From Exploration to Explanation: Designing for Visual Data Storytelling

A dissertation presented
by
Nam Wook Kim
to
The School of Engineering & Applied Sciences

in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy
in the subject of
Computer Science

Harvard University
Cambridge, Massachusetts
May 2019

Thesis advisor: Professor Hanspeter Pfister Nam Wook Kim

From Exploration to Explanation: Designing for Visual Data Storytelling

Abstract

Past research in visualization has focused on supporting rapid analysis and exploration of data, involving comparing different visual representations for perceptual effectiveness and building usable analytic systems for domain experts. As data is becoming much more ubiquitous and relevant to our daily lives in recent years, there has been a growing need for communicating data to a general audience.

However, many visualization tools still lack support for communication that ranges from adding simple annotations to customizing visualizations and constructing stories. These explanatory components can significantly aid understanding and recall of information compared to traditional exploratory visualizations. Due to the absence of such support in existing visualization tools, people still rely on graphic design tools that do not have data-driven abstractions, requiring time-consuming and error-prone manual encoding procedures.

This thesis investigates how to enable individuals to design expressive visual data stories through a series of interactive systems. The first, Data-Driven Guides, provides data abstraction into a flexible design environment in which users can generate guides from data and draw custom shapes with the help of the guides. The second system, DataSelfie brings the expressivity of DDG into a personal context by allowing users to create a visual vocabulary to represent their personal data. Moving beyond the generation of a single visualization, DataToon offers fluid comic storyboarding by blending exploration and explanation in a unified environment supported by pen and touch interactions. Lastly, story curves allows storytellers to quickly grasp the narrative structure of a story by visualizing how events are temporally arranged.

The fundamental idea of this thesis is to go beyond traditional exploratory visualizations to create expressive, explanatory, and personal visual stories. The resulting systems establish a research framework where presentation and storytelling is a core part of visual data systems.

Contents

Listing of figures

Dedicated to my family.

Acknowledgments

This thesis would not have been possible without support from many indispensable individuals:

- my advisor Hanspeter Pfister who gave me a leg up in not only growing as an independent researcher but also pursing whatever I am interested in. Particularly due to my idiosyncratic ideas, there were a lot of chances that projects in this thesis could go sideways. I am grateful for his insight and vision that helped me stay on track.
- my co-advisor Krzysztof Gajos for not only supporting my research but also demonstrating many more important things other than research, including how to implement an inclusive and diverse teaching environment.
- my mentor Nathalie Henrie Riche for inspiring me with her passion, energy, and intellectual curiosity, along with a burst of insightful research ideas in every meeting we had.
- my mentor Leo Zicheng Liu for believing in my initially crazy idea about DDG and staying with me until publishing my first paper in IEEE VIS.
- my internship hosts Sasha Schriber and Markus Gross for providing me the most memorable experience in a wonderful country, Switzerland.
- Jeffrey Heer and Stuart Card who got me into this business in the first place.
- Many colleagues and collaborators who made this thesis together, including Hyejin Im, Benjamin Bach, Matthew Brehmer, Zoya Bylinskii, Michelle Borkin, Aude Oliva, Fredo Durand, Eston Schweickart, Mira Dontcheva, Wilmot Li, Jovan Popovic, Ken Hinckley, Michel Pahud, Haijun Xia, Michael J. McGuffin.
- high-school advisees, Alicia Wang and Guanpeng Xu, who gave me the joy of mentoring.
- study participants for my research, including workers from Amazon's Mechanical Turk and students from Harvard.
- friends from Harvard and worldwide, especially members of the Visual Computing group and Intelligent Interactive Systems group.
- my parents Seong Ho Kim and Seongja Jeong for their unconditional love and support.
- my brother Nam Kyun Kim, my nephew Doyun Kim, and sister-in-law Joomi Kim for being there for me.
- my wife Hyejin Im for putting in the hard work on multiple research projects that are part of my thesis, but more importantly, always caring about me and making me a better person.

My graduate studies were generously supported by Kwanjeong Educational Foundation and Siebel Scholars Foundation — thanks.

Author List

The following authors contributed to this thesis:

- Chapter 3: Eston Schweickart, Zhicheng Liu, Mira Dontcheva, Wilmot Li, Jovan Popovic, and Hanspeter Pfister.
- Chapter 4: Hyejin Im, Nathalie Henry Riche, Alicia Wang, Krzysztof Z. Gajos, and Hanspeter Pfister.
- Chapter 5: Nathalie Henry Riche, Benjamin Bach, Guanpeng Xu, Matthew Brehmer, Ken Hinckley, Michel Pahud, Haijun Xia, Michael J. McGuffin, and Hanspeter Pfister.
- Chapter 6: Benjamin Bach, Hyejin Im, Sasha Schriber, Markus Gross, and Hanspeter Pfister.

Technology should not aim to replace humans, rather amplify human capabilities.

Douglas Engelbart

1 Introduction

Today, data is everywhere. The quantity and quality of data produced in our society, from science, engineering, humanities, business, as well as everyday human activities, has been increasing at staggering rates. This data contains immense information about our lives and allows us to better understand ourselves and our communities. For instance, a large collection of health records can enable clinicians to discover recurring treatment patterns from patients and make informed decisions to improve their health care, while various sensors embedded in personal devices have also empowered individuals to

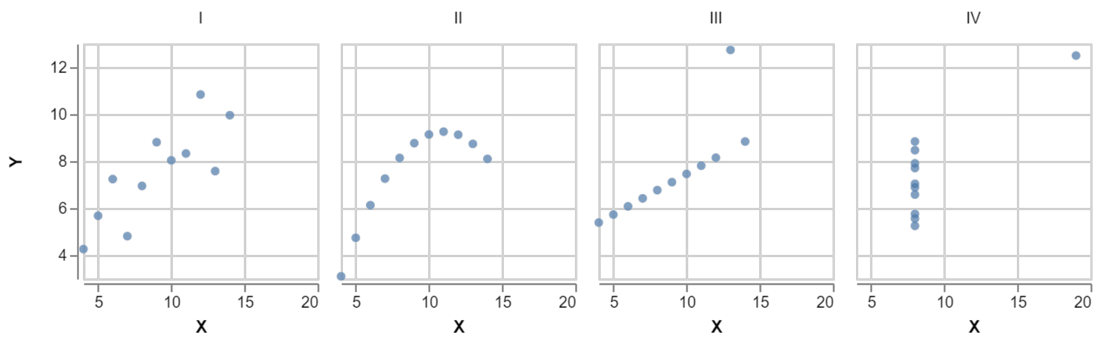


Figure 1.1: A trellis plot of Anscombe's Quartet[17]: four different datasets with same descriptive statistics reveal varying qualitative patterns; the second and fourth graphs reveal nonlinear relationships between the two variables, while the third and fourth graphs show the impact of outliers.

collect and analyze data about daily lives to increase self-knowledge. As data is becoming more ubiquitous and the world is becoming data-driven, the ability to understand and communicate data is becoming essential for everyone.

By leveraging our visual perception, visualization provides a powerful means for showing meaningful patterns in data. A classic example used to demonstrate the value of visualization is Anscombe's Quartet designed by F.J. Anscombe in 1973[17]. The quartet consists of four different datasets with identical descriptive statistics (e.g., means, standard deviations, and correlation coefficients), which may lead one to believe that the datasets are reasonably similar. However, once visualized, they reveal vastly different patterns (Figure 1.1). What this example essentially tells us is that we need to make both calculations and graphs, each of which contribute to our understanding of the data[17].

Most past research in visualization has been focused on finding effective visual representations by quantifying the perception of different visual encoding channels and supporting rapid analysis and exploration of data to discover new insights from the data. Likewise, many visualization systems are designed to support the analysis process of domain experts (e.g., biology researchers and financial analysts), ranging from exploring patterns and detecting outliers in data to formulating and confirming hypotheses about the data. However, these systems typically have complex designs that are unintuitive

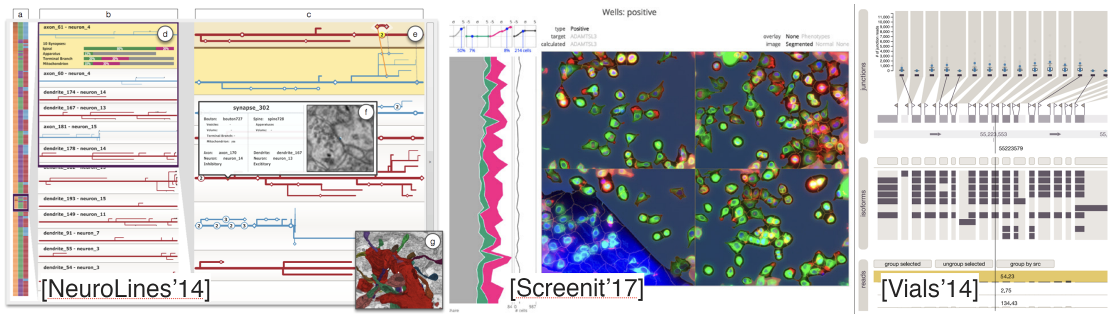


Figure 1.2: Examples of visual analytic systems for domain scientists: NeuroLines [13] for neuroscientists, and Screenit [59] and Vials [182] for genetic biologists.

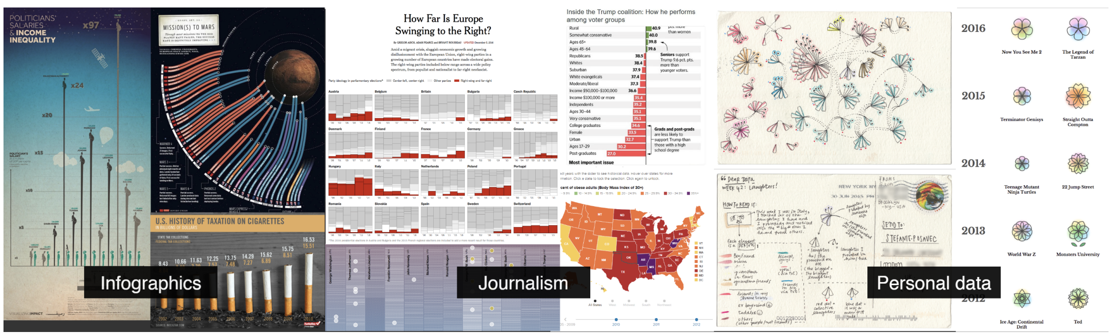


Figure 1.3: With the improved accessibility of data and the increasing demand for communicating the data, people from diverse backgrounds, including designers, journalists, and casual users, are generating expressive, explanatory, and personal visual data stories.

and cumbersome for non-experts as they are tailored to tasks and data to a specific domain.

On the other hand, Visualization is more and more being used to communicate messages derived from data. In particular, due to the increased accessibility of data in the public sector, visualization is reaching a broad audience including journalists, designers, and even casual users. For example, designers are creating compelling visualizations that are different from standard charts but more tailored to data types and message contexts, while journalists are frequently employing visualizations to tell compelling stories about data and to make the public informed. This growing need for communicating complex data led to the emergence of *data-driven storytelling* that goes beyond exploratory visualizations towards explanatory visualizations.

However, existing visualization tools provide poor support for communication. Creating explanatory visualizations need to consider more than perceptual effectiveness but also other design factors such as aesthetics, engagement, and memorability. People still often rely on graphic design tools that do not provide data-driven abstractions. As a result, when designing custom visualizations, it requires tedious and manual encoding process to draw graphics based on data. Things can be much more challenging when organizing multiple visualizations to construct a holistic data story. It involves developing a compelling narrative flow while reducing the complexity of the data and keeping multiple views consistent. This data storytelling capability is still vastly underexplored in visualization tools.

This thesis investigates how to enable individuals to design visualizations for communication and author engaging stories about data by developing a series of interactive prototypes. The thesis contributes novel methods for creating expressive data graphics that can communicate the semantics of data, as well as systems for creating data stories using elements of comics and analyzing the narrative structure of stories. The results of these methods and systems establish a research framework for developing visual data storytelling systems by advancing the spectrum of visualization design and transitioning from exploration to explanation.

Thesis statement: Visual data systems that enable people to design custom visualizations, author compelling data stories, and analyze the underlying structure of stories can support creative, expressive, and iterative visual data storytelling going beyond traditional exploratory visualizations toward explanatory visualizations.

1.1 Summary of Contributions and Thesis Outline

This thesis is divided into seven chapters and contributes four interactive artifacts that provide new knowledge by enabling novel ways for visual data storytelling.

Chapter 2 surveys prior work that this thesis builds on, including an extensive review of visualization tools and systems as well as 2D shape deformation techniques, personal informatics and visualizations, multimodal interaction techniques for content authoring, and visualizations for stories.

Chapter 3 introduces Data-Driven Guides (DDG), a technique for designing expressive data graphics in a graphic design environment. Instead of being confined by predefined templates, DDG allows designers to generate guides from data and use the guides to place and measure custom shapes accurately. Following the principles of information encoding, the guides can encode three main visual channels, length, area, and position, that can be combined to create a variety of visual structures and map these structures to data (Figure 1). When a user updates the data, DDG uses a deformation technique to transform custom shapes using the guides as the backbone of the shapes. DDG brings data abstraction into a designer's familiar authoring environment and addresses tedious manual encoding and lack of data-driven support in the existing design practice using freeform illustration tools.

Chapter 4 studies how to enable individuals to determine the representation of their own data. This work is motivated by the fact that existing personal informatics systems mostly focus on data collection, specifically automated tracking, use predefined presentations of the data. Users remain mostly passive and less engaged with the data as such. This chapter introduces DataSelfie that brings the expressivity of DDG into a personal context by augmenting a familiar survey editor with a drawing capability. Users can create any questionnaire to gather data about themselves and design a personalized visual vocabulary to represent the qualitative and nuanced aspects of the data. DataSelfie introduces a novel generative framework for producing a visualization, compared to descriptive approaches in existing tools including DDG. As a result, the construction of visualizations is not an afterthought, but a primary activity that users actively engage in while thinking about the goals of their data collection.

Chapter 5 presents DataToon, a data storytelling tool that blends exploration and explanation in a unified environment supported by pen and touch interactions. Most visualization tools including DDG and DataSelfie allow for the production of a single visualization which may be sufficient for conveying simple messages about the data. But, they cannot support the design of a fuller narrative and thus their ability to produce a comprehensive story is limited. Leveraging the well-established visual language of comics, DataToon allows a storyteller to rapidly generate visualization panels, annotate them, and arrange them on a canvas to produce a visually compelling narrative. To facilitate rapid data exploration, it offers recommendations of interesting patterns in the data. To allow fluid experimentation of alternative narrative structures, it enables direct manipulation of panels and their data contents with pen and touch interactions, as well as automatic transitions between panels.

Chapter 6 describes story curves, a visualization technique that shows the narrative structure of a story. Crafting a story is a creative, open-ended process that involves trial and error. Being able to quickly grasp the overall narrative structure can accelerate this iterative process. story curves focuses on revealing nonlinear narrative patterns by showing the order in which events are told and comparing them to their actual chronological order. The analysis of movie scripts using Story Curves unveiled new narrative patterns, e.g., merging or diverging zigzags, that have not been discovered in the literature. Story Explorer builds on story curves and provides a full-fledged system to explore movie scripts, as well as a curation interface to allow users to specify the chronological order of events in the scripts.

Lastly, Chapter 7 reflects on future research directions and opportunities that build on the ideas behind this thesis.

Solving a problem simply means representing it so as to make the solution transparent.

Herbert Simon

2

Related Work

This thesis builds on prior work in visualization, human-computer interaction, and computer graphics. This chapter provides a review of specific areas in these domains that are related to the topics discussed in this thesis.

2.1 Visualization for Communication

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[190]. Most visualization systems today are based on these principles that inform perceptually effective visual encodings of data[51,80,185]. It is only recently that researchers have started exploring other aspects of visualization design such as memorability[36,35], aesthetics[145], and engagement[79,38]. 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[27,78,92,33].

Although embellishments can have negative impacts on visual search time[33] or certain analytic tasks[178], 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[114,179] that are beginning to find applications in visual storytelling[120,115]. As noted by Bigelow et al.[29], 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. One of the goals of this thesis is to provide appropriate tools to alleviate error-prone manual operations required in the design of engaging and memorable custom infographics.

2.1.1 Visualization Design Tools

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[74]. 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[192], RAW [*], Plotly [†]). 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[181,171] are based on formal graphical specifications such as the grammar of graphics[201] or a declarative model[81,44]. Tableau [‡] follows a similar formulation, and has been designed to support rapid exploratory data analysis rather than custom visualization design. On the other hand, Lyra[171], built on the Vega grammar [§], provides a more accessible interface to customize visual encodings. Spritzer et al.[179] 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[98,168,120] or demonstrational interfaces[149]. Other tools allow users to interactively construct multiple, coordinated visualizations[160,Weaver]. More recently, Data Illustrator[129], DataInk[207], and Charticulator[161] allow further expressivity in terms of custom visual marks and custom layouts, while tools like ChartAccent[159] or DataWrapper[72] provide rich annotation options. 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[37,199],

[*] http://raw.densitydesign.org
[†] https://plot.ly/
[‡] https://public.tableau.com
[§] https://vega.github.io/vega/

and others provide composable operators for different visual encodings[82] or extensible visualization widgets[66]. 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[69,108]. However, they separate the construction interface from the canvas area (i.e., poor 'closeness of mapping' between the problem domain and the tool[76]). 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)¶ or require procedural thinking to define loops to generate visual marks parametrically‖.

Despite the wealth of tools available for creating visualizations, Bigelow et al.[29] 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[194,193] and manipulation of tangible materials[97,103,183,96,203].

In this thesis, DDG takes advantage of the flexibility and expressivity of manual encoding while alleviating its time-consuming and error-prone aspects. Instead of enforces a rigid order of operations, DDG provide guides generated from data that help designers draw their own visual marks and layouts in a flexible graphic design environment.

¶http://aprt.us

‖https://vimeo.com/66085662

2.1.2 Constraint Deformation for 2D 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 [101]. 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. [100] 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. [127] extended this work to allow for deforming vector graphics (e.g., Bézier splines).

DDG augments 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. [90]. 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.

2.2 Data-Driven Storytelling

Nowadays, journalists (e.g., from the New York Times and the Guardian), as well as designers, analysts, and casual users, are increasingly using visualizations to convey messages about data. A growing interest in how to communicate data more effectively led to the emergence of *data-driven storytelling*. Previous studies investigated the roles of visualizations in data stories [71,173,93], as well as different genres including animation [166], data videos [15,49], annotation [159], and data comics [19,20,22]. However, many visualization tools still lack support for crafting data stories, such as collecting and organizing story pieces from data exploration [122].

2.2.1 Tools for Authoring Data Stories

Most existing visualization tools allow for the production of one visualization at a time, making it difficult to design a comprehensive narrative.

Recent research has examined the integration of communicative visualization within a linear narrative sequence[94]. This research is reflected in another category of tools that focus on sequence and narration. These include commercial tools including Tableau's Story Points[184] and Bookmarks for Microsoft's Power BI[143], which provide interfaces for composing a sequence of story points with embedded visualizations. Meanwhile, tools emerging from the research community aim for greater expressivity. These include: Ellipsis[170] and Timeline Storyteller[40], which augment a sequence of visualizations with annotations and state-based scene transitions; DataClips[16], which focuses on sequencing data-driven video clips; and Vistories[75], which leverages the interaction history produced during data exploration to automatically generate a sequence that can be curated and annotated into a presentable story. In each of these tools, a set of annotated visualizations are arranged in a linear narrative sequence, revealed one at a time via stepping or scrolling interactions[140].

Unlike linear slideshows and scroll-based stories, the layout and juxtaposition of panels in a comic allows for non-linear narrative structures, in which a reader can consume narrative points in various orders in a glanceable format that affords both skimming and revisitation. Unfortunately, no single existing data-driven tool can produce such narrative structures. The sole existing data comics editor by Zhao and Elmqvist[211] allows for the composition of linear slideshow comics and the embellishment of visualization with speech bubbles and a narrator character. However, like illustration software, this tool requires the importation of preexisting visualization generated by other tools. In contrast, DataToon provides the first all-in-one visualization and narrative design tool where multiple panels can be arranged freely on a page.

Another assumption inherent to many existing data-driven storytelling tools is that the storyteller already has a preconceived story, perhaps developed in the course of data analysis performed with other tools, in consultation with data analysts and subject matter experts, or in some combination thereof. However, this separation of analysis and storytelling hampers rapid experimentation of alternative narrative structures and the process of refining a story. In other words, dedicated analysis tools often do not have flexible storytelling features while dedicated storytelling tools lack data exploration capabilities such as ways to collect and organize insights.

One of the benefits of an interface that allows for the flexible arrangement of comic panels is that storytellers can rapidly iterate with alternative narrative structures. Furthermore, they can quickly generate and discard panels the process of data exploration without disrupting completed parts of the comic. Finally, DataToon integrates automatic suggestions and transitions between panels as a way to scaffold a story, eliminating the tedium of alternating between a dedicated visualization tool and a dedicated storytelling tool.

2.2.2 Visualizations of Stories

Information visualization has been a valuable tool for humanities scholars, enhancing their interpretative activities for literary works[Jänicke et al.,139]. Consequently, there are numerous works for visualizing narrative contents with a majority of them focusing on visualizing story content rather than narrative.

Visualization has been used to show character interactions (e.g., multiple protagonists or relationship developments, etc). Individual characters are represented as timelines that converge and diverge to indicate interactions or co-occurrences at the scene or event level[186,128,187,89,157]; similar visual encoding methods are found in other domains as well, such as the temporal dynamics of family relationships[111] and the evolution of network structures over time[141,202,210]. A potential drawback of the line representation is that a character's absence is not encoded as the line persists from the start to the end of the narrative. Other visualizations have focused on the interplay between characters, employing matrix representations[les] or node-link diagrams[106]. Often, graph metrics for clustering and centrality are presented together.

Another line of examples focuses on visualizing metadata extracted from the movie script. Metadata includes character dialogs, sentiments (positive or negative)[ort,32,sta] and emotions (primary emotions such as joy, anger, sadness, fear, disgust)[56]. Existing script writing software also embeds visualizations to show narrative structures as well as semantic metadata such as character motivations and dramatic events that are manually annotated by screenplay writers themselves[sto] by, e.g., showing the evolution of a character's emotions through a line chart. Some visualizations directly show character dialogs on top of visual summaries of characters and scenes, enabling qualitative interpretations for readers[200,177]. In addition, a few works visualize metadata that is not part of the narrative, such as character co-mention networks in Twitter[gam] or movie reviews[150].

Yet, there is little work for visualizing nonlinear narratives. Most existing visualizations for nonlinear narratives are non-interactive and hand-crafted infographics for specific movies[inf]. For example, Sharma and Rajamanickam[174] use a straight horizontal timeline to indicate the storyline and arcs to show nonlinear time jumps using manually crafted data for the film *500 Days of Summer*. A visualization published by Carter et al.[43] in the New York Times relates the order of scenes in movie trailers to the their temporal position in the movie. However, it does not show story order of the actual scenes, but rather compares two narrative orders: the movie narrative and the trailer narrative. Thus, no narrative patterns are observable. While extending that technique to entire movies, we additionally visualize additional data such as characters, places, and times.

2.2.3 Interaction Design for Creativity Support

Historically, comics have been drawn by hand, and thus we gravitated to interfaces that could leverage expressive pen-based input for drawing and styling comic elements. Such interfaces have become increasingly popular in recent years, and along with them we have seen an emerging body of research that focuses on the combination of pen and touch interaction for content creation and manipulation. By combining these forms of interaction, users report feeling more directly engaged as compared to manipulating elements via a WIMP interface[207]. Hinckley et al. introduced a rich palette of compelling interaction techniques for manipulating content, all following the principle that the pen writes and touch manipulates[86,153]. Other research has sought to identify and evaluate pen and touch gestures for common operations on interactive surfaces, including selection, deletion, and copy / paste[148], and these gestures have applied such gestures in various applications including diagram editing[68], digital drawing[204,205], early-stage ideation[206], and active reading [85]. Visualization researchers are also beginning to take advantage of touch and pen interaction in various contexts[119], including visualization authoring[207], storytelling[121], and data exploration[208,105], though until DataToon, they have yet to apply such interaction to the creation of data comics.

2.3 Casual and Personal Visualization

Most conventional visualization systems are designed to support domain experts to perform analytic tasks. As visualization has become widespread among a general audience, we are beginning to see more diverse uses of visualization. Pousman et al. describe casual information visualization that often involves ambient and artistic representations of data[154]. This type of visualization targets a broader population rather than experts alone and also serves personal or even aesthetic purposes[191]. Like other traditional visualizations, it generates analytic, often implicit, insights but often focuses more on awareness and reflective insights[154].

2.3.1 Personal Informatics & Self-Monitoring

Over the past decade, research has been abundant in personal informatics[126,125], also known as various similar terms such as lifelogging and quantified self. A wide range of tools has been proposed to assist with collecting and managing various kinds of personal information, such as habits, activities, and moods, to encourage self-reflection and to promote self-knowledge and behavior change. Cho et al. characterize the design space of personal informatics based on whether data collection is fully automatic, semi-automatic, or fully manual[48]. Manual approaches include analog methods using pen and paper such as bullet journals[42] or digital methods using spreadsheets or note-taking apps. The manual approaches have high data capture burdens, often hindering people from sustaining long-term practices[48].

Automated tracking technology attempts to address this issue by leveraging personal devices embedded with various sensors to remove the need for manual data inputs[48]. However, complete automation of data collection often eliminates additional opportunities for engagement and reflection with personal data[48]. Also, sensors have limitations in the types of data they can collect, mainly focusing on collecting quantitative information, and fails to support people's practical goals and emotional needs[18,48]. Most semi-automated approaches seek to strike a balance between both ends of the spectrum[48,113]. For instance, SleepTight aims to make manual tracking easier rather than automate it by leveraging lock screen and home screen widgets[50], while OmniTrack provides manual trackers combined with triggers and external services that enable automated logging[112].

Existing self-tracking tools focus on data collection and management, not necessarily on how to display the data. As a result, the design of the tools dictates the presentation of the data, showing data summaries in the form of standard charts or tables. DataSelfie explores how to allow users to decide the representation of their data by augmenting a familiar survey tool with a drawing capability.

2.3.2 Visualization in Personal Informatics

In the field of personal informatics, visualization has gained growing popularity as it provides a means to quickly make sense of complex data without requiring advanced statistical literacy. Huang et al. provide an overview of various design dimensions of personal visualizations and personal visual analytics[91], including who the data is about (e.g., self, family, and community) and the degree of control over data collection.

Visualizations in personal informatics tools are often personalized by using unconventional encodings such as visual metaphors, pictograms, and abstract drawings. For instance, UbiFit Garden uses the metaphor of a garden that blooms based on the performance of a user's physical activities [53]. Paper bullet journalists employ personally meaningful representations to meet their practical and emotional needs in tracking different types of data [18]. Such subjective visualizations can convey a unique perspective in personal visualizations [189], encourage further exploration of data [196], and establish a better sense of identity [107], although perceptions and preferences regarding designs of personal visualizations may depend on individual personalities [172].

While the past research suggests potential benefits of having personal visualizations of the data, users still do not have full control over them in existing personal visualizations. While individuals can use standalone visualization tools like DDG to create visual representations for their personal data, these tools have not been considered in a personal context. That is, visualization construction is considered as an afterthought not being part of data collection. DataSelfie can allow for prescriptive, rather than descriptive, visualization design tightly coupled with a data collection plan, generating a final visual in real time based on data input from a user.

A graphical element may carry data information and also perform a design function usually left to non-data-ink.

Edward Tufte

3 Designing Expressive Data Graphics

This chapter present *Data-Driven Guides* (DDG), a novel technique for designing custom graphics driven by data. Standard charts like bar charts or scatter plots are now well-established and can be integrated into analytical tools to facilitate rapid exploration of data. On the other hand, thoughtfully crafted custom visualizations can be highly engaging and have the power to communicate the semantics of the data, that is not possible with the standard charts. However, existing visualization tools have limited support for customizing visual representations of the data, failing to meet this need.

Following the principle of information encoding, DDG takes a highly flexible approach to create custom visualizations by generating guides from data and using the guides to place and measure custom shapes accurately.

3.1 Motivation

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 [114]. 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 [145,36,79]. 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 [74]. 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. [29], the

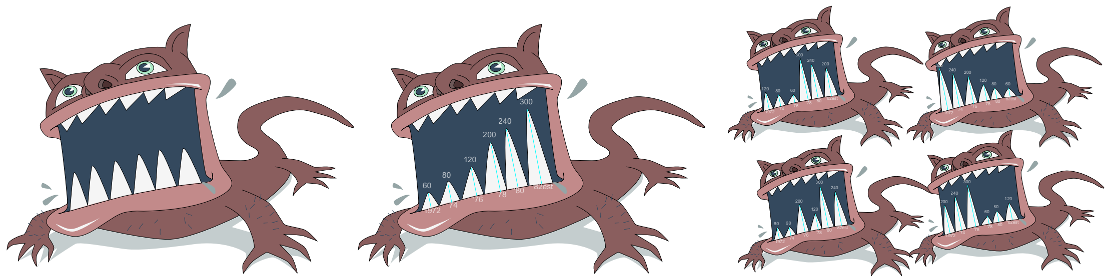


Figure 3.1: Nigel Holmes' *Monstrous Costs* chart, recreated by importing a monster graphic (left) and retargeting the teeth of the monster with DDG (middle). Taking advantage of the data-binding capability of DDG, small multiples are easily created by copying the chart and changing the data for each cloned chart (right).

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* 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.[29] 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 environment. 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

*http://www.adobe.com/Illustrator

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[28,136]. 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.

3.2 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

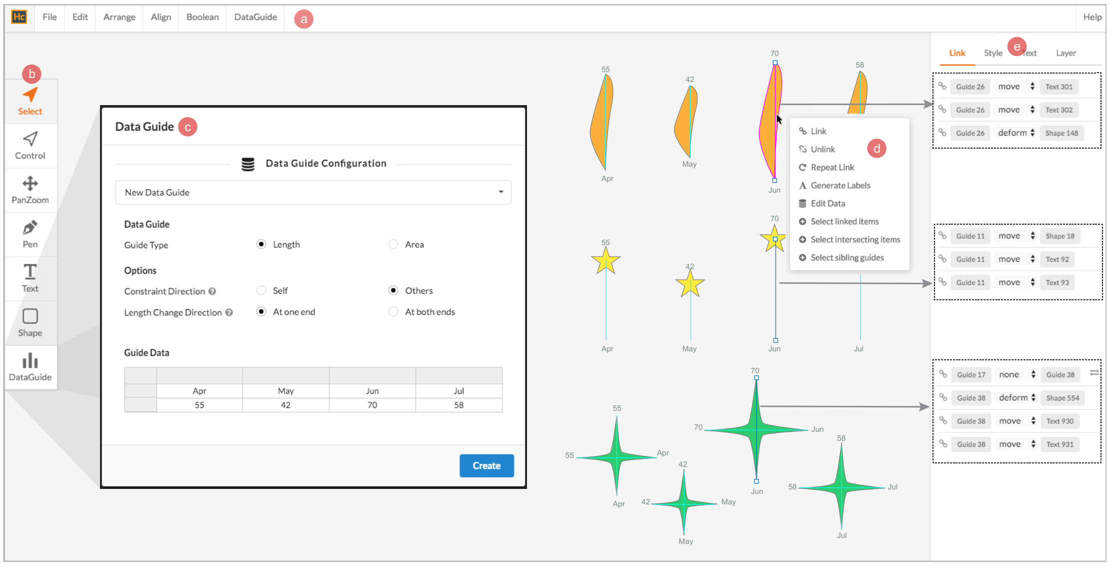


Figure 3.2: Overview of the vector drawing tool in which DDG is implemented: (a) top menu bar providing conventional features such as undo, redo, and SVG import & export operations. The menu item for DDG is the same as the context menu. (b) left toolbar providing the DDG tool as well as other selection 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.

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' existing 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 [29]. 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 [30] 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 [29]. 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.

3.3 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,

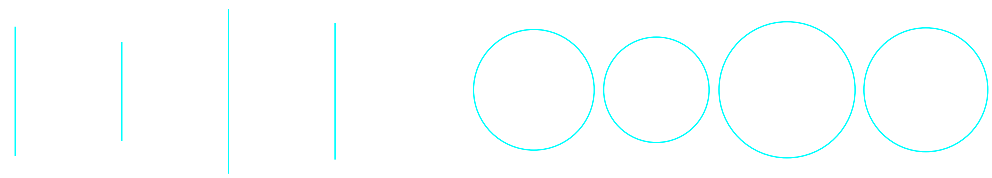


Figure 3.3: Length and area guides that can also be used as a position guide.

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 chapter.

We follow the theoretical frameworks of visual encodings[28,136] 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. 3.3). The resulting guides can be used to encode positions as well (See balloons in Fig. 3.11f). In addition, more than one guide can be combined to create more expressive visual structures (Fig. 3.11f-i). The visual variables—length, position, and area—are popularly used in infographic design[36], 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 [†]). 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.

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

3.3.1 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. 3.4).

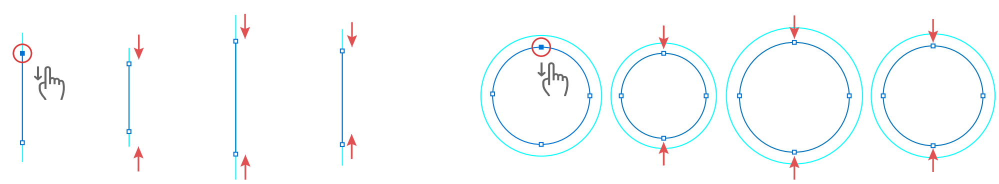


Figure 3.4: Changing a data guide will affect its siblings in order to preserve the relative size differences within the same group (blue colors represent new guide states and red colors represent the direction of user manipulation, e.g., grabbing and moving an anchor point.).

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. 3.1 right). Also, combining different groups of guides has the effect of combining different datasets graphically.

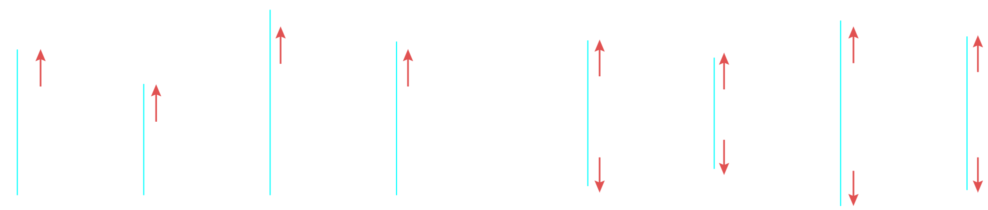


Figure 3.5: Users can specify the direction in which a length guide change its length: either endpoint (left) or both endpoints (right) of the line segment.

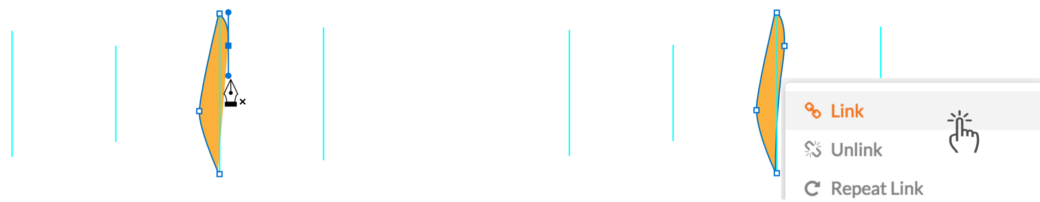


Figure 3.6: Drawing a shape directly on top of data guides will link the shape to the guide (left). Otherwise, users can explicitly link the shape with the guides (right).

3.3.2 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. 3.6 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. 3.1); 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. 3.9), 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. 3.11e). 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. 3.2c); i.e., both endpoints or either endpoint of the line segment (Fig. 3.5). 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[51].

Users can combine multiple data guides in order to construct more expressive structures (Fig. 3.11c, 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).

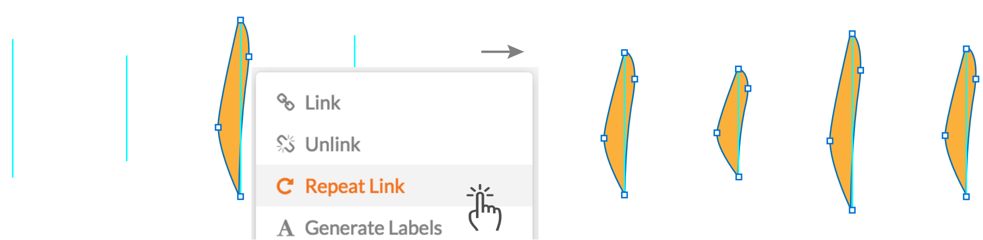


Figure 3.7: Selecting and repeating a shape will duplicate the shape over the sibling guides of its linked guides.

To further assist in drawing a data-driven graphic, we provide a number of visualization-specific

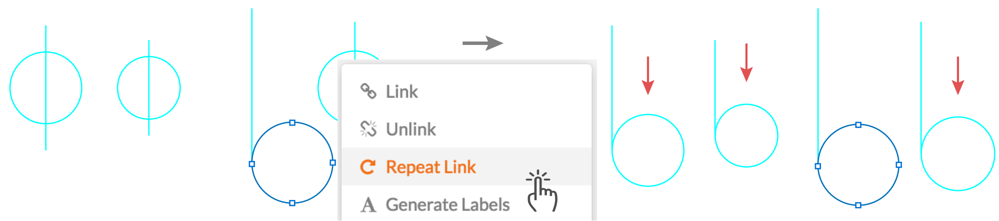


Figure 3.8: Selecting and repeating a guide (area) will reposition its sibling guides based on its position relative to the linked guide (length).

features. The repeat feature in DDG allows an associated shape to be repeated over its sibling guides in the same group (Fig. 3.7, 3.8). 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 3.11i). 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 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.

3.3.3 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. 3.10).

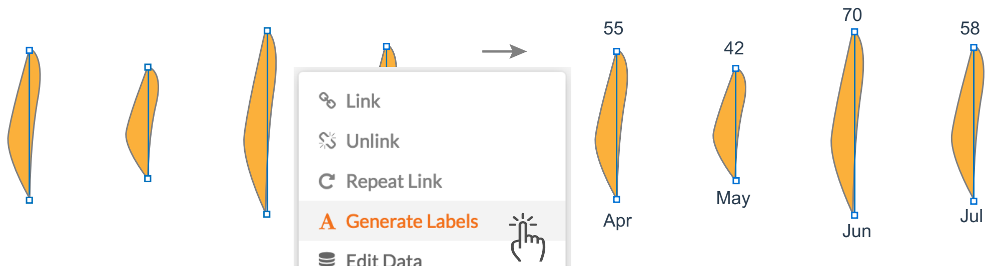


Figure 3.9: Labels are generated only when requested and are automatically linked to data guides when created.

To address the challenge of binding data to freeform shapes, we employ a deformation technique for vector graphics [100,127] 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.

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. 3.11g), and likewise multiple guides can be associated with a single shape (e.g., a cloud has four area guides attached in Fig. 3.11h). In the latter case, all the guides are used to transform

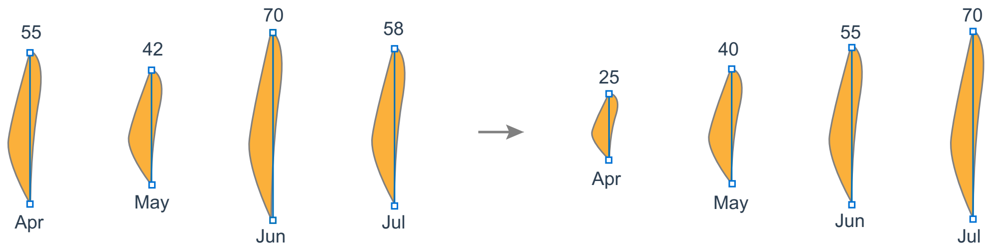


Figure 3.10: Updating data will transform shapes using the guides as the backbones of the shapes.

the shape. In addition, data guides can be linked to themselves, allowing more complex structures (Fig. 3.11c, f-i). For example, a length guide can be used to position an area guide (in Fig. 3.11g, i) or a group of guides (in Fig. 3.11f, 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.

3.4 Interface Design and System Implementation

Figure 3.2 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 3.2c). 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

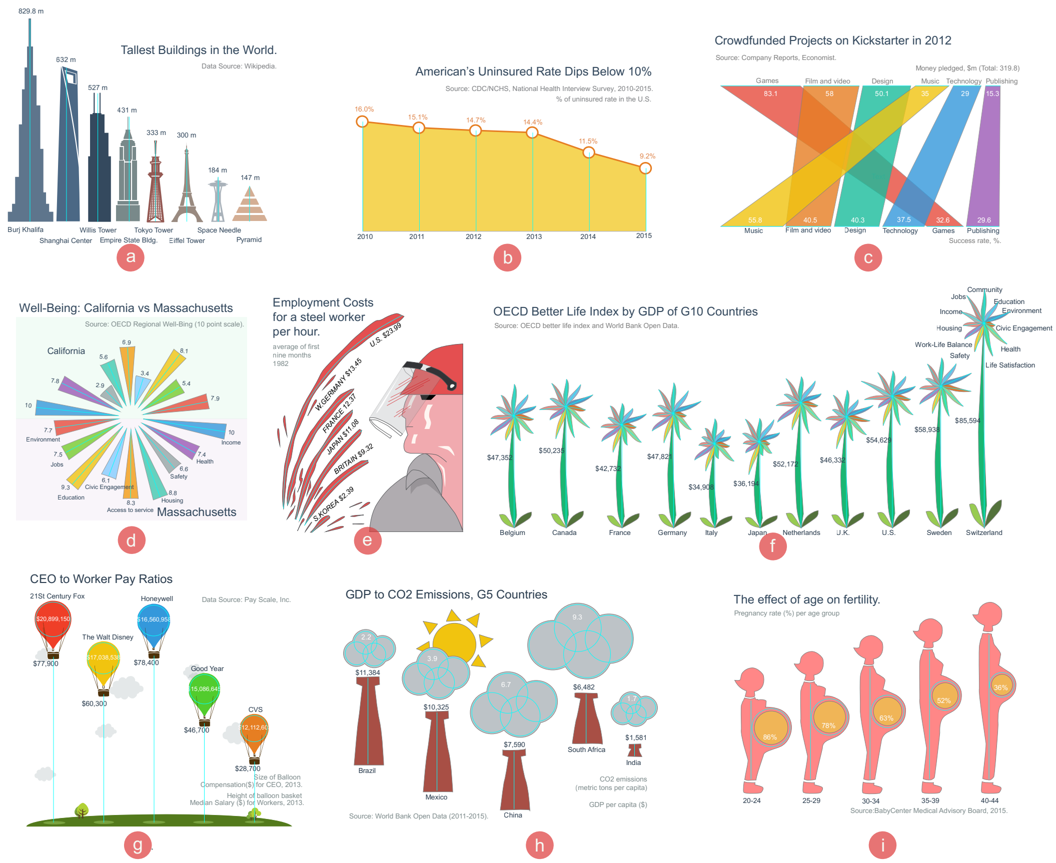


Figure 3.11: Examples created with DDG. (a) An isotype chart using data guides to measure the heights of imported building icons. (b) An area chart using a single stroke to draw the area and to encode slopes in the declining trend. (c) A sankey-style diagram where two DDG are juxtaposed to compare the 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.

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 3.2a, d).

In the right panel (Figure 3.2e), 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 3.2e). 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 [‡], 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 chapter.

3.5 Evaluation

3.5.1 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

[‡] http://paperjs.org/

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. 3.11. For example, they can create visual marks from a single stroke (Fig. 3.11b), using the repeat command (Fig. 3.11d), or by drawing different marks for individual guides (Fig. 3.11a). They also can create a layout using simple alignment and distribution features (e.g., aligned to bottom in 3.11c) that are typically available in any graphic design tool or using the layout functions (e.g., radial layout in Fig. 3.11d).

In addition to changing the basic styles of charts, users can create further embellished data graphics. We recreated two Nigel Holmes' infographics[87] that were used in a previous evaluation study investigating the effects of visual embellishment[27]. For the factory worker chart (Fig. 3.11e), 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. 3.1). 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. 3.1).

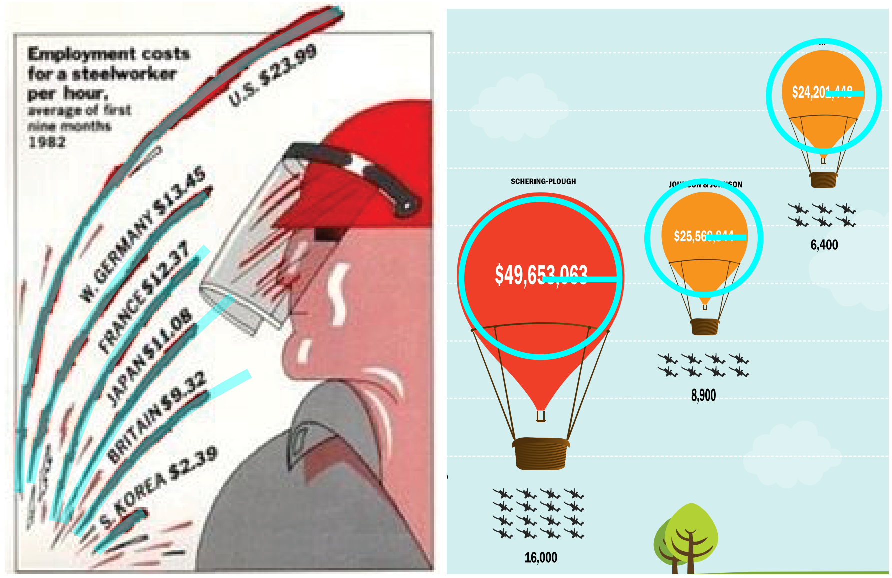
More complex visual structures can be constructed by combining multiple guides. In Fig. 3.11i, two groups of data guides were combined side-by-side (i.e., each area guide was linked to each length guide). A human shape was then linked to an area and length guide and repeated for sibling area and length guides; i.e., both the area and length guides act as the backbone of the shape. Another example is the flower chart (Fig. 3.11f) 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. 3.11f 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. 3.11i 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. 3.11c).

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. 3.12a) by Nigel Holmes may have an incorrect representation of the data. We also found a similar case in the balloon chart (Fig. 3.12b); 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.

3.5.2 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



(a) (b)

Figure 3.12: (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.

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. 3.2). 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 [§]. 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

[§]http://data.worldbank.org/

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).

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 3.2e), 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).

3.6 Discussion

3.6.1 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

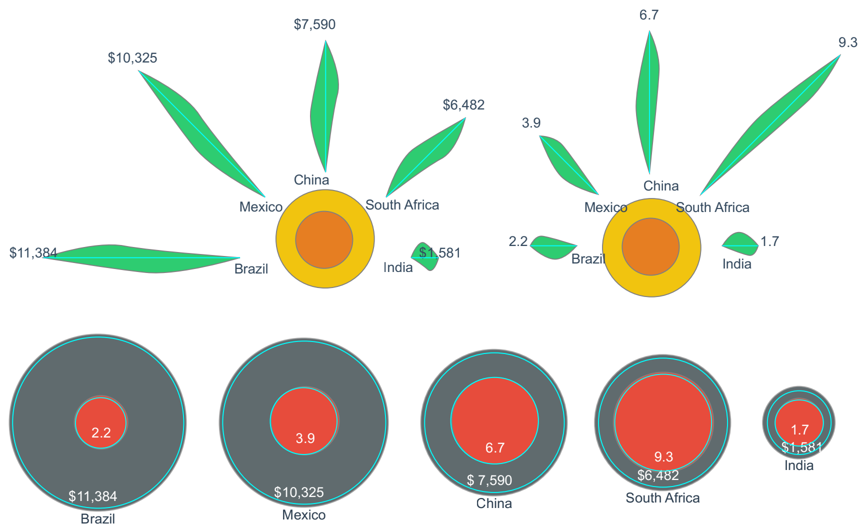


Figure 3.13: Participant-generated graphics in the third task in the user study, further embellished by the first author.

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[194] and tangible blocks[97] 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 [135] (e.g., calculating areas of visual marks to check whether they match actual data represented through data guides).

3.6.2 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 [¶] into visualization design environments. Directly manipulat-

[¶] http://paradrawing.com/

ing 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 [175]. For example, how can we develop computational tools to support freeform data sketching [‖] or tangible visualization construction [**] 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 [97] can make use of a large database of icons [††] as tokens to create diverse isotype charts.

3.7 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 [179]. 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

‖ http://www.dear-data.com/

** http://dataphys.org/

†† https://thenounproject.com/

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.

3.8 Conclusion and Future Work

In this chapter, we introduced 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.

The fundamental thing about sketching is that it is about asking not telling.

Bill Buxton

4 Creating a Personalized Visual Vocabulary

This chapter describes DataSelfie that enables individuals to design a visual vocabulary to represent their personal data. Visualization often serves casual and personal purposes. Individuals collect data about themselves, analyze and visualize the data to promote self-knowledge and share the visuals of the data to encourage conversation with others. However, existing personal informatics systems mostly focus on data collection, specifically automated tracking, and use predefined presentations of the data. Users remain mostly passive and less engaged with the data as such. DataSelfie augments a familiar questionnaire editor with a drawing tool, allowing users to actively construct visual mappings for the data to be collected.

4.1 Motivation

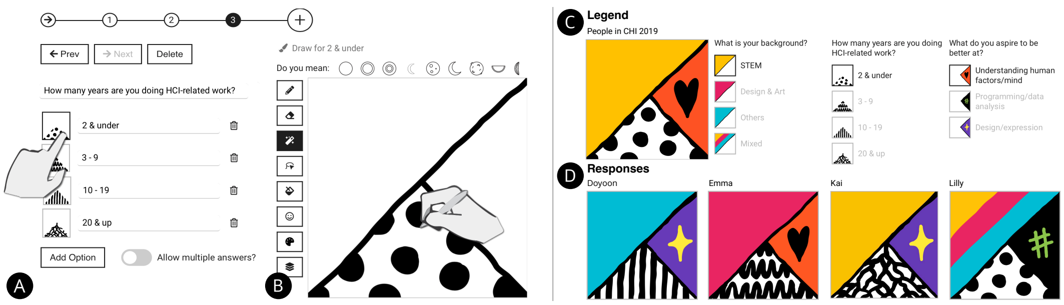


Figure 4.1: DataSelfie: A) The questionnaire editor for data collection. B) A visual mapping canvas in which a user can draw a unique personalized visual for a selected option. C) An interactive legend aids the interpretation of the visual mappings. D) Each questionnaire response generates a distinctive visual.

Data can capture a snapshot of the world and allow us to understand ourselves and our communities better. Giorgia Lupi, a renowned visualization artist, recently advocated for *data humanism*, a more personal approach to collecting, analyzing and visualizing data[131]. In contrast to traditional data systems that focus on rapid processing, automated analysis, and summary presentation of a large quantity of data, data humanism puts more emphasis on perhaps slow but deeper engagements with data, as well as unique expressive visuals that embrace the context and subjectivity of the data they represent.

Lupi's manifesto echoes in a lot of existing research in the areas of personal informatics, self-tracking, and casual visualization[91], including an abundance of tools for collecting and visualizing personal data[53]. However, most of these tools focus on data collection, specifically automated tracking, and use predefined presentations of the data. Users remain mostly passive and less engaged with the data as such. On the other hand, there has been a surge of construction tools allowing people to create custom visualizations of data [112,207]. Notwithstanding, these tools investigate authoring processes and tool expressiveness and have not been applied to a personal data context.

In this chapter, we are investigating the question of how to equip people with the ability to design their own visual vocabulary to represent qualitative and nuanced aspects of personal data. As an initial step toward understanding the empowerment through personal visualizations, we analyzed the Dear Data project [134] in which Giorgia Lupi and Stefanie Posavec collected, visualized, and shared the visual postcards of their data on a weekly basis for about a year. We performed open coding on the visual postcards as well as their reflections on the process of making them. We derived several design implications for personal informatics, including values of qualitative data for deeper reflection of self and common design elements used to personalize the visuals of the data.

Informed by the analysis of the Dear Data project, we developed DataSelfie, a web-based interactive system designed to enable any individuals to collect their data and decide how to visualize the data. DataSelfie combines a familiar survey authoring interface with a drawing capability so that users can create any questionnaire to ask questions about themselves and design a personalized visual vocabulary to represent the collected data. Unlike existing personal informatics tools, DataSelfie is offering users full autonomy over the visual presentation of their data. It also makes it possible to design the visual vocabulary at the time of creating the questionnaire; the final visual output is determined later on based on the response to the questionnaire. In this way, the construction of visualizations is not an afterthought, but a primary activity that users actively engage in while thinking about the goals of their data collection.

Users can use DataSelfie in both individual and collaborative scenarios. They can collect data on a recurring basis for self-tracking purposes or share a questionnaire with others to capture individual identities in a group setting. In the latter case, the small multiple of individual responses can create a single collective visualization representing the group. The users can also share their visual responses with each other, facilitating communication among individuals through data alone. Through a user

Figure 4.2: An example postcard with the theme *distractions*: Giorgia Lupi (left) and Stefanie Posavec (right).

study with 14 participants, we found that DataSelfie provides an easy-to-use and enjoyable interface to gather and visualize data. The variety of examples they created also suggests that DataSelfie enables to create expressive visualizations of data.

4.2 Analyzing Dear Data

To understand what types of data can capture personal lives and what kinds of visuals can be used to represent the data, we analyzed the Dear Data project[134] in which Lupi and Posavec collected data about themselves and drew custom visualizations with the data every week for a year.

4.2.1 Data

The project has a total of 104 postcards that spanned 52 weeks; each designer creates each half of them. The postcards contain hand-drawn visualizations of data about their lives along with a visual legend explaining the visual encoding and context of the data (Figure 4.2). The project also accompanies each designer's weekly retrospective, reflecting on the experience of creating and sharing the postcards, which is available on the website[*].

[*] http://www.dear-data.com/by-week/

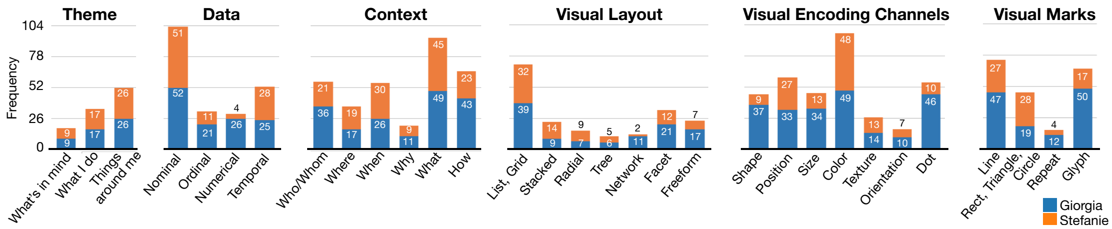


Figure 4.3: A histogram of categories identified through the open coding of 104 visual postcards.

4.2.2 Method

We used qualitative methods of open and axial coding to analyze both postcards and retrospective texts. While visual postcards revealed common design elements to represent personal data, retrospective texts allowed us to gain insights into the underlying rationale for the design choices.

4.2.3 Results on Visual Postcards

Figure 4.3 shows the overview of extracted categories and observation counts out of 104 visualizations. Among three major themes identified, *things around me* (52) and *what i do* (34)—external states—were more common than *what's in mind"* (18)—internal states. Regarding data scales, *nominal* (103) and *ordinal* (32) were used most often in comparison to *numeric* (30). In fact, numerical data was also often binned into ordinal categories (8 out of 30).

Regarding the types of contextual questions addressed, we found *what* (94, e.g., what I did? what for?) was the most frequently used. *how* (67, e.g., how long or how intense was it?), *who/whom* (56, e.g., who were involved?), *when* (56), and *where* (36) were next in the order of frequency, while *why* (20) was the least frequently observed. About half of the time (54), they used *visual annotations* (e.g., circling or giving a unique visual) to highlight specific items (e.g., husband or boyfriend) and to communicate further context (e.g., missing or uncertain data).

They regularly used standard layouts such as *grid* (50) and *list* (22). For example, when showing data over time, it was common to arrange items in chronological order or distribute them in a

2-dimensional space in which both axes represent time and day respectively. We also observed moderate uses of other layouts such as *facets* (33), *stacked* (23), *networks* (13), and *trees* (11). They also often used *freeform* layouts (24), such as randomly distributing items in space. In most cases, they used more than one layout in one postcard.

Most frequently used visual variables include *color* (97), *position* (60), *size* (47), and *shape* (46). Size encoding was not necessarily used for numerical data but also for categorical data as well. They also often used *texture* (27), *fill/no fill* (12), and *orientation* (17), which are not commonly observed in digital visualizations. Most often used visual marks include *lines* (74), *glyph* (67), and *dots* (56), while *circles* (32) and *rectangles* (14) were also common. We also observed *repeat* marks (16, multiple same elements are stacked to encode a single value) and *real objects* (3, e.g., cosmetics).

There was also a notable difference between Lupi and Posavec (Figure 4.3). For instance, Lupi used more numeric scales compared to Posavec (26 vs. 4). As a result, Lupi used more size encoding as well (34 vs. 13). Lupi's visuals were fine-grained while Posavec's visuals were vibrant but straightforward. For instance, Lupi also frequently used small visual attributes (e.g., *dots*: 46 vs. 10, and *glyphs*: 50 vs. 17) attached to a primary mark (see Figure 4.2).

4.2.4 Results on Retrospective Texts

We identified five different stages of the process. We summarize what each stage entails below. We use 'P' for Posavec, 'L' for Lupi, and 'W' for both along with a week number to indicate the source of evidence.

Preparation. Lupi and Posavec chose a theme in which they are both interested. They often decided the theme for a performative purpose as a way to promote certain behavior (e.g., *Being nicer* - W23). They interpreted the theme in their personal contexts, such as thinking about how they perceive it or reflecting on the past experience they had. The choice of the theme and the interpretation had an impact on what kinds of data to collect and tools to use. They often set up specific questions and categories for data tracking prior to data collection, while in other cases they jot down logs about the topic and sort through the logs afterwards.

Data collection. They mostly took qualitative methods for gathering data, employing both analog and digital tools (e.g., notebooks, Reporter, Moves). The methods were largely divided into two kinds: 1) surveying data at once and 2) tracking data over time. A side effect of this qualitative approach was that they were aware of gathering data, which often had an impact on their moods and behaviors throughout the week (e.g., recording positive feelings creates an optimistic mood - L31). This often made data collection *performative*, intentionally manipulating certain data points.

Struggles in the manual process involved capturing transient things and noting down every instance of a data point (e.g., having to pause and note a laugh in a circle of people - P42). In particular, the latter resulted in what they call *data-gathering fatigue*. They embraced this imperfect nature of the manual process; for instance, *data voids* indicating a special moment such as a wedding or being drunken. Similarly, they marked with a special mark to denote playful data manipulation by others (e.g., *Physical contacts* - W6).

While gathering data, they reflected on their characters, habits, and preferences. For instance, they reflected on how they organize things (P16), where they have come from (P22), and what makes them happy (L31).

Data processing. Once data is collected, they would organize the data to find interesting facts and stories that they found meaningful and want to share. They often added additional data to provide context (e.g., adding demographic information to friends - L25) or simplified the complexity of the data to ease the drawing or understanding of it (e.g., highlighting the top 5 emotions - P11). This stage also served as a firewall to ensure privacy by hiding certain information in the data (e.g., crossing out the husband's name - P14), which they enjoyed indicating it in the postcards. While aggregating and categorizing data, they discovered new aspects of themselves and had opportunities for self-reflection. The final organization had an influence on mostly visual layouts and groupings in postcards.

Visual encoding. They sketched and iterated on multiple ideas before reaching the final drawing of the data. They mostly used colored pencils, as well as pens & markers, but often incorporated new materials and drawing techniques (e.g., cut and collaged papers - W20 and lipsticks - L19). Most frustrations in this stage came from the hand-drawing of the data, including difficulties of redrawing mistakes, maintaining accuracy, and mapping too many data points.

A variety of factors inspired the visual representation of the data. These include: 1) visual metaphors for data or theme (e.g., scribbles alluding the textured sonic waveforms - P32), 2) personal visual metaphors for data or theme (e.g., musical scores for complaints - L07), 3) personal styles and preferences in general (e.g., plant-based shapes - P14 or abstract arts - L14), and 4) random and spontaneous drawings rather than dictated by data.

There was a tension between readability and aesthetics; e.g., they were not pleased with the legibility of data insights when they liked drawings, and vice versa. Often, aesthetic focus was intentional, such as to detract attention from data (P27) or to compensate for lack of patterns in the data (L14). In general, they appreciated having control over aesthetics by noting that both beauty and functionality are important and which one to emphasize depends on data and context.

Sharing & reflection. By sharing visual postcards, they learned about each other and improved self-knowledge via comparisons. They appreciated different ways of interpreting a theme (e.g., the definition of a nice act - W23), collecting data (e.g., close vs broad range of friends - W25), and drawing postcards (e.g., circular vs linear layouts - W06).

The postcards brought back vivid memories of each week, providing ample rooms for reflection. They also reflected on the impact of the Dear Data project on their lives, including changes in future behavior (e.g., "After this week, I really figured I could be nicer and credit people more often..." - L15) and the value of the postcards as personal records.

4.2.5 Implications for Personal Informatics

Below, we summarize key insights we learned from the analysis results. Please note that the project involved two professional designers and thus the results may not generalize to a general audience.

I1. Capture qualitative aspects of self. Our findings suggest that qualitative data may enable more nuanced and richer reflection that is not possible with current automated tracking. A challenge is that the data is often ephemeral or intrusive to gather.

I2. Reveal missing and uncertain data. We observed that inaccurate and missing data is embraced in a personal context. The mistakes and failures are part of the data and can provide additional insights into their lives.

I3. Provide different modes of data collection. Data gathering can be either tracking data over time or surveying it once. In the former case, it may start with a specific collection plan or involve iteratively refining data or discovering it during post-processing.

I4. Support data exploration for story harvesting. Visualization for communication and sharing would require more than listing all data points. Filtering & simplifying data and classifying it into categories are ways to experiment with different personal stories before the final visualization.

I5. Use visual annotations to encode moments. Annotation and highlighting are useful tools for indicating specific data and memorable moments in a personal visualization. They can be also used to add humor or a personal touch to the visualization.

I6. Enable designing personalized visuals. Designing a unique visual of personal data may increase enjoyment and attachment toward the data. It is also a means for self-expression, communicating a perspective on the data and revealing one's personality through the visual.

I7. Choose visual variables for personalization. We found that *color* and *shape* variables may be more

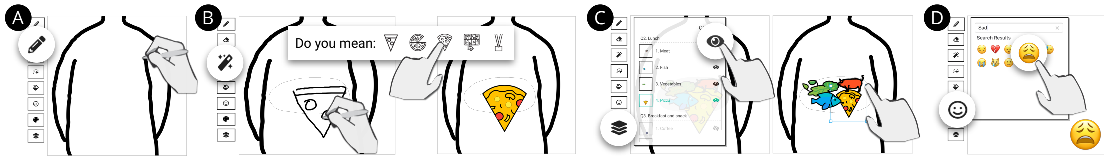


Figure 4.4: The drawing interface of DataSelfie: A) Drawing canvas, B) Auto-drawing, C) Layer view and selection, D) Emoji tool.

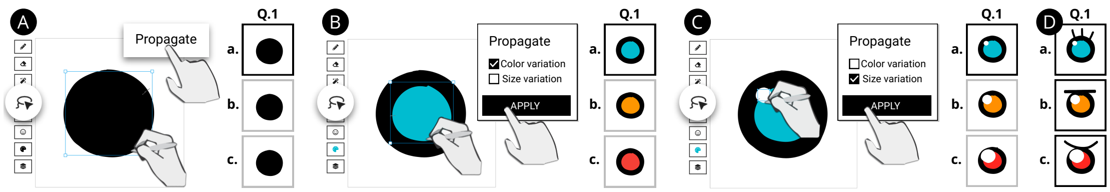


Figure 4.5: Automatic encoding support: A) Reusing a shape, B) Color encoding, C) Size encoding, D) Customization.

versatile for customization, while quantitative variables such as *position* and *size* are mostly governed by data. Traditionally underused variables (e.g., *texture* and *orientation*) due to lack of perceptual effectiveness might be acceptable in a personal context.

I8. Leverage visuals as personal documentaries. Our findings also point to the forgotten value of visualization; i.e., records. While communication and analysis are the focus of current visualization research, visual postcards demonstrate the use of visualizations to record personal memories.

I9. Address challenges with qualitative processes. We observed that main struggles lie in data collection and visual encoding stages. Our findings mostly confirmed previous research[125,29], including forgetting to gather data and recovering from mistakes. A notable difference is the difficulty of tracking transient things like emotions or smiles.

I10. Support conversations through data. A significant value of sharing is that it opens up opportunities to see one's data in the context of the other's data. Comparing and contrasting differences in data as well as in visuals can provide additional channel for self-reflection. Also, others might help to identify different data insights that are not intended by the author.

4.3 Design Decisions

Informed by the analysis of the Dear Data project, we wanted to build a system for individuals to engage with their personal data via creating expressive visual representations.

Our design goal is provide structured support for novices to crisply articulate the data they want to collect and visualize, as opposed to the Dear Data project that required substantial expertise in data collection and visualization.

We also draw inspirations from the work by Lupi and the Accurat team on automatically generating data portraits of individuals based on their responses to a survey [133,132]. We aim to generalize this idea to a general audience along with the insights learned from the Dear Data project.

D1. Using a questionnaire to collect qualitative data. (I1, I3, I9) To enable flexible collection of personal data, we seek to leverage the familiar experience of creating a questionnaire in a survey form. The questionnaire can serve as a data collection plan as observed in the preparation stage. It should be also editable to allow for revisions. The questionnaire is versatile to support contextual questions (e.g., who, what, why etc) and can capture diverse aspects of daily lives.

D2. Designing a personalized visual vocabulary. (I6, I7, I9) To allow people to craft unique and expressive visual representation of their data, we propose to augment the questionnaire editor with a drawing canvas in which a user can draw visual mappings for collected data. The visual mappings create a personalized visual vocabulary and should be also editable to allow for iterative design. The mappings can be constructed after data collection or at the time of creating the questionnaire; the latter has a benefit of visualizing the goals of data collection in advance.

D3. Support sharing visuals of data: (I8, I10) To foster collaborative reflection, we intend to support sharing data visuals with others. To assist in interpreting visual mappings, a legend should be generated and accompanied with the visuals. Also, sharing with more than two people should be possible.

4.4 The DataSelfie Interface Design

DataSelfie consists of three main components: a questionnaire editor (Figure 4.1A) and a canvas for drawing visual mappings (Figure 4.1B), an interactive legend (Figure 4.1C), and a response viewer (Figure 4.1D).

4.4.1 Creating a Questionnaire

The questionnaire editor has the same interface as other survey tools like Google Forms. It currently supports a multiple choice question (radio buttons) with an option to allow more than one answer (checkboxes). This type of question enables collection of mostly qualitative data (I1), i.e., categorical data, and also renders a rapid response by quickly selecting one. Although we do not support numerical data, users can still capture quantitative data by binning it into intervals (e.g., less than 10, 10 to 100, more than 100) as observed in the Dear Data project. While out of scope for this chapter, the questionnaire editor can be extended to incorporate automated tracking similar to OmniTrack[113].

Collecting Data. To collect a data point, a user simply needs to respond to the questionnaire, which is similar to filling out a survey. The user can respond multiple times, generating more than one data point. To assist data collection on a recurring basis (e.g., daily, weekly, or monthly), DataSelfie supports setting a reminder email to prompt for a new response (I9). DataSelfie also allows the user to refine the questionnaire during data collection (I3).

4.4.2 Drawing Visual Mappings

The questionnaire editor embeds a drawing canvas that affords free-form sketching so that a user can draw personalized visuals to represent each answer option per each question (I6, I7-shape). To construct a mapping from a question to a visual, the user can simply taps an option thumbnail and draw a corresponding visual on the canvas, and keep doing this for other options (Figure 4.1B).

In this way, the user can create a *personalized visual vocabulary* integrated with the questionnaire. That is, the visual vocabulary is a set of drawings tied to contextual questions for data collection, generating a unique visual per each response to the questions. DataSelfie allows iterative refinements of the visual mappings at any time (I9).

Drawing Tools. The drawing canvas comes with a set of tools including a pencil for sketching, selection tool, color fill, and palette (color, stroke, opacity). The user can use the layer view (Figure 4.4C) to see how the drawings for the currently selected option would interact with those for other options and questions. It is necessary since the same canvas area is shared across the questionnaire, stacking all drawing layers into a single layer to generate a final image when a user submits a response.

Assistive Drawing. To assist users with less experience in drawing, especially with a mouse, we integrated an auto-drawing feature [117] that suggests a predefined icon based on a series of strokes, dragging & dropping images, and a search box for adding emoji icons (Figure 4.4D). To enable reusing of the same shape across all options, we also support duplicating a shape across all options in a question, along with automatic size and color encodings (I7-color, Figure 4.5). The duplicate shapes compose a group constrained by the applied encodings; i.e., if a user changes the size of a shared shape, it also updates other shapes in the same group (I9).

Generating Visuals. Figure 4.6 shows a schematic diagram of how DataSelfie generates each visual response. Once a user submits a response to the questionnaire, it maps each answer option to a corresponding drawing while other drawings for other options are hidden in the final outcome. In this way, the visual vocabulary can generate a combinatorial number of visuals.

We currently take a layered approach to model the underlying visual vocabulary, which is similar to the layer model in a conventional drawing tool like Adobe Illustrator. Background and question layers are merged into a single layer, overlaid on top of each other in the order of questions. Each question layer can toggle the visibility of one or more option layers depending on the response.

4.4.3 Parametric Visualization Construction

Although simple, it is worth noting the way of generating a visualization is prescriptive than descriptive. The user create a visual vocabulary that can be used to generate a final visualization without actual data, rather than attempting to describe existing data. This is clearly different from typical visualization authoring tools as the users of those tools assume the existence of the data. In our case, the generation process is more close to parametric design. Each question can be considered as a parameter whose value is later specified by the user.

4.4.4 Sharing & collaboration with others

To facilitate a casual conversation through data, a user can share the questionnaire with others (I10). Each of them can then submit a response and receive a unique visual, contributing to a collage of visual responses (Figure 4.1D)[132]. DataSelfie also supports sharing each visual response with others using the hyperlink generated after submitting a response (I8). To assist in interpreting the data selfies, DataSelfie automatically generates an interactive visual legend from the visual vocabulary (Figure 4.1C).

4.5 Usage Scenario

We introduce individual and collaborative scenarios that correspond to two kinds of data collection scenarios we observed in the Dear Data project: tracking data over time and surveying data once.

4.5.1 Scenario 1: Self-Tracking

This scenario introduces how an individual user might use DataSelfie for self-tracking or lifelogging.

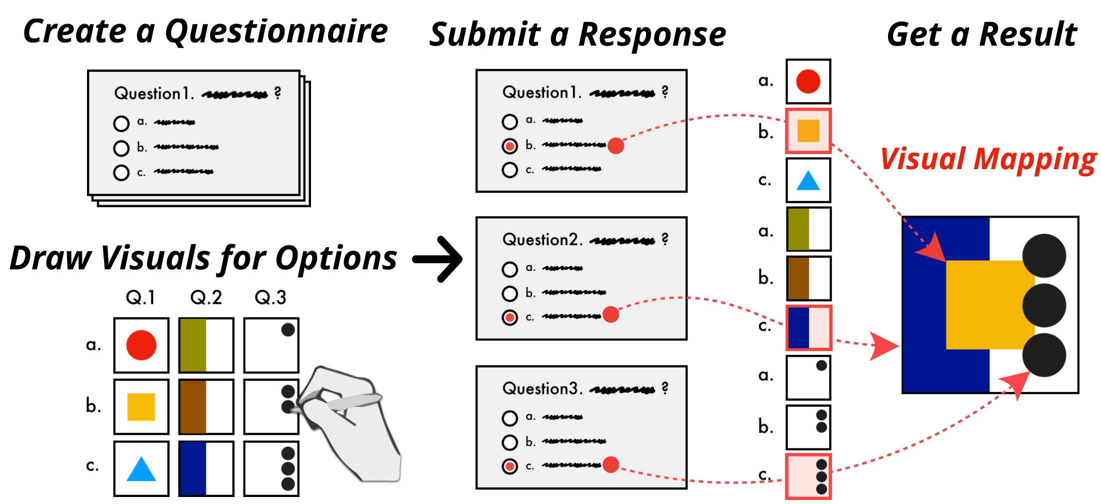


Figure 4.6: The production rule of a visualization in DataSelfie, generating a unique visual for each questionnaire response.

Sam is recently concerned that he eats pizza a lot. He would like to have a healthy diet but struggles to do so due to a busy school life, and decided to track what kinds of food he eats every day. He starts by creating a questionnaire with three simple questions asking what types of food he had during the morning, afternoon, and evening (Figure 4.4).

Once finishing the questionnaire, he begins drawing to visualize his personal goal of the questionnaire. He wants the visuals to reflect his diet directly. He first draws a background representing his body (Figure 4.4A). For each option, he draws a visual representing the type of food he may or may not want to eat. He thinks he is not good at drawing, so uses the auto-drawing feature to draw the pizza (Figure 4.4B) and adds an emoji to represent his feelings after the diet.

He set a daily reminder to fill out the questionnaire. When he received a reminder through an email, he taps the link using his mobile phone to go to the questionnaire to submit a response. He often intentionally eats healthy food in order to get the visual he wants, making the data collection performative. At the end of the month, he reviews all the responses in a collective form. He gets a sense of how he maintained his goal by quickly gauging the prominence of all the greens.

4.5.2 Scenario 2: Community Gathering

The second scenario demonstrates a collaborative activity, similar to the one demonstrated by Giorgia Lupi in which she used a questionnaire to survey fun facts about conference attendees [133].

Emily is an organizer of a HCI workshop. She wants to engage the attendees in a way that they get to know each other as they belong to the same research community. She thinks about the topic of the workshop and comes up with a set of questions that may be well suited to represent members of the group (Figure 4.1C). She draws a background that serves as a template for all questions like a coloring book. The drawings for each question fills a different region in the background (See divide lines in Figure 4.1B).

Since this is not about tracking data over time, she turned off reminders. Instead, she shares the web link to the survey with the attendees. Each attendee gets a unique visual that captures their identity based on their response (Figure 4.1). Emily prints this visual and make a personalized badge for each attendee. She also creates a collage of all responses showing the community collectively (Figure 4.1D).

4.6 Implementation

DataSelfie is a web-based application written in Javascript. It uses React.js [63] for building user interface components and Redux.js [64] for application state management. We use a Flask [167] web server and MongoDB [146] to persist the user and questionnaire data and enable sharing through hyperlinks. We heavily use Paper.js [124] to implement the drawing canvas integrated into the questionnaire editor. For assistive drawing, we leverage Autodraw [117] and Emoji Mart [144]. DataSelfie will be open-sourced and available online upon acceptance.

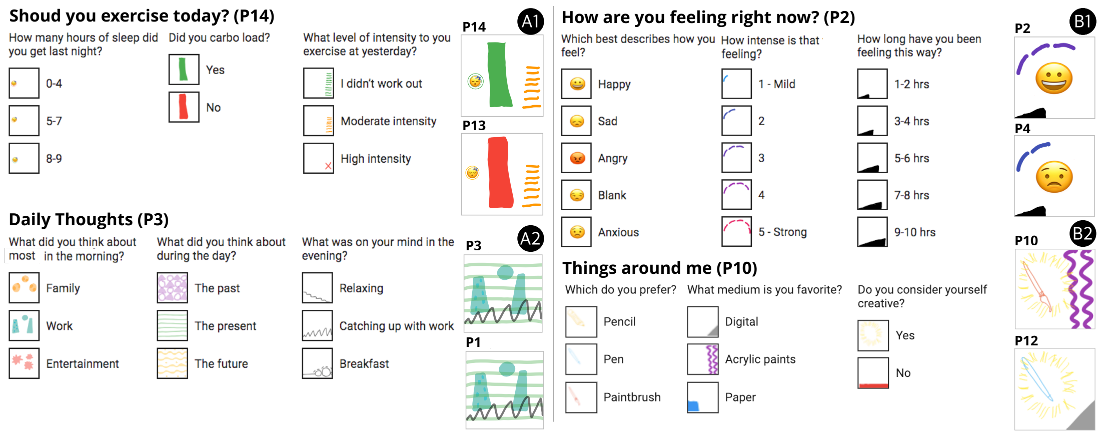


Figure 4.7: Four examples of questionnaires created by participants during the third task for two usage scenarios: A: tracking recurring states over time and B: capturing a current state.

4.7 User Study

We conducted a qualitative user study to evaluate the usability of DataSelfie and gain insights into its usefulness with an emphasis on lifelogging.

4.7.1 Participants

We had a screening survey to recruit 14 participants (nine female and five male, six aged 18-24 and seven aged 26-35, one aged 36-45) who have prior experience on manual journaling and automatic self-tracking, as well as basic tool literacy on survey tools and presentation software. We used Safari browser on iPad 12.9 inch using Apple Pencil. We paid participants with a $25 gift card for an hour-long session.

4.7.2 Procedure & Tasks

Each hour-long study session started with a background survey at the beginning. We had a short tutorial and four tasks in total. For the tutorial, we used an example adapted from an existing template[†].

In the first task, we asked participants to reproduce the example from the tutorial in order to make them familiar with DataSelfie. In the second task, they were asked to replicate a new example as quickly as possible without any guidance from a researcher. We varied the examples in the two tasks so that users can see two use case scenarios: 1) surveying a single state or current identity and 2) tracking recurring states over time. In the third task, they needed to design a new questionnaire with a minimum of three questions and craft the corresponding visual vocabulary. We assigned each participant to a specific scenario (A. *tracking* or B. *surveying*) and topic (1. *what's in mind*, 2. *what I do*, or 3. *things around me*).

We concluded the session with a usability survey adapted from the System Usability Scale[41] using a 5-point Likert scale (1—strongly disagree, 5—strongly agree), and a semi-structured interview discussing the overall experience and potential benefits in their current practice.

The fourth task happened later online, moderated by a researcher. We paired two participants who were assigned to the same scenario and topic in the third task. They submitted responses to each other's surveys and shared the final responses. We asked them to reflect on how they perceive the experience of sharing and exchanging the visual responses.

4.7.3 Results

All participants completed all the tasks. For the second task, it took 4 minutes on average (min=3, max=6), while the third task took 13 minutes on average (min=9, max=17). In the usability survey, participants rated higher on the ease of use (M=4.43, SD=0.76), the learnability (M=4.36, SD=0.74),

[†] https://ideas.ted.com/how-to-draw-your-own-selfie-using-your-personal-data/

and the usefulness of the tool (M=4.29, SD=0.83).

Diversity of questionnaires and visuals. They generated a variety of questionnaires (Figure 4.7, see the supplement for the full collection). The question topics include exercise, feelings, music, sleep, meeting, etc. A majority of questions were *how* and *what* questions, while some questions did not belong to 5W1H (e.g., Q: I like to take a shower. A: morning, afternoon, or before bed). In terms of visuals, all participants used *color* and *shape* variables while some also used the *size* variable using *repeat* (or *density*) marks (Figure 4.7B1). They also attempt to divide the canvas region to layout visual elements based on questions (Figure 4.7A1).

Ease of use & learnability. Participants had positive experiences with the tool with a few saying *"the interface is very intuitive"* - P1 and *"It is well-designed and fun to use"* - P5. One issue we found was foreseeing how layers will be merged, often ended up having next questions blocking previous ones. However, one participant deliberately manipulated this by having transparent elements to see through previous layers. Participants liked the assistive drawing support. One participant kept using the auto-drawing and said "I know it is not perfect, but I just like it" - P6, while another participant said *"I like having the emoji interface in a drawing tool"* - P8. One participant suggested to have drawing templates to further alleviate the fear of a blank canvas.

Enjoyment & engagement. Participants made positive comments on the engaging nature of the tool. For instance, they mentioned that *"I like how it integrates the answers into a succinct and amusing visual form"* - P5 and *"It reminds me of an art class. It would be useful for therapeutic purpose"* - P7. Other participants commented that *"I enjoyed the roughness of the hand-drawn sketches and I think the raw emotion feeling would be lost if the graphics were very clean cut and digitized."* - P4, and *"Freedom to draw anything to represent my data makes it fun and lively"* - P13.

Trade-offs in manual data collection. They liked the idea of collecting qualitative data, saying *"It's a good way to gain insights that can't be traced automatically"* - P11, and *"when it comes to qualitative* *data, it is more personal"* - P13. These comments are in line with the Dear Data project. However, a few participants said they would not use it for self-tracking due to lack of time for creating a questionnaire (e.g., *"I wouldn't use this for self-tracking. It takes too much time"* - P7).

Benefits of drawing personalized visuals. Participants' reaction to the drawing capability was mostly positive, saying *"it allows you to visualize goals and impacts in advance"* - P9, *"I like visuals as they invoke more thoughts"* - P7, and *"it would be good for emotion and well-being that lack clear forms"* - P8. Several participants also commented that *"it's more interactive and personal than existing (automatic) tracking tools"* - P13 and *"I think I would use it to track my menstrual cycles"* - P10. Others had mixed opinions: *"it's good for qualitative data, but probably not for quantitative data in which precision is important"* - P8, *"I would still like to see aggregate summaries"* - P12.

Benefits of sharing. They also made positive comments on sharing (P3, P11, P12), although it is too short of time to judge its true benefit. One participant mentioned that *"if there is one reason for me to use this tool, it would be the sharing aspect"* - P11; this participant expressed reluctance to use the tool because of potential privacy issues. Other participants commented that: *"surprised since she had the same response as me! I think the visual result helps to immediately tell how similar or different our responses were"* - P3 (Figure 4.7A2) and *"It was also very relatable to see how another (stranger) is feeling about the same subject"* - P4.

Other use cases. Several participants unexpectedly came up with different use cases that we did not intend to support, saying *"I can design a visual to lead to make a decision. It can make a real change on who you are in each date."* - P14 (See Figure 4.7A1). Other participants also mentioned, *"I think I can use this tool for testing different configurations for a garden I am designing."*- P1 and *"My psychologist friend would love to use this for her research tracking moods from subjects."* - P2.

4.8 Discussion

4.8.1 Limitations and Opportunities

The user study surfaced several opportunities for further improvements. We did not support question types for numerical data, as it can be binned to categories. However, we observed that participants were often annoyed as they need to write out all the intervals manually (e.g., Figure 4.7B1). It may be desirable to automatically generate these intervals while the user provides the min, max, and step size of the data. The main struggle, although not significant, was to come up with questions in the third task, particularly regarding the first topic: *what's in mind*. This topic was the least frequent in the Dear Data project as well. Providing questionnaire templates or allowing them to share with others would alleviate the struggle. Likewise, customizable drawing templates would also mitigate the perception of efforts necessary, as well as the concern about the lack of drawing skills.

4.8.2 Empowerment through Personalized Visuals

Our user study illuminated potential benefits of having control over the representation of personal data. One thing that recurrently stands out was the joyfulness of creating personalized visuals. The randomness of the outcome also seemed to contribute to it, as we saw some participants uttered their surprise when they see a visual response. This observation is also in line with previous research on the poetic use of unconventional encodings to create an element of surprise and stimulate reflection[189]. Another suggested benefit was being able to visualize the goals of data collection, as well as the impacts of the data on their lives. In particular, P14 articulated the use of our tool for visual decision making that can have an immediate impact on behavior change. Our study was qualitative and we believe it requires controlled experiments or a live deployment study to investigate the true potential of empowering people through personalized visuals more formally. The findings from our study provide initial hypotheses.

4.8.3 Beyond visuals and simple mappings

We focused on creating visuals in this chapter, but designing a multi-sensory experience using personal data is an exciting future direction. Similarly, one participant in the user study mentioned that sounds could often be much more emotional than visuals. It would be interesting to explore implications of using sounds [209] as well as tangible materials [188,104] in a personal context.

In addition, we believe there is an interesting avenue for exploring the concept of the parametric, decision-oriented visualization generation. Our current framework for visualization construction was based on simple linear mappings from an option to a visual in a single canvas. But there are fruitful opportunities to incorporate more advanced composition rules to the framework. For example, it can incorporate branches like a decision tree and allow for laying out the layers in multiple canvases that are further parameterized by the position and size.

4.9 Conclusion & Future Work

In this chapter, we analyzed the Dear Data project to gain insights on what is like to design custom visuals to represent personal data. We developed DataSelfie as a step toward empowering individuals to create personalized visuals to depict their data. We conducted a user study to understand the potential benefit of the empowerment. For future work, we plan to conduct a long-term deployment study to study its advantages and disadvantages in more depth for a self-tracking scenario.

The hardest part of design … is keeping features out.

Donald A. Norman

5

Authoring Data Stories using Comics

This chapter presents DataToon, a data storytelling tool for drawing data comics with pen + touch interaction. Comics are an entertaining and familiar medium for presenting compelling stories about data. DataToon seeks to incorporate elements of comics into the construction of data-driven stories. A single visualization is often not enough to break down the complexity of data and provide fluid guidance to an audience. DataToon supports the full lifecycle of the storytelling process from exploring data to discover insights and turning these insights into a narrative.

5.1 Motivation

Visualization is a pivotal component in data-driven storytelling, providing an audience with the means to understand patterns in data without requiring advanced statistical literacy [165]. One genre of data-driven storytelling [173] is the *data comic* [20], in which a narrative grounded in data is conveyed by leveraging the well-established visual language of comics [138]. Data comics integrate captions and annotations with visualization, suppressing the complexity of data by incrementally revealing aspects of the data across multiple panels, arranged thoughtfully on one or more pages [22,197].

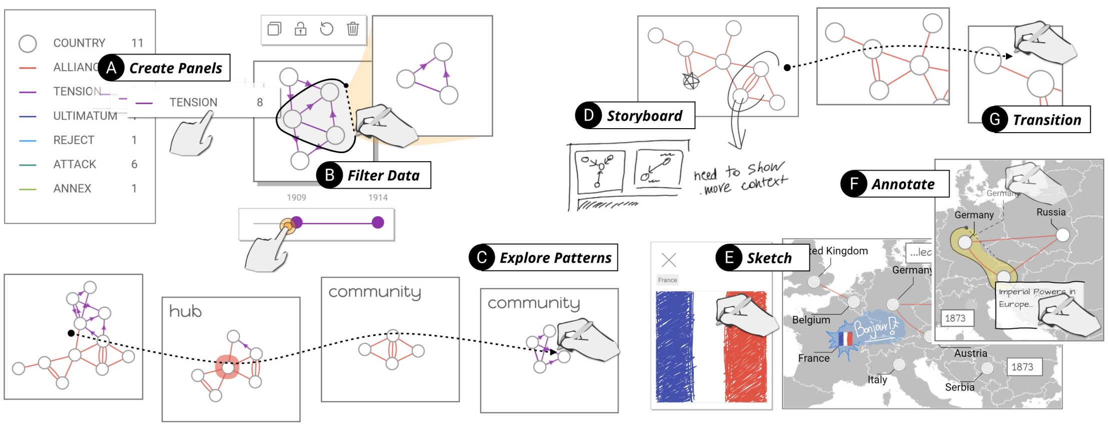


Figure 5.1: DataToon is a pen & touch environment for producing data comics. A storyteller can rapidly isolate aspects of their data via filtering and pattern detection, as well as assemble a rich narrative via annotation and automatic panel transitions.

A recently curated collection of manually-created data comics [23] demonstrates the richness of this genre and its applicability to telling stories about datasets of different natures. From a storytelling standpoint, one of the most challenging forms of data is a dynamic network. Dynamic networks appear in many contexts, from analyzing social networks to modeling neural connections in the brain. In addition to evolving in time, such networks may contain multiple types of nodes and links exhibiting different connectivity patterns. Due to this complexity, it is notoriously difficult to communicate insights about dynamic networks to a general audience with a single large visual representation. Since conventional comics often illustrate the dynamic nature of characters and the interactions that occur between them over time by identifying and sequencing salient moments, dynamic networks are ideally suited for a comic treatment [19].

However, producing a data comic is a difficult and laborious process, one that involves switching between visualization and graphic design tools [31], the former being ideal for generating accurate data representations, and the latter being ideal for stylizing visual elements and arrange panels in space. While several recent tools support the construction of visual data stories [112,171,207,160], they do not take advantage of the comic form as a storytelling medium. Thus authors have to resort to illustration and design tools such as Adobe Illustrator and Photoshop.

We contribute DataToon as a storytelling tool for producing data comics with a focus on dynamic networks. DataToon offers fluid storyboarding by blending analysis and presentation in a unified environment supported by pen and touch interactions. A storyteller can use DataToon to rapidly explore their data and generate visualization panels via interactive filtering and from recommendations of interesting data patterns, resulting in a visual story with custom annotations, automatic panel transitions, and layout templates.

The direct manipulation of panels and their data contents further facilitates the storytelling process. Natural touch interaction supports the iteration of story ideas by experimenting with different ways to compose panels and lay them out on a page. The use of a digital pen also allows storytellers to annotate panels with drawings and handwriting, or to draw custom glyphs for data entities. DataToon leverages the underlying data to eliminate the tedious duplication of actions necessary in conventional illustration software, such as propagating visual designs to other panels.

To demonstrate the expressivity of DataToon, we created a set of comics showing different rendering styles, panel layouts, visualization types, and narrative structures. Results from a reproduction study suggest that novice participants can successfully learn to use DataToon with minimal guidance to produce comics about dynamic networks. Insights about usability that led to improvements of

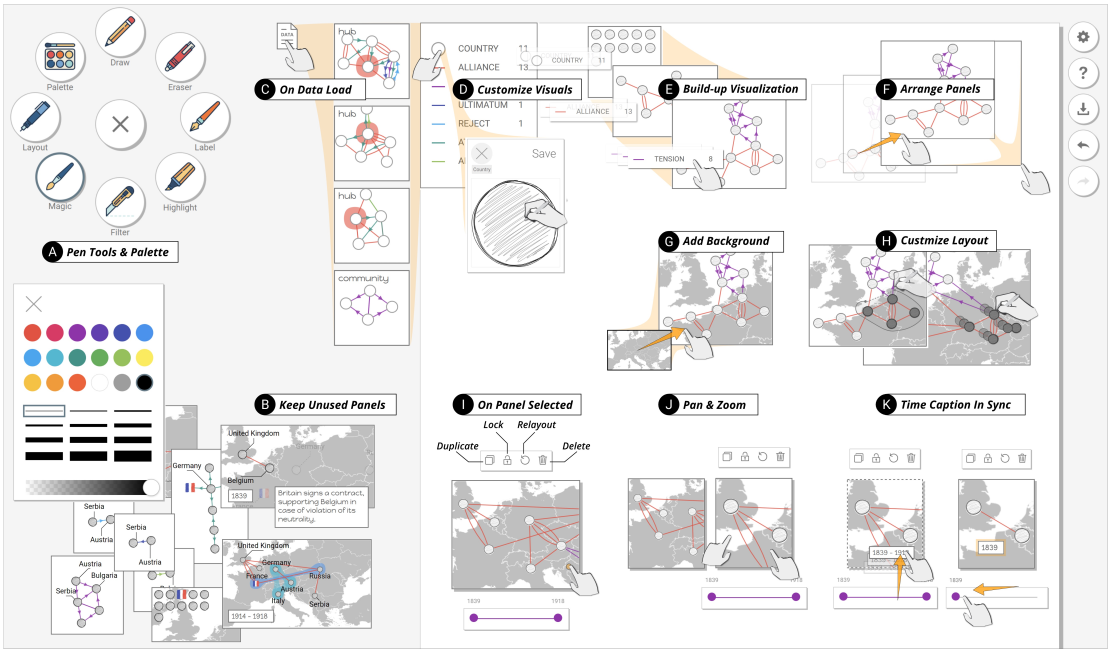


Figure 5.2: DataToon's interface: the pen can acquire different functions, such as labelling or filtering. The canvas area provides an infinite space for ideation and exploration, as well as a dedicated page area for presentation.

DataToon included difficulties in discovering features, the inconsistency of interactions, and the complexity of visualization contents.

5.2 Design Goals

We conceive DataToon to accomplish three main goals.

D1. Support the creation of data comics. Data comics have unique characteristics and components[20,22]. They expose the story via a juxtaposed and sequenced panels, each containing one (or a few) insight(s). Getting the reader to focus on each insight requires that the author carefully crafts the view of each panel, such as zooming in on an important part of a network. However, identifying different characters of interest (e.g., nodes) and tracking them across panels requires that the author gives each a custom visual style to maintain consistency[158]. DataToon aims to support the crafting of panels and the expressive design of characters by direct manipulation using pen and touch, while ensuring consistency with data-driven propagation.

D2. Enable data-driven design. Authoring tasks such as propagating visual designs across panels or creating transitions between panels are tedious and time-consuming. DataToon leverages the underlying data to automatically propagate visual designs of graphical elements and to generate textual labels and captions from data. DataToon also enables an author to generate transitional panels. For example, given two panels containing data at different times, DataToon can automatically create a series of panels representing the data at interim time points.

D3. Support exploration and authoring activities. Like any storytelling medium, the production of a comic is a creative and iterative process, often requiring the author to smoothly transition from reviewing the data and its patterns to styling them to craft a compelling story. DataToon facilitates the data exploration and review process by providing recommendations of interesting patterns in the data, such as a large cluster in the network, while enabling the rapid creation of visualizations using direct pen and touch manipulation. DataToon enables flexible workflows by providing a unique platform in which authors can review salient data aspects, rapidly generate and filter data visualizations, craft expressive visual design for data elements, compose a story by leveraging existing data comic templates, create a storyboard, or directly sequence and reorder panels.

5.3 Usage Scenario

To illustrate how DataToon accomplishes our goals and to describe key components of its interface, we describe the process taken by a hypothetical comic author named Aidan to create the comic in Figure 5.4, adapted from Bach et al. [19].

Aidan opens DataToon in his web browser on a pen + touch-enabled device. The pen-tools menu (Figure 5.2A) represents the set of instruments his digital pen can acquire. Aidan loads a dataset he previously created: a dynamic geographic and multi-faceted network containing countries and their alliances and oppositions over time. Dragging the file into the application instantiates both the legend panel and automatically generates a set of panels depicting notable structural patterns in the data (Figure 5.2C).

Aidan recalls that the evolution of alliances was interesting and he decides to explore these. He creates a new content panel by dragging the *Country* node type from the legend panel onto the canvas. Using the time slider for these panels (Figure 5.1B), he filters the data back and forth to explore how the alliances changed over time. He duplicates panels as he finds interesting times, jotting down notes on them using the pencil (Figure 5.1D).

Continuing this process of exploration, Aidan now has multiple panels with annotated insights. He proceeds to craft a story for his comic by organizing them on the canvas. He wants to start with an overview of the nations involved, so he drops an image file containing a map of Europe (Figure 5.2G) into a panel, adjusting the location of each node with the *Layout* pen (Figure 5.2H). He extracts country names from the data to place them on the map with the *Label* pen.

After establishing the context of the story, Aidan wants to show the evolution of alliances in Europe. He reuses panels that he created earlier, transforming his rapid handwritten annotations into visual clusters around nodes, captions and labels. While he pans and zooms his first panel to emphasize two nodes (Figure 5.2J), he realizes that the difference between this panel and the first one may be too great and that his audience may fail to see the connection. He acquires the *Magic* pen to interpolate between these panels and generate intermediary ones (Figure 5.1G).

He generates a time caption for the last panel by dragging the time label from the slider (Figure 5.2K, left). Duplicating this panel (Figure 5.2I) and adjusting the time automatically updates the caption (Figure 5.2K, right). Uncertain about what to show next, Aidan uses the *Magic* pen to trigger suggested panels with interesting patterns (Figure 5.1C).

As his comic nears completion, he customizes the node and link representations. For instance, he

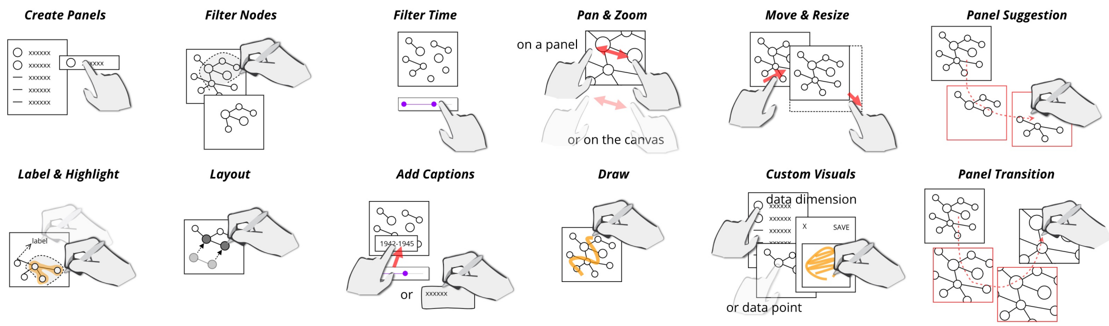


Figure 5.3: An overview of the pen + touch interactions supported by DataToon.

draws a custom sketchy representation for all nodes (Figure 5.2D). To emphasize France among all countries, he paints its flag (Figure 5.1E). This custom node representation is automatically propagated to all panels. Satisfied with his comic, he exports the comic as an image that he can share with his students.

5.4 DataToon

We designed DataToon for a broad range of people who wish to craft data comics that communicate insights about their data. This may include graphic designers without programming expertise, data analysts without design expertise tasked with communicating their findings, or educators seeking new ways to engage their students.

5.4.1 Data Abstraction

As mentioned in the introduction, we chose to focus on dynamic network data since it is poorly supported by existing communicative visualization and storytelling tools, and because of its inherent parallel to the dynamic interactions between characters appearing in comics. In particular, DataToon supports multivariate dynamic network data, which consists of nodes and links and their attributes. A node can have a label and a type, as well start and end dates. A link must have source and target nodes and may have a direction and a weight along with same set of attributes describing nodes. Given this criteria, DataToon also supports static networks, where neither nodes nor links have associated start or end times. Note that DataToon is primarily a storytelling tool, one suitable for communicating different aspects of dynamic network data; we do not address scalability issues and analysis tasks related to very large networks in this paper.

5.4.2 Story Abstraction

Like a comic book, a data comic consists of pages (Figure 5.4), in which each page can contain multiple panels. A panel is the essential building block of a narrative structure, which can in turn contain visualization, images, text captions, and annotation. The spatial arrangement of panels having varying size and content generates a unique narrative flow, enabling a nonlinear reading experience unlike linear sequences produced by other storytelling tools.

5.4.3 Interaction Design

DataToon is comprised of a canvas for storyboarding; content and legend panels for presenting visual representations of data; a set of pen tools for content creation and manipulation; and a contextual canvas for sketching custom visuals.

Figure 5.3 summarizes our pen + touch interactions. In general, our interaction design choices reflect the mantra: *the pen writes, touch manipulates*[86]. However, since individual nodes and links within panels are often too small to manipulate with a finger, the pen is also used to manipulate visual elements in some circumstances, as described below. Throughout the interface, we chose to visually expose interactive controls rather than rely on implicit multi-touch gestures that are difficult to discover and remember. Note that we describe our final system, which improves upon the version used in our reproduction study described below.

Interacting with the canvas. DataToon provides an infinite canvas to support flexible authoring and rapid ideation, transitioning from data exploration to authoring activities (D3). The author can navigate the canvas via pan and zoom gestures. Meanwhile, using the pen, the author can draw or write anywhere on the canvas, either to annotate content or to add storyboarding notes and ideas. The author can create an empty panel by simply drawing a rectangle, to be filled later with content, or create panels from data. Panels can be freely arranged and resized with touch interactions, leading to different layouts at any point in the authoring process.

Interacting with the legend panel. A legend panel is created when the author drags a dataset file onto the canvas. This panel provides an overview of the dynamic network, displaying a list of node types and link types along with the cardinality of each type. This legend also serves as an interface for creating content panels. Dragging a node or link type from the legend onto the canvas creates a new content panel displaying a filtered visualization of the data. Dragging a node type automatically creates a unit chart of all nodes in the data matching this type. Dragging a link type automatically creates a force-directed node-link diagram of nodes connected by this link type. Note that links can convey both weight and direction via line thickness and arrows, should these optional attributes be provided.

Node and link types can also be dragged from the legend panel to an existing content panel, whereupon its contents are updated to reflect the additional type (Figure 5.2E) and its layout recomputed. For instance, dragging a link type to a content panel containing a unit chart will convert the chart into a node-link diagram. Similarly, adding a link type to a panel containing a node-link diagram will generate multiple link types (see Figure 5.2).

Interacting with content panels. A content panel may contain visualizations, text, annotations, a background image, or some combination thereof (D1). Panels can be nested: drawing a rectangle inside a panel creates a child panel, which is useful for text captions or inset visualizations. It is possible to duplicate an existing panel, copying all of the content of the source panel to a new panel (Figure 5.2I). This interaction is particularly useful for progressively building a narrative using the previous panel as a starting point.

Tapping on a panel containing a visualization selects it and enables panning and zooming within the panel. This also reveals a time slider for the panel, which applies additional temporal filtering to nodes and links displayed within the panel. Dragging this slider onto the panel creates a nested time caption panel (D2), which remains updates as the user changes the time slider (Figure 5.2K).

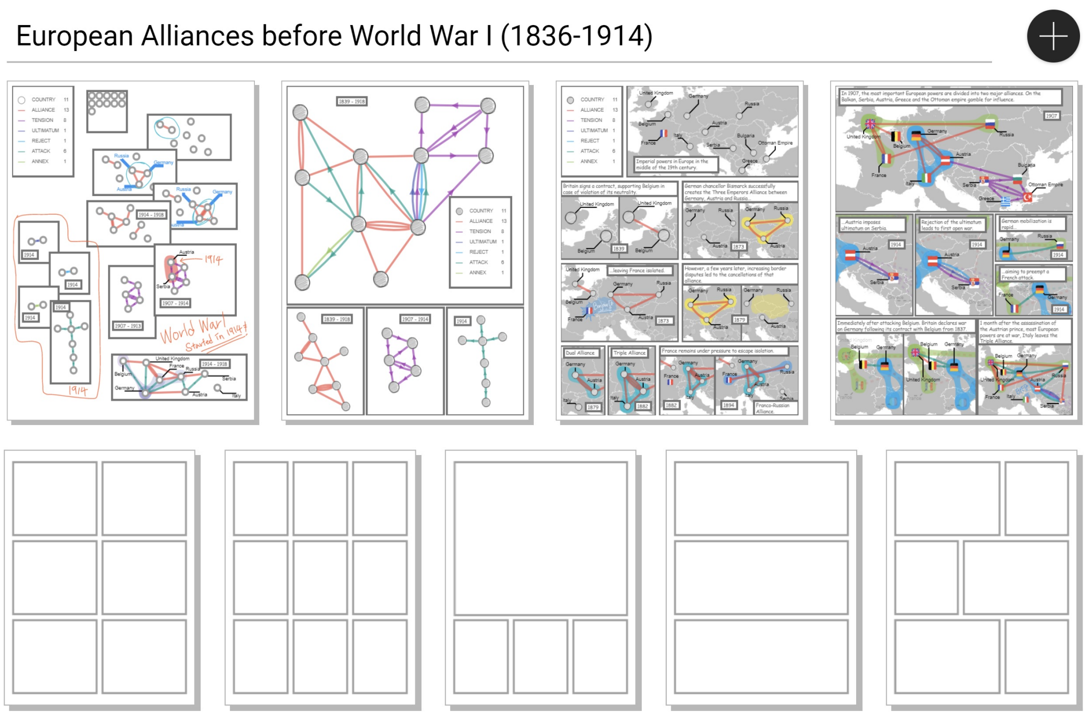


Figure 5.4: DataToon uses a comic book metaphor to allow authors to create multiple pages of data comics using more than one dataset. Each page can be created with a a predefined layout such as the ones shown in the second row.

Acquiring different pen modes. Content editing occurs through the use of the following pen modes:

Pencil mode is the default pen mode and allows for freeform inking on the canvas. If the author draws atop a content panel, the ink belongs to the panel and moves along with it when that content panel is manipulated.

Label mode generates a text label for nodes and links and allows for adjusting label positions (Figure 5.1F). Label and leader lines move when the element is moved.

Highlight mode allows the author to lasso a set of nodes to highlight them in a colored group (Figure 5.1F). As with labels, highlights also adjust when elements are moved.

Filter mode allows the author to lasso a set of nodes to filter them from a panel or to create a new panel with these nodes (Figure 5.1B). Filtering can de-clutter panels and help focus on different subsets of the network.

Layout mode leverages the high-precision of the digtial pen to adjust the positions of nodes and labels (Figure 5.2H).

Magic mode offers a way to automatically generate content panels (Figure 5.1C, D), described in more detail below.

Eraser mode deletes any item on the canvas including panels, ink, annotations, and highlights. This mode can also be invoked via the pen's eraser button, should it have one.

Palette mode allows the author to choose a different color, line thickness, and fill opacity, which will be applied to subsequent pencil strokes, annotations, and highlights.

Drawing custom node and link representations. A secondary canvas is invoked when the author taps on the iconic representation of a node or link type in the legend (Figure 5.2D). Within this canvas, the author can draw a new iconic representation for the selected node or link type, and this custom representation is immediately propagated across the comic for consistency (D2). The author can also invoke this canvas to change the representation of a single node or link, by tapping it with the pen while in pencil mode ().

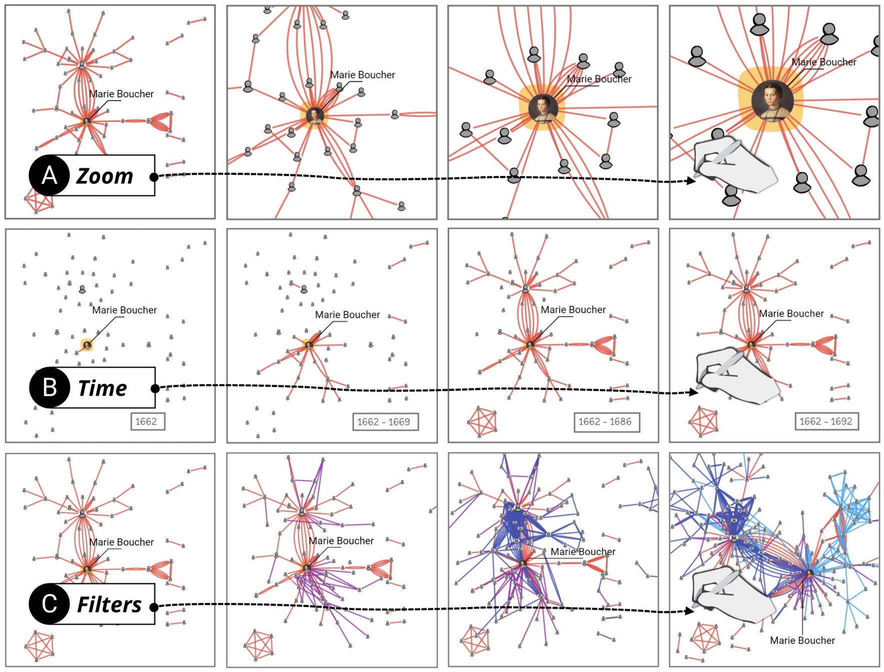


Figure 5.5: Automatic transitions between panels involve interpolating differences between panels, incorporating zoom levels, time ranges, filters, and combinations thereof.

5.4.4 Facilitating Ideation

DataToon provides several ways to scaffold and accelerate the data comics creation process (D1):

1. Multiple pages and layout templates facilitate the creation of multiple iterations of comics (Figure 5.4). Pages can load different datasets or contain different notes. Templates are a set of panel layouts commonly used in comics[22] such as grid, overview+detail, parallel, and staggered. When selected, the new page is auto-populated with empty content panels specified by the template. Authors can simply drag data on top of them to fill them.

2. Automatic transitions. DataToon creates intermediary panels by interpolating the difference between the two panels, taking into account their respective panel sizes, data filters, and zoom states (Figure 5.5). These transitions may also incorporate a temporal progression between two time filter states and the addition or removal of nodes and links[22]. It does not, however, attempt to interpolate annotations between two panels.

As an example, if a source panel and a target panel have zoom factors of 1.0 and 1.8 respectively, intermediary panels will have interpolated zoom factors between 1.0 and 1.8. In addition, if the source and target panels have different sizes and display data at different time periods, DataToon will interpolate over these dimensions as well, producing intermediary panels that would gradually increase in size and depict a progression over time intervals.

DataToon places intermediary panels along the path drawn by the author. Greater distances between panels results in more intermediary panels along the path. It discretizes the path and conducts a linear search for panel positions, while ensuring no overlap between intermediary panels.

3. **Panel suggestions** for new panels are shown in two contexts: either when loading a new dataset or when drawing a line out of a panel using the Magic pen. In the first case, it attempts to detect interesting subnetwork patterns in the whole dataset and places suggested panels at the left side of the canvas. This helps authors become familiar with the dataset and serves to kickstart the story design process. In the second case, it takes into account the current state of the panel and detects patterns within the panel to generate suggestions. It thereby enables authors to discover potential compelling directions for their story. Similar to transitions, these suggested panels are placed along the path the author draws, while preventing overlap between them.

To detect patterns in the network data, DataToon relies on a pattern detection engine. The engine accepts any network data including highlighted nodes as input and returns a list of detected patterns. It currently looks for four structural patterns including articulation points (or bridges), cliques, hubs, and communities (Figure 5.6). Once patterns are found, it heuristically prunes the results such as by removing overlapping patterns (e.g., a bridge and a hub can often show a completely identical structure), as well as cliques with less than four nodes. Finally, it ranks the patterns based on how much

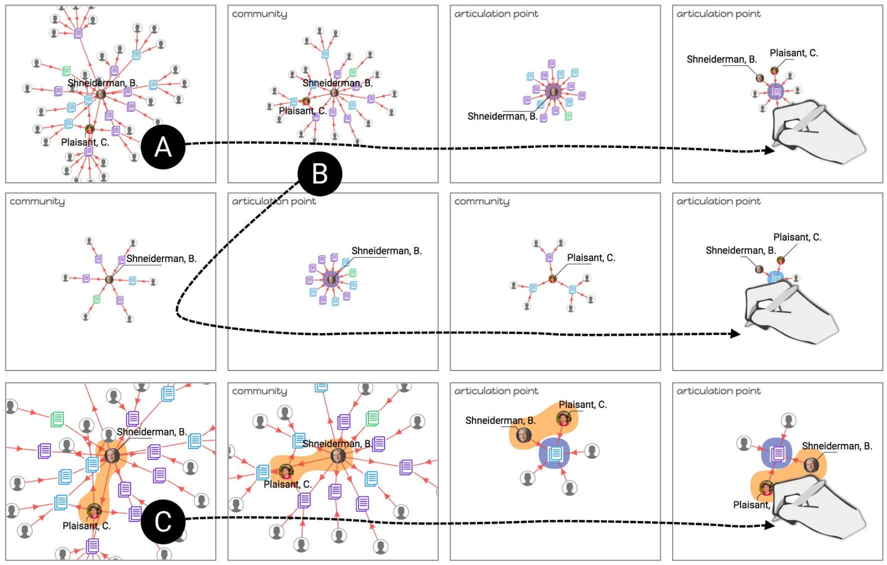


Figure 5.6: Automatic panel suggestions depicts structural patterns: communities, hubs, articulation points, and cliques. The author can trigger suggestions from existing panels (A) and suggested panels (B). Patterns are ranked based on network coverage and inclusion of highlighted nodes (C).

each pattern overlaps with the source data. This is to promote closure between the source panel and the suggested panel. Finally, the rankings give a high priority to patterns that contain highlighted nodes (Figure 5.6 C).

5.4.5 Implementation Details

DataToon is a browser-based application written in Javascript. It uses React.js for building user interface components and Redux.js for application state management. It uses WebCoLa [61] to generate the layout of the node-link diagram and a Javascript implementation [116] of Bubble Sets [52] to highlight a group of nodes as a cluster. DataToon consists of a front-end interface without a back-end server, though one can be attached if needed; currently, DataToon makes use of localStorage and indexedDB in HTML5 to persist the application state. The panel recommendation engine is implemented in Python and uses NetworkX to detect patterns [77].

5.5 Evaluation

Our evaluation methodology is representative of other recent evaluations of visualization authoring systems [162], in that we demonstrate the *expressiveness* of DataToon with an example gallery (Figure 5.7) and also evaluate its *learnability* and *usability* via a reproduction study.

5.5.1 Example Gallery

xFigure 5.7 shows example comics varying in their comic style, including diverse rendering styles (abstract, sketchy, realistic), panel layouts (inset, overlapping, staggered) [22], visualizations (unit charts or pictographs, maps, set visualizations, node-link diagram), and narrative structures (overview+detail, nonlinear-temporal, cut-out, build-up [22]). The gallery also exemplifies a diversity of datasets, including multivariate and temporal social networks and co-authorship networks.

5.5.2 Reproduction Study

To evaluate whether people can learn how to use DataToon to create comics from data, we conducted a qualitative reproduction study, in which participants are asked to reproduce them completed examples with DataToon. This type of study is particularly common in the evaluation of visualization authoring tools [162], having been used to evaluate Lyra [171], ChartAccent [159], DataInk [207], Data-Driven Guides [112], Data Illustrator [129], and Charticulator [161].

Participants. We recruited eight participants from a large software company in the United States. Half of the participants were graphic designers with limited data literacy (P1-P4: 3M, 1F; ages 30-−50, avg: 44), while the other half were data analysts with minimal experience in design tools (P5-P8: 2M, 2F; ages 31−42, avg: 37).

Apparatus. Participants used a earlier version of DataToon on a Microsoft Surface Studio with a 28-

inch screen at 4500×3000px (192 PPI), a device that enables simultaneous pen and multi-touch input.

Procedure and tasks. Beginning with a demographic survey, each study session lasted ~90 minutes, with two participants finishing in ~60 minutes, and one taking ~120 minutes. We asked them to reproduce two comics: 1) the first with guidance from us using the comic about *World War I alliances* (Figure 5.2) and 2) the second without any assistance using the comic inspired by Fathom's *Scaled in Miles* project[65] about the evolving instrumentation on Miles Davis' records (Figure 5.7-left). The first task served as a training session and lasted 30 to 40 minutes, which included a 15-minute tutorial, while the second task lasted between 30 and 50 minutes. The study ended with three Likert-style questions about ease of use & learning, and enjoyment, along with a semi-structured interview about their experience.

5.5.3 Observations

All participants successfully completed both comics, while we discovered several usability insights into the usability of DataToon. We describe our observations below.

Learning to interact with both pen and touch. All of the participants appeared to grasp DataToon's interaction design by the end of the study, except for P2, who had no prior experience with pen + touch devices. It took a long time (approx. 10 min) for P2 to complete the first task and the effort spent to learn the interactions are reflected in their low ease of learning (3/7) and use (4/7) scores. P4, P5, P6, and P7 also repeatedly appeared to be frustrated when attempting to use pen and finger interchangeably, with P7 stating *"I kept using my hand instead of the pen"*.

Participants bimanual pen and touch interaction to be engaging, with P8 remarking on the simplicity of the interactions: *"I love the power of just dragging* [shows fingers] *and creating* [shows the pen]*"*. P4 spoke about the empowering experience of bimanual pen and touch input for content creation, making DataToon *"unique"* and *"fun"* compared to other tools: *"I feel like a surgeon because I got precise and used both of my hands, not something I do ever. It's pretty cool!"*.

A focused tool set for design exploration. The graphic design participants all expressed that one notable strength of DataToon was a *"focused tool set"* (P1), its interface *"streamlining the set of tools"* (P4) compared to existing illustration tools. We observed that our interface enabled alternative workflows to achieve the same result, reflecting what Ren et al. refer to as the *flexibility* of a visualization authoring tool[162]. For example, several participants began with multiple panels, adjusted the content of each panel before customizing each of them in turn. Others would create and modify one panel until it was polished, only then duplicating to instantiate the next panel.

Participants' difficulties often related to feature discoverability, as not every pen mode was visually shown in the interface. For example, in the version of DataToon used in the study, the pen button was used to activate the highlight pen. P8 commented that *"Minimalism is in, it looks just like a simple drawing app, but then it can be intimidating because how do you achieve all of this?* [pointing to print out of the data comics] *I was nervous"*. Similarly, P1 commented that the principle of dragging and dropping elements into panels was violated in the case of time captions, which required a double tap, making it challenging to discover.

Closing the gap between analysis and storytelling. Participants appreciated the ability to discover patterns during the story authoring process, suggestive of a possible advantage over visualization authoring tools that are disjoint from data analysis tools. We observed that data analysts tended to explore the data before constructing on their comics. For example, P5 started by creating many panels (one per node type) and by commenting on the structural patterns in the data. P8 used a different strategy, adding each node type to a single panel in succession, where each node type was a different instruments featured on Miles Davis records; upon doing so, P8 stated that *"now I am beginning to see the relationships between instruments [...] I am going to move things around so I can understand my* *data"*. Finally, some participants noted the necessity of additional data abstraction. For example, P3, looking at a particularly dense node-link diagram, said *"I wish there were a way to untangle that because that is a super full graph"*, suggestive of a need for capabilities that aggregate nodes and links.

5.5.4 Lessons Learned from the Reproduction Study

The results of the study illuminated a set of usability insights regarding the difficulty of discovering features, the inconsistency of pen and touch interactions, and the complexity of visualization contents. These insights led to several improvements to the design of DataToon.

To address the feature discoverability issue, we ensured that all pen modes are visible (Figure 5.2A) without cluttering the interface and degrading the authoring experience. In addition, after observing participants repeatably attempting to use fingers where we enforced use of the pen, which initially included panning and zooming panel content, we opted to accommodate more touch interaction, reflecting *the pen writes, touch manipulates* mantra[86]. We also replaced the double-tap gesture for creating time captions with a consistent drag gesture. Finally, to handle the visual complexity issue, we added the ability to filter nodes of interest from a panel, as well as the panel suggestion functionality for assisting with exploring complex data.

5.6 Discussion and Future Work

We now reflect on broader issues and opportunities that arose during the development and evaluation of DataToon.

Generalization to other data and visualization types. The design of DataToon (See Figure 5.3) is mostly data-type-agnostic and generalizable beyond network data, such as gestures, panel manipulation, annotations, time sliders, highlighting and removing data elements, etc. Accommodating other data types (e.g., tabular data), visualizations (e.g., bar charts), and specific components (e.g., axes and

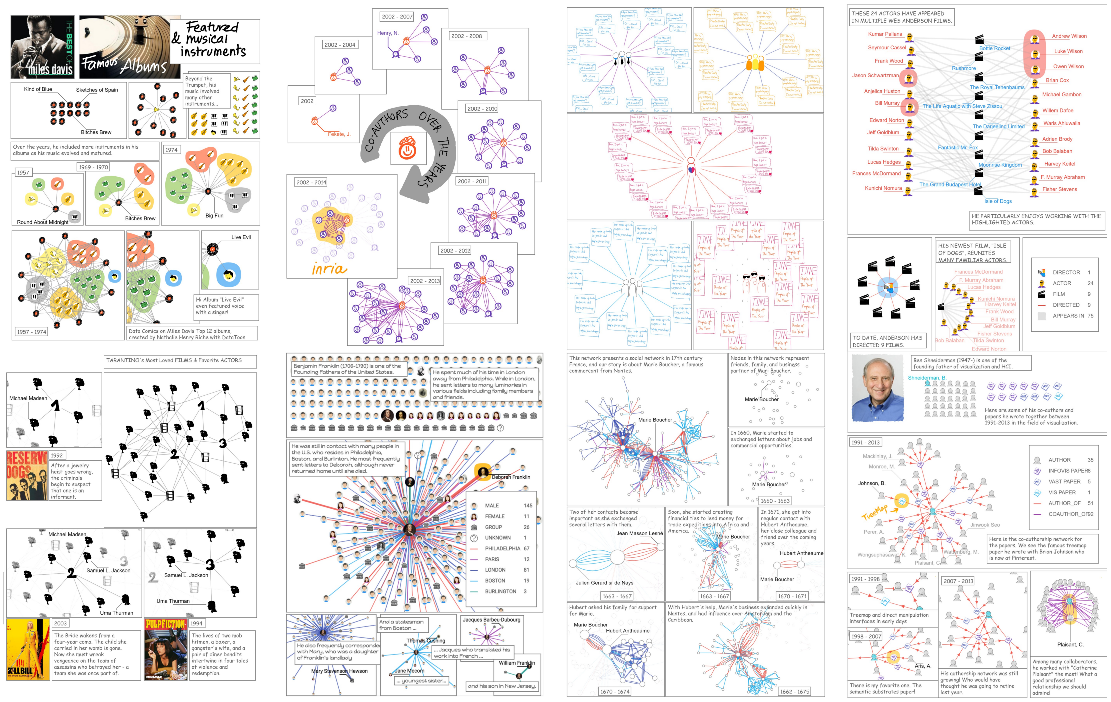


Figure 5.7: A gallery of eight data comics created with DataToon using different comic styles and datasets.

scales) is an interesting research direction. The overarching research question in this space would be how to enable fast and fluid creation and manipulation of such panels, as visualizations can involve complex grouping and filtering operations on data. However, core "comic" features such as automatic propagation of style, automatic generation of transitions and panel recommendation would require minimum redesign.

Pen + touch interaction for data-driven storytelling. Pen + touch interaction was seen as engaging and flexible by our study participants, advantages that may prove beneficial beyond data comics to other data-driven storytelling experiences. While novices require time to acclimate to pen + touch interfaces, we observed that after an initial learning phase, this input modality stimulates creativity and encourages experimentation. However, to promote learning and discoverability, we must design appropriate visual cues and affordances that remind users of their capabilities. In our study, we observed that people initially want to use pen and touch interchangeably to accomplish a single action. This observation warrants further investigation and a revisitation of the *pen writes, touch manipulates* mantra[86].

Beyond traditional comics. DataToon exports pages as static images, like traditional comic books. While being respectful of this tradition, creating dynamic and interactive data comics is an interesting research direction. For example, a "presentation mode" might allow for presenters or viewers to touch parts of the comic and reveal content on demand, or add annotations as part of an active reading process[195]. Alternatively, DataToon could export comics as websites that invite viewer interaction, potentially by integrating techniques such as brushing and linking across panels.

Toward higher-level narrative design support. Our design considerations emphasized use of the structural elements of comics such as panels and captions to convey a data story. However, producing an engaging story still depends on the contents of the data and the creativity of the author. DataToon does not explicitly incorporate higher-level narrative design patterns[22,21] into its interface.

The automatic transition, suggestion, and layout template features are a step toward narrative design guidance, but there are further opportunities to improve . For example, we might auto-populate an entire template with visualization content as a way to seed a story, though we must take care to not absolve authors of their creative agency. Similarly, generating panel transitions that precisely match an author's narrative intent is challenging, but such transitions can be used as a way discover new narrative directions. Also, being able to quickly evaluate the overall narrative structure will greatly aid the iterative process of crating a story[110].

5.7 Conclusion

We contributed DataToon, an interactive system for producing comics about dynamic networks. It leverages the form of comics to construct a narrative structure and offers a flexible pen + touch authoring interface for content creation and manipulation. DataToon provides automatic transitions and panel recommendations for narrative ideation and accelerated storyboarding. We plan to extend our evaluation to study the authoring process in a more longitudinal *free-form study* [162], focusing on comprehensive evaluation metrics for visualization authoring tools [14].

The greatest value of a picture is when it forces us to notice what we never expected to see.

John Tukey

6
Visualizing Story Structures

This chapter describes story curves, a visualization technique for showing the narrative structure of a story. Crafting a story is a creative, open-ended process that involves trial and error. Being able to quickly grasp the overall story structure can accelerate this iterative process. story curves allows a storyteller to rapidly evaluate the temporal dimension of the narrative and experiment with alternate structures through Story Explorer that embeds story curves with story contents.

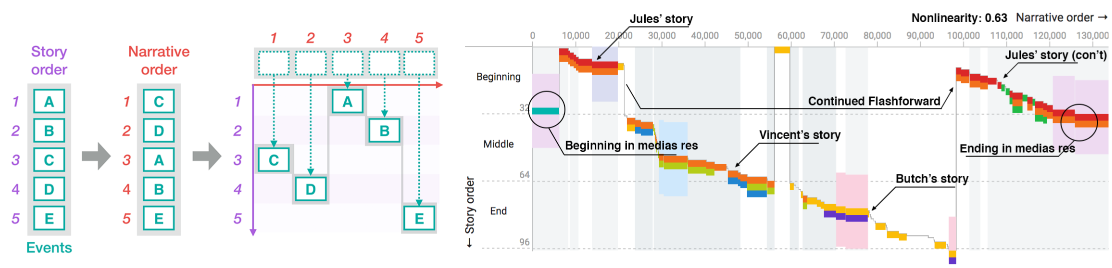


Figure 6.1: A schematic diagram showing how to construct a story curve from a sequence of events in story and narrative order (left). An example of a story curve of the movie *Pulp Fiction* (right) showing characters (colored segments), location (colored bands), and day-time (gray backdrop). A nonlinearity index is calculated based on the degree of deviation of narrative order from actual story order.

6.1 Motivation

A *narrative* specifies the way in which events in a story are told[70]. A *nonlinear* narrative is a narration technique portraying events in a story out of chronological order, such that the relationship among the events does not follow the original causality sequence. For example, a narrative can withhold information to maintain a sense of mystery, to keep tension high, and to keep the audience interested. Eventually, the narrative can flash back to the beginning of the story, releasing the tension. Such nonlinear narrative techniques are widely used in various types of storytelling genres, including literature, theater, movies, graphic novels, as well as hypertexts and other computer-mediated genres such as video games[62,47,11,73].

Understanding the wealth of patterns in narratives as well as their respective effects has been an ongoing effort in the humanities and the domains associated with the production of narratives[156,26,70]. For example, the French literary theorist, Gérard Genette described a recurrent set of nonlinear narrative patterns including flashbacks, flashforwards, and retrograde narratives[70]. Although there have been many computational approaches to analyze nonlinear patterns in stories[137,12,89,56], there are relatively few studies that focus on the temporal aspect of narratives. One explanation may be that explicit data about the chronological order of events is typically not available and difficult to infer from the

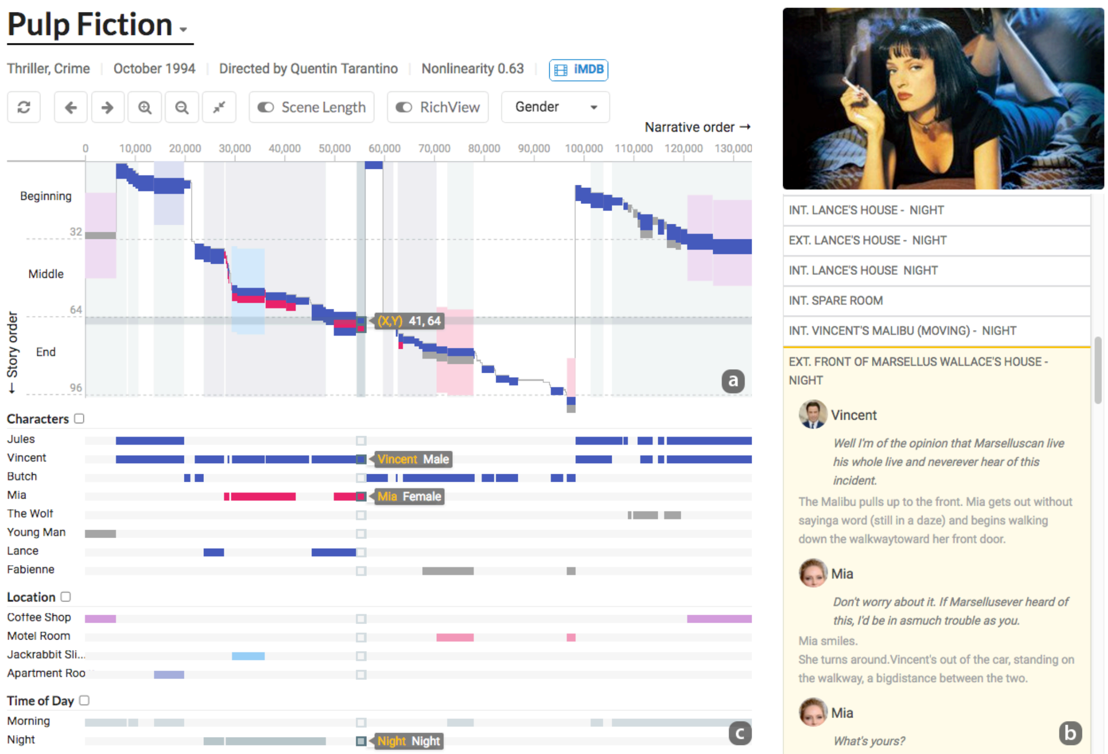


Figure 6.2: Overview of Story Explorer with three embedded views: (a) story curve view, (b) script view, and (c) metadata view. The story curve succinctly summarizes the nonlinear narrative of *Pulp Fiction* with additional metadata displayed along the story curve.

final narratives in film, theater, or video games[70,47]. The complexity of temporal disorientation makes it not only hard to write a nonlinear narrative but also difficult to understand it[non]. In addition, there are no computational tools that can ease exploration and communication of intricate nonlinear narrative patterns.

In this paper, we present *story curves* (Figure 6.1), a visualization technique to reveal nonlinear narrative patterns. We also describe Story Explorer (Figure 6.2), a tool that allows users to curate the chronological order of scenes in a movie script and explore the nonlinear narrative of the movie using story curves. Story Explorer automatically parses movie scripts and extracts essential story elements such as scenes (the story's events taking place in a specific time and location) and characters as well as their semantic metadata such as dialog sentiment and scene settings. It displays the movie script text (Figure 6.2b) alongside story curves with a set of visualizations of complementary information such as characters, times, and places (Figure 6.2c).

Story curves are the core of Story Explorer and visualize events (scenes) as points in a 2-dimensional plot according to their order in the narrative (horizontally, left-to-right) and their chronological order in the story (vertically, top-down). As users rearrange scenes into their chronological order, nonlinear narrative patterns become evident through the meandering shape of the story curve that connects scenes in both narrative and story order. A similar visualization has been used by the New York Times to visualize the narration in movie trailers[43]. Our story curves are the first scientific investigation and systematic exploration of this visualization technique. In addition, we encode additional story information such as characters, places, and periods of the day. Characters are represented using different colored curves, which communicate the number of characters in a scene through the thickness of the curve segment. Places are encoded as bands surrounding character curves, while day times are represented using vertical backdrops in the background.

To demonstrate the usefulness of the visualization technique, we generated story curves for 10 popular nonlinear movies. We show that the story curves do not only reveal nonlinear narrative patterns mentioned in Genette's framework[70] but also show several novel patterns that have not been discussed in the literature before. In order to evaluate the readability and learnability of story curves, we conducted a user study asking participants to answer 20 questions of a pattern reading task (e.g., how many flashbacks are there in a story curve?). Some participants had difficulty in reading both story and narrative order at the same time. Overall, however, participants showed good performance, correctly answering 80% of the questions on average. We highlight potential use cases of Story Explorer in screenplay writing and analysis, education, and film production based on informal discussions with experts including professional writers and a literary scholar.

6.2 Background

The terms *narrative* and *story* are often used interchangeably in informal settings, though the two are very different. A *story* is content (what is told) consisting of events (actions, happenings) and existents (characters and settings), while *narrative* is the expression (how it is told) concerning how the content is presented to readers (narrative voices, styles, plots)[47]. Strictly speaking, we only see the story through narrative and thus it is the narrative that determines our perception of story. As both story and narration unfolding are time-dependent, events in a story "happen", while the narrative is narrated. Every narration encompasses two temporal sequences: *Story time* is the chronological time in which the events happen (e.g., the year 1600, day 5 of the story, Monday, etc.), while *narrative time* is the time of the events being told (e.g. minute 4:21 of the movie, the beginning, middle, or end of a video game, etc.)[70].

Moreover, there are categories of narrative temporality: *order*, *duration*, and *frequency*[70]. Order describes the relation between the chronological sequence of the events of the story and the sequence within which the events are narrated to the audience. Duration compares the time an event spans in the story with the time it takes for it to be described in the narrative. And frequency contrasts the number of times an event occurs in the story to the number of times it is recounted in the narrative[164]. In this work, we are only concerned with order, i.e., the ordering of events in time, which is one of the most fundamental characteristics of any story[155,147].

In a linear narrative, the order of events in narrative time is the same as in story time. Nonlinear narrative is a storytelling device that depicts events out of chronological order, often employed for the purpose of increasing the suspense of a story[84,55]. Unlike a linear narrative, a nonlinear narrative does not have to follow direct causality patterns[54]. There may be more than one narrative describing the same story, such as leaving some events out to emphasize particular perspectives or rearranging events to create a sense of mystery[26].

According to Genette's typology[70], there are seven categories of the relationships between the temporal order of the events that are being told (story order) and the pseudo-temporal order of the narrative (narrative order). Montfort[147] concisely describes these patterns, which we summarize as follows:

Chronicle: Events are narrated in chronological order, i.e., there is temporal agreement in the order of the events between story and narrative. A unique order may not be specified as some events can happen simultaneously; they may be arranged in any order, relative to each other. Most movies belong to this category (e.g., natural disaster or folklore movies).

Retrograde: Events are narrated in reverse chronological order. For example, colored scenes in Christopher Nolan's movie *Memento* are portrayed backward, while black-and-white scenes are in the original order. Another historical example is Iliad, an ancient Greek epic poem, that begins in the middle of the Trojan War.

Zigzag: Events from a period are interleaved with those from another period as they are narrated in order, e.g., a narrative alternating between the past and present. The events that are paired must be semantically related, thus resulting in a temporal coordination similar to Syllepsis discussed later. For example, a past event is retrospective of a present event, and in the movie *Memento* chronological scenes are interleaved with reverse scenes.

Analepsis: Events are narrated that took place earlier than what is being narrated. It is more commonly referred as *flashbacks* that are used to recount events that occurred in the past to fill in crucial backstory[109] For instance, flashbacks are a major part of the TV show *Lost*, portraying what happened in the life of the main characters before they were stranded on the island.

Prolepsis: Events are narrated that take place later than what is being narrated. It is more commonly referred as *flashforwards* that are used to allude to events projected to occur in the future. For example, the film *Arrival* extensively uses prolepsis to show events that occur in the future.

Syllepsis: Events are grouped based on some criteria (e.g., spatial, temporal, thematic kinship). Thematic groupings are often used in the classical episodic novel where multiple stories are inserted and justified by analogy or contrast. Similar groupings are also found in films like *Pulp Fiction* and *Love Actually* that use multiple plotlines.

Achrony: Events are randomly ordered; thus the relationship between the order in which events are narrated and the order in which they occur is difficult or impossible to establish, possibly due to lack of temporal information available from the narrative.

Others have gone on to further extend this taxonomy to address the temporal irony that prevails in postmodern narratives such as time forks (e.g., *Inception*) and time loops (e.g., *Interstellar*)[84,163].

6.3 Design Goals

The overarching goal of this work is to develop a technique that facilitates the exploration, communication, and discovery of nonlinear narrative patterns.

Our work was inspired by Genette's analysis on nonlinear narrative patterns which was purely based on close reading of short text[70]. For instance, Genette uses textual symbols to represent narrative order and story order, such as "A2[B1]C2[D1(E2)F1(G2)H1]I2"(A-I: narrated events, 1: present, 2:past). Brackets and parentheses indicate flashbacks and flashforwards respectively. While close reading (reading an actual part of a text), is a crucial part in literary research, we wanted to design a computer interface for distant reading that can reveal and communicate patterns visually to scale beyond a few sequence[Jänicke et al.]. In order to design such a technique, we defined the following three design goals:

G1: Show events in both narrative and story order. Ordering of events is a key element for studying the temporal nonlinearity of narratives as highlighted by Genette[70]. Frequency and duration of events can only be considered after the order of the events is taken into account. Showing how narrated events are arranged in original story order demonstrates how the order of events has been dramatically manipulated into an engaging presentation of the story in the narrative.

G2: Present story metadata to reveal the semantic structure of narrative. The ordering of events in time alone does not provide a clear picture of what is told and how it is narrated. The raw material of a story, including not only actions but also actors, time, and location, is important for understanding the semantic structure of narrative[25]. Additional semantic metadata, such as character emotions (frown, fear, smile, etc) and traits (ambitious, charismatic, etc), could be also extremely useful. However, automatically extracting such high-level semantics from textual scripts is a challenging natural language processing problem and not the goal of our work.

G3: Let users access and read actual narrative texts. A traditional screenplay analysis involves closely reading scripts, similar to how humanities scholars analyze literary texts or text passages[Jänicke et al.]. While a high-level, abstract visual summary of the narrative structure can be useful in discovering global patterns, it is still important to show the raw text. A close reading of the script can enable deeper analysis of the context of global patterns[Jänicke et al.,139], e.g., reading actual conversations between two characters whose co-occurrence is prominent in the visual summary.

Based on these design goals, we infer four main visualization tasks. These tasks involve not only drilling down into a single character's progression across scenes but also making sense of the overall nonlinear structure.

T1: Identify nonlinear narrative patterns, i.e., identify how the order of the scenes in a film unfolds in story order. An example task is to find a flashback scene or a retrograde pattern [G1].

T2: Identify and compare character occurrences across different scenes in both story and narrative order [G1, G2].

T3: Compare the character occurrences in different scene settings including location and time [G1, G2].

T4: Read character dialogs and actions in a specific scene and identify the position of the scene in the global context [G1, G3].

These tasks are mostly in line with existing story visualizations outlined in Section subsection 2.2.2, except that we focus on the temporal nonlinearity of narratives. To support these tasks, we developed

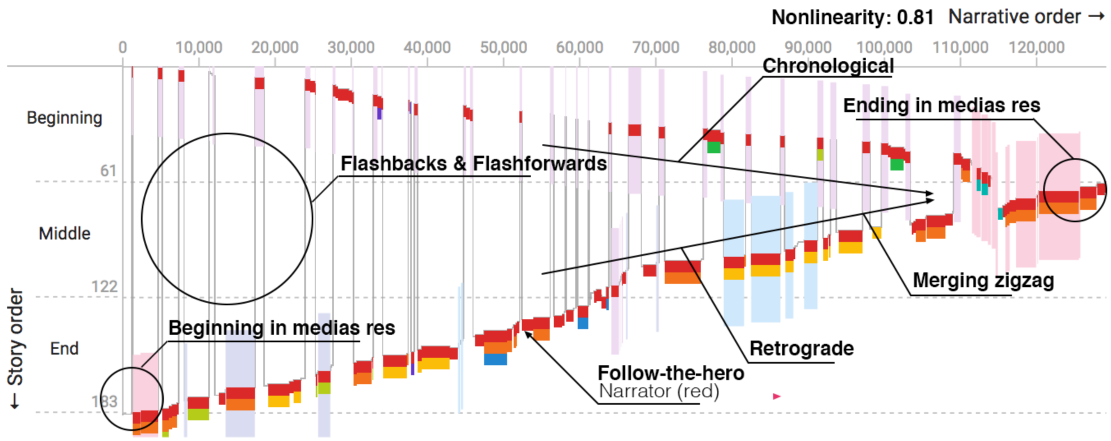


Figure 6.3: *Memento* with characters and scene locations superimposed onto the story curve.

story curves and Story Explorer, facilitating the visual exploration and communication of nonlinear narratives. We first describe the story curves visualization technique (Section section 6.4) and discuss the interactive exploration tool Story Explorer (Section section 6.5).

6.4 Story Curves

6.4.1 Revealing Nonlinear Patterns

A story curve provides a succinct visual summary of the order of events in a nonlinear narrative. Figure 6.1 shows a schematic diagram of how a story curve is constructed from a sequence of events in both story and narrative order (Figure 6.1 (left)). In the story curve (Figure 6.1 (right)) the events are arranged from the left to right (i.e., reading order in Western cultures) following the progression of the film narrative, while their story order is encoded from the top to bottom.

Story events are connected to form a curve such that the up-and-down movements of the curve's trajectory reveal nonlinear narrative patterns (**T1**). The duration of each event is encoded using the horizontal length of the corresponding visual mark. The length can also remain uniform across the

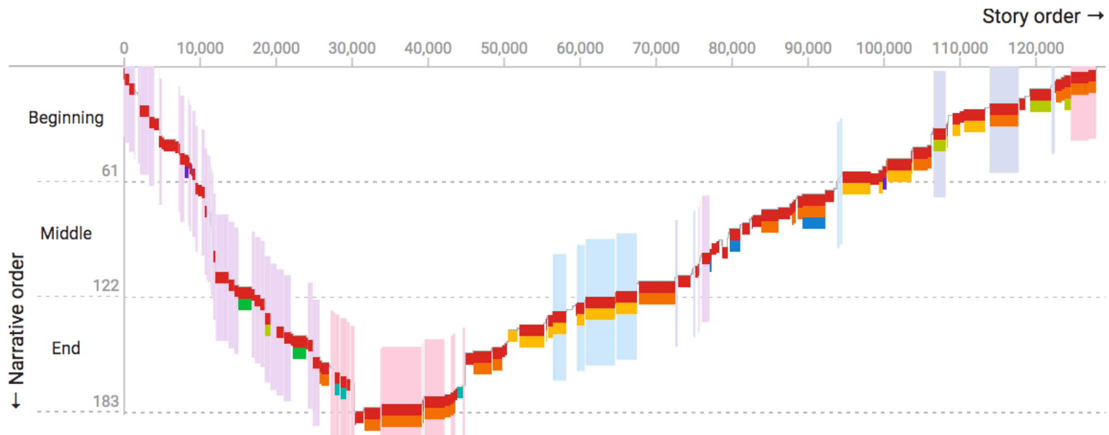


Figure 6.4: *Memento* with reverted axes showing the events in story order from the left to right.

events if necessary to facilitate the analysis of order alone (Figure 6.10b). The visual pattern of the story curve can be perceived by reading how much the curve deviates from the diagonal that represents the chronological timeline. For instance, the first scene and the last scene in the movie *Pulp Fiction* is located in the middle of the story (Figure 6.2), indicating that the narrative begins and ends at the same point of the story. An indexing degree of nonlinearity is calculated from the sum of distances of the events to the diagonal and shown above the story curve in Figure 6.2.

Another example, *Memento*, shows a completely different pattern (Figure 6.3). The narrative begins with the last event, and routinely flashes back to the beginning of the story. As a result, the story curve frequently moves up and down, creating a zigzag pattern. More interestingly, the flashback scenes are narrated in chronological order, while the flashforward scenes are narrated in reverse order. The two distinctive lines of scenes merge towards the end of the narrative. Surprisingly, this one movie contains almost all the patterns from Genette's framework, including chronology, retrograde, flashback, flashforward, and zigzag. A user can opt to flip the curve along the diagonal to read story order from left to right (Figure 6.4).

6.4.2 Visualizing Metadata on Story Curves

A story curve alone only communicates the nonlinear temporality of a narrative. To show a more comprehensive overview of the narrative structure, additional story metadata can be visualized along the curve. The goal is to selectively superimpose the metadata onto the curve while not overloading the user with too much information.

Individual characters are represented as colored segments placed on the curve to support understanding of character occurrences and their interactions (**T2**). The density of the color communicates who are prominent characters in the story based on the frequency of their appearances, while the thickness of the curve communicates how many characters co-occur in each scene. For example, in *Memento* (Figure 6.3), the red color shows that the main protagonist is the only character appearing in the flashback scenes (parts of the curve "spiking" upwards), while he interacts with other characters in the flashforward scenes.

Additional scene information, such as locations and periods of the day, can be added to the story curve as well (**T3**). Locations are represented as a band surrounding character segments. Where information is available, periods of the day are communicated through gray backdrops in the background (Figure 6.1). While not as prominent as characters, the setting information can also reveal interesting patterns. For example, the movie *Memento* begins and ends in the same location, shown by the red band surrounding the curve in Figure 6.3.

6.4.3 Design Alternatives

We considered different design alternatives for showing the temporal relationship between story and narrative (Figure 6.5). Our arc diagram (Figure 6.5a) is similar to the visualization by Sharma et al. [174]. It consists of a straight line from left to right representing story time, and circular arcs to different points along the story line to represent narrative time. Forward time jumps are represented in the

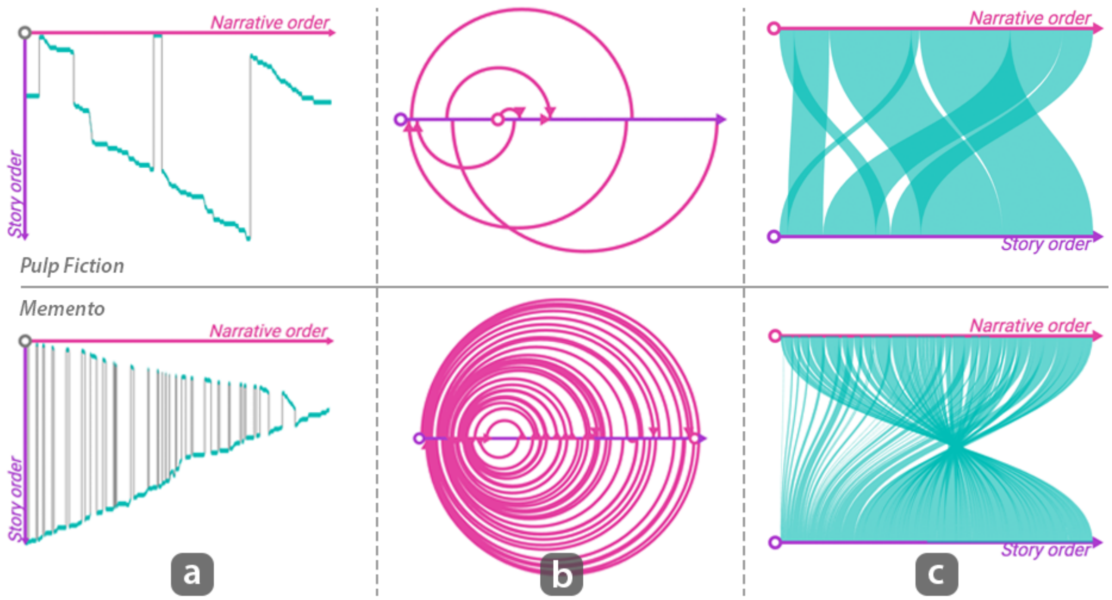


Figure 6.5: Story curves (a) compared to design alternatives (b,c) for *Pulp Fiction* (top) and *Memento* (bottom).

upper part and backward jumps are located in the lower part of the diagram. The bipartite graph (Figure 6.5c) is similar to the infographic by Syed[inf]. It has also two lines representing story time (bottom) and narrative time (top), respectively. Connecting edges are drawn to communicate the temporal connections between events in story and narrative times.

While visually interesting at a glance, both alternatives suffer from line crossings that can generate visual clutter in a complex narrative (Figure 6.5 bottom). In addition, following the narrative timeline and reading temporal patterns can be harder as they require frequent and longer eye movements, e.g., following the arcs back and forth or moving eyes up and down to find corresponding story points.

6.5 Story Explorer

Story Explorer (Figure 6.2) processes a movie script, extracts story elements (scenes, characters, settings), and visualizes the narrative of the movie alongside the script text. It supports the curation

Figure 6.6: The tagging interface that shows a parsed script for Eternal Sunshine of the Spotless of Mind. A user can modify the tag of each line using the dropdown selection on the right side.

of the chronological order of scenes and enables close reading of the script in both story and narrative order.

6.5.1 Extracting Narratives from Movie Scripts

Movie Script

A script, or screenplay, is written and intended for producing a movie or television program. It is usually formatted according to industrial standards[for] that stipulates how script elements are presented (Figure 6.6). The script elements include scene headings, actions, character names, dialogs, and other extra information (e.g., movie editing instructions).

A scene heading, also called slugline, introduces a scene by usually providing three pieces of information including whether the scene is inside (INT.) or outside (EXT.), and a location and time in which the scene takes place. An action describes the setting of the scene in more detail or often introduces characters if necessary. A character name specifies who speaks the dialog that comes after the name. A parenthetical remark is used to describe an attitude of the character; it is not used as frequently as other elements, though.

Processing Movie Scripts

To process movie scripts to extract story elements (scenes, characters, etc.) we implemented a parser for segmenting a script into script elements. We developed a similar method as Pavel et al.[152] that extracts scene headings, actions, character names, dialogs, and parentheticals. We ignore other elements, such as transitions (e.g., CUT TO, FADE TO) and shots (e.g., ANGLE ON), that are technical notes for directing a movie.

Our parser first segments a script into individual lines. It then computes features for each line, including whether the line is in all capital letters (e.g., character names, scene headings, transitions etc), contains a special marker such as *INT.* or *EXT.*, or is enclosed by parentheses. We also detect the left margin of the line. We also maintains a list of words indicating transitions and shots and ignore lines that contain such words.

Next, we separate the segmented script lines into two groups, where the first group contains the lines with all capital letters and the second group contains the rest (actions, dialogs, parentheticals). Using k-means clustering based on the left margin of the lines, it classifies scene headings and character names in the first group, and actions, character dialogs, and parentheticals in the second group. We further make use of the remaining features (i.e., interior/exterior, enclosing parenthesis) to resolve the tag for each line.

Unfortunately, not all scripts are well formatted. Some scripts deviate from the formatting standard (e.g., inconsistent left margins, non-capital letters for some character names, etc.). To work around this problem, we developed a tagging interface to fix the labels of the lines that are incorrectly tagged

Figure 6.7: Pulp Fiction's story curve showing character dialog sentiment; red: negative, gray: neutral, green: positive.

by the parser using a dropdown menu in the interface (Figure 6.6).

Extracting Story Metadata

Once scene and character information is parsed, our system further extracts semantic metadata from the script. From each scene heading, it retrieves the name of the location, the time of day, and whether the scene is inside/outside. The length of each scene is determined based on the amount of text in the scene. The system also derives the sentiment (negative, positive) of characters based on the sentiment of their dialogs in each scene using a simple Naive Bayes classifier trained on movie reviews [130].

Our system requests data from a public movie database [mov] using the movie title as a query, and merges the movie metadata (e.g., ratings, genres, director, cast, etc.) with the script data. To derive the gender of each character, we use the gender of the actor as the gender of the character.

6.5.2 Exploring Movie Narratives

When a user selects a movie in Story Explorer, it retrieves the preprocessed movie narrative data and presents three views: 1) story curves showing the arrangement of scenes in story and narrative

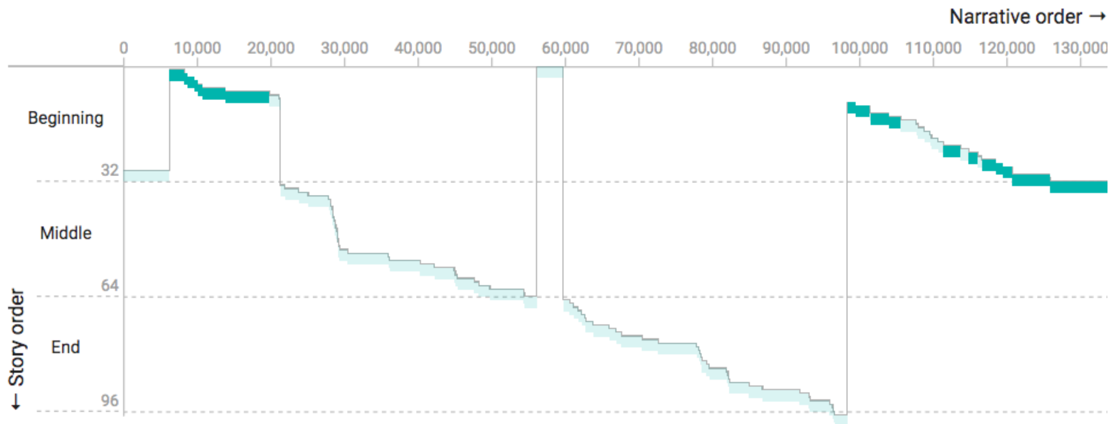


Figure 6.8: *Pulp Fiction*'s story curve showing the co-occurrence of the two main characters, *Jules* and *Vincent* in the morning.

order, 2) metadata view aligned with story curves displaying characters, locations, and periods of the day, and 3) script text as a list of segmented scenes (Figure 6.2).

Navigating Visualizations

The set of visualizations shows an overview of the narrative structure of the movie. The story curve view communicates the nonlinear temporal pattern of the narrative (**T1**), as can be seen in the top-left corner of Figure 6.2. Various modes of operations related to story curves are exposed through interface components. They include flipping axes to read the script in story order, switching to a rich view to place metadata on top of a story curve, and encoding scene length as the number of letters in the scene text. In addition, a user can choose different color encodings of the curve segments to display characters, character gender (Figure 6.2 a), and dialog sentiments (Figure 6.7).

To avoid visual clutter, scene setting information is initially displayed in the metadata view, separately from the story curve. The metadata view is vertically aligned with the story curves so that each column is a scene in both visualizations. A user can selectively project each character, location, or time onto the curve (Figure 6.2). This enables analysis of the story metadata in the context of the nonlinear timeline of the narrative (**T2, T3**). Instead of superimposing additional visual elements, the segments of the story curve are highlighted to show the co-occurrence of the metadata when a rich-view mode is not enabled, e.g., co-appearance of characters in a specific time and location (Figure 6.8).

Reading Movie Scripts

In addition to quickly grasping the narrative structure of the movie through the visualizations, a user can read the actual movie script in order to inspect the details of each scene (**T4**). The purpose of the script view is to supplement the visualizations by enabling close reading.

The script is shown as a segmented list of scenes, each of which corresponds to a column in the visualizations (Figure 6.2b). Both the visualizations and script reading interface are coordinated so that a user can dig into the script of each scene from the curve, and vice versa. While the scenes are initially arranged in the movie's narrative order, the user can read the script in story order by flipping the axes of the narrative curve.

Rearranging Scenes in Story Order

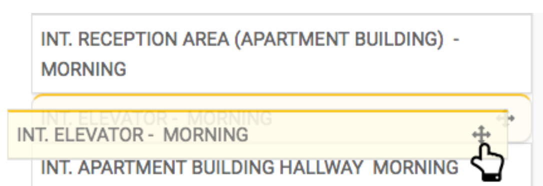


Figure 6.9: Arranging scenes in story order in the script reading interface.

The original script does contain the story order of the scenes. In order to reconstruct the arrangement of the scenes in chronological order, a user can drag and drop each scene segment in the script reading interface (Figure 6.9). This manual reordering can be cognitively demanding when lack of temporal information is available. For example, it took us 30 to 60 minutes on average to complete the rearrangement for each of the 10 selected movies that we analyze in section 6.6.

6.6 Story Curve Patterns for Nonlinear Films

We now report on narrative patterns that we could observe using story curves. Our intent here is threefold. We want to demonstrate how to generally read story curves; we want to show how the basic set of narrative patterns is represented (section 6.2); and finally we want to demonstrate the power of story curve visual patterns that led us to the discovery of additional narrative patterns that may have not been described in the literature.

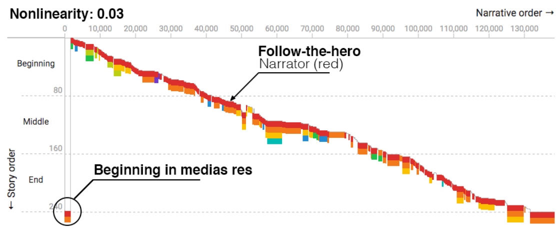
We created story curves for a selected set of popular nonlinear narrative movies: *Memento*, *Pulp Fiction*, *Eternal Sunshine of the Spotless Mind*, *The Usual Suspects*, *Reservoir Dogs*, *Annie Hall*, *500 Days of Summer*, *12 Monkeys*, *Fight Club*, and *Prestige*. We gathered the movie scripts from a public database[scr] and manually restored the story order of the scenes in each movie using our interface (Figure 6.5.2). The following description of our analysis refers to Figure 6.10.

6.6.1 Genette's Basic Structural Patterns

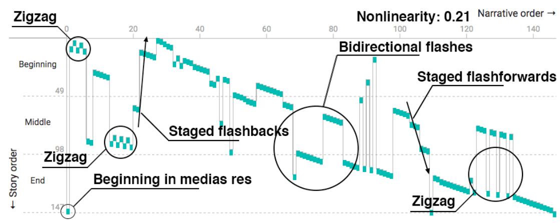
All the movies we considered contain at least a couple of Genette's basic patterns. In our analysis, we considered six patterns: chronological, retrograde, flashbacks (analepsis), flashforwards (prolepsis), zigzag, and syllepsis (section 6.2).

Flashbacks and flashforwards are most commonly observed in the movies, which can be easily identified by the up-and-down movements of the trajectory of the corresponding story curves. For instance, *Fight Club* begins with the last event and quickly flashes back to the beginning of the story; similar flashback patterns can be observed in *Annie Hall*, *500 Days of Summer*, and *12 Monkeys*.

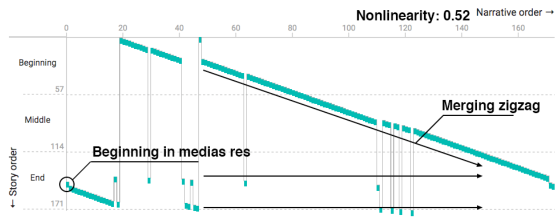
A more extreme use of flashbacks and flashforwards are found in *Memento* and *Eternal Sunshine of the Spotless Mind*, eventually creating a prominent zigzag pattern. Similarly, *The Usual Suspects* also shows a zigzag pattern interweaving two time periods. Other interesting cases can be found in *500 Days of Summer* and *Prestige* that both show multiple short zigzags.



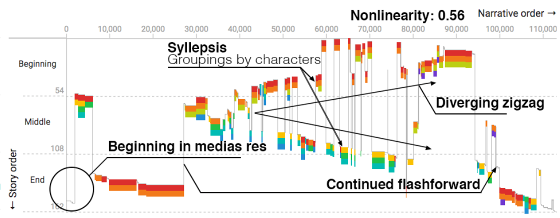
(a) *Fight Club*



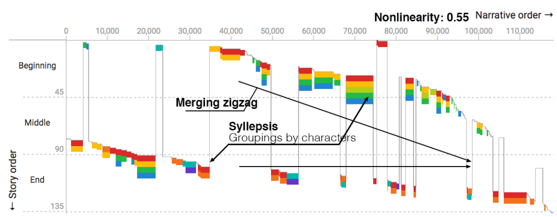
(b) *500 days of summer*



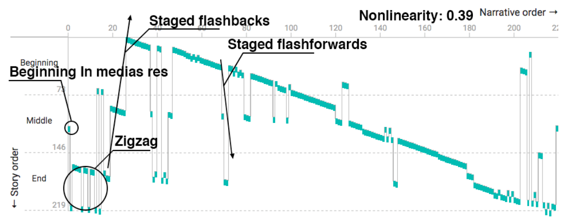
(c) *12 Monkeys*



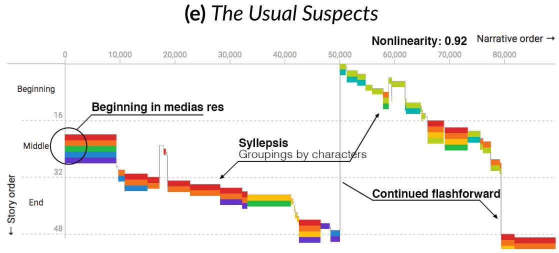
(d) *Eternal Sunshine of the Spotless of Mind*



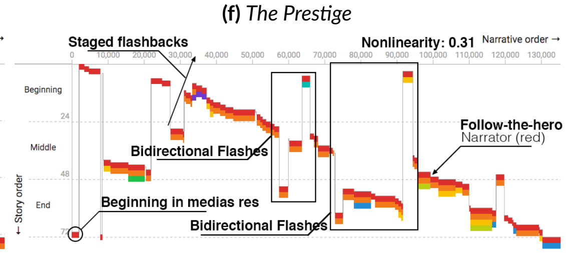
(e) *The Usual Suspects*



(f) *The Prestige*



(g) *Reservoir Dogs*



(h) *Annie Hall*

Figure 6.10: Examples of story curves for selected movies with nonlinear narratives. Narrative order advances horizontally from left to right, story order vertically from top to bottom. Colors on the curve indicate the presence of characters. The nonlinearity value is computed by the degree of deviation from the diagonal line (chronological timeline).

Syllepsis (groupings) are difficult to observe in the movies. Two notable cases are Quentin Tarantidno's movies *Pulp Fiction* and *Reservoir Dogs*. For instance, *Pulp Fiction* has three main interrelated stories with multiple protagonists (Vincent, Butch, and Jules). Based on the colors of the characters, it is easy to see that two groups of scenes belong to Jules (red color), one group belongs to Vincent (orange), and another group belongs to Butch (yellow).

6.6.2 Extensions to Genette's Structural Patterns

Genette's patterns mostly describe local patterns in the narratives, often spanning only a small set of scenes. Combining basic patterns can lead to higher-level structural patterns being composed of multiple other patterns that eventually can characterize an entire movie. Using story curves we were able to discern the following extended (global) patterns:

Beginning/ending in medias res: All the movies we had showed this narrative pattern. They begin at either the middle (e.g., *Pulp Fiction*, *Reservoir Dogs*) or the end (e.g., *Annie Hall*, *Fight Club*) of the story, and subsequently use flashbacks to explain earlier events in order to fill in the backstory. Most movies end at the last event of the story, except *Pulp Fiction* and *Memento*.

Continued flashbacks/flashforwards: These take the narrative back to the moment where an earlier flashback/flashforward ended, connecting disjointed, yet related groups of events that are separated apart in the narrative. For example, in *Pulp Fiction*, Jules' story consists of two parts of scenes that are disconnected by the interjected stories of Vincent and Butch; the second part is continued after Butch's story through a flashback. A similar case can be found in *Eternal Sunshine of the Spotless Mind*, where the narrative flashes back to explain why Joel and Clementine's memories were removed, and then jumps back to where it left off.

Staged flashback/flashforward: A flashback or flashforward followed by the same one and leading to a stepwise narration of events. For instance, *500 Days of Summer* shows this pattern using a series of explanatory flashbacks to show different points in the romantic relationship, creating a staircase-like pattern. Similarly, in the same movie, multiple flashforwards are used to come back to the most recent time. This is common in movies that have multiple time periods such as The *Prestige* and *Annie Hall*.

Bidirectional flashes: Flashbacks and flashforwards are intertwined with one another to reveal past and future events back and forth. This pattern is also commonly found in narratives that involves multiple time periods. It is similar to a zigzag pattern spanning more than two time periods, but the visual pattern is somewhat unique. The most prominent patterns of this category can be observed in *Annie Hall* and *500 Days of Summer*, e.g., the narrative travels through time to show the changes in characters' emotions or objectives, which are not necessarily temporally dependent.

Merging/diverging zigzags: This pattern shows that multiple groups of scenes create sub-narratives that diverge or converge. For example, in Memento, the flashback scenes and the flashforward scenes meet at the end of the movie, creating a converging visual pattern. On the other hand, *Eternal Sunshine of the Spotless Mind* shows the opposite pattern. While less unique than the two movies, *The Usual Suspects* and *12 Monkeys* also show similar patterns.

6.6.3 Semantic Patterns

So far, we have discussed structural patterns. Structural patterns represent how narrative order alters and jumps between story order. Semantic patterns include additional information about the movie, such as persons, places, and time of the day.

Follow-the-hero: This pattern jumps around in story order but always follows the main character in the story. It is found in many movies, including *Fight Club*, *Annie Hall*, and *Memento*. A common characteristic of these movies is that they have a single protagonist usually as a narrator, and begin the narrative with the most recent event followed by explanatory flashbacks. For instance, *Annie Hall* opens with a monologue by Alvy regretting breaking up with a former girlfriend Annie, and Alvy narrates his past relationship with Annie throughout the movie.

Grouped-by-character: As the narrative changes in story time so do the characters. The change of characters often reveal interesting patterns. For example, character occurrences in different events reveal the multiplot structure of *Pulp Fiction*, which is further accentuated by locations and time of day (i.e., a kind of syllepsis). *Eternal Sunshine of the Spotless Mind*, *Memento*, and *The Usual Suspects* also show interesting character groupings separated in two different time periods in a zigzag pattern.

6.7 Use Cases

In this section, we briefly describe potential use cases for story curves and Story Explorer that we identified through interviews with three professional writers (W1, W2, W3) and one literary scholar (L1). During the interviews, we introduced Story Explorer as well as the patterns we discovered. All experts had years of experience in writing, were very familiar with nonlinear narratives, and thus were easily able to grasp the overall idea of story curves.

All participants agreed that being able to see the overall temporal structure of a narrative is interesting and useful; e.g., W1: "*The visuals look like musical notes. A literary work has also rhythm. It is fantastic to see the narrative structure in this way.*", and L1: "*Finding patterns indicating differences between directors and even different storytelling styles is valuable*", W3: "*Very interesting, I think that Woody Allen's stand up comedy career might have influenced the bi-directional flashes in Annie Hall*".

Although the writers said that the tool may not be useful within the writing process (W2: "*When I write, I just write, everything else gets in the way*", W3:"*Some writers outline events in advance, but I usually don't*"), they all commented that it can be useful at a later stage such as revision or analysis; W1: "*As I refine my script, I rearrange scenes very frequently. I really like the rearranging interface*", W2: "I *often have to read and reorganize more than 70 pages and it is helpful to visually see the overview of a narrative.*", W3: "*With this kind of tool, reading existing scripts can be easier and entertaining*". This in part correlates with our prior assumption that writers often use markers to annotate different points in time[55,47].[*]

Two writers particularly commented on the potential usefulness of our work for students in film studies; W1: "*students often have a hard time writing a good narrative even if they have a good story. Writers are all about narratives, however. They especially don't know how to use time well and often overuse flashbacks. This tool can visually teach how time is manipulated in a narrative*", and W3: "*Being*

[*] https://cgblake.wordpress.com/2011/12/06/linear-vs-non-linear-narrative/

Table 6.1: A selected set of questions from the pattern reading task in the usability study with 13 participants; i.e., less discriminative and overlapping questions are not included. DF shows the difficulty of each question, i.e., a percentage of correct responds, while DC indicates how discriminative each question is in terms of distinguishing the high scoring and low-scoring participants.

Movie	Question	AVG.TIME	DF	DC
Pulp Fiction	1. How many flashbacks are there?	13 sec	0.85	0.08
Pulp Fiction	2. How many flashforwards are there?	10 sec	0.85	0.08
Memento	5. Among five basic nonlinear patterns, how many of them do you seen in this story curve?	32 sec	0.75	0.15
Memento	6. What is the overall pattern of the events belonging to the beginning of the story?	30 sec	0.54	0.23
Memento	8. In which range of narrative order can you find the longest retrograde?	23 sec	0.85	0.08
Fight Club	10. At which point of the story does the narrative start?	12 sec	0.46	0.15
12 Monkeys	12. In which range of narrative order can you find the flashback that goes farthest back in story time?	33 sec	0.77	0.15
500 Days of Summer	16. In which range of narrative order can you find a short zigzag jumping within the middle of the story?	20 sec	0.85	0.08

able to quickly see different narrative patterns in existing literature could be useful for them to visualize their own narratives". Based on this feedback, we later found a course material that specifically teaches nonlinear narrative, where our tool can be useful as a teaching aid[57].

Two participants also commented that our work can be useful at the production stage; W2: "*if the scenes are arranged in chronological order, it is easier for the director to decide which scenes to shoot together*", and L1: "*In a TV series, people could use it to help visualize the amount and type of nonlinearity that is typical in early episodes. Similarly, it could help someone who rearrange the rendered scenes and compare different arrangements of events*".

6.8 Readability Study

We were interested in assessing the readability and learnability of story curves. We conducted an online user study to see whether people with no expertise in visualization and narrative theory can learn to read story curves to perform low-level pattern reading tasks. As the result of this study, we also wanted to extract a systematically developed set of pattern reading questions (Table 6.1) appropriate for a literacy assessment test of story curves, inspired by Lee et al[123].

We recruited a total of 13 participants through various university mailing lists. We showed them story curves that we had created from actual movies and asked 20 questions that required correct interpretation of story curves. Participants were aged between 18 and 35 (8 female, 5 male) and were pursuing, or had pursued, a higher education degree in engineering or design. 12 were graduates, 1 was an undergraduate. The average proficiency in English was 4.69 on a 5-point Likert scale (5: native, 1: elementary).

6.8.1 Procedures

First, participants were introduced to basic nonlinear narrative patterns including flashbacks and flash-forwards as well as how they are represented through story curves. Participants were also shown how story curves are constructed, and then were required to complete two tasks.

A first task asked them to read a plot summary of the movie *Pulp Fiction* containing 13 events, each of which was described by a few sentences. Then, participants had to arrange these events in their chronological order, which is often used in teaching film studies[57]. Participants were then required to draw a corresponding story curve into a canvas with a grid in the background; the grid size was set to the number of events. The time limit for the entire reading-and-sketching task was set to 20 minutes. We expected this task to serve as active learning of story curves and only wanted to see how difficult the reordering task was.

After the sketching, participants were presented with individual story curves (without movie names) and a total of 20 multiple-choice questions, some of which are stated in Table 6.1. Most questions had 5 choices, while some had 3 choices. Questions required to understand and read patterns in the story curves for *Pulp Fiction*, *Memento*, *Eternal Sunshine of the Spotless Mind*, *12 Monkeys*, and *Fight Club* (Figure 6.10). Time limit for each question was 2 minutes.

6.8.2 Measures

We measured the task performance for the second task based on time and accuracy of the responses. The purpose of this measurement was to see if participants are able to read patterns from story curves even with relatively short learning time (i.e., reading the tutorial and completing the sketching task).

The accuracy score per participant was calculated based on the number of correct responses out of 20 questions. The score was corrected for guessing[67]. I.e., a participant had a 1/5 chance of getting a 5-choice question correct due to random guessing:

$$CS = R - \frac{W}{K-1},$$

where R is the number of correct responses, W is the number of wrong responses, and K is the number of choices for a question.

In order to judge performance, we also needed to consider the appropriateness of the questions we asked. Similar to the visualization literacy test by Lee et al.[123], we derived an *item difficulty index* (DF) and an *item discrimination index* (DC) for each question. The difficulty index ranges from 0.0 to 1.0 and is equal to the average ratio of correct responses per question:

$$DF_i = \frac{C_i}{N},$$

where C_i is the number of correct responses and N is the number of participants. The discrimination index shows how well the question distinguishes between high-scoring participants and low-scoring participants. The index ranges from -1.0 to 1.0 and is computed using:

$$DC_i = \frac{H_i - L_i}{N},$$

where H_i is the number of participants who answered the i-th question correctly in the high-scoring group, and L_i is the same number in the low-scoring group.

6.8.3 Results

The average score per participant for the pattern reading task was 16 (σ=3.37). The raw scores ranged from 9 to 20. The corrected score was 14.74 (σ=4.39). The average time taken to complete the whole task was 7.49 minutes per participant (σ=1.83, min=4.21, max=9.76). The learning time was around 17 minutes on average (reading the tutorial: 2.21 and completing the sketching task: 15.14 minutes). The overall task performance was a success, and participants were able to correctly answer 80% of the questions on average.

When we looked at individual questions, each question took 22.48 sec ($\sigma = 9.78$) on average per participant, showing that the time limit (2 minutes per question) was appropriate. The average difficulty index (or the percentage of participants who answered each question correctly) was 0.80 (σ=0.80, min=0.46, max=1.0). Based on the classification scheme used by Lee et al.[123], we had 6 easy ($DF_i > 0.85$), 12 moderate ($0.50 < DF_i <= 0.85$), and 2 hard ($DF_i <= 0.50$) questions. As with the overall performance, the results suggest that participants were able to fairly easily and quickly answer the pattern reading questions.

The discrimination index ranged from -0.08 to 0.31. We had 2 high-discriminative ($DC_i > 0.3$), 5 medium-discriminative ($0.1 < DC_i <= 0.3$), and 13 low-discriminative questions ($DC_i < 0.1$)[123]. We had four negative discrimination values ($DC_i < 0.0$), indicating that the corresponding questions may not be desirable and could be misinterpreted by participants. I.e., the low-scoring group would answer the question correctly but the high-scoring group would not. The rest of low-discriminative questions ($0.0 <= DC_i < 0.1$) suggests that all the participants performed well on the questions, thus less discriminative.

Based on these results, we selected representative questions summarized in Table 6.1 based on performance diversity (DF), question quality (DC), and question diversity (some questions were reused across different movies). This set of questions provides a basis for future readability studies regarding story curves and may allow researchers to compare other techniques.

Around half of the participants ($DF_{20} = 0.46$) were able to correctly answer the last question that is designed to measure the success of the sketching task. It was one of the most discriminative question ($DC_{20} = 0.31$). When we looked at the difference between participants who answered the last question correctly or incorrectly, the overall score was significantly different ($p < 0.05$). This suggests that the sketching had a positive effect on learning the mechanics of story curves.

In a follow-up survey with 5-point Likert scale questions (1-strongly disagree, 5-strongly agree), participants indicated that they are able to read story curves (mean=4.08, σ=0.64) and to apply story curves to represent nonlinear narratives in movies they had watched or will watch in the future (mean=4.00, σ=0.78).

In terms of the difficulty of the tasks (1-very easy, 5-very difficult), participants rated the sketching task slightly difficult (mean=3.90, σ=0.94). They indicated that figuring out the chronological order of events (μ=3.91, σ=0.54) is more difficult than the drawing of story curves (μ=2.27, σ=0.90). Most participants found the pattern reading tasks rather easy (μ=2.09, σ=0.54).

6.9 Discussion

6.9.1 Lessons Learned from the Readability Study

The results of our user study indicate that story curves are easily graspable. A number of participants commented that it is fun and enjoyable to learn about story curves; e.g., P5: *"[...] a lot of fun!"*, P10: *"[...] an interesting experiment to understand narrative and story in graphics."*, and P11: *"I was able to recognize Memento's curve. It is a totally fascinating idea."*. Some participants struggled to understand the design of story curves; e.g, P3: *"Putting the origin at the upper left corner was initially disorientating"*, P9: *"I needed to remind myself that one thing is narrative and the other is chronological."* These comments suggest that being able to control the origin of the axes and a visual aid for reading two axes (e.g., double crossline in Story Explorer) could be useful for reading story curves. Inspecting the responses, we also found that some participants were confused between flashforwards and flashbacks. One option to resolve that confusion might be arrows indicating the direction of time jumps.

One participant particularly commented on the sketching task that "*I thought it was very interesting, specially the part of drawing the story curve*" (E13). The overall results indicate that reconstructing original order of events is not easy even for such short narrative text. One participant suggested to provide a notepad and being able to directly drag events into the canvas, which we consider a useful future extension.

6.9.2 Extending Story Curves and Story Explorer

Story curves were developed mainly based on Genette's analysis on event order in narratives. Genette discussed other dimensions of temporal nonlinearity, such as frequency (e.g., repetitive descriptions of a single story event) and duration (e.g., time taken to narrate a story event). For example, in Pulp Fiction, a flashback scene explains an incident that happened several decades ago, while the story curve currently does not accurately communicate this amount of time. One could enrich story curves with additional temporal information instead of using event order alone. E.g., one could be using a log scale for story time while keeping a linear scale in narrative time.

There are some extensions to Genette's patterns in the literature[84,163]. While Genette's framework is mostly applicable to nonfictional narratives or realistic fictions, it may not generalize well to ones that contain numerous violations of realistic temporality[163]. Richardson describes various temporal paradoxes that can exist in narratives, including time loops and parallel timelines and there are several movies that have such temporal structures, e.g., *Looper* or *Inception*. Story curves can potentially be extended to represent such parallel, branching, and repeating patterns. However specific challenges may arise (e.g., layout, clutter, visual encoding) that require a careful visual design.

Eventually, given the challenges of restoring the actual chronological order of events, we plan to automatically assist the ordering of events in Story Explorer. While there are preliminary works on automatically inferring the causal relationship between events[46,45], an ideal solution would be a mixed initiative system leveraging human cognition.

6.9.3 Generalizations to Other Domains

Though our examples in this paper exclusively involved movies, story curves and Story Explorer are generalizable to other genres such as theater plays, novels, poems, and song lyrics. Of course, each of these genres may introduce specific structural patterns and information to visualize.

Beyond the domain of story narratives, story curves could be applied to other problem that involves the comparison of two orderings for the same set of elements. The data structure underlying story curves is essentially a bipartite graph with an ordering in both node sets and an identity relation as links. Examples include the comparison of rankings in sports analytics and demographics, or the visualization of a chromosome rearrangement in which the order of nucleotides is modified in the structure of the chromosome[142].

6.10 Conclusions

Nonlinear narratives are often complex to understand due to the disruption of the direct causal relationship between events in order to increase suspense. We developed story curves, a visualization technique for communicating nonlinear narratives. Based on the technique we built Story Explorer, an interactive system that allows users to curate the chronological sequence of scenes in a movie script and explore the nonlinear narrative of the movie using story curves. We illustrated several examples of story curves using popular nonlinear movies, evaluated the readability of story curves through a user study, and highlighted potential use cases in screenplay analysis and story production.

The ability to take data – to be able to understand it, to process it, to extract value from it, to visualize it, to communicate it's going to be a hugely important skill in the next decades.

Hal Varian

7

Conclusion & Future Work

This dissertation explored the design space of visualization systems for communication and storytelling through the development of a series of interactive prototypes. DDG enables the design of expressive data graphics by giving a flexible way to bind data to custom shapes. DataSelfie goes one step further to allow users to design a personalized visual vocabulary to represent their own data, which also tightly couples visualization design with a data collection plan. On the other hand, DataToon leverages elements of comics to create a full-fledged data story beyond a simple visualization image.

story curves reveals the narrative structure of the story to enable rapid evaluation and iteration of different story ideas. Overall, this thesis establishes a research framework for creating visual, data-driven systems for presentation and storytelling, going beyond traditional visualization systems.

This chapter concludes this thesis with discussions about important avenues for future work.

7.1 Future Directions

7.1.1 Explanatory and Exploratory Data Stories

The field of visual, data-driven storytelling is still nascent. What is particularly interesting, but under-explored is the idea of marrying both exploration and explanation into a cohesive data story. Design considerations for creating such bimodal data stories can be divided into two separate user dimensions: writers (authoring) and readers (consumption). In the reader's side, we are now seeing many innovative examples [180] (e.g., Explorable Explanations) but still creating the examples requires significant programming expertise from the author's side. DataToon is only an initial exploration of the integrated authoring space for both analysis and presentation.

What kind of tool support can we provide to a lay audience for authoring exploratory and explanatory stories? Our thinking and creativity are often limited by the tools we have. For instance, Word Processing Software like MS Word and Google Docs changed our ways of producing documents and is now deeply immersed in our lives. While tools for writing dynamic documents (e.g., web pages) are emerging, flexibility and expressivity are still significantly limited when it comes to the coupled interaction of texts and visualizations. In particular, nowadays visualizations are rarely used standalone but integrated into other types of media such as videos, comics, documents, spreadsheets, etc. Supporting the design of new interactive, data-driven media is an important research direction to equip everyone with the skills to work with data efficiently and effectively.

Data storytelling require a complex set of both analysis and creative skills, which is especially hard for a general audience to acquire. The recent advancement of artificial intelligence can shed light on bringing intelligent support into data storytelling tools from extracting insights to crafting a narrative from the insights. The automatic detection of interesting data patterns in DataToon and autocomplete of drawing in DataSelfie are an initial step in this direction. But numerous research opportunities lie ahead such as automatically finding interesting story pieces based on data patterns as well as providing higher-level narrative design support.

Data science is an emerging field whose results have important implications for our lives. However, much of the efforts in this field are currently geared towards on exploration and analysis of data. For instance, a recent study of 200 academic computational notebooks, a popular medium for documenting and sharing exploratory analysis, revealed that few describe analytical reasoning and results of the analysis [169]. The public communication of data science still has a lot of rooms to be improved, not only to communicate the results to the general public but also to improve knowledge transfer among data scientists. How can we provide structured support for analysts to encode analysis results in an easily consumable form? Addressing this question will also have implications for promoting reproducible science.

7.1.2 Collaborative & Immersive Data Storytelling

Storytelling is inherently a collaborative activity, involving both tellers and listeners through conversations. However, collaboration, in general, is vastly under-supported in most visualization tools except a few preliminary studies [99,83], particularly in the context of data storytelling. On the other hand, collaboration is a well-studied topic in computer-supported cooperative work [24]. We can draw inspirations from this field enable collaboration throughout the life cycle of data storytelling from insight discovery to narrative design. To support collaborative data storytelling, different design considerations can emerge, including user roles (e.g., analysts, storytellers, readers), activities (e.g., data exploration, narrative creation, and presentation & sharing), scenarios (e.g., author-driven, reader-driven, mixed stories), modes (e.g., synchronous vs asynchronous, remote vs collocated), as well as environments (e.g., desktops, mobiles & tablets, and AR & VR). A particularly exciting direction is how to enable conversations and collaborations between storytellers and readers. Can readers take part in enriching stories, potentially suggesting additional insights or providing fact-checking for other readers? The recent advancement of web technology is certainly favorable for this new research direction as witnessed by many other collaborative tools like Google Colaboratory, although lack of off-the-shelf programming interfaces and infrastructures for developers of collaborative software is still a major technical bottleneck.

Virtual and augmented reality technologies, as well as large displays and hand-held devices with multimodal interaction capability[60,105,88,176], are providing new ways to experience information and opening up a lot of research opportunities in visualization. These technologies have been considered in visual analysis contexts but not necessarily data storytelling. What could be the true benefits of these technologies for data storytelling contexts? For instance, AR & VR often focus on presenting information in a 3D space, but traditionally, visualization researchers avoided 3D representations unless the underlying data has inherent 3D spatial mappings. Given the past results of perception research, it is less likely useful for time-critical decision-making scenarios. However, this new emerging medium can provide an engaging way to create and consume data stories; for instance, how can we bring the interactivity of DataToon into AR and VR environments? What would be an efficient way to manage multiple-coordinated 2D visualizations in 3D space?

7.1.3 Science of Data Storytelling

Even with the widespread adoption of visualizations for data stories, we still do not have much knowledge about what makes an effective data story and how visualizations contribute to it. In order to support faster and more accurate analysis of data, there have been many perceptual studies to identify efficient preattentive visual variables to encode different types of data in a variety of analysis tasks.

When it comes to visualization for communication and storytelling, there are many more cognitive dimensions involved than perception alone, such as aesthetics[118], engagement[95] and memorability[35]. However, there is still no established set of cognitive design principles for designing effective visual data stories. New tools like DataToon and DataSelfie can allow for new types of studies focused on presentation and storytelling, but the design and user space of data stories can be too broad to run traditional controlled experiments at scale. A potential approach to address this issue is to crowdsource visualization experiments in which users test their own hypothesis to extract design guidelines, similar to Lab in the Wild[151]. Another direction is to leverage the well-established narrative theories in humanities and computational narratives, that are not deeply considered to date in the visualization community.

Once we establish the set of design principles, as well as new techniques for data storytelling, it becomes essential to educate the public about the new knowledge, often dubbed *visualization literacy*[39]. Being able to read and write visualizations and visual data stories is becoming important, as they become popular channels to access data. People nowadays rarely see raw data tables but visual representations of the data, especially through journalism media. To avoid information inequality, they should be able to decode visual mappings; e.g., areas represent numerical quantities in a treemap. Similarly, when encoding visual mappings and stories, people need to be aware of the pros and cons of their design decisions such as caveats of different encoding channels (e.g., the rainbow colormap[34]).

Creating a visual data story can be much more involved than designing a single visualization, such as keeping multiple views consistent[158] and understanding the double-edged sword aspect of a story[58]. The current approach to visualization literacy is to develop passive, instructional guidelines. This usually works for people with enough data literacy like analysts but not for novices such as students and children. We can envision a framework that takes the constructionism approach to allow learners to actively take part in the learning process (e.g., deconstructing or constructing graphs by themselves).

7.2 Outlook

The main idea of this dissertation is to go beyond traditional exploratory visualizations to create expressive, explanatory, and personal visual stories. In this way, presentation and storytelling become a core part of visualization systems. Rather than create a dichotomy between exploration and explanation, visual data storytelling tightly couples exploring data to discover insights with arranging the insights to tell stories as a cohesive process. The resulting systems are more expressive and powerful than traditional visualization software. I believe that this tight integration of analysis and presentation will be a powerful model for future visualization systems.

Bibliography

[inf] 16 complicated movie plots explained with infographic timelines. http://www.scribblelive.com/blog/2013/08/26/16-movie-timeline-infographics/. Accessed: 2017-03-31.

[gam] How every #gameofthrones episode has been discussed on twitter. https://interactive.twitter.com/game-of-thrones/. Accessed: 2017-03-31.

[scr] The internet movie script database. www.imsdb.com/. Accessed: 2017-03-31.

[les] Les misérables co-occurrence. https://bost.ocks.org/mike/miserables/. Accessed: 2017-03-31.

[non] Linear vs. non-linear narrative. https://cgblake.wordpress.com/2011/12/06/linear-vs-non-linear-narrative/. Accessed: 2017-06-24.

[ort] Lostalgic. http://intuitionanalytics.com/other/lostalgic/. Accessed: 2017-03-31.

[mov] The movie database. https://www.themoviedb.org/. Accessed: 2017-03-31.

[for] Screenplay format guide. http://www.storysense.com/SPFormat.pdf. Accessed: 2017-03-31.

[sto] Story touch - scriptwriting tools. http://www.storytouch.com/o-que-e.html. Accessed: 2017-03-31.

[sta] Visualizing star wars movie scripts. http://labs.precognox.com/star-wars-visualization/. Accessed: 2017-03-31.

[11] Aarseth, E. (2012). A narrative theory of games. In *Proceedings of the international conference on the foundations of digital Games* (pp. 129–133).: ACM.

[12] Abello, J., Broadwell, P., & Tangherlini, T. R. (2012). Computational folkloristics. *Communications of the ACM*, 55(7), 60–70.

[13] Al-Awami, A. K., Beyer, J., Strobelt, H., Kasthuri, N., Lichtman, J. W., Pfister, H., & Hadwiger, M. (2014). Neurolines: a subway map metaphor for visualizing nanoscale neuronal connectivity. *IEEE Transactions on Visualization and Computer Graphics*, 20(12), 2369–2378.

[14] Amini, F., Brehmer, M., Bolduan, G., Elmer, C., & Wiederkehr, B. (2018). Evaluating data-driven stories & storytelling tools. In N. Henry Riche, C. Hurter, N. Diakopoulos, & S. Carpendale (Eds.), *Data-Driven Storytelling*. A K Peters/CRC Press. https://aka.ms/dds_book.

[15] Amini, F., Henry Riche, N., Lee, B., Hurter, C., & Irani, P. (2015). Understanding data videos: Looking at narrative visualization through the cinematography lens. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)* (pp. 1459–1468).: ACM.

[16] Amini, F., Riche, N. H., Lee, B., Monroy-Hernandez, A., & Irani, P. (2017). Authoring data-driven videos with DataClips. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis)*, 23(1), 501–510. http://doi.org/10.1109/TVCG.2016.2598647.

[17] Anscombe, F. J. (1973). Graphs in statistical analysis. *The American Statistician*, 27(1), 17–21.

[18] Ayobi, A., Sonne, T., Marshall, P., & Cox, A. L. (2018). Flexible and mindful self-tracking: Design implications from paper bullet journals. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18 (pp. 28:1–28:14). New York, NY, USA: ACM.

[19] Bach, B., Kerracher, N., Hall, K. W. M., Carpendale, S., Kennedy, J., & Henry Riche, N. (2016). Telling stories about dynamic networks with graph comics. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)* (pp. 3670–3682). http://doi.org/10.1145/2858036.2858387.

[20] Bach, B., Riche, N. H., Carpendale, S., & Pfister, H. (2017). The emerging genre of data comics. *IEEE Computer Graphics and Applications (CG&A)*, 37(3), 6–13. http://doi.org/10.1109/MCG.2017.33.

[21] Bach, B., Stefaner, M., Boy, J., Drucker, S., Bartram, L., Wood, J., Ciuccarelli, P., Engelhardt, Y., Köppen, U., & Tversky, B. (2018a). Narrative design patterns for data-driven storytelling. In N. Henry Riche, C. Hurter, N. Diakopoulos, & S. Carpendale (Eds.), *Data-Driven Storytelling*. A K Peters/CRC Press. https://aka.ms/dds_book.

[22] Bach, B., Wang, Z., Farinella, M., Murray-Rust, D., & Riche, N. H. (2018b). Design patterns for data comics. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. https://doi.org/10.1145/3173574.3173612.

[23] Bach, B., Wang, Z., Riche, N. H., Farinella, M., Murray-Rust, D., Carpendale, S., & Pfister, H. (2018c). Data comics. http://datacomics.net.

[24] Baecker, R. M. (1994). *Readings in GroupWare and Computer-Supported Cooperative Work: Assisting Human-Human Collaboration*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1st edition.

[25] Bal, M. (2009). *Narratology: Introduction to the theory of narrative*. University of Toronto Press.

[26] Barthes, R. (1974). S/z, trans. *Richard Miller (New York: Hill and Wang, 1974)*, 76.

[27] Bateman, S., Mandryk, R. L., Gutwin, C., Genest, A., McDine, D., & Brooks, C. (2010). Useful junk?: The effects of visual embellishment on comprehension and memorability of charts. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)* (pp. 2573–2582). https://doi.org/10.1145/1753326.1753716.

[28] Bertin, J. (1983). Semiology of graphics: diagrams, networks, maps.

[29] Bigelow, A., Drucker, S., Fisher, D., & Meyer, M. (2014). Reflections on how designers design with data. In *Proceedings of the ACM Conference on Advanced Visual Interfaces (AVI)* (pp. 17–24). http://doi.org/10.1145/2598153.2598175.

[30] Bigelow, A., Drucker, S., Fisher, D., & Meyer, M. (2016). Iterating between tools to create and edit visualizations. *IEEE TVCG*, (pp. in press).

[31] Bigelow, A., Drucker, S., Fisher, D., & Meyer, M. (2017). Iterating between tools to create and edit visualizations. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis)*, 23(1), 481–490. http://doi.org/10.1109/TVCG.2016.2598609.

[32] Bilenko, N. (2016). The narrative explorer. Master's thesis, EECS Department, University of California, Berkeley.

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

[34] Borkin, M., Gajos, K., Peters, A., Mitsouras, D., Melchionna, S., Rybicki, F., Feldman, C., & Pfister, H. (2011). Evaluation of artery visualizations for heart disease diagnosis. *IEEE Transactions on Visualization and Computer Graphics*, 17(12), 2479–2488.

[35] Borkin, M. A., Bylinskii, Z., Kim, N. W., Bainbridge, C. M., Yeh, C. S., Borkin, D., Pfister, H., & Oliva, A. (2016). Beyond memorability: Visualization recognition and recall. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis)*, 22(1), 519–528. https://doi.org/10.1109/TVCG.2015.2467732.

[36] Borkin, M. A., Vo, A. A., Bylinskii, Z., Isola, P., Sunkavalli, S., Oliva, A., & Pfister, H. (2013). What makes a visualization memorable? *IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis)*, 19(12), 2306–2315. https://doi.org/10.1109/TVCG.2013.234.

[37] Bostock, M., Ogievetsky, V., & Heer, J. (2011). D³ data-driven documents. *IEEE TVCG*, 17(12), 2301–2309.

[38] Boy, J., Detienne, F., & Fekete, J.-D. (2015). Storytelling in information visualizations: Does it engage users to explore data? In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)* (pp. 1449–1458).: ACM.

[39] Boy, J., Rensink, R. A., Bertini, E., & Fekete, J.-D. (2014). A principled way of assessing visualization literacy. *IEEE transactions on visualization and computer graphics*, 20(12), 1963–1972.

[40] Brehmer, M., Lee, B., Henry Riche, N., Tittsworth, D., Lytvynets, K., Edge, D., & White, C. (2019). Timeline storyteller: The design & deployment of an interactive authoring tool for expressive timeline narratives. In *To appear in proceedings of the the Computation + Journalism Symposium*. https://aka.ms/TSCJ19.

[41] Brooke, J. et al. (1996). Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194), 4–7.

[42] Carroll, R. (2018). Bullet journal. https://bulletjournal.com/. Accessed: 2018-06-16.

[43] Carter, S., Cox, A., & Bostock, M. (2013). Dissecting a trailer: The parts of the film that make the cut. *The New York Times*.

[44] Castells, P., Szekely, P., & Salcher, E. (1997). Declarative models of presentation. In *Proc. of IUI* (pp. 137–144).: ACM.

[45] Chambers, N. & Jurafsky, D. (2008). : Citeseer.

[46] Chambers, N., Wang, S., & Jurafsky, D. (2007). Classifying temporal relations between events. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions* (pp. 173–176).: Association for Computational Linguistics.

[47] Chatman, S. B. (1980). *Story and discourse: Narrative structure in fiction and film*. Cornell University Press.

[48] Choe, E. K., Abdullah, S., Rabbi, M., Thomaz, E., Epstein, D. A., Cordeiro, F., Kay, M., Abowd, G. D., Choudhury, T., Fogarty, J., Lee, B., Matthews, M., & Kientz, J. A. (2017). Semi-automated tracking: A balanced approach for self-monitoring applications. *IEEE Pervasive Computing*, 16(1), 74–84.

[49] Choe, E. K., Lee, B., et al. (2015a). Characterizing visualization insights from quantified selfers' personal data presentations. *IEEE Computer Graphics and Applications*, 35(4), 28–37.

[50] Choe, E. K., Lee, B., Kay, M., Pratt, W., & Kientz, J. A. (2015b). Sleeptight: Low-burden, self-monitoring technology for capturing and reflecting on sleep behaviors. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '15 (pp. 121–132). New York, NY, USA: ACM.

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

[52] Collins, C., Penn, G., & Carpendale, S. (2009). Bubble sets: Revealing set relations with isocontours over existing visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 15(6), 1009–1016. http://doi.org/10.1109/TVCG.2009.122.

[53] Consolvo, S., McDonald, D. W., Toscos, T., Chen, M. Y., Froehlich, J., Harrison, B., Klasnja, P., LaMarca, A., LeGrand, L., Libby, R., Smith, I., & Landay, J. A. (2008). Activity sensing in the wild: A field trial of ubifit garden. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08 (pp. 1797–1806). New York, NY, USA: ACM.

[54] Dancyger, K. (2014). *The technique of film and video editing: history, theory, and practice*. CRC Press.

[55] Dancyger, K. & Rush, J. (2013). *Alternative scriptwriting: Beyond the Hollywood formula*. CRC Press.

[56] Denis, A., Cruz-Lara, S., Bellalem, N., & Bellalem, L. (2014). Visualization of affect in movie scripts. In *Empatex, 1st International Workshop on Empathic Television Experiences at TVX 2014*.

[57] Dharwadker, A., Dharwadker, V., Harrison, S., Lemaster, T., & Bourenane, H. D. (2012). Teaching the god of small things in wisconsin: A guide for educators.

[58] Dimara, E., Bezerianos, A., & Dragicevic, P. (2017). Narratives in crowdsourced evaluation of visualizations: A double-edged sword? In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17 (pp. 5475–5484). New York, NY, USA: ACM.

[59] Dinkla, K., Strobelt, H., Genest, B., Reiling, S., Borowsky, M., & Pfister, H. (2017). Screenit: Visual analysis of cellular screens. *IEEE Transactions on Visualization and Computer Graphics*, PP(99), 1–1.

[60] Drucker, S. M., Fisher, D., Sadana, R., Herron, J., & schraefel, m. (2013). Touchviz: A case study comparing two interfaces for data analytics on tablets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13 (pp. 2301–2310). New York, NY, USA: ACM.

[61] Dwyer, T. (2018). WebCoLa. http://ialab.it.monash.edu/webcola/. [Online; accessed 31-03-2018].

[62] ECKEL, J. (2014). Twisted times. *(Dis) Orienting Media and Narrative Mazes*, (pp. 275).

[63] Facebook (2018a). React: A JavaScript library for building user interfaces. https://reactjs.org/. [Online; accessed 16-Sep-2018].

[64] Facebook (2018b). Redux: a predictable state container for javascript apps. https://redux.js.org/. [Online; accessed 16-Sep-2018].

[65] Fathom (2015). Scaled in miles. https://fathom.info/miles/. [Online; accessed 11-Sep-2018].

[66] Fekete, J.-D. (2004). The infovis toolkit. In *Prof. IEEE InfoVis* (pp. 167–174).

[67] Frary, R. B. (1988). Formula scoring of multiple-choice tests (correction for guessing). *Educational Measurement: Issues and Practice*, 7(2), 33–38.

[68] Frisch, M., Heydekorn, J., & Dachselt, R. (2009). Investigating multi-touch and pen gestures for diagram editing on interactive surfaces. In *Proceedings of the ACM Conference on Interactive Tabletops and Surfaces (ITS)* (pp. 149–156). http://doi.org/10.1145/1731903.1731933.

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

[70] Genette, G. (1983). *Narrative discourse: An essay in method*. Cornell University Press.

[71] Gershon, N. & Page, W. (2001). What storytelling can do for information visualization. *Communications of the ACM*, 44(8), 31–37.

[72] GmbH, D. (2018). Datawrapper gmbh. https://www.datawrapper.de. [Online; accessed: 2018-08-04].

[73] Göbel, S., Salvatore, L., & Konrad, R. (2008). Storytec: A digital storytelling platform for the authoring and experiencing of interactive and non-linear stories. In *Automated solutions for Cross Media Content and Multi-channel Distribution, 2008. AXMEDIS'08. International Conference on* (pp. 103–110).: Ieee.

[74] Grammel, L., Bennett, C., Tory, M., & Storey, M.-A. (2013). A Survey of Visualization Construction User Interfaces. In *EuroVis - Short Papers* (pp. 19–23).: The Eurographics Association.

[75] Gratzl, S., Lex, A., Gehlenborg, N., Cosgrove, N., & Streit, M. (2016). From visual exploration to storytelling and back again. In *Computer Graphics Forum (Proceedings of EuroVis)* (pp. 491–500). http://doi.org/10.1111/cgf.12925.

[76] Green, T. R. G. & Petre, M. (1996). Usability analysis of visual programming environments: a 'cognitive dimensions' framework. *Journal of Visual Languages & Computing*, 7(2), 131–174.

[77] Hagberg, A., Schult, D., & Swart, P. (2018). NetworkX. https://networkx.github.io/. [Online; accessed 31-03-2018].

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

[79] Harrison, L., Reinecke, K., & Chang, R. (2015). Infographic aesthetics: Designing for the first impression. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15 (pp. 1187–1190). New York, NY, USA: ACM.

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

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

[82] Heer, J., Card, S. K., & Landay, J. A. (2005). Prefuse: a toolkit for interactive information visualization. In *Proc. of CHI* (pp. 421–430).: ACM.

[83] Heer, J., van Ham, F., Carpendale, S., Weaver, C., & Isenberg, P. (2008). *Creation and Collaboration: Engaging New Audiences for Information Visualization*, (pp. 92–133). Springer Berlin Heidelberg: Berlin, Heidelberg.

[84] Heise, U. K. (1997). *Chronoschisms: Time, narrative, and postmodernism*, volume 23. Cambridge University Press.

[85] Hinckley, K., Bi, X., Pahud, M., & Buxton, B. (2012). Informal information gathering techniques for active reading. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)* (pp. 1893–1896). http://doi.org/10.1145/2207676.2208327.

[86] Hinckley, K., Yatani, K., Pahud, M., Coddington, N., Rodenhouse, J., Wilson, A., Benko, H., & Buxton, B. (2010). Pen + touch = new tools. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)* (pp. 27–36). http://doi.org/10.1145/1866029.1866036.

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

[88] Horak, T., Badam, S. K., Elmqvist, N., & Dachselt, R. (2018). When david meets goliath: Combining smartwatches with a large vertical display for visual data exploration. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18 (pp. 19:1–19:13). New York, NY, USA: ACM.

[89] Hoyt, E., Ponto, K., & Roy, C. (2014). Visualizing and analyzing the hollywood screenplay with scripthreads. *Digital Humanities Quarterly*, 8(2).

[90] Hsu, S. C. & Lee, I. H. (1994). Drawing and animation using skeletal strokes. In *Proc. of ACM SIGGRAPH* (pp. 109–118).: ACM.

[91] Huang, D., Tory, M., Aseniero, B. A., Bartram, L., Bateman, S., Carpendale, S., Tang, A., & Woodbury, R. (2015). Personal visualization and personal visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 21(3), 420–433.

[92] Hullman, J., Adar, E., & Shah, P. (2011). Benefitting infovis with visual difficulties. *IEEE Transactions on Visualization and Computer Graphics*, 17(12), 2213–2222.

[93] Hullman, J. & Diakopoulos, N. (2011). Visualization rhetoric: Framing effects in narrative visualization. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis)*, 17(12), 2231–2240. https://doi.org/10.1109/TVCG.2011.255.

[94] Hullman, J., Drucker, S., Riche, N. H., Lee, B., Fisher, D., & Adar, E. (2013). A deeper understanding of sequence in narrative visualization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12), 2406–2415. https://doi.org/10.1109/TVCG.2013.119.

[95] Hung, Y.-H. & Parsons, P. (2017). Assessing user engagement in information visualization. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '17 (pp. 1708–1717). New York, NY, USA: ACM.

[96] Huron, S., Carpendale, S., Thudt, A., Tang, A., & Mauerer, M. (2014a). Constructive visualization. In *Proc. of DIS* (pp. 433–442).: ACM.

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

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

[99] Isenberg, P., Elmqvist, N., Scholtz, J., Cernea, D., Ma, K.-L., & Hagen, H. (2011). Collaborative visualization: Definition, challenges, and research agenda. *Information Visualization*, 10(4), 310–326.

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

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

[Jänicke et al.] Jänicke, S., Franzini, G., Cheema, M. F., & Scheuermann, G. On close and distant reading in digital humanities: A survey and future challenges.

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

[104] Jansen, Y., Dragicevic, P., Isenberg, P., Alexander, J., Karnik, A., Kildal, J., Subramanian, S., & Hornbæk, K. (2015). Opportunities and challenges for data physicalization. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15 (pp. 3227–3236). New York, NY, USA: ACM.

[105] Jo, J., L'Yi, S., Lee, B., & Seo, J. (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 (pp. 2660–2671). New York, NY, USA: ACM.

[106] Kaminski, J., Schober, M., Albaladejo, R., Zastupailo, O., & Hidalgo, C. (2012). Moviegalaxies - social networks in movies.

[107] Kang, J., Binda, J., Agarwal, P., Saconi, B., & Choe, E. K. (2017). Fostering user engagement: Improving sense of identity through cosmetic customization in wearable trackers. In *Proceedings of the 11th EAI International Conference on Pervasive Computing Technologies for Healthcare*, PervasiveHealth '17 (pp. 11–20). New York, NY, USA: ACM.

[108] Kazman, R. & Carriere, J. (1996). Rapid prototyping of information visualizations using vanish. In *Proc. of IEEE InfoVis* (pp. 21–28).

[109] Kenny, R. (2004). *Teaching TV production in a digital world: Integrating media literacy*. Libraries Unlimited.

[110] Kim, N. W., Bach, B., Im, H., Schriber, S., Gross, M., & Pfister, H. (2018). Visualizing nonlinear narratives with story curves. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis)*, 24(1), 595–604. http://doi.org/10.1109/TVCG.2017.2744118.

[111] Kim, N. W., Card, S. K., & Heer, J. (2010). Tracing genealogical data with timenets. In *Proceedings of the International Conference on Advanced Visual Interfaces*, AVI '10 (pp. 241–248). New York, NY, USA: ACM.

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

[113] Kim, Y.-H., Jeon, J. H., Lee, B., Choe, E. K., & Seo, J. (2017b). Omnitrack: A flexible self-tracking approach leveraging semi-automated tracking. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 1(3), 67:1–67:28.

[114] Kosara, R. (2016). Presentation-oriented visualization techniques. *IEEE Computer Graphics and Applications*, 36(1), 80–85.

[115] Kosara, R. & Mackinlay, J. (2013). Storytelling: The next step for visualization. *Computer*, 46(5), 44–50.

[116] Krause, J. (2018). Bubble sets. https://github.com/JosuaKrause/bubblesets-js. [Online; accessed 31-03-2018].

[117] Lab, G. C. (2018). Autodraw. https://www.autodraw.com/. [Online; accessed 16-Sep-2018].

[118] Lau, A. & Moere, A. V. (2007). Towards a model of information aesthetics in information visualization. In *2007 11th International Conference Information Visualization (IV'07)* (pp. 87–92).: IEEE.

[119] Lee, B., Isenberg, P., Riche, N. H., & Carpendale, S. (2012). Beyond mouse and keyboard: Expanding design considerations for information visualization interactions. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis)*, 18(12), 2689–2698. https://doi.org/10.1109/TVCG.2012.204.

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

[121] Lee, B., Kazi, R. H., & Smith, G. (2013b). SketchStory: Telling more engaging stories with data through freeform sketching. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis)*, 19(12), 2416–2425. http://doi.org/10.1109/TVCG.2013.191.

[122] Lee, B., Riche, N. H., Isenberg, P., & Carpendale, S. (2015). More than telling a story: Transforming data into visually shared stories. *IEEE Computer Graphics and Applications*, 35(5), 84–90. http://doi.org/10.1109/MCG.2015.99.

[123] Lee, S., Kim, S. H., & Kwon, B. C. (2017). Vlat: Development of a visualization literacy assessment test. *IEEE Transactions on Visualization and Computer Graphics*, 23(1), 551–560.

[124] Lehni, J. & Puckey, J. (2018). Paper.js: The swiss army knife of vector graphics scripting. http://paperjs.org/. [Online; accessed 16-Sep-2018].

[125] Li, I., Dey, A., & Forlizzi, J. (2010). A stage-based model of personal informatics systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10 (pp. 557–566). New York, NY, USA: ACM.

[126] Li, I., Dey, A. K., & Forlizzi, J. (2011). Understanding my data, myself: Supporting self-reflection with ubicomp technologies. In *Proceedings of the 13th International Conference on Ubiquitous Computing*, UbiComp '11 (pp. 405–414). New York, NY, USA: ACM.

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

[128] Liu, S., Wu, Y., Wei, E., Liu, M., & Liu, Y. (2013). Storyflow: Tracking the evolution of stories. *IEEE Transactions on Visualization and Computer Graphics*, 19(12), 2436–2445.

[129] Liu, Z., Thompson, J., Wilson, A., Dontcheva, M., Delorey, J., Grigg, S., Kerr, B., & Stasko, J. (2018). Data illustrator: Augmenting vector design tools with lazy data binding for expressive visualization authoring. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18 (pp. 123:1–123:13). New York, NY, USA: ACM.

[130] Loria, S. (2014). Textblob: simplified text processing. *Secondary TextBlob: Simplified Text Processing*.

[131] Lupi, G. (2017). Data humanism, the revolution will be visualized. http://www.printmag.com/information-design/data-humanism-future-of-data-visualization/.

[132] Lupi, G. (2018). A data wallpaper for story: soft and small data to capture people's vision for the future of work. http://giorgialupi.com/work/#/collaborative-data-wallpaper-for-story/.

[133] Lupi, G. & Accurat (2017). Data portraits at ted 2017. https://www.accurat.it/works/ted/.

[134] Lupi, G. & Posavec, S. (2016). *Dear Data*. Chronicle Books.

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

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

[137] Mani, I. (2012). Computational modeling of narrative. *Synthesis Lectures on Human Language Technologies*, 5(3), 1–142.

[138] McCloud, S. (1993). *Understanding Comics: The Invisible Art*. Harper Collins.

[139] McCurdy, N., Lein, J., Coles, K., & Meyer, M. (2016). Poemage: Visualizing the sonic topology of a poem. *IEEE Transactions on Visualization and Computer Graphics*, 22(1), 439–448.

[140] McKenna, S., Riche, N. H., Lee, B., Boy, J., & Meyer, M. (2017). Visual narrative flow: Exploring factors shaping data visualization story reading experiences. *Computer Graphics Forum (Proceedings of EuroVis)*, 36(3), 377–387. http://doi.org/10.1111/cgf.13195.

[141] Meidiana, A. & Hong, S.-H. (2015). Multistory: Visual analytics of dynamic multi-relational networks. In *2015 IEEE Pacific Visualization Symposium (PacificVis)* (pp. 75–79).

[142] Meyer, M., Munzner, T., & Pfister, H. (2009). Mizbee: A multiscale synteny browser. *IEEE Transactions on Visualization and Computer Graphics*, 15(6), 897–904.

[143] Microsoft (2018). Use bookmarks to share insights and build stories in power bi. https://docs.microsoft.com/en-us/power-bi/desktop-bookmarks. [Online; accessed 13-09-2018].

[144] Missive (2018). Emoji mart. https://missive.github.io/emoji-mart/. [Online; accessed 16-Sep-2018].

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

[146] MongoDB, I. (2018). Mongodb. https://www.mongodb.com/. [Online; accessed 16-Sep-2018].

[147] Montfort, N. (2007). Ordering events in interactive fiction narratives. In *Proceedings of the AAAI Fall Symposium on Intelligent Narrative Technologies* (pp. 87–94).

[148] Morris, M. R., Wobbrock, J. O., & Wilson, A. D. (2010). Understanding users' preferences for surface gestures. In *Proceedings of Graphics Interface (GI)* (pp. 261–268). http://dl.acm.org/citation.cfm?id=1839214.1839260.

[149] Myers, B. A., Goldstein, J., & Goldberg, M. A. (1994). Creating charts by demonstration. In *Proc. of CHI* (pp. 106–111).: ACM.

[150] Na, J.-C., Thet, T. T., Khoo, C. S., & Kyaing, W. Y. M. (2011). Visual sentiment summarization of movie reviews. In *International Conference on Asian Digital Libraries* (pp. 277–287).: Springer.

[151] Oliveira, N., Jun, E., Croxson, T., Gajos, K. Z., & Reinecke, K. (2017). Labinthewild: How to design uncompensated, feedback-driven online experiments. In *Companion of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, CSCW '17 Companion (pp. 25–28). New York, NY, USA: ACM.

[152] Pavel, A., Goldman, D. B., Hartmann, B., & Agrawala, M. (2015). Sceneskim: Searching and browsing movies using synchronized captions, scripts and plot summaries. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (pp. 181–190).: ACM.

[153] Pfeuffer, K., Hinckley, K., Pahud, M., & Buxton, B. (2017). Thumb + pen interaction on tablets. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)* (pp. 3254–3266). http://doi.org/10.1145/3025453.3025567.

[154] Pousman, Z., Stasko, J., & Mateas, M. (2007). Casual information visualization: Depictions of data in everyday life. *IEEE Transactions on Visualization and Computer Graphics*, 13(6), 1145–1152.

[155] Prince, G. (1974). *A grammar of stories: An introduction*, volume 13. Walter de Gruyter.

[156] Propp, V. (2010). *Morphology of the Folktale*, volume 9. University of Texas Press.

[157] Qiang, L., Bingjie, C., & Haibo, Z. (2017). Storytelling by the storycake visualization. *The Visual Computer*.

[158] Qu, Z. & Hullman, J. (2018). Keeping multiple views consistent: Constraints, validations, and exceptions in visualization authoring. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis)*, 24(1), 468–477. http://doi.org/10.1109/TVCG.2017.2744198.

[159] Ren, D., Brehmer, M., Lee, B., Höllerer, T., & Choe, E. K. (2017). ChartAccent: Annotation for data-driven storytelling. In *Proceedings of the IEEE Pacific Visualization Symposium (PacificVis)* (pp. 230–239). http://doi.org/10.1109/PACIFICVIS.2017.8031599.

[160] Ren, D., Höllerer, T., & Yuan, X. (2014). iVisDesigner: Expressive interactive design of information visualizations. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis)*, 20(12), 2092–2101. http://doi.org/10.1109/TVCG.2014.2346291.

[161] Ren, D., Lee, B., & Brehmer, M. (2018a). Charticulator: Interactive construction of bespoke chart layouts. *IEEE Transactions on Visualization and Computer Graphics*, (pp. 1–1).

[162] Ren, D., Lee, B., Brehmer, M., & Riche, N. H. (2018b). Reflecting on the evaluation of visualization authoring systems. In *Workshop Proceedings of Evaluation and Beyond - Methodological Approaches for Visualization (BELIV)*. https://aka.ms/renbeliv18.

[163] Richardson, B. (2002a). Beyond story and discourse: Narrative time in postmodern and nonmimetic fiction. *Narrative dynamics: Essays on time, plot, closure, and frames*, (pp. 47–63).

[164] Richardson, B. (2002b). *Narrative dynamics: essays on time, plot, closure, and frames*. Ohio State University Press.

[165] Riche, N. H., Hurter, C., Carpendale, S., & Diakopoulos, N., Eds. (2018). *Data-Driven Storytelling*. AK Peters. CRC Press. https://aka.ms/dds_book.

[166] Robertson, G., Fernandez, R., Fisher, D., Lee, B., & Stasko, J. (2008). Effectiveness of animation in trend visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6).

[167] Ronacher, A. (2018). Flask: A microframework for python. http://flask.pocoo.org/. [Online; accessed 16-Sep-2018].

[168] Roth, S. F., Kolojejchick, J., Mattis, J., & Goldstein, J. (1994). Interactive graphic design using automatic presentation knowledge. In *Proc. of CHI* (pp. 112–117).: ACM.

[169] Rule, A., Tabard, A., & Hollan, J. D. (2018). Exploration and explanation in computational notebooks. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18 (pp. 32:1–32:12). New York, NY, USA: ACM.

[170] Satyanarayan, A. & Heer, J. (2014a). Authoring narrative bisualizations with Ellipsis. *Computer Graphics Forum (Proceedings of EuroVis)*, 33(3), 361–370. http://doi.org/10.1111/cgf.12392.

[171] Satyanarayan, A. & Heer, J. (2014b). Lyra: An interactive visualization design environment. *Computer Graphics Forum (Proceedings of EuroVis)*, 33(3), 351–360. http://doi.org/10.1111/cgf.12391.

[172] Schneider, H., Schauer, K., Stachl, C., & Butz, A. (2017). Your data, your vis: Personalizing personal data visualizations. In *IFIP Conference on Human-Computer Interaction* (pp. 374–392).: Springer.

[173] Segel, E. & Heer, J. (2010). Narrative visualization: Telling stories with data. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis)*, 16(6), 1139–1148. https://doi.org/10.1109/TVCG.2010.179.

[174] Sharma, R. & Rajamanickam, V. (2013). Using interactive data visualization to explore non-linear movie narratives. *Parsons Journal for Information Mapping*.

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

[176] Sicat, R., Li, J., Choi, J., Cordeil, M., Jeong, W.-K., Bach, B., & Pfister, H. (2019). Dxr: A toolkit for building immersive data visualizations. *IEEE transactions on visualization and computer graphics*, 25(1), 715–725.

[177] Sinclair, S. & Ruecker, S. (2006). The digital play book: An environment for interacting with play scripts. *Canadian Theatre Review*, 127, 38–41.

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

[179] Spritzer, A. S., Boy, J., Dragicevic, P., Fekete, J.-D., & Freitas, C. M. D. S. (2015). Towards a smooth design process for static communicative node-link diagrams. *Computer Graphics Forum (Proceedings of EuroVis)*, 34(3), 461–470. http://doi.org/10.1111/cgf.12658.

[180] Stolper, C. D., Lee, B., Henry Riche, N., & Stasko, J. (2016). *Emerging and Recurring Data-Driven Storytelling Techniques: Analysis of a Curated Collection of Recent Stories*. Technical report. https://www.microsoft.com/en-us/research/publication/emerging-and-recurring-data-driven-storytelling-techniques-analysis-of-a-curated-collection-of-rec

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

[182] Strobelt, H., Alsallakh, B., Botros, J., Peterson, B., Borowsky, M., Pfister, H., & Lex, A. (2016). Vials: visualizing alternative splicing of genes. *IEEE transactions on visualization and computer graphics*, 22(1), 399–408.

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

[184] Tableau (2018). Story Points. https://tabsoft.co/2jxghOC. [Online; accessed 13-09-2018].

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

[186] Tanahashi, Y. & Ma, K.-L. (2012). Design considerations for optimizing storyline visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 18(12), 2679–2688.

[187] Tapaswi, M., Bauml, M., & Stiefelhagen, R. (2014). Storygraphs: visualizing character interactions as a timeline. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 827–834).

[188] Thudt, A., Hinrichs, U., Huron, S., & Carpendale, S. (2018). Self-reflection and personal physicalization construction. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18 (pp. 154:1–154:13). New York, NY, USA: ACM.

[189] Thudt, A., Perin, C., Willett, W., & Carpendale, S. (2017). Subjectivity in personal storytelling with visualization. *Information Design Journal*, 23(1), 48–64.

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

[191] Viégas, F. B. & Wattenberg, M. (2007). Artistic data visualization: Beyond visual analytics. In *Proceedings of the 2Nd International Conference on Online Communities and Social Computing*, OCSC'07 (pp. 182–191). Berlin, Heidelberg: Springer-Verlag.

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

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

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

[195] Walny, J., Huron, S., Perin, C., Wun, T., Pusch, R., & Carpendale, S. (2018). Active reading of visualizations. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis)*, 24(1), 770–780. http://doi.org/10.1109/TVCG.2017.2745958.

[196] Wang, S., Tanahashi, Y., Leaf, N., & Ma, K.-L. (2015). Design and effects of personal visualizations. *IEEE Computer Graphics and Applications*, 35(4), 82–93.

[197] Wang, Z., Wang, S., Farinella, M., Murray-Rust, D., Riche, N. H., & Bach, B. (2019). Comparing effectiveness and engagement of datacomics and infographics. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*.

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

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

[200] Wilhelm, T., Burghardt, M., & Wolff, C. (2013). To see or not to see-an interactive tool for the visualization and analysis of shakespeare plays.

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

[202] Wu, Y., Pitipornvivat, N., Zhao, J., Yang, S., Huang, G., & Qu, H. (2016). egoslider: Visual analysis of egocentric network evolution. *IEEE Transactions on Visualization and Computer Graphics*, 22(1), 260–269.

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

[204] Xia, H., Araujo, B., Grossman, T., & Wigdor, D. (2016). Object-oriented drawing. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)* (pp. 4610–4621). http://doi.org/10.1145/2858036.2858075.

[205] Xia, H., Araujo, B., & Wigdor, D. (2017a). Collection objects: Enabling fluid formation and manipulation of aggregate selections. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)* (pp. 5592–5604). http://doi.org/10.1145/3025453.3025554.

[206] Xia, H., Hinckley, K., Pahud, M., Tu, X., & Buxton, B. (2017b). WritLarge: Ink unleashed by unified scope, action, and zoom. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)* (pp. 3227–3240). http://doi.org/10.1145/3025453.3025664.

[207] Xia, H., Riche, N. H., Chevalier, F., Araujo, B., & Wigdor, D. (2018). DataInk: Direct and creative data-oriented drawing. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*. https://doi.org/10.1145/3173574.3173797.

[208] Zgraggen, E., Zeleznik, R., & Drucker, S. M. (2014). PanoramicData: Data analysis through pen & touch. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis)*, 20(12), 2112–2121. http://doi.org/10.1109/TVCG.2014.2346293.

[209] Zhao, H., Plaisant, C., & Shneiderman, B. (2005). ”i hear the pattern”: Interactive sonification of geographical data patterns. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '05 (pp. 1905–1908). New York, NY, USA: ACM.

[210] Zhao, J., Glueck, M., Chevalier, F., Wu, Y., & Khan, A. (2016). Egocentric analysis of dynamic networks with egolines. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16 (pp. 5003–5014). New York, NY, USA: ACM.

[211] Zhao, Z., Marr, R., & Elmqvist, N. (2015). *Data Comics: Sequential Art for Data-Driven Storytelling*. Technical report, Human Computer Interaction Lab, University of Maryland. http://www.cs.umd.edu/hcil/trs/2015-15/2015-15.pdf.