DataInk: Direct and Creative Data-Oriented Drawing

Haijun Xia[1], Nathalie Henry Riche[2], Fanny Chevalier[1,3], Bruno De Araujo[1], Daniel Wigdor[1]

[1]University of Toronto
{haijunxia|fanny|brar|daniel}@dgp.toronto.edu

[2]Microsoft Research
nath@microsoft.com

[3]Inria

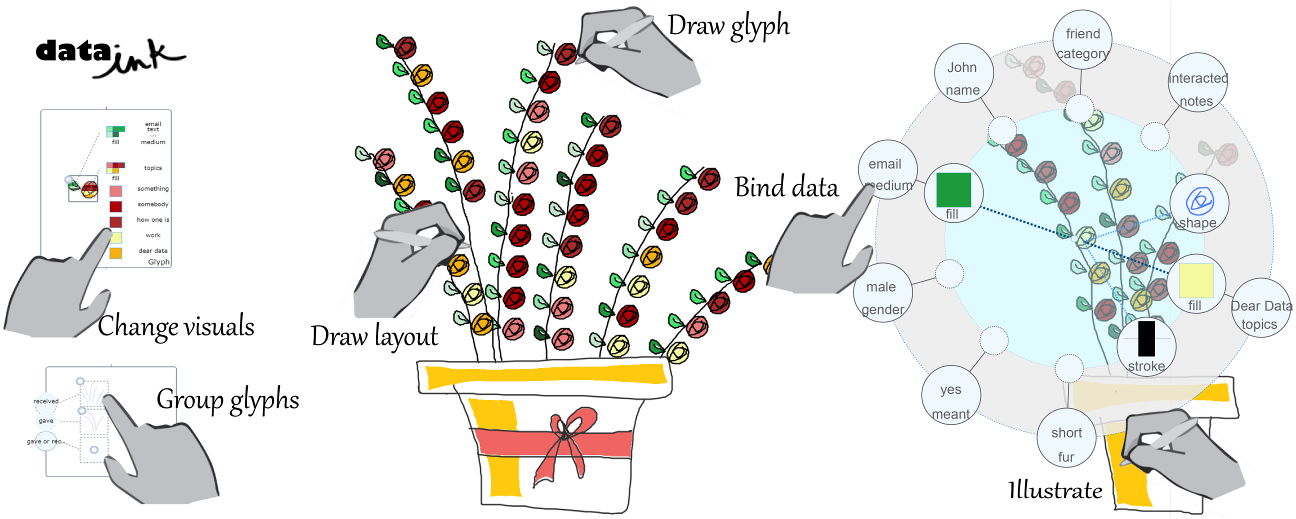

Figure 1 DataInk enables easy creation of visually creative data visualizations. Visualization authors can interleave illustration (right, lower), data binding (right, upper), layout configuration (center, middle) to quickly create visualizations that can be iteratively specified through direct manipulation on the levels of data point (right) or data dimension (left).

ABSTRACT

Creating whimsical, personal data visualizations remains a challenge due to a lack of tools that enable for creative visual expression while providing support to bind graphical content to data. Many data analysis and visualization creation tools target the quick generation of visual representations, but lack the functionality necessary for graphics design. Toolkits and charting libraries offer more expressive power, but require expert programming skills to achieve custom designs. In contrast, sketching affords fluid experimentation with visual shapes and layouts in a free-form manner, but requires one to manually draw every single data point. We aim to bridge the gap between these extremes. We propose *DataInk*, a system supports the creation of expressive data visualizations with rigorous direct manipulation via direct pen and touch input. Leveraging our commonly held skills, coupled with a novel graphical user interface, *DataInk* enables direct, fluid, and flexible authoring of creative data visualizations.

CHI 2018, April 21–26, 2018, Montreal, QC, Canada

ACM ISBN 978-1-4503-5620-6/18/04…$15.00
https://doi.org/10.1145/3173574.3173797

INTRODUCTION

Visual representations of data are a powerful communication medium for presenting insights and ideas in an understandable form. Beyond traditional information visualization, designers, artists and enthusiasts alike also leverage this mode of expression to craft meaningful, beautiful, and memorable pieces [10, 14, 29, 33, 49]. While compelling, authoring visualizations that embody creative, artistic influences is presently a laborious task, typically requiring one to alternate between different tools to form an enticing, yet factual representation of the data [4].

Tools that aim at supporting design creativity in information visualization should encompass *design expression*, i.e., the crafting of highly customized and stylized visuals and *rigorous execution* of the principles of information visualization to ensure that the visual properties of visuals comply with data they encode [5]. Our goal is to also encompass support for creativity [48], and thus empower users with *creative exploration* through the fluid and spontaneous experimentation with glyphs and layouts, to facilitate ideation and iterative design.

While several tools exist to assist with the generation of data visualizations, there is a lack of software that supports design creativity. The tools used by visualization authors today typically enable for either rigorous visualization, or creative design expression, but none address our goal of facilitating data visualization as both a medium and a tool for creative expression.

Visualization toolkits (e.g. D^3 [8]), graphical interfaces (e.g. Lyra [46]) and online applications (e.g. RAWGraphs [16]) address the rigorous execution of visualizations by enabling users to automatically apply visual encodings to their data. These tools offer different trade-offs between power and simplicity of use. Programming toolkits enable for the implementation of any custom data visualization, however often have a steep learning curve. Online applications are limited to the creation of a small set of predefined, mutable data visualizations. The lack of flexibility to customize visuals in a fluid manner makes these tools poorly suited for creative exploration and design expression.

In contrast, graphic design and illustration software such as Adobe Illustrator [1] enables rich visual expression, but lacks support to leverage data bindings to aid in drawing. Further, the lack of support for bindings hampers iterative design, since simple changes require significant work. The visualization research community has recently started to address this by bridging diverse types of interfaces [4] and augmenting illustration software with specific widgets [27]. These latter efforts, however, rely on a rigid, complex workflow that hampers creative exploration. This work thus set out to support design expression and rigorous execution while offering flexibility, plasticity and freedom to manipulate for creative exploration via iterative design.

The inherent freeform nature of sketching enables visualization authors to follow their inspiration and quickly model and remodel different designs, or discard them without second thoughts [6, 42]. Such unconstrained means of expression can also foster an author's expressivity and promote understanding when drawing with data [56]. We articulate a set of five design decisions rooted in the literature of supporting creativity via direct manipulation and contextualized in the domain of data visualization. Embodying these design decisions, DataInk [1], a pen-and-touch enabled user interface that leverages drawing on a digital canvas and the use of direct manipulation to seamlessly access visual properties of graphics and bind them to data. A user study with eight designers and non-experts demonstrates that DataInk affords direct, fluid, and flexible interactions for the authoring of creative and expressive data visualizations.

RELATED WORK

Many authoring tools have been proposed to assist users in creating visualizations for their data. Creating a visual representation of data requires one to specify a *visual-data mapping*: associating data dimensions to visual variables such as color, shape, size, or position [3, 11]. This encoding produces *marks* that visually represent the data, as a one-to-one correspondence between each data point and its visual representation (i.e. glyph-based representation) or as a groups of data points (e.g. bar charts). Grammel et al. [18] surveyed visualization and HCI publications, and reported on authoring strategies of over 60 visualization creation tools. Online tools also proliferate and spark practitioners' discussions on their merits and limitations [43]. We review visualization authoring tools with regard to the purposes they address, delineating the gap our work fills.

Rigorous Execution

Deeply rooted in scientific practice—where visualizations are seen as functional assets for analysis—the vast majority of tools developed by the research community support the systematic building of a visualization from data. This ensures that the visual properties of marks scrupulously comply to the data they encode (i.e., rigorous execution).

Visualization toolkits and charting libraries (e.g., D3.js [8], Vega [45], ggplot2 [58]), and advanced systems for data analysis (e.g., Microsoft Excel [37], Tableau [50], PowerBi [38], Lyra [46], iVisDesigner [40]) afford some level of design expression. However, creating custom graphics and layouts requires substantial training such as learning to program, or having good command of the vast functionality present in a full-featured graphical user interface. Trading power for simplicity, many online systems (e.g. ManyEyes [54], EasyCharts [17], RAWGraphs [16]) enable non-expert audiences to create visualizations of their data in just a few steps, i.e., load data, choose a template, and select the data dimensions to represent. The range of possibilities these applications support, however, is limited to commonplace visualizations. The integration of custom glyphs and layouts is usually limited or not supported.

In such tools, creative exploration is hindered due to the large number of steps required to generate, evolve, and refine design alternatives. Moreover, most design decisions have to be made before the user can see an actual visual representation of their data. Users not versed in the art of crafting data visualizations may not fully comprehend what each decision entails or how the series of decisions they made will impact the resulting visual representation [36]. Méndez et al. proposed iVolver [35] to alleviate the need to making decisions beforehand, following the principle of constructive visualization [21]. While this approach affords more expressivity than other systems, it still requires one to learn the workflow language to achieve the envisioned designs, making it difficult to pursue creative exploration.

Design Expression

Beyond their functional purpose, visual representations of data have increasingly been viewed as a means of expression, where pursuits for aesthetics can further yield unique, beautiful pieces [53]. Many designers and artists leverage design expression to craft personally-relevant and evocative visualizations susceptible to provoking emotional responses from the audience. One example demonstrating a high level of design expression is the Dear Data project [33], featuring a collection of hand-drawn data visualizations. Recent studies investigating visual thinking on whiteboards [9] and data sketching [56] highlighted the expressiveness of sketched visuals when working with data.

[1] http://datainkresearch.github.io

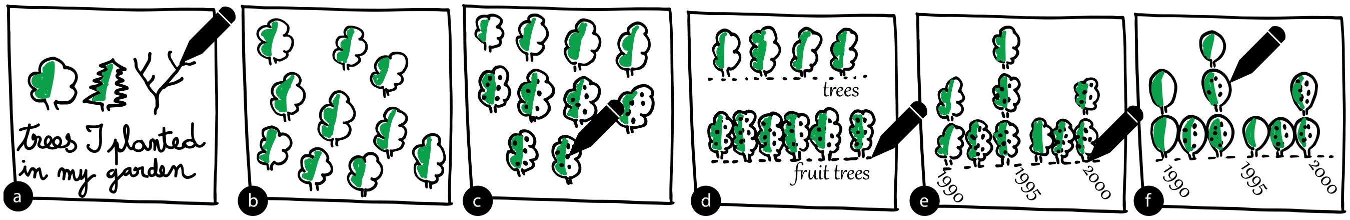

Figure 2 A storyboard illustrating data-oriented drawing with direct input following the principles of direct manipulation. a) Sketching visual designs using digital ink on a canvas enables one to experiment with various shapes. b) Selecting one of the shapes to specify a data-visual mapping automatically populates the canvas with all relevant data points. c) Sketching compound glyphs to represent additional data dimensions. d) Drawing a layout based on a data dimension to structure the data spatially. e) Redrawing the layout to map the data to a different data dimension. f) Redrawing a visual mark for a different data dimension.

Sketching is widely recognized as an excellent instrument for the quick generation of multiple designs due to its inherent freeform and effortless nature [6, 42]. Yet, the lack of automation makes it hard to envision how resulting visuals might be mapped to actual data.

The same issue holds for graphics design and illustration tools such as Adobe Illustrator [1], which focus on design expression but lack support for mapping realized visuals to data. Bigelow et al. [4] provide insights on the laborious process in which designers alternate back and forth between graphics authoring and data visualization tools to realize stylized visuals. Recent efforts have tried to simplify this process by providing bridges between tools [5] or integrating data-driven widgets in illustration interfaces [27]. Such efforts are great avenues to reconcile the expressive power of graphic design software to generate data visualizations. Yet, they impose a high threshold [51] on users, as they still require to master complex functionalities, making it laborious to generate alternative designs, which can inhibit creative exploration.

Creative Exploration

Pen and touch enabled interfaces leverage natural human sketching and physical manipulation skills, empowering users to pursue creative tasks in a fluid workflow [20]. The visualization community sees these interfaces as a promising alternative to WIMP UIs, allowing analysts to focus on the data under study, rather than how to operate the interface [30]. Several visualization tools take advantage of pen and touch input for data exploration [9, 15, 23, 44, 62] and its presentation to an audience [31, 32].

However, pen usage is often limited to making simple annotations on customized data views. Only rarely is the expressiveness of sketching exploited to create new simple stroke-based visuals (e.g. [31]). In the context of graphics authoring, direct manipulation approaches such as object-oriented drawing [60] have been introduced to facilitate the creative exploration of visual designs by providing a novel and more direct interaction metaphor to manipulate the visual properties of graphics. The present work continues to pursue this line of research, investigating the power of pen and touch interaction to enable users to create personalized, expressive data visualizations.

DATAINK AND DIRECT MANIPULATION

We share the spirit that direct manipulation can enable users to easily express their intent, and hence support their creativity [48]. However, creating a direct manipulation system that not only supports easy entry for novice users but also complex functionalities and flexible workflow for experts is very challenging. Despite the general claim of supporting direct manipulation, many of the systems are not direct enough, especially ones using WIMP UI [5, 27], suffering from the intrinsic indirectness of WIMP [2].

We seek to rigorously support direct manipulation to allow users to fluidly experiment with visual designs and visual-to-data bindings to create expressive data visualizations (Figure 2). Informed by literature, we articulate a set of five design decisions illustrated in Figure 3.

D1. Create and manipulate with direct input

We focus on a new UI with direct pen + touch input to leverage commonly held skills [52], while keeping the interaction vocabulary simple to ease the learning.

We also draw inspirations from designers' and artists' general workflows [4], where designs are first explored by sketching on paper or whiteboards. A similar workflow should be supported to foster the rapid experimentation with visual marks, by providing freeform sketching with direct pen input on a digital canvas (Figure 3:1a), through easy access to the visual properties of graphics via direct touch input (Figure 3:1b).

To compose compelling visual designs for a given dataset, authors usually take several iterations to explore different visual encodings [4]. To support this, the users should be able to access data dimensions and visual properties of graphics at any time through direct physical manipulation (Figure 3:1c) [20, 60].

D2. Interact with objects – data points as glyphs

To empower users to craft expressive visual representations of their data, we propose a focus on glyph-based visualizations [7]. In this type of visual representations, each data point is objectified as a *glyph*, which turns aggregation of abstract numbers into a physical object that one can direct interact with [60]. The visual properties of compound glyphs are dictated by one or more dimensions

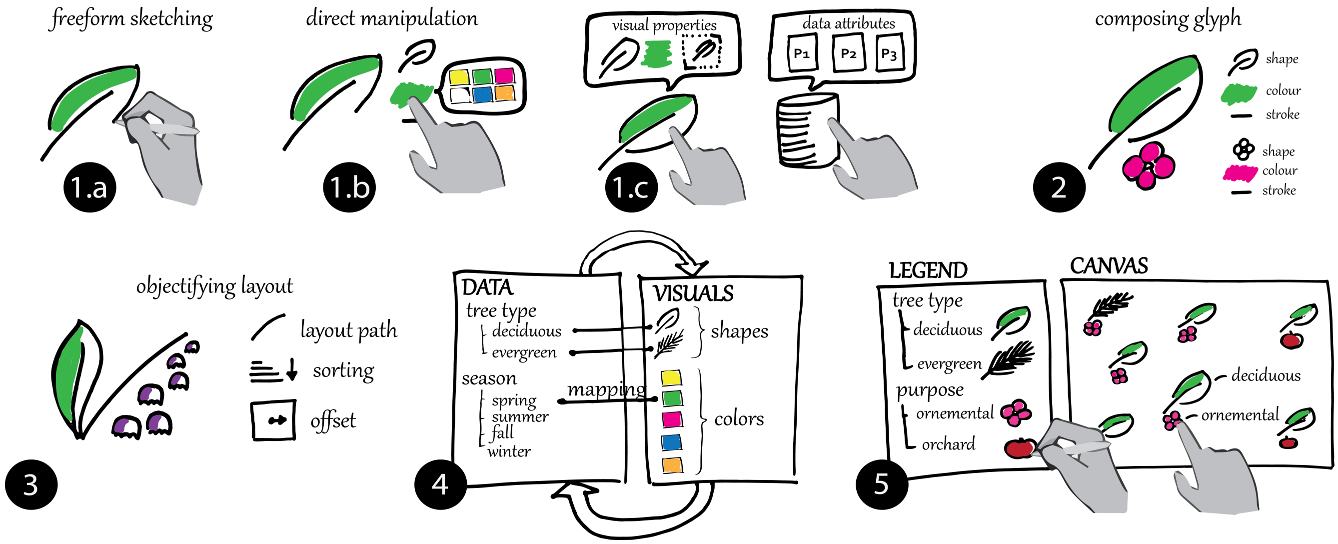

Figure 3 The design decisions of DataInk. 1) Freeform sketching (1.a), direct manipulation (1.b, 1.c) to enable flexible authoring of visual designs, and quick access to both visual properties of glyphs and data attributes. 2) Composing glyphs specifying visual properties to data dimensions. 3) creating layouts and editing them as objects. 4) specifying visual-data mapping from visual variables or data dimensions. 5) supporting multiple workflows from the legend or the canvas.

of the data point it represents. Glyph-based representations [34, 41] are commonly found in visualization for communication purposes [39] as they afford expressiveness in the composition of visual marks, while also offering a concise presentation of multivariate information. They also simplify the specification of the visual-data mapping since they do not require complex specification of grouping.

We propose to empower users to create visually-rich marks portraying multiple facets of data points (Figure 3:2), by iteratively incorporating components.

D3. Interact with objects – layouts as objects

Configuring data-driven layout is another challenging and abstract task. To make it as direct as possible, we propose objectify layout as objects, following the Object-Oriented approach [12, 60], to provide the suitable representation for interaction and reasoning. This suggests considering the layout as a visual mark that a user can directly draw on the canvas, manipulate the properties of, and bind a data dimension to. The visual mark can also serve to select groups of glyphs for direct manipulation. We see properties of the layout such as the distribution path and the order of glyphs along the path similarly to the visual properties of glyphs (e.g. shape and color).

We seek to empower users to create a spatial structure encoding different dimensions of the data by iteratively composing the outcome they envision (Figure 3:3).

D4. Support flexible workflow – bidirectional mapping

The realization of a data visualization requires one to specify a set of mappings between data dimensions and visual variables [13]. To provide maximum flexibility, we propose to enable users specifying visual-data mappings from either direction (Figure 3:4). In simple terms, a user may start from a visual property (e.g. shape, color) and experiment with different data dimensions it could encode; or they may start from a data dimension and experiment with different visual encodings that could represent it.

D5. Support flexible workflow – multiple abstraction levels

To foster flexible exploration [48], it is essential to support the workflows of creating data visualizations on the levels of data point and data dimension.

Data visualization experts, may use a generative approach, making decisions on the data dimension level. They may choose visual encodings for each data dimension and values successively (e.g., map each value of 'tree type' to a shape, then map each value of 'temperature' to a color, etc.). Put differently, they decide on a set of rules to generate glyphs: the *legend*. Designers on the other hand, may take more of a design-by-example approach, making decisions at the data point level, by iterating on the design of the final glyph for individual data points (e.g., draw a glyph representing a tree that grows in a warm environment and produces fruit). Put differently, they decide on the designs of glyphs first: the *visualization,* and derive generative rules later.

Supporting users to follow either one of these workflows or fluidly switch between them, we propose using a legend and visualization canvas metaphor (Figure 3:5). Interaction within the legend enables specification of mappings on the data dimension level, while interaction within the canvas enables them to specify mappings on the data point level.

To summarize, the five design decisions support the creative authoring [48] of data visualizations by providing:

- *low threshold*: easy entry for novices,
- *high ceiling*: experts to achieve sophisticated pieces,
- *wide walls*: a wide range of possible explorations,
- *many styles and many ways*: multiple workflows.

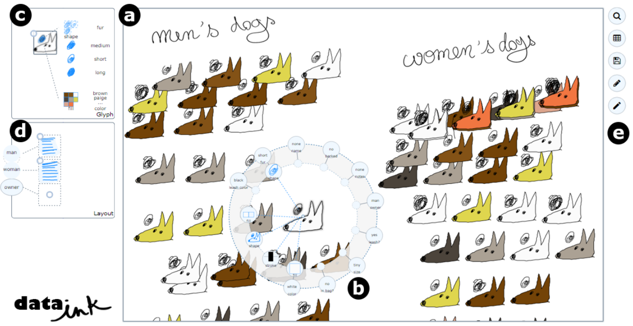

Figure 4 DataInk: a) the main canvas. b) the visual-data palette to edit visual properties and encode the mapping. c) glyph panel to define mappings like a legend. d) layout panel to structure data dimensions. e) menu to load data, search for vector graphic visuals and save visualizations.

DATAINK WORKFLOW

DataInk (Figure 4) is an application that incorporates our design decisions. We first describe the workflow to create a data-driven drawing, then delve into the design of each interface component.

To engage with her colleagues at her new job, Emma decided to create a fun, whimsical visual representation of her personal data. Inspired by Dear Data [33], she recorded the dogs she encountered during her daily commute during the week in a spreadsheet, where each row is a dog and each column represents one of its characteristics (e.g. color, length of fur, whether their owner was a man or a woman).

Emma opens DataInk on her pen-and-touch enabled tablet and loads her spreadsheet. She starts by experimenting with different dog shapes by drawing on the canvas (Figure 4:a). Looking at the different shapes she drew, she taps on her preferred design and binds it to data dimensions. The system instantaneously duplicates the dog drawing, and spread the newly created collection of dogs randomly on the canvas. Each drawing represents a data point.

As Emma taps on a dog, a visual-data palette (Figure 4:b) appears depicting the visual properties of the drawing in the inner ring, and the values for each data dimension in the outer ring. Working at the data point level, she can modify the color of her drawing from white to brown because the data value of this dog shows that it was brown. She then can create a mapping from the color visual encoding to the color data dimension by aligning the inner and outer rings.

The legend (Figure 4:c) thus updates to display the color mapping. Emma taps the legend to access the list of colors and sets them successively. She then decides to encode a second data dimension, the fur length, by adding a shape to her dog drawing. She taps a dog, draws on the canvas just above its head and map the shape of this mark to the corresponding attribute using the visual-data palette. She then taps a different dog, and changes the shape by drawing in a contextual interactive panel invoked directly from the touching the visual shape item in the palette or legend.

Satisfied by the design of her current glyph, Emma decides to organize the glyphs. She does not really know what data dimension would provide an interesting structure. Working at data dimension level, she taps the layout legend and browses through the different data dimensions (Figure 4:d). As the data dimensions are in focus, DataInk previews the different groupings in the canvas. Given the number of dogs she has in her visualization, Emma opts for structuring the space by the gender of the dog owners. The dogs appear now into two distinct groups on the canvas. Emma can touch each group and move them in the canvas. As she taps the layout legend, she can select a particular group and simultaneously draw distribution path in the canvas. This enables Emma to experiment with different layout shapes for each group. She concludes by saving her work and sending it to the large format printer (Figure 4:e). See supplemental material for the video of this scenario.

DATAINK USER INTERFACE

DataInk is composed of five UI components (Figure 4). The visualization canvas (Figure 4a) shows the generated visual representation of the data, as well as any visual elements not bound to data such as handwritten text or illustrative marks. A contextual visual-data palette appears when a user taps on a visual element, enabling them to set appropriate mappings (Figure 4b). The legend depicting the sets of mapping between visual encodings and data dimensions are grouped in two panels: one for the glyph (Figure 4c) and one for the layout (Figure 4d). A side menu (Figure 4e) allows the users to access functionalities such as searching vector graphics, loading data, saving the visuals, as well as providing access to several illustration tools.

SKETCHING

DataInk offers *freeform sketching* (D1) to enable users to ideate on graphical elements that can later serve to compose data visualizations. Users can generate these graphical elements by directly drawing on the canvas with a digital pen or writing in a search box to import vector graphics. Visual elements on the canvas can be accessed and modified by touch interaction. DataInk follows an object-oriented drawing approach [60], materializing properties of the visual elements as interactive cards that can be directly manipulated to adjust their values.

Figure 5 Object-Oriented Drawing: sketching or reusing visuals and directly editing visual attributes on the canvas.

Creating Data-Driven Glyphs

The users can select any visual marks to represent the data, using them as a glyph. To create a glyph, the user can first lasso the visual marks with the stylus to indicate a selection. She can then drag and drop the selected marks into the glyph panel to create a glyph. Alternatively, she can hold the glyph panel with her non-dominant hand, and lasso the visual marks with the stylus using her dominant hand, to transfer the selected visual marks into the glyph panel.

Immediately after the assignment, all the data points (e.g., rows of the data table) are represented by the glyph and spread on the canvas. Tapping each glyph on the canvas reveals the data values of the data point it represents, and existing visual encodings, if any, via a contextual visual-data palette (Figure 6).

Visual-Data Palette

We designed the visual-data palette to enable flexible *bidirectional mappings* between data dimensions and visual properties of the glyph.

Inspect and Edit the Attributes

The palette is composed of two rings (Figure 6), whereby the outer ring depicts a list of data values if this visual represents a data point, and the inner ring depicts a list of visual attributes for the selected elements. DataInk currently supports shape, fill, stroke, and size for strokes. Existing mappings are always shown to inform the users of how data dimensions have been encoded. The user can lasso select any element to inspect its visual attributes (Figure 7). Changes of the visual attributes of a data-driven glyph will be automatically applied to all others.

Create and Remove a Mapping

The user can specify a mapping by dragging a visual property (inner ring) and drop it into the slot next to the data property (outer ring). Once mapped, both items are visibly linked and form a new mapping object that rides across the rings. The user can break a mapping by holding and dragging either of its attribute off from the other. Creation and deletion of a mapping is automatically propagated to the corresponding visual attribute and data value in other data-driven glyphs. Changes of the visual attribute of a mapping will only be applied to glyphs that have the same data value. In Figure 6, the data dimension 'color' is encoded by the fill color of the shape.

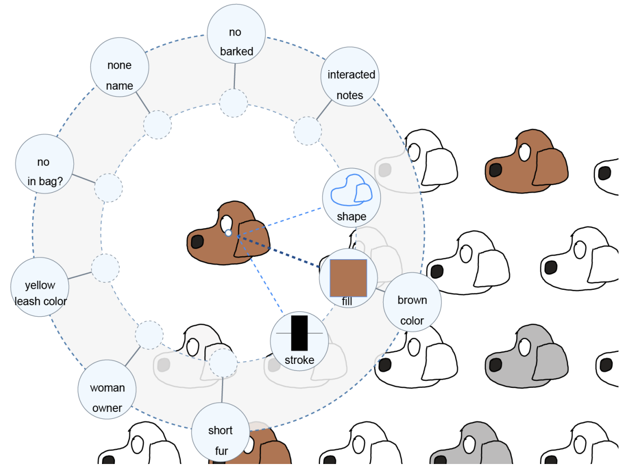

Figure 6 Visual-Data Palette - visual attributes distributed along the inner ring, data attributes along the outer ring.

Transfer a Mapping

A user may iterate over different encodings of a data dimension. For example, the user may want to experiment with encoding the data dimension 'color' to the fill color of a different stroke. Instead of removing the old mapping and creating a new one, she can transfer mappings she has previously created to new components. She can hold the existing mapping with one hand, and lasso select the strokes she likes with another hand to transfer the mapping to a new set of shapes (Figure 7). Similarly, this change is automatically propagated to all other glyphs.

Compose Compound Glyphs and Mappings

A user can sketch freely on the canvas, even on top of the glyphs as annotations. However, when the palette is activated, the user can directly draw on the glyph to add additional strokes to that glyph. The new strokes are again automatically prorogated to all the glyphs. This enables the users to iterate on the glyph design for complex visual effects even after the glyph is created.

DataInk builds upon the *aggregated attributes* from Collection Objects [61]. When multiple strokes are selected, the aggregated visual attributes are shown, each of which can be mapped to a data dimension. This allows for complex mappings between data properties and the visual properties of any sets of strokes. In Figure 6, the data dimension 'color' is mapped to the fill color of two shapes.

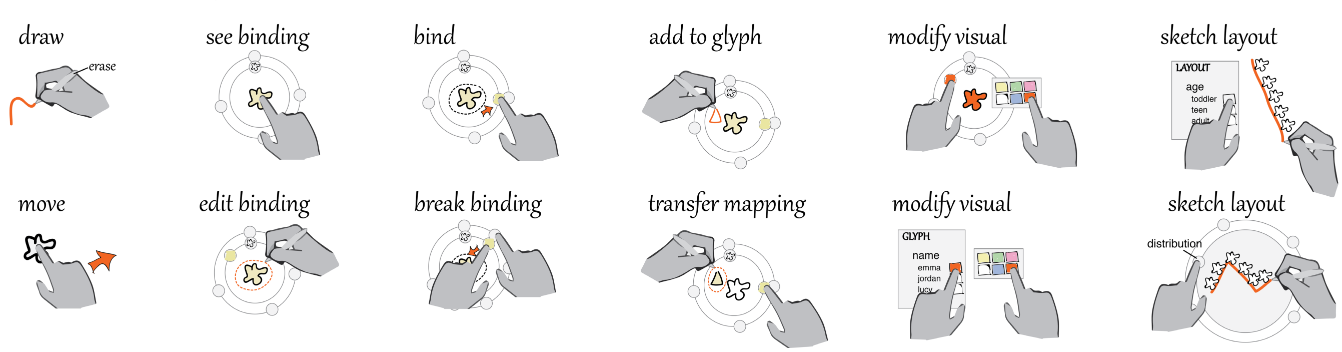

Figure 7 Pen and Touch Interactions in DataInk

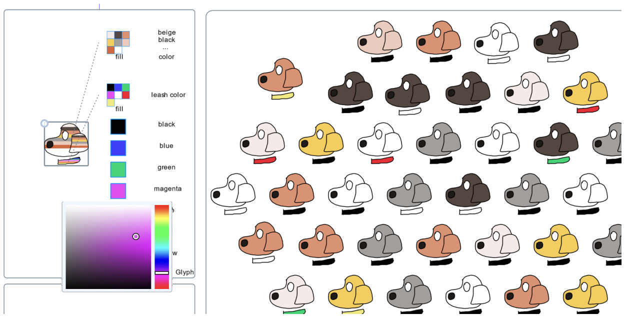

Figure 8 Editing visual properties from the glyph legend.

Legends

The visual-data palette enables users to reason at the data point level, on the canvas. To provide a *flexible workflow* (D5), DataInk also enables users to work on the data dimension level, inside the legends, which show the overview of all existing mappings. DataInk provides two types of legends: a glyph legend and a layout legend.

Glyph Legend

The glyph legend depicts all the visual-data mappings of the glyph (Figure 8). When the user creates a mapping using the visual-data palette, an aggregated mapping object is created inside the glyph panel. Tapping on the mapping object expands it and reveals the subordinate mapping objects for categorical data dimensions, the user then can set the mappings for each value of a data dimension successively (e.g., assigning a color to each data value; Figure 8). For quantitative data dimensions, a panel is shown for users to specify the mappings for a small set of values (min, max and several intermediate values) and interpolate the property for the entire continuous interval.

When configuring a mapping, a user may first work on the data point level, experimenting with visual-data palette on the canvas. Once deciding on the mapping, she can then work on the data dimension level to effectively map the complete set of data values. Interaction with the visual-data palette and the glyph panel can be interleaved, with the changes properly synchronized with each other.

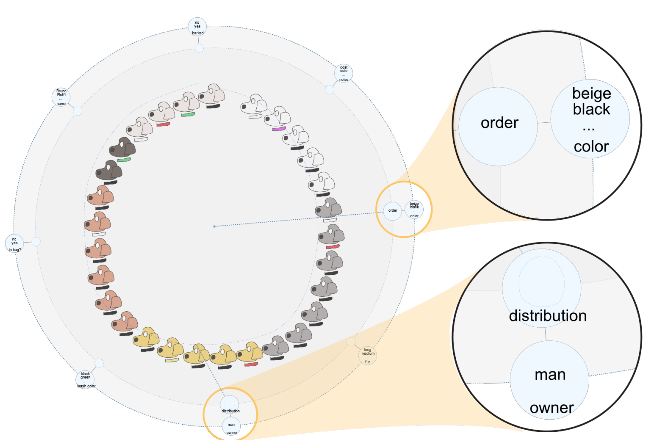

Figure 9 Visual-Data Palette of a group of glyphs, with two new structural attributes: distribution and sorting order.

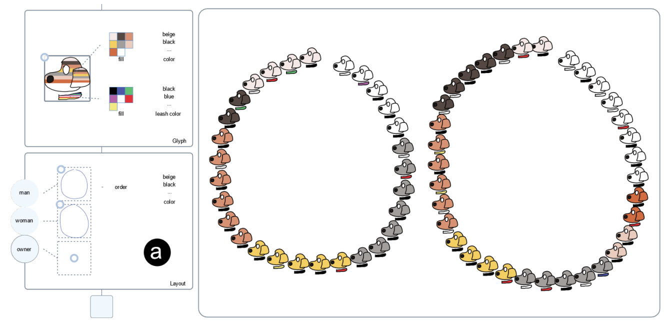

Figure 10 Specifying distribution path for a data dimension from the layout legend (a).

Layout Legend

The layout legend depicts the visual-data mappings for the layout of the visualization (Figure 10). The user can preview the effects of different groupings by browsing the set of data dimensions available from the data items on the left of the legend (Figure 10a). Once a data dimension is selected, glyphs on the canvas are separated into several groups according to the data values of the data dimension. Touching one of the values selects the corresponding set of glyphs on the canvas. Users can then hold the data values on the legend and draw distribution paths for each group directly on the canvas to distribute them.

We see a set of grouped glyphs as a glyph of a higher level. A group of glyphs can be freely moved on the canvas by dragging. Tapping on the group reveals the visual-data palette on the group level (Figure 9), with the aggregated data attributes on the outer ring, and two new structural attributes on the inner ring: the distribution geometry and the sorting order of the glyphs along the path. The user can modify the distribution of a group. Tapping on the distribution attribute calls out a drawing lens covering the group, where the user can erase and redraw new distribution paths. In a similar spirit to the glyph legend, additional structural properties of the layout, the distribution path and sorting order, are also visually represented in the legend.

DataInk supports flexible workflow on various levels of the glyphs on the canvas. The user may want to adjust the visual-data mappings of individual glyphs while working on the layout configuration of a group with the palette. She can dive onto the level of individual glyph by directly lassoing the glyph or its components. The palette will then shrink to the individual glyph level. After the changes, she can go back to the group level by tapping the group to expand the palette back to the group. The direct indication of objects and levels of interest as well as the rapid shrink and expansion of the palette, allow the users to flexibly switch their intention on different aspects of their design.

USER STUDY

We conducted a user study to gain insights into the potential of DataInk in supporting design expression, rigorous execution, and creative expression, as well as the usability of the system for authoring data visualizations. We targeted at designers to understand tradeoffs of DataInk compare to graphics design tools, and invited non-experts (limited experience in programming, data analysis and design) to assess the receptiveness by a general audience.

Apparatus

DataInk was implemented as a web application using D3.js, running on a Surface Studio, where finger and stylus input can be simultaneously detected and differentiated.

Participants

We recruited 8 participants (3 males; Age:18 to 39, Average: 26). Four of them (D1-D4) were designers who reported having more than 7 years of experience using digital drawing applications. The others were non-experts (P1-P4) who reported having no or very limited experience in digital drawing. All participants reported having no prior experience in creating data visualizations, except P2 who had experience using D3.js to create charts. Participants received $50 for an approximately 90-minute session.

Protocol

Participant first filled a consent form and a demographics questionnaire, then completed three tasks with DataInk, concluding by a feedback questionnaire and an interview.

Demonstration and Training (25 min).

The experimenter demonstrated the underlying concepts of DataInk, including basic sketching, glyph composition, visual-data mapping and layout via a walkthrough of an example. Participants were instructed to replicate the exact same example, seeking guidance whenever necessary. The experimenter also encouraged them to freely explore the interface to try every functionality.

Replication and Iteration Exercises (20 min)

The experimenter then proceeded to the replication task. This task consists in asking participants to replicate a data visualization for a new dataset without guidance. Participants received the target visualization on a paper sheet, along with an accompanying legend describing the visual encodings. We design this exercise to assess the learning curve necessary for people to use DataInk to recreate a design from scratch and bind data to its visual elements. The second task required participants to alter and iterate on their design, requiring them to make four changes to the glyph composition and three changes to the layout given an example provided by the experimenter.

Freeform Exploration (25 min)

After completing the replication and iteration exercises, the experimenter ask participants to design their own data visualizations. Half of the participants (D1, D2, P1, P2) started with the dataset used in the previous stage to give us an opportunity to assess the expressivity of the system. We provided the other half (D3, D4, P3, P4) with a new dataset to observe the entire creation process.

Questionnaire and Interview (20 min)

Participants concluded the study by a questionnaire, addressing expressive power and usability of our prototype using a 5-point Likert scale (1–strongly disagree, 5–strongly agree). The experimenter then conducted a semi-structured interview to collect qualitative comments on the expressiveness, utility, and usability of DataInk.

Results

Overall, all participants successfully completed the tasks and created data visualizations for each dataset. We did not observe any notable differences in terms of DataInk usage or quality of the outcomes among designers and non-experts, though our sample size is small. In the following, we report subjective ratings and qualitative comments made by participants that suggest that DataInk enabled them to easily get started (i.e. had a low threshold) and create expressive data visualizations (i.e. had a high ceiling), while supporting a wide range of data-visual mappings (i.e. had wide walls) in a fluid and flexible workflow (i.e. supporting many paths, many styles).

Low Threshold - Learning Curve and Usability

All participants found the interface easy to learn (5/8 strongly agree and 3/8 agree) and easy to use (5/8 strongly agree and 3/8 agree). As D2 noted, "*your tool to Illustrator is like SketchUp to AutoCAD.*" They also found the interface had "*no high-skill cap*" (D3) and was "*fun to use*" (D2, D3, D4, P3, P4).

Notably, the usability of the Visual-Data Palette was a recurrent theme discussed in the open comments. Several participants found the drag-and-drop nature of the visual attributes to data attributes to be "*very easy and intuitive*" (D2, D3, D4, P1, P2, P4). D4 noted "*it's like tether the actual data and the visual representation*".

The locality and the design of the Visual-Data Palette enabled participants to "*quickly pick up the functionalities of the system*" (P1), as "*it doesn't distract [them] at all*" (P1). When compared to traditional graphic editing software, P2 noted: "*I can't remember all the stuff and things are usually several click away*". In contrast, participants found the palette enabled them to "*have an overview of all the available data attributes*" (P1), and that "*the slot on the inner ring is good visual indicator of the mapping*" (D4). D1 and D3 noted the "*new set of tools*" provided by the interface are "*unfamiliar*" and "*different*", but the DataInk is "*simple, straightforward, consistent, and easy to grasp*".

High Ceiling – Expressive Power

Participants responded positively about the expressiveness the interface, strongly agreeing (6/8) or agreeing (2/8) that it was easy to create the desired visuals, and strongly agreeing (5/8) or agreeing (3/8) that they were satisfied with the range of visualization designs the system is capable of. Participants highlighted the range of possibilities that DataInk affords: *"You have the micro of what you can change, which is the glyph, right? And you have the macro which is the overall layout with the shapes, you can get really crazy funky with it."*

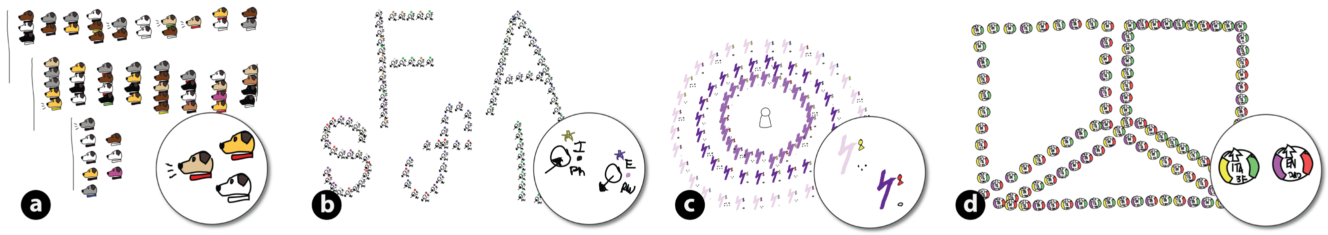

Figure 11 Examples of design realized by participants during the freeform exploration tasks.

Participants also favored the set of consistent interactions to specify the visual-data mappings, which enabled them to achieve sophisticated outcomes, e.g., "*The foundation is really good and I can see how it scales up, because you can just populate the functionalities with more and more attributes, but they all work the same way. I can totally see its potential"* (D4). Participants commented the sketching and rapid generation of data-driven graphics support their creativity, e.g., *"I feel like I could get more creative. I like that outcome which was easy to do, easy to read, info clearly communicated, and visually appealing." (P3)*

Wide Walls, Many Paths, Many Styles– Creative Exploration

For the Iteration task, all participants successfully completed the requested changes, and rated that it was easy to change glyphs (4/8 strongly agree and 4/8 agree) and layout (all strongly agree).

Positive comments on the ability to experiment with alternatives: "*I like the flexibility to try out different things"*(P3), and ratings on the statement that *it was easy to experiment with different alternatives* (5/8 strongly agree and 3/8 agree) suggests that DataInk supported their creative explorations. Figure 11 features several visualizations designed at the Freeform Exploration stage. Participants were all satisfied by their final creation (5/8 strongly agree and 3/8 agree).

During the tasks, we observed different workflows and creation strategies. All participants started by drawing basic shapes to act as a glyph representing a data point. After the glyphs populates the canvas, instead of electing to see the data attribute of a data point, D1 and P3 started with the layout, as they "*want[ed] to know how many different things are in the dataset, like how many data attributes, how many different values of them, and how many data points in each group"*. After seeing the different categories, D1 continued exploring layouts, while P3 switched to exploring how to compose the glyphs. The rest started by composing the glyphs and then the layout. All participants went back and forth between the glyph and layout editing to see whether there were data attributes left to be visualized.

We also observed participants fluidly switching between visual-data palette and legend. Participants typically experimented with new drawings mappings directly on the canvas. Then, they switched to the legend to set the mapping for every value of a data dimension, e.g., "*the dragging and the coding on the legend make it very simple and fast to do what I want to do" (D3)*. Participants commented that always being able to see the data dimensions was useful for their creative process on the canvas (e.g., "*…This helps me to think about the creation of the glyph, deciding which data attributes to pick and what would be the most appropriate visual form or layout to represent them" (P1)),* and legend *(e.g., "Looking at the attributes, I really have the urge to map them all!!" (P3)).*

Every participant explored different mappings for glyphs and layouts (Figure 11). The ability to propagate changes to other glyphs and setting the mapping on the attribute level in the legend were reported as key factors to support rapid exploration of different alternatives, e.g., *"I like the capability the tools provide for easily duplicating glyphs and mapping data attributes to visual elements. I am happy to be able to create a data visualization containing so much data in such short time."* (P1), and *"It's fast. You can change the drawing and mappings easily. It's something everyone can easily put together." (D3).*

Suggestions for Improvement

We observed several usability issues, such as quickly locating a data dimension of interest. The DataInk palette orders existing mappings in relation to the part of the glyph they apply to. However, this design induces changes in the order of items in the outer ring. D1 and P2 suggested keeping this order persistent instead, sorting them in alphabetical order for easy navigation. D4 suggested having a search function of data attributes for large datasets. Participants also suggested several features such as multiple glyph and layout panels (P3) to easily change and compare different designs, and moving these legend inside the canvas (D3) to take advantage of the entire screen space.

DISCUSSION AND FUTURE WORK

The study suggests that DataInk is promising to support creative exploration, design expression, and the rigorous execution of creating data visualizations. We reflect on each of these goals, outlining limitations, challenges, and opportunities for future work.

Limitation and Opportunities

When considering design expression, DataInk only supports glyph-based visualizations with a subset of the features available in professional illustration software. While there is a trade-off between low thresholds (i.e., learning curves) and high ceilings (i.e., expressivity), we believe that there are opportunities to support greater expression without compromising the interaction paradigm. We ground our discussion with examples from Dear Data [33].

Glyphs

DataInk supports a subset of the visual properties available for encoding. Adding support for additional ones such as rotation or opacity is directly achievable. Another interesting addition is to incorporate procedural drawing techniques such as those demonstrated in Vignette [26] and Para [22]. To reproduce the glyph made of concentric circles in Figure 12:a, DataInk requires users to draw a different shape for each data value. This process forces the users to provide a categorical visual encoding for continuous data value, while procedural drawing enables the generation of continuous ones.

Generalizing DataInk to encompass every data visualization is out of scope. We reflect on a few straightforward extensions, and more challenging ones. A simple modification to our prototype would enable users to create visual marks representing aggregates used in standard representations such as bar or bubble charts. DataInk could support this by enabling users to bind properties of visuals to the results of operations on groups of data points. For example, one could create groups of dogs by fur length (via the same mechanism provided to group glyphs for the layout). Then, by providing a set of operations on groups (e.g. count), one could map the size of a drawing to the results of this operation. DataInk would thus generate one drawing per group, rather than one per data point.

Other types of visualizations would require more profound changes. For example, node-link diagrams (Figure 12:b) rely on two types of glyphs: nodes and links. This may result in a more complex creation process, as it introduces dependencies between nodes and links glyphs when generating the visualization.

Layout

One of the more challenging features to tackle pertains to the specification of the layout of visual marks in space. Our current prototype enables the users to specify a single level of grouping of glyphs as well as their distribution path and sorting order. Supporting the hierarchical and nested spatial structure that can be seen in several examples of Dear Data (Figure 12:cd) would require a user to specify multiple

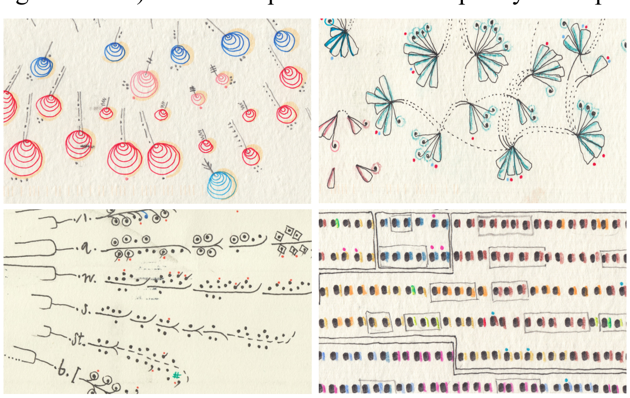

Figure 12 Examples from Dear Data requiring more features: a) shape generation for continuous data; b) node-link diagrams; c) hierarchical or d) nested layouts.

levels of groupings, distribution paths, and orders. In our prototype, this could be achieved by specifying multiple layout cards, however, the order of layout cards introduces dependencies and constraints regarding the possibilities of the nested layout structure. Conveying these constraints to the users in a transparent way, and enabling them to fluidly experiment with layout hierarchies, raises new challenges.

Blurring the Line with Data Exploration

When considering creative exploration, Design Decision 1, 4, and 5 suggested interactions to explore the different data dimensions and fluidly experiment with visual-data mappings. Reflecting on our study observations, we believe that we can enrich DataInk to provide more data exploration capabilities. We plan to enrich the interactions with the legend to enable users to browse and manipulate data dimensions (e.g. filtering, grouping, and sorting). The goal of our current prototype is to probe and evaluate a new interaction concept rather than engineering a full-featured system. Assured by the research community, in the future, we seek to support the entire workflow from collecting and editing data to creating and sharing visualizations.

Bringing Creativity to the Next Level

DataInk goes beyond the generation of static visualizations. We envision many opportunities to support creative and expressive designs for interactive and dynamic data visualizations. The philosophy and general directness of our approach combines well with techniques such as Draco [25] or Kitty [24], enabling users to specify the dynamic and interactive behavior of graphics through sketching. A direct opportunity for future research would be to explore how interactions afforded by these systems can be integrated into DataInk and augmented to support data binding.

CONCLUSION

We propose DataInk, a system for authoring whimsical and personalized visual representations of data. DataInk aims at supporting the creative process while affording design expression to craft visually-rich graphics via freeform sketching and rigorous execution by maintaining a tight coupling between visuals and data via direct manipulation. The design decisions of supporting direct manipulation are embodied as a simple set of pen-and-touch interaction techniques and a fluid graphical user interface, which enable both rich expressiveness and effortless execution of data visualizations. A user study with eight designers and non-experts suggests that this approach is promising and demonstrate that DataInk allows users to unleash their creativity to experiment with differently visual designs and create a diverse set of glyph-based visualizations.

ACKNOWLEDGEMENTS

We would like to thank members of DGP lab for their input and the reviewers for the valuable feedback. Special thanks to Peter Hamilton for his assistance with the video, Michael Wang for assisting the implementation, as well as Giorgia Lupi and Stefanie Posavec for permission to use images from Dear Data.

REFERENCES

1. Adobe Illustrator 2017, Retrieved September 16, 2017 from www.adobe.com/Illustrator
2. Michel Beaudouin-Lafon. 2000. Instrumental interaction: an interaction model for designing post-WIMP user interfaces. In Proceedings of the SIGCHI conference on Human Factors in Computing Systems (CHI '00). 446-453. http://dx.doi.org/10.1145/332040.332473
3. Jacques Bertin. 2010. *Semiology of Graphics: Diagrams, Networks, Maps (1st ed.)*. ESRI Press, Redlands, Calif.
4. Alex Bigelow, Steven Drucker, Danyel Fisher, and Miriah Meyer. 2014. Reflections on How Designers Design with Data. In *Proceedings of the International Working Conference on Advanced Visual Interfaces* (AVI '14), 17–24. https://doi.org/10.1145/2598153.2598175
5. Alex Bigelow, Steven Drucker, Danyel Fisher, and Miriah Meyer. 2017. Iterating between Tools to Create and Edit Visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 23(1), 481-490. https://doi.org/10.1109/TVCG.2016.2598609
6. Bill Buxton. 2010. *Sketching user experiences: getting the design right and the right design*. Morgan Kaufmann.
7. Rita Borgo, Johannes Kehrer, David H.S. Chung, Eamonn Maguire, Robert S. Laramee, Helwig Hauser, Matthew Ward, Min Chen. 2013. Glyph-based Visualization: Foundations, Design Guidelines, Techniques and Applications. *Eurographics State of the Art Reports*, 39-63.
8. Mike Bostock, V. Ogievetsky, and J. Heer. D^3 Data-Driven Documents. 2011. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309. https://doi.org/10.1109/TVCG.2011.185
9. Jeffrey Browne, Bongshin Lee, Sheelagh Carpendale, Nathalie Riche, and Timothy Sherwood. 2011. Data analysis on interactive whiteboards through sketch-based interaction. In *Proceedings of the International Conference on Interactive Tabletops and Surfaces* (ITS '11), 154-157. https://doi.org/10.1145/2076354.2076383
10. Alberto Cairo. 2012. *The Functional Art: An introduction to information graphics and visualization*. New Riders.
11. Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman, eds. 1999. *Readings in information visualization: using vision to think*. Morgan Kaufmann.
12. Marianela Ciolfi Felice, Nolwenn Maudet, Wendy E. Mackay, and Michel Beaudouin-Lafon. 2016. Beyond Snapping: Persistent, Tweakable Alignment and Distribution with StickyLines. In *Proceedings of the ACM Symposium on User Interface Software and Technology* (UIST '16), 133-144. https://doi.org/10.1145/2984511.2984577
13. William S. Cleveland and Robert McGill. 1987. Graphical Perception: The Visual Decoding of Quantitative Information on Graphical Displays of Data. *Journal of the Royal Statistical Society. Series A (General)* 150(3):192–229. https://doi.org/10.2307/2981473
14. Darius Coelho, Ayush Kumar, and Klaus Mueller. 2015. Data Memes for Personal Visualization. *Proceeding of IEEE VIS 2015 workshop Personal Visualization: Exploring Data in Everyday Life.*
15. Andrew Crotty, Alex Galakatos, Emanuel Zgraggen, Carsten Binnig, and Tim Kraska. 2015. Vizdom: interactive analytics through pen and touch. In *Proceeding of VLDB Endow*, 8(12):2024-2027. http://dx.doi.org/10.14778/2824032.2824127
16. DensityDesign Research Lab. RAWGraphs. Retrieved September 16, 2017 from http://rawgraphs.io/
17. Drupal. EasyCharts. Retrieved September 16, 2017 from http://www.easychart.org/
18. Lars Grammel, Chris Bennett, Melanie Tory, and Margaret-Anne Storey. 2013. A Survey of Visualization Construction User Interfaces. In *EuroVis - Short Papers*. http://dx.doi.org/10.2312/PE.EuroVisShort.EuroVisShort2013.019-023
19. Lars Grammel, Melanie Tory, and Margaret-Anne Storey. 2010. How Information Visualization Novices Construct Visualizations. *IEEE Transactions on Visualization and Computer Graphics* 16(6): 943–952. https://doi.org/10.1109/TVCG.2010.164
20. Ken Hinckley, Koji Yatani, Michel Pahud, Nicole Coddingotn, Jenny Rodenhouse, Andy Wilson, Hrvoje Benko, and Bill Buxton. 2010. Pen + touch = new tools. In *Proceedings of the ACM Symposium on User interface software and technology* (UIST '10), 27–36. https://doi.org/10.1145/1866029.1866036
21. Samuel Huron, Sheelagh Carpendale, Alice Thudt, Anthony Tang, and Michael Mauerer. 2014. Constructive Visualization. In *Proceedings of the Conference on Designing Interactive Systems* (DIS '14), 433–442. https://doi.org/10.1145/2598510.2598566
22. Jennifer Jacobs, Sumit Gogia, Radomir Mech, and Joel Brandt. 2017. Supporting Expressive Procedural Art Creation through Direct Manipulation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '17), 6330-6341. https://doi.org/10.1145/3025453.3025927
23. Jaemin Jo, Sehi L'Yi, Bongshin Lee, and Jinwook Seo. 2017. TouchPivot: Blending WIMP & Post-WIMP

Interfaces for Data Exploration on Tablet Devices. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (CHI '17), 2660-2671. https://doi.org/10.1145/3025453.3025752

24. Rubaiat Habib Kazi, Fanny Chevalier, Tovi Grossman, and George Fitzmaurice. 2014. Kitty: Sketching Dynamic and Interactive Illustrations. *Proceedings of the ACM Symposium on User interface software and technology* (UIST '14), 395–405. https://doi.org/10.1145/2642918.2647375
25. Rubaiat Habib Kazi, Fanny Chevalier, Tovi Grossman, Shengdong Zhao, and George Fitzmaurice. 2014. Draco: bringing life to illustrations with kinetic textures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '14), 351-360. https://doi.org/10.1145/2556288.2556987
26. Rubaiat Habib Kazi, Takeo Igarashi, Shengdong Zhao, and Richard Davis. 2012. Vignette: interactive texture design and manipulation with freeform gestures for pen-and-ink illustration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, (CHI '12), 1727-1736. https://doi.org/10.1145/2207676.2208302
27. Nam-Wook Kim, Eston Schweickart, Zhicheng Liu, Mira Dontcheva, Wilmot Li, Jovan Popovic, Hanspeter Pfister. 2017. Data-Driven Guides: Supporting Expressive Design for Information Graphics, *IEEE Transactions on Visualization & Computer Graphics*, 23(1):491-500. https://doi.org/10.1109/TVCG.2016.2598620
28. https://doi.org/10.1145/2207676.2208302
29. Randy Krum. 2013. *Cool infographics: Effective Communication with Data Visualization and Design*. Wiley.
30. Bongshin Lee, Petra Isenberg, Nathalie Henry Riche, and Sheelagh Carpendale. 2012. Beyond Mouse and Keyboard: Expanding Design Considerations for Information Visualization Interactions. IEEE Transactions on Visualization and Computer Graphics 18(12): 2689-2698. https://doi.org/10.1109/TVCG.2012.204
31. Bongshin Lee, Rubaiat Habib Kazi, and Greg Smith. 2013. SketchStory: Telling More Engaging Stories with Data through Freeform Sketching. *IEEE Transactions on Visualization and Computer Graphics* 19(12):2416-2425. https://doi.org/10.1109/TVCG.2013.191
32. Bongshin Lee, Greg Smith, Nathalie Henry Riche, Amy Karlson, and Sheelagh Carpendale. 2015. Sketchinsight: Natural data exploration on interactive whiteboards leveraging pen and touch interaction. In *Proceeding of the IEEE Pacific Visualization Symposium* (PacificVis '15), 199–206. https://doi.org/10.1109/PACIFICVIS.2015.7156378
33. Giorgia Lupi, and Stephanie Posavec. 2016. *Dear Data*. Princeton Architectural Press.
34. Eamon Maguire, Philippe Rocca-Serra, Susanna-Assunta Sansone, Jim Davies, and Min Chen. 2012. Taxonomy-Based Glyph Design with a Case Study on Visualizing Workflows of Biological Experiments. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2603-2612. https://doi.org/10.1109/TVCG.2012.271
35. Gonzalo G. Méndez, Miguel A. Nacenta, and Sebastien Vandenheste. 2016. iVoLVER: Interactive Visual Language for Visualization Extraction and Reconstruction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '16), 4073–4085. https://doi.org/10.1145/2858036.2858435
36. Gonzalo G. Méndez, Uta Hinrichs, and Miguel A. Nacenta. 2017. Bottom-up vs. Top-down: Trade-offs in Efficiency, Understanding, Freedom and Creativity with InfoVis Tools. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '17), 841-852. https://doi.org/10.1145/3025453.3025942
37. Microsoft Excel 2016, Retrieved September 16, 2017 from https://products.office.com/en-us/excel
38. Microsoft Power BI, Retrieved September 16, 2017 from https://powerbi.microsoft.com/en-us/
39. Otto Neurath. 2011. *From Hieroglyphics to Isotype: A Visual Autobiography*, eds. Matthew Eve and Christopher Burke, Hyphen Press.
40. Donghao Ren, Tobias Höllerer, and Xiaoru Yuan. 2014. iVisDesigner: Expressive Interactive Design of Information Visualizations. IEEE Transactions on Visualization and Computer Graphics, 20(12):2092–2101. https://doi.org/10.1109/TVCG.2014.2346291
41. William Ribarsky, Eric Ayers, John Eble, and Sougata Mukherjea. 1994. Glyphmaker: creating customized visualizations of complex data. *Computer*, 27(7):57–64. https://doi.org/10.1109/2.299412
42. Yann Riche, Nathalie Henry Riche, Ken Hinckley, Sheri Panabaker, Sarah Fuelling, and Sarah Williams. 2017. As We May Ink? Learning from Everyday Analog Pen Use to Improve Digital Ink Experiences. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '17), 3241-3253. https://doi.org/10.1145/3025453.3025716
43. Lisa Rost. Blog Post. Retrieved 12 September 2017 from https://source.opennews.org/articles/what-i-learned-recreating-one-chart-using-24-tools/
44. Ramik Sadana and John Stasko. 2014. Designing and implementing an interactive scatterplot visualization for a tablet computer. In Proceedings of the 2014 International Working Conference on Advanced Visual

Interfaces (AVI '14). ACM, New York, NY, USA, 265-272. https://doi.org/10.1145/2598153.2598163

45. Arvind Satyanarayan, Ryan Russell, Jane Hoffswell, Jeffrey Heer. 2016. Reactive Vega: A streaming dataflow architecture for declarative interactive visualization. *IEEE Transactions of Visualization and Computer Graphics,* 22(1): 659–668. https://doi.org/10.1109/TVCG.2015.2467091
46. Arvind Satyanarayan and Jeffrey Heer. 2014. Lyra: An interactive visualization design environment. *Computer Graphics Forum*, 33(3):351–360. https://doi.org/10.1111/cgf.12391
47. Carlos Scheidegger, Huy Vo, David Koop, Juliana Freire, and Claudio Silva. 2007. Querying and creating visualizations by analogy. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1560–1567. https://doi.org/10.1109/TVCG.2007.70584
48. Ben Shneiderman et al. 2006. Creativity support tools: Report from a U.S. National Science Foundation sponsored workshop. *International Journal of Human-Computer Interaction*, 20(2):61-77. http://dx.doi.org/10.1207/s15327590ijhc2002_1.
49. Julie Steele, and Noah Iliinsky. 2010. *Beautiful visualization: looking at data through the eyes of experts*. O'Reilly Media, Inc.
50. Tableau Software, Retrieved September 16, 2017 from https://www.tableau.com/products/desktop
51. Maryam Tohidi, William Buxton, Ronald Baecker, and Abigail Sellen. 2006. Getting the Right Design and the Design Right: Testing Many Is Better Than One. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI ’06), 1243-1252. https://doi.org/10.1145/1124772.1124960
52. Andries van Dam. 1997. Post-WIMP user interfaces. Commun. ACM 40, 2 (February 1997), 63-67. http://dx.doi.org/10.1145/253671.253708
53. Andrew Vande Moere, and Helen Purchase. 2011. On the role of design in information visualization. *Information Visualization,* 10(4): 356-371. https://doi.org/10.1177/1473871611415996
54. Fernanda B. Viegas, Martin Wattenberg, Frank van Ham, Jesse Kriss, and Matt McKeon. 2007. ManyEyes: a Site for Visualization at Internet Scale. *IEEE Transactions on Visualization and Computer Graphics* 13(6):1121–1128. https://doi.org/10.1109/TVCG.2007.70577
55. Jagoda Walny, Sheelagh Carpendale, Nathalie Henry Riche, Gina Venolia, Philip Fawcett. 2011. Visual Thinking In Action: Visualizations As Used On Whiteboards. *IEEE Transactions on Visualization & Computer Graphics*, 17(12):2508-2517. https://doi.org/10.1109/TVCG.2011.251
56. Jagoda Walny, Samuel Huron, and Sheelagh Carpendale. 2015. An Exploratory Study of Data Sketching for Visual Representation. *Computer Graphics Forum*, 34(3). https://doi.org/10.1111/cgf.12635
57. Jagoda Walny, Bongshin Lee, Paul Johns, Nathalie Henry Riche, and Sheelagh Carpendale. 2012. Understanding Pen and Touch Interaction for Data Exploration on Interactive Whiteboards. *IEEE Transactions on Visualization and Computer Graphics,* 18(12):2779-2788. https://doi.org/10.1109/TVCG.2012.275
58. Hadlet Wickham. 2010. *ggplot2: Elegant Graphics for Data Analysis (Use R!). Second edition*. Springer.
59. Kanit Wongsuphasawat, Dominik Moritz, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. 2016. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):649–658. https://doi.org/10.1109/TVCG.2015.2467191
60. Haijun Xia, Bruno Araujo, Tovi Grossman, and Daniel Wigdor. 2016. Object-Oriented Drawing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI ’16), 4610-4621. https://doi.org/10.1145/2858036.2858075
61. Haijun Xia, Bruno Araujo, and Daniel Wigdor. 2017. Collection Objects: Enabling Fluid Formation and Manipulation of Aggregate Selections. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)., USA, 5592-5604. https://doi.org/10.1145/3025453.3025554
62. Emanuel Zgraggen, Robert Zeleznik, and Steven M. Drucker. 2014. PanoramicData: data analysis through pen & touch. IEEE Transactions on Visualization and Computer Graphics (TVCG) 20, 12: 2112-2121. https://doi.org/10.1109/TVCG.2014.2346293