Comparison in visualization – An overview

Jonas Oesch*

Institute for Interactive Technologies, FHNW

February 2018

Abstract

In this paper, we present a high-level overview of comparison techniques in visualization. Because *comparison* is one of the principal visualization tasks, many examples from a wide range of application domains can be found in literature. We therefore concentrate on theoretical frameworks which help us evaluate the strengths and weaknesses of the different approaches in a general fashion. We find that visualizations can either be compared on the image level through *juxtaposition*, *superposition* and *static animation*. Comparison can also happen on the data level. The results of this comparison metric can then again be visualized. The latter is called *explicit encoding* and requires the comparison metric (*what* ought to be compared) to be known in advance.

In the second part, we present recurring themes from the literature and their associated research challenges. We find that comparison is especially difficult between graphs and trees because of their high information density. Showing the change history of a visualization is another theme found in different contexts. Finally, we present recent uses of implementations of reverse visualization pipelines which have been used to restyle and merge visualizations.

Part I

Theoretical framework

The design space of comparative visualization is huge and heterogeneous. Hundreds of approaches can be found in the literature [1]. In order to, gain an understanding of the general properties and challenges involved in comparative visualization, we present three high-level views on comparison and relate them to each other in this part.

*jonas.oesch@fhnw.ch

1 The comparison task

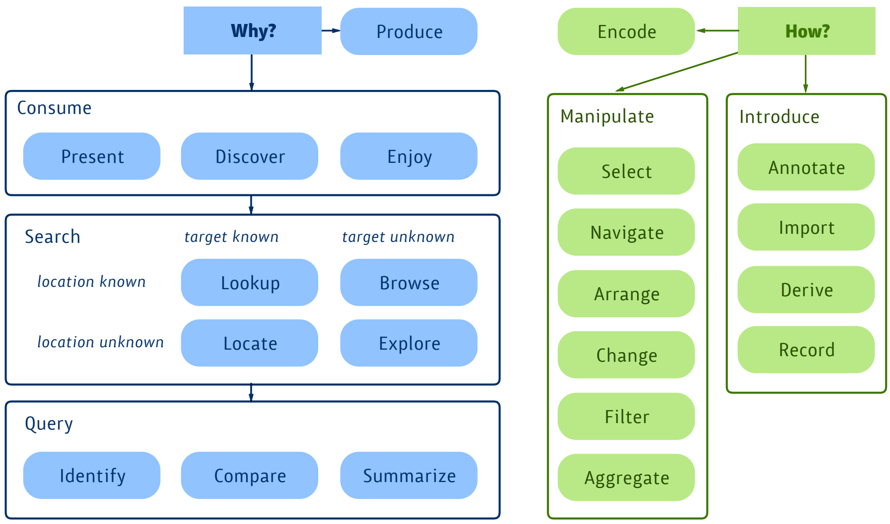


Figure 1: A topology of visualization tasks which synthesizes a large body of previous literature has been proposed by Brehmer et al. [2]

Many researchers have found it useful to analyse the field of information visualization by *task type*. A meta-analysis by Brehmer et al. [2] integrates 30 existing classifications into a unified topology of visualization tasks. (see figure 1) They classify comparison as one of eleven principal goals for the use of visualization.

Andrienko and colleagues [3] make a clear distinction between *comparison* and *relationship-seeking*. In the first case, the subsets of the data are given and the reader looks for previously unknown relationships between them; in the second case the reader looks for a subset of the data which corresponds to a given relationship (see figure 2). This presupposed relationship will be called the *comparison metric* for the rest of this paper for simplicity.

Tominski et al. [4] counter that the distinction between comparison and relationship-seeking is often blurry in practice and even absent in exploratory visualization because neither subsets nor comparison metrics are known in advance. They argue that this uncertainty is what makes it difficult to develop generalized methods to compare visualizations.

In a recent paper, Gleicher [5] describes six different comparison goals. We suspect that these would turn out to be a subset of the visualization goals described by Brehmer and colleagues [2] (for a side-by-side listing see table 1). But further research would be needed to clearly demonstrate this.

Andrienko et al.	*Gleicher*	*Brehmer et al.*
Comparison	**Identify** the relationships among items	**Identify**
Relationship-seeking	Measure / Quantify / **Summarize** those relationships	**Summarize**
Relationship-seeking	**Dissect** a relationship; that is, to examine the relationship in detail to understand it better.	Discover or Explore
Relationship-seeking	**Connect** relationships, for example, to put multiple differences together to assemble a more complete concept, or to understand the variety within a set of items.	Compare
Relationship-seeking	**Contextualize** how a similarity/difference fits into the bigger object of which it is part.	**Locate**
Relationship-seeking	**Communicate** / Illuminate a relationship (i. e. explain it to others).	**Present**

Table 1: A side-by-side listing of Gleicher's [5] goals for comparison with the goals for visualization by Brehmer et al. [2]. All but the first goal assume that one or more comparison metrics are known in advance. They are therefore classified as *relationship-seeking* according to Andrienko et al. [3]

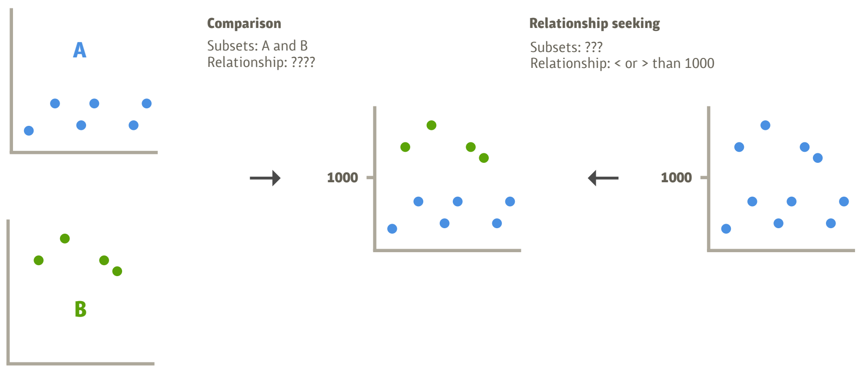


Figure 2: Andrienko et al. [3] differentiate between *comparison* where datasets are compared to find previously unknown relationships and *relationship-seeking* where the reader looks for a subset of the data that corresponds to a given relationship.

Gleicher also states that most prior work has been focused on *identifying* relationships among items. The other five goals he describes all require that one or multiple comparison metrics are known in advance. Comparison metrics therefore present a challenge for future research.

2 Visual approaches

In their seminal paper on comparison in information visualization, Gleicher et al. [1] propose a surprisingly simple taxonomy. It consists only of tree basic categories: *Juxtaposition*, *superposition* and *explicit encoding*. Their proposition is based on a sample of 173 comparative visualizations. The evaluated designs cover a wide range of dataset types and domains, from software source code [6] to phylogenetic trees [7].

The usefulness of this taxonomy according to Gleicher and colleagues is to have a general understanding of the strength and weaknesses of each comparison strategy. This in turn will help with the evaluation of future comparison solutions.

2.1 Juxtaposition

Also called *small multiples* by Tufte [8]. The visualizations are placed next to each other (see figure 3). Gleicher et al. include animation in this

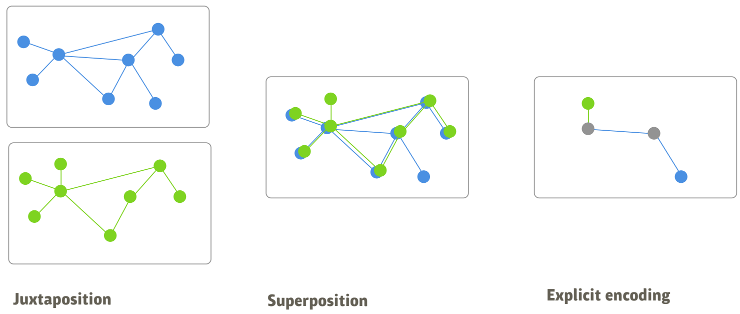


Figure 3: Gleicher et al. [1] find that there are three essentially different techniques for visualization comparison: *Juxtaposition*, *superposition* and *explicit encoding*

category with the argumentation that it is juxtaposition in time. We argue that animation should be a category because it has different strengths and weaknesses than juxtaposition. This is discussed in more detail in section 2.6.

The strength of juxtaposition is that it can be easily applied to any kind of visualization. Additionally, the approach can be scaled to an arbitrary number of objects.

The main challenge in juxtaposition according to Gleicher et al. is that it relies on pattern recognition and visual memory of the reader. Both of which are prone to constraints in the perceptual system (see chapter 9.1 in the appendix).

2.2 Superposition

Superposition or *overlay* designs place the objects to be compared in the same space (see figure 3). The strength of this approach is, that related objects are placed close to each other. By this, differences can be perceived through pattern recognition and change blindness is avoided (see chapter 9.1 in the appendix). For visualizations where the spatialization is not given (like in networks), it is important to ensure that the layout is consistent, so that corresponding objects between the compared visualizations stay close to each other.

The main challenge for superposition designs, according to Gleicher et al. [1] is *occlusion*; especially when the data density is high. Methods to address this challenge are transparency, colour weaving [9] or slight modifications in placement (see figure 3). The last is often used in cartography [10] or in

2.5D visualizations [11].

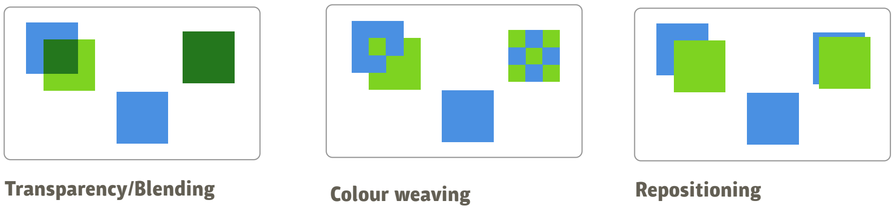


Figure 4: Different ways to deal with occlusion in superposition designs. *Blending* makes it difficult to use colour as a channel to encode other information. Others [9, 10] have proposed *colour weaving* or slight *repositioning*.

2.3 Explicit encoding

One measure for the relationship between the visualizations is calculated (a comparison metric). Instead of the original visualization, only the visually encoded relationship is shown (see figure 3). As described in Gleicher et al. [1], this offers maximum readability because the burden of comparison is removed from the reader.

A first challenge with explicit encoding is, that the relationships that should be compared need to be known in advance. This is necessary to match objects in the visualization and calculate comparison metrics. Often this is not the case [4] or it is very hard to match related objects [12]. We discuss this problem more in depth later in section 3.2.

The second challenge is that explicit encoding creates a new visualization. The original context is lost in the process. Gleicher et al. see this as the reason why explicit encoding is rarely used alone.

The authors also mention *additive encoding* as a style of explicit encoding. Additive encoding means that the original visualization is kept and the explicitly encoded relationships are added to it. Zaman [13] proposes *subtractive encoding* where the members that are related are removed from the original visualization. He argues that this is a useful way to reduce clutter and therefore occlusion, which is a problem with additive encoding.

2.4 Combined approaches

Gleicher and colleagues [1] go on to propose that the shortcomings of each individual approach can be addressed by combining it with another approach

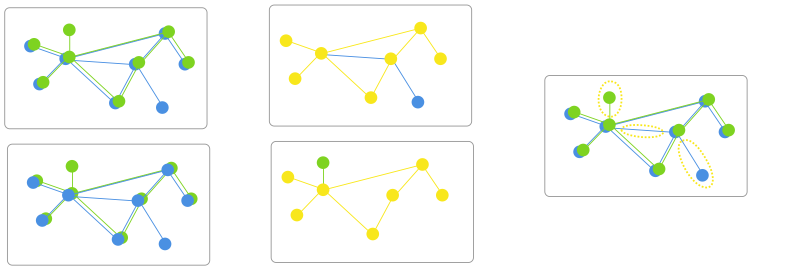


Figure 5: Gleicher et al. observe that the basic techniques (juxtaposition, superposition and explicit encoding) are often combined in order address the shortcomings of each other. Especially juxtaposition and explicit encoding are often used jointly

with differing strengths. They thus extend the taxonomy with the combined categories *juxtaposition and superposition*, *juxtaposition and explicit encoding*, *superposition and explicit encoding* (see figure 5). The hybrid most commonly found by the authors is *juxtaposition and explicit encoding* where each technique seemingly compensates very well for the weaknesses of the other. They also find examples for the rather contradictory *juxtaposition and superposition* hybrid and note that it is commonly used for comparisons of comparisons.

2.5 Validation of the taxonomy

A paper by Javed et al. [14], which was published almost at the same time as Gleicher et al. [1], explores the design space of composite visualization. Their taxonomy is very like the one proposed by Gleicher et al. It contains five categories of views: *juxtaposed*, *superimposed*, *integrated*, *overloaded* and *nested* (see table 2).

Juxtaposed and *superimposed views* as described by Javed et al. are essentially the same as the ones in Gleicher et al. *Integrated views* correspond closely to the *juxtaposition and explicit encoding* combined category. This is verified by the fact that both papers mention the *VisLink* [17] technique as an example for this respective category.

Overloading is described as a visualization superposed on another but without spatial relationship. It seems then most closely to map to *superposition and explicit encoding*. Javed and colleagues use the example of placing scatterplots on top of parallel coordinates (SPPC) [18] to illustrate this category. Yet

Gleicher et al.	*Javed et al.*	*Beck et al.*	*Kerracher et al.*
Juxtaposition	Juxtaposed views	Juxtaposed node-link diagram, layered matrices	Juxtaposition
Superposition	Superimposed views	Superimposed views	Superposition
Explicit encoding			Merged, Time-as-a-node
- additive	Nested views*	Integrated views, intra-cell-timeline*	Nested*
Juxtaposition & Superposition		Hybrid Layout	Additional spatial dimension
Juxtaposition & Explicit encoding	Integrated views	Compound graphs	
Superposition & Explicit encoding	Overloaded views*		
Juxtaposition & Superposition & Explicit Encoding			
**	**	Animation	Sequential view

Table 2: Our proposition on how taxonomies by other authors [14–16] might relate to the one proposed by Gleicher et al. [1]
* The assignment might be contestable.
** Animation is mentioned in both papers but excluded from the taxonomy for different reasons

Gleicher et al. classify SPPC simply as a *superposition* design.

Nested views use the elements of one visualization as the canvas for another. It is therefore a way of putting sub-visualizations in the context of another. This is very like *additive encoding* as described in Gleicher et al. Yet a commonly cited example (NodeTrix [19]) is there classified as *superposition*. Gleicher et al. acknowledge that additive encoding may look very much like superposition.

Further proof of the usefulness of the taxonomy in Gleicher et al. is found in two other papers: "The design space of temporal graph visualization" by Kerracher et al. [16] and "The state of the art in visualizing dynamic graphs" by Beck et al. [15]. Both apply the categories from Gleicher et al. and Javed et al. to classify techniques for visualizing changes in graphs over time.

2.6 Animation

The use of animation in visualization is prevalent enough, that there are multiple authors who study its effectiveness in comparison with other techniques. Tversky et al. [20] remark that animation seemingly is a natural choice when visualizing *processes of change* because it is *congruent* with our experience. They argue that most of the benefits found for animation are due to dynamic transitions which convey additional information to static displays. A smoothly animated acceleration contains more information than a few in-between snapshots of the same movement. This differentiation is supported by a small study by Robertson and colleagues [21] which found that juxtaposed *trendlines* led to faster recognition of trends and lower error rates than animation with dynamic transitions. Due to the use of trendlines, all information from the animation was also visible in the static view. Trendlines that were superposed performed at the same level as animation.

We differentiate therefore between *dynamic animation* where transitions are smooth and *static animation* which is more like presenting a slideshow.

The difference between dynamic and static animation is studied in Heer et al. [22] and Bach et al. [23]. Both authors found that performance is better for dynamic animation. Heer et al. also note that dynamic animation led to enthusiastic user feedback.

From the findings above, we theorize that dynamic animation is perceptually closer to superposition and non-dynamic animation closer to juxtaposition. Further research would be needed to support this claim.

3 Levels of comparison

Verma et al. [24] make another important distinction between three levels of comparison. In this they reflect and extend prior work on the issue of comparison in visualization by Pagendarm et al. [12].

3.1 Comparison on the image level

Verma and colleagues [24] state that comparison on the image level is the most common and most simple approach. Often the comparison is done by juxtaposition, in other cases also by superposition or by displaying a difference image. Often these methods need the images to be aligned.

To quantify the difference between two images, the authors mention *summary image statistics* like the root mean square or correlation analysis.

The shortcoming of image level comparison is that not the data but the impression of the visualization is being compared. The authors hold that the difference is important because design choices (resolution, interpolation, non-data elements or perceptual limits) can impact the outcome of the comparison.

3.2 Comparison on the data level

Instead of comparing visible pixels, the data can also be compared directly. For this, some idea of a relationship between the two datasets needs to be defined in advance. Comparison on the data level works therefore only for *relationship-seeking* as described by Andrienko et al. [3]. Comparison on the data level is also a prerequisite of the *explicit encoding* approach defined by Gleicher et al. [1].

Although this approach avoids the perceptual problems of image level comparison, the authors note its main disadvantage: finding suitable comparison metrics is not straightforward. This leads to very specialized algorithms with only narrow application domains in practice.

One example for this is TreeJuxtaposer by Munzner and colleagues [7]. The authors developed an algorithm for comparing the structures of phylogenetic trees. Still, later studies [25] proposed different algorithms because they wanted to compare different tree features than Munzner et al.

3.3 Comparison on the feature level

Features like area or colour are extracted from images and then compared on the data level. But this is not the same as the data level comparison described before. Image features in visualizations usually have some relationship to the underlying data but mostly this relationship is not known. Pagendarm et al. [12] mention the use of a *reverse visualization pipeline* to re-establish this relationship. Savva et al. [26] implement such a pipeline to estimate the data behind common bar and pie charts and redesign them.

4 Summary

The three presented high-level views on comparison are obviously in many ways related. Yet each of them offers its own insights for creating comparative visualizations. An important distinction in practice is, if the comparison metric is known in advance or not. If the comparison metric is not known, *juxtaposition*, *superposition* and *static animation* are the only comparison approaches that are available. Conversely: to use *explicit encoding* there are two prerequisites: the comparison metric needs to be known and the comparison needs to happen on the data level.

The main challenges for visual comparison are then:

- Perceptual constraints for juxtaposition and static animation designs
- Occlusion and layout for superposition designs
- Finding comparison metrics for explicit encoding and dynamic animation designs

We conclude the *explicit encoding* in comparison holds more promise for future research, because the challenges of this approach are more conceptual than perceptual. A further observation is, that comparison designs are often implemented in the context of change visualization which could benefit from the use of dynamic animation.

Part II

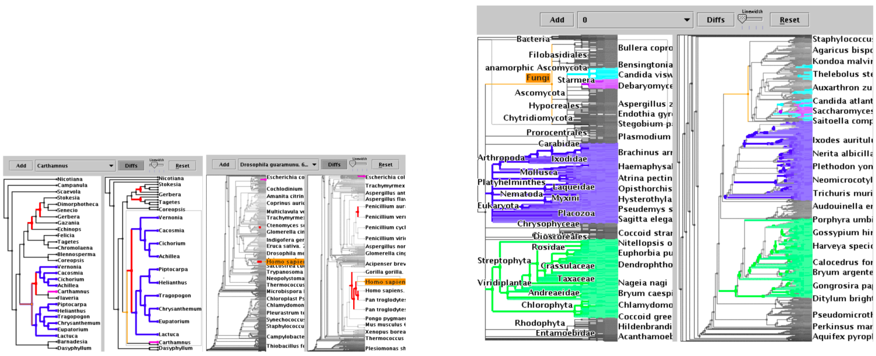
Themes

After studying the general properties of comparison in the first part, we focus on current research topics and open questions. For each general theme, we briefly present a few recent examples. Based on these, we single out future research directions as noted by the authors.

5 Graphs and Trees

Among the most widely researched data types when it comes to comparison are graphs and trees. Stability of layout (also referred to as the readers' *mental map*) is a major concern when comparing graphs and trees through juxtaposition or superposition. Another issue is the high data-density of graphs and trees. These very dense visualizations have problems with occlusion and don't leave much room for encoding additional attributes. [15, 27]

5.1 TreeJuxtaposer



(a) Visually matched subtrees

(b) Selected subtrees stay visible

Figure 6: The TreeJuxtaposer interface presented by Munzner et al. [7] allows biologists to compare large phylogenetic trees

Munzner et al. [7] present an approach for comparing large (>100'000 nodes) phylogenetic trees. They calculate similarity scores and visually match selected nodes. Nodes that are selected are kept visible while the rest of the nodes are being compressed if necessary. The authors call this "guaranteed visibility". Visually matching nodes helps readers compare trees with differing layouts. Guaranteed visibility alleviates problems with information density.

The authors note that finding good similarity metrics (or: comparison metrics) should be a topic for future research. Additionally, further exploration of the concept of guaranteed visibility is recommended.

5.2 Graph Comics

Storytelling or narrative visualization has been a topic of interest in visualization fro some years [28]. This has also resulted in attempts to apply the visual language of comics to the concerns of visualization. [29, 30]. Even though *comic strips* in the form of small multiples have been used in data visualization before [31], the more complete exploration of graphical devices of comics is a recent development [29].

Some notable examples are Temporal Summary Images by Bryan et al. [32] and a browser extension to create Data Comics by Zhao et al. [33]. Both use panels as well as integrated text as storytelling devices. Chu et al. [34] proposed an algorithm to transform temporal image sequences (e. g. videos) into comics. They make use of comic design principles to decide on the optimal layout and the optimal placement of speech bubbles.

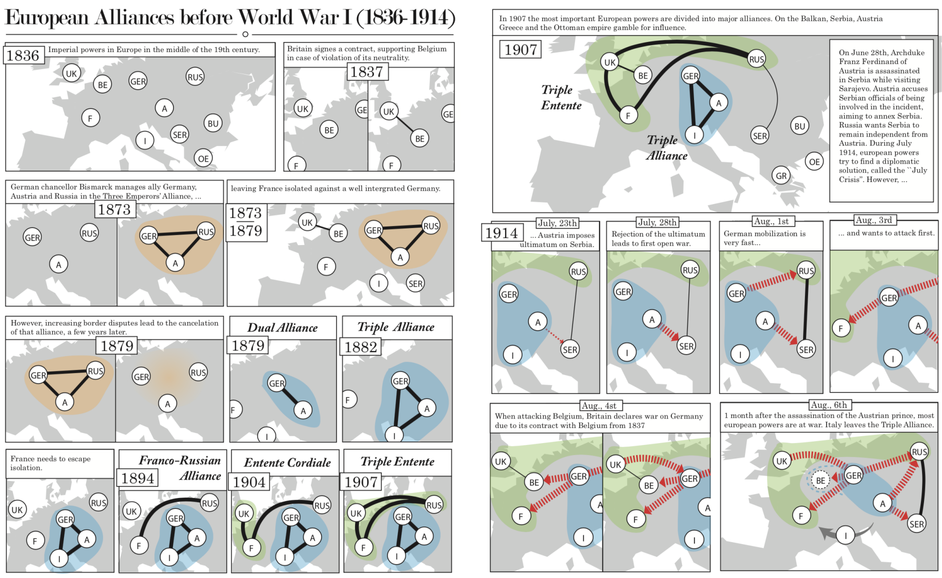


Figure 7: An example of a graph comic by Bach and colleagues [35]. The graph comic illustrates the changing alliances between European countries before World War I.

In their paper on *Graph comics*, Bach et al. [35] evaluate a wide spectrum of comic idioms, including the use of glyphs, to depict changes in networks. The authors identify five specific challenges. Four of them stem from the abstractness of graphs as opposed to the content traditionally depicted in comics (e. g. identifying a same node from one panel to the next). The fifth challenge comes from high data density. The authors address these issues with a set of design factors. For example, by designating some nodes to be *main characters* through visual emphasis; others may get summarized as an anonymous *crowd*.

The graphical vocabulary used by Bach et al. is still very limited when compared to the range proposed by McCloud in his seminal work on comics

[36]. McCloud discusses for example the effect of the style of lines on the perception of the reader. A stylistic device which has recently been applied by Wood and colleagues [37]. They used sketchy rendering to depict imprecision in visualizations. We conclude, that further exploration is needed to fully evaluate the usefulness of different comic idioms in (graph) visualization.

For the future, Bach et al. propose more sophisticated authoring solutions for graph comics. Some of their ideas are reflected in another recent work on the design of static node-link diagrams [38].

6 History

Even for datasets that have no temporal dimension, visualizing change may be important. History accrues also when a reader explores an interactive visualization or when an author edits a visualization.

6.1 Edit history

Heer et al. [31] research the use of graphical histories in exploratory data visualization. According to them creation and exploration go hand in hand most of the time in an iterative process. At each step, new questions arise and old questions are answered. A graphical history mechanism permits the author to go back to previous stages of their exploration, understand how she came to a conclusion and communicate her thinking process to others.

The authors use small screenshots to indicate steps in the edit history. They are juxtaposed along the lower edge of the main visualization window. A prior state of the visualization can be restored by clicking on one of the small screenshots.

Heer and colleagues discuss the difficulty in finding the right amount of detail for the history steps. They use a heuristic which groups actions that would be perceived as related, like a sequence of keystrokes.

Su et al. [40] use explicit encoding in the form of glyphs to show the edit history for a vector graphic.

In future work, they also mention the need for a more general approach to defining history steps and finding metrics to define the salience of a history step. Questions which are further explored by Bryan et al. [32].

As another research direction, Heer et al. also see a need for facilitating the creation of presentations from analysis histories.

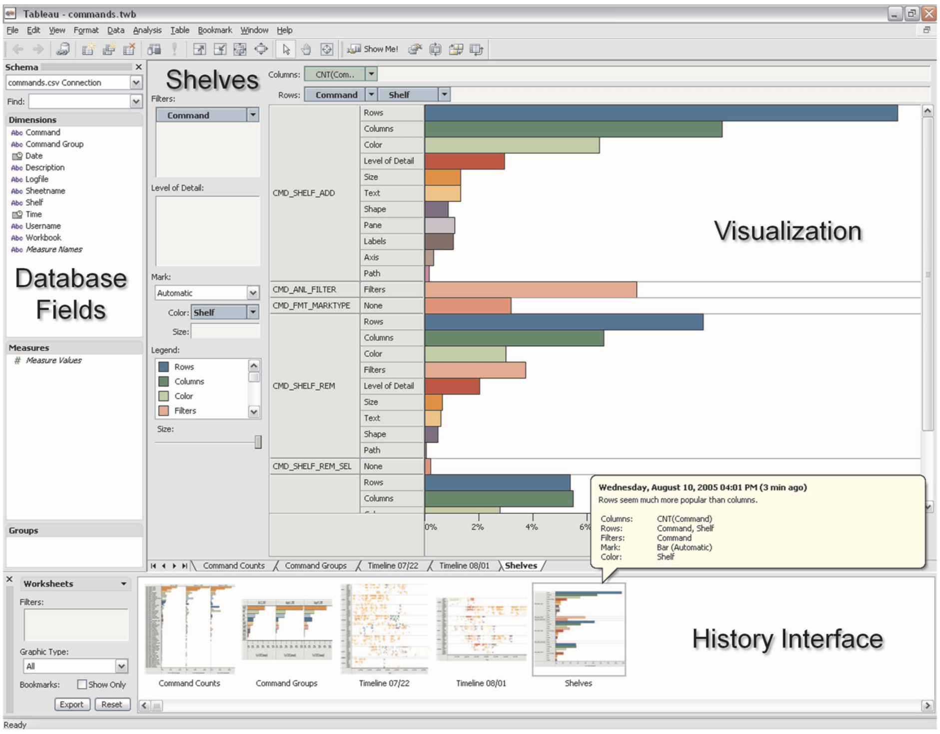


Figure 8: Heer et al. [31] presented a history interface for the visualization tool *tableau* [39]. It shows previous steps in the visualization process as small multiples.

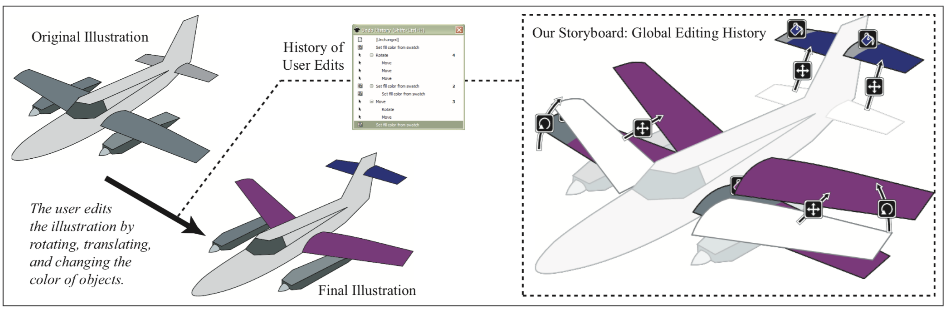


Figure 9: Su et al. [40] added a visual history mechanism to a vector graphics editor. They use glyphs to show how an object has changed from a previous step.

6.2 Interaction history

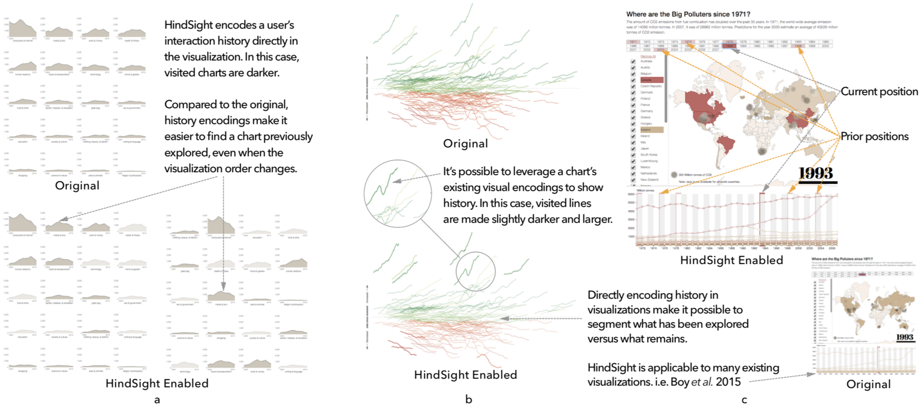


Figure 10: Hindsight is a mechanism to explicitly encode interaction history in a visualization. Elements that the user has interacted with change colour or are otherwise marked.

Related to the topic of edit history is interaction history. In their paper, Feng et al. [41] evaluate the effects of explicitly encoding 'visited' and 'unvisited' objects in visualizations. Their design is based on the idea of *wear* which has successfully been applied in interaction design according to the authors.

In agreement with Heer and colleagues [42] they also mention the difficulty in finding the right 'unit' and interaction duration to determine a history element.

The authors discuss different means of explicitly encoding the *visited* state: *Augmentation* where unused visual channels are used to encode the state, *addition* which exploits unused space in the chart to indicate the state and the *adaptation* of the visualization which becomes necessary when no visual channels are available.

In their evaluation, the authors find that encoding the interaction history in a visualization leads readers to explore it more widely. Recording the exploration process has also been explored by Robinson et al. [43]. They used a superposition design (traces on top of maps) to help readers retrace their exploration process.

7 Interaction

Interaction might be one way to offset some of the weaknesses of juxtaposition, superposition and static animation designs. Intriguingly this leads to a generalized comparison solution, one which is not tailored to a specific kind of visualization or domain.

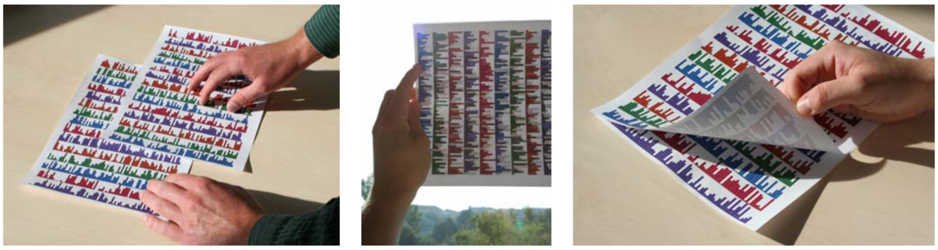


Figure 11: Examples for the ways researchers compare visualizations on paper. Tominski et al. [4] adapted these techniques for their interactive comparison tool

One such proposition can be found in a paper by Tominski et al. [4]. The authors take their inspiration from the way readers compare visualizations on paper. They identify three ways: *side-by-side*, *shine-through* and *folding* (see figure 11). These can roughly be mapped to the *juxtaposition*, *superposition* and *static animation* categories proposed earlier.

Comparison happens in a zoomable visualization space where visualizations can be placed freely. Views are based on selections either in the image space or in the data space. Snapping to different image properties (grid, axes, objects, image features) helps with alignment.

As a result of their user testing, Tominiski et al. note that the whole range proposed comparison techniques is used by the readers. Each for a different purpose. Smaller views with trends were compared using juxtaposition. Superposition served to identify regions with changes while folding was used for more regional comparison.

In future work, they mention that users might need more assistance in finding similar regions than currently provided by their tool. They propose viscosity when moving a visualization on top of another to *snap* a view region to another that is similar.

8 Reverse visualization pipeline

A last big theme we have observed is the *reverse visualization pipeline* mentioned in the chapter on *feature extraction*. Relating visualization features to the underlying data may offer new possibilities for generalized visualization tools that use *explicit encoding*.

8.1 Visualization by Demonstration

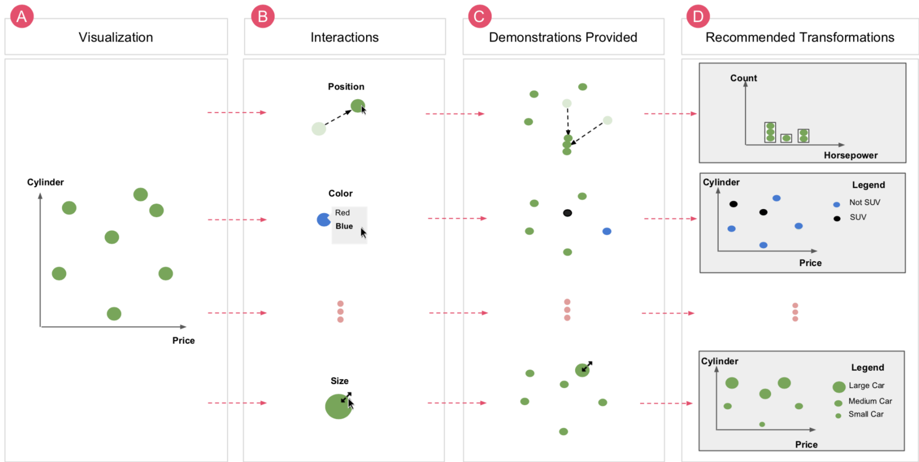


Figure 12: Example of a visualization by demonstration workflow [44]. Readers can change the graphical attributes of the elements of a visualization. Based on their changes, an algorithm tries to deduce new visual mappings

Saket et al. [44] let the viewer interact with positions and graphical attributes in a scatterplot. From the difference, they try to deduce the viewers' intent and propose new axes or a different visualization layout (see figure 12. When translated to our terminology, we could say that the viewer visually defines a new comparison metric.

As a challenge the authors explain that it is often hard to deduce a clear recommendation when there is too little user input. As the viewer goes on providing more and more information on her intent, ambiguity can get resolved.

The authors have implemented their approach only for scatterplots and bar charts. A future research challenge, according to them, would be to extend their principle of 'visualization by demonstration' to other visualization types.

8.2 Extracting features and restyling

Savva et al. [26] classify commonly found visualizations like bar and pie charts found on the internet. They then extract image features and labels with traditional image processing techniques to deduce the underlying data. From the deduced data, they create alternative visualizations (see figure 13).

Their technique is still extremely limited. For example, it currently only

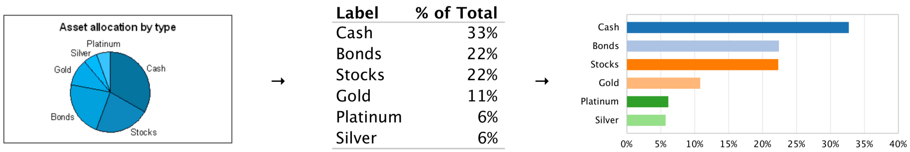


Figure 13: Through machine learning and image processing techniques, Savva et al. [26] extract data from common visualizations they find on the internet. From the extracted data, they create alternative visualizations.

works for bar and pie charts. It misses small data elements and text labels had to be tagged manually by the authors. Future work would need to further evolve their approach to make it more stable and more generally applicable.

8.3 Iterative visualization editing

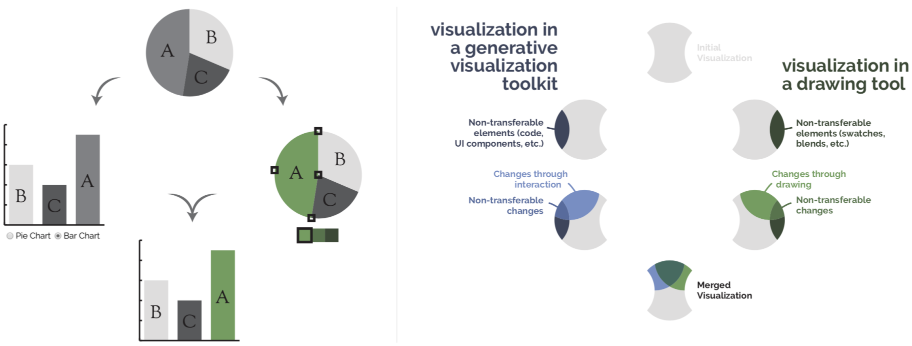


Figure 14: Bigelow et al. [45] demonstrate a visualization design process that involves multiple tools. They address the challenge of how to merge changes from different visualization tools

Bigelow et al. [45] use comparison on the feature and on the data level to enable iterative editing of visualizations. Their observation is, that designers use multiple tools to create visualizations, but they often lose much of their work, when moving between tools. The authors propose a bridge model between generative visualization tools like D3 [46] and drawing tools like Adobe Illustrator [47] (see figure 14). They use SVG as a *shared data representation*. SVG elements describe visual elements and are bound to data points via an *ID*.

Their approach is interesting because it seemingly works for a wide range visualizations. The authors don't mention the limited applicability found by

Saket et al. [44] and Savva et al. [26]

Bigelow and colleagues note that for successfully calculating the difference between two visualizations, it is important to have the right *granularity* of objects and attributes. Also, the objects must clearly be attributable to a data point. The question of the choice of granularity is reminiscent of the challenges seen in section 6.

The authors also make an important distinction between *data-bound* and *non-data-bound elements* which exist simultaneously in the same visualization. Often non-data-bound elements like labels are nonetheless connected to data-bound elements like the bars in a barchart. How to relate these to each other is an open question for the authors (like Savva et al. [26]). They mention a shared data representation as another topic of future research.

In future work, the authors propose the development of new bridges between generative tools and drawing tools. Notably the creation of bridge for pixel based visualizations. They also see a need for a standard format for visualizations which brings together data and graphical objects.

9 Summary and research directions

In the first part, we have established the theoretical framework to understand comparison and the challenges it poses to visualization. We concluded that there are four basic approaches to comparison: juxtaposition, superposition, animation and explicit encoding. We also speculated that the design space of explicit encoding might be equal to the design space of visualization in general. We further differentiated between image level, data level and feature level comparison and showed that data level comparison is a prerequisite of explicit encoding.

In the second part, we have presented themes that we have encountered related to visual comparison. From the difficulties that the authors have faced, we deduce the following directions for future research:

9.1 Research directions

Collect many examples and generalize

- Further explore the comparison goals which involve *relationship-seeking* and explicit encoding. The goal would be to build a subclassification of **explicit encoding approaches**. This would help to understand if the design space of explicit encoding is equal to that of visualization in general

Work on perception

- Further explore the literature and study the benefits and challenges of **animation for comparison**.

- Find ways to find good **comparison metrics** without prior knowledge of the type of data or type of visualization. Relating for example the deltas between two visualizations on the feature and on the data level to create *difference visualizations*
- Extend the concept of **guaranteed visibility** from trees to graphs
- Find and evaluate applications of the **graphical language of comics** to visual comparison
- Building tools to help authors **narrate change** in visualizations
- Finding metrics to **identify salient history steps** in a changing visualization
- Using **feature extraction** to find similar regions between two visualizations and with this assisting comparison by juxtaposition and superposition.
- Research properties of an image format for visualizations that relates image features to underlying data.

Appendix

Perceptual constraints

In visual comparison, human perceptual features and limitations play an important role. The literature distinguishes between *implicit* and *explicit* detection of changes/differences. Some visual features (like hue or orientation) can supposedly be perceived implicitly at the preattentive level [48]. Implicit perception is very rapid and parallelized. The evidence for implicit detection of changes between two views is very weak [49]. Yet there are hints that dynamic changes (<300ms) can be perceived implicitly and draw the attention of the viewer. [50]

Explicit perception is a slower process which is also called *visual search*. Explicit perception seems to be dependent on attention. Attention at the same time is limited to around four spatial regions (1–2 in worst case; up to 9 in best case scenarios) [51]. Comparison happens against a limited set of objects in *visual short term memory*. The number of objects that can be held in this memory is dependent on task type [52] and the complexity of the objects [53]. It can be as high as five and as low as one object.

An often-cited effect of these attention and memory limitations is called *change blindness* [54]. Viewers can miss even large changes in a view if their attention is not focused on the region and the features that change. The classical example is a black gorilla walking through the scene unnoticed when a viewer focuses on basketball players in white shirts.

Prior knowledge by the reader seems to *reduce the complexity* of the objects. Relatively complex objects (like Chinese signs) can thus be perceived like a single object (also called a *chunk*) [50]. What combinations of features can be chunked and how efficiently is still an open question in research [52].

References

[1] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. D. Hansen, and J. C. Roberts, “Visual comparison for information visualization.,” English, *Information visualization*, vol. 10, no. 4, pp. 289–309, 2011.

[2] M. Brehmer and T. Munzner, “A Multi-Level Typology of Abstract Visualization Tasks.,” *Ieee trans. vis. comput. graph.*, vol. 19, no. 12, pp. 2376–2385, 2013.

[3] N. V. Andrienko and G. L. Andrienko, “Exploratory analysis of spatial and temporal data - a systematic approach.,” 2006.

[4] C. Tominski, C. Forsell, and J. Johansson, “Interaction Support for Visual Comparison Inspired by Natural Behavior.,” *Ieee transactions on visualization and computer graphics*, vol. 18, no. 12, pp. 2719–2728, 2012.

[5] M. Gleicher, “Considerations for Visualizing Comparison,” *Ieee transactions on visualization and computer graphics*, vol. 24, no. 1, pp. 413–423,

[6] D. J. Barnes, M. T. Russell, M. W. E. Conference, and 1988, *Developing and adapting UNIX tools for workstations.*

[7] T. Munzner, F. Guimbretiere, S. Tasiran, L. Zhang, and Y. H. Zhou, “TreeJuxtaposer: Scalable tree comparison using Focus+Context with guaranteed visibility,” English, *Acm transactions on graphics*, vol. 22, no. 3, pp. 453–462, Jul. 2003.

[8] E. R. Tufte, *The visual display of quantitative information.* Cheshire, CT: Graphics Press, 2001.

[9] H. Hagh-Shenas, S. Kim, V. Interrante, and C. G. Healey, “Weaving Versus Blending - a quantitative assessment of the information carrying capacities of two alternative methods for conveying multivariate data with color.,” *Ieee trans. vis. comput. graph.*, vol. 13, no. 6, pp. 1270–1277, 2007.

[10] J. Wood, J. Dykes, A. Slingsby, and K. Clarke, “Interactive Visual Exploration of a Large Spatio-temporal Dataset: Reflections on a Geovisualization Mashup.,” *Ieee transactions on visualization and computer graphics*, vol. 13, no. 6, pp. 1176–1183, Sep. 2017.

[11] U. Brandes, T. Dwyer, and F. Schreiber, “Visualizing Related Metabolic Pathways in Two and a Half Dimensions.,” *Graph drawing*, 2003.

[12] H. G. Pagendarm and F. H. Post, *Comparative visualization: Approaches and examples*, 1995.
[13] L. Zaman, "User Interfaces and Difference Visualizations for Alternatives," 2015.
[14] W. Javed and N. Elmqvist, "Exploring the design space of composite visualization.," *Pacificvis*, pp. 1–8, 2012.
[15] F. Beck, M. Burch, S. Diehl, D. W. E. STAR, and 2014, *The state of the art in visualizing dynamic graphs*.
[16] N. Kerracher, J. Kennedy, K. C. EuroVis, and 2014, "The design space of temporal graph visualisation," *Academia.edu*,
[17] C. Collins and S. Carpendale, "VisLink: Revealing Relationships Amongst Visualizations," *Ieee transactions on visualization and computer graphics*, vol. 13, no. 6, pp. 1192–1199,
[18] X. Yuan, P. Guo, H. Xiao, H. Zhou, and H. Qu, "Scattering Points in Parallel Coordinates," *Ieee transactions on visualization and computer graphics*, vol. 15, no. 6, Nov. 2009.
[19] N. Henry, J.-D. Fekete, and M. J. McGuffin, "NodeTrix: A hybrid visualization of social networks," *Ieee transactions on visualization and computer graphics*, vol. 13, no. 6, pp. 1302–1309, Nov. 2007.
[20] B. Tversky, J. B. Morrison, M. B. I. j. o. human, and 2002, "Animation: can it facilitate?" *Elsevier*,
[21] G. G. Robertson, R. Fernandez, D. Fisher, B. Lee, and J. T. Stasko, "Effectiveness of Animation in Trend Visualization.," *Ieee trans. vis. comput. graph.*, vol. 14, no. 6, pp. 1325–1332, 2008.
[22] J. Heer and G. G. Robertson, "Animated transitions in statistical data graphics," *Ieee transactions on visualization and computer graphics*, vol. 13, no. 6, pp. 1240–1247, 2007.
[23] B. Bach, E. Pietriga, and J. F. on, "GraphDiaries: animated transitions andtemporal navigation for dynamic networks," *Ieeexplore.ieee.org*,
[24] V. Verma and A. Pang, "Comparative flow visualization," English, *Ieee transactions on visualization and computer graphics*, vol. 10, no. 6, pp. 609–624, 2004.
[25] J. Gomez, A. Buck-Coleman, and C. Plaisant, "Interactive visualizations for comparing two trees with structure and node value changes," 2012.
[26] M. Savva, N. Kong, A. Chhajta, F.-F. Li, M. Agrawala, and J. Heer, "ReVision - automated classification, analysis and redesign of chart images.," *Uist*, p. 393, 2011.
[27] *Visual Graph Comparison*, Jun. 2009.
[28] N. Gershon and W. Page, "What storytelling can do for information visualization," *Communications of the acm*, vol. 44, no. 8, pp. 31–37, 2001.

[29] B. Bach, N. H. Riche, S. Carpendale, and H. Pfister, "The Emerging Genre of Data Comics.," *Ieee computer graphics and applications*, vol. 37, no. 3, pp. 6–13, 2017.

[30] B. Victor, *Scientific Communication as Sequential Art*, Oct. 2017.

[31] J. Heer, J. D. Mackinlay, C. Stolte, and M. Agrawala, "Graphical Histories for Visualization: Supporting Analysis, Communication, and Evaluation," *Ieee transactions on visualization and computer graphics*, vol. 14, no. 6, pp. 1189–1196, 2008.

[32] C. Bryan, K.-L. Ma, and J. Woodring, "Temporal Summary Images - An Approach to Narrative Visualization via Interactive Annotation Generation and Placement," *Ieee trans. vis. comput. graph.*, vol. 23, no. 1, pp. 511–520, 2017.

[33] Z. Zhao, R. Marr, and N. Elmqvist, "Data Comics: Sequential Art for Data-Driven Storytelling," pp. 1–12, Oct. 2015.

[34] W.-T. Chu, C.-H. Yu, and H.-H. Wang, "Optimized Comics-Based Storytelling for Temporal Image Sequences.," *Ieee trans. multimedia*, vol. 17, no. 2, pp. 201–215, 2015.

[35] B. Bach, N. Kerracher, K. W. Hall, S. Carpendale, J. Kennedy, and N. Henry Riche, "Telling Stories about Dynamic Networks with Graph Comics," in *The 2016 chi conference*, New York, New York, USA: ACM Press, 2016, pp. 3670–3682, ISBN: 9781450333627.

[36] S. McCloud, *Understanding comics*, 1993.

[37] J. Wood, P. Isenberg, T. I. 0001, J. Dykes, N. Boukhelifa, and A. Slingsby, "Sketchy Rendering for Information Visualization.," *Ieee trans. vis. comput. graph.*, vol. 18, no. 12, pp. 2749–2758, 2012.

[38] A. S. Spritzer, J. Boy, P. Dragicevic, J.-D. Fekete, and C. M. Dal Sasso Freitas, "Towards a smooth design process for static communicative node-link diagrams," English, *Computer graphics forum*, vol. 34, no. 3, pp. 461–470, Jul. 2015.

[39] "Tableau Software,"

[40] S. L. Su, S. Paris, F. Aliaga, C. Scull, S. J. M. Retrieved, and 2009, "Interactive visual histories for vector graphics," *Researchgate.net*,

[41] M. Feng, C. Deng, E. M. Peck, and L. Harrison, "HindSight - Encouraging Exploration through Direct Encoding of Personal Interaction History.," *Ieee trans. vis. comput. graph.*, vol. 23, no. 1, pp. 351–360, 2017.

[42] J. Heer, F. B. Viégas, and M. Wattenberg, "Voyagers and voyeurs - supporting asynchronous collaborative information visualization.," *Chi*, p. 1029, 2007.

[43] A. C. Robinson and C. Weaver, "Re-visualization: Interactive visualization of the process of visual analysis," *Personal.psu.edu*,

[44] B. Saket, H. Kim, E. T. Brown, and A. Endert, "Visualization by Demonstration - An Interaction Paradigm for Visual Data Exploration.," *Ieee trans. vis. comput. graph.*, vol. 23, no. 1, pp. 331–340, 2017.

[45] A. Bigelow, S. M. Drucker, D. Fisher, and M. D. Meyer, "Iterating between Tools to Create and Edit Visualizations.," *Ieee trans. vis. comput. graph.*, vol. 23, no. 1, pp. 481–490, 2017.
[46] M. Bostock, V. Ogievetsky, and J. Heer, "D^3 Data-Driven Documents," *Ieee transactions on visualization and computer graphics*, vol. 17, no. 12, pp. 2301–2309, 2011.
[47] "Adobe Illustrator," *Adobe.com*,
[48] C. G. Healey, K. S. Booth, and J. T. Enns, "High-Speed Visual Estimation Using Preattentive Processing.," *Acm trans. comput.-hum. interact.*, vol. 3, no. 2, pp. 107–135, 1996.
[49] S. R. Mitroff, D. J. Simons, and S. L. Franconeri, "The siren song of implicit change detection.," English, *Journal of experimental psychology-human perception and performance*, vol. 28, no. 4, pp. 798–815, 2002.
[50] R. A. Rensink, "Change detection," *Annual review of psychology*, vol. 53, pp. 245–277, 2002.
[51] S. L. Franconeri, *The Nature and Status of Visual Resources*, Oxford University Press, 2013.
[52] J.-S. Hyun, G. F. Woodman, E. K. Vogel, A. Hollingworth, and S. J. Luck, "The comparison of visual working memory representations with perceptual inputs.," English, *Journal of experimental psychology-human perception and performance*, vol. 35, no. 4, pp. 1140–1160, 2009.
[53] G. A. Alvarez and P. Cavanagh, "The Capacity of Visual Short-Term Memory is Set Both by Visual Information Load and by Number of Objects," English, *Psychological science*, vol. 15, no. 2, pp. 106–111, 2004.
[54] D. J. Simons, "Current Approaches to Change Blindness," English, *Visual cognition*, vol. 7, no. 1-3, pp. 1–15, Jan. 2000.