

Narvis: Authoring Narrative Slideshows for Introducing Data Visualization Designs

Qianwen Wang, Zhen Li, Siwei Fu, Weiwei Cui, Huamin Qu

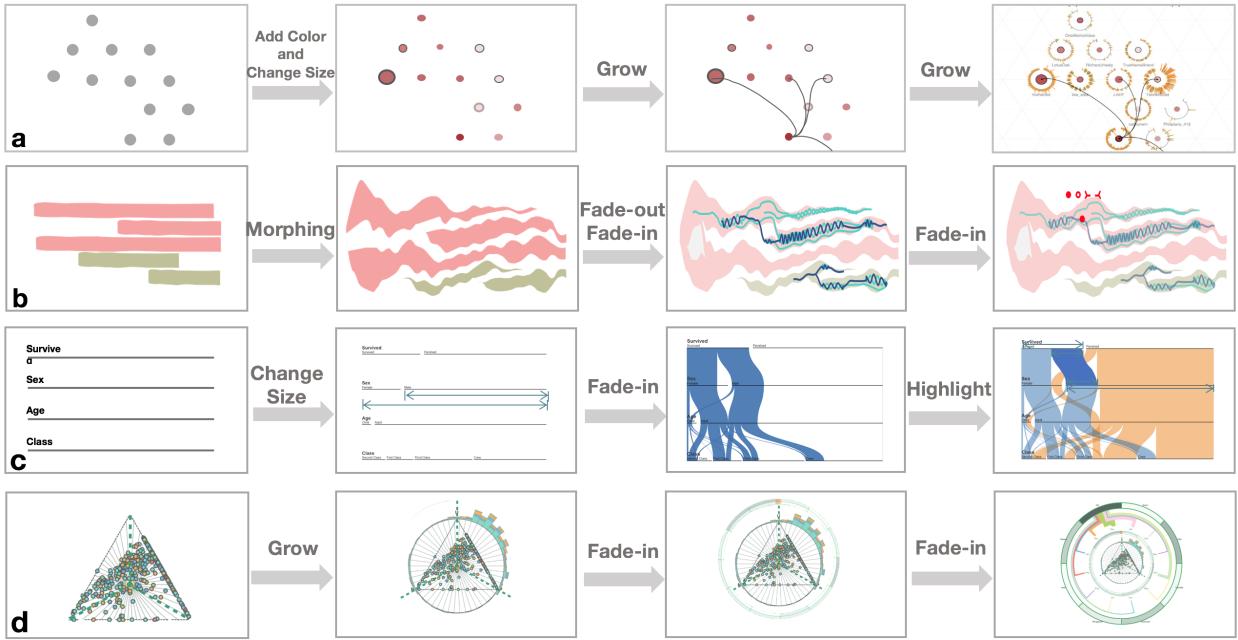


Fig. 1. Examples of the slideshows created by Narvis. Snapshot images illustrate the content of four slideshows that introduce four designs, (a) TargetVue [8], (b) Textflow [11], (c) Parallel Sets [13], and (d) OpinionSeer [41], respectively. They demonstrate how different visual channels (*i.e.*, color, size, shape, position) and visual units (*i.e.*, dots, lines, bars, flows) are added sequentially for a progressive introduction. These slideshows use seven types of animation templates provided by Narvis: fade-in, fade-out, growing, changing size, adding color, morphing, and highlighting.

Abstract— Visual designs can be complex in modern data visualization systems, which poses special challenges for explaining them to the non-experts. However, few if any presentation tools are tailored for this purpose. In this study, we present Narvis, a slideshow authoring tool designed for introducing data visualizations to non-experts. Narvis targets two types of end users: teachers, experts in data visualization who produce tutorials for explaining a data visualization, and students, non-experts who try to understand visualization designs through tutorials. We present an analysis of requirements through close discussions with the two types of end users. The resulting considerations guide the design and implementation of Narvis. Additionally, to help teachers better organize their introduction slideshows, we specify a data visualization as a hierarchical combination of components, which are automatically detected and extracted by Narvis. The teachers craft an introduction slideshow through first organizing these components, and then explaining them sequentially. A series of templates are provided for adding annotations and animations to improve efficiency during the authoring process. We evaluate Narvis through a qualitative analysis of the authoring experience, and a preliminary evaluation of the generated slideshows.

Index Terms—Education, Narrative Visualization, Authoring Tools.

1 INTRODUCTION

Data visualization is widely used in data presentation and analysis. While basic visualization methods (*e.g.*, line charts, bar charts, donut charts) have demonstrated their utility and effectiveness, they may

• Q. Wang, Z. Li, S. Fu, and H. Qu are with Hong Kong University of Science and Technology. Email: {qwangbb, zlidn, sfugg}@connect.ust.hk, huamin@cse.ust.hk.

• W. Cui is with Microsoft Research Asia. Email: Weiwei.Cui@microsoft.com.

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

fail to present complex data. For example, they can hardly give a comprehensive illustration of how the topics in social media split, merge and compete with each other.

Over the years, many advanced visualization methods have been devised for communicating and analyzing these complex data. However, these advanced visualizations are usually composed of numerous visual components, have diverse visual encodings, and thus are not intuitively understandable (*e.g.*, Fig. 1(a) and (d)). Consequently, these advanced visualization techniques haven't been widely exposed to the public, even though their utility has been verified by experts from various domains [40].

Imagine the following scenario: Jessica, an infovis teacher, wants to introduce a novel visualization technique described in an academic article in her class. The visual design is complex, as it is composed of many visual components and various visual encodings that the stu-

dents likely have never seen before. She anticipates they might have immediate questions, such as “*what do the links mean?*”, “*what do the positions of nodes mean?*”. So she decides to create a tutorial to guide her students through the different parts of the visualization to help them better understand it.

To create a tutorial, a common solution is to accompany a visual design with a textual description. For example, a whole section is dedicated to describing the visual encodings in the academic article. But learning through an extensive and monotonous textual description is seldom pleasant, and often inefficient. Videos, on the contrary, are relatively engaging, attractive, and comprehensible [42]. Like many other visualization papers, the academic article that Jessica wants to introduce has an accompanying video that demonstrates how to interpret the visual design. However, the pacing and the content of this video cannot be easily edited to fit the needs of the class. Reproducing a video from scratch is generally difficult. Slideshows are a favorable alternative. As with videos, slideshows reveal information progressively in an engaging and appealing manner while being easier to produce. Moreover, the pace of slideshows can be easily controlled by either the teacher or the students.

However, crafting a comprehensible and attractive tutorial slideshow can be challenging. First, to make the slideshow attractive and engaging, some teachers may apply graphic editing techniques such as “*highlighting*”, “*morphing*” and “*zooming in*”. However, these techniques can be tedious and time-consuming to achieve using existing tools. Second, due to “*the curse of knowledge*” [18], these teachers might unknowingly assume that their students have the background to understand and unconsciously omit the explanation for certain visual encodings and components. Third, collecting feedback from the students can also be difficult for teachers, which directly hinders the teachers from improving the quality of their tutorials.

Moreover, it is also challenging to make the slideshow tutorial comprehensible to the students. First, the students can easily get overwhelmed if they are inundated with all the information simultaneously. At the same time, an improper narrative sequence will confuse the students considering the logic dependence existing among visual components. For example, nodes in a node-link diagram should be introduced before the links connecting them. In an advanced visualization design, which has more components than merely nodes and links, identifying a proper explanation sequence can be more challenging and demanding. In addition, the students can be easily distracted and become lost, thereby preventing effective learning.

To tackle these challenges, we propose Narvis, a slideshow authoring tool that assists in introducing advanced visualization designs. Those who craft tutorials are the direct end users of Narvis. Meanwhile, it is important to take those who read these tutorials into account when designing a narrative data visualization [31]. Thus, two types of end users are identified for Narvis: teachers and students. We conduct interviews with these two groups of end users to understand their expectations, identify the problems in existing tools, and then distill the design requirements for Narvis. At the same time, to form a clear logic in the tutorial slideshows, we borrow lessons from previous work about the design space of a data visualization [28, 38], combine them with our empirical observations, and propose an approach to decompose a visualization design into several components and to introduce these components progressively. Based on the aforementioned analysis, we design and implement Narvis, aiming to offer an efficient, expressive and user-friendly authoring tool for introducing data visualization designs. We evaluate Narvis in terms of authoring experiences and the quality of generated slideshows.

Our contributions are as follows: 1) An analysis of the requirements for an authoring tool tailored for introducing a visualization design. These requirements are obtained from the perspectives of teachers and students. 2) An approach to hierarchically decompose a visualization design and introduce its compositions progressively. 3) The design and implementation of Narvis, a prototype authoring tool to generate and edit tutorial slideshows for introducing visualization designs. We believe our work can facilitate wider adoption of advanced visualization designs.

2 RELATED WORK

In this section, we discuss the prior studies that are closely related to our work.

2.1 Structures of Narrative Data Visualizations

Narrative is an effective tool for information communication [12] and has been widely studied in the fields of literature, comics, and cinema [10, 37]. With the increasing importance of data communication, researchers in the visualization community are studying how narrative can be used in data visualization [6, 27, 31, 39].

Some researchers borrowed the concepts from other fields to guide the design of narrative data visualization. For instance, Amini et al. [1] borrowed concepts about narrative categories from comics to analyze the structure of narrative data visualizations. Wang et al. [39] adopt two representative tactics, time-remapping and foreshadowing, from cinematographers to organize structures of narrative data visualization for better information communication. Some researchers focused on the narrative structures exclusively for data visualization. Satyanarayan and Heer [34], through interviews with professional journalists, defined the core abstractions of narrative data visualization as state-based scenes, visualization parameters, dynamic graphical and textual annotations, and interaction triggers. By identifying the change in data attributes, Hullman et al. [21] proposed a graph-driven approach to automatically identify effective narrative sequences for linearly presenting a set of visualizations. These works, however, mainly focus on conveying the insight discovering process and rarely discuss the narrative structures used for explaining visualization designs.

Maybe the closest to our work is the narrative introduction proposed in the work of Boy et al. [6], which provides initial insights and introduces different visual encodings for three online data visualizations. The authors found that these narrative introductions, even though increased uptime and number of visits, didn’t increase user-engagement in exploration. However, they focused on studying the effect of narrative introductions on user-engagement and paid little attention to analyzing the proper structure of a narrative introduction.

Compared with previous studies, we aim to help people identify a proper narrative sequence for introducing the visual encoding of a visualization design.

2.2 Decomposing Data Visualizations

The clarification of the design space of a data visualization design can help people get a better understanding of this design. Munzner [28] proposed that a data visualization design “*can be described as an orthogonal combination of two aspects: graphical elements called marks and visual channels to control their appearance*”. Borrowing the concept of physical building blocks such as Lego, Huron et al. [22] extended the design space of a data visualization, defining the components of a data visualization as a token, token grammar, environment and assembly model. Javed et al. [24] focused on a high-level structure, exploring the composition of multiple visualization views. These theoretical works motivate the designers of visualization tools to develop efficient high-level visualization systems [5, 29].

On the other hand, theoretically identifying the basic components of a data visualization enables people to physically extract them. Harper and Agrawala [17] contributed a tool that extracts visual variables from existing online visualization designs to generate a new design. Huang et al. [20] proposed a system that recognizes and interprets imaged infographics from a scanned document. ReVision [35] applied computer vision methods to recognize the types, marks, encodings of a data visualization, and allows the users to create a new design based on these data.

Narvis is inspired by previous theoretical analysis. However, instead of decomposing a current design and mapping it to an alternative one, Narvis extracts the basic compositions of a visual design for introducing this design progressively and structurally.

2.3 Authoring Tools for Narrative Visualizations

The extensive requirements of data communication have motivated researchers to investigate techniques for authoring narrative visualization.

User experience represents a vital concern when utilizing an authoring tool. SketchStory [25], with its free-form sketching of interaction, provides an engaging approach to creating and presenting narrative visualizations. DataClips [2] lowers the barrier of crafting a narrative visualization by providing a library of data clips, thereby allowing non-experts to be involved in producing narrative visualizations. Lyra [33] allows designers to author expressive visualization designs through drag-and-drop interactions without writing code.

Information delivery is the core consideration of an authoring tool. Existing authoring tools typically select a specific type of narrative visualization based on the information type [2, 16]. Meanwhile, integrating an authoring tool for narrative visualization with a data analysis tool is becoming a trend since it effectively bridges the gap between data analysis and communication [7, 15, 26].

These tools offer inspiring user interaction design and provide favorable examples for implementing narrative visualization. However, these tools treat visual encoding as a cognitively obvious attribute that can be universally recognized without a formal introduction. Thus, these tools are ill-suited for introducing visualization designs. Contrary to the aforementioned tools, Narvis enables people to produce narrative tutorial slideshows, which introduce visualization designs by decomposing a visualization design and explaining their visual components progressively .

3 UNDERSTANDING END USERS

To better guide the design and implementation of Narvis, we conducted interviews with two groups of end users (*i.e.*, teachers and students) to investigate their current practice of producing tutorials and understanding visualizations. Six design requirements for Narvis were derived from our interviews.

3.1 Users

To investigate the current practice of making tutorials and the experience of reading tutorials, we collaborated with the two types of end users of Narvis, *i.e.*, teachers who craft tutorials and students who read tutorials. In this study, we assume that teachers have extensive experience in visualization designs and students have little prior knowledge of data visualization. The editor group consisted of two teaching assistants (TAs) in a data visualization course, one professor in the field of data visualization, and two employees from a commercial data visualization website. All the five teachers, denoted as T1 to T5, have the needs to produce tutorials for introducing visualization designs. A slideshow is their preferred type because it is “*more appealing and comprehensible than textual descriptions and easy to edit than videos*” (T2).

The students group consisted of seven undergraduate students, denoted as S1 to S7, from three departments (*i.e.*, biology, finance, computer science). All these students had less than three-months experience in data visualization. They recently took a data visualization course, where they were required to survey visual designs for their assignments. Thus they learned many visual designs from the internet and watched many tutorials.

3.2 Interviews

We conducted two-part, semi-structured interviews with teachers and students separately.

In the first part, each teacher described their recent experience in producing an introduction slideshow. We encouraged them to open this slideshow and recall as many details as possible. Then, we inquired more about the organization of the narrative sequence in their tutorials, such as “*why the encoding of color is introduced before the encoding of position?*” In the second part, teachers enumerated all the tools they have ever used, identified the most frequently-used tool, and provided their reasons for their choice. Then, we asked them about the obstacles they encountered when using this tool, and how they overcame these obstacles. The entire interview lasted approximately 40 minutes for each participant.

In the interview with the students, we discussed with them the tutorials they had learned from. We first asked their general comments about the different types of tutorials (*e.g.*, online blogs, course lectures,

papers, blogs). Then each student identified one positive example and one negative example they encountered.

These tutorials are mainly about how to read a visualization design as a combination of graphics. For example, they regarded the encoding explanation (0:52-1:35) in a video ¹ and the encoding explanation in a blog² as positive examples that are “*comprehensible and helpful*”.

We discussed example tutorials with them, listened to their comments, and identified the obstacles toward their understanding. The length of the interviews varied from person to person, lasting from 18 to 43 minutes.

We took notes and videos during interviews for later analysis.

3.3 Design Requirements

Narvis aims to help teachers produce narrative slideshows for introducing data visualization designs. Thus, support for common operations and guidance to avoid common mistakes should be provided in Narvis.

Based on our observations during the interviews, we categorized six design considerations for Narvis, denoted as R1 to R6.

R1. Enable Efficient and Expressive Graphic Editing. T1, T2 and T3 (two TAs and one professor) typically used generalized presentation tools (*e.g.*, PowerPoint, Keynote, Prezi) because of their high efficiency, although the graphic editing capacities of these tools are limited. “*Highlighting a subarea is effective while introducing a visual design, yet it can hardly be achieved in PowerPoint*”, commented by T1. Owing to the complexity of the operation, professional graphic editing software (*e.g.*, Illustrator, Photoshop) were only used for special cases, such as “*a demo influencing the next year’s funding*” (T3, the professor). T4 and T5, two employees from an online data visualization platform, preferred to use professional software. But the producing process was time-consuming, taking nearly “*two weeks to produce one tutorial*”. A gap remains between the efficiency of general tools and expressiveness of professional graphical tools.

R2. Avoid Unconscious Overlooking. T1, T2, and T4 all mentioned that they might unconsciously miss the explanation of certain visual encodings without a good preparation. Since an advanced visualization design usually consists of various visual components with varying visual encodings, it can be challenging to ensure that every visual component and every visual encoding are properly explained. Moreover, with extensive expertise in data visualization, teachers might treat certain visual encodings as self-evident and offer no additional explanation. However, the incompleteness of information impedes the quality of produced tutorials.

This phenomenon is also known as “*the curse of knowledge*” [18]. Well-informed agents (*e.g.*, experts in data visualization) typically assume the less-informed agents (*e.g.*, students with little prior knowledge of data visualization) have the background to understand.

R3. Collect Feedback. For teachers, communicating with their students can be difficult, especially when the student size is large or when the student is remote in time and space (*e.g.*, the student of an online course). “*It is hard to make sure that all people understand all the visual encodings correctly. The thought that they may interpret the data falsely always bothers me*”, remarked T4, an employee from an online data visualization platform.

However, teachers require student feedback to evaluate and improve their tutorials. All interviewees in the teachers group stated that they would show their produced slideshows to their friends or colleagues for quality evaluation and further revision. But “*such evaluation methods can be biased since my friends are already familiar with visual designs*” (T5), and “*it is hard to collect feedback in a wide range*” (T2).

R4. Avoid information overload. Six out of seven participants in the students group complained that they experienced information overload when reading tutorials, such as a paper that “*uses one even two pages to describe a visualization design*” or slideshows that “*put too many things in one slide*”. S3 stated that “*I just skip some parts when I am inundated with too much information*.”

¹<https://www.youtube.com/watch?v=XQ6xPkAZsPU>

²<http://www.dear-data.com/week-08-1/>

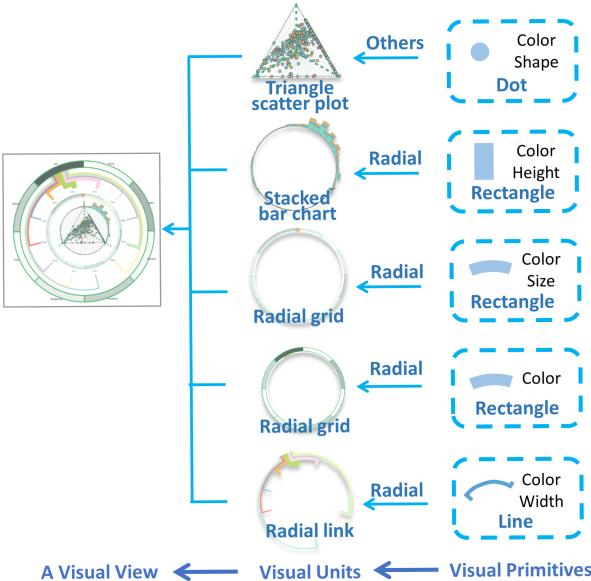


Fig. 2. After decomposing OpinionSeer [41], we obtain the hierarchical structure of this visualization. It consists of five visual units and employs three kinds of visual primitives.

Complex visual designs contain numerous visual components with varying visual encodings, thus imposing a cognitive burden on the students. To make a tutorial slideshow comprehensible, these visual components should be introduced progressively to avoid overloading the students.

R5. Provide Clear Narrative Logic. The interview with students revealed that many web-based data visualization systems and visualization designs were rarely accompanied by detailed, comprehensive tutorials. Students complained a lot about the lack of clear logic in these tutorials. “*Sometimes, I had to read a tutorial several times, reorganize all the information myself to fully understand a visual design*”, stated by S4. When commenting on a tutorial, S2 said, “*It first explains ‘A’, then it explains ‘B’, and suddenly it goes back to explaining ‘A’ again. Maybe it has its own logic, but I am totally confused.*”

Creating a tutorial slideshow with clear logic and informing students of this logic will facilitate their understanding of a visualization design.

R6. Keep the Sense of Overview. When receiving a large amount of information (e.g., a visualization design contains many visual components with varying visual encodings), the students can be easily distracted or forget previous information. In this situation, informing the students of the overall structure of a visualization design and reminding them of the previous messages can aid the perception process.

4 UNDERSTANDING DATA VISUALIZATION

According to the aforementioned design requirements, we realize that a comprehensive understanding of data visualizations is crucial for designing and implementing Narvis. Only by understanding a visualization design can Narvis know how to guide the teachers to form a clear narrative sequence (R2), assist them in making a progressive introduction (R4), prevent them from omitting certain visual encodings (R5), and help them keep the students concentrated (R6).

In this section, we aim to understand a visualization design by answering the following three questions: “*What are the basic components that compose a data visualization?*”, “*What is the relationship between these components?*”, “*How should we deal with these relationships when introducing a data visualization?*”

4.1 Components of a Visualization

Efforts have been made to identify the atomic building blocks of a visualization [3, 29, 38]. In this study, we extend the previous work by

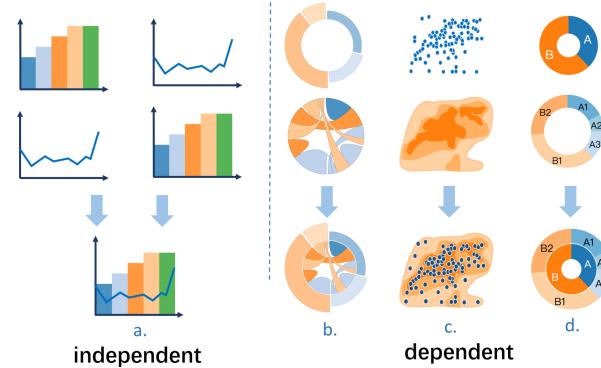


Fig. 3. Each visualization on the bottom is comprised of two visual units. The relationship between these two visual units can be independent (a), which means that the explanation order of these two units can be arbitrary, or dependent (b, c, and d), which means that one unit must be explained before another.

- 1) proposing a hierarchical structure for decomposing a visual design,
- 2) including the logic relationship between components. Here, we define a visualization as a single-view, static presentation of data.

In our model, a visualization is a hierarchical structure of three levels, namely, visual primitives, visual units, and visual views. Taking OpinionSeer [41] as an example, we apply the hierarchical model and decompose this design into five visual units, as depicted in Fig. 2.

A **Visual Primitive** is a graphical element whose visual channels, such as color, width, and height, are mapped to data attributes with certain visual grammars. Visual channels are visual properties that control the appearance of a graphical element, whereas a visual grammar describes the way a visual channel represents a data attribute. For instance, a point is a visual primitive, size is a visual channel, and “size indicates the importance score” is a visual grammar.

A **Visual Unit** is an assembly of visual primitives that are bound with the same data attributes. For example, one dot is a visual primitive, while the dots in a scatter plot constitute a visual unit. A visual unit is the smallest functional unit of a visualization. A visual primitive alone (e.g., a point) has no meaning, and it only has meaning as a part of a visual unit (e.g., an outlier point in a scatter plot).

A **Visual View** can be considered a combination of visual units. A simple visual view contains only one visual unit (e.g., the lines in a line chart) whereas an advanced visual view typically combines multiple visual units. For example, OpinionSeer [41] is composed of five visual units, as illustrated in Fig. 2.

4.2 Relationships between Components

We first describe the relationships between conceptual components, and then offer suggestions for constructing a narrative sequence based on these relationships.

4.2.1 Relationships between Visual Channels

For a visual primitive, various channels are encoded with different data attributes. Thus, these visual channels usually have no logical dependency between themselves. Determining a narrative sequence from their inner logical dependency is difficult.

Therefore, we define two metrics to arrange visual channels: **the complexity of their encoded information** and **saliency of their visual appearance**. In this study, “saliency” represents the degree of difficulty for people to notice a certain channel. The visual saliency of different channels is relatively constant and well-defined [9, 28].

These two metrics are used for the following reasons. First, the order of decreasing visual saliency can facilitate graphical perception [9]. Different channels have intrinsically different perceptual saliences and channel with high salience will suppress the expression of others. This salience strength can be influenced in a task-dependent manner [30]. After introducing the channel with high saliency first, we remove this

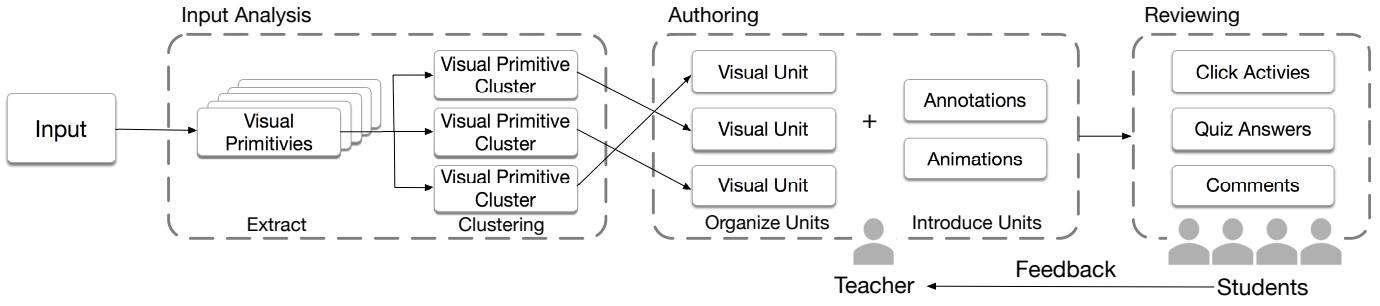


Fig. 4. Abstract representation of the workflow of Narvis, which consists of Input Analysis, Authoring, Reviewing. Two types of end users, namely, teachers and students, are involved. In Input Analysis, Narvis automatically detects visual primitives from the input and clusters them. In Authoring, teachers craft an introduction slideshow by organizing visual units and explaining them through animated transitions and annotations. In Reviewing, students watch the produced slideshow and offer feedback, which help the teachers refine the slideshows

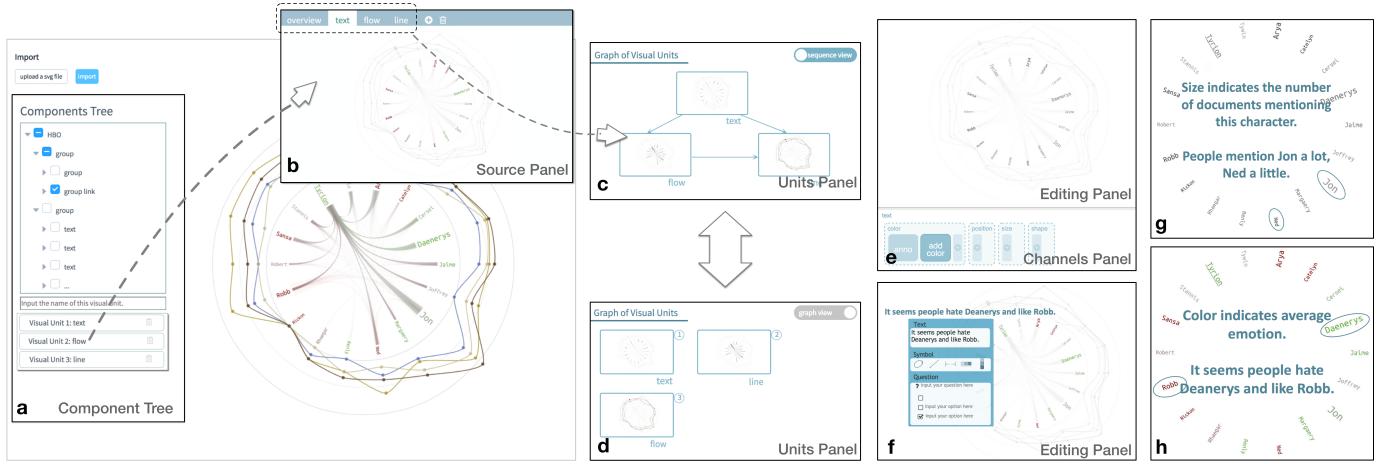


Fig. 5. The workflow of creating a slideshow with Narvis to introduce the visualization proposed by Scharl et al [36]. Teachers select visual units from Component Tree (a), identify the logical relationship between visual units (b) and get a narrative sequence (c). They then add animated transitions (e) and annotations (f) for explaining the visual encodings in these visual units. (g) and (h) are examples of produced slides.

channel from the task list in our mind [23], decrease its saliency and allow other channels an opportunity to attract the limited human attention.

Second, the order of increasing complexity leads to an effective learning process. The easy-to-difficulty procedure has been confirmed as effective for learning new tasks [4].

4.2.2 Relationships between Visual Primitives

Visual Primitives assemble to form a visual unit by following different construction rules. For example, dots can constitute a scatter plot, a spiral dot chart, or a circle packing chart by following radial, orthogonal, or metric-based construction rules, respectively [?].

4.2.3 Relationships between Visual Units

A visual view can be specified as the combination of several visual units. We identify two types of relationships between visual units: dependent and independent relationships.

Independent relationship refers to a relationship that no logic dependency exists between two visual units. The two visual units should be explained together but the sequence can be arbitrary. For instance, a unit of “line”, which indicates the temperature over a time period, and a unit of “bar”, which denotes the precipitation over a time period, are jointly placed in one visual view, sharing the same x-axis (Fig. 3(a)). The relationship between the two units is independent. In this situation, the unit “lines” can be introduced before or after another unit without impeding understanding.

Dependent relationship refers to a relationship wherein one visual unit “A” depends on another visual unit “B”. The understanding vi-

sual unit “B” is the prerequisite of understanding “A”, thus “B” should be explained before “A”. For example, in Fig. 3 (b), a unit of “flows” represents the correlation between “bars”. The students need to understand the meaning of “bars”; thus, flows among these “bars” can be meaningful.

5 NARVIS: A SLIDESHOW AUTHORIZING TOOL

Guided by the design requirements discussed in Section 3, as well as the theoretical model discussed in Section 4, we design and implement Narvis, a slideshow authoring tool for the introduction of data visualization designs. The workflow of Narvis consists of three phases (Fig. 4), i.e., Input Analysis Phase, Authoring Phase, and Viewing Phase.

5.1 Phase1: Input Analysis

In the Input Analysis Phase, Narvis processes a visualization design, extracts the graphic elements from the visual design, and classifies these elements into groups for further editing, as presented in Fig. 7.

Narvis uses a scalable vector graphics (SVG) file as input because SVG is employed in a wide range of data visualizations, and provides a complete scene graph for describing two-dimensional based vector graphics. A SVG file describes a scene graph through two types of elements: a) shape elements such as `<circle>`, `<rect>`, and `<path>`, which create basic shape elements on screen; b) group elements `<g>` which are the containers used to group shape elements. Each shape element has attributes (e.g., fill, stroke) that define its visual appearance and attributes (e.g., class, id) that describe its function.

Narvis processes a SVG file by extracting its shape elements and their attributes. Since shape elements are essential components that define a

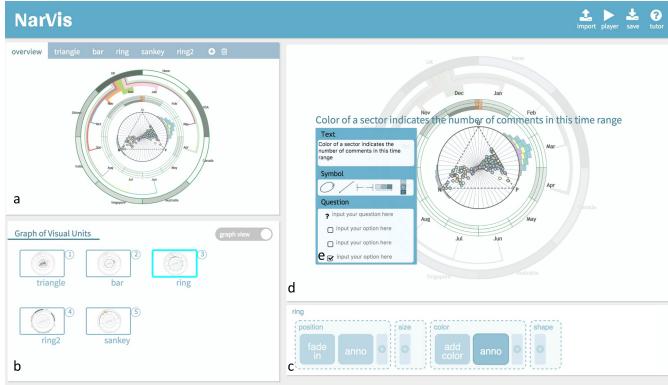


Fig. 6. Annotated screenshot of the interface of Narvis: a) Source Panel where Narvis process the input data, b) Units Panel where teachers define the relationships between units and obtain a narrative sequence, c) Channels Panel where teachers add annotations and animated transitions for the explanation of each unit, d) floating annotation window that offers options for adding annotations, e) Editing Panel where teachers modify the added animations and annotations.

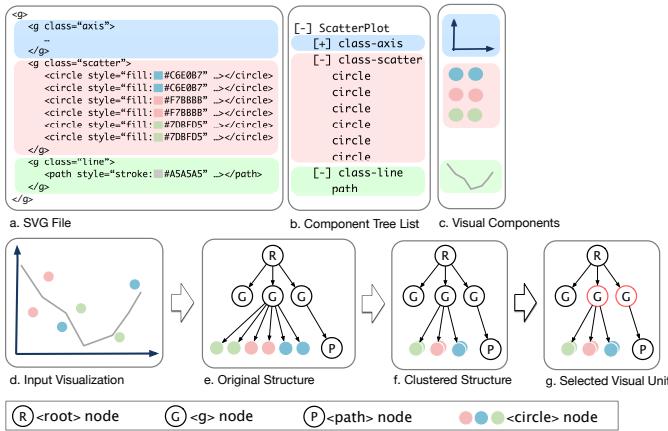


Fig. 7. Narvis automatically detects and clusters the graphic elements (c) in an SVG file (a) and organizes them in the form of an SVG tree (d-g). Teachers can examine, select, and organize needed visual units from a checkable list (b).

SVG, Narvis is able to parse any SVG file regardless of its generator. The extracted shape elements are designated as visual primitives. They are grouped based on, firstly, their original groups (*i.e.*, `<g>`) and classes, secondly, their element types (*e.g.*, `<circle>`, `<rect>`), and thirdly, their visual appearance (*e.g.*, color). It's worth noting that original groups and classes don't necessarily exist in a SVG file. After clustering, we get a hierarchical structure of these visual primitives.

To identify the visual units in a visualization design, teachers manipulate on the tree list in Components Tree View, as shown in Fig. 5(a).

This tree list depicts extracted visual primitives and their hierarchical structures (Fig. 7). Teachers identify visual units by selecting root nodes of subtrees in the tree list. All leaf nodes (*i.e.*, visual primitives) of a selected subtree then form a visual unit. In case teachers are not satisfied with the automatic clustering results, they are allowed to modify this tree list, including split, merge, and remove nodes. Besides, hovering over a node in the tree list will highlight all visual primitives that are descendants of this node, which displays at the right side of the tree list (Fig. 5(a)).

These visual units are displayed at the **Source Panel**, where each tabbed panel contains a cluster of visual primitives and acts as a visual unit (Fig. 5(b)). Each tabbed panel can be renamed by the teachers to facilitate further authoring process.

Through input analysis, teachers identify the visual units of the input

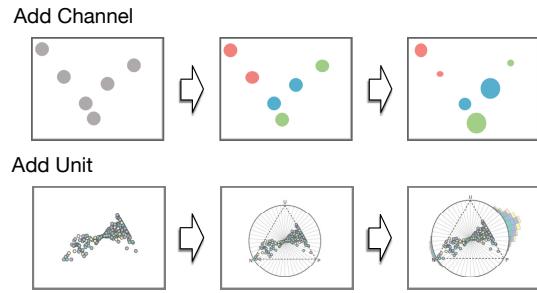


Fig. 8. Demonstration of the progressive introduction of visual channels (top) and visual units (bottom).

visualization design. Graphic elements belonging to the same visual unit are bound together and can be edited together during explanation (R1). Thus, Narvis allows the teachers an efficient and structural manipulation in the succeeding Authoring Phase.

5.2 Phase2: Authoring

In the Authoring Phase, teachers create an introduction slideshow by manipulating the visual units extracted in the Input Analysis Phase. In this section, we demonstrate the workflow of the Authoring Phase, which comprises two steps, namely, organizing units and introducing units (Fig. 4).

5.2.1 Organizing Units

The teachers define the relationships among the visual units, and then organize a narrative sequence for introducing these units on the basis of the defined relationships. In the generated slideshow tutorial, the visual units are added to the scene one by one following this sequence, preventing the addition of too much information at one step (R4).

This step is conducted in **Units Panel**, which consists of two views, a node-link view and a sequence view (Fig. 5(c) and (d)). In the node-link view, teachers define relationships between visual units, which can be independent (linked with a double arrow line), dependent (linked with an arrow line) or undefined (no link) (Section 4.2.3). In the sequence view, Narvis suggests one narrative sequence on the basis of these relationships using a topological sort algorithm (R2). Teachers are allowed to adjust this sequence as long as the adjustment doesn't conflict with the previously defined relationships. This sequence can be inserted in the generated slideshow to inform the students of the overall structure (R6).

5.2.2 Introducing Units

After determining the narrative sequence, teachers craft slides for the introduction of each unit.

Narvis allows teachers to add annotations and animated transitions for crafting a narrative introduction. Annotation is a common and important technique used in narrative data visualization [27, 31]. The effect of animated transition on improving perception and facilitating learning processes has been discussed by previous research [14, 19, 32].

Two panels, *i.e.*, **Channels Panel** and **Editing Panel**, are involved in this step.

Channels Panel (Fig. 6(c)) displays possible channels for a unit, once it is selected in the Units Panel (Fig. 6(b)).

These possible channels are detected by the analysis of the attributes of visual primitives in a visual unit. For example, if visual primitives in a visual unit have different colors, color will be detected as a possible visual channel that needs to be explained.

We explicitly enumerate the possible visual channels, instead of requiring the teachers manually add them one by one. We believe that this mechanism can reduce, if not eliminate, the teachers' overlooking of some important visual encodings (R5). By default, these channels are arranged in the decreasing order of visual salience. Teachers can freely change the channels' order (*e.g.*, explain simple encodings first), add and delete visual channels in the Channel Panel. To explain a

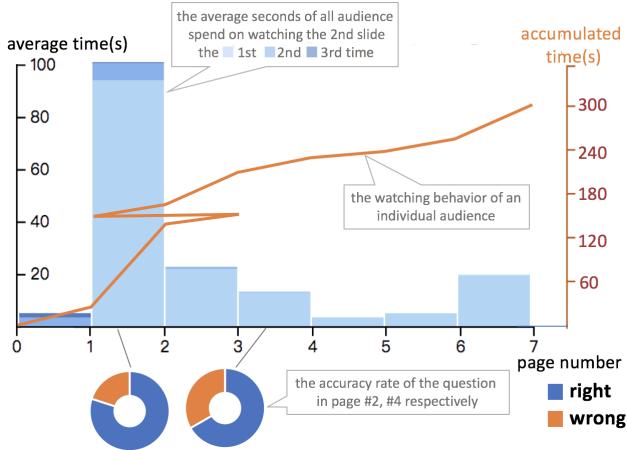


Fig. 9. Narvis visualizes the feedback data. The stacked bar chart describes the average watching time for different slides. The line chart indicates the watching behavior of an individual student. The donut chart represents the accuracy rate of student's answers if any question is inserted in the slideshow by the editor.

certain channel, teachers are required to add animated transitions and annotations. Added animated transitions and annotations appear as tabs in the corresponding dotted box of the corresponding channel (Fig. 6(c)).

Editing Panel displays an annotation or an animated transition (Fig. 6(d)), once it is selected in the **Channels Panel**. In **Editing Panel**, the teachers preview these annotations and animated transitions, and then perform modifications such as resizing and moving symbol annotations, revising text annotations, and restricting an animated transition to specific primitives.

For an efficient authoring process, Narvis provides templates for adding annotations and animated transitions. We propose these templates based on previous study [27, 31] and iterate over the design of these templates based on close discussions with three teachers mentioned in Section 3 (T1, T3, T4). Currently, Narvis supports seven types of animated transitions: fade-in, fade-out, growing, changing size, adding color, morphing, and highlighting, five types of symbol-based annotations: color legends, circles, arrow lines, double arrow lines, freeform lines, and several text-based annotations for the explanation of different channels. Fig. 8 demonstrates the progressive introduction of visual channels and visual units using the animated transitions provided by Narvis. Teachers can also insert multiple/single choice questions in their slideshow tutorials by using the question annotation template. These questions can remind the students of the previously mentioned information (R6). Meanwhile, Narvis collects the student's answers to these questions, thereby helping the teachers evaluate their slideshows (R3).

5.3 Phase3: Viewing

The produced slideshow can be either saved locally as an HTML file, or saved in Narvis and watched online. For online viewing, Narvis collects the click activities of students (*e.g.*, when they click a button to start a new slide or revert to a previous slide), their comments for this tutorial, and their answers to inserted questions, if any.

The feedback data is visualized by Narvis in the form of a stacked bar chart, a line chart and donut charts (Fig. 9). The x-axis represents the page number of slides in the bar chart and the line chart. In the line chart, the y-axis represents the accumulated watching time of a particular student. By contrast, the y-axis in the bar chart denotes the average watching time overall students for different slides. Each bar is split into colored bar segments. The bottom bar segment illustrates the time spent watching the slide the first time. If the student reverts and watches the slide for a second time, then a bar segment with a dark color is placed above the previous one, and so on. The rate of accuracy

to each question is visualized as one donut chart linked with the slide containing the question.

The feedback data enables teachers to observe the students' behavior in watching the tutorial, check the students' understanding of the visualization design, thus generating ideas for later revisions (R3).

6 EVALUATION

In this section, we demonstrate the utility of Narvis through evaluating its authoring experience and the quality of the generated slideshows.

Narvis is tailored for creating introduction slideshows for visualization designs. To the best of our knowledge, there is no other software that is specified for the same purpose. A formal comparative study of Narvis with general presentation tools, such as Prezi and PowerPoint, would be biased, since these tools lack the graphic editing capabilities that Narvis has. Comparing Narvis with professional graphic editing tools, such as Adobe Illustrator and Adobe Photoshop, would also be unfair, since Narvis is tailored for introducing visual designs and provides more efficient operations. Thus, we believe that it is more meaningful to show the utility of Narvis in qualitative studies.

6.1 Study Design

We conducted two sessions, *i.e.*, Authoring Session and Viewing Session, to evaluate the authoring experience of Narvis and quality of the generated slideshows, respectively.

We recruited five participants as teachers (two females and three males) between 25 and 40 years old, denoted as PT1 to PT5, to produce introduction slideshows with Narvis. The five teachers were all researchers or postgraduate students with considerable experience in data visualization. We also sent emails to students in our university and recruited 20 volunteers (seven females and thirteen males) as students, denoted as PS1 to PS20, to evaluate the quality of the generated slideshow. The twenty students were undergraduate or postgraduate students between 18 and 30 years old, with no prior background in data visualization.

6.1.1 Material Preparation

The authoring experience of Narvis is examined by asking five PTs to produce tutorial slideshows using Narvis. We prepared a SVG file and a textual introduction in advance so that teachers can better focus on producing tutorial slideshows. We chose *TextFlow* [11] as the example for teachers to explain. Two types of materials were prepared: 1) a textual description that explains the visual encodings of *TextFlow*. This description is directly extracted from the original paper without any modification; 2) an SVG file generated by ourselves using D3.js. We used the same data as used in *TextFlow*, namely, 2980 articles related to “Obama” from January 25 to February 10 in Bing News³. The visualization is rendered as a web page. We zoomed into topic flows with split/merge patterns and got the visualization as shown in Fig. 10. We used the “inspect elements” function in Chrome DevTools⁴ to locate the SVG elements and the “copy elements” function to save them in a SVG file.

6.1.2 Authoring Session

The Authoring Session consisted of three phases: (1) **Sketching Phase**, (2) **Authoring Phase**, (3) **Feedback Phase**. PTs were asked to think aloud during the whole session. We were present in the room for the whole session to observe and take notes.

In the **Sketching Phase**, PTs first learned a new visual design, *TextFlow* proposed in [11], through reading the literature description extracted from the paper. This learning phase cost about 20 minutes for each PT. Then, they were asked to sketch ideas for introducing *TextFlow*. They were required to enumerate all visual encodings and encouraged to consider (i) conveying insights to people with less experience in data visualization, (ii) organizing a clear narrative structure, and (iii) providing additional annotations and animated transitions. This phase took about 15 minutes for each PT.

³<https://www.bing.com/news>

⁴<https://developers.google.com/web/tools/chrome-devtools/>

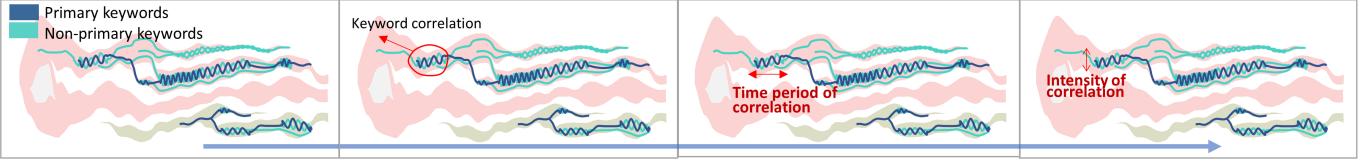


Fig. 10. Slides extracted from the slideshow created by PT3 with Narvis. These slides aim to explain two channels (*i.e.*, color and size) of the “lines” in *TextFlow* [11]

Table 1. Observation of five slideshows

	Slide 1	Slide 2	Slide 3	Slide 4	Slide 5
Types of Animated Transitions	Fade-in, High-highlight	N/A	Fade-in, High-highlight, Add-color	Fade-in, High-highlight, Morphing	Fade-in, Add-color
Number of Animated Transitions	2	0	4	4	3
Number of Symbol-based Annotation	6	7	8	6	9
Number of Text-based Annotation	6	2	8	6	3
Reading Time (S) (mean, standard deviation)	127.05, 22.03	156.78, 14.08	169.33, 21.49	128.84, 19.43	143.79, 17.21
Information Missing (in Slideshow /in Sketch)	1/2	0/1	0/2	0/0	0/0

Table 2. Overall evaluation of slideshows

	Readability	Utility	Aesthetics	Attractiveness	Quiz
Mean	4.00	4.15	3.95	4.05	4.475
Standard Deviation	0.56	0.37	0.60	0.68	0.47

In the **Authoring Phase**, PTs implemented the ideas in their sketches with Narvis. We first demonstrated the workflow of Narvis and allowed PTs to familiarize themselves with the Narvis using three basic visual designs, a chord diagram, a node-link diagram, and a parallel set. During Authoring, PTs were asked to speak out i) ideas in their sketches that they failed or found inconvenient to implement with Narvis; ii) ideas that occurred to them while authoring with Narvis. Examples of generated slideshows are demonstrated in Fig. 1 and Fig. 10.

In the **Feedback Phase**, PTs first filled out a questionnaire, which evaluated Narvis on a five-point Likert scale. Then, we asked these PTs open-ended questions to collect detailed comments and suggestions for Narvis.

6.1.3 Viewing Session

We ran a between-subject study in the Viewing Session. Each PS watched one slideshow, which was randomly picked from the slideshows produced in Authoring Session. After watching, PSs were asked to finish a quiz, with a full mark of five, to check their understanding of the design of *TextFlow*. Then, they accomplished a questionnaire to rate the quality of the slideshow from one (very poor) to five (excellent) regarding readability (*e.g.*, is it easy to read and follow the logic?), utility (*e.g.*, does it help you understand this visual design?), aesthetics (*e.g.*, does it look pretty and pleasant?) and attractiveness (*e.g.*, does it attract your interest?). We asked them open-ended questions to collect detailed feedback and explanations for their rating.

6.2 Results

Overall, participants are satisfied with the design of Narvis, regarding to the authoring experience of Narvis and the quality of the generated slideshows. Participants in the teachers group commented that Narvis “is well-designed”, “has clear UI”, and “is easy to operate”. In the questionnaire, which was based on a 5-point Likert scale ranging from strongly agree (5) to strongly disagree (1), they confirmed that the interaction is easy in general (mean 4, standard deviation 0) and they were able to craft a slideshow with Narvis after a short training period (4.8, 0.45). They commented that the produced slideshows were visually pleasing (5, 0) and they will use Narvis in the future (4.4, 0.55). We also evaluated the five slideshows generated in Authoring Session from the independent opinions of students. Overall, 20 students were satisfied with the quality of the produced slideshows, as shown in Table 2. The high scores they obtained in the subsequent quiz demonstrated that they were able to read the visualization design and obtain insights after watching the tutorials.

In the study, we have observed that the proposed design requirements (Section 3) are fulfilled to some extent. For example, **R1** and **R3** are met based on the results of the questionnaire and the quiz. For **R1**, PTs agreed Narvis can implement the ideas they have sketched (mean 4.8, standard deviation 0.45) (*i.e.*, expressive editing), reduce time and workload (4.8, 0.45) (*i.e.*, efficient editing). For **R3**, PTs appreciated the function of collecting feedback from the students, agreeing that it would help them revise and improve the introduction slideshow (4, 0.71).

All PTs (strongly) agreed that Narvis helped them clarify their logic (4.8, 0.45) (**R5**), and offered a clear overview of the design (4.5, 0.5) (**R6**). PSs also confirmed that the produced slides were easy to read (4, 0.56). Four PTs (PT1, PT2, PT3, PT5) clearly expressed their appreciation for these designs in their detailed comments. PT4 commented, “The Unit Panel helped me decide what should be introduced first and what should be introduced later.” PT5 commented, “This whole sequence organizer function is a refreshing idea. It alone can attract me to Narvis.”

The fulfillment of **R4** is implicit, which is indicated by whether the produced slides are comprehensible. PSs commented that the produced slides were easy to read (4, 0.56) and helped understand a visual design (4.15, 0.37). The scores (4.475, 0.47) PSs got in the quiz also proved that they understood the introduced visualization design. We plan to exploit explicit measurement to evaluate the fulfillment of **R4**.

We examine the fulfillment of **R2**, avoid unconscious ignorance, based on our observation during Authoring Session. For example, PT2 overlooked the encoding of glyph size in his sketch. He stopped with obvious hesitation when editing in the Channels Panel. He then went back to check the provided textual description and added this missing encoding. According to Table 1, failing to mention the encoding of certain channels when sketching occurred to three out of five teachers. All the three teachers added several, if not all, missing channels during authoring with Narvis, which indicates that Narvis is able to remind teachers of the visual encoding they ignored. Meanwhile, we also got negative feedback from two PTs about the restriction used in **R2**, indicating the need for further improvement about the designs related to **R2**.

7 DISCUSSION

7.1 Reflection on Evaluation Feedback

We reflect on users' feedback on using Narvis and derive several design lessons, which can guide the further improvement of Narvis and inform other designs of visualization introduction.

First, the same type of users may have different, even opposite, preferences when editing information at different levels of details. To avoid information overload, Narvis allows only one visual channel and one visual unit to be explained at a time. While PTs highly appreciate the restricted explanation order of *visual units*, they showed less interest in the restricted order of *visual channels* and even complained that this restriction limited their editing. A possible reason is that people prefer well-defined restrictions on high-level information (*i.e.*, visual units) and enjoy flexibility in editing detailed information (*i.e.*, visual channels).

Second, different types of users may have different opinions for the same setting. For example, participants in the students group were satisfied with the fact that only one visual channel was explained at one time. However, some participants (PT2, PT4) in the teacher group thought such a restriction was too limited, commenting that they "*might want to explain two channels at the same time*". This phenomenon indicates the importance of identifying different types of end users and understanding their different preferences.

7.2 Encoding vs. Insight.

When introducing a visualization design, the explanation of visual encodings and the introduction of insights are sometimes mixed. The mixed introduction of insights and encodings was also observed at three PTs (PT1, PT3, PT4) when using Narvis. For example, after the annotation "*height stands for the intensity of correlation*", which explained the visual encoding, PT3 added another annotation "*the keyword 'reagan' has a weak correlation with keyword 'debt' and a strong correlation with keyword 'leader'*" to introduce an insight.

Narvis is designed to help users organize an explanation of visual encodings, which is the foundation to understand a visualization design and to obtain insights from it. But Narvis also supports the introduction of insights through functions such as annotating and highlighting. Users can introduce the patterns displayed by highlighting relevant components and adding annotations at appropriate time points (*e.g.*, after introducing corresponding visual encodings). We will explore how to offer more guidance for insight discovery in future work.

7.3 Limitations

Narvis has several limitations.

First, the usage context of the current version of Narvis may be limited. From the perspective of end users, we merely interviewed students and teachers groups to derive design requirements. In this scenario, students have strong motivation to learn the visualization design and understand visual encodings. However, visualizations targeted at different end users may have different requirements for explanation. For example, readers in data journalism may care more about the insights a visualization conveys, and pay less attention to remembering visual encodings. Therefore, a pattern-based, or insight-based, explanation may be preferred. In future research, we plan to generalize the application of Narvis by studying how visualization is explained and interpreted in various scenarios. From the perspective of form, we only discussed tutorials in the form of slideshows, which are preferred by our interviewees for the explanation of complicated visualization designs. However, other forms of tutorials can also be efficient in certain working scenarios. For example, students in the interviews commented that legends can be helpful and efficient when the visual encodings are relatively simple. It is beyond the scope of this work to identify suitable working scenarios for different forms of tutorials. In this paper, we assume that the explained visualizations are complicated enough and can hardly be explained by, for example, a simple legend.

Second, the evaluation lacks external validity. We evaluated authoring experience and quality of the tutorials mainly based on self-reported perceptions. As with all self-reported data, results of this evaluation

have the potential for recalling bias, under-reporting, or over-reporting. Meanwhile, whether the produced tutorials improve understanding of visualization designs is evaluated based on PSs' scores in a quiz where a baseline comparison is missing. While the quiz results showed that participants in the students group correctly understood the visualization design, it is not clear the extent to which Narvis enhances the understanding of visualization designs. Thus, current results should be treated with caution and we plan to investigate whether our insights can apply more generally in future work. Nevertheless, we are encouraged by the fact that the current version of Narvis was appreciated by the initial users and got positive feedback.

Third, the input analysis method we implemented can be further improved. Currently, Narvis processes the input visualization as an SVG file. The components in the original visualization that are rendered as HTML elements (*e.g.*, tooltip as a $<div>$) are lost during the input analysis. Meanwhile, the analysis results are influenced by the structure of the input SVG. More specifically, if the input SVG has nested groups, the tree list will be complex and require more efforts from the teachers to select visual units. These problems can be alleviated by using more sophisticated input analysis methods.

8 CONCLUSION AND FUTURE WORK

In this paper, we present Narvis, an authoring tool to generate slideshows for explaining visualization designs. The design and implementation of Narvis are guided by our understanding of two types of end users, namely, teachers and students. Narvis provides a sequence organizer for clear narrative structures and a series of templates for easy implementation of common operations. Moreover, Narvis offers mechanisms for avoiding information overload and unconscious information omission. Thus, using Narvis results in an efficient crafting process and a high-quality generated slideshow. User studies have confirmed the utility and effectiveness of Narvis. To the best of our knowledge, this study is the first presentation tool tailored for the introduction of visual designs.

We envision improving Narvis in several directions. First, we will improve the design of Channel Panel to allow users more freedom of editing and to provide editing guidance without undermining flexibility. Second, we plan to better support the introduction of insights in Narvis. A possible solution is to automatically detect patterns (*e.g.*, outliers, clusters, extremes) and give hints to introduce insights after the explanation of corresponding visual channels. Third, we are interested in analyzing the differences between data visualizations targeted at different end users (*e.g.*, readers of data journalism, subject specialists) and their requirements of explaining. By doing so, we aim to extend the usage context of Narvis.

ACKNOWLEDGMENTS

The authors would like to thank all the participants involved in the studies and the reviewers for their valuable feedback. This work is partially supported by funding from the Theme-based Research Scheme of the Hong Kong Research Grants Council, project number T44-707/16-N.

REFERENCES

- [1] F. Amini, N. Henry Riche, B. Lee, C. Hurter, and P. Irani. Understanding data videos: Looking at narrative visualization through the cinematography lens. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI '15*, pp. 1459–1468. ACM, 2015.
- [2] F. Amini, N. H. Riche, B. Lee, A. Monroy-Hernandez, and P. Irani. Authoring data-driven videos with DataClips. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):501–510, 2017.
- [3] J. Bertin. Semiology of graphics: diagrams, networks, maps. *University of Wisconsin Press*, 1983.
- [4] J. P. Bliss, D. R. Lampton, and J. A. Boldovici. The effects of easy-to-difficult, difficult-only, and mixed-difficulty practice on performance of simulated gunnery tasks. Technical report, Army Research Institution for the Behavior and Social Sciences, 1992.
- [5] M. Bostock and J. Heer. Protovis: A graphical toolkit for visualization. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1121–1128, 2009.

- [6] J. Boy, F. Detienne, and J.-D. Fekete. Storytelling in information visualizations: Does it engage users to explore data? In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 1449–1458. ACM, 2015.
- [7] C. Bryan, K.-L. Ma, and J. Woodring. Temporal summary images: An approach to narrative visualization via interactive annotation generation and placement. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):511–520, 2017.
- [8] N. Cao, C. Shi, S. Lin, J. Lu, Y. R. Lin, and C. Y. Lin. TargetVue: Visual analysis of anomalous user behaviors in online communication systems. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):280–289, 2016.
- [9] W. S. Cleveland and R. McGill. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American Statistical Association*, 79(387):531–554, 1984.
- [10] N. Cohn. Visual narrative structure. *Cognitive Science*, 37(3):413–452, 2013-01.
- [11] W. Cui, S. Liu, L. Tan, C. Shi, Y. Song, Z. Gao, H. Qu, and X. Tong. TextFlow: Towards better understanding of evolving topics in text. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2412–2421, 2011.
- [12] L. S. Cunningham and J. J. Reich. *Culture and Values: A Survey of the Humanities*. Cengage Learning, 2009-20.
- [13] J. Davies. A visualization technique for multidimensional categorical data. <https://www.jasondavies.com/parallel-sets/>. Accessed: 2017-09-20.
- [14] C.-E. Dessart, V. Genaro Motti, and J. Vanderdonckt. Showing user interface adaptivity by animated transitions. In *Proceedings of the 3rd ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, EICS ’11, pp. 95–104. ACM, New York, NY, USA, 2011.
- [15] R. Eccles, T. Kapler, R. Harper, and W. Wright. Stories in GeoTime. In *2007 IEEE Symposium on Visual Analytics Science and Technology*, pp. 19–26, 2007.
- [16] J. Fulda, M. Brehmel, and T. Munzner. TimeLineCurator: Interactive authoring of visual timelines from unstructured text. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):300–309, 2016.
- [17] J. Harper and M. Agrawala. Deconstructing and restyling d3 visualizations. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST ’14, pp. 253–262. ACM, 2014.
- [18] C. Heath and D. Heath. The curse of knowledge. *Harvard Business Review*, 84(12):20–23, 2006.
- [19] J. Heer and G. Robertson. Animated transitions in statistical data graphics. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1240–1247, 2007.
- [20] W. Huang and C. L. Tan. A system for understanding imaged infographics and its applications. In *Proceedings of the 2007 ACM Symposium on Document Engineering*, DocEng ’07, pp. 9–18. ACM, New York, NY, USA, 2007.
- [21] J. Hullman, S. Drucker, N. H. Riche, B. Lee, D. Fisher, and E. Adar. A deeper understanding of sequence in narrative visualization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2406–2415, 2013.
- [22] S. Huron, S. Carpendale, A. Thudt, A. Tang, and M. Mauerer. Constructive visualization. In *Proceedings of the 2014 Conference on Designing Interactive Systems*, DIS ’14, pp. 433–442. ACM, 2014.
- [23] L. Itti and C. Koch. Computational modelling of visual attention. *Nature Reviews Neuroscience*, 2(3):194–203, 2001.
- [24] W. Javed and N. Elmqvist. Exploring the design space of composite visualization. In *IEEE Pacific Visualization Symposium*, pp. 1–8. IEEE, 2012.
- [25] B. Lee, R. H. Kazi, and G. Smith. SketchStory: Telling more engaging stories with data through freeform sketching. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2416–2425, 2013.
- [26] B. Lee, N. H. Riche, P. Isenberg, and S. Carpendale. More than telling a story: Transforming data into visually shared stories. *IEEE Computer Graphics and Applications*, 35(5):84–90, 2015.
- [27] S. McKenna, N. Henry Riche, B. Lee, J. Boy, and M. Meyer. Visual narrative flow: Exploring factors shaping data visualization story reading experiences. In *Computer Graphics Forum*, vol. 36, pp. 377–387. Wiley Online Library, 2017.
- [28] T. Munzner. *Visualization Analysis and Design*. CRC Press, 2014-01.
- [29] G. G. Mndez, M. A. Nacenta, and S. Vandenheste. iVoLVER: Interactive visual language for visualization extraction and reconstruction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI ’16, pp. 4073–4085. ACM, 2016.
- [30] H.-C. Nothdurft. Salience from feature contrast: variations with texture density. *Vision Research*, 40(23):3181–3200, 2000.
- [31] N. H. Riche, C. Hurter, N. Diakopoulos, and S. Carpendale. *Data-Driven Storytelling*. CRC Press, 2018.
- [32] P. Ruchikachorn and K. Mueller. Learning visualizations by analogy: Promoting visual literacy through visualization morphing. *IEEE Transactions on Visualization and Computer Graphics*, 21(9):1028–1044, 2015.
- [33] A. Satyanarayan and J. Heer. Lyra: An interactive visualization design environment. *Computer Graphics Forum*, 2014.
- [34] A. Satyanarayan and J. Heer. Authoring narrative visualizations with ellipsis. *Computer Graphics Forum*, 33(3):361–370, 2014-01.
- [35] M. Savva, N. Kong, A. Chhajta, L. Fei-Fei, M. Agrawala, and J. Heer. ReVision: Automated classification, analysis and redesign of chart images. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST ’11, pp. 393–402. ACM, 2011.
- [36] A. Scharl, A. Hubmann-Haidvogel, A. Jones, D. Fischl, R. Kamolov, A. Weichselbraun, and W. Rafelsberger. Analyzing the public discourse on works of fiction—detection and visualization of emotion in online coverage about hbos game of thrones. *Information Processing & Management*, 52(1):129–138, 2016.
- [37] D. Shen and P. Huhn. The living handbook of narratology. *Hamburg University Press*, 2011.
- [38] J. von Engelhardt. *The language of graphics: A framework for the analysis of syntax and meaning in maps, charts and diagrams*. Yuri Engelhardt, 2002.
- [39] Y. Wang, Z. Chen, Q. Li, X. Ma, Q. Luo, and H. Qu. Animated narrative visualization for video clickstream data. In *SIGGRAPH ASIA 2016 Symposium on Visualization*, SA ’16, pp. 11:1–11:8. ACM, 2016.
- [40] G. H. Weber, S. Carpendale, D. Ebert, B. Fisher, H. Hagen, B. Shneiderman, and A. Ynnerman. Apply or die: On the role and assessment of application papers in visualization. *IEEE Computer Graphics and Applications*, 38(3):96–104, 2017.
- [41] Y. Wu, F. Wei, S. Liu, N. Au, W. Cui, H. Zhou, and H. Qu. OpinionSeer: Interactive visualization of hotel customer feedback. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1109–1118, 2010.
- [42] A. Yadav, M. M. Phillips, M. A. Lundeberg, M. J. Koehler, K. Hilden, and K. H. Dirkin. If a picture is worth a thousand words is video worth a million? differences in affective and cognitive processing of video and text cases. *Journal of Computing in Higher Education*, 23(1):15–37, 2011.