Int. J. Human-Computer Studies 82 (2015) 11–20

Contents lists available at ScienceDirect

Int. J. Human-Computer Studies

journal homepage: www.elsevier.com/locate/ijhcs

Sketcholution: Interaction histories for sketching ☆

Zhenpeng Zhao [a], William Benjamin [b], Niklas Elmqvist [a,c,*], Karthik Ramani [b]

[a] Department of Computer Science, University of Maryland, 2117H Hornbake Building, South Wing, College Park, MD, USA
[b] School of Mechanical Engineering, Purdue University, West Lafayette, IN, USA
[c] College of Information Studies, University of Maryland, College Park, MD, USA

ARTICLE INFO

Article history:
Received 12 March 2014
Received in revised form
17 February 2015
Accepted 15 April 2015
Communicated by P. Mulholland
Available online 30 April 2015

Keywords:
Sketching
Interaction history
Animation
Summaries
User study

ABSTRACT

We present Sketcholution, a method for automatically creating visual histories of hand-drawn sketches. Such visual histories are useful for a designer to reflect on a sketch, communicate ideas to others, and fork from or revert to an earlier point in the creative process. Our approach uses a bottom-up agglomerative clustering mechanism that groups adjacent frames based on their perceptual similarity while maintaining the causality of how a sketch was constructed. The resulting aggregation dendrogram can be cut at any level depending on available display space, and can be used to create a visual history consisting of either a comic strip of highlights or a single annotated summary frame. We conducted a user study comparing the speed and accuracy of participants recovering causality in a sketch history using comic strips, summary frames, and simple animations. Although animations with interaction may seem better than static graphics, our results show that both comic strip and summary frame significantly outperform animation.

1. Introduction

Sketching is commonly defined as rapidly created freehand drawing that is not intended to yield finished work, but rather to visually explore ideas (Ullman et al., 1990), and is a common tool for early design and creativity, regardless of discipline (McKim, 1972). While pen and paper remain the most common medium for such activities, digital media for sketching has several compelling benefits beyond paper. One such is the ability to capture not just the final state of a sketch, but also every intermediate state along the way. Based on this idea, we propose *Sketcholution*, an automatic visual interaction history of how a sketch has evolved over time (Fig. 1). Whereas a paper-based visual history would require the designer to take regular photocopies or digital pictures of the sketch being worked on, Sketcholution runs unobtrusively in the background of the digital sketch tool, capturing every single stroke made by the designer. This interaction history can then be played back, stroke by stroke, to show how the sketch was created and evolved over time. This would allow the designer to, for example, recall progress made during an earlier sketch session, communicate a particular idea to a collaborator or a stakeholder, or access an earlier version of a sketch to either revert to or fork from that version (similar to source control systems).

Animation with interactive control is a seemingly obvious choice over static representation given that animations have been shown to improve understanding of spatiotemporal information (Rieber, 1990; Zhu et al., 2011). However, merely playing back an animation of the interaction history for a sketch is not necessarily the optimal presentation method. Complex animations can be difficult to perceive accurately (Dragicevic et al., 2011) and are also potentially time-consuming to view in their entirety. For that reason, we propose two new *static* techniques for automatically summarizing sketch history captured during one or several design sessions: a *comic strip* of representative frames (or highlights) during the history, and a *summary frame* that annotates the changes made to the sketch in a single image. Both approaches rely on a bottom-up agglomerative clustering algorithm (Jain et al., 1999) that combines adjacent frames (each representing a stroke) into *frame aggregates* while retaining the causal sequence of the interaction history. The decision of which frames to combine depends on the distance between consecutive frames as computed

☆This paper has been recommended for acceptance by P. Mulholland.

* Correspondence to: College of Information Studies, University of Maryland, College Park, 2117H Hornbake Building, South Wing, College Park, MD 20742, USA. Tel.: +1 301 405 7414; fax: +1 301 314 9245.
E-mail addresses: zhaoz@umd.edu (Z. Zhao), william@purdue.edu (W. Benjamin), elm@umd.edu (N. Elmqvist), ramani@purdue.edu (K. Ramani).

http://dx.doi.org/10.1016/j.ijhcs.2015.04.003
1071-5819/

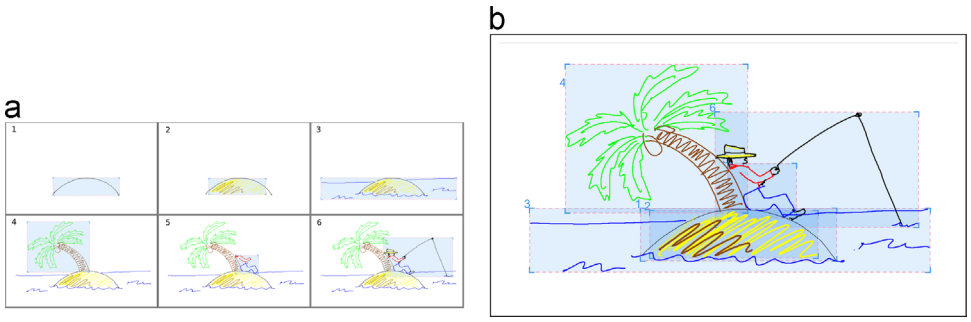

Fig. 1. Our two Sketcholution visual history mechanisms illustrating the evolution of a sketch. (a) Comic strip (6 frames), (b) summary frame (events 1–6).

by a *frame distance function*. The resulting aggregation dendrogram can be cut at any level to yield a desired number of frames (for comic strips), or a particular distance threshold between events (for summary frames).

To determine which history presentation—animation, comic strip, or summary frame—is most efficient, we conducted a user study comparing the completion time of participants recovering the causal sequence of visual components in a sketch. Results show that both comic strip and summary are significantly faster than animation. These results also provide compelling evidence to the controversy surrounding animation for comprehension (Tversky et al., 2002).

2. Related work

Our sketch captury and summary mechanism lies at the intersection of sketching, early design, and interaction histories. Below we review relevant work in these research areas.

2.1. Sketching and cognition

A *sketch* is a rapidly created freehand drawing that is not intended to create finished work, but rather to visually explore ideas (Ullman et al., 1990; McKim, 1972). Sketches can be used in the design of electrical, mechanical, scientific, mathematical, and software artifacts (Sutherland, 1964). For this reason, sketches are often used for idea generation and recording in early design—see below.

The order in which we sketch and draw reflects how we think. Taylor and Tversky (1992) studied how people create regional maps and observed that the order in which people draw reflects their mental organization of the space. Regions which had features at multiple scales were depicted starting with larger features first followed by smaller ones. Sketches themselves have a definite structure similar to language and consists of basic elements such as lines and blobs (Tversky et al., 2000). Tversky (1999) showed that the order of sketching elements reveals the designer's underlying conceptual organization.

2.2. Sketching for early design

Design is a gradual, iterative process, often beginning from ill-defined or difficult problems that are decomposed, explored, and integrated in turn to yield many possible solutions (Taborda et al., 2012). Sketches play an important role in externalizing ideas during early design (McKim, 1972), providing a "visible graphic memory" (p. 127) that facilitates creativity by providing an easily accessible database of generated ideas and by stimulating building on earlier ideas. Studies show that pictorial representations in general, and sketching in particular, are more effective than any other representations during early phases of ideation and creativity (Mckoy et al., 2001). Furthermore, pen and paper remain the most common tools for sketching in early design (Ullman et al., 1990). Greenberg et al. (2011) present a collection of methods to illustrate how to design with sketching.

Nevertheless, with the recent proliferation of pen-input devices, many efforts have been made to develop sketch-based interfaces for early design in a wide variety of domains such as architecture (Dorsey et al., 2007), automotive design (Kara and Shimada, 2008), and software design (Chen et al., 2003). These approaches all aim at replicating the good properties of paper—such as minimal learning curve, natural and precise interaction, and physical affordances—while retaining the unique benefits afforded by digital media, such as replication and composition of sketches.

2.3. Interaction logs and graphical histories

Interaction logs and histories are common in human–computer interaction due to their relation to undo and redo operations, and modern user applications typically support multi-level versions of these, sometimes of a selective nature (Berlage, 1994). Similarly, navigation histories are central to web browsers, allowing users to easily go back and forward while browsing the Web. Heer et al. (2008) carefully review the design space of interaction histories; we refer to their survey for further details on interaction capture and recall. Compared to Heer et al., our work targets a different media type and uses an aggregation algorithm to chunk the history.

Interaction data can be used to even greater effect. *Graphical histories* do not just maintain a list or stack of interactions, but also show them using a graphical summary (Kurlander and Feiner, 1988). This is most commonly done using a thumbnail image of previous state (Heer et al., 2008; Ma, 1999), and allows for capturing interactions over time (Nancel and Cockburn, 2014). In recent work, Heer et al. (2008) propose a comic strip-style graphical history using thumbnails of previous visualizations. Further, recent work has shown that augmenting sketch histories with contextual information such as pictures audio and videos improves the effectiveness of sketching for communication in early design (Li et al., 2012). Proper segmentation with users' guidance makes the graphical history easier to understand (Noris et al., 2012).

Beyond interaction data, histories have also been used for summarizing other media types. For example, histories for binary image files can be modeled using a directed acyclic graph to store temporal and semantic relationships (Chen et al., 2011). For video histories, Barnes et al. (2010) proposed continuous zooming to support navigation in time. Building on this, Ajmal et al. (2012) give a comprehensive introduction to video summarization techniques, of which cluster-based and color-based methods are partially similar to our proposed aggregation approach. Eccles et al. (2008) integrated geotemporal information into storytelling and presented stories with data such as behaviors and events. However, compared to all of the above techniques, our aggregation

approach is different, our data type is sketch strokes, and we also focus on presentation techniques for the cluster data, which we evaluate with a user study.

3. Sketcholution

Sketcholution is an automatic technique for capturing and visualizing the history of a sketch for the purpose of reflection, collaboration, and revision. Based on our literature review, we formulate the following design goals for a sketch-based history mechanism:

- *Stroke-level events:* A truly useful sketch history requires fine-grained history capture down to the level of individual strokes and formatting operations. This would enable the designer to reflect on each individual change in a sketch.
- *Chronology-preserving:* The causal order of how a sketch was constructed is a vital part of providing an accurate history. The chronology may also give insight into the creative process behind the sketch beyond the final result.
- *Efficient screen usage:* The sketch history should be space-efficient and adaptable to any screen area; in fact, it may become integrated into the sketch editor itself.
- *Efficient time usage:* An efficient history mechanism for sketches should support quick references so that it can become part of the sketch flow.

Many versioning systems allow users to revert and fork the version history from earlier revisions. However, such advanced version control operations are beyond the scope of this work; here we only concern ourselves with capturing and summarizing sketching.

3.1. Data model

We define a *digital sketch* as a *canvas* and an ordered list of *sketch operations* that, when executed in sequence by an appropriate 2D rendering engine, yields a visual representation of the sketch on the canvas. Many types of sketch operations are potentially relevant; in this work, we include draw strokes, erase strokes, and formatting (color, transparency, stroke thickness). Both draw and erase strokes are modeled as pairs of 2D points (lines), where the former add digital ink to the canvas, and the latter remove it.

Using this definition of a digital sketch, it is clear that the sketch itself is also a stroke-by-stroke history of how it was constructed, from the first drawing operation to the last. We therefore call the list of sketch operations in the sketch a *sketch history*. This also means that rendering a sketch consisting of N operations from the beginning yields N consecutive *frames*—as in an animation—where the last frame F_N is the current state of the sketch: $F_1, F_2, \ldots, F_N$. A frame F_t can be seen as the set of sketch operations added to the sketch up to a time t. To complete this interpretation, we also include time stamps t with each individual operation in the list, allowing us to exactly animate both the order and timing of the sketch.

3.2. Sketch capture

Whereas most versioning systems (such as SVN, git, and Subversion) as well as cloud storage systems (Dropbox and Google Drive) operate on a file level, Sketcholution must integrate with the sketch editor itself in order to collect stroke-level data. We focus on two main sketch operations—drawing and erasing—as well as changing the color and thickness of the drawing stroke, and the size of the eraser (however, other operations are also possible to capture and recall). All sketch input results in sequences of stroke segments (two points forming a line) being added to the sketch history as draw or erase strokes (depending on the chosen tool).

Storing a Sketcholution sketch amounts to serializing the canvas (dimensions, background, title, etc.) as well as the entire sketch history. For long sketch histories, it may be impractical to execute or store the entire sequence of operations from beginning to end due to high rendering time and memory consumption, respectively. This is particularly true when many draw strokes have since been erased. For this purpose, we use the SVG format to store a cached version of the current state of a sketch's visual representation.

3.3. Sketch summary

The design space of interaction histories is large (Heer et al., 2008), but our design goals limit the options to consider. Since our intention with this work is not to support forking past states, there is no branching model and thus no need for a hierarchical visual representation. Rather, the sketch history is a linear list, and the challenge instead becomes how to reduce the number of frames that is shown in the summary, from showing all N frames to just showing one.

In light of these constraints and goals, we formulate the following main types of sketch summary mechanisms (discussed below):

- *Animation:* Play back the entire sequence as a smooth animation. This option corresponds to one end of the above design spectrum: including all N frames.
- *Comic strip:* Render a subset of n representative frames as a static comic strip. The number would be selected as $n \in (1, N)$, so this constitutes a design compromise.
- *Summary frame:* Create a single frame that captures the entire sequence and enumerates the order of events. This design is at the other end of the spectrum: $n = 1$.

3.3.1. Animated playback

Animated playback is the most straightforward of all: summarize the history of the sketch being drawn by simply replaying the operation sequence and show how the sketch evolves as strokes and formatting changes are executed. There is no attempt to reduce the number of frames, yet several design issues must still be resolved, the primary ones being timing, animation speed, and user control.

Timing refers to how to map the timestamped sketch operations in the sketch history to animated playback. While it may be useful to see the relative timing of operations (i.e., that the user took more time drawing one part of the sketch than another), most of the time the timestamps can be ignored and the operations simply be drawn in order with uniform timing. Speed, on the other hand, refers to the overall speed with which the animation is played back. Typically, an animated history will be played back faster than real time, although this could be left in the hands of the user. In addition, easing the speed of the animation in and out using temporal distortion may help perception (Dragicevic et al., 2011). Finally, user control concerns the interaction model that is used to control the animation; we use a media player metaphor, including play, pause, and rewind buttons, as well as a scrubber bar for directly controlling frame position.

3.3.2. Sequential aggregation

Before being able to summarize a sketch using either a comic strip or a summary frame, we need to find a way to reduce the

number of frames in the sequence, ideally without losing too much of the information encoded in the history. Furthermore, our design constraints mandate that we preserve the chronological order in the history, yet provide a flexible data structure that can adapt to available screen space. To achieve this, we use a variant of standard agglomerative (or hierarchical) clustering (Jain et al., 1999) where we preserve the order by only aggregating adjacent frames. This *sequential aggregation* algorithm preserves the chronology of frames, whereas traditional hierarchical clustering typically considers the distances between *all* items at each iteration of the algorithm.

For the aggregation to begin, we first need to compute the distances between each neighboring frame in the sketch history. The distance between two sketch frames $d(F_{t-1}, F_t)$ is a measure of how much has changed between the two consecutive points in time, with the understanding that larger differences are more likely to be interesting than smaller ones. Several different distance metrics are possible (for example, how far the user's pen has traveled from one frame to the next); we explore this in more detail later in this paper.

With the distances computed, the aggregation algorithm proceeds in the normal way by iteratively combining the two adjacent frames with the shortest distance into a single aggregate with the two frames as children. Distances for the new aggregate are taken from the child items, so no new distances need to be computed. The hierarchy created during this process is stored in memory, and the iterative merging is repeated until only one aggregate remains (the root of the aggregation hierarchy). Fig. 2 shows how a sketch is iteratively clustered bottom-up into an aggregation hierarchy.

When creating a new aggregate, an impostor frame is chosen to represent the whole aggregate. Impostor selection should be stable, i.e., the impostor for an aggregate should be one of the two impostors of its children, otherwise the sketch history may change radically when expanding or collapsing an aggregate and thereby cause confusion. Of the two candidates taken from the children of an aggregate, our algorithm chooses the one with the most visible strokes because this branch of the aggregation hierarchy has the most information to convey to the user. Naturally, impostors for the original frames in the sketch history are those frames themselves.

3.3.3. Comic strip

The basic idea behind the comic strip history mechanism is to select a subset of $n \in (1, N)$ frames and show them in a static comic strip, i.e., in a list or grid where time runs from left to right, top to bottom. Fig. 3 shows a four-frame comic strip of a sketch.

Of course, selecting *which* frames from the sketch history to include is the real challenge. This is where the sequential aggregation algorithm presented above comes in useful. Given a particular distance function, the dendrogram resulting from this cluster analysis constitutes a level-of-detail tree that can be cut at any level depending on the amount of screen space available. Such a cut (temporarily disregarding the hierarchy above the tree cut) yields a *forest* of n subtrees. The impostor frame for each subtree root becomes a representative frame to use in the comic strip. This way, the sketch history can be represented by any number of frames $n \leq N$, including $n=1$ as well as $n=N$. For example, the comic strip in Fig. 3 resulted from cutting Fig. 2 at depth 3, yielding 4 subtrees represented by frames 2, 7, 8, and 12 from the sketch sequence.

3.3.4. Summary frame

Our intention with the summary frame was to provide a single frame where all events were captured. This would represent a compact, space-efficient visualization of a dynamic event sequence that is diametrically opposite from animation, which uses all frames. We achieve this by computing the union of all strokes (even those who were later erased) drawn at different points in time during the sketch history into a single frame, and then adding visual annotations to highlight and enumerate the major events during the history. Fig. 4 shows an example of a summary frame for Fig. 2.

Again we come back to the problem of identifying the major representative events during the sketch history, and the sequential aggregation algorithm again provides the solution. Let the *diameter* of an aggregate be defined as the sum of the sketch distances between all of its children. Instead of the user explicitly choosing the number of frames to display in a comic strip, we traverse the dendrogram depth first, only stopping the traversal for each branch when the current aggregate has a diameter less than or equal to a threshold value. The resulting list of frames becomes the events to highlight. For example, given a threshold of .45, such a traversal of the dendrogram in Fig. 2 would yield frames 2, 5, 8, and 12. We then draw the bounding box of the strokes for each frame in the single summary frame and add an ordinal number. We use a green box and numeral for added strokes and red for erased strokes.

To be able to separate drawing and erasing into different events that can be conveyed with different colors, we must also use a distance metric that differentiates between these two types of

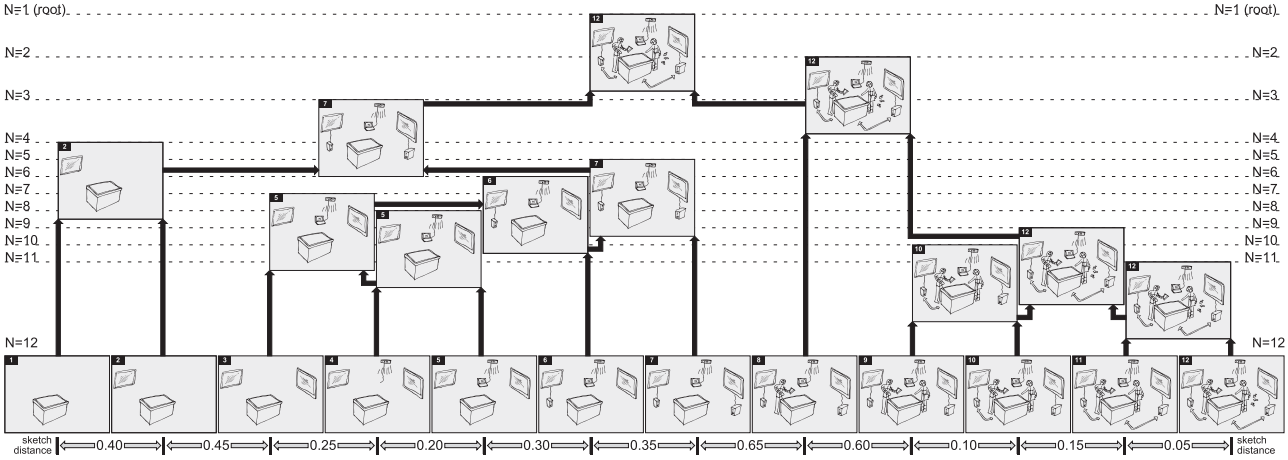

Fig. 2. Visual dendrogram of sequential agglomerative clustering of a sample sketch. The distances between individual frames (bottom of image) only have to be computed once. Each sketch impostor shown in the dendrogram (above the bottom sequence) represents a cluster of two child sketches, and is chosen from the two children as the one with the most strokes (i.e., most information).

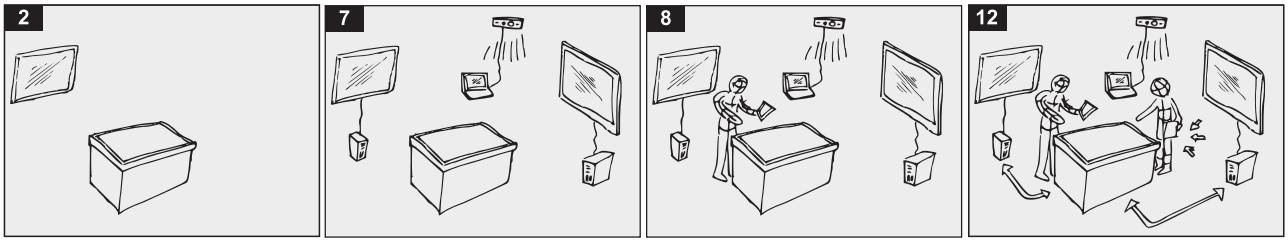

Fig. 3. Four-frame comic strip representation of the sketch shown in Fig. 2. The four frames were chosen by cutting the aggregation tree (dendrogram) in Fig. 2 at depth 3 ($N=4$), yielding four impostor children.

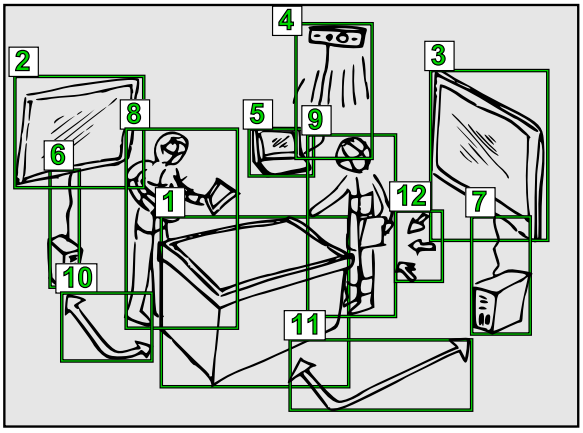

Fig. 4. Summary frame for Fig. 2 with ordinal numbers given the temporal sequence of events. The green bounding box communicates objects being added; removed objects (none in this sequence) would be drawn using a red color instead. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

events. This in turn will cause the aggregation hierarchy to become a forest instead of a single tree; a tree consisting of drawing strokes simply cannot be merged with a tree consisting of erasing strokes. A long sequence of alternate drawing and erasing will thus yield many events in the resulting summary frame, even if each is minor.

3.4. Implementation

While we expect a production sketch tool to run on a mobile tablet, our prototype Sketcholution implementation (Fig. 1) is built in Java using the Piccolo2D (Bederson et al., 2004) library for vector graphics. The main component in the Sketcholution software is the sketch history, implemented as a queue of operations using the Command design pattern. Because early design requires minimally-invasive interfaces and as close to natural interaction as possible, we chose not to expose this functionality in our prototype.

4. Sketch distance metrics

Our sketch summarization approach has two major advantages compared to more traditional summarization approaches (Gonzales and Woods, 2008): it is bottom-up, and it can be performed online. To achieve this, we need appropriate sketch distance metrics $d(F_{t-1}, F_t)$ that accurately portray the absolute difference between two consecutive sketch frames at timestamps $t-1$ and t. Below we review some of the distance metrics we have experimented within our prototype:

- *Temporal distance:* The temporal order in how people sketch and draw reflects how we think (Taylor and Tversky, 1992), which indicates that the time spent in drawing parts of a sketch may be an interesting distance metric to explore.
- *Euclidean distance:* Another simple metric is defined as the 2D distance (in pixels) that the user's pen has moved between the frames. This metric will cluster adjacent strokes tighter than distant ones, and is used in Fig. 1.
- *Perceptual distance:* Euclidean distance alone will not distinguish among color, thickness, and transparency, but such formatting changes are often important in segmenting a sketch history. A perceptual distance metric extends the Euclidean metric by assigning weights based on the visual saliency of the chosen stroke format.
- *Salience:* A more complex distance metric would use the stroke salience, where the salience of a stroke at a specific point is related to the instantaneous curvature of the stroke. This local salience can be found robustly by carrying out a local Principal Component Analysis. We do this by constructing a covariance matrix M of the coordinates within a small time window $N(p_i)=\{p_j \mid j=i-w, j, i+w\}$ of width w and then performing an eigendecomposition of M to yield eigenvectors (v_S, v_L) and eigenvalues (λ_S, λ_L) of M. We define the salience at an instant as the ratio of the eigenvalues:

$$\alpha = \frac{\lambda_S}{\lambda_L} \tag{1}$$

For a perfectly straight line $\lambda_S = 0$, while for a circular region $\lambda_S \approx \lambda_L$. The distance between adjacent frames can be found by computing the α difference:

$$d(F_{t-1}, F_t) = |\alpha_{t-1} - \alpha_t|. \tag{2}$$

This method is computationally efficient using a small frame window width w. From a pilot study (discussed below) we found that a frame width of 5–10 strokes is optimal to model intended sketch curvatures while filtering out small erratic moves.

Sketching is a highly subjective and application-dependent process. Hence it is difficult to come up with a single distance metric that fits all users and applications. In fact, composite distance metrics that weigh together more than one of the above metrics are often good compromises. A general methodology to select distance metrics given an application is left for future work.

In many real-life design scenarios, such as that discussed by Schmidt et al. (2007), continuous refinement and correction are an essential sketch activity. The final outcome has all the details and color on top of the original design scaffolding, which makes it hard for the user to review and have a quick overview for the whole process. In this case, a good distance metric may be based on color and thickness combined with temporal distance, as sketchers typically change their pens for either another color or thickness, and may often pause before starting another stage in the layered sketch.

5. User study

At this point, we have derived three different techniques for conveying sketch history—animation, comic strip, and summary

frame—but have no data on which of these techniques perform best under different conditions. Our intuition is that comic strips and summary frames will outperform animation when the testing sample is long and complex. On the other hand, for short and simple tasks, we expect that animation will be fastest since such tasks do not require summarization (Ajmal et al., 2012), and animation is also more familiar to participants.

To find the answer to this question, we performed a controlled user study that compared the completion time of participants recovering the temporal sequence of visual components being added and removed from a sample sketch.

5.1. Pilot study

Prior to performing the full experiment, we first conducted an in-depth pilot study involving 16 participants. In the pilot study, we used a simple task sketch history recall task (similar to the actual study below), but used the pilot to determine suitable factors and conditions. During the pilot study, we found that the length of a sketch history and the context components that serve as background are interesting factors affecting the overall difficulty of the tests. However, the initial experimental design for the pilot study was not able to adequately distinguish between different difficulty levels. After increasing these two factors, we were able to obtain results so that groups were well separated based on completion time. We carefully selected the parameters to keep a balance so that participants will have enough challenge in the hard cases while their performance will not be affected by fatigue. This increased difficulty leads to increased session duration. As a consequence, we were forced to take fewer participants in the formal user study due to limited time and budget. We explain the details of the two experimental factors below.

For the techniques of Single Frame and Comic Strip, we also observed that obtaining the correct number of frames is essential to their effectiveness, i.e. the time for the participants to finish certain task using such techniques. Therefore, we developed an automatic segregation approach that separates the sketch into discrete frames. We tested this approach in the pilot study to reach an optimal separation rate, and used the parameters to the actual study with minor changes.

All of the parameters for factors in the below text were chosen to best separate the efforts of participants while keeping the tasks manageable.

5.2. Participants

We recruited 12 paid participants (6 male, 6 female) for this user study (each paid $10). The participants were self-selected from the student population at our university, were aged between 20 and 31 years of age, had normal or corrected-to-normal vision, and were proficient computer users (all demographics were self-reported). A majority of participants were engineering students with a design background.

5.3. Apparatus

We conducted the experiment on a standard desktop computer equipped with a 24-inch LCD screen (resolution 1600×1200), a standard keyboard, and a mouse. The Sketcholution software was maximized to fill the entire screen during the experiment.

5.4. Task and data

Based on previous work on graphical histories (Heer et al., 2008) and causality in analytics (chul Kwon et al., 2012), we selected a task based on recovering the causality of events in an interaction history. In our case, each trial consisted of the participant using one of the three sketch history mechanisms to view the evolution of a sample sketch.

The sample sketch consisted of several smaller "sketch objects": D distractor objects, A added objects, and R removed objects. A sketch object is defined as a relatively small (approximately 100–300 pixels in dimension) sketched component of a clearly distinguishable physical item, such as car, face, or heart shape. All objects were recorded using one-pixel wide black strokes for simplicity. A sample sketch was constructed by assembling a sequence of A sketch objects being added to the sketch, and R objects being removed. This design was chosen to allow us to procedurally generate new trials from scratch. Sketch objects were selected from a pool of 23 discrete objects of different complexity, and each object was used at most once per trial. The order of object additions and removal was randomly permuted, whereas distractor objects never changed during a trial.

The user study task was to use the given sketch history mechanism to recover the order of added and removed sketch objects in the overall sketch. Participants were given a randomly ordered list of the sketch objects involved in the sample sketch and were asked to specify the order. This was done by simply selecting a number representing its order of appearance or disappearance for each object using combo boxes (Fig. 5). A trial could only be completed when the exact order for each sketch was given (incorrect clicks were recorded). The experimental platform silently recorded the completion time from the beginning of the trial to the moment the user clicked the submit button once the answer was correct.

5.5. Factors

We included the following factors in the experiment:

5.5.1. History mechanism (H)

This factor modeled the sketch history mechanism H used:

- *Animation:* Basic animation playback where the participant can play, pause, and rewind the action, fast-forwarding, fast-rewinding, as well as moving the seeker bar to any time in the animation. All these controls were available throughout the trials. The playback window was stretched to fill the entire available screen space, minus the playback control interface.
- *Comic strip:* The Sketcholution comic strip where the available screen space is subdivided into smaller frames, each showing a representative snapshot of the history. The participant was able to control the number of frames using a slider; the initial number was five. Increasing the number caused the individual frames to become smaller, but showed a larger number of history highlights (and vice versa).
- *Summary frame:* The Sketcholution summary frame where a single annotated picture of the sketch history was stretched to fit the entire screen. The number of significant events to annotate in the summary was determined from distance thresholds found using a pilot. This number could also be controlled by the participant using a slider similar to the slider used for the comic strip; increasing the event number would simply highlight more events in the summary frame.

Even if our sketch objects were discrete, we did not artificially add delimitations between them, but used the *complete* stroke-level history of the sketch objects constituting the history.

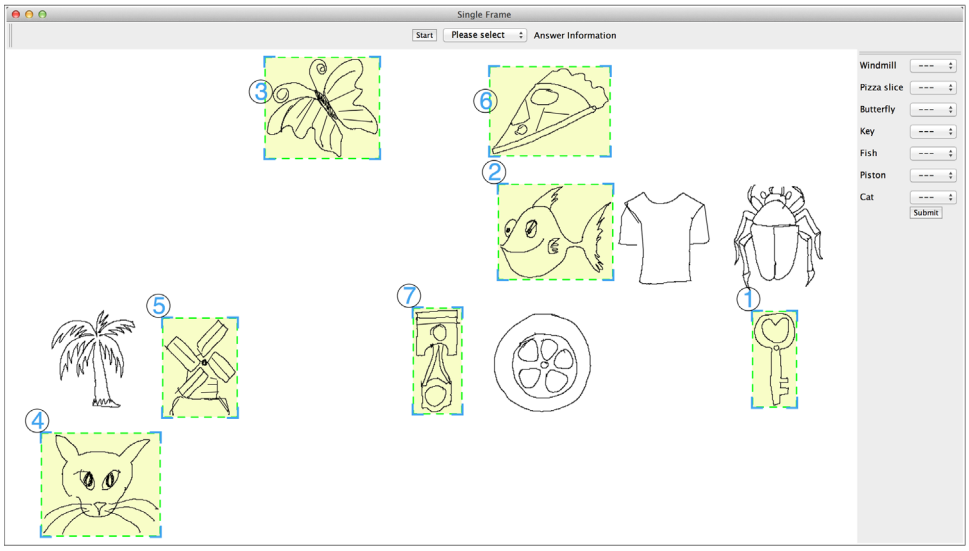

Fig. 5. Screenshot from the study setup using the summary frame technique. In this example, we chose both temporal and Euclidean distance metrics for better segmentation. Temporal distance is added with a weight of .3 for better segmentation rates based on results from our pilot study.

5.5.2. Sketch complexity (C)

The length of a sketch history will clearly have an impact on ; the difficulty of recovering the sequence of events; the longer the history, the more elements to survey and remember. To model this, we introduced a sketch complexity factor C as the sum $C = A + R$, i.e., the total number of added and removed sketch objects constituting the history. Given C, values for A and R were selected randomly. The pilot study had a low Sketch Complexity: 3, 5, 8. We increased it to 5, 8, 11 in the actual user study to increase the difficulty.

5.5.3. Distractor level (D)

Sketches often contain components that either serve as context (for example, reference figures or background scenery), are part of earlier work, or were created by others in a collaborative setting. These visual components still add to the visual complexity of the sketch. We speculate that these invariant components in a sketch that do not change over the course of the sketch history may therefore impact performance. We thus added a distractor factor D that modeled the number of such distractors that are visible in the sketch. We selected the distractor factor as a set of low (2 distractors), medium (4 distractors), and high (6 distractors) in the pilot study. We found that this configuration was not enough to serve its purpose, so we increased the number of distractors to 2 (low), 4–6 (medium), and 8–10 (high) distractors.

5.6. Experimental design

We used a full-factorial within-participants design:

	12	participants
×	3	*History mechanisms H* (anim, strip, frame)
×	3	*Sketch complexities C* (5, 8, 11)
×	3	*Distractor levels D* (*low*, *medium*, *high*)
	324	Total trials (27 per participant, training excluded)

Trials were organized in blocks for each history mechanism. Block order was balanced using Latin square participants to counteract learning effects; other factors were randomized. Trial completion time (in seconds) was the single dependent variable.

5.7. Procedure

An experimental session started with the participant arriving, reading and signing the consent form, and being assigned an identifier and history mechanism block order. The administrator then explained the general goals and task for the experiment. Each block started with the administrator demonstrating the use of the history mechanism and showing how to utilize it to solve a practice trial. The participant was then given two practice trials to solve. These trials were not timed and the participant was allowed to ask questions about the history mechanism and the task during this time. Since trials were timed individually, participants could rest between any two trials at will.

Individual trials started with an empty intermission screen that instructed the participant to click on a button to start the next trial. Clicking on this button caused the empty screen to be replaced by the history mechanism interface. A separate panel provided a randomly ordered list of the names of the sketch objects involved in the trial. The participant could select the ordinal number for each object's appearance or disappearance during the sketch history using a dropdown menu. There was no need to specify whether the object had been added or removed during the sequence. Participants clicked on a submit button to finish the trial, but were allowed us to proceed only if the provided order was correct.

After finishing a full block of 3×3 conditions per mechanism, participants were asked to rate their perception of the efficiency, ease of use, and enjoyability of the history mechanism on a 1–5 Likert scale. After finishing all three blocks, they were asked to provide any general comments or feedback on any of the history mechanisms or the experiment. A full experimental session lasted approximately 60 min, including training and questionnaires.

5.8. Hypotheses

We formulate the following hypotheses about the study:

H1 *Both Sketcholution interfaces will be significantly faster than animation.* We believe that the visual summaries provided by the comic strip and summary frame will be more efficient representations of sketch history than animation.

H2 *High sketch complexities will cause significantly longer completion time than low complexities.* The number of discrete

components being added or removed from the sketch will directly influence the time to solve a trial.

H3 *Many distractors will cause significantly longer completion time than few distractors.* Analogously, many distractors will make the task more difficult, and thus slower.

6. Results

We here report on the completion times and subjective ratings for the three different history mechanisms.

6.1. Completion time

We analyzed completion times using a repeated-measures analysis of variance (RM-ANOVA, all assumptions fulfilled) and found a significant main effect of history mechanism H on the completion time ($F(2,22)=106.59, p<.0001$). Fig. 6 shows a visual summary of the completion times; averages of 61.7 s for animation, 34.2 for summary frame, and 32.1 for comic strip. A post hoc Tukey HSD indicated that the difference between animation and comic strip as well as between animation and summary frame was significant ($p<.05$), with animation on average having twice as long completion time as the other two mechanisms. There was no significant difference between completion times for comic strip and summary frame ($p=.60$).

Not surprisingly, the sketch complexity C had a significant main effect on completion time ($F(2,22)=215.67, p<.0001$). A post hoc Tukey HSD showed that all complexity levels had significant pairwise differences ($p<.05$) in the order $5<8<11$ in ascending completion time. Similarly, analysis of the distractor level D on completion time yielded a main significant effect ($F(2,22)=4.89, p<.01$). A pairwise Tukey HSD showed significant differences for $2<10$ number of distractors ($p<.05$).

Finally, we found a significant interaction between history mechanism H and complexity C ($F(4,44)=8.16, p<.0001$). A Tukey HSD post hoc test indicated that animation had comparable performance to both comic strip and summary frame at low complexity (5 events), but was significantly ($p<.05$) slower for 8 and 11 events.

6.2. Subjective ratings

Fig. 7 depicts boxplots of the subjective ratings for each history mechanism on efficiency, ease of use, and enjoyability. We analyzed the 5-point Likert scale of subjective ratings for the different history mechanisms and found that the efficiency and ease of use rating were significantly different across the three mechanisms (Friedman test, $p<.05$), but the enjoyability had no significant difference (Friedman tests, $p=.12$ for enjoyability). A Friedman post hoc test showed significant differences of efficiency between comic strip and animation ($p<.05$), and significant differences of ease of use between comic strip and animation ($p<.001$) and between comic strip and summary frame ($p<.05$).

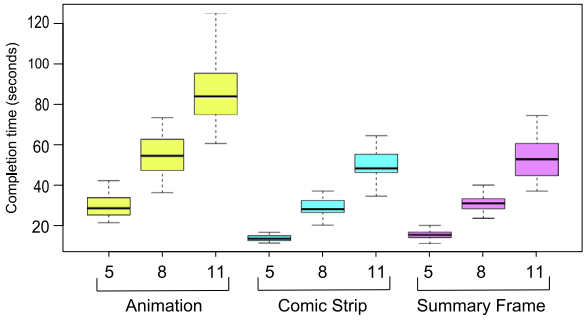

Fig. 6. Average completion time as a function of history mechanism H and complexity C. Comic strip and summary frame show significantly faster times than animation.

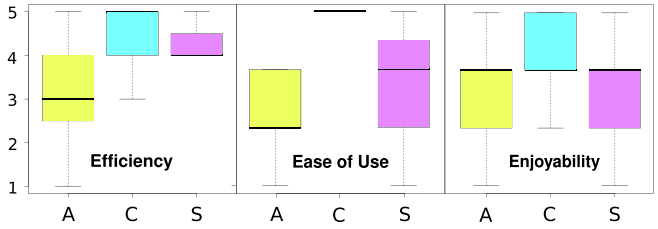

Fig. 7. Subjective ratings (Likert 1–5). Comic strip and summary frame are higher than animation for efficiency and ease.

7. Discussion

We can summarize our findings as follows (discussed below):

- Both comic strip and summary frame are considerably faster than animation (accepting **H1**);
- The complexity of trials (i.e., the number of components) had a direct impact on trial difficulty (accepting **H2**); and
- The number of distractors in the sketch history had significant impact on completion time (accepting **H3**).

7.1. Explaining the results

Our findings were in line with our initial hypotheses. Both summary frame and comic strip showed better performance than animation. The most likely explanation for these results is that our two novel methods allow the user to overview the entire event sequence without relying on a dynamically changing visual representation and requiring no user interaction. Users were simply faster in recovering causal order using Sketcholution for these reasons.

We made several interesting observations on how participants used the different techniques to solve the task. For comic strip, participants seemed to move from frame to frame using the bounding boxes to mark changes. The comic strip partitions events into an orderly grid, and so is faster than an animation. Similarly, the numbered bounding boxes in the summary frame impose structure on the event sequence, relieving the user from having to replay the sketching history and explicitly remember the ordering.

It is hardly surprising that the complexity of a trial is indeed governed by the number of sketch components that are added or removed during the history, and this is a direct impact from the need to provide more information to the testing platform to recover the sequence. Animation of complex sketches involved a noticeably large amount of user interaction to recognize and recall events in the sketching sequence. The summarized nature of comic strip and summary frame made it significantly easier to identify the sketching sequence in the presence of complexity.

Distractors have a significant impact on completion times. This is again not surprising: a higher number of distracting items clearly make it more difficult to memorize the event sequence, especially for animation. On the other hand, comic strip and summary frame have bounding boxes to indicate the changes which makes the distractors less effective. During the actual test, the combination of high sketch complexity and distractor level significantly increased the completion time.

7.2. Generalizing the results

History mechanisms are pervasive in HCI, so our findings are potentially important beyond the sketch interactions discussed

here. For example, it is notable that comic strips are common in several graphical history mechanisms (Heer et al., 2008). Even though animation has comparable enjoyability to summary frame, it requires user interaction to start, stop, and rewind. In contrast, both summary frame and comic strips are static and require no user navigation.

On a more general level, both Sketcholution techniques proposed in this paper are examples of *spatialization* (Javed and Elmqvist, 2013): transforming temporal data into spatial representations. For Sketcholution, the stroke-level interaction sequence represents the temporal data, and the techniques only differ in which spatial representation is used. Comic strips use small multiples, whereas the summary frame instead orders the temporal data in the same image. Both approaches have their own pros and cons, and more research is needed to investigate the relative merits of each technique in detail.

However, the common denominator for all spatialization methods is that they run the risk of introducing high visual clutter and complexity. This is certainly also true for the Sketcholution techniques; the comic strip scales down and multiples the sketch canvas, yielding increased clutter proportional to the number of frames as well as smaller visual resolution for individual frames, and the numbering and bounding boxes used in the summary frame will cause similar levels of increased clutter and visual complexity. The saving grace for sketches in comparison to, for example, digital video is that the overall level of visual details tends to be relatively low for sketches. However, it is equally true that a sketch will become increasingly complex as it is worked on over a longer period of time. Our Sketcholution techniques will not scale well with such long sequences and more work is needed to accommodate them.

There are several strategies that we could adopt to combat scale and complexity in the future. Simple optimization rules could be applied to the interaction sequence that eliminates redundancy, such as discarding strokes that were immediately deleted or undone. Multiple operations could be *chunked*, similar to our work in the skWiki system (Zhao et al., 2014), where a series of consecutive stroke or erase operations are grouped together into a single meta-operation. Finally, as observed earlier in this paper as a rationale for introducing distractors in our user study, there are often parts of a sketch that do not change much. Such invariant components could be ghosted so that they are less visually salient and thus less of a distraction.

Another limitation of the Sketcholution approach is that sequential agglomerative clustering may not work well with holistic sketching where the illustrator evenly adds detail across the entire canvas (i.e., a form of breadth-first sketching) as opposed to working on different parts of the sketch separately (i.e., depth-first). We may have to relax the chronology-preserving design goal to better accommodate such behavior. We leave this for future work.

Finally, our intended use-case for Sketcholution is to embed the methods into the skWiki collaborative multimedia system (Zhao et al., 2014). skWiki stores persistent versions of sketches, rich text, and images using the concept of a path as a sequence of operations yielding a particular data object. We plan to use the Sketcholution method to visualize the evolution of a sketch in the system. While our study shows clear advantages to the comic strip and summary frame techniques proposed in this paper, a realistic implementation in the skWiki system may still provide all three options. In fact, hybrid solutions may also be desirable: for example, an animation could highlight recent changes using bounding boxes similar to the comics strip and summary frame. Another approach may be to chunk the animation into individual segments, resulting in a hybrid animated comic strip.

8. Conclusions and future work

We have presented Sketcholution, a new approach to capturing and summarizing the interaction history for digital sketching. The approach is based on an order-preserving agglomerative clustering algorithm that summarizes the stroke-level evolution of a sketch from the bottom up, yielding a visual summary that can be adapted to any desired level of detail. Comic strip and summary frame are two concrete history mechanisms that use the aggregation tree resulting from this cluster analysis, and they present the results as a sequence of highlighted frames, or as a single annotated frame, respectively. Results from a controlled user study indicate that both comic strip and summary frame significantly outperform animation. Furthermore, both are static and require no user interaction.

In the future, we will keep exploring applications in the domain of artistic sketching where our techniques can be integrated into a tool of segmenting sketches based on their time stamps. Our techniques can also be handy in the field of technical sketches and sketches of process schematics since such sketches tend to have multiple layers where our tools can separate the overlapped layers in certain cases. Our future work will also study how to use findings from the user study in a revision control framework for sketches. It should be possible to integrate our Sketcholution techniques directly into the editor interface. Furthermore, beyond vector drawing, automatically extracting animation summaries is appealing for many domains, including storyboarding, general interaction histories, and video summarization.

Acknowledgments

This work is partially supported by the NSF Award Numbers 1245780 and 1422341 through Intelligent Information Systems, the NSF Award No.1227639 Division of Undergraduate Studies and the Donald W. Feddersen Chair Professorship support from the School of Mechanical Engineering. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

Appendix A. Supplementary data

Supplementary data associated with this article can be found in the online version at http://dx.doi.org/10.1016/j.ijhcs.2015.04.003.

References

Ajmal, M., Ashraf, M., Shakir, M., Abbas, Y., Shah, F., 2012. Computer Vision and Graphics. Video Summarization: Techniques and Classification, vol. 7594. Springer, Berlin, Heidelberg, pp. 1–13.

Barnes, C., Goldman, D.B., Shechtman, E., Finkelstein, A., 2010. Video tapestries with continuous temporal zoom. ACM Trans. Graph. 29, 89:1–89:9.

Bederson, B.B., Grosjean, J., Meyer, J., 2004. Toolkit design for interactive structured graphics. IEEE Trans. Softw. Eng. 30 (8), 535–546.

Berlage, T., 1994. A selective undo mechanism for graphical user interfaces based on command objects. ACM Trans. Comput.–Hum. Interact. 1 (3), 269–294.

Chen, Q., Grundy, J., Hosking, J., 2003. An e-whiteboard application to support early design-stage sketching of UML diagrams. In: Proceedings of the IEEE Symposium on Human Centric Computing Languages and Environments, pp. 219–226.

Chen, H.-T., Wei, L.-Y., Chang, C.-F., 2011. Nonlinear revision control for images. ACM Trans. Graph. 30, 105:1–105:10.

chul Kwon, B., Javed, W., Ghani, S., Elmqvist, N., Yi, J.S., Ebert, D., 2012. Evaluating the role of time in investigative analysis of document collections. IEEE Trans. Vis. Comput. Graph. 18 (11), 1992–2004.

Dorsey, J., Xu, S., Smedresman, G., Rushmeier, H., McMillan, L., 2007. The mental canvas: a tool for conceptual architectural design and analysis. In: Proceedings of the IEEE Pacific Conference on Computer Graphics and Applications, pp. 201–210.

Dragicevic, P., Bezerianos, A., Javed, W., Elmqvist, N., Fekete, J-D. 2011. Temporal distortion for animated transitions. In: Proceedings of the ACM Conference on Human Factors in Computing Systems, Vancouver, BC, Canada, pp. 2009–2018.

Eccles, R., Kapler, T., Harper, R., Wright, W., 2008. Stories in geotime. Inf. Vis. 7 (1), 3–17.

Gonzales, R.C., Woods, R.E., 2008. Digital Image Processing, Pearson. Prentice Hall, Upper Saddle River, NJ.

Greenberg, S., Carpendale, S., Marquardt, N., Buxton, B., 2011. Sketching User Experiences: The Workbook. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Heer, J., Mackinlay, J.D., Stolte, C., Agrawala, M., 2008. Graphical histories for visualization: supporting analysis, communication, and evaluation. IEEE Trans. Vis. Comput. Graph. 14 (6), 1189–1196.

Jain, A.K., Murty, M.N., Flynn, P.J., 1999. Data clustering: a review. ACM Comput. Surv. 31 (3), 264–323.

Javed, W., Elmqvist, N., 2013. ExPlates: spatializing interactive analysis to scaffold visual exploration. Comput. Graph. Forum 32 (2), 441–450.

Kara, L.B., Shimada, K., 2008. Supporting early styling design of automobiles using sketch-based 3D shape construction. Comput.-Aided Des. Appl. 5 (6), 867–876.

Kurlander, D., Feiner, S., 1988 Editable graphical histories. In: Proceedings of the IEEE Workshop on Visual Language, New Orleans, Louisiana, USA, 1988, pp. 127–134.

Li, G., Cao, X., Paolantonio, S., Tian, F., 2012. SketchComm: a tool to support rich and flexible asynchronous communication of early design. In: Proceedings of the ACM Conference on Computer Supported Cooperative Work, Seattle, Washington, USA, pp. 359–368.

Ma, K.-L., 1999. Image graphs—a novel approach to visual data exploration. In: Proceedings of the IEEE Conference on Visualization, San Francisco, California, USA, pp. 81–88.

McKim, R.H., 1972. Experiences in Visual Thinking. Brooks/Cole Pub. Co, Monterey, CA.

Mckoy, F.L., Vargas-Hernández, N., Summers, J.D., Shah, J.J., 2001. Influence of design representation on effectiveness of idea generation. In: Proceedings of the ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Pittsburgh, PA, USA, pp. 1–10.

Nancel, M., Cockburn, A., 2014. Causality: a conceptual model of interaction history. In: Proceedings of the ACM Conference on Human Factors in Computing Systems, Toronto, Ontario, Canada, pp. 1777–1786.

Noris, G., Skora, D., Shamir, A., Coros, S., Whited, B., Simmons, M., Hornung, A., Gross, M., Sumner, R., 2012. Smart scribbles for sketch segmentation. Comput. Graph. Forum 31, 2516–2527.

Rieber, L.P., 1990. Using computer animated graphics in science instruction with children. J. Educ. Psychol. 82 (1), 135.

Schmidt, R., Isenberg, T., Jepp, P., Singh, K., Wyvill, B., 2007. Sketching, scaffolding, and inking: a visual history for interactive 3D modeling. In: Proceedings of the ACM Symposium on Non-Photorealistic Animation and Rendering, San Diego, California, USA, pp. 23–32.

Sutherland, I.E., 1964. Sketch pad a man-machine graphical communication system. In: Proceedings of the SHARE Design Automation Workshop. ACM, pp. 6.329–6.346.

Taborda, E., Chandrasegaran, S.K., Ramani, K., 2012. ME 444: Redesigning a toy design course. In: Proceedings of the International Symposium on Tools and Methods of Competitive Engineering, Karlsruhe, Germany, pp. 597–608.

Taylor, H., Tversky, B., 1992. Descriptions and depictions of environments. Mem. Cognit. 20 (5), 483–496.

Tversky, B., 1999. What does drawing reveal about thinking? In: Gero, J.S., Tversky, B. (Eds.), Visual and Spatial Reasoning in Design, Sydney, Australia, pp. 93–101.

Tversky, B., Zacks, J., Lee, P.U., Heiser, J., 2000. Lines, blobs, crosses and arrows: diagrammatic communication with schematic figures. In: Proceedings of the International Conference on Theory and Application of Diagrams. Springer, Berlin, Heidelberg, pp. 221–230.

Tversky, B., Morrison, J.B., Betrancourt, M., 2002. Animation: can it facilitate? Int. J. Hum.–Comput. Stud. 57 (4), 247–262.

Ullman, D.G., Wood, S., Craig, D., 1990. The importance of drawing in the mechanical design process. Comput. Graph. 14 (2), 263–274.

Zhao, Z., Badam, S.K., Chandrasegaran, S., Park, D.G., Elmqvist, N., Kisselburgh, L., Ramani, K., 2014. skWiki: a multimedia sketching system for collaborative creativity. In: Proceedings of the ACM Conference on Human Factors in Computing Systems. ACM, Toronto, ON, Canada, pp. 1235–1244.

Zhu, B., Iwata, M., Haraguchi, R., Ashihara, T., Umetani, N., Igarashi, T., Nakazawa, K. 2011. Sketch-based dynamic illustration of fluid systems. ACM Trans. Graph. 30, 134:1–134:8.