

Narvis: Authoring Narrative SlideShows for Presenting Data Visual Designs in A Constructing Way

Category: Application

Paper Type: application/design study

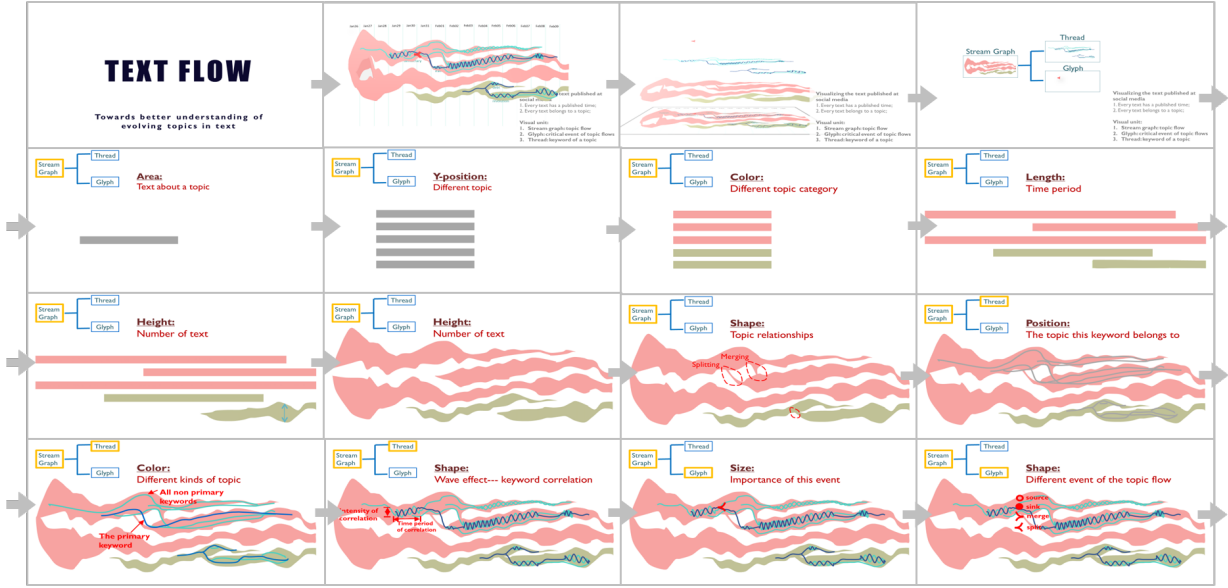


Fig. 1. Extracted slides from an introduction slideshow of *TextFlow* [14] produced by an expert in data visualization using Narvis. This slideshow illustrates the overview through a decomposition animation (the first line), introduces a visual unit (i.e. the stream graph) through revealing visual channels gradually (the second line), and introduces a visualization through constructing three visual units, a stream graph, a thread, and a glyph.

Abstract—Visual designs can be complex in modern data visualization systems, which poses special challenges for explaining them to the general audience. However, there is few theoretical work or presentation tool tailored for introducing a data visualization design. In this study, we present Narvis, an authoring tool for the crafting of narrative slideshows that introduces a visualization design. In Narvis, a visualization is specified as a combination of visual units and demonstrated in a constructing way. To better guide the crafting of an introduction slideshow, we incorporate lessons from previous work with our observation and propose a hierarchical constructing model, which consist of: conceptual components at different hierarchical levels, the process that components assemble another component at a higher level, and suggestions for the utility of narratives when introducing different components. Guided by this model, we implement a library of templates in Narvis. It enables the editors crafting an introduction slideshow through assembling these templates, thus achieves a level of expressiveness while improving efficiency. Narvis is designed and implemented specified for text visualizations but can be generalized to other types of visualizations. We evaluate Narvis through a preliminary evaluation of the authoring experience, a quantitative analysis of the generated slide show, and a qualitative analysis of the generated slideshow in the aspect of aesthetic, engagement, readability and utility.

Index Terms—Illustrative Visualization, Presentation, User Interface

1 INTRODUCTION

Simple data visualizations, such as bar chart and line chart, are not sufficient to meet various needs of end users. By introducing metaphors borrowed from nature [8, 28], applying carefully designed layout algorithms [11, 49], and sophisticatedly combining existing visualizations [51], novel visual presentations help users identify patterns, trends and correlations in data. However, these novel visualizations are sometimes complex and require steep learning curve, which hinder users from grasping insights into them.

As the field of data visualization is getting mature, many researchers have moved their focus from analysis to presentation. From the perspective of visualization experts, they are willing to generalize novel designs to other domains, and make their designs visible to general audience. However, they face three challenges when introducing novel yet complex visual designs to general audience. First, complex visual

designs attempt to deliver a great amount of information, which would overload the audience if all information is dumped at one time. Second, even if experts try to explain complex visual designs with an order. Considering the logic dependence existing, an improper explanation might confuse the audience. For example, in a node-link diagram, a node should be introduced before the links connecting it. In a complex visual design, which has more components than just nodes and links. To identify a proper logic for explanation is challenging. Third, given considerable amount of information, the audience can easily get lost or forget previous information. How to engage the attention and attract their attention is challenging.

Narrative data visualization is becoming an emerging topic [30]. Narrative, which means “connected events presented in a sequence”, has long been used to share complex information [43]. Many efforts have been made to define, classify, and provide design suggestions for

narrative data visualization [20, 26, 44]. Some visualization systems have already incorporated narrative modules into their design [7, 18]. However, most prior arts focus on communicating the conclusion of data analysis, and few discussions aim at how to guide the audiences to learn a visual design.

In this paper, we present a framework to introduce new visual designs. Based on our analysis of the structure, logic dependency, and visual distractions existing in a visual design, we develop an authoring tool, Narvis, to decompose a visualization, reorganize extracted visual elements, and explain their visual grammars one by one through animated transition in the form of a slideshow. Through incorporating a narrative sequence, appropriate chunks of information, rather than all the information, are delivered to the audience at one time, which avoids information overload. Visual attention guidance, such as flickering, highlighting, and morphing are used to lead audiences' attention to newly added information. We ground this framework on text visualizations, which are widely applied to designed to explore and analyze textual data. But we believe our work generalizes to other kinds of visualizations.

To the best of our knowledge, this is the first attempt to introducing a visual design in a constructing way. Our contributions are as below: 1) A paradigm for decomposing visualizations. It analyzes the hierarchical structure of its components, the relationships between components, and possible visual distractions. 2) A framework for explaining a visual design, which is the result of consulting theory from graphical perception process, techniques in narrative visualization, various attention cues in animation, and empirical observations of numerous visual designs. 3) An authoring tool to generate and edit the narrative visual encoding explanation. We conjecture our work can motivate and enable people to use more advanced visual designs, supporting the democratization of data visualization.

2 RELATED WORK

In this section, we review prior arts of the analysis of narrative structure in data visualization, animation in data visualization, and existing authoring tools for narrative visualizations.

2.1 Structure of Narrative Data Visualization

Narrative is as old as human history [15]. People in the fields of literature, comics [13] and cinema [43] have gone to great lengths to analyze the sequencing and forms of grouping used in a narrative, as well as how they affect the meaning a narrative tries to deliver.

Researchers in the community of data visualization are greatly inspired by work in other fields. Amini et al. [1] borrow concepts from comics [13] to classify and analyze the structure of data videos. Wang et al. [47] adopt two representative tactics, time-remapping and foreshadowing, from cinematographers to organize a narrative sequence for visualizing temporal data.

Some researchers, on the other side, focus on the narrative structures exclusively for data visualization. Satyanarayan and Heer, through interviews with professional journalists [41], define 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. [26] propose a graph-driven approach to automatically identify effective narrative sequences for linearly presenting a set of visualizations.

These works, however, rarely discuss the narrative structures used for explaining visual designs. We propose a narrative framework specified for presenting visual designs to fill this gap.

2.2 Authoring Tools for Narrative Visualizations

The extensive needs of data communication exist not only in the data visualization field but also in journalism, media, etc. This has motivated researchers to investigate ways for authoring narrative visualization.

User experience is of great concern when utilizing an authoring tool. Sketch story [33], with its freeform sketch interaction, provides a more engaging way to create and present narrative visualization. Dataclips [2] lowers the barrier of crafting a narrative visualization by

providing a library of data clips, allowing non-experts to be involved in the production of narrative visualization.

However, it is information delivery that is the core consideration of an authoring tool. Existing authoring tools usually choose a specific type of narrative visualization based on the information type [2, 19]. Meanwhile, integrating an authoring tool for narrative visualization with a data analysis tool has become a trend since it effectively bridges the gap between data analysis and data communication [7, 18, 34].

These tools offer inspiring user interaction design as well as good examples to implement narrative visualization. However, they treat visual encodings as cognitively obvious attributes that can be universally recognized without a formal introduction, which is not appropriate for introducing visual designs.

2.3 Decomposition of Data Visualization

Clarifying the design space of a data visualization can help people get a better understanding of how it is constructed. Munzner [35] proposes that it "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. [27] extends the design space of a data visualization, defining the components of a data visualization as a token, token grammar, environment and assembly model.

Such theoretical work motivates the designers of visualization tools to contribute efficient high-level visualization systems rather than low-level graphical systems [5, 36].

On the other hand, theoretically identifying the basic components of a data visualization enables people to physically extract them, and remap them to an alternative design without involving any programming work. Harper and Agrawala [21] contribute a tool that extracts visual variables from existing online visualization designs to generate a new design. Huang et al. [23] propose a system that recognizes and interprets imaged infographics from a scanned document. Revision [42] applies 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.

However, these decomposing methods exclusively focus on simple visualization designs, such as bar chart, line chart, dot chart, and are not applicable for complex visualization designs, which assemble miscellaneous visualization approaches to realize a novel presentation. Moreover, these methods are not specified of explaining a visual design, thus giving no consideration for graphical perception process and visual attention shift.

3 INTRODUCING A DATA VISUALIZATION

To help people better understand a data visualization design, we propose a model that introduces a data visualization through constructing, which has been proven as an effective teaching method [9, 27]. To build a constructive model for presenting visual designs, there are three questions we need to answer: "what are the basic components that compose a data visualization?", "what is the relationship between these components?", "How should we deal with these relationships in our narrative?". At the same time, considering a large number of graphical elements employed in a data visualization design, we should eliminate the visual distraction to keep audience's focus on the target.

We propose this model based on our observation of the 375 text visualizations collected and classified in the survey made by Kucher [32], as well as the lessons from previous work.

3.1 Compositions of a Visualization

Several work has taken efforts to identify the atomic building blocks of a visualization [3, 36]. This model is the extension of previous work by 1) proposing a hierarchical structure; 2) including the relationship between components; 3) considering the requirements for composing a comprehensive narrative sequence. In our model, a visualization is decomposed into a hierarchical structure of three levels, including visual primitives, visual units, and an advanced visualization design. Taking OpinionSeer [50] as an example, we apply the hierarchical model and decompose it into five visual units, as shown in Figure 2.

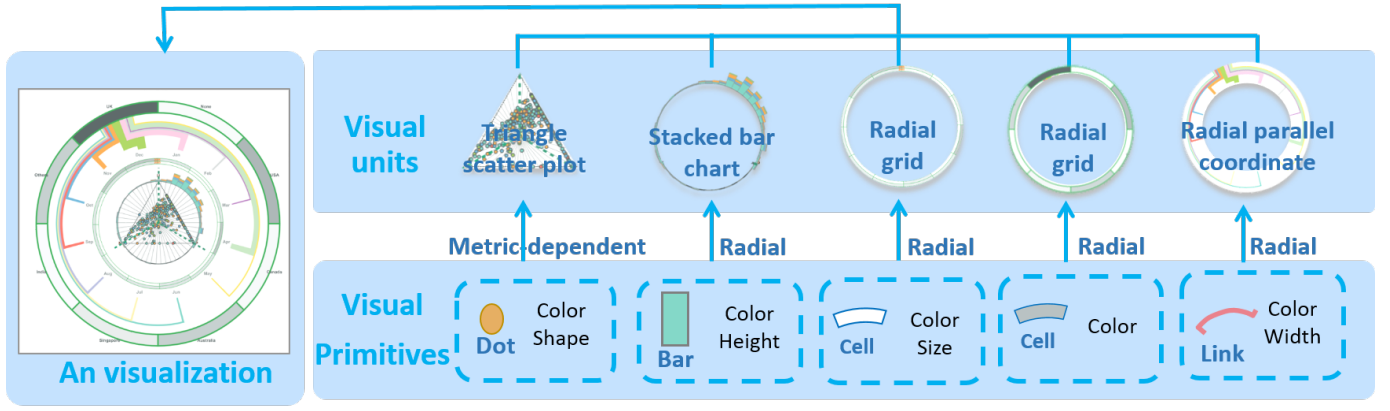


Fig. 2. After decomposing OpinionSeer [50], we obtain the hierarchical structure of this visualization. It consists of five visual units, and employs three kinds of visual primitives.

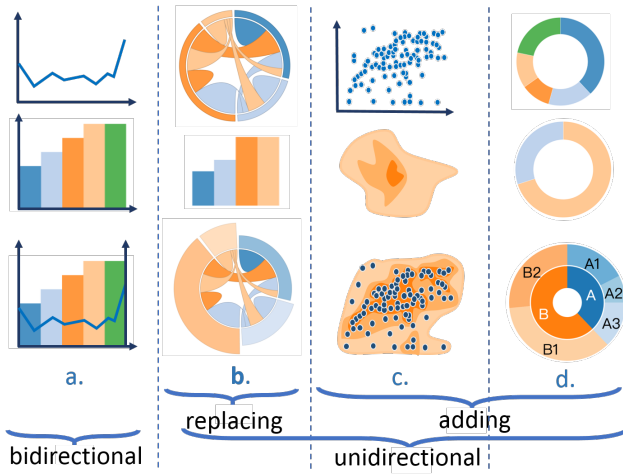


Fig. 3. Illustration of the two kinds of relationship between visual units

A **visual primitive** is one graphical element whose visual channels, such as color, width, height, are mapped to data attributes with certain visual grammars. Visual channels are visual properties that control the appearance of a graphical element, and a visual grammar describes how a visual channel indicates 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 the assembly of one kind of visual primitives based on a certain construction rule, as Table 1 shows. Considering the fact that new designs are emerging, the purpose of this table is not to cover all visual units but to demonstrate how a visual unit can fit in a table cell. Visual primitives of one kind can constitute different visual units by following different construction rules. For example, dots can constitute scatter plot, spiral dot chart, or circle packing chart by following radial, orthogonal, or metric-based construction rules, respectively. A visual unit is the smallest functional unit of a visualization.

A **visualization** can be treated as the combination of visual units. A simple visualization contains only one visual unit while an advanced one is usually the combination of multiple visual units.

3.2 Relationships Between Compositions

We first describe the relationship between conceptual compositions, then offer suggestions for narrative sequence based on these relationships. Notice that we skip the relationship between visual primitives since there is only one kind of visual primitives in a visual unit.

Table 1. A taxonomy of visual units.

	Radial	Orthogonal, Align, Map	Metric-based
Dot	Spiral	Dot Chart, Scatter Plot, Bubble Chart, Bubble Map	Circle packing
Line	Radar Chart, Spiral Plot	Line Chart, Parallel Coordinates, Arc Diagram	Force-directed Node-link graph
Flow	Chord Diagram	Parallel Sets, Sankey Diagram, Flow Map	
Area	Area Spiral Chart	Stream Graph	Contour Map
Bar	Radial Bar Chart, Spiral Bar Chart	Candlestick Chart, Bar Chart	
Cell	Sunburst Diagram	Matrix, Tree Map	
Wedge	Pie Chart, Donut Chart		
Text	People Spiral in [17]	Sentence Tree	Word Cloud

3.2.1 Relationships Between Visual Units

A visualization can be specified as the combination of several visual units. We go through all the text visualizations collected in Kucher and Kerren’s survey [32], and identify two types of relationships between visual units: unidirectional relationship and bidirectional relationship. Since visual units are the combination of construction rules and visual primitives, we can describe the relationship between two visual units through describing the relationship between their visual units and constructing rules.

Bidirectional relationship refers to that the dependency of two visual units is mutual. They share the same construction rule and their visual primitives indicate separate data attributes in a dataset. For example, in Figure 3(a) a line chart and a bar chart are put together with the same construction rule.

Unidirectional relationship refers to that one visual unit “A” depends on another visual unit “B”. This dependency is established either by “replacing” or “adding”. A dependency established by “replacing” refers to that “A” replaces the original graphical elements in “B” to build a new visual design. For example, in Figure 3(b), a bar chart replaces the node segments in a chord diagram. This kind of dependency widely exists in the advanced visualization design, such as the heat map mapped upon the streams in a theme river [49] and usage of glyphs to replace the nodes in a multidimensional scaling (MDS) plot [10]. A dependency established by “adding” means that “A” is added to reveal

a more detailed description of “B”. Such description can be the more detailed information from dataset, or information obtained from the previous visualization (Figure 3(c)). For example, in Figure 3(c), a contour map is mapped upon a MDS map to illustrate the area with high distribution density, which information is obtained from the layout. While in Figure 3(d), a donut chart is added to another donut chart to reveal more hierarchies, which information is obtained from the dataset.

A proper narrative sequence of visual units should take these two types of relationships into consideration. For a unidirectional relationship, the dependent should be explained after the visual unit it depends on. For a bidirectional relationship, these two visual units should be explained together but the sequence can be arbitrary. We display relationships between units through a tree diagram where every visual unit is represented by a tree node. Sibling nodes stand for visual units with a bidirectional relationship while a child node and a parent node stand for two visual units that have a unidirectional relationship. A proper narrative sequence can be obtained by running a deep first search on this tree diagram.

3.2.2 Relationships Between Visual Channels

For a visual primitive, different channels are encoded with different data attribute. Thus, they are usually separated and have no logic dependency upon others. It’s hard to determine a narrative sequence from their inner logical dependency. Therefore, we define two metrics to order the explaining of visual channels: **the complexity of their encoded information** and **saliency of their visual appearance**.

First, the order of decreasing visual saliency can facilitate graphical perception [12]. Even though different channels have intrinsically different perceptual saliency and channel with high saliency will suppress the expression of other, such saliency strength can be influenced in a task-dependent manner [37]. By introducing the channel with high saliency first, we remove it from the task list in our mind [29], decrease its saliency and give other channels more chance to attract the limited human attention.

Second, the order of increasing complexity leads to an effective learning process. Easy to difficult practice has been long used and confirmed to be effective for learning new tasks [4].

The visual saliency of different channels is relatively constant and well defined [12, 35], while their information complexity varies in different designs. And an effective narrative sequence is a trade-off between these two metrics.

3.3 Attention Orientation

To keep the audience focusing on target objects, it is necessary to identify and avoid visual distractions. We identify two kinds of visual distractions: the one from context and the one from sibling channels, which refer to the visual channels belonging to the same visual primitives.

3.3.1 Visual Distraction from the Context

This kind of distraction has been widely discussed in the field of object detection and human visual attention [37, 45]. Its intensity is mainly determined by spatial distance and appearance similarity [48]. Focus + Context, which might be the most popular techniques for this problem, make uneven use of graphic resources to discriminate focus from their context. At the same time, adding dynamic changes to focus elements has also been demonstrated as effective under various conditions [46]. We support easy application of these techniques in our system.

3.3.2 Visual Distraction from Sibling Channels

A visual primitive usually has more than one visual channels. Thus, when recognizing one primitive, the channels with high visual saliency can significantly influence the expression of other channels. For example, color can be a strong noise when the focus is supposed to be the shape. Our system is able to eliminate this kind of distractions by exaggerating or suppressing specific visual channels.

4 DESIGN CONSIDERATIONS

In this section, we first describe our understanding of two groups of end users, i.e., editors and the general audiences. Then, we distil design tasks to guide our design and development of Narvis.

4.1 User Perspectives and Methods

Narvis aims to offer an efficient, expressive and friendly authoring tool for experts in data visualization, assisting them to create a slideshow to introduce advanced visual design to the general audience. Hence, we identify two different user perspectives: the editors and the general audience perspectives. Editors are visualization experts who have the need to create a slideshow to present visual designs. The general audiences have no prerequisite for visualization. They gain an understanding of a visual design through the slideshow created by the editor.

To understand the current practice of making slideshows and the experience of reading tutorials, we collaborated with four teaching assistants (TAs) of a data visualization course and seven undergraduate students (UGs) taking this course. The four TAs are postgraduate students whose research interests are information visualization. Their duty of this course involves making slides to introduce visual designs from major publications in the field of visualization. The slides should cover fine-grained description to help students review them after class. The seven UGs have no prior experience in visualization, and have taken this course for no more than one month.

We began by conducting semi-structured interviews with TAs, whom we identified as editors, and UGs, as general audiences. During the interviews with TAs, we asked their workflows of making slideshows and explaining visual design. To identify opportunities for Narvis, we also asked them to enumerate a list of challenges faced in the workflows. The interviews with UGs are semi-structured as well. We asked their comments in reading the slideshows and attending course lectures. Then, we used mind-mapping to find clusters in their comments that defined goals for an ideal slideshow.

4.2 Design Tasks

Based on our observations and the interviews, we categorize six design tasks to guide the design of Narvis. Three tasks, denoted as DE, are originated from the interview with editors, i.e., the four TAs, and other three tasks (DA) are from the general audiences (UGs).

DE1. Emphasis on efficiency. TAs used presentation tools, such as Power Point¹ and KeyNote² to introduce visual designs. However, these tools are for general purpose and not tailored for visualization presentation. For example, “focus + context” techniques are widely used in data visualization to guide the users attention to the region of interest [16, 31]. It requires exaggeration or suppression of the visual channels, like hue, luminance, sharpness, or size, of graphical elements, which are hard to perform in these common presentation tools.

DE2. Suggest options. People with extensive experience in designing data visualization can have little knowledge about how to present a visual design. Thus, suggesting design options to them for creating a comprehensive slideshow is demanding. Many presentation tools already offer this kind of service. For example, Power Point Designer³ automatically generate a list of professionally designed layouts based on the contents. However, these suggestions focus on general issues, especially on aesthetics, and give no special consideration for presenting a visual design. For example, composing a clear narrative sequence from all the visual grammar employed can facilitate the perception process. A list of design options can help editors quickly ideate on how to organize the narrative sequence and convey the insights of a visual design. However, no available presentation tool supports such service.

DE3. Collect feedback. “When students read my slides, I do not know whether they can follow the logic, or whether the slides cover enough details for them to grasp the visual design.”, one TA commented.

¹<https://office.live.com/start/PowerPoint.aspx>

²<http://www.apple.com/keynote/>

³<https://support.office.com/en-us/article/About-PowerPoint-Designer-53c77d7b-dc40-45c2-b684-81415eac0617>

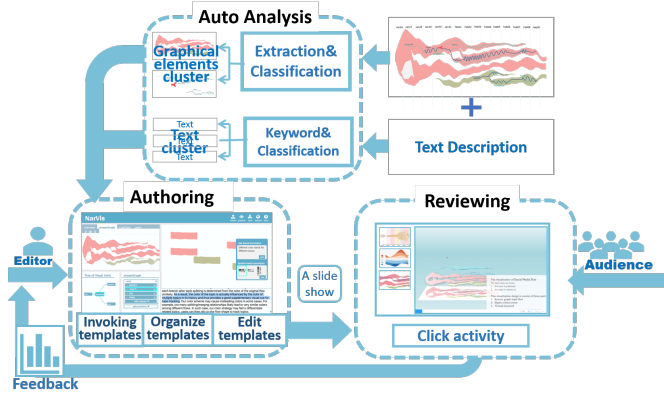


Fig. 4. An abstract representation of the workflow of Narvis: Auto Analysis, Authoring, Reviewing, and how the two kinds of users: editors and audiences, are involved.

Collecting feedbacks of audiences is crucial for editors to revise their slideshow, making it more understandable and attractive.

DA4. Avoid information overload. All UGs complained that they had experienced information overload in reading slides. When the information in one slide is overwhelming, it is common for them to miss important visual encodings. The slideshow should be well designed to ensure that the amount of new information in each slide is appropriate.

DA5. Avoid unconscious ignorance. Experts in data visualization, i.e. the TAs of the course, prone to treat some visual grammars as self-evident that need no explanation. However, the lack of information confuses the UGs, who have no prior knowledge of visualization. Considering the importance of information integrity to a comprehensive slideshow, we need a mechanism to guarantee that all visual grammars are explained.

DA6. Keep the sense of overview When conveying a considerable amount of information to the audiences, they can easily get distracted or forget previous information. In this situation, taking some measures to inform the audiences of the overall structure and the current state can help the perception process.

5 NARVIS: SYSTEM DESIGN AND IMPLEMENTATION

Guided by the theory model discussed in Section 3, as well as the design tasks discussed in Section 4 we design and implement Narvis, an authoring tool for crafting slideshows for the presentation of visualization. The workflow of Narvis consists of three phases (Figure 4), i.e., Automatic Analysis Phase, Authoring Phase, and Viewing Phase.

5.1 Phase1: Auto Analysis

The input of Narvis includes two parts: one image presenting a visual design (mandatory) and a piece of text describing the design (optional). Considering that the vast majority of visualizations are only available as bitmap images, we offer an algorithm to detect and extract the graphical elements from the input image. Here, we explain the basic idea of how Narvis analyzes these input sources to facilitate further authoring.

5.1.1 Analysis of Input Image

The analysis of the input image includes two steps, object detection and object clustering.

Object detection. We iterate through all pixels, and clusters all the pixels that are i) neighbors and ii) have the similar color through a modified breath-first search algorithm. Simple objects, such as a bar in a bar chart, a node in a scatter plot, are detected and extracted after this step.

Object clustering. All the objects with spatial and appearance similarity are clustered to allow an efficient manipulation, as described in Algorithm 1. For example, all the nodes in a scatter plot should be put in one cluster, instead of letting the editor add them one by one manually.

Algorithm 1 Object Clustering

Data: A list of objects: A , the number of clusters: N

Result: A list of objects: B

```

 $E \leftarrow \{\}$ 
foreach:  $a_1 \in A$  do
  foreach:  $a_2 \in A$  and  $distance(a_1, a_2) \leq L$  do  $\triangleright L$  is a parameter
    that accelerates our calculations
    if  $a_1$  and  $a_2$  have similar color then
       $d \leftarrow$  the distance between  $a_1$  and  $a_2$ 
       $E \leftarrow E \cup \{(a_1, a_2, d)\}$ 
    end if
  end for
end for
 $B \leftarrow A$ 
 $E \leftarrow$  sort  $E$  by the  $d$  of each elements in descending order
foreach:  $(a_1, a_2, d) \in E$  do
   $b_1 \leftarrow b | b \in B, a_1 \subseteq b$ 
   $b_2 \leftarrow b | b \in B, a_2 \subseteq b$ 
  if  $b_1 \neq b_2$  then
     $B = (B - \{b_1\} - \{b_2\}) \cup \{b_1 \cup b_2\}$ 
  end if
  if  $|B| \leq N$  then
    break
  end if
end for
return  $B$ 

```

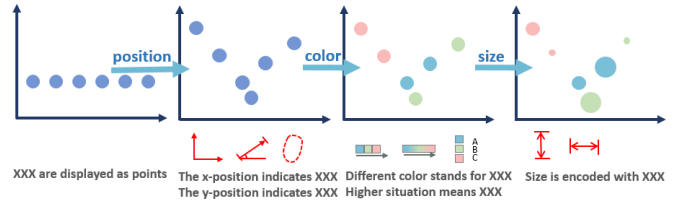


Fig. 5. Demonstration of the template for bubble chart, which is composed of series of slides (top), symbol-based annotation for each visual channel (middle), and text-based annotation for each visual channel (bottom).

5.1.2 Analysis of Input Textual Description

For the inputted textual description, we offer a keyword detection and classification algorithm, which is based on a dictionary of terms that are identified as highly related with visual grammars. e.g. the word “length” and “encodes”, are highly correlated with the visual grammar of size. We extract all the sentences containing the terms in our dictionary, and cluster them based on the terms they have.

The algorithm we proposed is a compromise between efficiency and performance. It is a heuristic algorithm for images with high quality and clear edges, but its performance can be improved by adopting other well-established algorithms, such as an algorithm based on patch detection and clustering [42] and an algorithm based on edge maps [24].

5.2 Phase2: Authoring

In the Authoring Phase, editors craft an introduction slideshow by constructing built-in blocks called templates in Narvis. We first explain how we design and organize the templates in Narvis, then demonstrate the workflow of this phase, which includes three steps, i.e., invoking templates, organizing templates and modifying templates, as illustrated in Figure 4.

5.2.1 A Library of Templates

We propose a library of templates for the narrative explanation of a visualization. A template is a set of slides that intends to introduce a visual unit. Since an advanced visualization design is the assembly of miscellaneous visual units, such templates can achieve a high level of

Table 2. A summary of the animations provided by Narvis

Animation	Engaging	orientate attention	perception	working scenario	ref
Morphing	✓	✓	✓	grammar of size, grammar of shape	[22,40]
Blur		✓		focus+context	[38]
Flicker		✓		focus	[46]
Motion	✓	✓	✓	grammar of position	[25]
Zoom-in/out	✓	✓		focus	
Annotation		✓	✓	textual explain	[44]
Fade in/out		✓			
Decompose	✓		✓	Show how a visualization is composed by visual units	A novel design by us

efficiency for the explanation of a visualization (DA.1). Furthermore, to adapt to various usage scenarios, Narvis allows users to modify and refine templates through rich interactions.

Types of templates. Narvis organize the provided templates with a 8*3 matrix, where the 3 columns stand for 3 construction rules and the 8 rows stand for 8 visual primitives, as shown in Table 1. Narvis is extensible, we plan to introduce new templates in the future. At the same time, it is desirable to keep the set of supported marks small and well organized, so as to avoid overwhelming users with a cornucopia of confusing options.

Templates design. We apply the analysis and theory model in Section 3 for the design of templates. A template has four core components: 1) a well-considered narrative sequence for visual grammar explanation; 2) exaggeration or suppression of certain visual channels in some slides; 3) a series of narrative techniques such as attention cues, animated transitions, information repetition, to orientate visual attention and facilitate perception (DE.1); 4) Hints for adding annotations (DE.2) in each slide.

With a visual unit, more specifically, a set of graphic elements, as input, a template will generate a slideshow and each slide illustrates one visual grammar(DE.1). These slides are sorted based on the narrative sequence we discussed in section 3.3. In each slide, we offer hints to guide the annotation process (DE.2). These hints are a sentence with blanks to fill in, heuristic questions, or a list of suggestion symbols. A visual channel is suppressed until its grammar has been explained. For example, in Figure 5, before we introduce the visual grammar of color, all the object will be blue. The graphical elements in different slides, which have different visual appearances due to the applied exaggeration or suppression of certain visual channels, are perceptively connected through morphing animation.

Animation embedded in templates Narvis provides 8 types of animation, implement them in templates based on their effects on human attention and perception (DA.1), which has been widely discussed in previous work [22,39,46]. We also provide a novel decomposition animation, which displays how a visualization is decomposed to several visual units. This animation, at the beginning of the introduction slideshow, aims to engage the audience as well as to help them get a sense of overview (DA.6). Animation is a double-edge sword, which brings both benefits and pitfalls. We are not discussing its effects here but leave it to the judgment of editors by giving them the freedom to remove it.

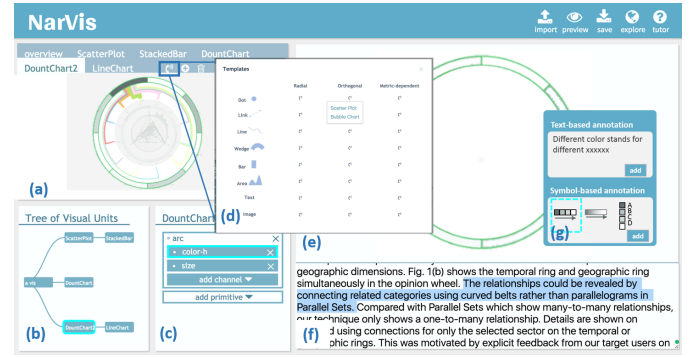


Fig. 6. Annotated screenshot of the interface of Narvis: a) Source Panel, b) Tree Panel, c) Unit Panel, d) the library of templates, e) Edit Panel, f) text area where the related sentence is highlighted from input textual description, and g) a floating annotation window that offers options for adding annotation

5.2.2 Invoking Templates

After graphical elements are extracted and clustered based on visual representation, each cluster appears as a tabbed panel in the *Source Panel* (Figure 6(a)).

Editors can switch between these tabbed panels, add or delete graphical elements in each panel, making sure that 1) all the graphical elements of the same visual unit are in the same panel; 2) every graphical element belongs to one and only panel. Then, for each visual unit, editors call a template from all the templates provided by Narvis (see in Figure 6(d)).

The relationship between graphical elements and templates is similar to the one between data and function. Templates contain a set of operations to produce a sequence of slides from the input graphical elements. For example, in Figure 5, the “scatter plot” templates modify the appearance of the input picture and outputs 4 slides that describe different visual grammars.

5.2.3 Organizing Templates

Once invoked, a template will show on the *Tree Panel* (Figure 6(b)) as a tree node. By dragging and dropping these nodes, editors organize the structure of the tree diagram, which reflects the relationship between visual units and determines the narrative sequence of the slideshow. This tree diagram will be automatically inserted in the generated slideshow, demonstrating the overall structure to the audience (DA. 6).

5.2.4 Modifying Templates

Narvis provides templates to generate slideshows with high efficiency. It also supports flexible modification of templates for expressiveness. Editors can edit a template in the *Unit Panel* (Figure 6(c)) by selecting a node on the *Tree Panel*. In each template, all possible visual grammars are enumerated. Editors can delete unused one themselves, instead of adding the ones used, thus eliminating the unconscious omission of crucial information (DA.5). It also recommends a narrative sequence of visual grammars, based on the metrics we mentioned in Section 3.2 (DA.2). In the *Editor Panel* (Figure 6(e)), with the hints from Narvis, editors add annotations to facilitate graph and chart comprehension. For each slide, which is defaulted to explain one visual grammar in our templates, Narvis offers questions or sentence with blank for adding text-based annotation and a list of suggested design options for symbol-based annotation (DE. 2) that can be added into the slide by a simple one-click (DE.1), as show in Figure 6(g).

5.3 Phase3: Viewing

The slideshow produced by editors will then be watched by the audiences. By clicking the “explore” icon on the right top, audiences will be directed to a new window, where Narvis exhibits all the slideshows produced and uploaded. Audiences can choose one slideshow for watching, click buttons to move forward or backward to view all the

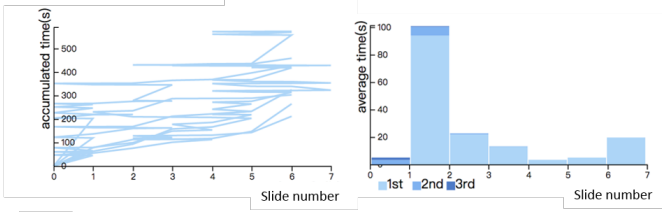


Fig. 7. The click stream data of one slideshow visualized as a line chart (left) and a stacked bar chart (right). The x-axis stands for the number of slides. In the line chart (left), the y-axis stands for the accumulated time an audience spent for watching, while in the bar chart (right), the y-axis stands for the average time all audience spent on watching this slide.

slides it contains. Their click activity will be recorded automatically by Narvis for the analysis of their watching behavior.

These clickstream data will be visualized in the form of a stacked bar chart and a line chart (see in Figure 7). The x axis represents the slide's number in both charts. In the line chart, the y axis represents the accumulated watching times, and each line refers to the watching behavior of one individual audience. In the bar chart, the y axis represents the average watching time of all users for a certain slide. Each bar is split into colored bar segments. The bottom bar segment represents the time audience spent the first time for watching this slide, if they go back and watch this slide for the second time, a bar segment with darker color will be placed on the top of the previous one, and so on.

The line chart emphasizes on the watching behavior of every individual while the bar chart focuses on the description of every single slide. With the help of these two charts, the editor is able observe how the audiences watch his slideshow and generates ideas for later revision (DE.3).

5.4 A Working Scenario

Jessica has extensive experience in the field of data visualization, and has implemented a visual analytics tool for a review service website based on the design of OpinionSeer [50], which has five visual units as demonstrated in Figure 2. To help audiences better understand this design, she needs to publish a tutorial accompanied with it. First, she loads the screenshot of her system, as well a textual description, into Narvis. After a few seconds, the system automatically extracts the graphical elements. Jessica first adds five tabbed panels (see in Figure 6(a)), since she identifies five different visual units in OpinionSeer. At each panel, where the uploaded image shows with half-transparency as background, Jessica adds graphical elements by clicking it. As in Figure 6(a)), the “geometry ring” is added to a tabbed panel and highlighted. Note that Narvis pre-clusters some graphical elements to convenient the users. For example, all the dots in “scatter plot” will be highlighted just by clicking on one dot.

After some editing, each tabbed panel includes all graphics elements belonging to one visual unit. Now, she chooses templates for each visual unit. She first chooses a visual unit by clicking its tab, then clicks the “phone” icon. A table (Figure 6(d)) jumps up, which categorizes all the templates as the 8*3 table we described in Section 3.1. For example, when Jessica clicks on the (2,1) cell, a dropdown list that contains two templates, the templates for bubble chart and scatter plot, will appear.

One by one, Jessica invokes 5 templates, all then show as tree nodes in *Tree Panel* as children of the “a vis” node. Jessica reorganizes the structure of the tree diagram (see in Figure 6(b)) by dragging and dropping based on her understanding of this visual design.

Moreover, Jessica edits the narrative templates based on her design. She goes through all five templates in the *Unit Panel* by clicking the corresponding node in *Tree Panel*, and deletes the visual channels with no encodings, such as the size in the template of scatter plot.

Jessica further adds annotations at each slide with the help from an annotation window (see in Figure 6(g)). This annotation windows offers some design options for adding text-based annotations as well

as symbol based annotations. The text area Figure 6(f) also offer hints for the addition of annotation by highlight the corresponding textual description.

To refine the readability of the tutorial, Jessica asks several friends, who have little experience in data visualization, to watch the tutorial before release. Narvis collects their viewing behavior from click activities, generates statistics results, and visualize it in the form of a stacked bar chart and a line chart (Figure 7). From these two charts, Jessica finds out that people spent significantly longer time on reading the 4th slide, which points out that information in this slide is not clear or too complicated. Jessica then revises her slideshow accordingly.

6 EVALUATION

We conducted a user study to evaluate Narvis through comparing the authoring experience and output generated by Narvis and a general presentation tool, i.e., PowerPoint. Our study was a between-subjects design with two sample groups: one group of participants using Narvis, the other group using PowerPoint. We report our qualitative observations during the authoring process, and provide insights on the quality of the slideshows generated from both groups.

6.1 Participants

We invited the same four TAs mentioned in Section 4 as editor participants, denoted as PC1 and PC2 for the control group, and PE1 and PE2 for the experiment group. We also sent emails to students in the data visualization course we mentioned before and recruited 20 volunteers as audiences, denoted as PAs, to evaluate the quality of the generated slideshow.

6.2 Material

We extracted the visual design and the corresponding literature description from TextFlow [14]. This visualization design conveys multiple level results of topic evolution analysis: a set of topics with splitting/merging relationships among each other, which encodes a series of topic flows, a set of critical events, which encodes glyphs, and the keyword correlations, which encode threads. We chose TextFlow because it is one of the visual designs that our experts found it hard to explain during their previous TA experience.

6.3 Procedure

6.3.1 Generating Slideshows

We ran 100-min long sessions for the four editor participants separately. This session consisted of 3 phases: (1) *Learning Phase*, (2) *Sketching Phase*, (3) *Preparing Phase*, (4) *Authoring Phase*. We were present in the room for the whole session to observe and took videos.

In the *Learning Phase*, participants read the literature description extracted from the paper, which offered a detailed description of the visual design with diagrams. This phase ended when the participants reported us that they finished reading and understand this visual design. This phase took about 15 minutes for each participant.

In the *Sketching Phase*, participants were asked to sketch ideas for introducing *TextFlow* in 15 minutes. They were encouraged to give considerations to (i) convey the insight to the people with less experience in data visualization; (ii) organize a clear narrative structure; (iii) think about additional annotation and animation required. Participants were asked to think aloud in this phase.

In the *Preparing Phase*, the participants got familiar with the authoring tools they would use in following the *Authoring Phase*. For PEs, we demonstrated the working flow of Narvis through an automatic step by step tutorial included in Narvis. For PCs, we asked them a series of questions to make sure that they were familiar with authoring PowerPoint. This phase took about 10 mins for both groups.

In the *Authoring Phase*, participants implemented the ideas in their sketches in one hour to produce a narrative slideshow that is self-explanatory. We sent each participant a PNG file and a txt file as the raw material for authoring. PCs used PowerPoint while PEs used Narvis as their authoring tools. Both PEs and PCs were allowed to ask questions about their authoring tools in during authoring.

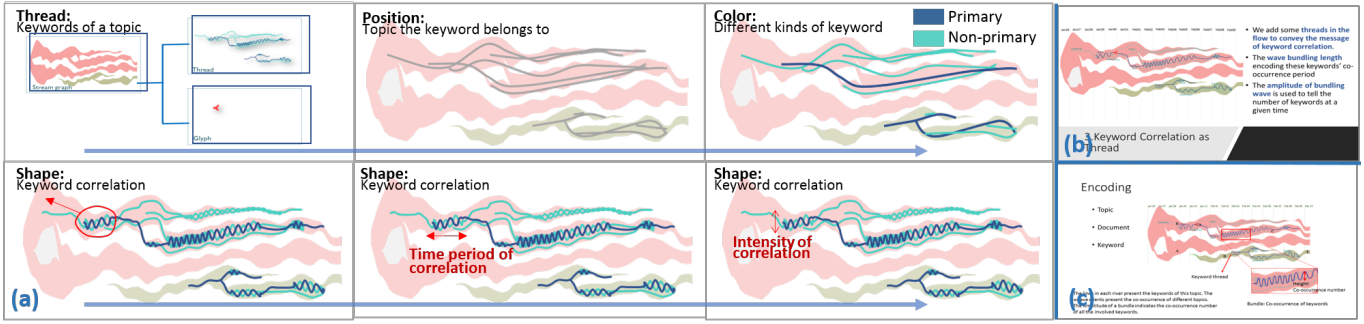


Fig. 8. The slideshows produced by (a) Narvis and (b),(c)Power Point to introduce a visual unit (thread) in *TextFlow* [14]. Note that (b) and (c) both miss the visual grammar of thread color and (c) forgets to mention the visual grammar of wave bundling length.

Table 3. A summary of 4 slideshows

	SE1	SE2	SC1	SC2
Number of Slides	29	22	7	3
Average Reading Time(Total,s)	327.05	156.78	169.33	128.84
Average Reading Time(Per Slide, s)	11.27	7.09	24	42
Information Missing (in Slideshow/in Sketch)	1/4	0/3	2/3	2/2
Average Length of Text (per Slide)	10.7	12.3	32	47

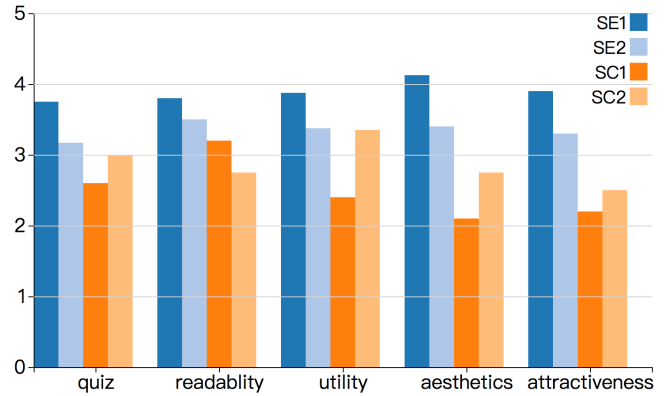


Fig. 9. Comparing the PA's scores in the quiz and their rating from four aspects, i.e., readability, utility, aesthetics, and attractiveness, after watching SE1, SE2, SC1, SC2, respectively.

6.3.2 Evaluation Methods

The evaluation focused on two parts, the authoring experience of Narvis and the quality of the generated slideshows. We reported our observation of the authoring experience based on an interview with the two PEs and the video we took during the authoring session. To evaluate the slideshows generated from both groups, we first analyzed the slideshows and reported some quantitative observations, such as the number of slides. Then, to get an independent opinion, we asked 20 PAs to evaluate the generated slideshows. To eliminate the bias introduced by irrelevant variables (e.g. watching the slideshow in Narvis or in PowerPoint), this part was conducted in a website we built. Each PA was randomly assigned a slideshow when visited this website. They watched all the slides by clicking two buttons, “next” and “previous”, and their click activities were automatically recorded by a background program. After watching the slideshow, they were asked to finish an online questionnaire composed of 2 parts: 1) a quiz about the visual design of “TextFlow” (with a full mark of five); 2) rate the slideshow from one (very poor) to five (excellent) at various aspects.

7 RESULTS

7.1 Generated Slideshow

We obtained four slideshows, two from the control group, denoted as SC1 and SC2, and two from the experiment group, denoted as SE1 and SE2 (see in Figure 8). We evaluated these four slideshows from our observation and the independent opinions from PAs.

7.1.1 Observation from Experimenters

We reported our observation of the 4 slideshows based on: 1) the generated slideshows and their sketches; 2) the videos and notes we took during experiments; 3) the click activities of the PAs; 4) interviews with PEs.

SE1 and SE2 are similar since they were conducted with the same templates in Narvis. However, SE1 included all animations in templates

while SE2, whose creator preferred an abstract introduction, had no animations. SC1 explained the visual design with long, detailed textual description that was formatted with bullet points. SC2 mainly used symbol-based annotations for explaining, and re-edited the image we offered in PowerPoint. Figure 9 gives a quantitative report of the four slideshows.

We listed all the visual grammars applied in “TextFlow” and compared with the sketches from editor participants. Information omission occurred at all four sketches, even though their creators were given the freedom to check with the provided material. For example, three sketches (sketches for SE1, SE2, and SC1) failed to mention the visual grammar of the glyph size, two sketches (sketches for SE1 and SE2) omitted the visual grammar of threads color. Compared with their sketches, SE1 and SE2 corrected three out of four, three out of three omissions, respectively. The editors, PE1 and PE2, noticed these omissions when deleting unemployed visual channels in the *Unit Panel*. They hesitated whether to delete a certain channel and went back to check with the provided materials. For SC1 and SC2, only one omission in the sketches got noticed and corrected by its editor while the others remained the same.

With the same authoring time, SE1 had 29 slides, SE2 had 22 slides, SC1 had 7 slides, and SC2 had 3 slides. PC2 (the creator of SC2) spent most of the time to add symbol-based annotation, edited the image to realize techniques such as zoom-in, which was a time-consuming operation in PowerPoint, thus had little time to organize the textual annotation. Huge blocks of text were put arbitrarily in SC2.

7.1.2 Evaluation from PAs

Figure 9 presents the results of the questionnaire. PAs showed a strong preference for the slideshows produced with Narvis. Meanwhile, PAs

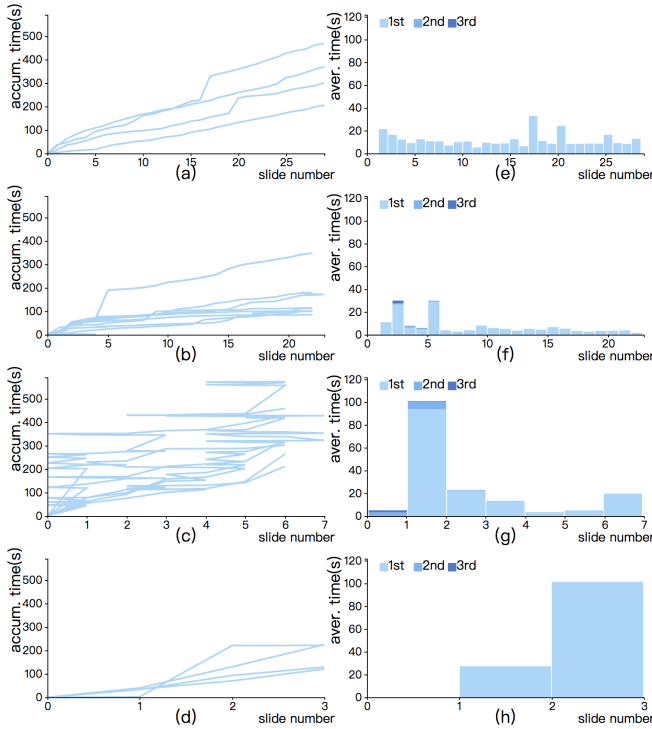


Fig. 10. The visualization of clickstream data when PAs watch the slideshows created by PE1(a)(e), PE2(b)(f), PC1(c)(g), and PC2(d)(h), respectively.

that watched the slideshows produced with Narvis got a higher average score in the quiz, confirming Narvis’s capacity for producing a comprehensive slideshow. For SEs, the PAs were excited about the animations applied and one PA even asked us about the source code. No complaint was made about the relatively long watching time for SEs, which might be due to “the transition is smooth and the structure is well organized” (from one PA) and the short staying time at each slide. PEs also appreciated that textual descriptions were brief and were separated into different slides. We got valuable suggestions for the improvement of Narvis, such as the inclusion of interaction and the implementation of a progress bar to demonstrate when an animation ends.

For SCs, PAs complained about the huge blocks of text frequently. One PA commented that “it is hard to read and I have to admit I skipped some parts, thus still confused about this design.” They enjoyed the symbol-based annotation and the way the creator edited the image in SC2. However, such operation was time-consuming in PowerPoint, resulted in a short, unfinished slideshow with unformatted text.

7.2 Authoring Experience of Narvis

All 2 PEs were impressed by the overall Narvis design, mentioning that the workflow was intuitive and that the *Tree Panel* is inspiring. They confirmed that they were able to craft a slideshow with Narvis after a short training period, and understood the purpose and main capacity of each panel. They appreciated the way Narvis reduced the workload for crafting an engaging slideshow through providing various templates. They showed special interest in the decomposition animation and the morphing transition, thought such animations would be effective methods to attract the general public and to facilitate the perception process of complex visual designs. Their feedbacks for the click stream data were highly positive. One PE quickly identified the problem of his slideshow with the visualized click stream data. “Some audiences went back to check the second slide when watching the third one, which reminded me that the contents in these two slides failed to keep consistent”.

They also offered suggestions for improving the authoring experience of Narvis. They pointed out that the interaction in the *Source Panel* should be more smooth. “I can notice the delay the system needed to respond the mouse event and highlight the related graphical elements”, one PE commented. We plan to solve this problem by implementing a more robust algorithm.

8 LIMITATION AND DISCUSSION

Results of our study indicate that editors could make slideshow more efficiently with Narvis than with professional software. In addition, audience report that the perceived quality of slideshow generated with Narvis is higher than those created with Other tools. However, we identify four limitations existed in Narvis.

First, the library of templates cannot cover all innovative designs. In the current prototype, we allow users to extend existing templates to alleviate this issue. Second, Narvis is limited for static visual design in current prototype. We plan to extend Narvis for interactive visualizations in future research. Further, the results of decomposition highly depend on the quality of the input image. Thus, users are required to import images with high resolution. The third limitation of our study is the evaluation. As all user studies are small in sample size, these results should be treated with caution and Narvis warrants further evaluation to confirm our initial findings. Another limitation in the evaluation is the metrics for assessing the quality of slideshow, which remains an opening question. Our study makes an attempt to assessing its quality but does not delve into all relevant metrics (e.g. memorability).

9 CONCLUSION AND FUTURE WORK

We have presented Narvis, an authoring tool to generate slideshow for explaining visual designs. The design and implementation of Narvis is guided by a constructive model we proposed for introducing visual designs, as well as the collaboration with end users. Narvis allows users to create a self-explanatory slideshow through assembling various templates, supports easy creation of animations tailored for visual designs, suggests design options based on content, and results an efficient crafting process. The user studies have confirmed the utility and effectiveness of Narvis. To the best of our knowledge, this is the first framework designed for presenting visual designs. Though Narvis is designed for making introduction slideshow, it can be generalized to create other forms of presentation, such as narrative posters, comic strips, and data videos, and to produce self-explanatory tutorial, which is in high demand because of the rapid growth of visual analysis tools.

In future work, we envision three main research directions. First, we will dive into click behaviors of audiences to broaden our understanding of audience, and the way of making compelling and comprehensive introduction slideshows for visual design. Second, with the popularity of visualization toolkits, such as D3 [6], an increasing number of visual designs are deployed online. To support a border usage scenario, we plan to equip Narvis with the ability to parse and analyze online visualizations. Third, apart from visualizing the click stream data, we aim at offering suggestions for editors to revise their slideshow based on user behavior.

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. doi: 10.1145/2702123.2702431
- [2] F. Amini, N. H. Riche, B. Lee, A. Monroy-Hernandez, and P. Irani. Authoring data-driven videos with DataClips. 23(1):501–510. doi: 10.1109/TVCG.2016.2598647
- [3] J. Bertin. *Semiology of graphics: diagrams, networks, maps*. 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.
- [5] M. Bostock and J. Heer. Protovis: A graphical toolkit for visualization. 15(6):1121–1128. doi: 10.1109/TVCG.2009.174

- [6] M. Bostock, V. Ogievetsky, and J. Heer. D³ data-driven documents. *IEEE transactions on visualization and computer graphics*, 17(12):2301–2309, 2011.
- [7] C. Bryan, K. L. Ma, and J. Woodring. Temporal summary images: An approach to narrative visualization via interactive annotation generation and placement. PP(99):1–1. doi: 10.1109/TVCG.2016.2598876
- [8] N. Cao, Y. R. Lin, X. Sun, D. Lazer, S. Liu, and H. Qu. Whisper: Tracing the spatiotemporal process of information diffusion in real time. 18(12):2649–2658. doi: 10.1109/TVCG.2012.291
- [9] M. Chapman. *Constructive Evolution: Origins and Development of Piaget's Thought*. Cambridge University Press. Google-Books-ID: 7WgC-nXmdX1MC.
- [10] Q. Chen, Y. Chen, D. Liu, C. Shi, Y. Wu, and H. Qu. PeakVizor: Visual analytics of peaks in video clickstreams from massive open online courses. 22(10):2315–2330. doi: 10.1109/TVCG.2015.2505305
- [11] M. T. Chi, S. S. Lin, S. Y. Chen, C. H. Lin, and T. Y. Lee. Morphable word clouds for time-varying text data visualization. 21(12):1415–1426. doi: 10.1109/TVCG.2015.2440241
- [12] W. S. Cleveland and R. McGill. Graphical perception: Theory, experimentation, and application to the development of graphical methods. 79(387):531–554. doi: 10.2307/2288400
- [13] N. Cohn. Visual narrative structure. 37(3):413–452. doi: 10.1111/cogs.12016
- [14] 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. 17(12):2412–2421. doi: 10.1109/TVCG.2011.239
- [15] L. S. Cunningham and J. J. Reich. *Culture and Values: A Survey of the Humanities*. Cengage Learning. Google-Books-ID: wXcEAAAQBAJ.
- [16] H. Doleisch, M. Gasser, and H. Hauser. Interactive feature specification for focus+ context visualization of complex simulation data. In *VisSym*, vol. 3, pp. 239–248, 2003.
- [17] M. Drk, D. Gruen, C. Williamson, and S. Carpendale. A visual backchannel for large-scale events. 16(6):1129–1138. doi: 10.1109/TVCG.2010.129
- [18] 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. doi: 10.1109/VAST.2007.4388992
- [19] J. Fulda, M. Brehmel, and T. Munzner. TimeLineCurator: Interactive authoring of visual timelines from unstructured text. 22(1):300–309. doi: 10.1109/TVCG.2015.2467531
- [20] N. Gershon and W. Page. What storytelling can do for information visualization. 44(8):31–37. doi: 10.1145/381641.381653
- [21] 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. doi: 10.1145/2642918.2647411
- [22] J. Heer and G. Robertson. Animated transitions in statistical data graphics. 13(6):1240–1247. doi: 10.1109/TVCG.2007.70539
- [23] 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. doi: 10.1145/1284420.1284427
- [24] W. Huang, C. L. Tan, and W. K. Leow. Model-based chart image recognition. In *International Workshop on Graphics Recognition*, pp. 87–99. Springer, 2003.
- [25] D. E. Huber and C. G. Healey. Visualizing data with motion. In *VIS 05. IEEE Visualization, 2005.*, pp. 527–534. doi: 10.1109/VISUAL.2005.1532838
- [26] J. Hullman, S. Drucker, N. H. Riche, B. Lee, D. Fisher, and E. Adar. A deeper understanding of sequence in narrative visualization. 19(12):2406–2415. doi: 10.1109/TVCG.2013.119
- [27] 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. doi: 10.1145/2598510.2598566
- [28] S. Huron, R. Vuillemot, and J. D. Fekete. Visual sedimentation. 19(12):2446–2455. doi: 10.1109/TVCG.2013.227
- [29] L. Itti and C. Koch. Computational modelling of visual attention. *Nature reviews neuroscience*, 2(3):194–203, 2001.
- [30] R. Kosara and J. Mackinlay. Storytelling: The next step for visualization. 46(5):44–50. doi: 10.1109/MC.2013.36
- [31] R. Kosara, S. Miksch, and H. Hauser. Focus+ context taken literally. *IEEE Computer Graphics and Applications*, 22(1):22–29, 2002.
- [32] K. Kucher and A. Kerren. Text visualization techniques: Taxonomy, visual survey, and community insights. In *Visualization Symposium (PacificVis), 2015 IEEE Pacific*, pp. 117–121. IEEE, 2015.
- [33] B. Lee, R. H. Kazi, and G. Smith. SketchStory: Telling more engaging stories with data through freeform sketching. 19(12):2416–2425. doi: 10.1109/TVCG.2013.191
- [34] B. Lee, N. H. Riche, P. Isenberg, and S. Carpendale. More than telling a story: Transforming data into visually shared stories. 35(5):84–90. doi: 10.1109/MCG.2015.99
- [35] T. Munzner. *Visualization Analysis and Design*. CRC Press. Google-Books-ID: dznSBQAAQBAJ.
- [36] 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. doi: 10.1145/2858036.2858435
- [37] H.-C. Nothdurft. Saliency from feature contrast: variations with texture density. 40(23):3181–3200. doi: 10.1016/S0042-6989(00)00168-1
- [38] Y. Pinto, C. N. Olivers, and J. Theeuwes. Selecting from dynamic environments: Attention distinguishes between blinking and moving. *Attention, Perception, & Psychophysics*, 70(1):166–178, 2008.
- [39] G. Robertson, R. Fernandez, D. Fisher, B. Lee, and J. Stasko. Effectiveness of animation in trend visualization. 14(6):1325–1332. doi: 10.1109/TVCG.2008.125
- [40] P. Ruchikachorn and K. Mueller. Learning visualizations by analogy: Promoting visual literacy through visualization morphing. 21(9):1028–1044. doi: 10.1109/TVCG.2015.2413786
- [41] A. Satyanarayan and J. Heer. Authoring narrative visualizations with ellipsis. 33(3):361–370. doi: 10.1111/cgf.12392
- [42] 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. doi: 10.1145/2047196.2047247
- [43] J. N. Schmidt. the living handbook of narratology.
- [44] E. Segel and J. Heer. Narrative visualization: Telling stories with data. 16(6):1139–1148. doi: 10.1109/TVCG.2010.179
- [45] D. I. Standage, T. P. Trappenberg, and R. M. Klein. Modelling divided visual attention with a winner-take-all network. 18(5):620–627. doi: 10.1016/j.neunet.2005.06.015
- [46] M. Waldner, M. L. Muzic, M. Bernhard, W. Purgathofer, and I. Viola. Attractive flicker #x2014; guiding attention in dynamic narrative visualizations. 20(12):2456–2465. doi: 10.1109/TVCG.2014.2346352
- [47] 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. doi: 10.1145/3002151.3002155
- [48] J. M. Wolfe. Guided search 2.0 a revised model of visual search. 1(2):202–238. doi: 10.3758/BF03200774
- [49] Y. Wu, S. Liu, K. Yan, M. Liu, and F. Wu. OpinionFlow: Visual analysis of opinion diffusion on social media. 20(12):1763–1772. doi: 10.1109/TVCG.2014.2346920
- [50] Y. Wu, F. Wei, S. Liu, N. Au, W. Cui, H. Zhou, and H. Qu. OpinionSeer: Interactive visualization of hotel customer feedback. 16(6):1109–1118. doi: 10.1109/TVCG.2010.183
- [51] J. Zhao, N. Cao, Z. Wen, Y. Song, Y. R. Lin, and C. Collins. #x0023;FluxFlow: Visual analysis of anomalous information spreading on social media. 20(12):1773–1782. doi: 10.1109/TVCG.2014.2346922