

Visualizing, Editing, and Inferring Structure in 2D Graphics

Sara L. Su

Computer Science & Artificial Intelligence Lab.
Massachusetts Institute of Technology
Cambridge, MA, USA
sarasu@mit.edu

ABSTRACT

Vector graphics software provides powerful tools for specifying the hierarchical structure of objects in an illustration document. This structure is useful for future editing, but building it explicitly (*e.g.* by grouping and ordering objects) is a tedious process for the user. We seek methods for inferring structure in a document and exploiting it. We tackle three key problems: First, we introduce a *storyboard* visualization that enables non-sequential browsing and manipulation of graphical editing history, that is, the actions used to create an illustration. Second, we discuss using this history to infer structure. Finally, we discuss methods for using this structure to explore the design space of vector graphics. Our broad aim is to help users take fuller advantage of illustration on the computer by introducing methods to visualize, manipulate, and explore alternative versions of their designs.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces - *Graphical user interfaces*.

General Terms: Algorithms, Design, Human Factors

Keywords: Vector graphics, design history, undo, storyboards, visualization, interaction techniques

INTRODUCTION

Digital tools have introduced great flexibility to the illustrator's workflow. In addition to providing a rich toolbox of graphical elements, programs such as Adobe Illustrator, Macromedia FreeHand, and Microsoft PowerPoint facilitate exploration, trial-and-error editing, and refinement of designs. Our goal is to improve these interactions with an illustration by leveraging its editing history. In this article, we summarize our ongoing and planned work on three problems: visualization of graphical history, inference of an illustration's structure, and generation of design variations.

First, we propose a metaphor for visualizing and interacting with the history of an illustration. With the ubiquitous *undo* command, users can revisit the recent history of a document. Unfortunately, this is typically displayed as a stack of text lines that poorly conveys the creative process behind it. A number of systems have been developed to provide more descriptive lists [5, 17]; we focus on improving interaction by visualizing the editing history in spatial context.

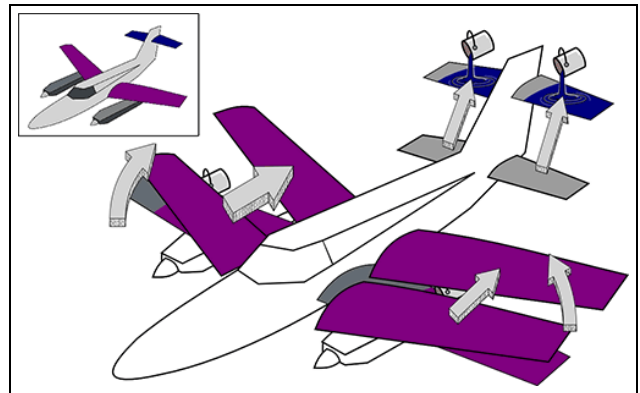


Figure 1: Automatically-generated visualization of the editing history of an illustration (boxed, upper left). Arrows and icons depict spatial transforms and color change actions performed by the user. (Plane used under the GNU Free Documentation License: <http://commons.wikimedia.org/wiki/Image:Achsen-cessna2.svg>)

We are inspired by Kurlander and Feiner, who introduced *graphical histories* [11]. Their system visualizes history in a series of panels, each one showing the before and after an action. This direct mapping (abstracting context and focusing on the modified element) highlights the action but can result in a proliferation of panels. Our solution is a hybrid layout that shows multiple actions per panel (Fig. 1). We will describe how this *storyboard* interface enables non-sequential access and modification of the drawing's history.

In addition to the history, we are interested in the *structure* of documents. As in 3D modeling and animation, a hierarchy of the objects in an illustration is valuable for future editing. Unfortunately, building this structure is tedious for the user and requires significant foresight. We argue that there is much that can be inferred about the document's structure from observing the actions used to create it. In particular, we look at selections. We observe that objects that belong to the same selection at some point in time can be considered as implicitly grouped. By noting the user's selections we can progressively infer document structure.

Finally, we want to help users explore the space of possible designs. Given the document structure, our goals are to generate variations from the parametric design space, cluster similar versions, and help the user navigate this space. The challenge is not only to generate variations, but to determine which are meaningful and how they should be presented.

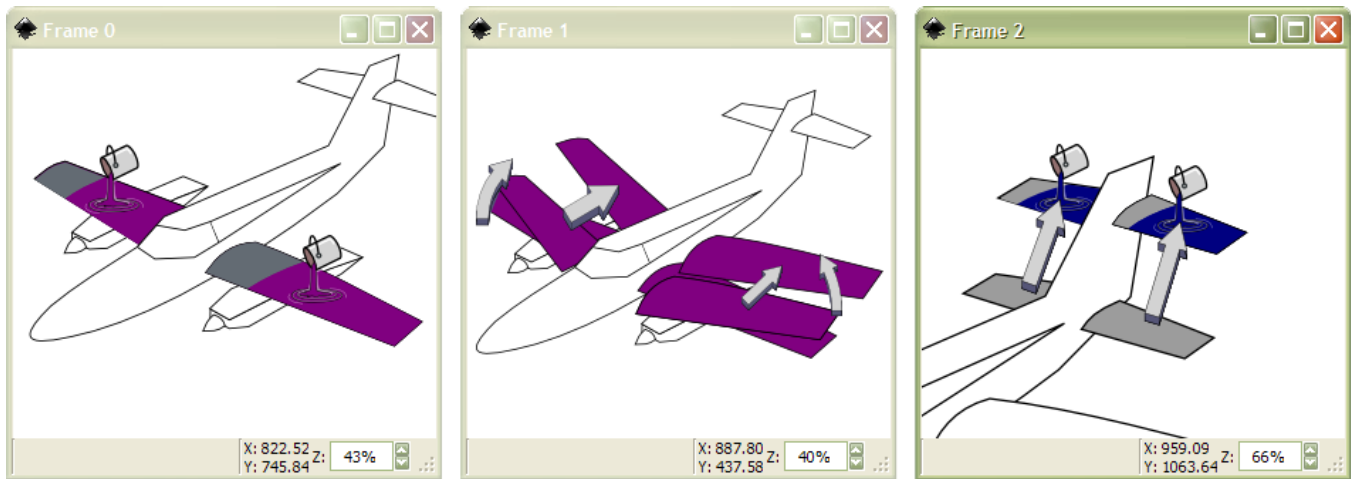


Figure 2: Storyboard visualization. The user can click on any depiction to undo the corresponding action. Non-sequential undo is minimally destructive; not all subsequent actions will be canceled, only those of the same dependency class.

STORYBOARDING FOR NON-SEQUENTIAL INTERACTION

All modern text and graphics editors support a notion of history. The undo mechanism not only makes it easy to discard recent mistakes, it allows the user to compare the design before and after a modification. In many programs, this has been extended to storing the full history of actions, and users can roll back to arbitrary points in time. However, users are limited to sequential and causal exploration of this history.

We present a new visualization that enables the user to easily select and undo previous actions in a non-sequential manner. Our work is inspired by assembly instructions [2], maps [3], comics [14], and storyboards used in filmmaking [8].

Visualizing History in Context

In the *storyboard mode* of our software, the user sees the illustration’s history in one or more frames, each containing *action depictions*—graphical metaphors of editing actions (Fig. 1). This interface provides instant access to any action, allowing the user to consider it in context rather than in a disconnected list view. It has been shown that in-place visualization of user interface transitions [4] and design variations [19] improves comprehension. Guided by this work and by principles of cognitive psychology [9, 20], we have designed the action depictions to be simple, yet dynamic and intuitive. Together, these depictions can be seen as the “assembly instructions” for an illustration.

Of the exciting recent work on history of text [12], spreadsheets [10], and graphics [15], our approach most closely follows from Chimera [11], which displays the history of an illustration as a series of panels. Every user action is visualized with a before-and-after in a separate panel. We extend this work with depictions that show dynamics of the actions and with a storyboard showing multiple depictions per frame. This requires careful evaluation of the degree of clutter; we use a simple but effective bounding box test on the depictions to determine when a new frame should be created (Fig. 2).

Our decision to depict multiple actions per frame is inspired by the recent work of Goldman *et al.* on *schematic story-*

boards [7]. Our interface shares some of the same annotation elements, but while schematic storyboards visualize an existing video sequence as a static image, our goal is to visualize the steps of constructing an illustration.

Users interact with the storyboard in an intuitive manner. The storyboard mode can be activated at any time during the editing process, and once activated, action depictions appear on top of the document for selection. Our approach is most relevant to illustrative figures for a slide presentation or article, where the drawing complexity is moderate. Highly complex, fine art illustrations raise scalability issues that are beyond the scope of this work. We are continuing to refine the appearance to improve user comprehension and differentiate the depictions from the underlying illustration.

We now describe two useful applications of the storyboard visualization: a non-sequential undo mechanism and color-coding for collaborative editing.

Non-sequential Undo

Sequential undo, found in all existing illustration programs, removes actions from the top of the undo stack, and upon returning to a previous state, all actions between then and the current state must be canceled as well. With our storyboard, the user can click on the depiction for any previous action to index into the undo stack. He can undo any action, even if it is not the most recent one in the document.

Our *non-sequential undo* mechanism is minimally destructive. Rather than canceling all actions after the selected one, it cancels only those of the same type, those belonging to the same *dependency class*.

The first storyboard frame in Fig. 2 shows a color change of the wing, followed by a rotation, then a translation. A standard undo of the color would cancel all three actions. In contrast, non-sequential undo of the color (intuitively) does not affect the two spatial transforms. We define the following dependency classes: spatial transforms (translation, rotation), appearance changes (fill style, stroke style), and shape modifications (control point editing, scaling).

Collaborative Editing

The storyboard facilitates editing by multiple users in a manner similar to the “track changes” feature available in text editors such as Microsoft Word. When starting an editing session, a user identifies himself and each of his actions is tagged with this author ID. In storyboard mode, actions are then color-coded by author, thereby making visible the interactions between all the collaborators (Fig. 3). Currently, we support asynchronous collaboration across several editing sessions. Synchronous editing raises new challenges for undo that are worth investigating, including user understanding of the undo mechanism and user intent [1].

INFERRING STRUCTURE FROM INTERACTION

Our storyboards enable new modes of interaction with a drawing’s history. We now turn our attention to what the history can tell us about a drawing’s structure.

Many users organize their documents into components that can be individually changed, grouped, or added to a hierarchy of objects. As in animation, this hierarchical structure is extremely useful for future editing. For example, one might hierarchically group the parts of the plane (propeller belongs to engine belongs to wing belongs to body) so that translations to the body propagate to its children. Another common task is to change the style of all members of a group.

Building this structure into a document is a tedious task for the user that requires planning ahead. We wish to leverage structure without having to specify it explicitly. A key observation about drawing structure is that, in addition to explicit groupings, objects that belong to the same selection at some point in time can be considered as implicitly grouped.

Implicit Structure from Selections

We will develop methods for inferring implicit structure in an illustration from the actions used to create it. We will record user actions applied to multiple objects, taking our cue from work on structured text, such as Miller’s LAPIS system [16], whose *lightweight structure* enables reuse of text in a way not handled by basic pattern matching. Graphical documents are a logical next step to consider, as notions of grouping and regions are less constrained than in text.

We will consider user selections by object, region, and property (e.g. color or object type). The results of these selections implicitly build structure in the document. Our strategy is to use these results to extract the full implicit hierarchy of the document a piece at a time.

We do not seek a fully automatic method for inferring structure, as many users respond negatively when automatic systems do not behave as they expect; expert users may even resent systems that try to be “too smart” or that do not provide the fine level of control they desire. Our approach is to make adaptive suggestions based on the user’s previous actions. We can learn users’ preferences when they are presented with alternatives, for example, which of the possible outcomes for an ambiguous undo operation does the user choose? Each of these instances will provide examples of local structure, and we can use this partial information to progressively refine our understanding of the structure of the entire document.

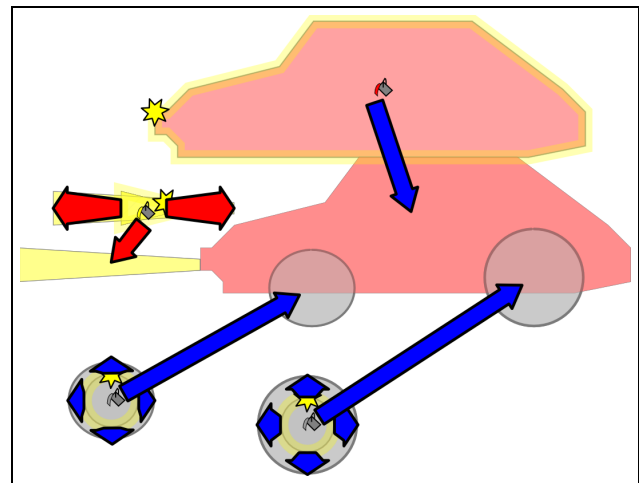


Figure 3: In this collaborative editing scenario, action depictions are color-coded by author. Depicted are object creation, translation, scaling, and color change.

Flexible Data Structures

Explicit spatial grouping by the user is a simple tree and does not allow an object to belong to more than one group. We will develop a tree-like structure that allows grouping across branches and direct access, to facilitate future manipulation of the structure. We will store what we have extracted up to a certain point in this tree. This will enable user queries such as “Show me the possible groups for this object.”

EXPLORING THE DESIGN SPACE

One advantage of digital over traditional media is its encouragement of users to think in terms of structure and variations of a design. A common strategy is to save “snapshots” of an illustration as separate documents. For example, an architect considering three options for a house entryway might make copies of the house blueprints for three teams to work on separately. These files are independent, and while tools such as Adobe Version Cue can help keep them organized, there is no direct way to merge them or use elements of one’s structure in another. We aim to make these processes possible.

Given the design history and the implicit structure of a document, we will build tools to help users explore alternative versions of a given document. Having access to combinatorial possibilities will aid users in prototyping documents. We believe it would be valuable to a user to see how a local change (an undo or other action) would propagate through the document structure. This would be particularly useful for collaboration, when users may want to independently pursue branches in the design. We would like to show a user which parts of the document can be edited independently.

We want to develop tools for computing the possible versions of a document, clustering similar versions, and helping the user navigate this space of possible designs. Challenges include determining which of these alternatives are meaningful (that is, defining similarity metrics for clustering) and how they should be presented. In the planning stage of this work, we look to prior work on variations [18, 19] and design galleries [13] for inspiration.

DISCUSSION

We believe that the techniques we have proposed will help users take fuller advantage of design on the computer by making the process more flexible and by helping them explore alternative versions of their designs. Our target audience includes anyone who has ever wished that their Illustrator or PowerPoint figures were easier to edit. To this end, we have built a software prototype that visualizes the history of an illustration. This work has applications in several common usage scenarios, including:

- **Prototyping.** Storyboards encourage exploratory design by enabling the user to easily survey previous actions in spatial context and selectively undo them. Design galleries would facilitate user exploration of variations.
- **Manipulation.** An object hierarchy is valuable for future editing of a document. We will try to suggest this structure to novice users who have not built it explicitly.
- **Collaboration.** Color-coding of depictions in the storyboard facilitates asynchronous collaborative editing. This “track changes” feature would benefit, for example, multiple authors working on the same figure for a paper.
- **Education.** An expert’s storyboards could help novices learn about an illustration by revealing the structure and editing process.
- **Redoing.** In addition to undoing an action, it would be useful to be able to modify its parameters or apply it to another object. Dealing with such modifications and propagating their effects on subsequent actions motivates future work on action macros and redo.

We are excited about this work and hope to demonstrate the storyboards, discuss applications, and get feedback on all three projects. We would be happy to discuss both low-level details and system-level design. We are particularly interested in different modalities of interaction (*e.g.*, restricting the scope of the storyboard with a zoom or “magic lens” [6]).

We are exploring ways to use interaction to infer an object hierarchy and methods to help users explore the space of possible designs. We believe that this work will result in useful, tangible artifacts for the UI community and could open up exciting avenues for future research. Currently, we are preparing for formal evaluation of the storyboards and plan a set of directed and undirected editing tasks for novice and expert users. We welcome feedback on the design of these user studies and on all components of this work.

ACKNOWLEDGMENTS

Sara Su is a Ph.D. candidate advised by Frédo Durand at MIT. The storyboards are joint work with Sylvain Paris and Frédo Durand. We thank Rob Miller for advice on designing user studies, Eugene Hsu for discussion about usage scenarios, and the MIT Graphics Group for feedback on this article.

REFERENCES

1. G. D. Abowd and A. J. Dix. Giving undo attention. *Interacting with Computers*, 4(3):317–342, 1992.
2. M. Agrawala, D. Phan, J. Heiser, J. Haymaker, J. Klingner, P. Hanrahan, and B. Tversky. Designing effective step-by-step assembly instructions. In *SIGGRAPH’03*.
3. M. Agrawala and C. Stolte. Rendering effective route maps: Improving usability through generalization. In *SIGGRAPH’01*.
4. P. Baudisch, D. Tan, M. Collomb, D. Robbins, K. Hinckley, M. Agrawala, S. Zhao, and G. Ramos. Phosphor: explaining transitions in the user interface using afterglow effects. In *UIST ’06*.
5. T. Berlage. A selective undo mechanism for graphical user interfaces based on command objects. *TOCHI*, 1(3), 1994.
6. E. A. Bier, M. C. Stone, K. Pier, W. Buxton, and T. D. DeRose. Toolglass and magic lenses: The see-through interface. In *SIGGRAPH’93*.
7. D. B. Goldman, B. Curless, D. Salesin, and S. M. Seitz. Schematic storyboarding for video visualization and editing. In *SIGGRAPH’06*.
8. J. Hart. *The Art of the Storyboard: Storyboarding for Film, TV, and Animation*. Focal Press, 1999.
9. J. Heiser, D. Phan, M. Agrawala, B. Tversky, and P. Hanrahan. Identification and validation of cognitive design principles for automated generation of assembly instructions. In *AVI’04*.
10. Y. Kawasaki and T. Igarashi. Regional undo for spreadsheets. In *UIST ’04*.
11. D. Kurlander and S. Feiner. A visual language for browsing, undoing, and redoing graphical interface commands. *Visual Languages and Visual Programming*, 1990.
12. R. Li and D. Li. A regional undo mechanism for text editing. In *ECSCW’03*.
13. J. Marks, B. Andalman, P. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Rummler, K. Ryall, J. Seims, and S. Shieber. Design galleries: a general approach to setting parameters for computer graphics and animation. In *SIGGRAPH’97*.
14. S. McCloud. *Understanding Comics: The Invisible Art*. Harper Perennial, 1993.
15. C. Meng, M. Yasue, A. Imamiya, and X. Mao. Visualizing histories for selective undo and redo. In *APCHI’98*.
16. R. C. Miller. *Lightweight Structure in Text*. PhD thesis, Carnegie Mellon University, May 2002.
17. B. A. Myers, R. G. McDaniel, R. C. Miller, A. S. Ferreny, A. Faulring, B. D. Kyle, A. Mickish, A. Klimovitski, and P. Doane. The Amulet environment: New models for effective user interface software development. *Software Engineering*, 23(6), 1997.
18. M. Terry and E. D. Mynatt. Side views: persistent, on-demand previews for open-ended tasks. In *UIST ’02*.
19. M. Terry, E. D. Mynatt, K. Nakakoji, and Y. Yamamoto. Variation in element and action: supporting simultaneous development of alternative solutions. In *CHI ’04*.
20. B. Tversky, J. Zacks, P. Lee, and J. Heiser. Lines, blobs, crosses, and arrows: Diagrammatic communication with schematic figures. In *Theory and Application of Diagrams*. Springer, 2000.