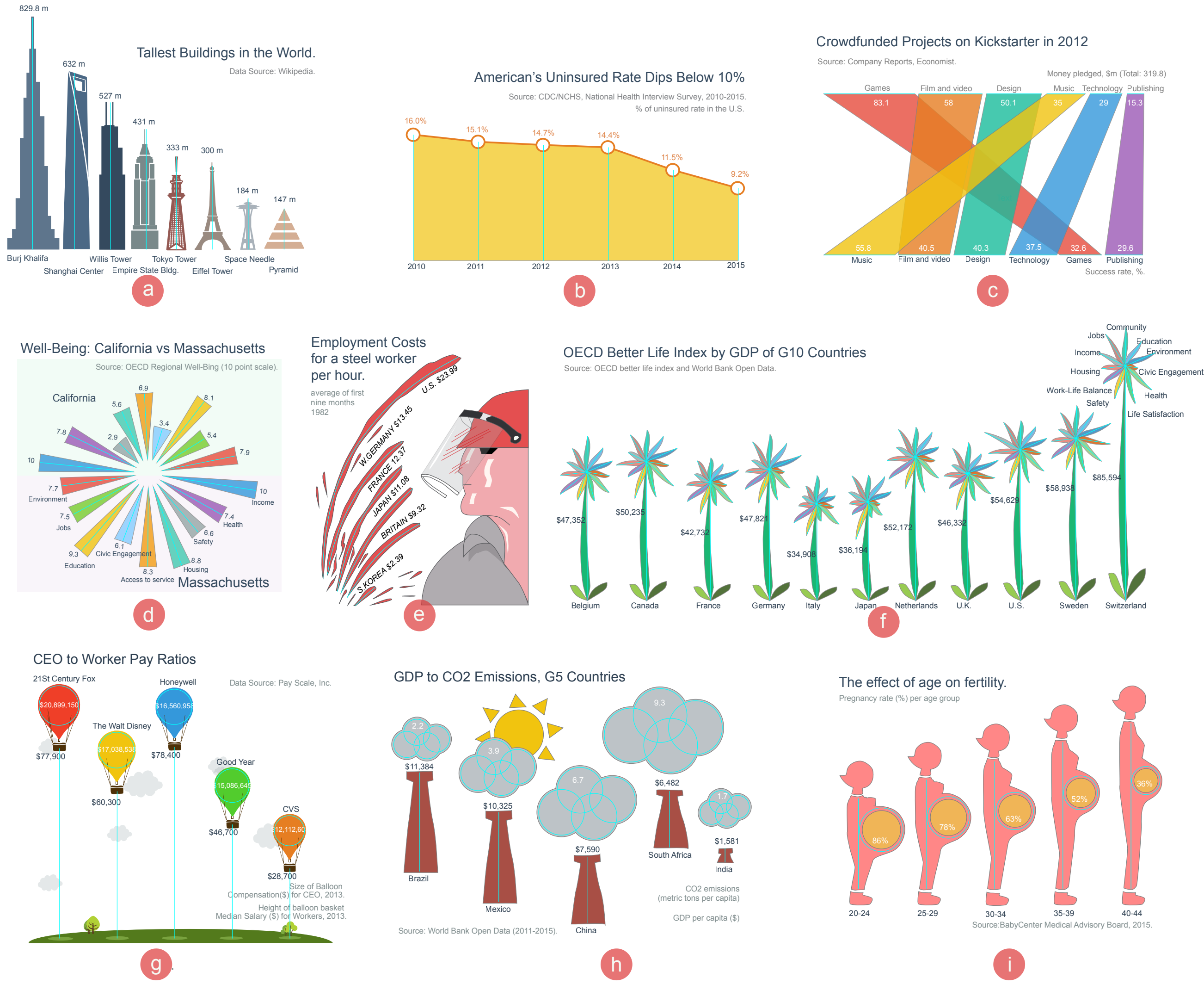


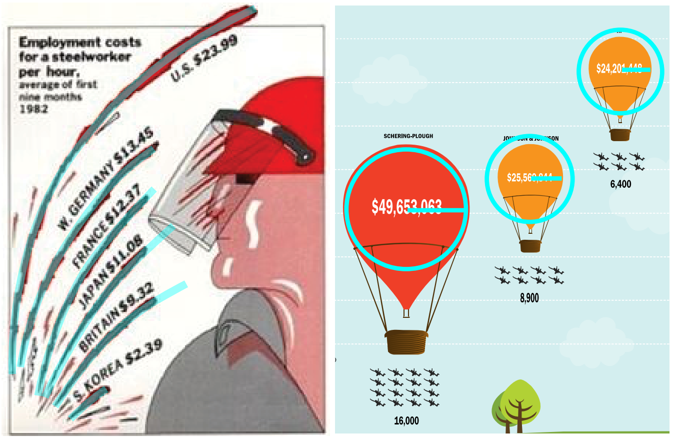
Fig. 12: Examples created with DDG. (a) An isotype chart using data guides to measure the heights of imported building icons. (b) An area chart using a single stroke to draw the area and to encode slopes in the declining trend. (c) A sankey-style diagram where two DDG are juxtaposed to compare the rankings of two different metrics. (d) A radial chart created using the radial layout function we provide. (e) Nigel Holmes' factory worker chart using curved DDG to encode data. The incorrect representation in the original chart is fixed in our version. (f) A flower chart using a parent DDG to encode stems, while multiple child DDG are used for flowers (i.e., hierarchical dataset). (g) A balloon chart using area DDG for the size of balloons and position DDG for the location of the balloons. (h) A cloud and chimney chart where four DDG created from the same dataset are used to encode each cloud. (i) A customized isotype chart using both length and area DDG to encode a pregnant woman's height and belly respectively.

infographics [21] that were used in a previous evaluation study investigating the effects of visual embellishment [1]. For the factory worker chart (Fig. 12e), data guides were adjusted to be curved lines on which gushed lava was drawn; however, the curved lines may not be desirable in most other cases. The monster chart was created differently (Fig. 1). Instead of drawing it from scratch, we imported a monster graphic into the tool and repurposed it with data guides by adjusting the teeth to match the size of the guides. We can then reuse the chart for different datasets by simply copying and pasting the chart (Fig. 1).

More complex visual structures can be constructed by combining multiple guides. In Fig. 12i, two groups of data guides were combined side-by-side (i.e., each area guide was linked to each length guide). A human shape was then linked to an area and length guide and repeated for sibling area and length guides; i.e., both the area and length guides act as the backbone of the shape. Another example is the flower chart (Fig. 12f) where all guides in each child group (flower) are linked to each guides in the parent group (stem). The guides in the child group were laid out using the radial layout function.

The graphics will be dynamically updated based on the changes in the underlying dataset. For example, when the dataset for the bottom guide group in Fig. 12f is changed, the positions of the flowers as well as the sizes of the stems will be updated accordingly. Likewise, changing the dataset for the length guide group in Fig. 12i will update the positions of the heads and the heights of the bodies, while the sizes of the bellies will remain the same. However, the current version of DDG does not handle the cardinality change in the dataset well and we discuss this problem in the limitation section (e.g., inserting additional data values in Fig. 12c).

We also found that some existing infographics were potentially inaccurately designed. For example, when we juxtaposed data guides on top of the original image we found that the factory worker chart (Fig. 13a) by Nigel Holmes may have an incorrect representation of the data. We also found a similar case in the balloon chart (Fig. 13b); that is, the radius of the balloon instead of the area was used to represent the data value. This case is actually a commonly found mistake in existing infographic design practice.



(a) (b)

Fig. 13: (a) When recreating the factory worker chart, we found that the lengths of three lava marks representing France, Japan, and Britain do not match the size of data guides; the baseline is not clear however. (b) With DDG, we found that the radius of balloons was used instead of the area.

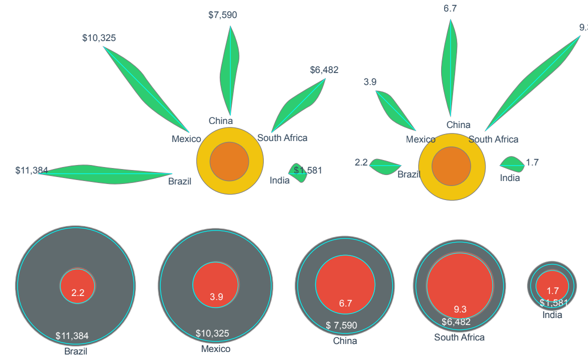


Fig. 14: Participant-generated graphics in the third task in the user study, further embellished by the first author.

7.2 Usability Study

To evaluate the usability of DDG for designing information graphics, we conducted a user study with 13 designers currently enrolled in professional schools of various design disciplines (e.g., architecture, design technology, urban planning, information design). Based on the pre-study survey, 5 participants (E1, E3, E4, E7, E11) had more than six years of experience in graphic design as well as information design and visualization, while 4 participants (E9, E12, E13) had less than two years of experience in both fields (i.e., both fields had similar participant distributions); other participants (E2, E5, E6, E8, E10), lie in-between. When specifically asked, all participants mentioned that they mostly create charts and graphs as side work. This is in line with real-world designers such as those found on freelancing platforms (e.g., UpWork), who create not only infographics but also other graphic design works such as logos or posters. In terms of frequently used design tools, participants specified that they used vector drawing tools, image editing tools, presentation software, and spreadsheets in the order listed. Only two participants had experience with programming. Each participant received a $20 gift card for their time.

Procedure. A 60-minute study session started with a 15 minute tutorial introducing the tool interface and a handful of examples demonstrating how to work with DDG (e.g., drawing data graphics from scratch and repurposing existing artworks). Using the datasets for the CO2 emissions and GDP of G10 countries, participants were first asked to recreate two graphics (similar to shown in Fig. 3). In the third task, they were asked to create their own graphic using a smaller dataset of G5 countries. All the datasets were extracted from the World Bank Open Data [14]. The first two tasks were intended to make sure that participants experienced all aspects of DDG, while the last task was to see whether DDG enabled expressive infographic design. The study setup was informal, allowing participants to interrupt at any time to ask questions during the tasks. They were asked to complete a post-study survey and were debriefed at the end.

All participants completed the two replication tasks with minimal guidance, taking roughly 15 minutes in total. The third task was open-ended and often involved an interview-like conversation between participants and the study moderator to understand their thought processes and derive useful feedback for the tool. The tasks were not strictly timed. In 5-point Likert scale questions during the post-study survey (1-strongly disagree, 5-strongly agree), participants highly rated their experience with DDG: interactions with DDG were intuitive (μ=4.0, σ=0.71), DDG is useful for positioning and measuring custom shapes based on data compared to rulers or grids (μ=4.7, σ=0.63), DDG is useful for designing creative and expressive infographics (μ=4.9, σ=0.38), and DDG would improve their current design practice of creating custom data graphics (μ=4.5, σ=0.81).

[14] `http://data.worldbank.org/`

The overall reactions from the participants were very promising as well. Although we did not specifically ask during the study, almost all participants said that drawing with DDG was fun and enjoyable (e.g., E6: *"I am having so much fun with this!"*). In addition, most participants asked us whether the tool was available for use and said they would like to use it when available (e.g., E4: *"I'm looking forward to a live release some day."*). Some participants suggested that DDG could be implemented in existing graphic design tools (e.g., E8: *"I like the idea of data guide, and it could even be a function or a plug in in other programs potentially."*). Another participant (E3) who studies architecture said that *"I think that this would be a wonderful aid in creating graphics for architectural representations as well."*.

They also made other positive comments in the post-study survey. Two participants commented regarding the tool interface that *"data guides provide a simple and straightforward interface"*(E1), and *"overall it was very intuitive, and it has a wonderfully simple and pleasing interface."*(E2). Similarly, two participants also commented on the usefulness of DDG: *"even though the tool has some bugs, it would already be a huge improvement to the infographic workflow."*(E6) and *"this would be an incredibly useful tool for making infographics, there's almost no learning curve."*(E8). Others expressed more specific use cases that *"data guides would allow me to represent data in a much more compelling and highly consumable way. I need this in my life!"*(E9) and *"currently, I need a calculator to make graphics that respond to data in Illustrator and you have to modify each element individually, which is pretty arduous. This tool makes it much easier to try things out and experiment with the graphics."*.

We also learned lessons that imply further necessary improvements to DDG. First of all, although participants found the interface was easy-to-use, we observed that they often struggled with keeping tracking of the links with data guides. Although we highlight linked items and provide a panel for inspecting the links (Figure 3e), the learning curve for this seems to be steep for beginners. This was especially noticeable in the third task where participants tried to create complex structures. During the study, we also observed instances where participants attempt to create a cohesive layout using two groups of guides while we do not have an appropriate support. Another shortcoming was that the result of shape deformation was not always perfect as it is still in the development phase. Other minor issues included missing features that were available in other full-fledged graphic design tools such as Adobe Illustrator. Since our focus was not to recreate an existing design tool, we ignore these issues in this work. One participant (E12) made the interesting suggestion that it would be useful to have a variety of templates while also allowing users to create custom visual structures. Lastly, one participant also expressed her concern that *"I usually do very analytical infographics, using traditional forms like bars or circles. Because of that, I'm not quite sure if data guides might be very useful"* (E5).

8 DISCUSSION

8.1 Benefits and Challenges of Manual Visual Encoding

An inherent benefit of allowing manual visual encoding is creative freedom in design, which makes it possible to create a wide variety of visual representations of data. In the third design task in the user study, we observed that none of the designs created by participants were the same although they were not fully embellished due to the time constraint. While our study is limited to a graphic design environment, previous research similarly found that tool flexibility led to more expressive designs, in which pencil & paper sketching [51] and tangible blocks [25] were used to construct a visualization. A potential advantage of having flexibility in design is the ability to develop context-specific visual representations. In a sense, this characteristic differentiates infographics from traditional visualization techniques that are generalizable across different datasets but often impose a loss of data context at the same time.

A main downside of manual encoding of data is that the process is tedious and time-consuming especially when dealing with precision or large data. While existing visualization creation tools address this problem by automating visual mapping process, we approach it in a very different way by providing helper guides driven by data. The guides are inherently different from construction user interfaces used in the existing tools, which usually enforce a rigid order of operations and do not give the feeling of directness. While we focused on supporting three visual variables—length, area, and position, it would be interesting to think about how to generalize the concept to other variables such as color, slope, or angle.

The flexibility of DDG comes with a caveat, however. It is still possible for users to create inaccurate representations as DDG does not prevent it. It would be beneficial to have intelligent agents or systems that provide design critiques based on design principles established in the visualization community [35] (e.g., calculating areas of visual marks to check whether they match actual data represented through data guides).

8.2 Opportunities for new visualization design tools

There is still a much unexplored gap in how designers create innovative visualizations and how currently available tools mandate the process of generating visualizations. Most existing visualization creation tools are based on formal specifications for rapidly generating traditional statistical graphics. However, designers still engage in manual encoding in order to design unique visual representations of data that are often found to be more attractive, engaging and easier to remember. In the formative study, we only investigated a subset of infographic authoring processes in the wild, which informed our design goals and led to the development of DDG. Novel infographics often involve a wide variety of different authoring techniques, most of which are still unknown to the visualization research community. Further investigations will be necessary to address the challenges and unearth the benefits of such visual mapping techniques, which will also provide insights on new visualization design tools.

DDG can be considered as a constraint-based drawing technique in a sense that the form and size of a data guide constrain the appearance of an associated shape. The use of a deformation technique enabling fine-grained constraint behavior differentiates our work from existing constraint-based drawing tools that mostly provide object-level constraint transformations (i.e., translate, rotate, scale constrained objects). In the same vein, there are still opportunities incorporating different design paradigms such as parametric drawing [15] into visualization design environments. Directly manipulating geometry using data as parametric constraints might be a possible solution in this direction.

The most common reaction we observed during the user study was that the participants seemed to enjoy working with DDG particularly in the third creative design task regardless whether they succeeded or not. This suggests that there may be an interesting avenue for developing creativity support tools for data graphic design [42]. For example, how can we develop computational tools to support freeform data sketching [16] or tangible visualization construction [17] whose inherent expressivity makes it appropriate for creative activities? Although such tools may not serve analytic purposes, they may find their usefulness in casual or personal contexts. For instance, a digital alternative to the use of tangible tokens [25] can make use of a large database of icons [18] as tokens to create diverse isotype charts.

9 LIMITATIONS

DDG has inherent limitations. First, DDG currently operates on a simple tabular dataset. Therefore, it is impossible to create certain types of charts that work on multivariate data (e.g., scatter plots) and graph data (e.g., networks). For such complex data structures, it may make more sense to create visualizations automatically, and then manually embellish them for communication [44]. Second, data guides allow freeform manipulations for flexible layouts, meaning that they may not be always axis-aligned. This makes it difficult to not only generate guide elements such as axes but also determine the position and orientation of a new guide when a new column is added to the dataset. In the similar vein, because of the flexibility in constructing novel visual structures, DDG is currently limited in supporting the data cardinality change (i.e., ambiguities in whether it is necessary to automatically generate marks and links for new guides). Lastly, DDG currently only supports length, area and position visual variables, requiring manually encoding other frequently used variables such as color.

Most of the limitations we found through the user study were related to the tool maturity. First, the current interaction model for linking shapes to a data guide through the context menu and keeping track of the links on the inspection panel needs further refinements. That is, the direct manipulation of data guides successfully reduced the gulf of execution, but there is still room for narrowing the gulf of evaluation in assessing the link states. Second, our tool does not provide advanced layouts of data guides except the simple linear and radial layouts. Currently, manually manipulating a large number of guides is cumbersome especially if it involves more than one group of guides. Lastly, the user study was rather limited, calling for more focused and task-oriented studies to better evaluate the effectiveness of DDG.

10 CONCLUSION AND FUTURE WORK

In this paper, we introduce DDG, an interaction technique for designing custom data-driven graphics. DDG is designed to addresses issues in the current design practice where designers manually encode data into custom graphics. Unlike traditional guides such as rulers or grids, data guides are generated from data and enable direct manipulation for intuitive interaction. DDG maintain a flexible design process by allowing users to draw custom shapes on top of guides or to use the guides to repurpose existing artworks. DDG's data binding support for freeform shapes further improves the design process by alleviating manual encoding when data is changed as well as increasing the reusability of custom charts. We demonstrate the expressiveness and usability of DDG through example graphics and a user study.

For future work, we plan to improve the accessibility of DDG by addressing the limitations we learned from the user study, including creating links and assessing the link states, coordinating the layouts of multiple groups of data guides, and providing predefined guide structures for novice users. We believe that there are still many opportunities in this less-studied area as we have outlined in the discussion and limitation section. We plan to further investigate the existing infographic design practice, focusing on the visual mapping process, in order to inform the design of next generation visualization design tools.

ACKNOWLEDGMENTS

The authors wish to thank Jean-Daniel Fekete, Jeremy Boy, Johanna Beyer, Kasper Dinkla, Hendrik Strobelt, and James Tompkin for valuable feedback on this project. This work was supported in part by a grant from the Kwanjeong Educational Foundation.

[15] http://paradrawing.com/

[16] http://www.dear-data.com/

[17] http://dataphys.org/

[18] https://thenounproject.com/

References

[1] S. Bateman, R. L. Mandryk, C. Gutwin, A. Genest, D. McDine, and C. Brooks. Useful junk?: the effects of visual embellishment on comprehension and memorability of charts. In *Proc. of CHI*, pages 2573–2582. ACM, 2010.

[2] J. Bertin. Semiology of graphics: diagrams, networks, maps. 1983.

[3] A. Bigelow, S. Drucker, D. Fisher, and M. Meyer. Reflections on how designers design with data. In *Proc. of AVI*, pages 17–24. ACM, 2014.

[4] A. Bigelow, S. Drucker, D. Fisher, and M. Meyer. Iterating between tools to create and edit visualizations. *IEEE TVCG*, page in press, 2016.

[5] R. Borgo, A. Abdul-Rahman, F. Mohamed, P. W. Grant, I. Reppa, L. Floridi, and M. Chen. An empirical study on using visual embellishments in visualization. *IEEE TVCG*, 18(12):2759–2768, 2012.

[6] M. A. Borkin, Z. Bylinskii, N. W. Kim, C. M. Bainbridge, C. S. Yeh, D. Borkin, H. Pfister, and A. Oliva. Beyond memorability: Visualization recognition and recall. *IEEE TVCG*, 22(1):519–528, 2016.

[7] M. A. Borkin, A. A. Vo, Z. Bylinskii, P. Isola, S. Sunkavalli, A. Oliva, and H. Pfister. What makes a visualization memorable? *IEEE TVCG*, 19(12):2306–2315, 2013.

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

[9] J. Boy, F. Detienne, and J.-D. Fekete. Storytelling in information visualizations: Does it engage users to explore data? In *Proc. of CHI*, pages 1449–1458. ACM, 2015.

[10] P. Castells, P. Szekely, and E. Salcher. Declarative models of presentation. In *Proc. of IUI*, pages 137–144. ACM, 1997.

[11] 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.

[12] J.-D. Fekete. The infovis toolkit. In *Prof. IEEE InfoVis*, pages 167–174, 2004.

[13] I. Fujishiro, Y. Ichikawa, R. Furuhata, and Y. Takeshima. Gadget/iv: a taxonomic approach to semi-automatic design of information visualization applications using modular visualization environment. In *Proc. of IEEE InfoVis*, pages 77–83, 2000.

[14] L. Grammel, C. Bennett, M. Tory, and M.-A. Storey. A Survey of Visualization Construction User Interfaces. In *EuroVis - Short Papers*, pages 19–23. The Eurographics Association, 2013.

[15] T. R. G. Green and M. Petre. Usability analysis of visual programming environments: a cognitive dimensions framework. *Journal of Visual Languages & Computing*, 7(2):131–174, 1996.

[16] S. Haroz, R. Kosara, and S. L. Franconeri. Isotype visualization: Working memory, performance, and engagement with pictographs. In *Proc. of CHI*, pages 1191–1200. ACM, 2015.

[17] L. Harrison, K. Reinecke, and R. Chang. Infographic aesthetics: Designing for the first impression. In *Proc. of CHI*, pages 1187–1190. ACM, 2015.

[18] L. Harrison, F. Yang, S. Franconeri, and R. Chang. Ranking visualizations of correlation using weber's law. *IEEE TVCG*, 20(12):1943–1952, 2014.

[19] J. Heer and M. Bostock. Declarative language design for interactive visualization. *IEEE TVCG*, 16(6):1149–1156, 2010.

[20] J. Heer, S. K. Card, and J. A. Landay. Prefuse: a toolkit for interactive information visualization. In *Proc. of CHI*, pages 421–430. ACM, 2005.

[21] N. Holmes. *Designer's guide to creating charts & diagrams*. Watson-Guptill, 1984.

[22] S. C. Hsu and I. H. Lee. Drawing and animation using skeletal strokes. In *Proc. of ACM SIGGRAPH*, pages 109–118. ACM, 1994.

[23] J. Hullman, E. Adar, and P. Shah. Benefitting infovis with visual difficulties. *IEEE TVCG*, 17(12):2213–2222, 2011.

[24] S. Huron, S. Carpendale, A. Thudt, A. Tang, and M. Mauerer. Constructive visualization. In *Proc. of DIS*, pages 433–442. ACM, 2014.

[25] S. Huron, Y. Jansen, and S. Carpendale. Constructing visual representations: Investigating the use of tangible tokens. *IEEE TVCG*, 20(12):2102–2111, 2014.

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

[27] A. Jacobson, I. Baran, J. Popović, and O. Sorkine. Bounded biharmonic weights for real-time deformation. *ACM Trans. Graph.*, 30(4):78:1–78:8, July 2011.

[28] A. Jacobson, Z. Deng, L. Kavan, and J. Lewis. Skinning: Real-time shape deformation. In *ACM SIGGRAPH 2014 Courses*, 2014.

[29] Y. Jansen and P. Dragicevic. An interaction model for visualizations beyond the desktop. *IEEE TVCG*, 19(12):2396–2405, 2013.

[30] R. Kazman and J. Carriere. Rapid prototyping of information visualizations using vanish. In *Proc. of IEEE InfoVis*, pages 21–28, 1996.

[31] R. Kosara. Presentation-oriented visualization techniques. *IEEE CG&A*, 36(1):80–85, 2016.

[32] R. Kosara and J. Mackinlay. Storytelling: The next step for visualization. *Computer*, (5):44–50, 2013.

[33] B. Lee, R. H. Kazi, and G. Smith. Sketchstory: Telling more engaging stories with data through freeform sketching. *IEEE TVCG*, 19(12):2416–2425, Dec. 2013.

[34] S. Liu, A. Jacobson, and Y. Gingold. Skinning cubic bézier splines and catmull-clark subdivision surfaces. *ACM Trans. Graph.*, 33(6):190:1–190:9, Nov. 2014.

[35] K. Luther, J.-L. Tolentino, W. Wu, A. Pavel, B. P. Bailey, M. Agrawala, B. Hartmann, and S. P. Dow. Structuring, aggregating, and evaluating crowdsourced design critique. In *Proc. of CSCW*, pages 473–485. ACM, 2015.

[36] J. Mackinlay. Automating the design of graphical presentations of relational information. *ACM Trans. Graph.*, 5(2):110–141, 1986.

[37] A. V. Moere and H. Purchase. On the role of design in information visualization. *Information Visualization*, 10(4):356–371, 2011.

[38] B. A. Myers, J. Goldstein, and M. A. Goldberg. Creating charts by demonstration. In *Proc. of CHI*, pages 106–111. ACM, 1994.

[39] D. Ren, T. Hollerer, and X. Yuan. ivisdesigner: Expressive interactive design of information visualizations. *IEEE TVCG*, 20(12):2092–2101, 2014.

[40] S. F. Roth, J. Kolojejchick, J. Mattis, and J. Goldstein. Interactive graphic design using automatic presentation knowledge. In *Proc. of CHI*, pages 112–117. ACM, 1994.

[41] A. Satyanarayan and J. Heer. Lyra: An interactive visualization design environment. *Computer Graphics Forum (Proc. EuroVis)*, 33(3):351–360, 2014.

[42] B. Shneiderman. Creativity support tools: Accelerating discovery and innovation. *Communications of the ACM*, 50(12):20–32, 2007.

[43] D. Skau, L. Harrison, and R. Kosara. An evaluation of the impact of visual embellishments in bar charts. *Computer Graphics Forum (Proc. EuroVis)*, 34(3):221–230, 2015.

[44] A. S. Spritzer, J. Boy, P. Dragicevic, J.-D. Fekete, and C. M. Dal Sasso Freitas. Towards a smooth design process for static communicative node-link diagrams. *Computer Graphics Forum (Proc. EuroVis)*, 34(3):461–470, 2015.

[45] C. Stolte, D. Tang, and P. Hanrahan. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *IEEE TVCG*, 8(1):52–65, 2002.

[46] S. Swaminathan, C. Shi, Y. Jansen, P. Dragicevic, L. A. Oehlberg, and J.-D. Fekete. Supporting the design and fabrication of physical visualizations. In *Proc. of CHI*, pages 3845–3854. ACM, 2014.

[47] J. Talbot, V. Setlur, and A. Anand. Four experiments on the perception of bar charts. *IEEE TVCG*, 20(12):2152–2160, 2014.

[48] E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, USA, 1986.

[49] F. B. Viegas, M. Wattenberg, F. Van Ham, J. Kriss, and M. McKeon. Manyeyes: a site for visualization at internet scale. *IEEE TVCG*, 13(6):1121–1128, 2007.

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

[51] J. Walny, S. Huron, and S. Carpendale. An exploratory study of data sketching for visual representation. *Computer Graphics Forum (Proc. EuroVis)*, 34(3):231–240, 2015.

[52] C. Weaver. Building highly-coordinated visualizations in improvise. In *Proc. of IEEE InfoVis*, pages 159–166.

[53] H. Wickham. *ggplot2: elegant graphics for data analysis*. Springer Science & Business Media, 2009.

[54] L. Wilkinson. *The Grammar of Graphics (Statistics and Computing)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[55] T. Wun, J. Payne, S. Huron, and S. Carpendale. Comparing Bar Chart Authoring with Microsoft Excel and Tangible Tiles. *Computer Graphics Forum (Proc. EuroVis)*, 2016.