Narvis: Authoring a Narrative Slide Show for Introducing Data Visualization in A Constructing Way

Category: Research

Paper Type: please specify

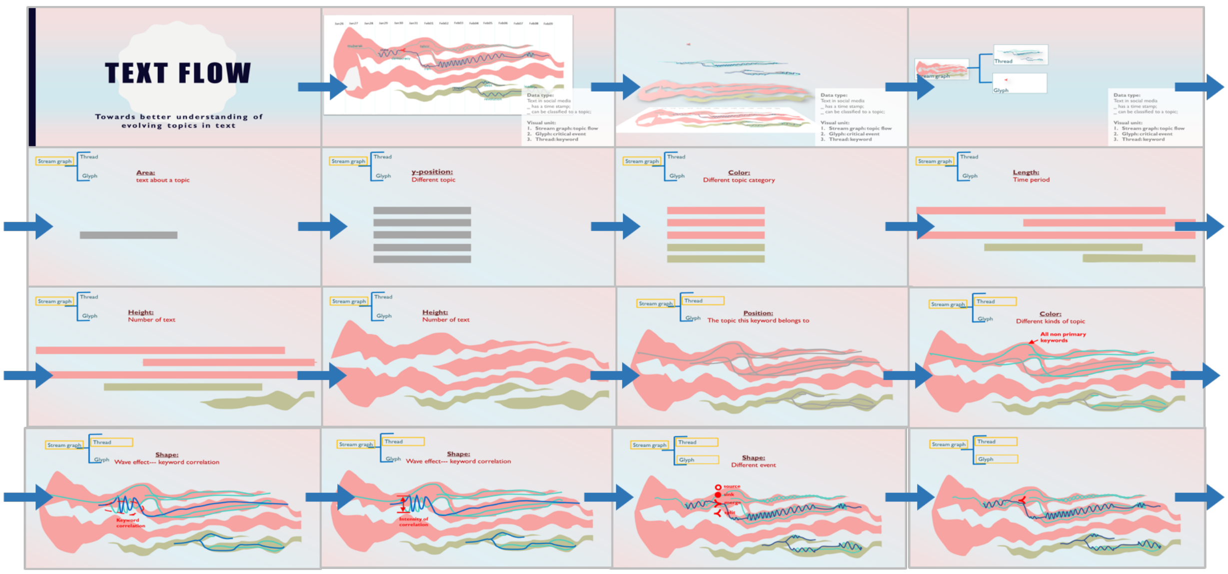

Fig. 1. Example of an introduction slide show of *TextFlow* [14] generated by an expert in data visualization using Narvis. This slideshow consist of (a) a cover, (b) a decomposing animation, (c) introducing the design in a constructing way.

Abstract— The democratization of data visualization is readily motivated by current data deluge. However, learning a visualization is error prone especially when the design is complicated. 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 a narrative slide show 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 slide show, we incorporate lessons from previous work with our observation and propose a hierarchical constructing model, which is 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 for explaining different components. Guided by this model, we implement a library of templates in Narvis. It enables the editors crafting an introduction slide show through assembling these templates, thus achieves a level of expressiveness while improving efficiency. 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—User Interface, Visualization System adn Toolkit Design,

1 Introduction

For data with complicated structure, naive data visualization like bar chart and pie chart maybe unsatisfying for a comprehensive display. By introducing metaphors borrowed from nature [7, 25], applying carefully designed layout algorithms [10, 43], and sophisticatedly combining existing visualizations [45], novel visual presentations help people identify patterns, trends and correlations hidden in data. However, these advanced visualizations are usually not intuitively recognizable. Users need to go through some training, for example, reading a long and boring literal description, before they grasp the knowledge required to understand and freely explore a visualization.

What is more, even designers of these advanced visualizations suffer when they are required to introduce their design, especially when the visual encoding has complicated logic dependency, or when their audience have little prior knowledge about visualization techniques.

As a result, these advanced visualization technologies, in spite of the fact that their utility has been verified by domain experts from various fields, gain little exposure outside the visual community. Main stream media is still dominated by naive visualizations, such as bar charts, pie charts and so on.

For a visualization, its core design space can be described as the orthogonal combination of two aspects: graphical elements called marks and visual channels to control their appearance [29]. But why the explanation of these two things is so complicated?

This problem mainly arises from the fact that advanced visualization designs usually attempt to delivery a great amount of information. First, it would overload an audience if we inundated them with all the information at one time. Second, even if we tried to explain it sequentially, considering the logic dependency existing among visual elements, an improper explanation could totally confuse the audience. For example, in a node link diagram, a node should be introduced before the links connecting it. In an advanced visualization design, which has more components than just nodes and links, it is challenging to identify a proper explaning oder. Third, when digesting such a considerable amount of information, audiences can easily get distracted or forget previous information.

Thus, to better introduce an advanced visualization, we should convey its information sequentically and in a specific order. Attention guidance and reminders are also needed to make sure that audiences are following this order, not getting distracted or forgetting previous information.

Narrative, which means connected events presented in a sequence, has long been used to share complex information. [37]As the data visualization field is maturing, many researchers have moved their focus from analysis to presentation, making narrative data visualization an emerging topic [26]. Many efforts have been made to define, classify, and provide design suggestions for narrative data visualization [17, 23, 38]. Some visualization systems have already incorporated narrative modules into their design [6, 15]. However, current work is mainly focused on communicating the conclusion of analyses, rather than guiding the audience how to read a visualization.

Here, we present a prototype to introduce new visualization design. Based on our analysis of the structure, logic dependency, and visual distraction existing in a visualization design, we develop an authoring tool to decompose a visualization, reorganize extracted visual elements, and explain their visual encodings one by one through animated transition in the form of slideshow. Through incorporating a narrative sequence, appropriate chunks of information, rather than all the information, is delivered to the audience at one time, effectively avoiding information overload. Reminders, such as questions, summarizations and repetitions are woven into the narrative sequence to enhance the audiences memory while visual attention guidance, such as flickering, highlighting, and morphing are used to lead their attention to newly added information. ()

To the best of our knowledge, this is the first attempt to explain visual encoding with narrative. Our contributions are as below: 1). A paradigm for decomposing visualizations. It analyzes the hierarchical structure of its components, the relationships between components, and visual distraction existing. 2) A framework for explaining visualization 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 visualization 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 visualization designs, supporting the democratization of data visualization.

2 Related work

In this section, we provide an overview of prior research around 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. [cite something] People in the fields of literature, comics [12] and cinema [37] have gone to great lengths to analyse the sequencing and forms of grouping used in a narrative, as well as how they affect the meaning a narrative tries to deliver.

Some people believe that work from other fields can inspire researchers in the visual data community. Amini et al [1] borrow concepts from comics [12] to classify and analyse the structure of data videos. Wang et al [41] adopt two representative tactics, time-remapping and foreshadowing, from cinematographers to organize a narrative sequence for visualizing temporal data.

Other researchers, on the other side, focus on the narrative structures exclusively for data visualization.

Satyanarayan and Heer, through interviews with professional journalists [34], define the core abstractions of narrative data visualization as state-based scenes, visualization parameters, dynamic graphical and textual annotations, and interaction triggers. Hullman et al [23], by identifying the change in data attributes, 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 visual encoding scheme, which is fundamental to a visualization. We hope our work can fill this gap.

2.2 Animation for data visualization

There is a wide discussion about the effects of animation when used in a data visualization environment.

Animation can facilitate the cognitive process. Heer and Robertson [19] confirm the effectiveness of animation when relating data visualizations backed by a shared dataset. Ruchikachorn et al [33], going a step further, design morphing animations which bridge the gap between a familiar visualization and an unfamiliar one, thus introducing a new visualization design through animation. Graphdiaries [3] uses animation to help audiences track and understand changes in a dynamic visualization.

On the other hand, animation can be an effective tool to attract and guide visual attention. Huber et al [22] study the perceptual properties of different kinds of animation, as well as their effects on human attention. Waldner et al [40] focus on a specific animation: flicker. By dividing the animation into an orientation stage and an engagement stage, they strike a good balance between the attraction effectiveness and annoyance caused by flickering.

It is, however, noteworthy that animation, in spite of all the advantages mentioned above, can bring about negative effects when used improperly [32]. Our work is based on the results of these researches, which provide a guideline on how to implement animations in our system.

2.3 Authoring tools for narrative visualization

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

User experience is of great concern when utilizing an authoring tool. Sketch story [27], with its freeform sketch interaction, provides a more engaging way to create and present narrative visualization. Dataclips [2]lowers the barrier of crafting 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, 16]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. [6, 15, 28]

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, making them inapplicable in our case.

2.4 Decompose a data visualization

Clarifying the design space of a data visualization can help people get a better understanding of how it is constructed. Tamara [29] 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 [24] 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, although varying from one to another, motivate the designers of visualization tools to contribute efficient high-level visualization systems rather than low-level graphical systems. [5, 30]

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 [18] contribute a tool that extracts visual variables from existing D3 visualization designs to generate a new design.Huang et al [21] propose a system that recognizes and interprets imaged infographics from a scanned document. Revision [36] applies computer vision method 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 advanced visualization designs, which assemble miscellaneous visualization approaches to realize a novel presentation. Moreover, these methods are put forward for the purpose of constructing a visualization, instead of explaining an already existing one, 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 method that introduces a data visualization through constructing, which has been proven as an effective teaching method [8, 24] . Thus, 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 the 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.

3.1 Compositions of a Visualization

We propose a model that decomposes a visualization into three levels of structure: visual primitives, visual units, and then an advanced visualization design. We apply this hierarchical structure theory to "Opinionseer" and decompose it, as shown in Fig.2.

A visual primitive is one graphic element whose visual channels, such as color, width, height, are mapped to data attributes. We employ the definition in previous work [24, 35], use the term "grammar" to describe how the visual channels of a visual primitive is influenced by data. For instance, a point whose size and color are encoded is a visual primitive. How the two visual channels, size and color, are related to data attributes is its visual grammar.

A visual unit is the assembly of visual primitives based on a certain construction rule, as tab.1 show. A visual primitive can assemble different visual units by following different constructing rule, as demonstrated in Fig.3. We are not pretending that our table includes all existing visual units, since new design is proposed constantly. A visual unit is the smallest functional unit of a visualization. Note that people might employ two visual primitives in a visualization design. For example, *Visual Sedimentation* [25] employs two visual primitives, bar and dot, to construct a novel design.

A visualization can be treated as the combination of visual units. An naive visualization can be as simple as one visual unit while an advanced one is usually the combination of several units. It doesn't simply put all visual units together but construct them with certain connections with each other, which is detailedly discussed in section 3.1.2.

3.2 Relationships between compositions

3.2.1 Relationships between Visual Units

An advanced visualization can be specified as the combination of several visual units. Through observing the approaches people apply to design new visualizations, we define four types of relationship between visual units: irrelevance, relevance, enhancement, and dependency.

Irrelevance refers to that two visual units have no correlations in the visual grammar. It is a bi-directional relation. For example, 2 donut charts, Fig.4(a) and Fig.4(b), are applied to illustrate the distribution of age and gender groups respectively in a population. They are put together in Fig.4(c) just for space-efficiency and there is no correlation between these two charts.

Relevance refers to that two visual units share some visual grammar and it is a bi-directional relation. For example, a line chart, Fig.4(d), and a bar chart, Fig.4(d), share the same encoding of horizontal position and they are put together in Fig.4(e). According to our survey, color and position are the most commonly shared visual encodings, which might be the result that color and position usually encoded with simple while crucial information.

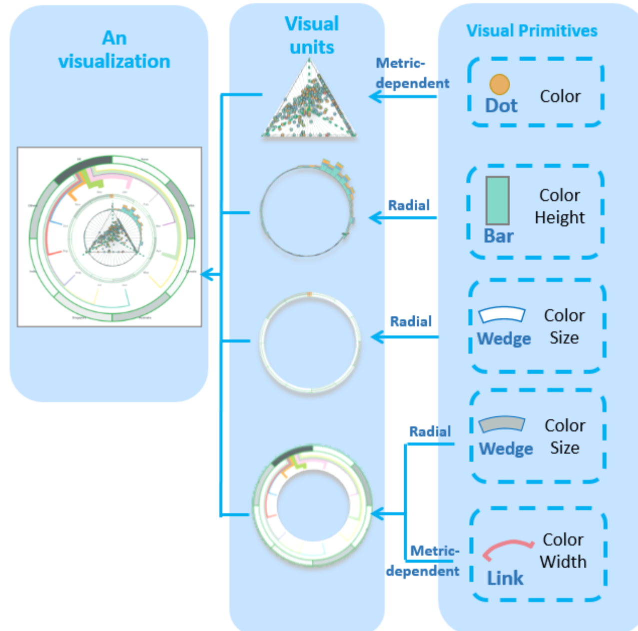

Fig. 2. An example of the hierarchical structure of a visualization, Opinion Seer [44]

Enhancement is an one-way relationship. If one visual unit "A" is the enhancement of another visual unit "B", it means that "A" is imported into "B", replaces some graphical elements of "B", thus enables the representation of some data attributes that "B" alone fails to convey. Suppose there are 5 types of area in a park. A bar chart, Fig.4(h), illustrates their average price per unit area, a chord diagram, Fig.4(g), illustrates how passengers travel through each area. In Fig.4(i), the bar chart take the place of node segments in a chord diagram, resulting in a novel and informative visualization . Some actual examples are the heat map mapped upon the steams in a theme river [43] and usage of glyphs to enhance the meaning of nodes in a multidimensional scaling plot. [9]

Dependence is an one-way relationship. If one visual unit "A" is dependent on "B", it means that "A" reveal some information that results from the visualization of "B". For example, a multiple dimensional scaling (MSD) map, Fig.4(j), shows the similarity between each restaurants in a city. A heat map, Fig.4(h), is then added to the MSD map to show the most common type of restaurants, which information can hardly be obtained from the dataset but quite evident from the MSD map, as in Fig.4(i). The biggest difference between "**enhancement**" and "**dependence**" is that **enhancement** still illustrate the data attributes in the dataset, while **dependency** reveals the new knowledge we obtain from adopting a previous visualization to the dataset.

3.2.2 Relationships Between Visual Primitives

The inner relationship between visual primitives is relatively simple. A visual unit usually has 1 or 2 visual primitives.

The relationship between the 2 visual primitives, if there are two, are usually self-evident. One primitive either has no dependency on the other one, such as the bar and line in *Candlestick Chart*, or has high and evident dependency. This dependency can be logical, such as the line and bar in *error bar chart*, or spacial, such as the node segment and arc in *chord diagram*, or temporal, such as the stream and dot in **[?]**

3.2.3 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

Table 1. A taxonomy of visual units.How to avoid the name ambiguities

	Polar Coordinates		Orthogonal Coordinates			Metric Dependent		
	Radial	**Spiral**	**Orthogonal**	**Parallel Align**	**Map**	**Cluster**	**Force-direct**	**Others**
Dot		Spiral Dot Chart	Scatter Plot, Bubble Chart	Dot Plot	Bubble Map	Circle packing	TopicPanorama [?]	
Line	Radar Chart	Spiral Plot	Node-link Diagram, Line Chart	Parallel Coordinates, Arc Diagram				
Flow	Chord Diagram			Parallel Sets, Sankey Diagram	Flow Map			
Area		Area Spiral Chart	Stream Graph					
Bar	Radial Bar Chart	Spiral Bar Chart	Candlestick Chart	Bar Chart				
Cell	Sunburst Diagram		Matrix, Tree Map					
Wedge	Pie Chart, Donut Chart							
Text		Parallel Tag Cloud [13]		Sentence Tree		Word Cloud		

Fig. 3. Illustration of the 4 types of relationship between visual units

dependency upon others.

3.3 Design considerations of narrative sequence

3.3.1 Narrative Sequence for Visual Channels

An arbitrary explaining order of visual channels may lead to an inefficient information delivery, especially when this visual primitive has multiple channels, which is common in an advanced visualization.

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, a proper explanation should follow the order of decreasing visual saliency. [11] Even though different channels have intrinsically different perceptual salience and channel with high salience will suppress the expression of other, such salience strength can be influenced in a task-dependent manner. [31] By introducing the channel with high saliency first, we remove it from the task list in our mind [**?**], decrease its saliency and give other channels more chance to attract limited human attention.

Second, we should follow the order of increasing complexity. 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 [11, 29], while the information complexity varies in different designs.

3.3.2 Narrative Sequence for Visual Primitives

As discussed in section 3.2.2, relationships between visual primitives are self-evident, thus need no special consideration for the narrative sequence.

3.3.3 Narrative Sequence for Visual Units

As discussed in section 3.1.2, there are four types of relationships between visual units, and they will influence the order of a narrative explanation. Thus, we display the correlations between units in a tree diagram where a child node is the enhancement/dependence of its parent node and sibling nodes have logic dependences. as show in FigWhen explaining these visual units, we can simply follow a deep first search (DFS) order to visit all the visual units.

3.3.4 Non-linear Sequence

So far, all the narrative explanation we discussed is linear. However, reading a lengthy, extremely detailed instruction maybe tedious. A good narrative explanation should include non-linear design, allowing users to skip uninterested parts, go back to previous information and freely switch between different parts. Also, users should be allowed the flexibility to choose explanations at different levels of details.

3.4 Attention Orientation

To keep audience focus on the target object, it is necessary for us to identify existing visual distraction and take measures to suppress them. We identify two kinds of visual distractions: visual distraction from context and visual distraction from sibling channels (sibling channels refer to the channels belonging to the same visual primitives).

3.4.1 Visual distraction from the context

This kind of distraction has been widely discussed in the field of object detection and human visual attention. [31, 39]Its intensity is mainly determined by spatial distance and appearance similarity. [42] For example, when we try to focus on a green rectangel, a red triangel near by it can lead to visual distraction. And the intensity of such distraction is determined by the distance and the appearance similarity between the two graphics. Do i need this example 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 [40].

3.4.2 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 focus is supposed to be the shape. By applying animated transition and revealing only one channel at a time, as demonstrated in Fig1, the second line, we are able to reduce such distraction.

4 Design Considerations

In this section, we first describe our understanding of two groups of end users, i.e., editors and general audience. Then, we distill 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 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 communicate visual design. General audience have no prerequisite for visualization. They gain understanding of the visual design by watching the slideshow.

To understand the current practice of making slideshows and the experience of reading tutorials, we collaborated with two teaching assistants (TAs) of a Data Visualization course and seven undergraduate students (UGs) taking this course. The two TAs are postgraduate students whose research interest is information visualization. Their duty of this course involves making slides for explaining visual design appears in major publications in the field of data visualization. The slides should cover fine-grained description to help students review them after class. The seven UGs have little background in visualization before the class, and have taken this course for less than one month.

We began by conducting semi-structured interviews with TAs, whom we identified as editors, and UGs, who are general audience. 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. Two tasks, denoted as DE, are originated from the interview with editors, i.e., the two TAs, and other four tasks (DA) are from audience (UGs).

DE1. Keep efficiency. TAs used presentation tools, such as Power Point[1] and KeyNote[2] to introduce visual design. However, these tools are for general purpose and not tailored for visualization. Splitting a visualization into fine-grained graphical elements and combining them into logic flow are tedious and time-consuming with existing tools. Therefore, automatic decomposition of visualization and composition of graphical elements are necessary for facilitating design and production of a slideshow.

DE2. 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. However, Collecting feedbacks of audience is crucial for editors to revise their slideshow, making it more understandable and attractive.

[1]https://office.live.com/start/PowerPoint.aspx

[2]http://www.apple.com/keynote/

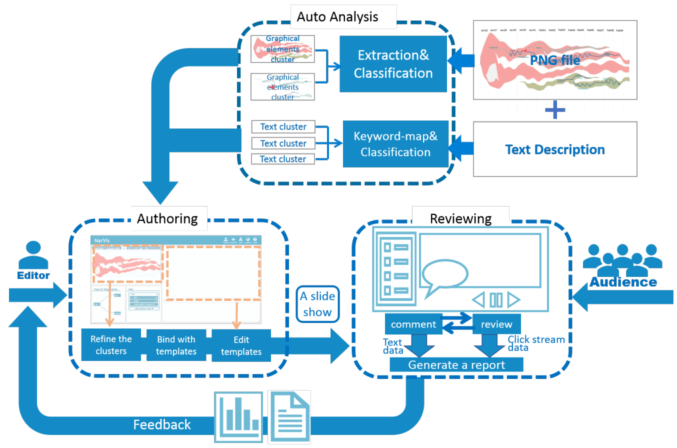

Fig. 4. The system overview

DA3. Avoid information overload. All UGs complained that they had experienced information overload in reading slides. Though they might separate an informative slideshow into parts and read one part at a time, they got lost if the information in one slide is overloaded. To reduce mental effort of audience, one slide should limit the amount of information, revealing one visual grammar of one graphical element.

DA4. Avoid unconscious ignorance. Experts in data visualization prone to treat some visual encodings as straightforward, and unconsciously ignore some crucial encodings when presenting a visualization. However, the lack of information confuses the UGs with no prior knowledge of visualization. A comprehensive slideshow is crucial to communicate a visual design to novice.

DA5. Keep logic flow. The complicated relationship, spatial and logical alike, between different graphic elements is one of the biggest barriers that impedes a smooth communication of visual encoding scheme.

DA6. Emphasis on conveying intuitive concepts. Although algorithm might be crucial for the achievement of a visualization design, some of the interviees show little interest in it.

5 Narvis: System Design and Implementation

With the considerations above in mind, we implement the workflow of Narvis consisting of three phases (Figure ref), i.e., Automatic Analysis Phase, Manual Editing 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). In this phase, Narvis accepts the input image and extracts graphical elements to facilitate further authoring. If text is provided, editors are able to add and edit annotations in the Manual Editing Phase.

5.1.1 Analysis of input image

The analysis of input image includes three steps, i.e., object detection, object clustering and object recovery.

Object detection. Narvis iterates through all pixels in the input image. At every iteration, we first check to see if this pixel has already been tagged as part of an object. If not, we know that this pixel forms a part of a new object. We explore the colors of the neighboring pixels, where the neighbors are chosen such that the distance between the current pixel and a potential neighbor is less than 3. If the difference in color between a neighbor pixel and the current pixel is less than a threshhold, the neighboring pixel is tagged as part of the same object. Once all neighbors have been classified as either part of the same object or not, we choose another pixel that was classified as part of the same object and apply this algorithm again. This is a modified BFS algorithm and allows us to identify all unique ojects in the given visualization.

Object clustering. Once all the objects have been detected, we have to extract a target object. To extract an object means to only select the pixels that are classified as part of this object, so we should remove all objects that are not part of this object and we should extract objects that are inside our target object.It is trivial to set all pixels that are not within or part of our object to have color white. For objects that are inside our target object, i.e. those objects that are clustered with out target object, we will first detect that object then programmatically change its pixels to white.

Object recovery. Once we have completed extraction, we have the issue of these white spaces. The reason this is an issue is because an extracted object might have been dividing two objects, and so when it is extracted, we lose the boundary between our target object and another object, which can cause confusion as to whether that white space should be colored in or not. To solve this boundary problem, we create a queue of the white spaces, with each data point giving the starting and ending point of that space. We then look at the intervals between enclosed white spaced objects, if that interval is above a threshhold, we take that white space to not be part of our object. If it is below our threshhold, then we enclose the white space with the target objects color, creating a boundary for it. The main difference is that for objects not within our target object, we do not create a boundary, whereas objects within our target object are enclosed with the target objects color.

5.1.2 Analysis of input textual description

For the input textual description, we offer a basic text detection and classification algorithm, which uses a dictionary of terms that are highly correlated with certain channels. E.g. the word "length" is highly correlated with the size channel. To do the text detection, we first classify each sentence depending on whether it contains any of the key words in our dictionary. If it contains a key word for one of the channels, the sentence is tagged as being a description of that channels visualization. Once we have tagged all the sentences, whenever a channel is selected, we show the entire text that was inputted and highlight the text that has been tagged as descriptive of that channels visualization.

The algorithm we proposed is a compromise between efficiency and performance. At this time point, it is limited for image with high quality and clear edges, but its performance can be improved by applying a more advanced algorithm, such as the method based on patch detection and clustering mentioned in Revison [36].

5.2 Phase2: Manual Editing

The workflow of this phase includes three steps, i.e., visual unit generation, visual unit organization and template modification, as illustrated in Figure ref. We further introduce a library of templates for sth

5.2.1 Visual Unit Identification

After graphical elements are extracted and clustered based on visual representation, each cluster appears as a tabbed view in the Source Panel (Figure ref). Editors add, delete and modify the graphical elements associated with each tab, making sure that 1) All the graphical elements of the same visual unit is with one tab 2) every graphical element belongs to one and only one tab. For each visual unit, the user call a template from our library in a drop-down list. where is the drop-down list?

5.2.2 Visual Unit Organization

Editors are allowed to adjust the relationships between different visual units througn interacting in the *Unit Tree* panel, where all visual units are shown as tree nodes. With drag and drop, editors organized and display the structure of the visual units, like what we have discussed in section 3.3 (DA5). To help editors better identify the relationship between visual units, which might be a new concept for them, we include a tutorial here. Even though learning the relationship between visual units requires extra effort and time, we believe it is worthwhile since it can give people a better understanding about the structure of a visualization. you spend too much words describing the tutorial. what is it? is it important? Narvis will organize the narrative sequence of visual units based on the tree diagram. sequential order is easy to understand. how about hierarchical order? which do you present first? do you mentioned this point in section 3?

5.2.3 Template Modification

Narvis provides templates to generate slideshows with high efficiency. It also supports flexible modification of templates for enough expressiveness. Editors can edit a template in the *Unit Panle* by selecting a node on the *Tree Panel*. For each visual units, the template enumerates possible encodings and leave the users to delete unused ones, thus eliminating the unconscious missing of crucial information (DA4). It also recommends a narrative sequence based on the metrics we mentioned in section 3.1.4 (DE1, DA3, DA5). In the editor panel, users can further access the *grammar* of each visual primitives, adding a short annotation to describe it (DA6), refine or remove the animation we embedded in a template. add figure reference for each panel

5.2.4 A library of templates

We propose a library of templates for the narrative explanation of a visualization. A templates is a set of slides that tends to introduce a visual unit, which can be descibed as an orthogonal combination of a visual primitive and a construction rule, as shown in tab.1. Since advanced visualization design is the assembly of miscellaneous visual units, we conjecture such templates can achieve a high level of efficiency for the explanation of a visualization. (DA1)Meanwhile, allowing users a high flexible, friendly interface to edit offered templates, Narvis maintains a considerable level of expressiveness and accessibility.

Types of templates

The initial set of templates provided by Narvis can be described as a 9*4 matrix, 9 types of visual primitives and 4 types of construction rules. Narvis is extensible, new templates can be added by its developer through programming, or by end users through uploading their modified templates. At the same time, all the supported templates are classified into a certain cell of the 9*4 matrix, 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 three components: 1) a well-considered narrative sequence for visual grammar explanation, which is discussed in section 3.3 and reveal encoding grammar gradually(DA3); 2) Embedded a series of narrative techniques such as attention cues, animated transitions, information repetition, to orientate visual attention and facilitate perception; 3) Formatted sentence for annotations (DA6) that will be gradually disclosed in the slide show. (DA3)

With a visual unit, more specifically, a set of graphic elements, as input, a templates will generate a series of slide show and each slide is responsible for the explaining of one visual grammar. A visual properity show on a slide only after its grammar has been explained. For example, if we havn't explain the encoding of color, all the object in current slides will be gray. These slides are sorted based on the narrative sequence we discussed in section 3.3. The graphical elements in different slides, which might have different visual appearance, 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(DA1), which has been widely discussed in previous work. [20,32,40]We also provide an novel decomposition animation at the beginning of the introduction slide show to engage the audience as well as to help them get a sense of overview.

Animation is a double-edge sword, which introduces both benefits and pitfalls. We are not discussing the effects of animation here. Editors can choose to remove these animation if they prefer an abstract slide show or they are suspicious of the effects of animation.

Table 2. A summary of animation provided

Animation	Engaging	orientate attention	perception	working scenario	ref
Morphing	✓	✓	✓	grammar of size, grammar of shape	[33]
Blur		✓		focus+context	
Flicker		✓		focus	
Motion	✓	✓	✓	grammar of position	
Zoom-in/out	✓	✓		focus	
Annotation		✓	✓	textual explain	
Fade in/out		✓			
Decompose	✓		✓	Show how a visualization is composed by visual units	A novel design bu us

5.3 Phase3: Viewing

5.3.1 The interface for audience

The interface of audience is composed of two panels.

***Gallery:*the collection of generated slide show**

Gallery exhibit all the slide show produced by editors and saved in Narvis. Every slide show is presented by a image, the visualization it tends to explain. By clicking on the image, users can watch this slide show in the *Screen* panel.

***Screen:* review and comment** Every slide show displayed in *Gallery*is a series of slides, each of which is responsible for the delivery of one simple encoding information, for example, the horizontal position indicates time. In the *Screen* panel, users click buttons to move forward or backward to view these slides.

5.3.2 Generated Report

The report visualize the click activity of audience in the form of a stacked bar chart. The heigh of the bar indicates the time spent on watching this slide. If audiences go back to revist a slide while viewing, a bar will be stacked on the top of previous one. If there are animation in the slide show, a white line will be drawn on the bar chart, indicating the animation playing time of each slide, thus can indicate whether an animation is too fast or too slow. (DE2)

5.4 A working scenario

Jessica has extensive experience in the field of data visualization, and has implement a visual analytics tool in a review service website based on the design of OpinionSeer [44]. To help audience better understand this design, she needs to publish a tutorial accompanied with it.

First, she loads the screen-shot of her system, as well as a piece of textual description, into Narvis. After a few seconds, the system automatically extracts the graphics elements and clusters them based on features. As Figureref shows, Jessica obtains four clusters.

Then, she defines visual units based on clusters. By default, each cluster includes all graphics elements belonging to one visual unit. However, she observes that geographic ring and calendar ring are in the same cluster due to their similar appearance. Therefore, she divides it into two clusters, containing geographic ring and calendar ring respectively.

Next, she chooses narrative templates for each visual unit. Moreover, Jessica edits the narrative templates based on her design. She goes through all four templates in the "what is in-unitin-unit", and deletes the visual channels with no encodings, such as sth. Through drag and drop, Jessica further organizes the structure of the unit tree based on the relationships between units. For example, some example

Jessica further improves the quality of animation by adding annotations and strengthening the binding between data and graphic elements.

To refine the readability of the tutorial, Jessica asks several friends, who have no 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 stacked bar chart , which helps Jessica answer questions like *"which slides do they skip?"*, *"which slides do they review several times?"*, and *"which slides do they stay for a long time?"*.

6 Evaluation

6.1 Participants

There are two kinds of participants, editors and audiences, in our user study.

Editors: they are experts in data visualization. They will be divided into two groups and exploit either Narvis or PowerPoint to generate a slide show that explains a visualization design.

Audience: they have no previous experience in data visualization. A questionnaire is conducted to investigate their knowledge about visualization. They will review the slide show produced by the experts, rank it, give subjective comments, and answer a series of questions to check their understanding of this visualization.

For editors, we have 4 postgraduate students, aging between 22-30, and all of them have more than one year experience in data visualization.

For audiences, we have 20 under graduate students, whose majors vary from business to biology. According to the questionnaire, none of them have accessed advanced data visualization before. Only 13% students know the tree map, and none can give a accurate explanation of theme river with topic splitting and merging.

6.2 Material

We extract the visualization design and the corresponding literature description from a visualization design paper by Cui et al [14]

We choose this visual design based on two considerations. First, it's not too difficult for a laymen but still a novel design that requires extra effect to clarify its encoding scheme. Second, it is a typical abstract data visualization that is fully consist of graphical element, not involving 3D image or real world image such like satellite map, which is beyond the coverage of our edge detection algorithm.

This visualization design is aimed at providing a better understanding about topic evolution in large text collections. It 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 encode glyphs, and the keyword correlations, which encode threads.

6.3 Procedure

6.3.1 Producing

We run a two-hour long sessions, which is consist of 3 phases: (1)*learn visualization*, (2)*idea generation and sketch*, (3)*authoring*.

In the *learn visualization* phase, participants read the literature description we extract from the paper, which is two-page long and describes the visual design with diagrams. This phase ends when the participants report the experimenters that they finished reading and understand this visual design. This phase takes about 15min, since all the participants are experts in data visualization and familiar with reading such papers.

In the *idea generation and sketch* phase, participants are asked to sketch ideas for introducing *TextFlow* to general public. They are encouraged to give considerations to (1) knowledge base of the audience, (2) information complexity of different visual encodings, (3) attention cues to orientate audience's attention. Participants are asked to think aloud and experimenters are present in the room to observe.

In the *authoring* phase, participants implement the ideas in their sketch as many as possible in a one-hour-long session. Participants in control group use Power Point, a presentation making tool that all the participants are familiar with. In experimental group, before authoring, experimenters demonstrate the capacity of Narvis through an automatic step by step tutorial included in Narvis, using intro.js. This training lasts about 15 min and is not counted in the one-hour authoring session. Participants are also allowed to ask additional questions in the authoring phase.

6.3.2 Reviewing and feedback

We conducted a first pilot study to ensure the clarity of the instructions and control the time of experiments.

We asked a group of 20 volunteers to evaluate the quality of the generated slide show. We conducted a questionnaire in advance to make sure that they all have no experience or knowledge in advanced data visualization. In a one-hour session, they are asked to view, comment, and rate these slide shows. They also answer a series of questions to check their understanding of the visualization design.

We record video during this session with the participants permission. For participants who review the slide show generated by Narvis, their click activity will be recorded automatically and they can make comments on the slides. These click stream data, as well as the comments stream, will be used to generate a report, which will then send to its editor.

To conclude the user study, the experimenters conduct an interview with the participants about their authoring experience, the issues they encountered, if there are any, and the feedback report Narvis generated.

6.4 Results

We analyzed the following material: 1) video and notes that the experimenters took during the user study session, which the participants consented to. 2) the slides and the sketch created by participants, 3) the interview with the editor participants, 4) the ranking, comments, answers, click stream data from the reviewer participants. While analyzing, we focus on extracting information on the following aspects: 1)

6.4.1

6.4.2

xuke
reading 15min
draft 5min
making slides 40min
qiaomu
reading 14min
draft 5min
making slide 40min

6.4.3 Generated slideshow

6.4.4 Authoring experience

7 Limitation and Discussion

We are not pretending that Narvis are exclusive for all types of visualization design. However, by allowing users a high flexibility to create and edit templates, we believe its coverage will quickly broaden as more and more users contribute their own templates to our library.

Metaphor for aesthetic purpose. Our algorithm, not applicable for 3d rendering picture. In our model, we focus on statistic image and leave dynamic interaction at this time point, which is an important feature for advanced data visualization design.

8 Conclusion and Future Work

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] B. Bach, E. Pietriga, and J. D. Fekete. GraphDiaries: Animated transitions andTemporal navigation for dynamic networks. 20(5):740–754. doi: 10.1109/TVCG.2013.254

[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] 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

[7] 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

[8] M. Chapman. *Constructive Evolution: Origins and Development of Piaget's Thought*. Cambridge University Press. Google-Books-ID: 7WgC-nXmdX1MC.

[9] 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

[10] 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

[11] 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

[12] N. Cohn. Visual narrative structure. 37(3):413–452. doi: 10.1111/cogs.12016

[13] C. Collins, F. B. Viegas, and M. Wattenberg. Parallel tag clouds to explore and analyze faceted text corpora. In *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on*, pp. 91–98. IEEE, 2009.

[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] 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

[16] 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

[17] N. Gershon and W. Page. What storytelling can do for information visualization. 44(8):31–37. doi: 10.1145/381641.381653

[18] 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

[19] J. Heer and G. Robertson. Animated transitions in statistical data graphics. 13(6):1240–1247. doi: 10.1109/TVCG.2007.70539

[20] J. Heer and G. Robertson. Animated transitions in statistical data graphics. 13(6):1240–1247. doi: 10.1109/TVCG.2007.70539

[21] 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

[22] 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

[23] 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

[24] 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

[25] S. Huron, R. Vuillemot, and J. D. Fekete. Visual sedimentation. 19(12):2446–2455. doi: 10.1109/TVCG.2013.227

[26] R. Kosara and J. Mackinlay. Storytelling: The next step for visualization. 46(5):44–50. doi: 10.1109/MC.2013.36

[27] 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

[28] 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

[29] T. Munzner. *Visualization Analysis and Design*. CRC Press. Google-Books-ID: dznSBQAAQBAJ.

[30] 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

[31] H.-C. Nothdurft. Salience from feature contrast: variations with texture density. 40(23):3181–3200. doi: 10.1016/S0042-6989(00)00168-1

[32] 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

[33] 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

[34] A. Satyanarayan and J. Heer. Authoring narrative visualizations with ellipsis. 33(3):361–370. doi: 10.1111/cgf.12392

[35] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer. Vega-lite: A grammar of interactive graphics. 23(1):341–350. doi: 10.1109/TVCG.2016.2599030

[36] 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

[37] J. N. Schmidt. the living handbook of narratology.

[38] E. Segel and J. Heer. Narrative visualization: Telling stories with data. 16(6):1139–1148. doi: 10.1109/TVCG.2010.179

[39] 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

[40] 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

[41] 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

[42] J. M. Wolfe. Guided search 2.0 a revised model of visual search. 1(2):202–238. doi: 10.3758/BF03200774

[43] 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

[44] 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

[45] 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