

Enhanced Visual Authoring Using Operation History

by

Sara Lee Su

B.S., University of Washington, 2002

S.M., Massachusetts Institute of Technology, 2005

Submitted to the

Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2009

© 2009 Massachusetts Institute of Technology. All rights reserved.

The author hereby grants to MIT permission to reproduce and to distribute publicly paper and electronic copies of this thesis document in whole or in part in any medium now known or hereafter created.

Author
Department of Electrical Engineering and Computer Science
May 22, 2009

Certified by
Frédo Durand
Associate Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by
Terry P. Orlando
Chairman, Department Committee on Graduate Students

Enhanced Visual Authoring Using Operation History

by

Sara Lee Su

Submitted to the Department of Electrical Engineering and Computer Science
on May 22, 2009, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computer Science

Abstract

Graphical editors have introduced great flexibility to the designer's workflow, providing powerful digital tools and enabling the creation of complex and compelling designs. This thesis presents methods for improving these interactions by leveraging operation history. Much instrumentation and activity logging in software has been for the purpose of debugging, that is, for the benefit of the programmer or analyst. Our work addresses the mining of operation history for the benefit of the end user. We present three main contributions in this area. First, we introduce *selection expansion*, a method for facilitating the reuse of complex multiple-item selections by identifying items that are likely to be edited together. We then discuss an extension of this work, *soft grouping*, which gives users more control than standard selection and more flexibility than standard grouping. Finally, we present an interactive visualization of operation history, *interactive storyboards*, which enables in-context browsing and manipulation of operation history. We demonstrate these approaches in the context of vector graphics editing and present the results of pilot studies using our software implementation. While this thesis focuses on the usage patterns of graphic designers, many of the strategies could be generalized to other domains.

Thesis Supervisor: Frédo Durand

Title: Associate Professor of Electrical Engineering and Computer Science

Reader: Professor Maneesh Agrawala, University of California, Berkeley

Reader: Professor Robert C. Miller, Massachusetts Institute of Technology

Reader: Doctor Sylvain Paris, Adobe Systems

Vita

Sara Su was born in January 1982 in Jackson, Mississippi, and completed her secondary education in Bellevue, Washington. In June 2002, she received the Bachelor of Science degree in Computer Engineering from the University of Washington, then moved to the Massachusetts Institute of Technology for graduate studies. In January 2005, she received the Master of Science degree in Electrical Engineering and Computer Science from M.I.T. On April 8, 2009, she successfully defended the work presented in this dissertation in a public seminar at the M.I.T. Computer Science and Artificial Intelligence Laboratory.

Acknowledgments

Thanks to my advisor, Frédo Durand, for guiding my work since my arrival at MIT, for pushing when needed, and for demonstrating how to be a good scientist.

Thanks to Rob Miller and Maneesh Agrawala for supporting my transition to user interface research and for their input as members of my committee. Thanks to Sylvain Paris for hosting me at Adobe and co-supervising much of this work.

Thanks to David Salesin for making an internship at Adobe possible, to Steve Johnson and Frederick Aliaga for graphic design on the storyboards, and to Craig Scull for assistance with the user study. I appreciate those who have provided feedback on the work in this thesis, including Mira Dontcheva, Steve Feiner, Patrick Baudisch, Beth Mynatt, and the UIST Doctoral Consortium. Thanks to the Open Clip Art Library for source drawings and to the Inkscape developers for providing an excellent graphics editor upon which to build my prototypes.

I acknowledge the NSF Graduate Research Fellowship and the MIT Presidential Fellowship Program for providing the financial freedom to pursue my work.

Thanks to Jovan Popović and Seth Teller for making the Computer Graphics Group a welcoming place, and to Victor Zue for academic guidance throughout my time at MIT. From coordinating user studies to printing theses, Bryt Bradley has provided invaluable help cutting through red tape. Thanks to my early mentors in computer science research: Richard Anderson, Ed Lazowska, David Notkin, and Linda Shapiro at the University of Washington; and Ying-Qing Xu and Harry Shum at Microsoft Research Asia. Thanks to Cynthia Skier and the Women's Technology Program for the opportunity to teach.

Thanks to my contemporaries in the Graphics Group for being collaborators, proofreaders, pilot users, and good friends: Addy, Alec, Bennett, Daniel, Emily, Eric, Eugene, Ilya, Jonathan, Kevin, Marco, Paul, Robert, Soonmin, Tilke, Vlad, Yeuhi, Zhunping. I have had as role models our lab's postdocs and an earlier generation of Ph.D. students: Barb, Bob, Hector, Jaakko, Jan, Jason, Jernej, Jingyi, John, Manish, Matthias, Peter, Rob, Sam, Sylvain, Tom, Wojciech. Thanks to the many friends who have made my time in Cambridge so enjoyable.

Finally, thanks to my parents Bernard and Patricia, my brother Jonathan, and my grandparents for a lifetime of support.

To Jy-Ching Su and Pao-Fong Li

Contents

1 Introduction	23
1.1 Reusing Selections and Groups for Efficiency	24
1.2 Visualizing History for Non-Linear Interaction	25
1.3 Dissertation Context	25
1.4 Dissertation Outline	27
2 Related Work	29
2.1 Operation History	29
2.2 Selections and Grouping	30
2.3 Inferring Relations	31
2.4 General Design Goals	31
3 History-Based Selection Expansion	33
3.1 Motivation	33
3.2 Related Work	35
3.2.1 Revealing Occluded Content	35
3.2.2 Adapting User Interfaces	36
3.2.3 Generalizing Selections	36
3.3 Method	37

3.3.1	Selection Expansion Strategy	37
3.3.2	Design Considerations and Implementation	41
3.4	Evaluation of QuickSelect Interface	42
3.4.1	Participants	42
3.4.2	Apparatus	43
3.5	Selection Reuse with Existing Histories	43
3.5.1	Conditions	43
3.5.2	Tasks	43
3.5.3	Design and Procedure	44
3.5.4	Hypotheses	45
3.5.5	Results	46
3.6	Selection Authoring and Reuse in Free Drawing	47
3.6.1	Procedure	47
3.6.2	Hypotheses	47
3.6.3	Results and Feedback	48
3.7	Discussion	50
3.7.1	Strengths	50
3.7.2	Limitations and Future Work	51
3.8	Summary of Selection Expansion	52
4	Soft Groups for Multiple Selection Authoring and Reuse	57
4.1	Introduction	58
4.2	Related Work	59
4.2.1	Selecting, Grouping, and Tagging	59
4.2.2	Relation Building from Operation History	60

4.3	Soft Group Creation and Review	61
4.3.1	Soft Group Creation	61
4.3.2	Soft Group Review	61
4.3.3	Implementation in a Graphical Editor	63
4.4	Exploratory Evaluation of Soft Groups	63
4.4.1	Procedure	64
4.4.2	Participants	64
4.4.3	Apparatus	65
4.5	Observations and Feedback from User Study	65
4.5.1	Strengths	65
4.5.2	Limitations and Possible Solutions	66
4.6	Discussion and Future Work	67
4.7	Summary of Soft Groups	69
5	Interactive Visual Histories	71
5.1	Motivation	71
5.2	Related Work	72
5.3	Overview of Approach	73
5.4	Design Goals	73
5.5	Storyboard Visualization	76
5.5.1	Action Depictions	76
5.5.2	History in Context	78
5.6	Application: Non-Sequential Undo	79
5.6.1	Dependencies	80
5.6.2	Algorithm	80

5.7	Exploratory Evaluation of History Storyboards	80
5.7.1	Participants	81
5.7.2	Apparatus	83
5.7.3	Design	83
5.7.4	Results	83
5.7.5	Strengths	84
5.7.6	Limitations	85
5.8	Extensions	85
5.8.1	Multi-Frame Storyboards	85
5.8.2	Editing in Multiple-User Environments	86
5.9	Discussion	86
5.9.1	Future Work	87
6	Conclusions	91
6.1	Expressibility of Selections and Soft Groups	91
6.2	Expressibility of Visual Histories	92
6.3	Directions for Future Work	93
A	Stimuli	95
	Bibliography	117

List of Figures

1-1	The techniques presented in this dissertation address the related problems of reusing user operations and reusing user selections.	26
3-1	QuickSelect usage in Inkscape. In this vector graphics application, the user selects an object, then uses our method to expand the selection to a larger set. The selection expansion is driven by the operation history, and editing operations can be performed on the expanded selections like on any manual selection. For clarity, selection boxes have been highlighted in red in this manuscript.	35
3-2	Selection expansion example: Visualization of the document's operation history. A '1' indicates that an object was modified in that operation. An empty cell indicates that the object was not affected by that operation.	38
3-3	Selection expansion example: Left: An excerpt of the history showing operations affecting the query (e). Right: Collapsing identical lines for efficiency.	39
3-4	Selection expansion example: Query (e) yields three candidate objects to add to the selection: d, f, and g.	39
3-5	Selection expansion example: Query (e, d) yields two candidate objects to add to the selection: f and g.	40
3-6	Selection expansion example: Query (e, d, g) yields only one candidate: f.	40

3-7 Selection expansion example: Query selection size vs. Maximum selection frequency.	41
3-8 The seats task: “Delete only the chairs at the center of the room.” We show the existing selections that can be reused for a particular object. Starting with one chair, the user can expand the selection to view previous selections. In this case, the given task could be completed with the set given after the second expansion. If the user expands too far, unwanted objects can be removed from the set by standard deselection.	44
3-9 Selection reuse with existing histories. When the user has understood the objective, he or she clicks “Start”. The objective is accomplished using the specified selection interface. When the task is complete, the user clicks “Stop”. Timing data is recorded by the system. While selection boxes have been highlighted in red in this manuscript, they had the conventional appearance in the software used in the study.	45
3-10 Performance comparison for selection reuse using existing histories. Shown are the median task completion times for two interface conditions. Participants completed the selection authoring tasks faster using the QuickSelect interface.	46
3-11 Tasks for which QuickSelect noticeably increased efficiency. (a) QuickSelect enabled users to re-select occluded balls. (b) Standard selection required more zooming to access differently-sized states. (c) The overlapping hair was easier to manipulate with QuickSelect. . .	54
3-12 Tasks for which QuickSelect did not noticeably affect performance. The composition of these drawings is simple enough that direct selection methods could be used quickly.	55

4-1	Soft Group creation in Inkscape. (a) The user first makes a multiple selection using an existing method, in this case, the marquee tool. (b) The creation command is called to mark this three-item selection as a soft group. (c) The three items now constitute a soft group. The members of a soft group can continue to be manipulated independently. For clarity, selections have been highlighted in red.	60
4-2	Soft Group review in Inkscape. In this example, we assume the user has previously created soft groups by several applications of the process shown in Figure 4-1. (a) The user selects a query item. (b) The user triggers two review queries to retrieve two soft groups containing the query item. (c) A returned soft group can be used like any regular selection. The members of a soft group can continue to be manipulated independently of one another.	61
4-3	Several study participants requested visual confirmation that a soft group had been created. We modified the prototype so that that bounding box of the selection flashes when it is made into a soft group. Participants who used the modified prototype did not have complaints about visibility.	67
5-1	Overview of storyboards. This automatically-generated visualization shows the editing history of an illustration. Arrows and icons depict spatial transforms and color change actions performed by the user. The user can click on any depiction to selectively undo the corresponding action.	74
5-2	Action depictions: Translation and rotation.	76
5-3	Action depiction: Resizing.	77
5-4	Action depiction: Change in fill style.	78
5-5	Action depiction: Deletion.	78
5-6	Per-object history. Once in storyboard mode, the user can restrict the history view (Figure 5-1) to a selected object. Here, only the history of one of the tail elevator pieces is shown.	79

5-7	Pseudo-code of the selective undo algorithm.	81
5-8	The storyboard visualization provides a natural interface for non-sequential interaction, specifically selective undo. Note that the selective undo of the earring movement (an action occurring farther back in the history stack) does not affect any other object. In contrast, standard (sequential) undo would have required also undoing all subsequent actions.	82
5-9	Participants in the user study were asked to recreate drawings typical for them. Over the course of the editing sessions, they were asked to make modifications to their drawings in an effort to encourage the use of the visualization and undo.	84
5-10	Multi-frame storyboard. This storyboard summarizes the editing history of a document, with time progressing from left to right. In addition, this storyboard shows an alternative depiction style we have considered.	86
5-11	Pseudo-code of the multi-frame storyboard creation process.	87
5-12	Collaborative editing scenario. Two users have edited this drawing, and the storyboard displays their actions color-coded in a “track changes” style.	88
5-13	A sampling of vector drawings made by participants in our study. All were amateurs who had prior experience with Microsoft PowerPoint’s drawing tools, and some were familiar with Adobe Illustrator. All were first-time users of our software, yet were able to create a wide range of drawings in a short period of time.	90
A-1	Task card: “Make all of the diamonds red.” To complete this task, ten objects (the diamonds) must be selected. Using the pre-existing history, this could be accomplished with a three-step selection expansion.	96
A-2	Task clock: “Change the color of the time.” To complete this task, five objects (the clock digits) must be selected. Using the pre-existing history, this could be accomplished with a three-step selection expansion.	97

A-3	Task filtration: “Move all explanatory text to the top of the diagram.” To complete this task, fourteen objects must be selected. Using the pre-existing history, this could be accomplished with a four-step selection expansion.	98
A-4	Task foot: “Bend the left leg at the knee.” To complete this task, thirty objects (the bones) must be selected. Using the pre-existing history, this could be accomplished with a four-step selection expansion.	99
A-5	Task giraffe: “Remove the giraffe’s spots.” To complete this task, fourteen objects must be selected. Using the pre-existing history, this could be accomplished with a three-step selection expansion.	100
A-6	Task girls: “Change the hair of both girls to brown.” To complete this task, eight objects must be selected. Using the pre-existing history, this could be accomplished with two two-step selection expansions.	101
A-7	Task jeep: “Delete the triangular pieces on the doors.” To complete this task, sixteen objects must be selected. (The brown filled regions and black outlines are separate drawing objects.) Using the pre-existing history, this could be accomplished with a five-step selection expansion.	102
A-8	Task map: “Color all states on the East coast green.” To complete this task, fifteen objects must be selected. Using the pre-existing history, this could be accomplished with a four-step selection expansion.	103
A-9	Task melon: “Make the black seeds white.” To complete this task, twelve objects must be selected. Using the pre-existing history, this could be accomplished with a two-step selection expansion.	104
A-10	Task mickey: “Make the centers of the eyes blue.” To complete this task, two objects must be selected. Using the pre-existing history, this could be accomplished with a two-step selection expansion.	105

A-11 Task mobile: “Delete the first row of buttons (the un-numbered ones).” To complete this task, four objects must be selected. Using the pre-existing history, this could be accomplished with a three-step selection expansion. 106

A-12 Task page: “Delete all text on the pages.” To complete this task, twenty objects must be selected. Using the pre-existing history, this could be accomplished with a three-step selection expansion. . . . 107

A-13 Task protein: “Delete the blue ovals, including their text.” To complete this task, twelve objects must be selected. Using the pre-existing history, this could be accomplished with two three-step selection expansions. 108

A-14 Task recycle: “Change the color of the entire PEBD (#4) symbol.” To complete this task, three objects must be selected. Using the pre-existing history, this could be accomplished with a two-step selection expansion. 109

A-15 Task ribs: “Color the white ribs blue.” To complete this task, fourteen objects must be selected. Using the pre-existing history, this could be accomplished with a three-step selection expansion. . . . 110

A-16 Task runway: “Change all of the yellow lines to black.” To complete this task, thirty-three objects must be selected. Using the pre-existing history, this could be accomplished with a three-step selection expansion. 111

A-17 Task seats: “Delete only the chairs at the center of the room.” To complete this task, twelve objects (the chairs and their shadows) must be selected. Using the pre-existing history, this could be accomplished with a four-step selection expansion. 112

A-18 Task sheep: “Change the color of the legs and feet.” To complete this task, eight objects must be selected. Using the pre-existing history, this could be accomplished with a four-step selection expansion. 113

A-19 Task truck: “Make the leftmost wheel green.” To complete this task, seven objects must be selected. Using the pre-existing history, this could be accomplished with a two-step selection expansion. . . 114

A-20 Task uranium: “Delete all of the red balls.” To complete this task, twenty objects must be selected. Using the pre-existing history, this could be accomplished with two two-step selection expansions, plus a single selection. 115

Chapter 1

Introduction

An important advantage of computer-based tools over their analog predecessors is the ease with which they store and can revisit state. With the ability to save copies of a document and to undo changes, software frees users to experiment with variations of a design with fewer consequences of making mistakes. Our work focuses on further extending the flexibility of these interactions by visualizing, manipulating, and reusing *operation history*.

There are many existing uses of history. Activity logging is built in to much of the software we regularly use. Often, instrumentation is for the purposes of error reporting, debugging, or optimization, that is, to support the developers or analysts of the software. For example, in the event of a system crash, a crash dump file is often created, and the user is prompted to allow it to be sent to the software company for analysis. While such logging data is valuable for the software engineers and support personnel, its benefit to the end user is often indirect, and in its raw form, it is can be difficult to make sense of. In this dissertation, we focus on how the history of a document can benefit the document's author in direct and visible ways. Instead of logged system events, we consider the editing operations the user has made directly on a document. This type of operation history drives the *undo* mechanism we are all familiar with. It is also useful for versioning systems and for tutorials recording what a user has done.

Among the most frequent operations performed by users are *selections* and *grouping*. Users can edit several items at once using *multiple selections* or group these items to build structure into their documents. These interactions are intu-

itive and widely used but do have limitations. Undo provides a way for the user to revisit the history of a document, but this is limited to sequential browsing. Selections are lightweight and are created as needed, but as they are not persistent, remaking multiple selections is tedious. Groups are persistent, but interactions with them are governed by strict rules. For example, an object can belong to only one group, hence changing the membership of a group can be tedious.

We address these limitations by showing that reusing operations, selections, and groups from a document's history can improve interaction for the end user. We will discuss strategies for increasing user efficiency in repetitive editing tasks and for supporting user browsing and manipulation of history. This work has also involved an important human component. In the iterative design process, we have been guided by principles of cognitive psychology (e.g. [Tversky et al. 2006]), as well as feedback from users of our prototype software.

1.1 Reusing Selections and Groups for Efficiency

One thread of our work is applying operation history to improve user efficiency. We have evaluated methods for visual reminders and shortcuts in the context of selections and grouping. It is common to apply a change to multiple items in a document at once. Many tools exist for selecting sets of items, but directly selecting large sets can be cumbersome and repetitive. We look to a document's operation history to reuse selections, observing that items edited together in the past are good candidates to be edited together again. We propose a method and interface for reusing complex user selections. When the user requests it, items that have previously been edited with the current selection can be added to it.

This approach does not require a semantic model of the document or relations between items. Instead, the results of each user query are generated by what he or she has done so far to create the document. Interviews with users indicate that this *selection expansion* approach, using their previous actions, is easy to learn and to apply. The study also suggested areas for improvement and motivated work on *soft grouping*, a technique that provides users with more control than standard selection and more flexibility than standard grouping. Overall, feedback from study participants was positive, and they provided a number

of examples of scenarios from their own work in which the history reuse techniques would apply.

1.2 Visualizing History for Non-Linear Interaction

A second thread of our work is developing better visualizations of operation history. Most editing software supports a notion of history; with the undo command users can revisit the recent history and reverse any unintended edits made. The user can discard recent mistakes and 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. While this exploration is intuitive, users are limited to sequential and causal exploration of the history.

We propose a new interaction metaphor and visualization for operation history that enables in-context browsing and manipulation of the user's previous editing actions. This design is inspired by the visual language of film storyboards and assembly instructions and provides an *interactive graphical history*. At any point in editing, the user can access a storyboard history visualization annotated with graphical metaphors of their editing actions. The storyboard summarizes the editing of the entire document or a selected object and enables the user to consider the operation history in spatial context. This metaphor provides instant access to any past action, and we demonstrate that this is an intuitive interface to a selective undo mechanism.

1.3 Dissertation Context

The techniques presented in this dissertation address the related problems of reusing user operations and reusing user selections. The three main contributions can be considered in the space described by Figure 1-1.

We demonstrate these history-based techniques in the context of vector graphics editing, a domain in which digital tools have introduced significant flexibility to the designer's workflow. In addition to providing a rich toolbox of graphical elements, presentation and graphics software such as Illustrator [Adobe Systems

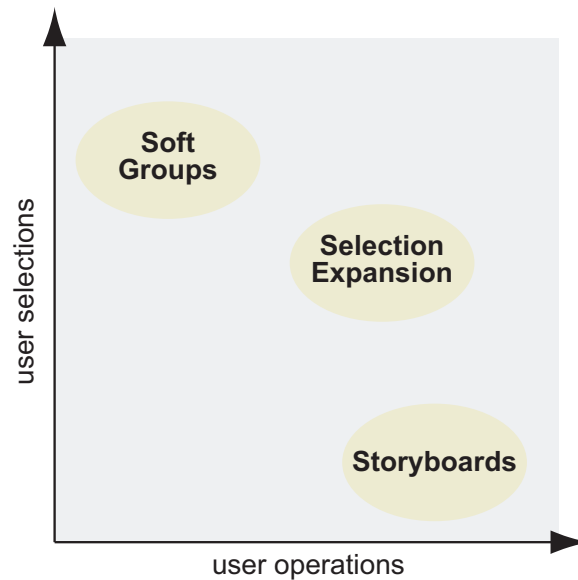


Figure 1-1: The techniques presented in this dissertation address the related problems of reusing user operations and reusing user selections.

1988–] and PowerPoint [Microsoft 1990–] facilitate exploration, trial-and-error editing, and refinement of designs.

We have implemented the techniques as extensions to Inkscape, a popular vector graphics editor [2003–]. Inkscape is an open source drawing program that uses the standard scalable vector graphics (SVG) format and provides many of the basic features of commercial packages such as Illustrator and PowerPoint. It is implemented in C++, and our prototype implementation reuses many of the Inkscape drawing primitives (as well as its home-grown renderer) to display the graphical elements of the new interface designs.

While this dissertation focuses on the usage patterns of designers and authors of graphical documents, the techniques we have developed could be translated to other domains. We believe that mining operation history can lead to enhanced human-computer interaction by revealing information about the document, the user, and the task. We hope that the history-based techniques presented in this dissertation demonstrate the utility of instrumentation for the end user and will stimulate further research in this area.

1.4 Dissertation Outline

Chapter 2 begins with an overview of related work. More detailed discussion is provided in the context of our contributions in Chapters 3, 4, and 5.

Chapters 3 and 4 address the problem of multiple selection. Chapter 3 introduces a technique for helping users reuse complex selections by expanding the set of currently selected items. It describes our simple but effective strategy for *selection expansion*, which takes a query set of items and generates a larger selection containing items most often edited with them (§3.3.1). The chapter discusses the design considerations of implementing this algorithm as a tool in Inkscape (§3.3.2). It then presents the results of a pilot study in which non-expert participants compared selection interfaces in the context of fixed drawing tasks (§3.5) and free drawing (§3.6). The stimuli used in the fixed drawing tasks are presented in Appendix A. Chapter 3 concludes with a discussion of the strengths and weaknesses of selection expansion in the context of the user study (§3.7).¹

Chapter 4 first discusses the limitations of manual grouping and manual selection (§4.1), then introduces a solution to help users bookmark and reuse selections. The chapter presents the iterative design of *soft grouping* (§4.3.2) and its implementation in Inkscape (§4.3.3). It then presents the results of an exploratory study in which participants tried soft grouping during unconstrained drawing (§4.4). The chapter concludes with observations on the feedback from the user study (§4.6), including comparison to selection expansion.

Chapter 5 focuses on *interactive visual histories*. It introduces our *interactive storyboard* interface, which serves as both a visualization and a new mode of interaction with a document's operation history. The chapter discusses the design goals driving the work (§5.4) the design of the visual elements (§5.5.1), and interaction (§5.5.2) of the storyboard. It describes how the storyboard interface enables non-sequential browsing and editing (§5.6). The chapter presents the results of an exploratory study using storyboards in a free drawing context (§5.7) and two extensions to the interface motivated by the user feedback (§5.8).

Finally, Chapter 6 summarizes our contributions and outlines future work and applications of reusing operation history.

¹The work described in this chapter was presented at Graphics Interface 2009 [Su et al. 2009].

Chapter 2

Related Work

Digital tools enable us to select and work with several components of a design at once. This dissertation focuses on three particular types of interaction: re-visiting history, selecting items, and grouping items. In this chapter, we survey these interactions, their limitations, and related work that seeks to address these limitations. Further discussion of related work is given in the context of the techniques proposed in Chapters 3, 4, and 5.

2.1 Operation History

Undo provides a way for the user to revisit the history of a document. While intuitive, this interaction is limited to sequential browsing. In addition, undoing one operation also cancels all operations after it. There has been work to add selective undo to graphical [Berlage 1994] and text editors [Li and Li 2003]. We build on this work and propose a graphical interface to make it easier to identify the operation to undo.

We are not the first to work in this area. In previous work on graphical histories, Meng et al. [1998] showed the changing state of a document. Their system provided a visualization of history at a summary level, but the “snapshots” did not show individual operations explicitly. Our work follows most closely from the Chimera system [Kurlander and Feiner 1992; Kurlander 1993], which shows

graphical history in a series of panels, each one showing the before and after states of one operation with a text description.

We have designed our history visualization to show explicit individual operations, to provide a summary with multiple operation depictions in a single view, and to emphasize visual depictions over text annotations. We propose this fully graphical storyboard as an alternative, complementary visualization. We look to the visual language of film storyboards [Hart 1999; Tumminello 2005] and assembly instructions [Agrawala et al. 2003] to design them.

We will show that our interactive storyboards are an interface to a more flexible, non-sequential undo mechanism. In CAD, programs such as SolidWorks [1995–] have shown the value of maintaining visual histories. The object tree in such systems provides an interface for selective undo and parametric changes [Cicirello and Regli 1999] but requires the user to structure the document. One of our goals in designing the interactive storyboard visualization was to offer some of the flexibility of these systems at a lower cost to the non-expert user.

2.2 Selections and Grouping

On the subject of structure, we consider the common operations of selection and grouping. Selections are lightweight and are created as needed, but they are not persistent. As discussed in the previous chapter, the ability to select and repeatedly edit the same set of objects together is fundamental in editing. Reselecting this set can be labor-intensive, particularly when objects overlap. Groups are intuitive but rigid; an item cannot belong to more than one group at a time.

In the following chapters, we address these limitations by introducing methods for reusing user selections and groups. Our work on selection expansion draws from several areas of related research: strategies for providing access to occluded content, methods for adapting interfaces to users' past actions, and techniques for generalizing users' selections.

It is a well-known problem that selection is difficult when objects overlap. There have been many techniques proposed to reveal hidden content using transparency filters [Baudisch and Gutwin 2004; Ishak and Feiner 2004], spatial transforms [Ramos et al. 2006], and physical interaction metaphors [Beaudouin-Lafon

2001]. There has also been work on facilitating selection of user interface components by adapting them to the user's past actions. In this space, techniques have been proposed to resize or rearrange menus to reduce target acquisition time [Bederson 2000] and to dynamically organize menu items based on usage patterns [Greenberg and Witten 1985; Mitchell and Shneiderman 1989]. Similar in spirit, we look to operation history to reduce the time to reselect items.

2.3 Inferring Relations

There has been recent work on building relations between digital objects from operation history. Notably, Pedersen and McDonald [2008] have proposed a metric of document usage time to infer relations between documents. In the domain of social computing, there has been work to determine the strength of links between individuals from frequency of activity in online social networks.

Focusing on selections, there has been significant recent work on inference of multiple selections from user examples [Miller and Myers 2002; Ritter and Basu 2009] and selection authoring by declarative query [Heer et al. 2008a]. In contrast to these approaches, our strategy does not take a programming by demonstration approach. Instead, suggestions for sets of objects come directly from previous usage.

2.4 General Design Goals

In addition to related work in human-computer interaction, our design strategy is informed by the cognitive psychology literature. It has been shown that in-place visualization of user interface transitions and variations [Baudisch et al. 2006; Terry et al. 2004] improves comprehension by exploiting spatial memory. This has led us to design our visualizations of user operations and selections to be in place, providing users with spatial context. In addition, we have designed graphical elements of the visualizations to have a consistent look and feel to distinguish them from user-created content. We rely on well-accepted schematic representations, such as arrows, to aid comprehension [Tversky et al. 2000].

Chapter 3

History-Based Selection Expansion

When editing a graphical document, it is common to apply a change to multiple items at once, and a variety of tools exist for selecting sets of items. However, directly selecting large sets can sometimes be cumbersome and repetitive. We propose a method for helping users reuse complex selections by expanding the set of currently selected items. We analyze a document's operation history to determine which items have been frequently edited together. When the user requests it, items that have been previously edited with the current selection can be added to it. The new selection can then be manipulated like any other selection. The *selection expansion* approach we present in this chapter does not require a semantic model of the document or relations between items. Rather, each expansion is based on what the user has done so far to create the document.

We demonstrate this approach in the context of vector graphics editing. Results from a pilot study were encouraging. Reusing selections with pre-existing histories, users were more efficient at editing tasks with our *QuickSelect* tool. Subjective preferences from a usability study in a free drawing context indicate that selection expansion is easy for users to learn and to apply.

3.1 Motivation

Graphical editing software enables users to create complex and extensible documents. Among the facilities editors offer is applying a change to several items at

once. Most support *multiple selections*, that is, selections of more than one item, by several methods. These include selection by region (lasso or marquee tool) or shared attribute, and by clicking on items to add them to the current selection. Selections are flexible but ephemeral. It is common to want to edit the same set of objects together over the course of an authoring session, and recreating the members of this set can become laborious and repetitive.

One solution is to group items. However, this requires foresight, and groups built for a given edit might not be adaptable for later editing tasks. Groups are reversible, but repeated grouping and ungrouping can become tedious, particularly for detailed designs.

In this chapter, we describe a lightweight method for simplifying the reuse of complex selections. The core idea of our approach is to take the document operation history into account, analyzing what the user has done so far and which objects have been edited together in the past. We observe that objects that have been selected together in the past are good candidates to be manipulated together in the future.

We propose a method for *selection expansion*: From one or more selected items, our method generates a larger selection containing items most often edited with this input item set. We describe a simple yet effective strategy for determining the members and size of each selection expansion based on past edits (§3.3.1). We demonstrate our approach in the context of 2D graphical editing (§3.3.2). Our *QuickSelect* tool can be activated at any point in editing an illustration. When the user clicks on any selection, our method expands it to a larger one. The new item set is a normal selection and can be manipulated as such, including being expanded again.

To evaluate the usability of selection expansion, we have conducted a pilot study with non-expert users. Feedback was generally positive, with participants finding the *QuickSelect* tool in Inkscape useful and easy to learn. When editing drawings with existing operation histories, users were faster at completing fixed tasks using *QuickSelect* (§3.5).

If history is to be valuable for selection, people must repeatedly select the same sets of items. After the fixed tasks, we observed users creating drawings from scratch. The majority said that selection reuse was common enough in their work that they could see the benefit of incorporating *QuickSelect* (§3.6).

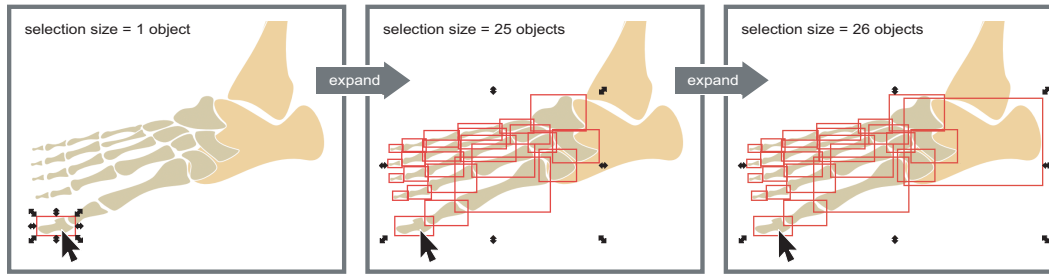


Figure 3-1: QuickSelect usage in Inkscape. In this vector graphics application, the user selects an object, then uses our method to expand the selection to a larger set. The selection expansion is driven by the operation history, and editing operations can be performed on the expanded selections like on any manual selection. For clarity, selection boxes have been highlighted in red in this manuscript.

We will discuss the strengths and limitations of selection expansion, as well as the potential for broader use of history to improve interaction (§3.7).

3.2 Related Work

Our work draws from three areas of related research: strategies for providing access to occluded content, methods for adapting interfaces to users' past actions, and techniques for generalizing users' selections.

3.2.1 Revealing Occluded Content

Many techniques have been introduced to improve selection interfaces and to address the problem of obscured content in graphical interfaces. Magic Lens filters [Bier et al. 1993] and multiblending [Baudisch and Gutwin 2004] reveal hidden content using transparency filters. While these designs focus on displaying hidden content, others have addressed the problem of selecting it.

For window management, Beaudouin-Lafon [2001] proposes interaction techniques that extend a paper metaphor to “peel back” windows. The Exposé feature of the Mac OS X operating system [Apple 2003–] resizes and translates windows to provide access to occluded objects. Content-aware free-space trans-

parency [Ishak and Feiner 2004] uses a physical interaction metaphor and provides a set of tools for users to probe portions of the screen for hidden content.

Addressing 2D drawing, the Tumbler and Splatter tools [Ramos et al. 2006] help users access occluded content by, respectively, cycling through layers of a document and “splating” objects into an exploded view. Schwarz et al. [2007] limit the need for multiple item selections by applying properties to the canvas background. Our work is complementary to these approaches as it aims to reduce the overhead of selections by reusing them.

3.2.2 Adapting User Interfaces

We aim to improve performance by adding interface shortcuts based on usage patterns. There is a long history of work in adapting interfaces to the user’s past actions. Improvements to menus have been proposed to resize or rearrange items with the goal of reducing time to target acquisition (e.g. fisheye menus [Bederson 2000] and Flexcel [Thomas and Krogsæter 1993]). Greenberg and Witten [1985] and Mitchell and Shneiderman [1989] have investigated the effects of dynamically organizing menu items based on frequency and recency of use. Sears and Shneiderman [1994] have shown that split menus, with high-frequency items at the top, significantly improve selection times. These studies have demonstrated the potential performance advantages of frequency-ordered content and suggest that this criterion could be applied to other types of selections. Findlater and McGrenere [2004] have found that dynamically rearranging menus can negatively affect performance, possibly disrupting spatial memory. This informs our strategy of avoiding rearranging the drawing objects themselves in favor of a lightweight use of the existing selection facilities.

3.2.3 Generalizing Selections

In their LAPIS text-editing system, Miller and Myers [2002] demonstrated “selection guessing”, a method of inferring a multiple selection user-provided positive and negative examples. Ritter and Basu [2009] recently continued in this theme with their selection classifier but focused on selections of discrete items (files) rather than arbitrary text. Our work differs from these methods as it does not use

inference and programming by demonstration, but rather offers suggested selections directly from the set of selections previously used. In the future, it would be interesting to explore combinations of these techniques in different types of editors and file managers.

Heer et al. [2008a] recently introduced a set of techniques for selection authoring by declarative query. Their interface supports selection by attribute and enables generalization of such selections by interactive query relaxation. They have shown that this approach enables users to make and reuse more accurate, “nuanced” selections, particularly in information visualization applications. Our approach differs in that the process for generating selection expansions does not use a semantic model of the underlying data. Rather, selection sets come directly from the document history.

In its lack of strict hierarchical structure, our work is related to Saund et al.’s ScanScribe editing platform, which supports flexible, overlapping groups [2003]. However, while ScanScribe seeks to find perceptually significant structure in documents, our equivalent of structure is determined by user selection patterns.

3.3 Method

Our approach is a combination of interaction history and direct manipulation. We look to a document’s editing history to reuse selections, observing that items edited together in the past are good candidates to be edited together again. We propose a selection expansion strategy: The user selects one or more items and invokes the expansion function. From this query, we generate a larger selection by adding items that were often previously edited with the query set. We now present the general algorithm and its implementation in Inkscape.

3.3.1 Selection Expansion Strategy

Our algorithm starts with a query selection, a set of one or more items. It then looks in the operation history for the best single item to add to the set. Candidates include any item that has been edited together with the query set. Of

these, we pick the item that appeared with the set most frequently and expand the selection by one.

To reduce tedium of repeated queries, we combine sequential expansion steps when possible: one query can grow the selection by several items. Our strategy is to consider the maximum selection frequency (given by the operation history) at each step of the expansion. If the frequency plateaus over two or more steps, we combine these into a single larger step.

To illustrate the algorithm, we consider an example expansion. We analyze the document's operation history, the beginning of which is visualized below (Figure 3-2). Columns are objects in the document and rows are user editing operations. In this visualization, a '1' indicates that an object was modified in that operation/row. An empty cell indicates that the object was not affected by that operation. For instance, operation 1 modified objects d and e.

		objects									
		a	b	c	d	e	f	g	h	i	j
operations	0				1	1					
	1				1	1					
	2	1						1			
	3		1	1			1	1	1	1	1
	4		1								
	⋮										

Figure 3-2: Selection expansion example: Visualization of the document's operation history. A '1' indicates that an object was modified in that operation. An empty cell indicates that the object was not affected by that operation.

Suppose the user makes the query (e). Below is an excerpt containing only operations affecting (e) (Figure 3-3). For efficiency, we compress the matrix, combining identical lines and encoding the frequency of selection sets, e.g. lines 0, 1, and 24 are collapsed into line 0.

		objects									
		a	b	c	d	e	f	g	h	i	j
operations	0				1	1					
	1				1	1					
	10				1	1	1	1			
	11					1			1		
	20				1	1	1	1			
	24				1	1					
	30					1				1	
	32					1					

→

		objects									
		a	b	c	d	e	f	g	h	i	j
operations	0				3	3					
	10				2	2	2	2			
	11					2			2		
	32					1					

Figure 3-3: Selection expansion example: Left: An excerpt of the history showing operations affecting the query (e). Right: Collapsing identical lines for efficiency.

With the query (e), there are three candidates to add to the set: d, f, and g (Figure 3-4). The set (e, d) occurred in line 0 (3 times) and in line 10 (2 times), for a total frequency of 5. The set (e, f) occurred in line 10 (2 times). The set (e, g) occurred in line 10 (2 times) and line 11 (2 times), for a total frequency of 4.

		objects									
		a	b	c	d	e	f	g	h	i	j
operations	0				3	3					
	10				2	2	2	2			
	11					2			2		
	32					1					

OR

		objects									
		a	b	c	d	e	f	g	h	i	j
operations	0				3	3					
	10				2	2	2	2			
	11					2			2		
	32					1					

OR

		objects									
		a	b	c	d	e	f	g	h	i	j
operations	0				3	3					
	10				2	2	2	2			
	11					2			2		
	32					1					

Figure 3-4: Selection expansion example: Query (e) yields three candidate objects to add to the selection: d, f, and g.

Because its set had the highest frequency, we add object d. To continue the expansion, the new query is (e, d). The candidates to add are f and g (Figure 3-5). In both cases, the frequency is 2.

		objects									
		a	b	c	d	e	f	g	h	i	j
operations	0	3		3							
	10			2	2	2	2				
	11			2	2						
	32			1							

or

		objects									
		a	b	c	d	e	f	g	h	i	j
operations	0	3		3							
	10			2	2	2	2				
	11			2	2						
	32			1							

Figure 3-5: Selection expansion example: Query (e, d) yields two candidate objects to add to the selection: f and g.

Frequencies being equal, we randomly choose to add object g. To continue the expansion, the new query is (e, d, g). The only candidate to add is object f (Figure 3-6), so we have (e, d, g, f).

		objects									
		a	b	c	d	e	f	g	h	i	j
operations	0	3		3							
	10			2	2	2	2				
	11			2	2						
	32			1							

Figure 3-6: Selection expansion example: Query (e, d, g) yields only one candidate: f.

We now see if any of these intermediate steps (e) → (e, d) → (e, d, g) → (e, d, g, f) can be combined into larger expansion steps. Consider the plot of the query selection size versus the maximum selection frequency (Figure 3-7). A frequency plateau occurs between steps 3 (adding g) and 4 (adding f). Because it only appeared as much as the next larger set, the set at step 3 is likely to be an intermediate one with the larger set being the one of more interest to the user. To reduce the number of queries necessary to reach it, we combine steps 3 and 4. The query (e, d) will return the new selection (e, d, g, f).

We observed frequency plateaus in drawing histories we analyzed, suggesting that this strategy could be broadly useful. For example, in Figure 3-1, rather than 25 expansion steps from selecting the big toe to selecting all bones in the foot, our algorithm requires only two queries from the user.

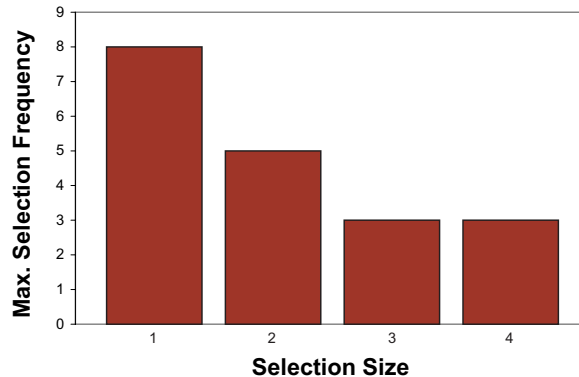


Figure 3-7: Selection expansion example: Query selection size vs. Maximum selection frequency.

3.3.2 Design Considerations and Implementation

We demonstrate our approach in the context of 2D graphical editing. We have implemented selection expansion as a feature in the Inkscape vector graphics editor [2003–]. In the interaction design process, we considered several interfaces.

A primary design goal was to limit disruption of the user’s workflow, so we found an initial design using a menu button to step through selection expansions cumbersome. Based on feedback from pilot users, we determined that the interface easiest to learn was a keyboard shortcut. At any point in editing an illustration, the user can select one or more objects using existing selection methods, then use our feature to expand the selection. Each subsequent key press triggers a new expansion query that could grow the selection by one or more objects. The items selected at any step can be manipulated like any other selection.

While pilot users found the keyboard shortcut easy to learn and use, some suggested a more active use of the mouse pointer similar to standard “shift+click” selection. These comments motivated the design of an alternative mouse-based selection interface that could provide more of a reminder of the initial object of interest.

Our initial design for the mouse interface followed the overloaded click model of text editors such as Microsoft Word, where the user clicks once to place the cursor, twice rapidly to select a word, and three times rapidly to select a para-

graph. This translates to our application by first entering a selection mode, then using repeated clicks to cycle through selection sets. Pilot users found the click interface intuitive but disliked having to enter a distinct mode to activate the selection behavior. On the other hand, making this the default mouse behavior could disrupt existing selection methods. Ultimately, we eliminated the mode requirement and simplified the mouse interface so that at any point in editing the user can click on an object, then scroll the mouse wheel to expand it.

3.4 Evaluation of QuickSelect Interface

To evaluate the new selection feature, we conducted a user study in which we asked subjects to perform specified selection and editing tasks, as well as unstructured exploratory drawing and editing.

The study was designed to fit within a 1-1.5 hour session, and participants had individual sessions. Participants were first asked about their background, specifically the types of drawings they make and the tools they use. Before the computer-based portion of the study, subjects completed an interactive tutorial to familiarize them with Inkscape and with the new selection feature. As the primary evaluation objective of the study was usefulness of the feature, we helped subjects through any usability problems during the tutorial. We also simplified the Inkscape toolbar and menus to provide basic drawing functionality while limiting potentially distracting features.

The study was composed of two phases. The first compared performance (speed and accuracy of selection authoring) using QuickSelect and existing methods on a set of fixed editing tasks (§3.5). The objective of the second phase was to observe the use of these selection methods in a more natural environment, by asking subjects to recreate a “typical” drawing, identified during the background interview (§3.6). The session concluded with a debriefing session.

3.4.1 Participants

Eleven subjects (6 male, 5 female) between the ages of 22 and 55 ($M = 33.4$, $SD = 11.5$), participated in the study. They were recruited from the general population

through research participation mailing lists and received gratuities for their time. All had normal or corrected-to-normal vision, and normal color vision. While none had previously used Inkscape, all were familiar with at least one 2D vector drawing program such as PowerPoint or Illustrator.

3.4.2 Apparatus

The study was conducted in a conference room in our laboratory. Participants used a 2.33 GHz laptop running Windows XP with a 15" display at 1440 × 900 pixel resolution. They were asked to use the laptop keyboard and a peripheral optical mouse. Screen capture software was used to record complete user sessions [Adobe Systems 2006–; RenderSoft 2001–].

3.5 Selection Reuse with Existing Histories

The objective of this phase of the study was to compare user performance on selection authoring tasks in a controlled lab environment.

3.5.1 Conditions

The two selection interfaces being compared were QuickSelect and “standard selection”, that is, the existing “shift+click” and rectangular marquee tools. For consistency, in this phase of the study we only used the keyboard shortcut interface to QuickSelect. (Pilot users reported that the keyboard interface was somewhat easier to learn than the mouse interface.)

3.5.2 Tasks

Each subject completed a series of twenty selection authoring tasks. Each task presented a drawing that had been previously edited by another author and had some stored operation history. Source documents came from the Open Clip Art Library [2004–] and were edited by the authors. Examples are shown in Figures 3-9, 3-11, and 3-12.

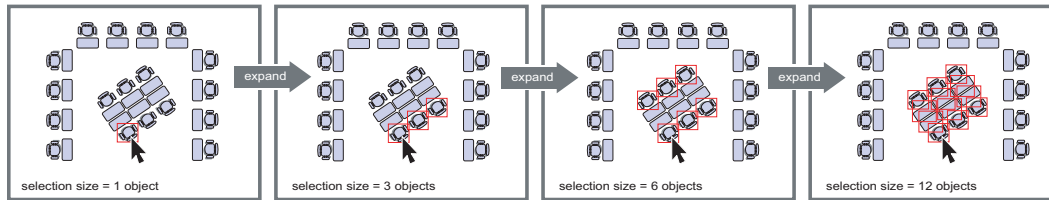


Figure 3-8: The seats task: “Delete only the chairs at the center of the room.” We show the existing selections that can be reused for a particular object. Starting with one chair, the user can expand the selection to view previous selections. In this case, the given task could be completed with the set given after the second expansion. If the user expands too far, unwanted objects can be removed from the set by standard deselection.

The histories attached to the trial drawings were recorded as the authors edited the files. These edits included simplification, rearrangement of objects, and attribute changes, and the task given to the subject was chosen with the history in mind. The task could always be completed using either standard selection or QuickSelect; with QuickSelect, more than one expansion step was always necessary to reach the target set. One example is shown in Figure 3-8

3.5.3 Design and Procedure

In each trial, the subject was asked to perform some editing operation (deletion, color change, or a spatial transform) on multiple objects in the drawing. The task objective was given in text and included the selection interface the subject should use. For the control condition, the type of standard selection used was recorded by the observer (e.g. “shift+click” or marquee or a combination). This was surprisingly consistent across users for each drawing.

The study design was within subjects. Each participant edited drawings with both conditions. Task order was randomized within each session to limit the learning effect. Subjects were asked to complete the tasks as quickly and as accurately as possible. Each trial proceeded in the following steps, also illustrated in Figure 3-9:

1. The document was loaded in Inkscape. The window contained a composition and a task objective, text at the top of the screen, describing the edits to make to the composition.

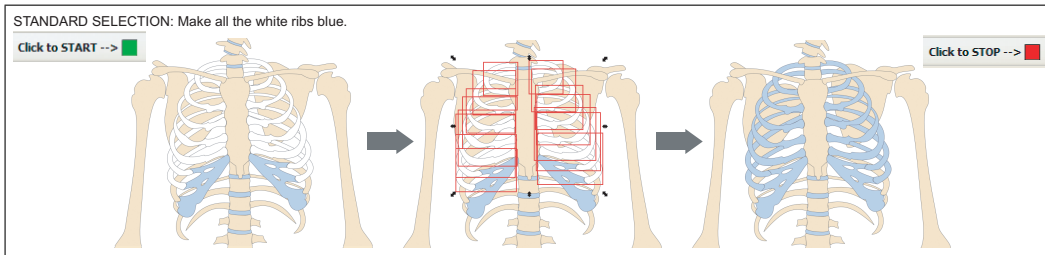


Figure 3-9: Selection reuse with existing histories. When the user has understood the objective, he or she clicks “Start”. The objective is accomplished using the specified selection interface. When the task is complete, the user clicks “Stop”. Timing data is recorded by the system. While selection boxes have been highlighted in red in this manuscript, they had the conventional appearance in the software used in the study.

2. When the user understood the task objective, after seeking any necessary clarification, he clicked the “Start” button.
3. The user made the prescribed edits to the composition.
4. To signal completion of editing, the user clicked “Stop”.
5. If the edits were correct, as determined by the observer, the task was complete.

The Inkscape software was instrumented to record timing data. Trial Time was counted from the “Start” click to the successful “Stop” click. In the trials where the automatic time stamp failed, timing data was retrieved from the session screen recording.

The observer recorded editing errors and mistrials. We define an editing error as an edit made on a selection that is not a subset of the target selection given in the task objective. We define a mistrial as an instance when the subject has clicked “Stop” before he or she has finished making all of the changes given in the objective. Mistrials were not included in the result timings.

3.5.4 Hypotheses

We hypothesized that the QuickSelect condition would result in shorter task completion times and fewer editing errors.

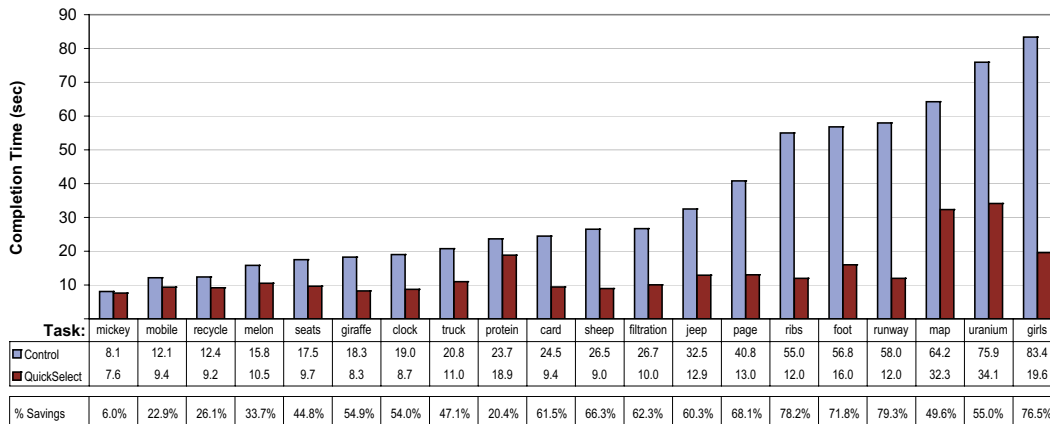


Figure 3-10: Performance comparison for selection reuse using existing histories. Shown are the median task completion times for two interface conditions. Participants completed the selection authoring tasks faster using the QuickSelect interface.

3.5.5 Results

We analyzed the performance data at a summary level by taking the median of the completion times over the 20 trials for each condition. We conducted an analysis of variance (ANOVA) on the dependent variable of Trial Time. We observed a significant main effect of selection interface ($F_{1,38} = 15.239, p < .0004$). Subjects completed the tasks faster with the QuickSelect interface than with the control. There was a notable difference in speed for the majority of the 20 tasks, as can be seen in Figure 3-10.

There are trade-offs to using pre-existing histories for these trials. The fixed tasks enabled direct comparison and guaranteed that subjects would try both standard selection and QuickSelect. While the timing results of these trials should not be considered conclusive evidence for validity of the technique, they suggest trends in performance to be explored further in unconstrained editing scenarios where the full history has been created by the user.

Our analysis revealed no main effect for number of editing errors. However, interestingly, several subjects *perceived* an improvement in their selection accuracy using QuickSelect. We discuss the implications of this in the next section.

3.6 Selection Authoring and Reuse in Free Drawing

The objective of the second phase of the study was to observe usage patterns during larger scale editing in an unconstrained context and to record subjects' subjective preferences for selection method.

3.6.1 Procedure

The subject was asked to recreate a drawing he or she described during the background interview. Based on time estimates from the interview, the experimenter selected a target drawing that could be completed in 15-20 minutes. For this part of the study, the mouse wheel interface to QuickSelect was introduced, and participants were asked to use whichever method they felt was more comfortable.

Subjects were told that the goals of this study were to observe their editing process and to give them enough practice with the selection tools so that they could comment on them at the end. There was no measure of success. The experimenter prompted the subject to use the selection tools as needed, usually by suggesting an edit to the document, but did not specify which method of selection to use.

Although we did not compare selection accuracy or time quantitatively for these trials, participants were interviewed after the computer-based portion and asked to rate the selection expansion feature. The purpose of this study was to record users' subjective impressions of the new selection feature after having used it for creation from scratch and for editing pre-existing documents (§3.5).

3.6.2 Hypotheses

We hypothesized that users would frequently use the new selections during unstructured drawing and would rate the feature highly for ease of acquisition, use, discoverability, applicability, and perceived performance. We hypothesized that users, when asked to compare QuickSelect with the control selection interface, would prefer to have both as options to use based on context.

3.6.3 Results and Feedback

Participants created a variety of designs including web-style graphics, diagrams, and poster layouts. While we did not count instances, most tried QuickSelect and standard selection without prompting. In some cases, the experimenter prompted a multiple selection by suggesting an edit, usually a return to a previous state of the drawing. In general, subjects were eager to try the new selection tool on their own drawings and did not require much prompting. User feedback on subjective measures supported our hypotheses.

Learnability. 10 out of 11 participants rated QuickSelect as easy to learn (*“It feels similar enough to [Microsoft] Office tools.”*), while one said *“It was easy to learn with some coaching”*.

Usability. 11 out of 11 participants rated QuickSelect as usable. All rated the keyboard shortcut as usable. While one subject expressed a preference of the keyboard shortcut over the mouse wheel because *“it is more controlled”*, the rest expressed slight preference for the mouse wheel because *“it feels faster”*.

Performance, speed. The subjects’ comments reflected their performance editing existing documents. All participants felt that QuickSelect improved their efficiency in both parts of the study.

Performance, accuracy. Although there was no measured effect, 7 out of 11 participants reported that QuickSelect improved the accuracy of their selections, both in the pre-made examples and on their own drawings. This is promising in that the feature may improve the confidence of novice users in experimentation in the software. One subject predicted that the tool would not necessarily increase his accuracy, because it improved his speed so much.

Discoverability. 7 out of 11 participants rated QuickSelect as discoverable. The remaining four found the behavior of the tool difficult to predict when editing other people’s drawings. All agreed that this was not a problem when they edited their own drawings.

Applicability. There was general consensus among participants that they could see themselves using QuickSelect if it were integrated in existing software. Several gave freeform feedback about situations where selection expansion would be most useful:

- *“It’s good for small objects and connected ones. Less fatigue.”*
- *“It’s good for reselecting. It would be easy to incorporate.”*
- *“For most drawings, I would probably stick to tried-and-true methods, but I would use this in cases where I had just made changes... It seems really useful for reaching embedded things.”*
- *“I maintain a lot of PowerPoint presentations. I’m motivated by expediency over perfection, so I think this will help me make changes quickly.”*
- *“I would use this more for touching up my own drawings than when creating them the first time.”*
- *“It seems best for things that are naturally grouped. I would definitely use this in [Adobe] Flash.”*

While we did not have a quantitative measure for this part of the study, we cross-referenced the experimenter’s notes with recorded history log files to determine how common multiple selections were in the participants’ workflows. A typical case is subject Y. Subject Y recreated a drawing of a lymph node from her work presentations. Over the course of this editing session, she performed 118 editing operations. Of these, 14 operations were applied to multiple selections. Of the multiple selections, a 2-item selection was repeated 7 times, a 4-item selection was repeated 2 times, and 5 selections were one-offs. This is typical across the participants in our study in that a few multiple selection would be reused over the course of the editing session. Future work is to analyze the statistics of longer editing sessions to determine applicability of multiple selection reuse for extended interaction.

3.7 Discussion

Results from the first part of the study were promising. While the use of existing histories might not be fully representative of real usage, the results suggest that QuickSelect can improve selection efficiency. User feedback from the free drawing sessions was consistently positive. We now discuss the strengths and weaknesses of selection expansion and consider directions for future work.

3.7.1 Strengths

Observing the user editing sessions, we identified common scenarios in which selection expansion offers high utility. We first consider the results for the 20 trial drawings with pre-existing histories. The timing results, summarized in Figure 3-10, suggest that the performance savings are larger for more complex drawings, if we assume that the completion time using the control condition is a measure of complexity. Figure 3-11 shows the three tasks that took the most time using the control condition.

Re-selecting occluded content

The uranium and girl tasks contain many overlapping objects. In this common scenario, the difficulty of selecting occluded objects can result in missed selections and subsequently longer editing times. While it does not eliminate the need for the initial selection, QuickSelect reduces the cost of re-selection in occlusion cases. User comments indicated consensus about the utility of QuickSelect for “embedded objects”.

Re-selecting objects of varying size

The map task (Figure 3-11) shows another common case where direct selection methods may be difficult. The target objects vary in size, and we observed subjects change the zoom level several times to complete the selections. QuickSelect mostly eliminated zooming during the selection process, but because of the object size difference, many subjects zoomed in to verify the edits. These examples

suggest that, while initial selection and verification costs are difficult to reduce, when these bottleneck costs are high, QuickSelect's speed-up for re-selections are more apparent.

Although several participants expressed concern after the first part of the study that they could not confidently predict how the selections would expand, the more natural "real world" tasks of the second part helped them realize applications of selection expansion in their own usage patterns. In addition to the measured improvement in speed in the fixed tasks, subjects expressed positive perceptions of their speed and accuracy with the QuickSelect tool.

3.7.2 Limitations and Future Work

Cases in which selection expansion did not offer a noticeable advantage include those in which there was little overlap of objects. In these cases, a rectangular marquee or individual selection could often be used quickly and accurately. Three such examples from the user study are shown in Figure 3-12. In addition, some compositions are simple enough that there is limited benefit of multiple object selections.

Study participants said they would use QuickSelect if it were integrated in the drawing software they normally use. Several subjects commented on feature extensions they would like to see, which we discuss now.

Predictability and error handling

To recover from an inadvertent addition to the selection set, the user can use the standard "shift+click" to remove unwanted items. An extension is to enable contraction as well as expansion of selections. To address users' concerns about predictability, a possible strategy is to provide visual cues for upcoming steps (information scent [Pirulli and Card 1999]). This would be useful in the case of editing another user's document, as well as when editing one's own document over an extended period of time.

In our exploratory study, we evaluated the accuracy of subjects using our selection feature. Future work is to evaluate the accuracy of the feature itself. Longer-term observation could reveal how well the selection expansions and

presentation order match the selections the user wants to make and, when they do not match, how much overhead is required to switch to another tool.

Combining selection tools

Several subjects thought it would be interesting to try selection expansion with tools such as select-by-attribute (e.g. Illustrator’s “Select Similar” [Adobe Systems 2005]). In the Inkscape implementation, we designed QuickSelect to work as a non-conflicting alternative alongside existing selection methods. Evaluating the impact of combinations of tools on the user experience is future work. In addition, it would be interesting to consider how manual grouping and selection expansion could be used together to resolve cases when the operation history shows an object to belong to several not necessarily nested sets. In Chapter 4, we introduce *soft groups*, which build on selection expansion but provide more user control and predictability.

Additional expansion heuristics

Another interesting direction for future research is incorporating additional user feedback into the selection expansion algorithm. The algorithm described in this chapter updates its suggested expansions as the user continues editing the document and using the expansion tool. Potential future work is to add user input and other heuristics to the algorithm (e.g. recency of use) and to apply more sophisticated learning.

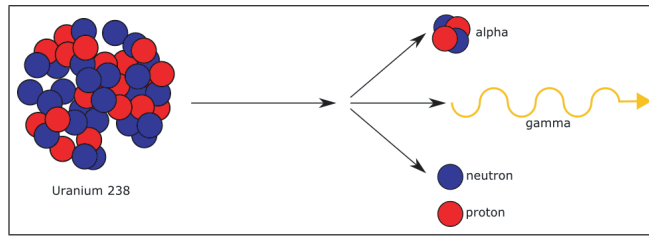
3.8 Summary of Selection Expansion

We have presented the selection expansion method, which combines direct manipulation with interaction history to improve selection efficiency. Results from the pilot study are encouraging and revealed common scenarios in which a selection expansion tool can improve both actual and perceived performance in selection authoring and manipulation tasks in vector graphics editing. In particular, it can reduce the time spent on repetitive, complex re-selections. We believe that selection expansion could benefit non-expert users and users who do a lot

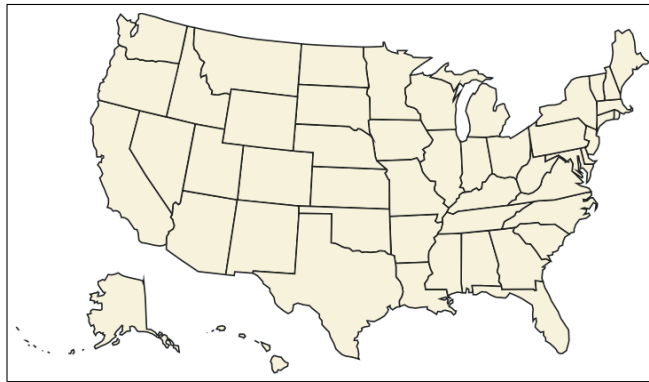
of prototyping in their designs. Several participants in our study said they would recommend the QuickSelect feature to friends (*“especially non-technical ones”*).

While we have shown one application in a vector graphics editor, the selection expansion algorithm could be applied to any interaction involving multiple selections. Prior work on generalizing selections has addressed text editors [Miller and Myers 2002] and file managers [Ritter and Basu 2009], and it would be interesting to explore how our selection expansion and their generalization techniques translate to different types of selections. Future work includes observing extended interaction sessions to determine how long it takes to build a usable history in different domains and at what point the cost of searching it becomes prohibitive.

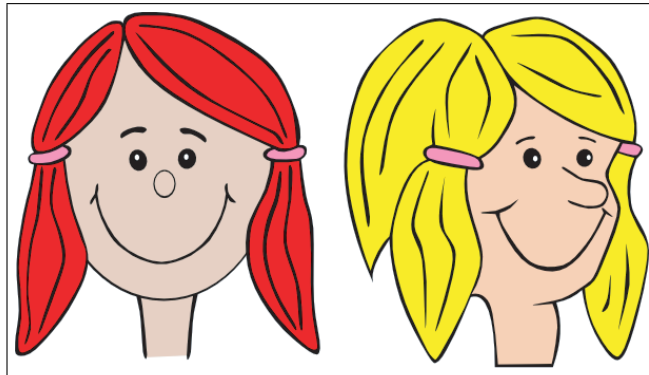
Looking at the bigger picture, we strongly believe that mining operation history can lead to enhanced human-computer interaction. The interaction history can reveal information about the document, the user, and the task. It can be used to leverage semantic properties without explicitly extracting them. Much instrumentation and logging in software today is done for the purposes of debugging and optimization, that is, for the benefit of the programmer. In this chapter, we have demonstrated one useful application of instrumentation to benefit the end user, and we hope that it will stimulate further research in this area.



(a) uranium task: "Delete all of the red balls."

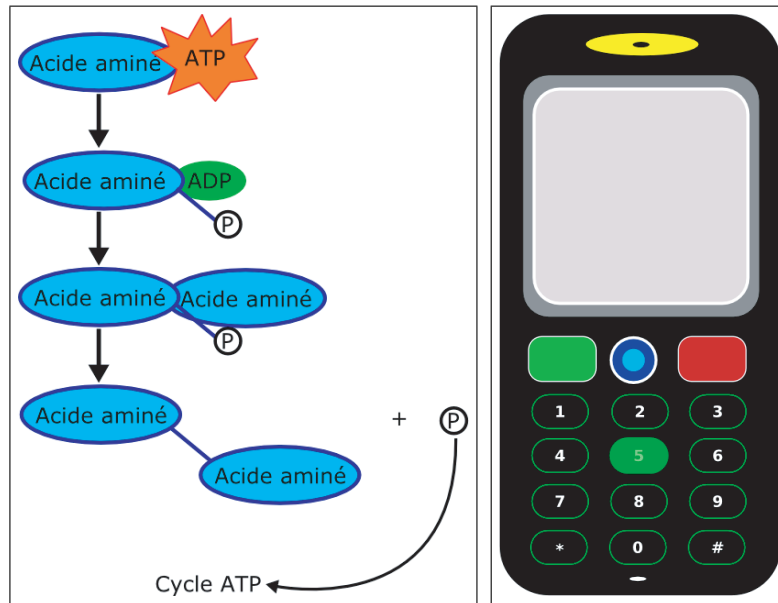


(b) map task: "Color all states on the East coast green."

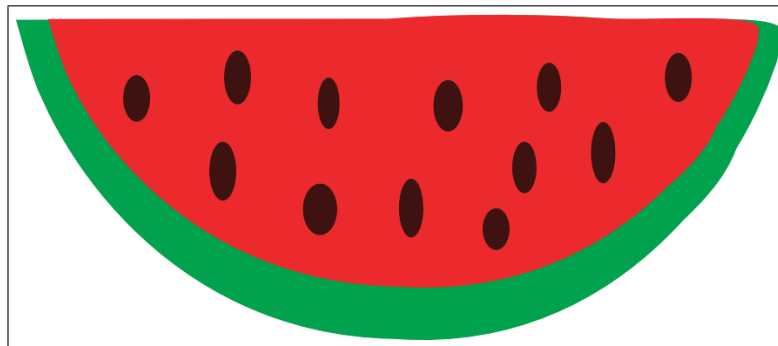


(c) girls task: "Change the hair of both girls to brown."

Figure 3-11: Tasks for which QuickSelect noticeably increased efficiency. (a) QuickSelect enabled users to re-select occluded balls. (b) Standard selection required more zooming to access differently-sized states. (c) The overlapping hair was easier to manipulate with QuickSelect.



(a) protein task: “Delete the blue ovals, including their text.” (b) mobile task: “Delete the first row of buttons (the un-numbered ones).”



(c) melon task: “Make the black seeds white.”

Figure 3-12: Tasks for which QuickSelect did not noticeably affect performance. The composition of these drawings is simple enough that direct selection methods could be used quickly.

Chapter 4

Soft Groups for Multiple Selection Authoring and Reuse

In this chapter, we introduce *soft groups*, a way to help users bookmark and reuse multiple selections. Soft groups enable easy reuse of complex selections by combining the benefits of standard groups and selections. Like standard groups, soft groups are persistent and easily reusable. Like selections, they are invoked on demand by the user, and an item can belong to more than one soft group. The user can select a number of items and mark this selection as a soft group. Later in the editing process, the user can review a list of the soft groups to which a query selection belongs. A soft group that comes up during this review process can be manipulated like any regular selection. This approach does not attempt to build a semantic model of the document or relations between items. Rather, the user notes which selections would be useful to return to.

We demonstrate this approach in the context of vector graphics editing and discuss the results of an exploratory evaluation comparing soft groups with existing selection and grouping methods, including selection expansion. Based on the user feedback, we discuss scenarios in which soft grouping would improve selection accuracy and efficiency.

4.1 Introduction

As we discussed in Chapter 3, selections, and multiple selections in particular, are a core component of many types of editing. It is common in the editing process to apply a change to several items at once, and one may wish to edit the same set of items together repeatedly. Re-selecting the members of this set can become tedious, particularly when the target items are at different scales or are partially occluded. The standard solution is to group items that are expected to be used together again [Golding 2005]. Large hierarchies of items can be created with the simple grouping command. However, this requires planning, as an item can belong to only one group at a time and there is a potentially high cost to ungrouping and regrouping items.

We propose a lightweight document structure, the *soft group*, that enables a user to bookmark meaningful multiple selections that they wish to reuse. The soft group is more persistent than a selection but less rigid than a standard group.

Soft grouping can be thought of as grouping by tagging. Consider the analogy of tags versus folder organization in media collections. A photo can belong to only one folder but can be selected using many tags. While we do not use tags in the sense of text labels, users can “tag” an item as a member of several soft groups simply by applying the group creation command to several selections. The creation interface is straightforward. The review behavior of soft groups is similar to that of selection expansion (Chapter 3), however, the soft groups presented and the presentation order are controlled by the user.

We demonstrate our approach in the context of vector graphics editing, implementing it as an extension to Inkscape [Inkscape 2003–]. Users can create and review soft groups in combination with existing selection and grouping tools. We have conducted a formative evaluation of soft groups with subjects who had beginner to intermediate experience with drawing software.

Overall, the study participants were positive about the soft grouping feature in Inkscape, and when asked to compare it with selection expansion, they could identify particular scenarios from their own experience in which each tool would be more applicable. We found that, in general, beginner-level users of drawing software preferred the efficiency of selection expansion, while intermediate-level users liked the greater precision afforded by soft groups. This suggests that soft

grouping addresses a weakness of selection expansion, the user's need for better predictability in the expansion steps. The pilot studies with our software prototype are promising and suggest that these tools are complementary to existing selection and grouping methods.

In the next section, we summarize related work in grouping and selection (§4.2). We then present the details of the soft group creation and review steps (§4.3.2) and their implementation as a feature in Inkscape (§4.3.3). Finally, we discuss the user feedback from the exploratory study (§4.4) and consider how selection expansion and soft grouping might be used together (§4.6).

4.2 Related Work

There are several areas of related previous work including selections, grouping, tagging for organization, and relation-building from interaction.

4.2.1 Selecting, Grouping, and Tagging

In addition to the standard single-click and region-based tools for multiple selections, some editors support select by attribute (e.g. Adobe Illustrator's "Select Similar" tool [Adobe Systems 1988–]). Recently, more sophisticated methods for *generalizing selections* have been proposed.

Miller and Myers [2002] have presented *selection guessing*, a method for inferring a multiple selections in a text-editing context. Heer et al. [2008a] have introduced a set of techniques for generating complex selection queries by generalizing from a simpler initial selection. Ritter and Basu [2009] have demonstrated their selection classifier for learning selection patterns in file manipulation.

Unlike these approaches, our method does not use inference or programming by demonstration. Instead, only those groups created by the user are presented for review. This simple yet effective strategy has the advantage of predictability; the behavior is easy for the user to understand. In the future, it would be interesting to explore uses of these techniques in combination.

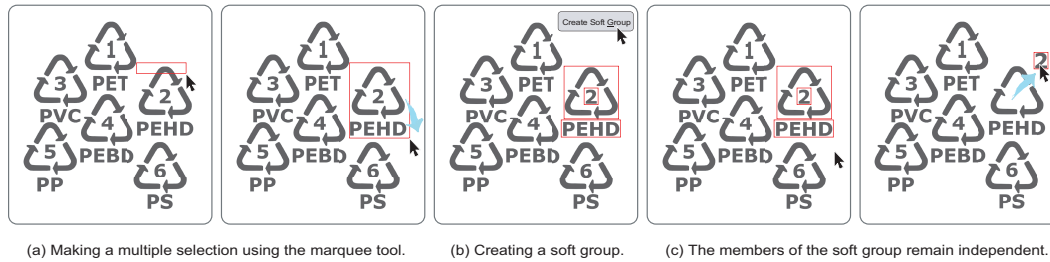


Figure 4-1: Soft Group creation in Inkscape. (a) The user first makes a multiple selection using an existing method, in this case, the marquee tool. (b) The creation command is called to mark this three-item selection as a soft group. (c) The three items now constitute a soft group. The members of a soft group can continue to be manipulated independently. For clarity, selections have been highlighted in red.

Like the ScanScribe editor [Saund et al. 2003], we aim to support a more flexible notion of grouping. While ScanScribe seeks perceptually significant structure in documents, our soft groups are determined by user selection.

We address the need for greater user control by allowing the user to annotate the history with which selections are worth returning to. This is similar in spirit to tagging systems (see e.g. Marlow et al. [2006], Golder and Hubermann [2006]), however we do not explicitly name or label the groups created.

4.2.2 Relation Building from Operation History

In another type of generalization, there has been recent work on building relations between digital objects based on monitoring of user activity. For example, Pedersen and McDonald [2008] have argued that a user’s interaction with documents can be interpreted as links. They have proposed a simple metric of document usage time to deduce relations between documents.

In Chapter 3, we introduced *selection expansion* to facilitate reuse of user selections. This approach takes the document history into account to determine which good candidates to be manipulated together in the future. A pilot study with non-experts suggested that selection expansion improves user efficiency in re-selection but that the expansion behavior is often difficult for users to predict. Our soft grouping provides users with more control by enabling them to specify which selections they would like to reuse.

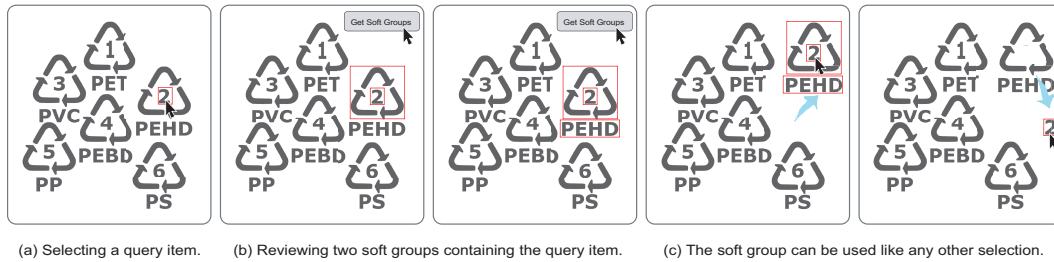


Figure 4-2: Soft Group review in Inkscape. In this example, we assume the user has previously created soft groups by several applications of the process shown in Figure 4-1. (a) The user selects a query item. (b) The user triggers two review queries to retrieve two soft groups containing the query item. (c) A returned soft group can be used like any regular selection. The members of a soft group can continue to be manipulated independently of one another.

4.3 Soft Group Creation and Review

We now describe the soft group technique. We first present the two steps of creation and review, then discuss the interaction design process of implementing the technique in the Inkscape vector graphics editor.

4.3.1 Soft Group Creation

The user creates a soft group by first making a selection using direct clicking or the marquee tool (Figure 4-1(a)), then calling the creation command (Figure 4-1(b)). There is no penalty to calling the function on a selection that has already been made into a soft group. In this interaction, creation of a soft group is like creation of a standard group.

4.3.2 Soft Group Review

The user reviews soft groups by first selecting one or more items of interest using existing selection methods (Figure 4-2(a)). Triggering a review query brings up a soft group containing the target selection. In Figure 4-2(b), we show two repeated review queries. A retrieved soft group can be used like any regular se-

lection (Figure 4-2(c)). Again, the members of a soft group can be manipulated as a unit or independently of one another.

The presentation order of the soft groups was determined through an iterative design process with pilot users. We tested three orderings that we believed would be easy for users to understand and predict: size, recency, and frequency. All of these orderings assume that the group creation step has already occurred.

Review by group size. This presentation order is straightforward. Soft groups are selected from the ones created by the user in order of increasing size. By size we mean the number of items belonging to the group. The motivation for this ordering is to progress intuitively from simple to complex selections. All of the pilot users found the ordering by size to be easy to understand and predict.

Review by recency (creation time). This presentation order was motivated by the hypothesis that users are more likely to revisit groups that were more recently used. Soft groups are presented in reverse chronological order, that is from most recently created to least recently created. The pilot users found this ordering easy to use, that is, it generated useful groups that they wished to revisit. However, they found it more difficult to predict than expansions by size, in part because the notion of expansion no longer holds.

Review by frequency (creation redundancy). This presentation order was motivated by the hypothesis that users will use the creation step repeatedly (and redundantly) on selections that they would like to revisit. Soft groups with higher frequency are presented first. Pilot users found this the most difficult to understand. For drawings of moderate complexity, they had little memory of which groups they had created more than once.

Based on the feedback from pilot users, we decided to order the presentation of soft groups by group size. To resolve ties, when there are multiple groups of the same size that could be presented, we use recency. As we will discuss in more detail later, users found this presentation order intuitive.

4.3.3 Implementation in a Graphical Editor

We demonstrate our approach in the context of vector graphics. We have implemented soft grouping as a feature in the Inkscape graphics editor [2003–]. In designing the interface, we sought to limit disruption to the user’s workflow. Based on interviews with pilot users, we determined that an interface easy to learn was a menu button and one easy to use was a keyboard shortcut.

Creating a soft group uses the same interaction as creating a traditional group. The user first selects the intended member objects, e.g. with the marquee tool. They then click the menu button or use the keyboard shortcut to make this selection into a soft group. Immediately after this, the selection/soft group can still be moved as a unit; the soft group creation step does not deselect the members. However, once deselected, the members can be clicked and moved independently (Figure 4-1(c)).

4.4 Exploratory Evaluation of Soft Groups

We conducted an exploratory user study in which we observed participants performing selection, drawing, and editing tasks in a lab environment. The goals of this study were to determine if the soft group interface improved efficiency with selection reuse, to identify scenarios in which soft groups are useful, and to draw our attention to weaknesses of the design.

We hypothesized that participants would find the soft groups interface easy to learn and use. We expected that, when asked to compare soft groups to selection expansion, participants would report that they would prefer to use both tools, but different ones in different usage scenarios.

Rather than comparing measurements such as success rates or task completion times, we tried to maximize ecological validity by recruiting a diverse group from the general population and by asking them to recreate drawings from their typical use [Winkel 1987]. We felt that free-form input from participants would be the most effective way of informing an evolving design. In this section, we will discuss recurring themes identified in the feedback.

4.4.1 Procedure

The study was designed to fit within a one-hour session, with each session involving a single participant. First, the participant was asked about the types of drawings he or she typically makes and the software tools used. Next, he or she completed an interactive tutorial to get familiar with Inkscape. As the objective of the study was to evaluate the soft grouping feature and not Inkscape, we helped participants through any usability problems encountered during the tutorial. We also simplified the Inkscape toolbar and menus to provide basic drawing functionality while limiting potentially distracting features.

The participant was asked to recreate one of the drawings described during the background interview. Observing the drawing process, the investigator encouraged the participant to use standard grouping and soft grouping. There was no measure of success. At the end of the editing session, participants were interviewed about their experience to record subjective preferences.

In the second part of the study, the participant was introduced to selection expansion. The participant was asked to continue editing the drawing while trying both soft groups and selection expansions. Although we did not perform a task-based quantitative comparison of the two methods, participants were interviewed after the drawing sessions and asked to compare them.

4.4.2 Participants

Nine participants (4 male, 5 female) participated in the study. They were between the ages of 20 and 36 ($M = 25.9, SD = 5.4$) and were recruited from the general population through an online ad. Subjects were paid for their time.

While only one participant was casually familiar with Inkscape, all had experience with at least one 2D vector drawing program such as Microsoft PowerPoint or Adobe Illustrator. Experience with this software ranged from beginner- to intermediate-level. The six intermediate-level users were employed in a creative field (website developer, cameraman, t-shirt artist, corporate designer, video student, logo designer). All had normal or corrected-to-normal vision.

4.4.3 Apparatus

Each session was conducted by one investigator with a single computer. Participants used a 2.33 GHz laptop running Windows XP with a 15" display at 1440 × 900 pixel resolution. They were asked to use a peripheral keyboard and mouse. Our modified version of Inkscape with soft groups was used for all drawing tasks. The investigator took extensive written notes during the sessions.

4.5 Observations and Feedback from User Study

Participants were asked to recreate drawings typical for them, resulting in a variety of designs including web-style graphics, poster layouts, and diagrams. Extensive notes taken during the study sessions were analyzed for themes across multiple participants that were then used to make conclusions about participants' experiences. While we did not count instances, most subjects incorporated standard selection, soft grouping, and selection expansion without prompting. In some cases, the experimenter suggested an edit involving a multiple selection. In general, subjects were eager to try the new features. Despite the range of participant backgrounds and drawings created, themes emerged from the debriefing interviews. We identified strengths and weaknesses of the soft group interface, which we summarize below.

4.5.1 Strengths

Simplicity and learnability. Soft grouping's straightforward interaction (creation and review) and similarity to manual grouping make it easy to learn. All nine study participants reported that the soft grouping feature was easy to pick up, and all appeared proficient with it after the tutorial.

Efficiency. After the first part of the study, we asked participants to comment on the effect of soft grouping on their editing efficiency. There was consensus among the participants that, while there was a fixed cost to creating the soft groups, once they had been created, the group review feature improved efficiency.

After selection expansion was introduced as an alternative, there was again consensus that it “*seemed faster*” than using soft grouping. However, users were divided over which method would be faster in practice. Interestingly, the feedback was split by experience level, with the more experienced users expressing concern that they would spend more time correcting for selection expansions that did not match their intent.

Control. Seven of the nine participants felt that soft grouping offered them greater control and precision than selection expansion. The remaining two felt there was no major difference in control.

Spatial memory cues. Participants commented that cycling through the retrieved soft groups provided a useful visualization of the history of changes. They used phrases such as “*visual history*”, “*visual reminder*”, “*reminders of what to use*”, “*re-editing*”, and “*history tool*” to describe what they liked about the soft groups.

4.5.2 Limitations and Possible Solutions

Error handling. To remove an item from a selection, the user can always use the standard “shift+click” to deselect it. Several participants expressed concern that there was no way to remove a soft group or mark it as uninteresting and not worth returning to. An obvious solution is to add a group removal command that can be called on any group that comes up in the review step. However, an extension to enable faster removal of a soft group, that is, without the constraint of being in the review step, would likely require scent [Pirolli and Card 1999] to identify candidates for removal. Additional future work is evaluation of the effect of the soft grouping feature on accuracy through longer-term observation.

Visibility and responsiveness. The first four study participants, while stating that the feature was generally easy to understand, shared a common concern that feedback for the created soft group was non-obvious. The feedback was text in the status bar. To fix this, we modified the prototype so that that bounding box of the selection would flash when it was made into a soft group (Figure 4-3). The

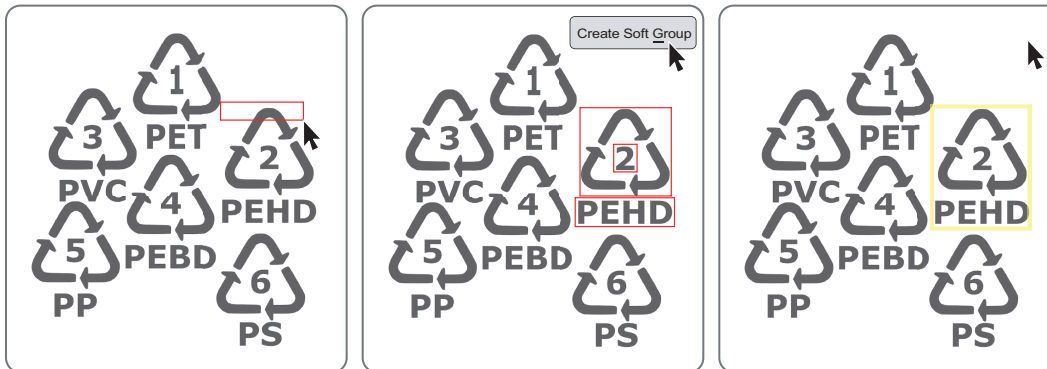


Figure 4-3: Several study participants requested visual confirmation that a soft group had been created. We modified the prototype so that that bounding box of the selection flashes when it is made into a soft group. Participants who used the modified prototype did not have complaints about visibility.

remaining six participants, who used the modified prototype, did not comment on the need for additional visual cues.

4.6 Discussion and Future Work

Overall, the feedback from the user sessions was positive, and several participants asked when the tool would be available in commercial software. The interviews also yielded examples of cases where soft grouping would be useful. Below is some of the freeform feedback we received.

- *“This saves me a lot of time clicking strokes and objects. I like the reminder of previous groups and reminders of what to use.”*
- *“This seems more useful for editing an existing drawing than for the first time creating it.”*
- *“It seems good for testing or iterating on a design.”*
- *“[Soft groups] would be useful when I am testing several swatches and repeatedly change the color.”*
- *“This would be good for wallpapers or when there are lots of small detailed objects.”*

- *“I would probably use this for Flash designs or icons. I think my more hard-core illustrator friends would use this more often.”*

We were particularly interested in participants’ thoughts on balancing efficiency and user control. When asked to compare soft grouping and selection expansion, beginner-level users stated a preference for the efficiency of the *“more automatic”* selection expansion, while intermediate users preferred the added control and predictability of the review process of soft groups. One user, a logo designer, said, *“I would use [selection expansion] for first drafts or sketches for customers, and soft groups for longer term projects.”*

Feedback from the study was promising and suggests that the appropriate use of soft groups is in combination with other selection and grouping methods. The study participants all agreed that soft grouping could co-exist with traditional methods if implemented in software they typically use. Future work includes longer-term observation to evaluate the impact of combinations of tools.

We consider a number of edge cases when using soft groups. In the case where a soft group is retrieved and a member of the group is then deleted, the remaining members are still grouped, following a “tagging” metaphor. However, we break this metaphor in the case where the user makes a copy of an object which is in a soft group; the new object is not a member of this group. These two examples illustrate our generally conservative grouping strategy, but future work is to add more sophisticated suggestion behavior to soft group creation. For example, we might suggest a soft group of objects with shared origins, for example, a result of copy-paste, creation by the same tool, or creation close in time. In addition, we could identify orthogonal groups.

Reviewing soft groups generates selections that can be used like any manual selection. Editing the selection does not change the soft group. Future work is to design the interaction for the case when the user does wish to edit an existing soft group. Presenting the soft groups for a particular object in an icon gallery could be a useful interesting alternative presentation for selecting and editing them. Several study participants suggested that, while the presentation order using increasing group size was most intuitive, it would be useful for them to see only the most recently used groups. It is worth investigating how far back in history we should go and what other cues in the history are relevant to the user.

4.7 Summary of Soft Groups

In this chapter, we have presented soft grouping, a technique for bookmarking and reuse of multiple selections. A soft group is more persistent than a standard selection but less rigid than a standard group. Like a group, a soft group is persistent. Like a selection, a soft group is recalled by the user. In many cases, soft grouping can increase user efficiency by providing an interface for reusing complex selections. In addition, soft grouping is more flexible than standard grouping, because members of a soft group can continue to be directly manipulated independently, and one item can belong to more than one soft group.

We have demonstrated the potential of soft groups as a complementary grouping technique in the Inkscape graphics editor. We believe that the strategy could be applied to any interaction in which there are multiple selections. Observing extended interaction sessions in domains such as text editing and file management is future work.

Chapter 5

Interactive Visual Histories

Presentation and graphics software enables users to experiment with variations of illustrations. They can revisit recent editing operations using the ubiquitous undo command, but they are limited to sequential exploration. We propose a new interaction metaphor and visualization for operation history. While editing, a user can access a history mode in which actions are denoted by graphical depictions appearing on top of the document. Our work is inspired by the visual language of film storyboards and assembly instructions. Our storyboard provides an *interactive visual history*, summarizing the editing of a document or a selected object. Each view is composed of *action depictions* representing the user's editing actions and enables the user to consider the operation history in context rather than in a disconnected list view. This metaphor provides instant access to any past action and we demonstrate that this is an intuitive interface to a selective undo mechanism.

5.1 Motivation

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, Corel Draw, and Microsoft PowerPoint facilitate exploration, trial-and-error editing, and refinement of designs. We aim to add to the flexibility of these interactions by leveraging the editing history of an illustration.

All modern text and graphics editors support a notion of history. The standard 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 argue that the creative process, which is often inherently nonlinear, should be supported by tools that are nonlinear. We present a new visualization that shows the actions in context and enables nonlinear exploration of history.

We show the illustration's history as a storyboard annotated with *action depictions*, graphical metaphors of editing actions such as fill color changes or spatial transformations (Figure 5-1). Our *graphical history mode* can be activated at any time during the editing process, and once activated, action depictions appear on top of the document. Users can view the history of the entire document or restrict it to a particular object or region.

5.2 Related Work

There has been much work on the history of web pages [Klemmer et al. 2002], graphic art [Meng et al. 1998], and views of datasets [Heer et al. 2008b]. Our work follows most closely from the Chimera system [Kurlander and Feiner 1992; Kurlander 1993], which shows graphical history in a series of panels, each one containing the before and after of an action. Chimera depicts several “actions” in the sense that the object selection or caret placement is displayed at the same time as the illustration modification (translation, color change, etc.). The action is described by a text annotation.

Our approach further develops the graphical aspect of the visualization, showing multiple actions at the same time and representing them by in-place graphical depictions rather than text annotations. We propose this fully graphical storyboard as an alternative, complementary visualization.

Recently, Nakamura and Igarashi [2008] presented a visualization system for generic Java AWT/Swing applications. While they focus on visualizing GUI events, such as mouse movement, we focus on visualizing changes to the document and

assisting the user's spatial memory of it.

We will discuss an application of the storyboard, using visual histories as an interface to non-sequential undo. The Photoshop history panel [Adobe Systems 1990–] and SolidWorks object tree [SolidWorks Corporation 1995–] have shown the value of maintaining and editing visual histories, and there has been significant progress made in 3D modeling and CAD [Hall et al. 1991; Bussan and Hall 1993; Cicirello and Regli 1999; Cicirello and Regli 2000; Schmidt et al. 2007].

The object tree in CAD systems provides a convenient interface for selective undo and parametric changes but requires significant structuring of the document. We want to offer some of the flexibility of these systems at a lower overhead to the user. Unlike the typical CAD file, illustrations in presentation slides are often done by casual users who are unlikely to create any hierarchy.

Our work differs in scope from Photoshop's History Brush [Adobe Systems 1990–], which lets users copy a previous state of an image into the current one. This can be seen as a per-pixel sequential undo because all modifications between the copied version and the current version are undone by the brush. Furthermore, Photoshop does not provide a visual representation of history.

5.3 Overview of Approach

Our storyboard is a visualization, as well as a new mode of interaction with a document's operation history. We describe how our storyboard interface enables non-sequential browsing and modification of a graphical content. We see the storyboard as complementary to existing interaction methods, and its visual histories are intuitive on the scale of the typical 2D vector illustration. In a usability study (§5.7), subjects easily understood how the storyboard relates to their actions and appreciated the new interaction modes. We target moderate complexity drawings on the scale of a typical clipart or PowerPoint illustration, which covers a large set of users. Comments from study participants also informed the design of extensions to the visualization (§5.8).

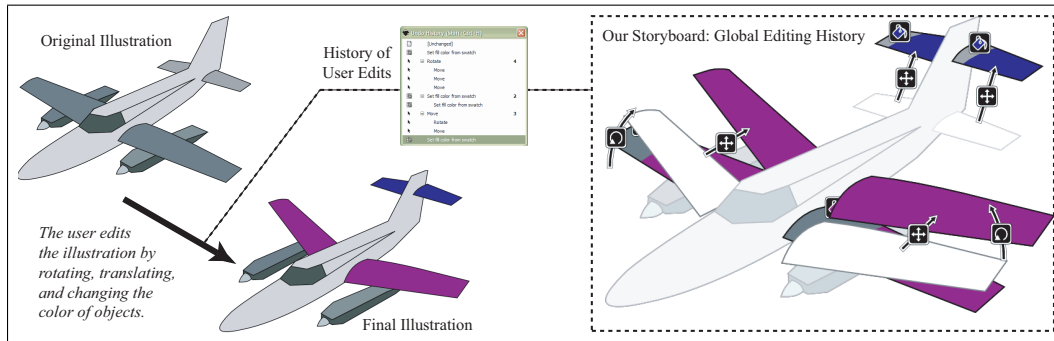


Figure 5-1: Overview of storyboards. This automatically-generated visualization shows the editing history of an illustration. Arrows and icons depict spatial transforms and color change actions performed by the user. The user can click on any depiction to selectively undo the corresponding action.

5.4 Design Goals

We have designed the storyboard annotations to be simple, with a common look and feel. The action depictions follow the visual language of assembly instructions [Agrawala et al. 2003], maps [Agrawala and Stolte 2001], comics [McCloud 1993], and film storyboards [Hart 1999; Tumminello 2005]. Together, the depictions can be seen as the “assembly instructions” for a document. Principles from cognitive psychology [Tversky et al. 2000; Tversky et al. 2006] inform our design strategy, summarized below.

Sequence of discrete steps. Humans naturally interpret and remember an event as a sequence of discrete steps [Zacks et al. 2001], and it has been shown that diagrams reflecting this structure are easier to comprehend [Seligmann and Feiner 1991; Agrawala et al. 2003; Heiser et al. 2004]. Our approach naturally achieves such sequential structure since the history of a vector graphics documents is a list of discrete steps. We refer to these steps as *actions*.

In place, static visualization. Our interface allows the user to consider actions in spatial context. It has been shown that in-place visualization of user interface transitions and variations [Terry et al. 2004] improve comprehension by exploiting spatial memory. Continuous changes over time, such as dragging an object to

translate it, could be depicted with animated visualizations. However, because experiments on the effectiveness of animation in explanatory diagrams are inconclusive [Tversky et al. 2002], we use static visualizations. Furthermore, one function of the storyboards is to enable users to select actions to undo, and a moving target could increase acquisition time.

Congruence, proximity, and comprehension. Most editing actions are visual in nature, and we rely on well-accepted schematic representations, such as an arrow for a translation [Tversky et al. 2000]. We have designed these visual elements to have a consistent look, exploiting the drawing primitives of the editing software itself.

Pilot users commented that “detached” icons were difficult to understand. In our storyboard, an action depiction is directly in contact with the object upon which it acts, e.g. a change of color is represented by a paint bucket overlaid on the object, visually relating the action to the object. We ghost unmodified regions to draw focus to objects actively being edited.

Before and after. We enable the user to compare the before and after states of a drawing in order to facilitate the decision to undo it or not. For example, we show the locations of an object before and after a translation, with “before” states rendered semi-transparent and in-place to limit screen clutter.

Saliency of depictions. We have designed the depictions to stand out from the document to avoid misinterpretation, and in the following section, we describe in more detail these design considerations. We have designed the storyboard such that all of the before-and-after object states are drawn in a single layer and all icons and arrows used in the depictions are in another layer. Practically, this allows the user to choose the amount of information to view: only the illustration, the illustration with just icon depictions, or the illustration with full depictions including before-and-after states.

Summarization. A single depiction limits the information provided to the user. Inspired by recent work on “reverse storyboarding” [Dony et al. 2005; Goldman et al. 2006], we summarize a segment of history with multiple depictions per

view. Our interface shares some of the annotation elements, but while these storyboards visualize an existing video sequence as a static image, our goal is to visualize the steps of constructing an illustration. To alleviate the visual clutter that can result from storyboarding a complicated document history, we support a “magic lens” interface [Bier et al. 1993; Hudson et al. 1997] for viewing the history of a selected object. User evaluation confirms that restricting the scope of the visualization is the preferred mode of interaction (§5.5.2).

5.5 Storyboard Visualization

We provide a *graphical history mode* that can be activated at any time when editing an illustration document. Our algorithm takes the operation history as input and creates a storyboard by traversing it from present to past. An entry in the history consists of a description of an action, its parameters, and a list of objects it acts on. For each type of action, we have designed an *action depiction*, a visual representation automatically generated based on the parameters of the action and its target objects. For instance, a translation of a polygon is depicted with a straight arrow. The action depictions are overlaid on the illustration. Among the challenges we address are designing visual metaphors to facilitate user comprehension and laying them out to minimize clutter.

5.5.1 Action Depictions

Our software prototype supports common actions on shapes, paths, and text. The actions we depict fall into the following categories: creations and deletions, spatial transforms, change of scale, fill and stroke attribute changes, and control point edits. Every action is responsible for its visualization. We have designed the action depictions with a common look and feel and display them in a layer on top of the illustration to emphasize figure-ground separation [Palmer 1999].

Spatial Transforms. We depict translation and rotation using arrows that share a similar visual style. We use thin arrows to avoid hiding the main illustration. Each arrow carries an icon indicating the transformation type. This icon also

provides a target easy to click. The arrows have a constant thickness to represent rigid transformations and a white outline to ensure visibility on any background.

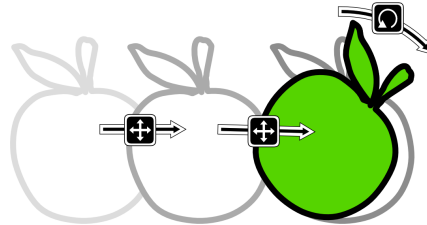


Figure 5-2: Action depictions: Translation and rotation.

For translations, we use straight arrows between the object’s old and new positions. The rotation arrow is similar; its length and orientation are the magnitude and direction of the transform. The arc shares the object’s center of rotation and its radius is the larger dimension of the object. For very small changes, such as keyboard “nudges”, we enforce a minimum arrow length.

Shape Modifications. The resizing depiction uses one to four axis-aligned arrows. To distinguish between scaling up and down and between scaling and other transforms, we use wide, tapered arrows: narrow-to-wide for up-scaling and wide-to-narrow for down-scaling.

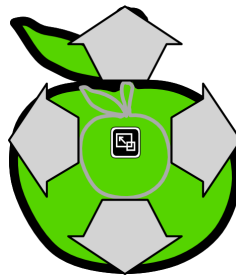


Figure 5-3: Action depiction: Resizing.

Editing control points, often used for refinement involving a series of many, in-place adjustments, is simply shown with the before and after states.

Appearance Changes. We depict changing fill and stroke style by overlaying a partial representation of the previous state on top of the current one. In practice, to create this half-object, we first create a cutting path as the diagonal half of the “after” object’s bounding box. We then take the Boolean difference of the object and the cutting triangle and overlay the result on the full “before” object. We complement the before/after overlay with icons, placed according to congruence rules, to provide an additional cue as to the type of change.

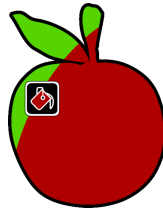


Figure 5-4: Action depiction: Change in fill style.

Creation and Deletion. Unlike other actions, creation and deletion do not affect any transformation on the object’s shape, position, or appearance style. Creation is easily identified due to the sequential appearance of the depictions, but for deletion we compensate by using an iconic cue. A gray, semi-transparent copy with an “X” icon represents a deleted object.

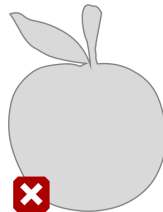


Figure 5-5: Action depiction: Deletion.

5.5.2 History in Context

Our storyboard provides an *interactive visual history* of a document, shown either as a summarizing *global storyboard* or a *local storyboard* for a selected object. From discussions with user experience designers, we understand that pro-

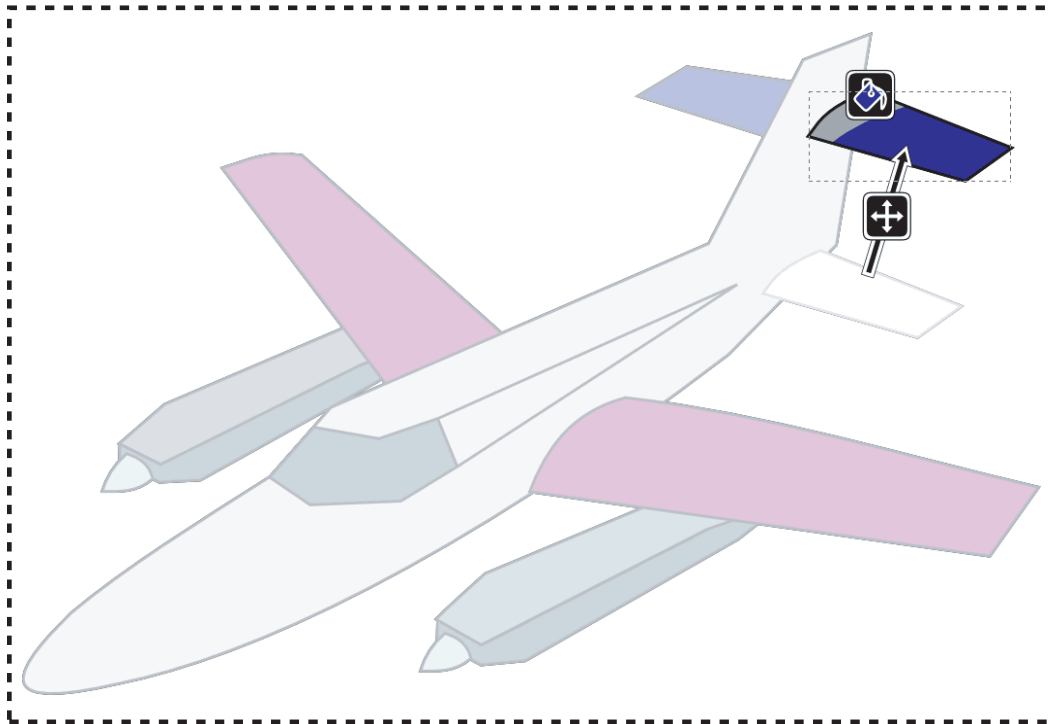


Figure 5-6: Per-object history. Once in storyboard mode, the user can restrict the history view (Figure 5-1) to a selected object. Here, only the history of one of the tail elevator pieces is shown.

viding both modes of interaction provides the best support for common producer (editing) and consumer (viewing) tasks.

The storyboard can be activated at any time during the editing process. Action depictions appear on top of the document, and users can select them to undo the corresponding actions. This allows the user to consider the actions in spatial context rather than in a disconnected list view. This visualization provides a natural interface to a selective undo mechanism.

5.6 Application: Non-Sequential Undo

Sequential undo, found in all existing illustration programs, removes the latest action from the top of the undo stack. To return to a previous state, the user has no choice but to lose all actions between then and the current state. A num-

ber of techniques have been proposed for less destructive, *non-sequential undo* mechanisms [Kurlander and Feiner 1990; Berlage 1994; Meng et al. 1998; Li and Li 2003; Kawasaki and Igarashi 2004]. We show that a useful application of our storyboard is as a front-end to non-sequential undo. The user can click on the depiction for any previous action to index into the history. Any action, not only the most recent one, can be undone. We briefly describe our use of storyboards for non-sequential undo in vector graphics editing.

5.6.1 Dependencies

We observe that many editing actions can be considered independently from each other. First, an action on one object can be safely undone without affecting actions on different objects. Second, we define the following *dependency classes*,

- Spatial transforms = {translation, rotation}
- Appearance changes = {fill style, stroke style}
- Shape modifications = {control point editing, resizing}

and argue that an action can be considered orthogonal to any action that does not belong to its class. We found that, in practice, these definitions are easy to understand. We keep as future work the study of alternative classes, e.g. setting fill and stroke independent.

5.6.2 Algorithm

Non-sequential undo is less destructive because, rather than canceling all actions after the selected one, it cancels only those acting on the same object that also belong to the same class. We execute a non-sequential undo command by issuing a number of queries to the undo stack. First, given the action depiction the user selected, we retrieve from the corresponding entry in the undo stack a list of objects affected by that action. Then, we query the stack for a list of actions subsequent to the undone actions that affected any of those objects. Among these actions, only those of the same class as the undone action are canceled. Figure 5-8 shows an example of the storyboard as an interface to selective undo.


```

SELECTIVEUNDO (Action  $a$ ):
  1.  $\{A_i\}$  = the list of objects  $a$  affects.
  2. Undo  $a$ , that is, remove it from the stack.
  3. Remove  $a$ 's depiction from the storyboard.
  4. Traverse the undo stack forward in time starting from  $a$ .
  5. For each action  $b$  that occurs later than  $a$ :
      (a)  $\{B_j\}$  = the list of objects  $b$  affects.
      (b) If  $\{A_i\} \cap \{B_j\} \neq \emptyset$ , and  $a$  and  $b$  belong to the same dependency class:
          i. SelectiveUndo ( $b$ )

```

Figure 5-7: Pseudo-code of the selective undo algorithm.

5.7 Exploratory Evaluation of History Storyboards

We have implemented the storyboards and non-sequential undo as an extension to the vector graphics editor Inkscape [2003–]. Our implementation reuses Inkscape drawing primitives to display the storyboard and create action depictions. For instance, we desaturate the background illustrations by overlaying a semi-transparent white rectangle and use Inkscape’s existing Boolean operations to create cutting paths for the ghost (“before”) objects.

We conducted a concept evaluation by observing first-time users of our prototype. User interaction with drawing tools is by nature complex. This suggests that using a standardized task for quantitative analysis would require a great simplification of this interaction [Olsen, Jr. 2007]. Rather than comparing measurements such as success rates or task completion times, we asked participants to recreate drawings from their typical use and performed a qualitative analysis by interviewing participants and extracting recurring themes.

5.7.1 Participants

Twelve participants (6 male, 6 female) aged 20-40 years were recruited for the study and received a gratuity for their time. Half were proficient computer users

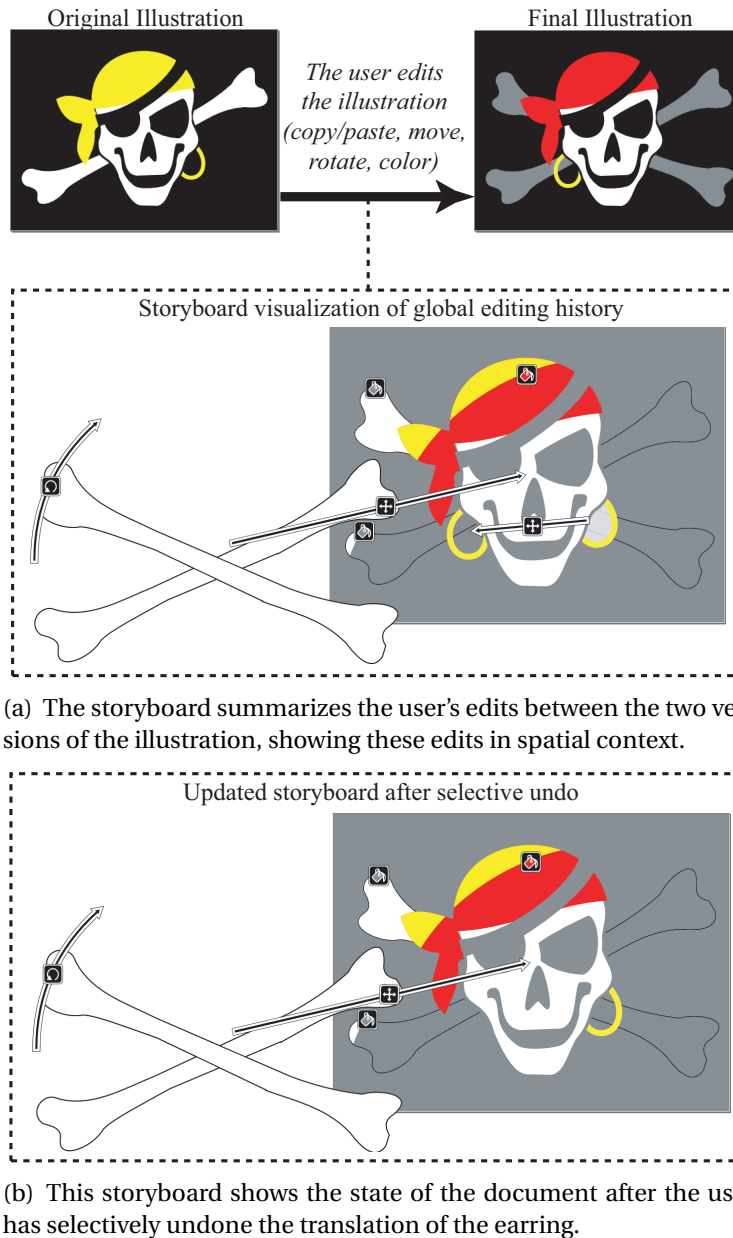


Figure 5-8: The storyboard visualization provides a natural interface for non-sequential interaction, specifically selective undo. Note that the selective undo of the earring movement (an action occurring farther back in the history stack) does not affect any other object. In contrast, standard (sequential) undo would have required also undoing all subsequent actions.

from a university and half were from the broader community. Some had familiarity with Illustrator; all were familiar with the drawing tools in PowerPoint. The university participants had prior experience creating figures for papers or presentations, and the participants from the community had a wide range of experience. None were graphic designers or creative professionals, and one was casually familiar with Inkscape. Despite the range of experiences, all participants were self-described “perfectionists” when it came to their drawings.

5.7.2 Apparatus

Subjects used a computer running Windows XP, with a display at 1280 × 1024 pixel resolution. Our modified version of Inkscape, with history storyboard and non-sequential undo extensions, was used for all drawing tasks. Because the goal of the study was to observe how participants used the storyboard, independent of the particulars of the Inkscape interface, we removed all non-essential widgets. In addition, we disabled several keyboard shortcuts and contextual menus to make non-sequential and standard (sequential) undo equally accessible.

5.7.3 Design

First, participants were asked about their background: occupation, types of drawings normally made, and tools used. Second, they were given a short tutorial on Inkscape and the history features and then asked to create one of the drawings they had described earlier. During the drawing task, participants were encouraged to try out the new history functions. Finally, participants were interviewed about their experience to record subjective preferences.

5.7.4 Results

Participants were asked to recreate drawings typical for them, resulting in a diverse sample of illustrations including: a party invitation, CPU and web architecture diagrams, clip art, a floor plan, a kitchen remodeling plan, concept and workflow models, an organization chart, and a neuron diagram. Some are shown

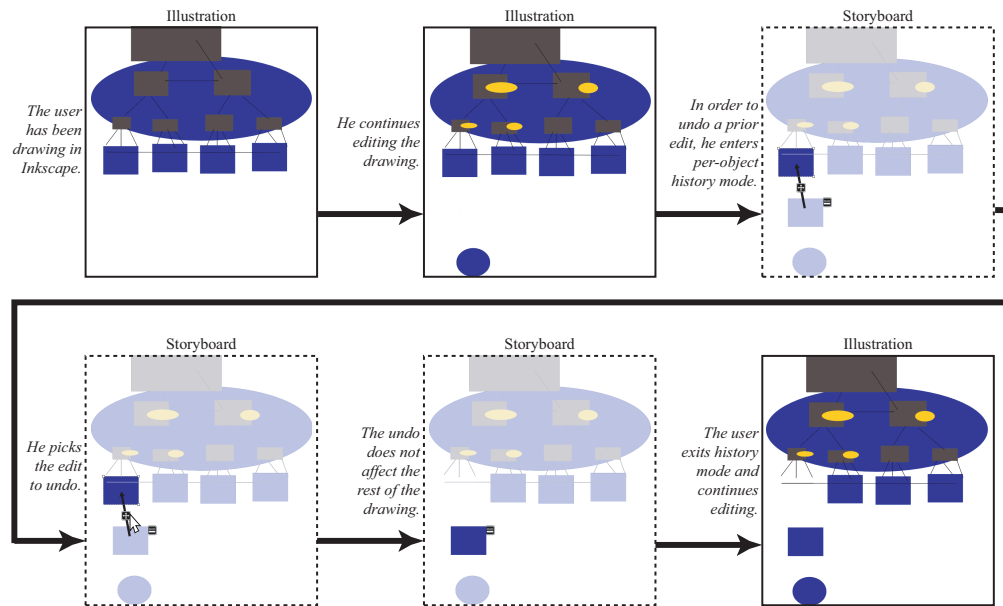


Figure 5-9: Participants in the user study were asked to recreate drawings typical for them. Over the course of the editing sessions, they were asked to make modifications to their drawings in an effort to encourage the use of the visualization and undo.

in Figures 5-9 and 5-13. Additional examples created by the authors are shown in Figures 5-1 and 5-8.

Extensive notes taken during the study sessions were analyzed for themes across multiple participants that were then used to make conclusions about participants' experiences. Despite the wide range of participant backgrounds and drawing experiences, themes emerged from the interviews.

5.7.5 Strengths

Free experimentation (nonlinear working style). Participants said that, compared to the tools they normally use, the storyboard allowed them to more freely experiment and try out new ideas. Participants often made a series of precise alignments, pixel-level adjustments, and precise color changes which would have been tedious to recreate in a traditional history that undoes every change after the one selected.

Spatial memory cues. Some participants commented that visualization of the history helped them to recall previously made changes, which they found helpful to reconsider.

Persistent history. Some participants described past situations in which they would have wanted to revert to changes from prior working sessions. One participant said, *“I have accidentally made changes to a file before, saved it over top of the original and then had no way to retrace steps to the original version.”*

5.7.6 Limitations

A common request from the study participants was for a shortcut or contextual menu to access the undo functions. Although these features are available in normal use, we had disabled them for the study to avoid bias. Several participants reported being overloaded by the number of depictions, particularly with the global storyboard, or felt that the history got in the way of the illustration; they would have liked a means to view their “pristine” drawing. We expected these comments as our approach deliberately adds visual depictions on top of the drawing. Nevertheless, to address these reactions, we added a shortcut to swap between the history and editing views. In addition, we developed a new multi-frame view in which the number of depictions is limited; we describe this extension in the following section.

5.8 Extensions

5.8.1 Multi-Frame Storyboards

A long illustration session can generate a dense storyboard when all actions are displayed at the same time. On the other hand, showing singular actions limits the information provided to the user. User feedback confirmed the need for balance in the visualization. We propose a hybrid storyboard that shows multiple actions per view or frame.

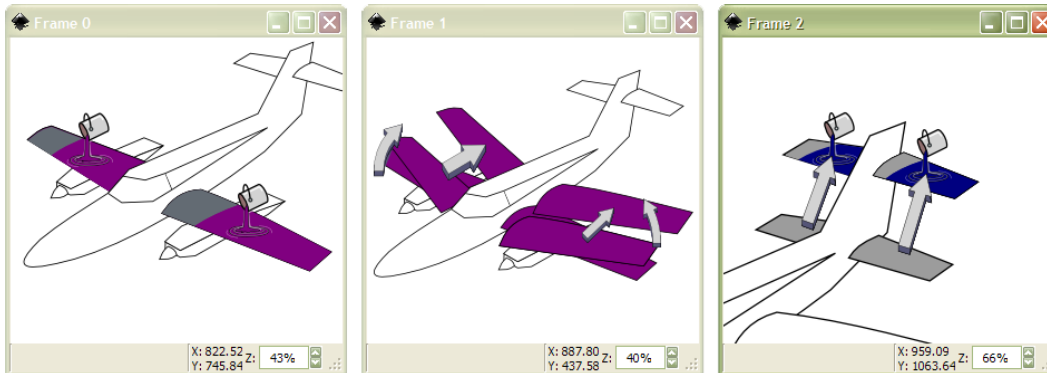


Figure 5-10: Multi-frame storyboard. This storyboard summarizes the editing history of a document, with time progressing from left to right. In addition, this storyboard shows an alternative depiction style we have considered.

To create this multi-frame storyboard (Figure 5-10), we traverse the history stack from the most recent action. We use a greedy strategy to avoid visual clutter when generating the depictions. If adding the next depiction would fully obscure an existing one or overlap more than a set maximum number of existing ones, we create a new frame; otherwise we draw in the current one. In addition to the visibility rule, we set a maximum number of depictions per frame. The viewport and zoom level are set so that all depictions and the objects they affect are visible. In addition, we set minimum and maximum zooms as functions of the overall drawing size.

5.8.2 Editing in Multiple-User Environments

The storyboard facilitates editing by multiple users in a manner similar to the “track changes” feature available in text editors such as Microsoft Word. Every editing action is tagged with the ID of the current user. In the storyboard, actions are color-coded by author, making visible the interactions between all the collaborators (Figure 5-12). This could, for example, assist multiple authors working on the same figure for a paper. 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 [Abowd and Dix 1992].

```
DRAWMULTIFRAMESTORYBOARD (HistoryStack  $H$ ):
```

1. For each action a in H :
 - (a) Create a new action depiction d_a .
 - (b) i_a = number of intersections of d_a with depictions in the current frame.
 - (c) If d_a fully obscures any existing depiction or if $i_a > \text{maxIntersections}$:
 - i. Create a new frame.
 - ii. Set the current frame to be the new frame.
 - (d) Draw d_a in the current frame.
 - (e) Adjust the zoom level.

Figure 5-11: Pseudo-code of the multi-frame storyboard creation process.

5.9 Discussion

Our storyboard visualization can 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. In particular, our user study has shown that people appreciate the additional freedom afforded by this interface to selective undo and that our storyboard aids their comprehension of the creation process as a whole, beyond the currently displayed illustration.

5.9.1 Future Work

Alternative Styles. A limitation of our approach is that the depictions may be misinterpreted when overlaid on an illustration that uses similar drawing components and conventions. In this chapter, we have shown examples of the “thin arrow + icon” style, as it minimizes clutter and fits well to most illustrations. A possible extension is to offer several depiction sets so that users can select a style not conflicting with their own illustration. For instance, movie makers would appreciate thick 3D arrows typical from movie storyboards [Hart 1999]. Figure 5-10

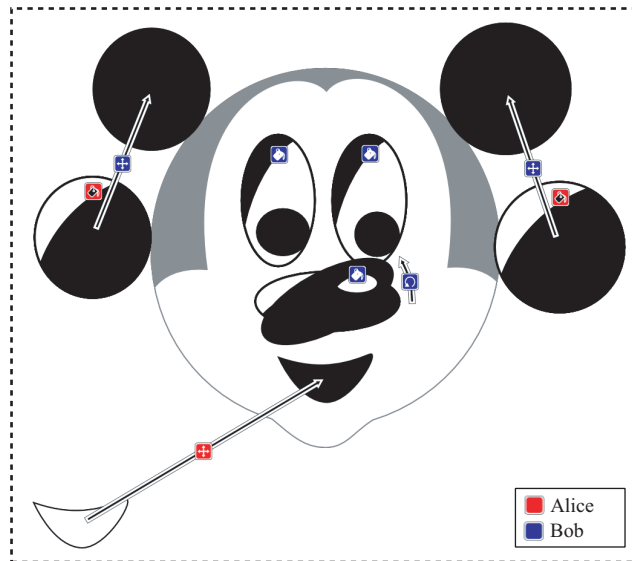


Figure 5-12: Collaborative editing scenario. Two users have edited this drawing, and the storyboard displays their actions color-coded in a “track changes” style.

shows such a style, which we have considered in our preliminary studies. We use bold arrows inspired by traditional storyboards. This makes the depictions easy to recognize and select and differentiates them from the underlying illustration. The size of the arrow scales with the size of the object and the editor viewport.

We have also experimented with transition effects [Baudisch et al. 2006] and text annotation. Text labels can provide additional cues as to the type of spatial transform and help differentiate between arrows in our visualization. Because pilot users reported that they found the arrows self-explanatory in most cases and did not need to refer to the labels, we disabled this feature for the full user study. However, it would be worth studying in what cases text annotation would add additional information. For instance, numbering operations could help with storyboard legibility.

Prototyping and Tutorial Applications. Storyboards encourage exploratory design by enabling the user to easily survey previous actions in spatial context and selectively undo them. An interesting direction for future work is to improve storyboards with design galleries [Marks et al. 1997] or other systems for visualizing variations to further facilitate prototyping.

In addition to undoing an action, it would be useful to modify its parameters or apply it to another object. Handling such modifications and propagating effects on actions motivates future work on macros and redo.

Another promising application is instruction. An expert's storyboards could serve as tutorials by revealing the editing process to less experienced users. Both transcription and translation of recorded interaction history for tutorial purposes involve challenges that should be interesting future work. While we have shown applications in vector graphics editing, we believe that these visualization techniques could be extended to aid prototyping, collaboration, and instruction in other domains.

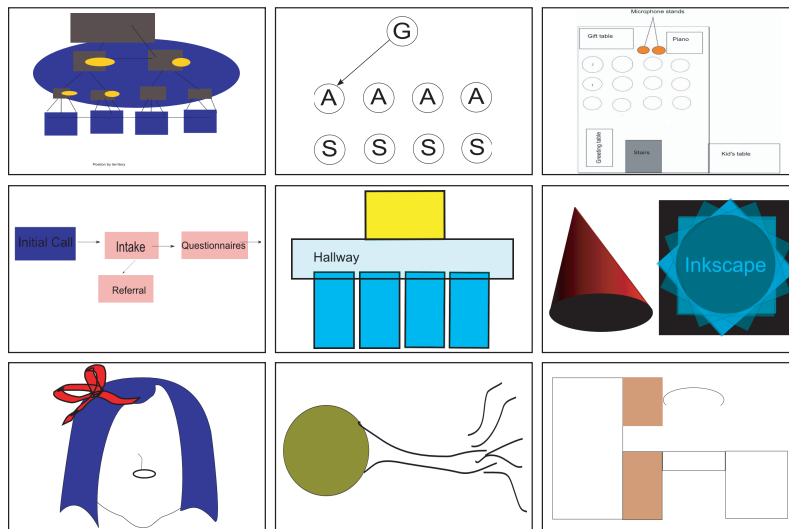


Figure 5-13: A sampling of vector drawings made by participants in our study. All were amateurs who had prior experience with Microsoft PowerPoint's drawing tools, and some were familiar with Adobe Illustrator. All were first-time users of our software, yet were able to create a wide range of drawings in a short period of time.

Chapter 6

Conclusions

In this dissertation, we have addressed the problem of mining of operation history for the benefit of the end user. We have presented strategies for increasing user efficiency in repetitive editing tasks and for supporting user browsing and manipulation of history. In the iterative design process, we have been guided by studies by cognitive psychologists on understanding of diagrams, as well as feedback from users of our prototype software.

This dissertation presents three main contributions in the field of computer graphics and interactive techniques. We have introduced *selection expansion*, a method for facilitating the reuse of complex multiple-item selections by identifying items that are likely to be edited together. We have presented the *soft grouping* technique, which gives users more control than standard selection and more flexibility than standard grouping. We have presented *interactive storyboards*, a visualization which enables in-context browsing and manipulation of operation history. We have demonstrated these techniques in the context of vector graphics editing with implementation in the Inkscape editor.

6.1 Expressibility of Selections and Soft Groups

We have shown that selection expansion is a simple but effective technique for retrieving frequently used selections (Chapter 3). Our formative user studies showed that selection expansion and soft groups can be useful in different sit-

uations. Both require the user to make manual selections initially, but over extended use, can improve user productivity.

Generally, users reported that selection expansion is more efficient, while soft group provides greater user control. Future work is to consider an intermediate design, perhaps using user-created soft groups as “seeds” to steer multiple possible selections expansions for a single object. Future work is to suggest soft groups or provide the option to group results of a copy-paste. Additional work lies in designing the interaction for cases in which the user wishes to edit an existing soft group. Presenting the soft groups for a particular object in an icon gallery could be an alternative presentation for selecting and editing them.

6.2 Expressibility of Visual Histories

As discussed in Chapter 5, in designing storyboard visual histories, we have targeted moderate complexity drawings on the scale of a typical PowerPoint illustration or clipart. Interviewing the dozen participants in our user study, we learned that they typically spend several hours on a drawing of such complexity, spread out over one to seven days. This provided a reference point when improving our design, especially to handle clutter. We feel that optimizing for this level of drawing is appropriate given its prevalence. In our qualitative study, users easily understood how our storyboard relates to their actions and appreciated the new interaction modes. We believe that our approach should appeal to the wide audience of presentation software users.

We do not claim that the depictions proposed in this dissertation are spanning, but we believe they are representative. We are confident that the framework could be extended to visualize a richer set of interactions. Further study is needed to determine the relative difficulty of visualizing particular classes of user interaction history and the added cognitive load of expanding the set of supported user operations.

Some operations can be supported as special cases of existing ones. For example, we handle a copy-object/paste-object sequence of operations as a special case of creation, and pasted objects are not highlighted with the typical creation icon. We plan to extend our set of depictions to include ones for these actions

and copy-property. Other operations are more difficult to visually represent with the existing arrows and icons in our storyboards. For instance, operations that relate two or more objects, such as alignment or clone, will require a more complex visual vocabulary to properly show the affected objects, the nature of the relation, and causality.

6.3 Directions for Future Work

We believe that the history-based techniques we have discussed will help users take fuller advantage of design on the computer by making the process easier, more flexible, and more controllable. In this dissertation, we have discussed integrating these techniques in an existing full-featured drawing package in order to involve real users in the iterative design. While we have thus far focused on graphical editing, the techniques have applications in many domains. Extending the ideas to improving the internet experience opens up a number of challenging research topics.

Storyboards encourage exploratory design by enabling the user to easily survey previous actions in spatial context, manipulate them, and selectively undo them. 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. Design galleries [Marks et al. 1997] and other visualization techniques would facilitate user exploration of variations.

Color-coding of depictions in the storyboard facilitates asynchronous collaborative editing. This “track changes” feature would benefit, for example, multiple authors editing a website. Extending this to the real-time, long-distance collaboration that the internet enables will require thoughtful design.

Reviewing the editing history of an experienced user can be an effective way for novices to learn a new technique or a trademark style. Unfortunately, raw recorded history can be too much data and can become for the end user to parse. Visualizations can go a long way to editing this into an understandable, educational form. An expert’s storyboards could help beginners learn about an illustration by revealing the structure and editing process. Designing context-specific

visualizations for data collected from online behavior is an interesting direction for future work.

Moving forward, we expect to continue conducting interviews and interaction analyses to better understand behavior during longer-term use of our software. This fieldwork is important to designing meaningful instrumentation and metrics for software, to sketching compelling visual representations of the data, and to developing technologies to support and enhance the user experience. We are excited about extending this work, and in addition to opening up exciting avenues for future research, we believe it will result in useful, tangible artifacts for the community.

Appendix A

Stimuli

In this appendix, we describe the stimuli used in the evaluation of selection expansion (“Selection Reuse with Existing Histories”, §3.4). In the user study, each participant completed a series of twenty selection authoring tasks. Each task presented one of the following twenty drawings that had been previously edited (Figures A-1 through A-20).

Source documents came from the Open Clip Art Library [2004–] and were edited by the authors. The histories attached to the drawings were recorded as the authors edited the files. These edits included simplification, rearrangement of objects, and attribute changes. The task given to the subject was chosen with the history in mind. The task could always be completed using either standard selection or selection expansion; with the latter, more than one expansion step was always necessary to reach the target set.

The following stimuli drawings are presented in alphabetical order, while in the experiment, the presentation order was randomized to minimize a learning effect. For each, we give the task statement presented to the study participant and briefly describe the history attached to the drawing.

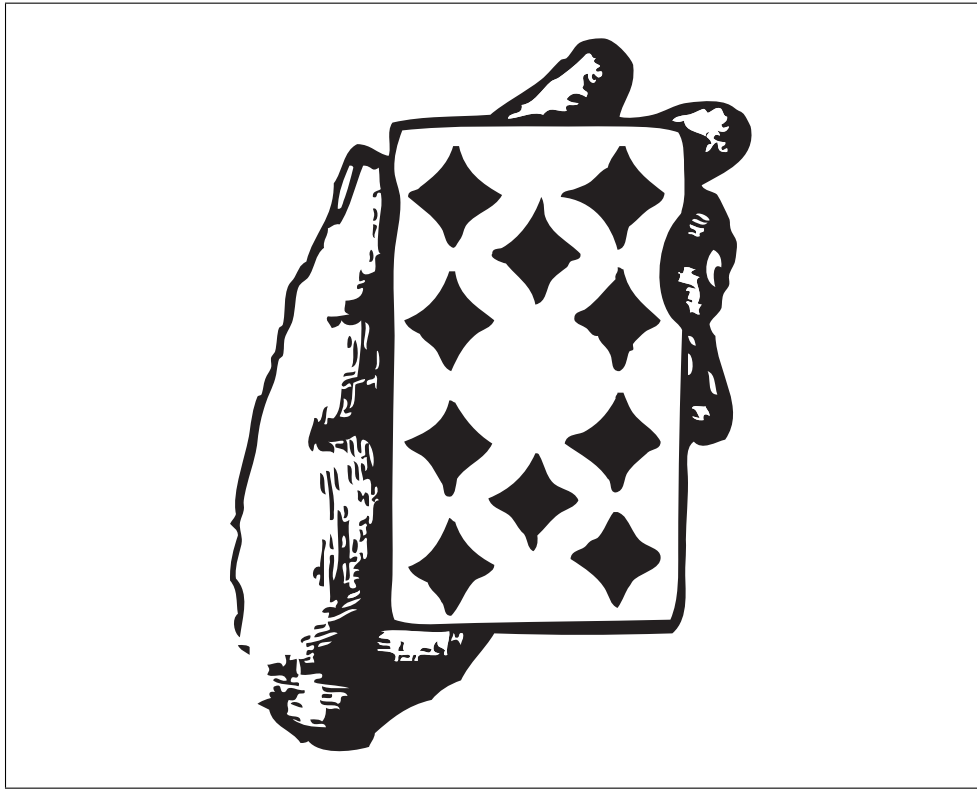


Figure A-1: Task card: “Make all of the diamonds red.” To complete this task, ten objects (the diamonds) must be selected. Using the pre-existing history, this could be accomplished with a three-step selection expansion.



Figure A-2: Task c1ock: "Change the color of the time." To complete this task, five objects (the clock digits) must be selected. Using the pre-existing history, this could be accomplished with a three-step selection expansion.

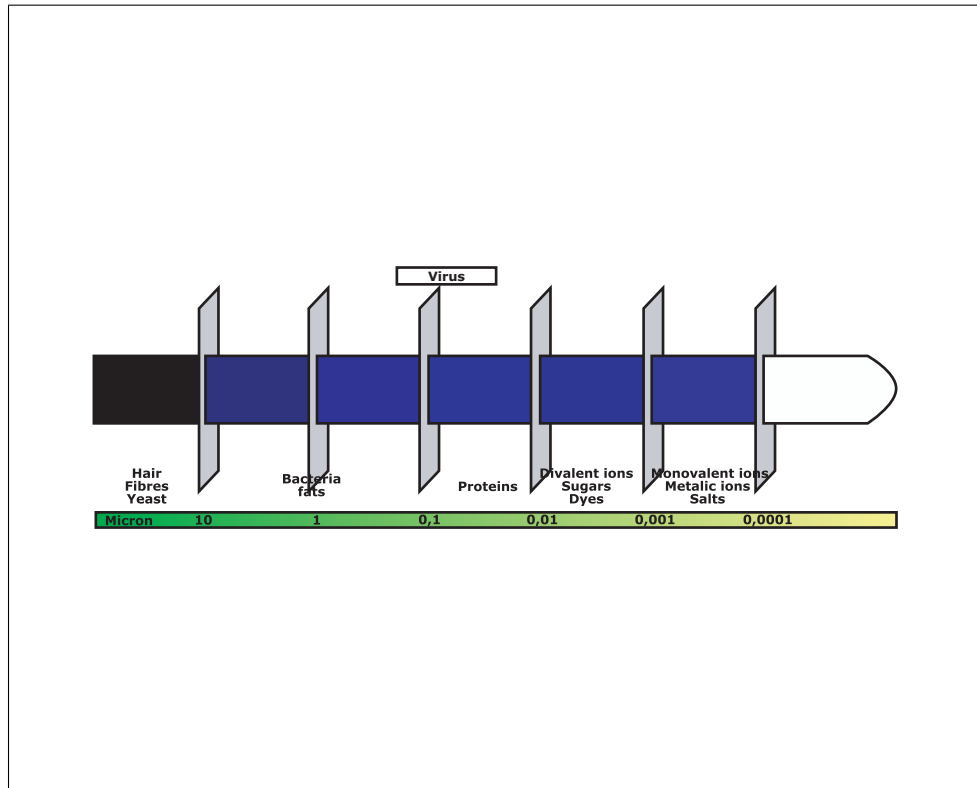


Figure A-3: Task filtration: “Move all explanatory text to the top of the diagram.” To complete this task, fourteen objects must be selected. Using the pre-existing history, this could be accomplished with a four-step selection expansion.

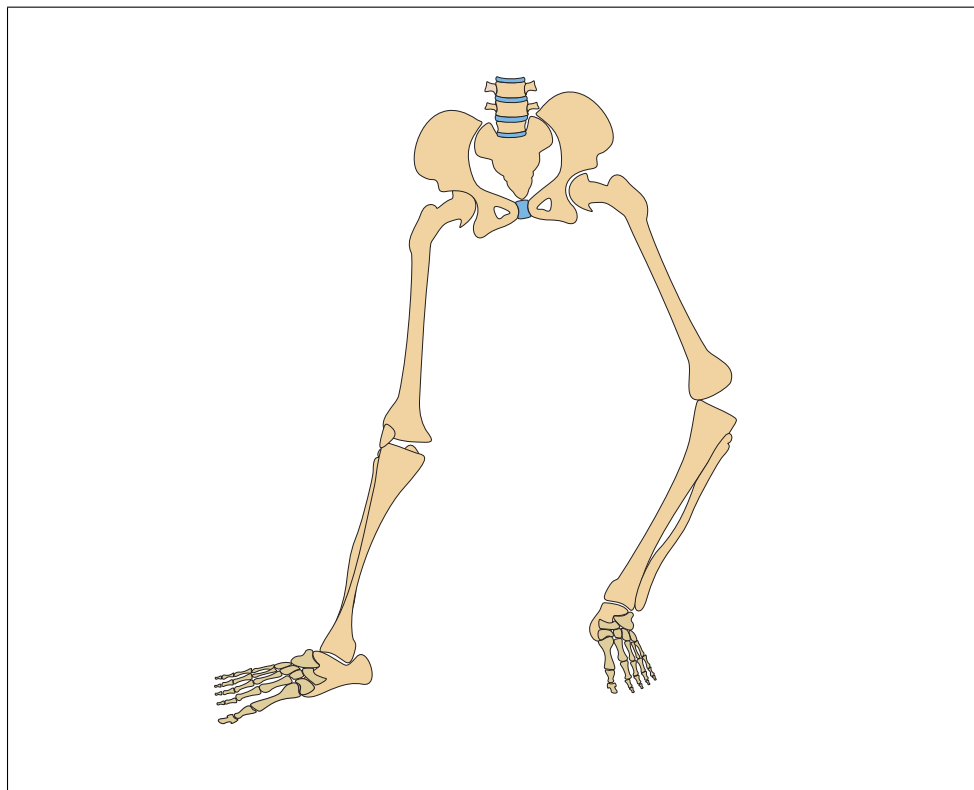


Figure A-4: Task foot: “Bend the left leg at the knee.” To complete this task, thirty objects (the bones) must be selected. Using the pre-existing history, this could be accomplished with a four-step selection expansion.

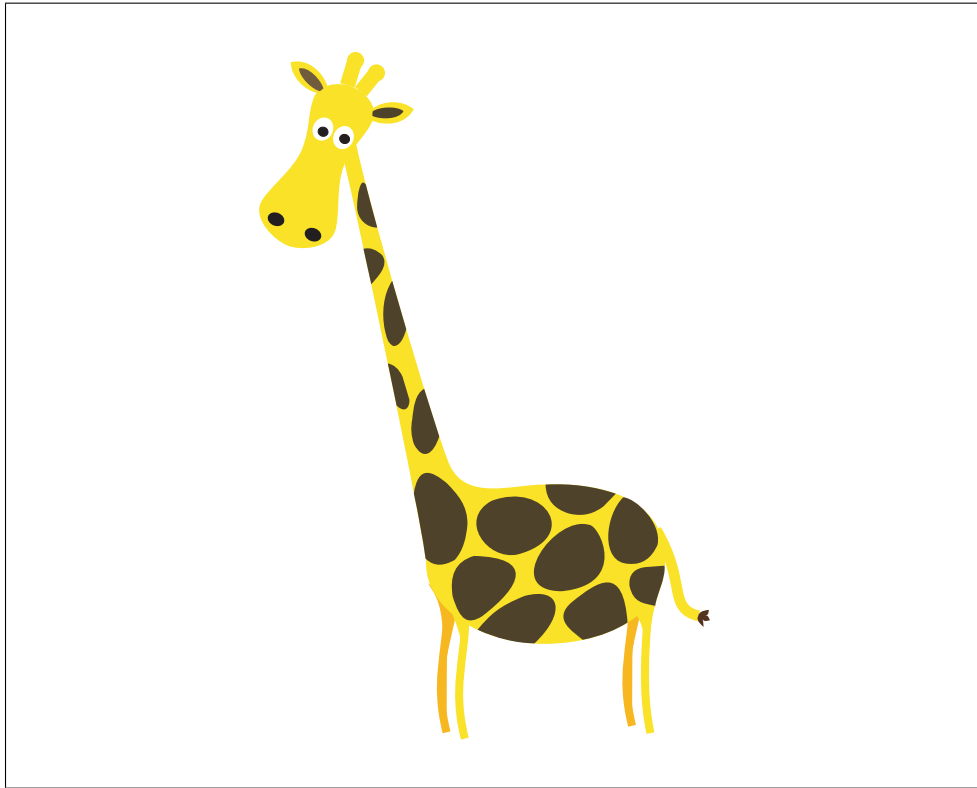


Figure A-5: Task giraffe: "Remove the giraffe's spots." To complete this task, fourteen objects must be selected. Using the pre-existing history, this could be accomplished with a three-step selection expansion.

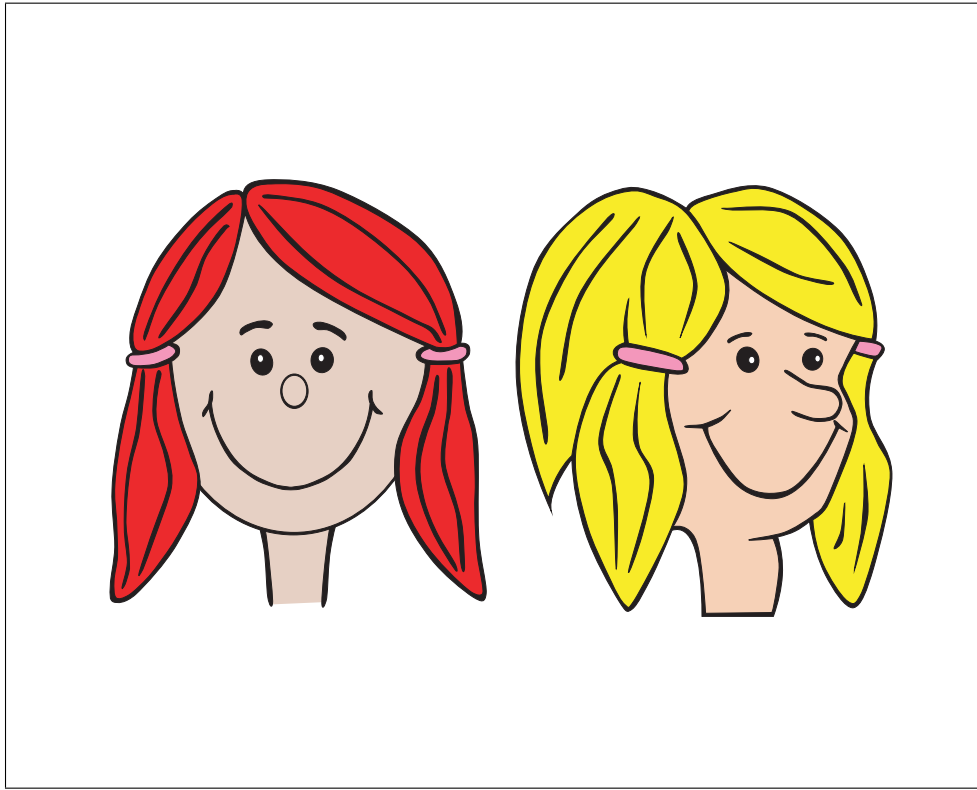


Figure A-6: Task girls: “Change the hair of both girls to brown.” To complete this task, eight objects must be selected. Using the pre-existing history, this could be accomplished with two two-step selection expansions.

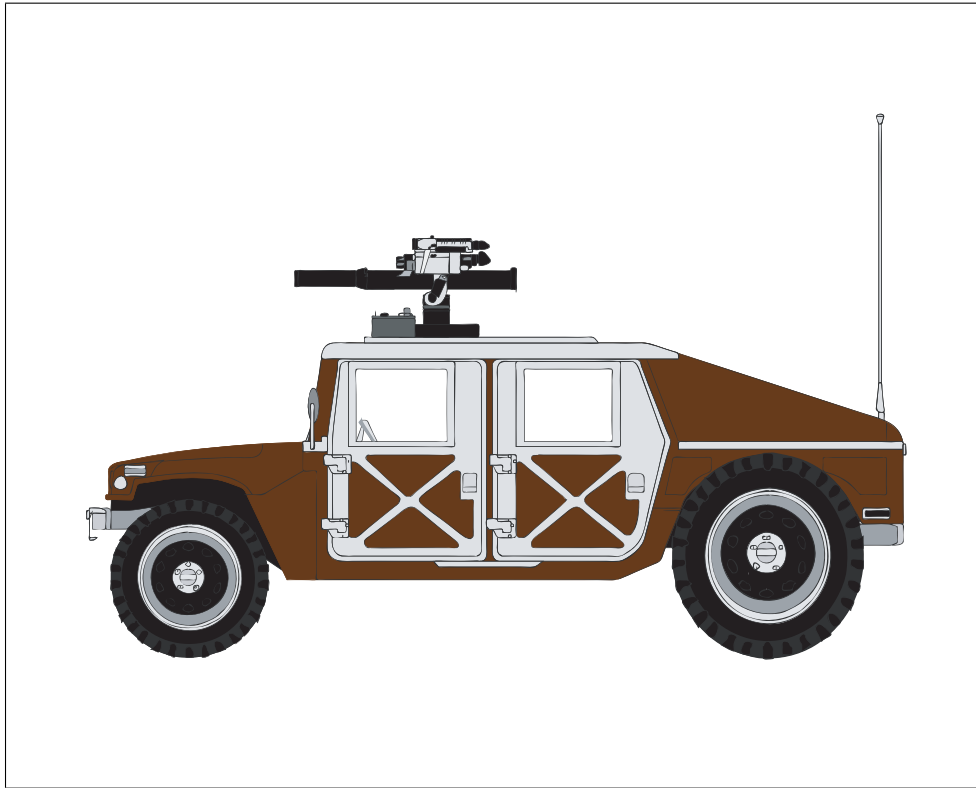


Figure A-7: Task jeep: “Delete the triangular pieces on the doors.” To complete this task, sixteen objects must be selected. (The brown filled regions and black outlines are separate drawing objects.) Using the pre-existing history, this could be accomplished with a five-step selection expansion.

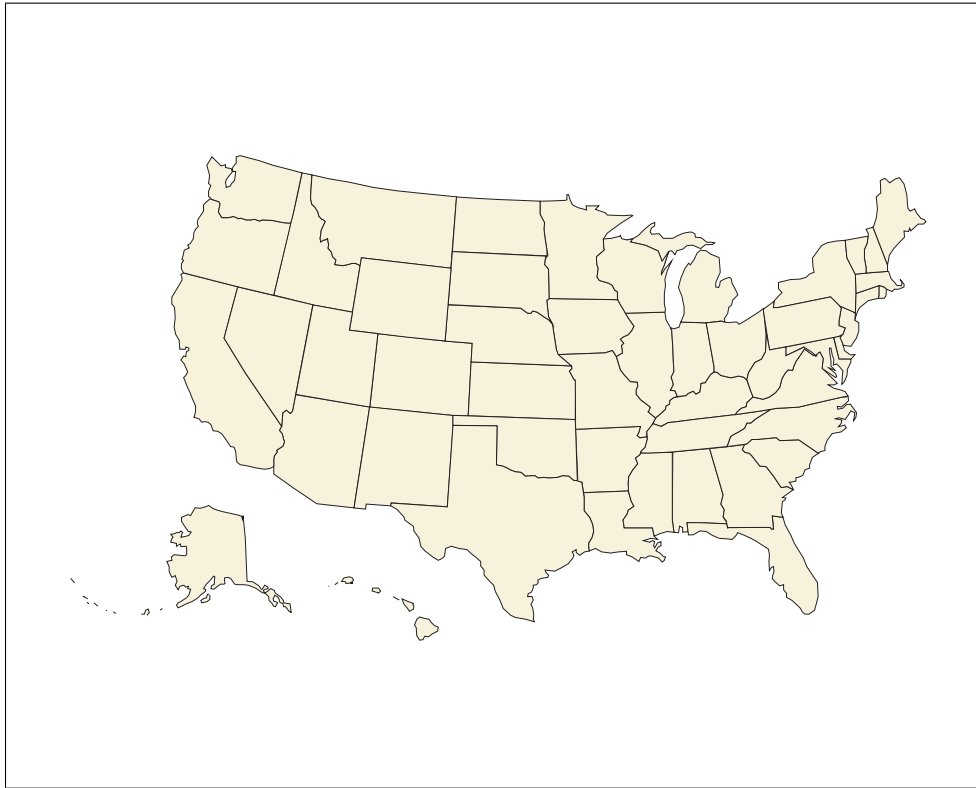


Figure A-8: Task map: “Color all states on the East coast green.” To complete this task, fifteen objects must be selected. Using the pre-existing history, this could be accomplished with a four-step selection expansion.

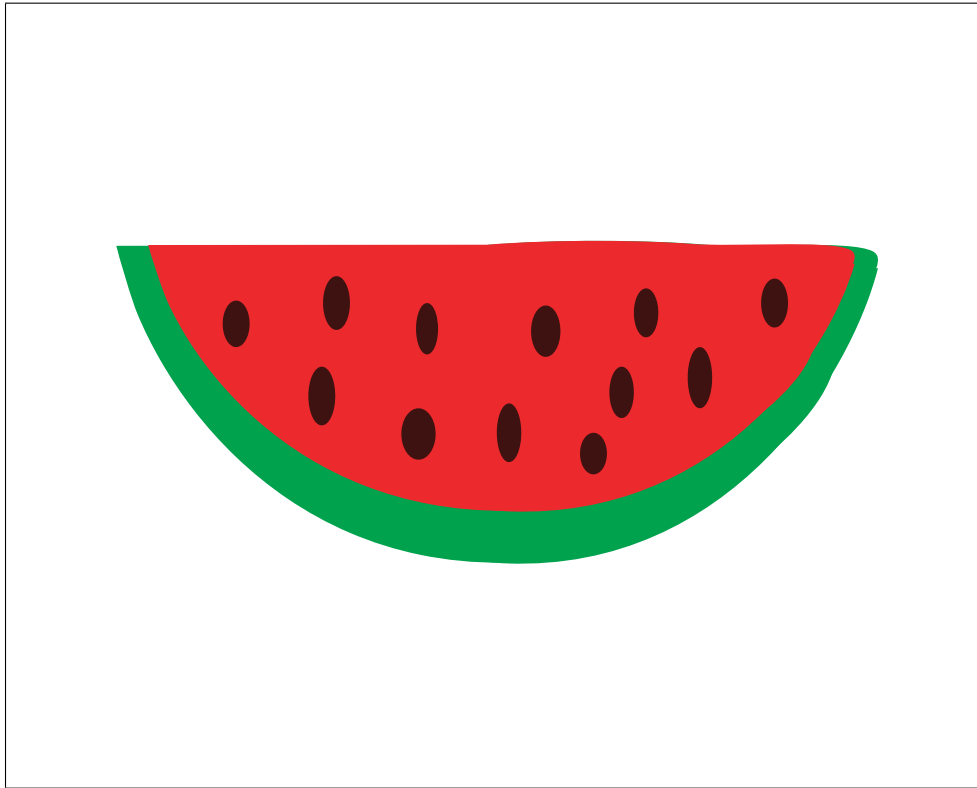


Figure A-9: Task me1on: “Make the black seeds white.” To complete this task, twelve objects must be selected. Using the pre-existing history, this could be accomplished with a two-step selection expansion.

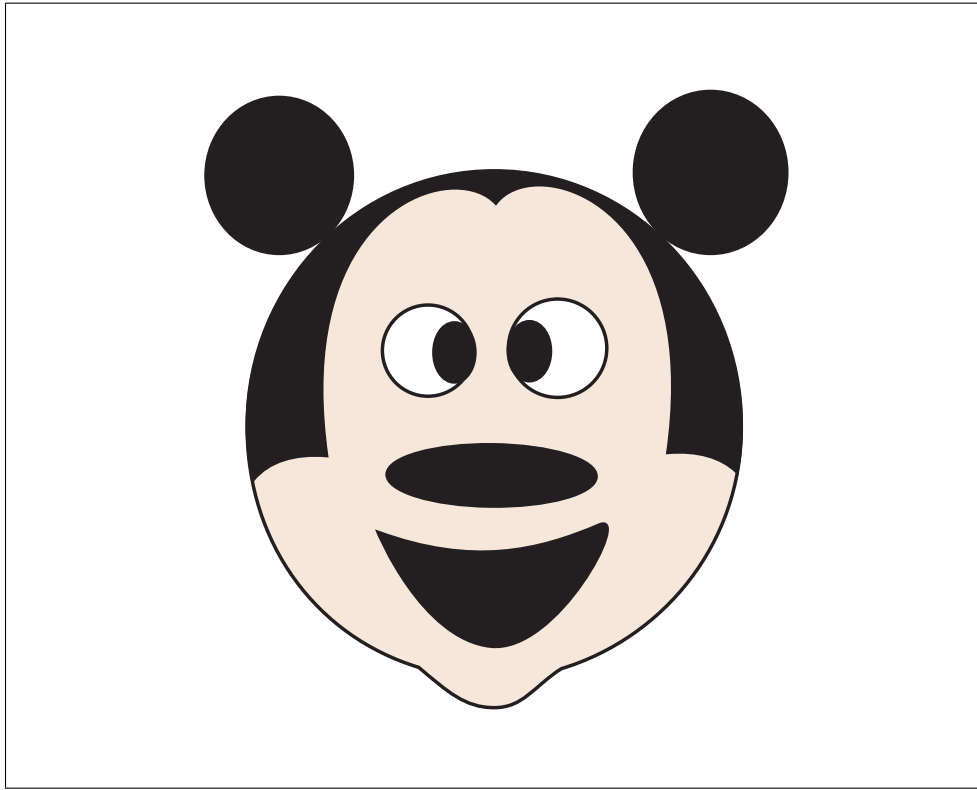


Figure A-10: Task mickey: “Make the centers of the eyes blue.” To complete this task, two objects must be selected. Using the pre-existing history, this could be accomplished with a two-step selection expansion.



Figure A-11: Task mobile: “Delete the first row of buttons (the un-numbered ones).” To complete this task, four objects must be selected. Using the pre-existing history, this could be accomplished with a three-step selection expansion.

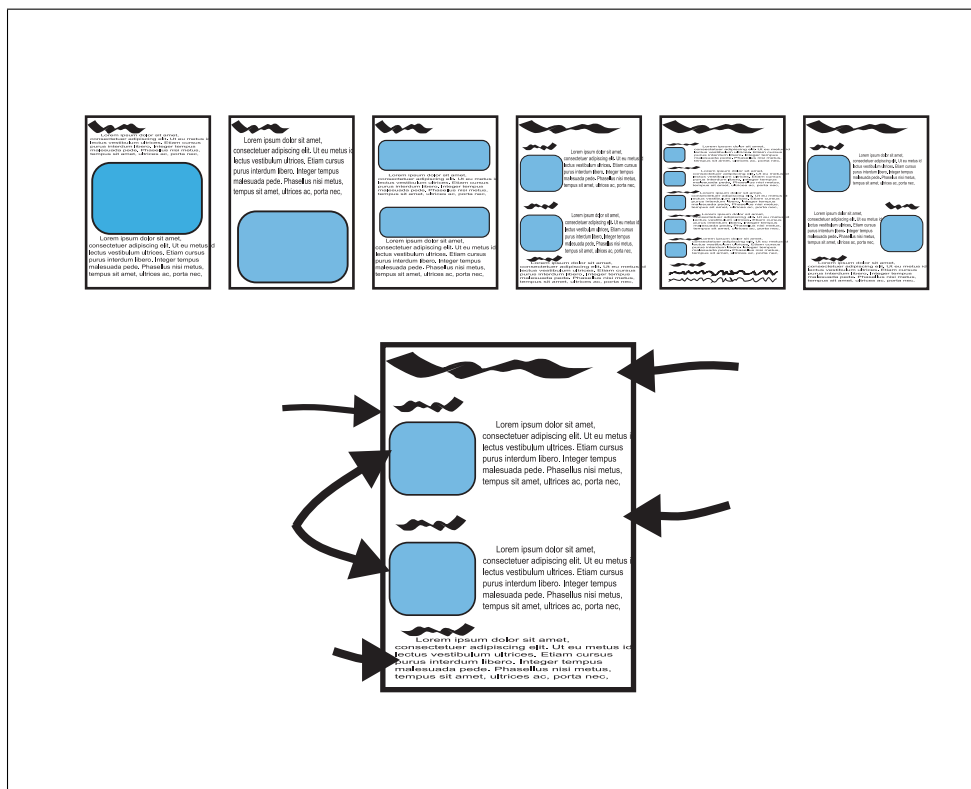


Figure A-12: Task page: “Delete all text on the pages.” To complete this task, twenty objects must be selected. Using the pre-existing history, this could be accomplished with a three-step selection expansion.

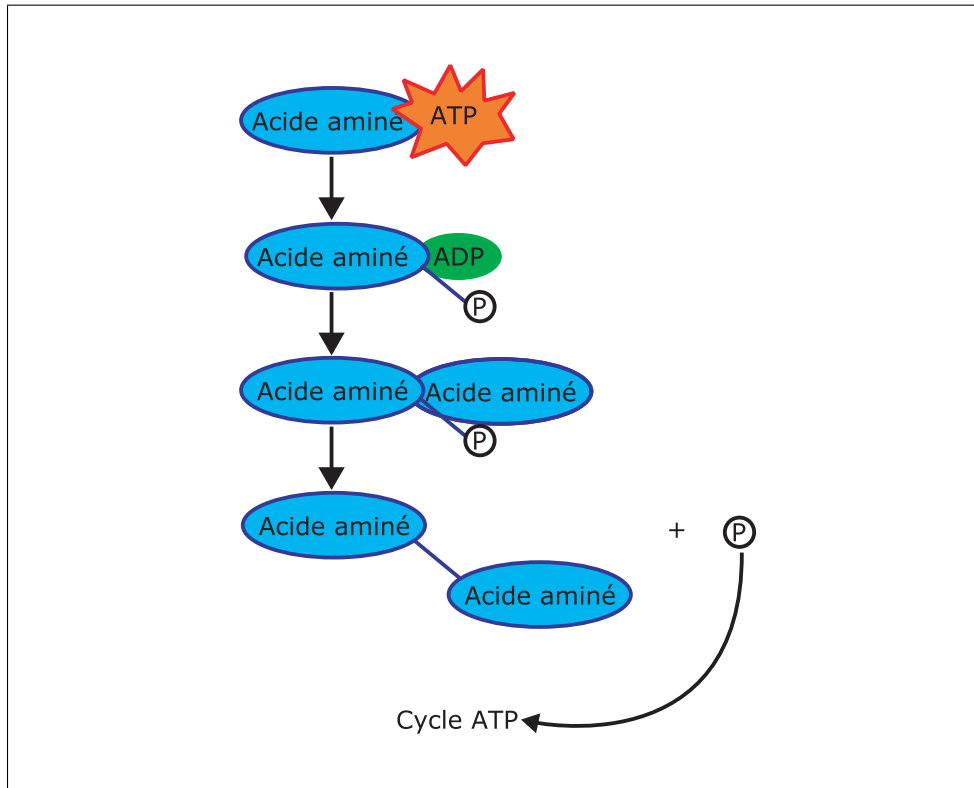


Figure A-13: Task protein: “Delete the blue ovals, including their text.” To complete this task, twelve objects must be selected. Using the pre-existing history, this could be accomplished with two three-step selection expansions.



Figure A-14: Task recyc1e: "Change the color of the entire PEBD (#4) symbol." To complete this task, three objects must be selected. Using the pre-existing history, this could be accomplished with a two-step selection expansion.

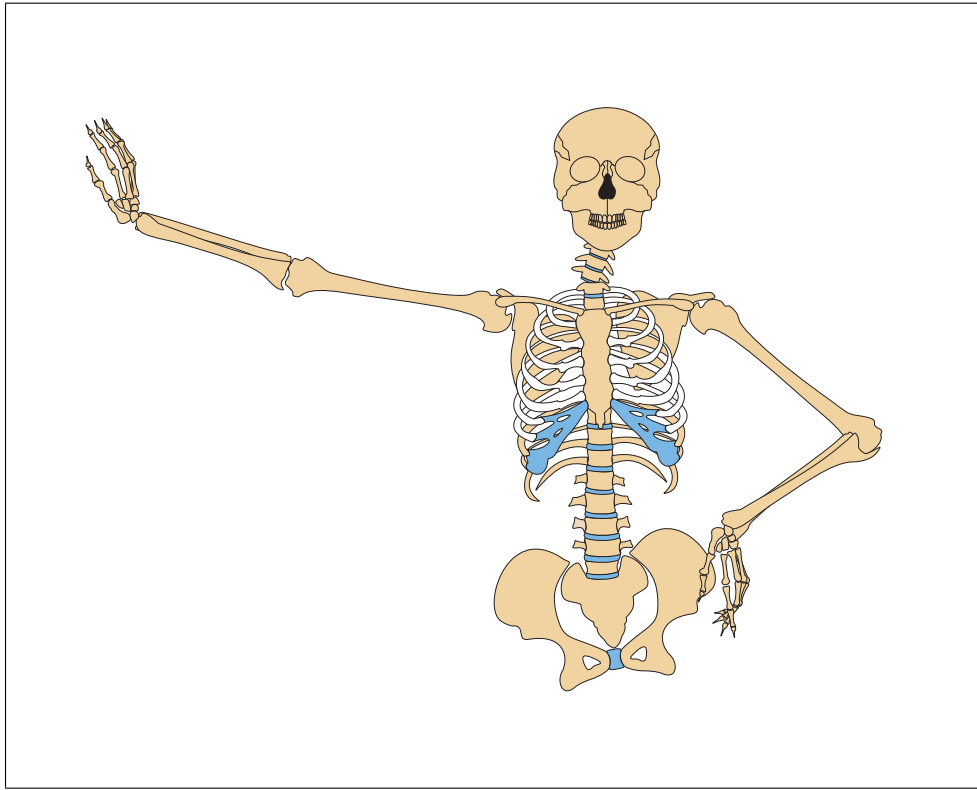


Figure A-15: Task `ribs`: “Color the white ribs blue.” To complete this task, fourteen objects must be selected. Using the pre-existing history, this could be accomplished with a three-step selection expansion.

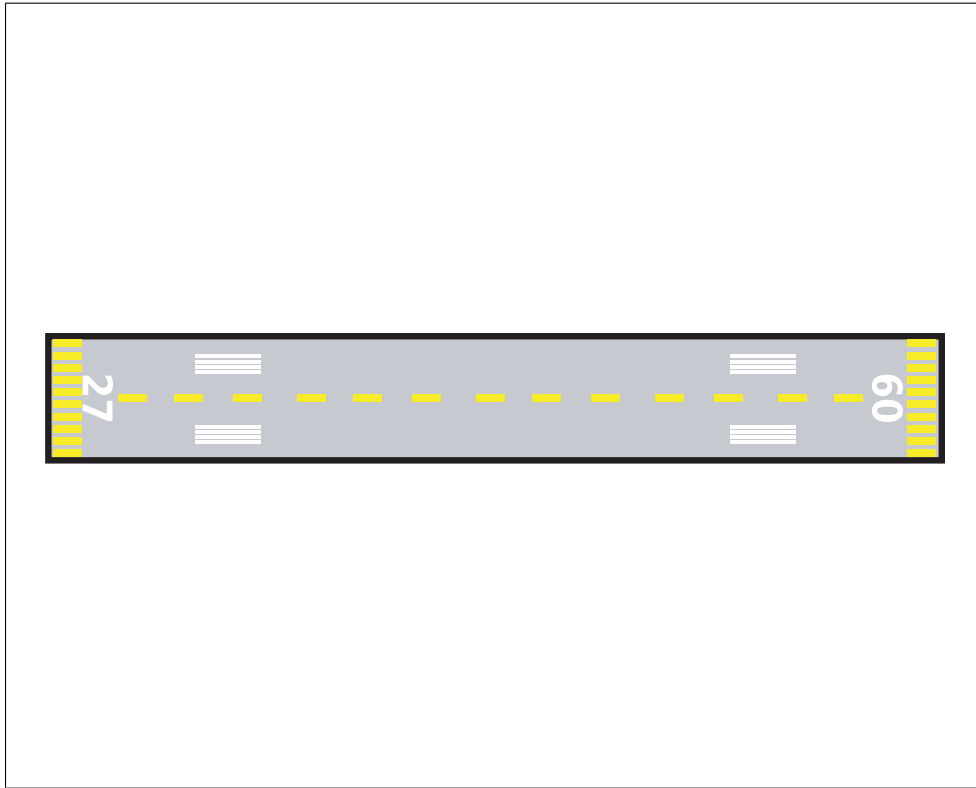


Figure A-16: Task runway: “Change all of the yellow lines to black.” To complete this task, thirty-three objects must be selected. Using the pre-existing history, this could be accomplished with a three-step selection expansion.

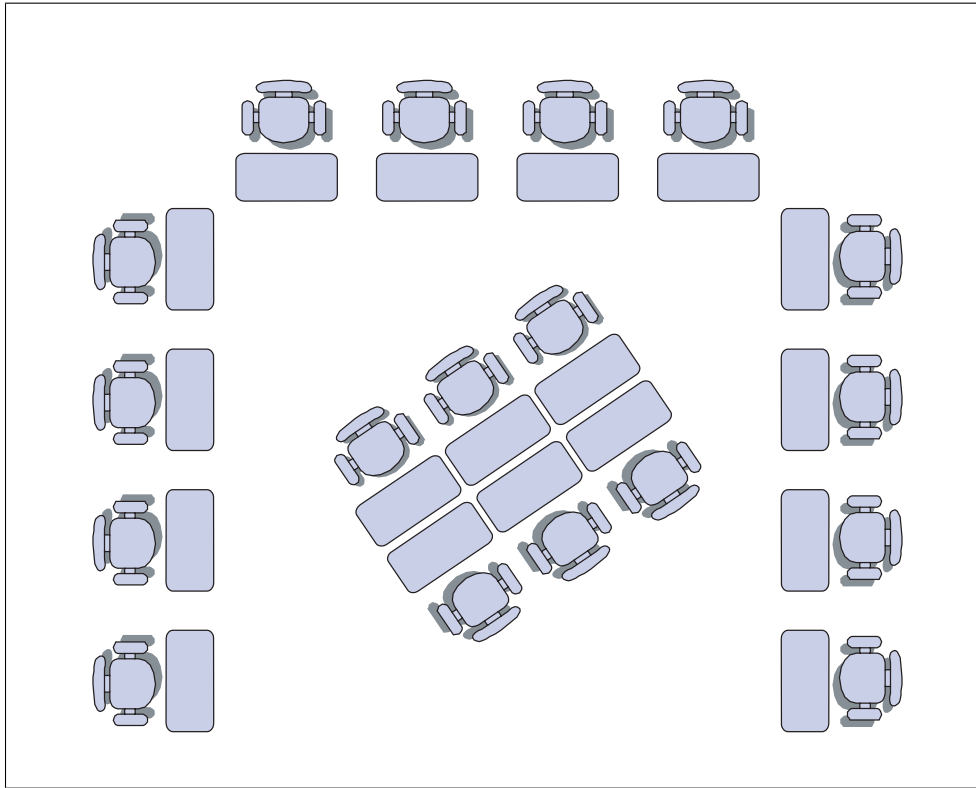


Figure A-17: Task seats: “Delete only the chairs at the center of the room.” To complete this task, twelve objects (the chairs and their shadows) must be selected. Using the pre-existing history, this could be accomplished with a four-step selection expansion.

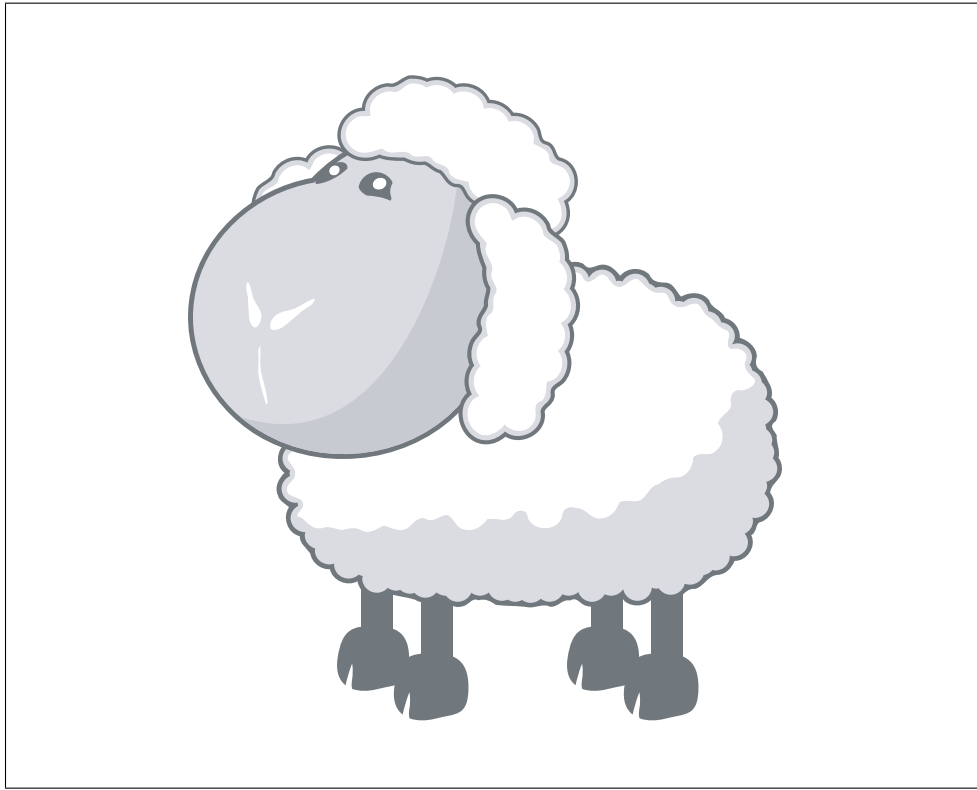


Figure A-18: Task sheep: “Change the color of the legs and feet.” To complete this task, eight objects must be selected. Using the pre-existing history, this could be accomplished with a four-step selection expansion.

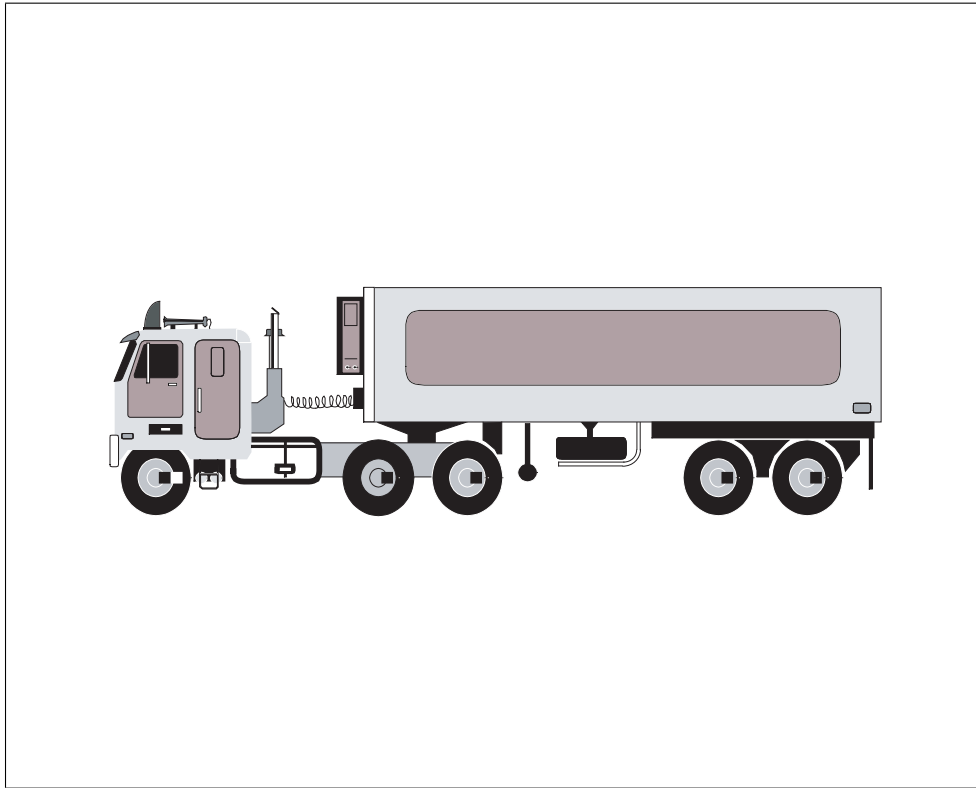


Figure A-19: Task truck: “Make the leftmost wheel green.” To complete this task, seven objects must be selected. Using the pre-existing history, this could be accomplished with a two-step selection expansion.

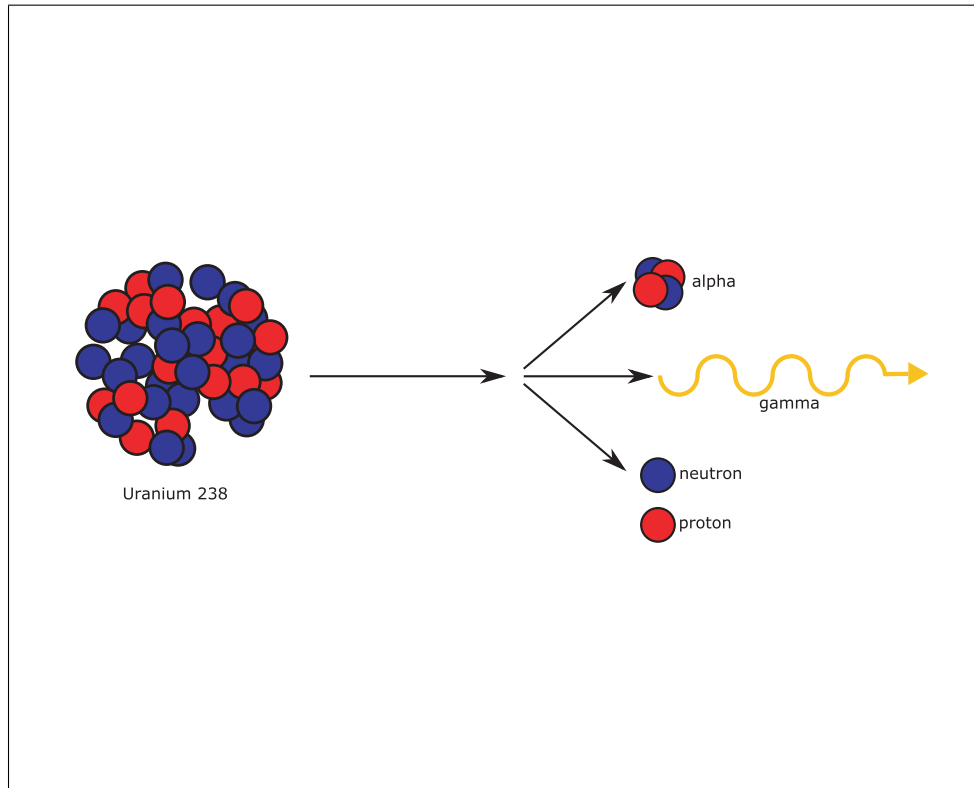


Figure A-20: Task uranium: "Delete all of the red balls." To complete this task, twenty objects must be selected. Using the pre-existing history, this could be accomplished with two two-step selection expansions, plus a single selection.

Bibliography

- ABOWD, G. D., AND DIX, A. J. 1992. Giving undo attention. *Interacting with Computers* 4, 3, 317–342.
- ADOBE SYSTEMS, 1988–. Illustrator. <http://adobe.com/products/illustrator>.
- ADOBE SYSTEMS, 1990–. Photoshop. <http://adobe.com/products/photoshop>.
- ADOBE SYSTEMS, 2005. Adobe Design Center Tutorials: How do I select just one kind of object from a design? Online article, July. <http://adobe.com/designcenter/illustrator/articles/illcs2qaarselect.html>.
- ADOBE SYSTEMS, 2006–. Acrobat Connect. <http://adobe.com/products/acrobatconnect>.
- AGRAWALA, M., AND STOLTE, C. 2001. Rendering effective route maps: Improving usability through generalization. In *Proceedings of ACM SIGGRAPH 2001, Computer Graphics Proceedings, Annual Conference Series*, 241–250.
- AGRAWALA, M., PHAN, D., HEISER, J., HAYMAKER, J., KLINGNER, J., HANRAHAN, P., AND TVERSKY, B. 2003. Designing effective step-by-step assembly instructions. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 22, 3, 828–837.
- APPLE, 2003–. Exposé. <http://www.apple.com/findouthow/mac/#tutorial=expose>.
- BAUDISCH, P., AND GUTWIN, C. 2004. Multiblending: displaying overlapping windows simultaneously without the drawbacks of alpha blending. In *CHI '04: Proceedings of the ACM Conference on Human Factors in Computing Systems*, 367–374.

- BAUDISCH, P., TAN, D., COLLOMB, M., ROBBINS, D., HINCKLEY, K., AGRAWALA, M., ZHAO, S., AND RAMOS, G. 2006. Phosphor: explaining transitions in the user interface using afterglow effects. In *UIST '06: Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, 169–178.
- BEAUDOUIN-LAFON, M. 2001. Novel interaction techniques for overlapping windows. In *UIST '01: Proceedings of the Annual ACM Symposium on User Interface Software and Technology*, 153–154.
- BEDERSON, B. B. 2000. Fisheye menus. In *UIST '00: Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology*, 217–226.
- BERLAGE, T. 1994. A selective undo mechanism for graphical user interfaces based on command objects. *ACM Transactions on Computer-Human Interaction* 1, 3, 269–294.
- BIER, E. A., STONE, M. C., PIER, K., BUXTON, W., AND DEROSE, T. D. 1993. Toolglass and magic lenses: the see-through interface. In *SIGGRAPH '93: Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, 73–80.
- BUSSAN, M., AND HALL, R. 1993. Abstraction, context and constraint. In *State of the Art in Computer Graphics*. Springer-Verlag.
- CICIRELLO, V. A., AND REGLI, W. C. 1999. Resolving non-uniqueness in design feature histories. In *SMA '99: Proceedings of the fifth ACM symposium on Solid modeling and applications*, 76–84.
- CICIRELLO, V. A., AND REGLI, W. C. 2000. Managing digital libraries for computer-aided design. *Computer Aided Design* 32, 2, 119–132.
- DONY, R. D., MATEER, J. W., ROBINSON, J. A., AND DAY, M. G. 2005. Iconic versus naturalistic motion cues in automated reverse storyboarding. In *Proceedings of 2nd IEEE European Conference on Visual Media Production*, 17–25.
- FINDLATER, L., AND MCGRENERE, J. 2004. A comparison of static, adaptive, and adaptable menus. In *CHI '04: Proceedings of the ACM Conference on Human Factors in Computing Systems*, 89–96.

- GOLDER, S. A., AND HUBERMAN, B. A. 2006. Usage patterns of collaborative tagging systems. *Journal of Information Science* 32, 2, 198–208.
- GOLDING, M. 2005. *Real World Adobe Illustrator CS2*. Peachpit Press.
- GOLDMAN, D. B., CURLESS, B., SALESIN, D., AND SEITZ, S. M. 2006. Schematic storyboarding for video visualization and editing. *ACM Transactions on Graphics* 25, 3, 862–871.
- GREENBERG, S., AND WITTEN, I. 1985. Adaptive personalized interfaces: A question of viability. *Behaviour & Information Technology* 4, 1, 31–45.
- HALL, R., BUSSAN, M., GEORGIADIS, P., AND GREENBERG, D. P. 1991. A testbed for architectural modeling. In *Proceedings of Eurographics '91*, 47–58.
- HART, J. 1999. *The Art of the Storyboard: Storyboarding for Film, TV, and Animation*. Focal Press.
- HEER, J., AGRAWALA, M., AND WILLET, W. 2008. Generalized selection via interactive query relaxation. In *CHI '08: Proceedings of the Twenty-Sixth ACM Conference on Human Factors in Computing Systems*, 959–968.
- HEER, J., MACKINLAY, J., STOLTE, C., AND AGRAWALA, M. 2008. Graphical histories for visualization: Supporting analysis, communication, and evaluation. *IEEE Transactions on Visualization and Computer Graphics* 14, 6 (Nov.-Dec.), 1189–1196.
- HEISER, J., PHAN, D., AGRAWALA, M., TVERSKY, B., AND HANRAHAN, P. 2004. Identification and validation of cognitive design principles for automated generation of assembly instructions. In *AVI '04: Proceedings of the Workshop on Advanced Visual Interfaces*, 311–319.
- HUDSON, S. E., RODENSTEIN, R., AND SMITH, I. 1997. Debugging lenses: a new class of transparent tools for user interface debugging. In *UIST '97: Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology*, 179–187.
- INKSCAPE. 2003–. <http://www.inkscape.org>.

- ISHAK, E. W., AND FEINER, S. K. 2004. Interacting with hidden content using content-aware free-space transparency. In *UIST '04: Proceedings of the Annual ACM Symposium on User Interface Software and Technology*, 189–192.
- KAWASAKI, Y., AND IGARASHI, T. 2004. Regional undo for spreadsheets. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*.
- KLEMMER, S. R., THOMSEN, M., PHELPS-GOODMAN, E., LEE, R., AND LANDAY, J. A. 2002. Where do web sites come from? capturing and interacting with design history. In *Proceedings of CHI 2002: ACM Conference on Human Factors in Computing Systems*.
- KURLANDER, D., AND FEINER, S. 1990. A visual language for browsing, undoing, and redoing graphical interface commands. *Visual Languages and Visual Programming*, 257–275.
- KURLANDER, D., AND FEINER, S. 1992. A history-based macro by example system. In *Proceedings of the 5th Annual ACM Symposium on User Interface Software and Technology*, 99–106.
- KURLANDER, D. J. 1993. *Graphical Editing by Example*. PhD thesis, Columbia University.
- LI, R., AND LI, D. 2003. A regional undo mechanism for text editing. In *Proceedings of the Workshop on Collaborative Editing Systems*.
- MARKS, J., ANDALMAN, B., BEARDSLEY, P., FREEMAN, W., GIBSON, S., HODGINS, J., KANG, T., MIRTICH, B., PFISTER, H., RUML, W., RYALL, K., SEIMS, J., AND SHIEBER, S. 1997. Design galleries: a general approach to setting parameters for computer graphics and animation. In *Proceedings of SIGGRAPH'97*, 389–400.
- MARLOW, C., NAAMAN, M., BOYD, D., AND DAVIS, M. 2006. HT06, tagging paper, taxonomy, Flickr, academic article, to read. In *HYPertext '06: Proceedings of the 17th Conference on Hypertext and Hypermedia*, 31–40.
- MCCLOUD, S. 1993. *Understanding Comics: The Invisible Art*. Harper Perennial.

- MENG, C., YASUE, M., IMAMIYA, A., AND MAO, X. 1998. Visualizing histories for selective undo and redo. In *Proceedings of the Third Asian Pacific Conference on Computer and Human Interaction*, 459–464.
- MICROSOFT, 1990–. PowerPoint. <http://office.microsoft.com/PowerPoint>.
- MILLER, R. C., AND MYERS, B. A. 2002. Multiple selections in smart text editing. In *IUI '02: Proceedings of the International Conference on Intelligent User Interfaces*, 103–110.
- MITCHELL, J., AND SHNEIDERMAN, B. 1989. Dynamic versus static menus: An exploratory comparison. *SIGCHI Bulletin* 20, 4, 33–37.
- NAKAMURA, T., AND IGARASHI, T. 2008. An application-independent system for visualizing user operation history. In *UIST '08: Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology*, 23–32.
- OLSEN, JR., D. R. 2007. Evaluating user interface systems research. In *UIST '07: Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, 251–258.
- OPEN CLIP ART LIBRARY. 2004–. <http://www.openclipart.org>.
- PALMER, S. E. 1999. *Vision Science: Photons to Phenomenology*. The MIT Press.
- PEDERSEN, E. R., AND McDONALD, D. W. 2008. Relating documents via user activity: the missing link. In *IUI '08: Proceedings of the 13th International Conference on Intelligent User Interfaces*, 389–392.
- PIROLI, P., AND CARD, S. K. 1999. Information foraging. *Psychological Review* 106, 643–675.
- RAMOS, G., ROBERTSON, G., CZERWINSKI, M., TAN, D., BAUDISCH, P., HINCKLEY, K., AND AGRAWALA, M. 2006. Tumble! Splat! Helping users access and manipulate occluded content in 2D drawings. In *AVI '06: Proceedings of the Workshop on Advanced Visual Interfaces*, 428–435.
- RENDERSOFT, 2001–. CamStudio. <http://www.camstudio.org>.
- RITTER, A., AND BASU, S. 2009. Learning to generalize for complex selection tasks. In *IUI '09: Proceedings of the Conference on Intelligent User Interfaces*, 167–176.

- SAUND, E., FLEET, D., LARNER, D., AND MAHONEY, J. 2003. Perceptually-supported image editing of text and graphics. In *UIST '03: Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*, 183–192.
- SCHMIDT, R., ISENBERG, T., JEPP, P., SINGH, K., AND WYVILL, B. 2007. Sketching, scaffolding, and inking: a visual history for interactive 3D modeling. *Proceedings of the 5th International Symposium on Non-Photorealistic Animation and Rendering*, 23–32.
- SCHWARZ, M., ISENBERG, T., MASON, K., AND CARPENDALE, S. 2007. Modeling with rendering primitives: an interactive non-photorealistic canvas. In *NPAR '07: Proceedings of the International Symposium on Non-Photorealistic Animation and Rendering*, 15–22.
- SEARS, A., AND SHNEIDERMAN, B. 1994. Split Menus: Effectively using selection frequency to organize menus. *ACM Transactions on Computer-Human Interaction* 1, 1, 27–51.
- SELIGMANN, D. D., AND FEINER, S. 1991. Automated generation of intent-based 3D illustrations. In *SIGGRAPH '91: Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques*.
- SOLIDWORKS CORPORATION, 1995–. SolidWorks. <http://www.solidworks.com>.
- SU, S. L., PARIS, S., AND DURAND, F. 2009. QuickSelect: History-based selection expansion. In *Proceedings of the 35th Annual Graphics Interface Conference*. In press.
- TERRY, M., MYNATT, E. D., NAKAKOJI, K., AND YAMAMOTO, Y. 2004. Variation in element and action: supporting simultaneous development of alternative solutions. In *CHI '04: ACM Conference on Human Factors in Computing Systems*, 711–718.
- THOMAS, C. G., AND KROGSÆTER, M. 1993. An adaptive environment for the user interface of Excel. In *IUI '93: Proceedings of the 1st International Conference on Intelligent User Interfaces*, 123–130.
- TUMMINELLO, W. 2005. *Exploring Storyboarding*. Thomson Delmar Learning.

- TVERSKY, B., ZACKS, J., LEE, P., AND HEISER, J. 2000. Lines, blobs, crosses, and arrows: Diagrammatic communication with schematic figures. In *Theory and Application of Diagrams*, M. Anderson, P. Cheng, and V. Haarslev, Eds. Springer, 221–230.
- TVERSKY, B., MORRISON, J. B., AND BETRANCOURT, M. 2002. Animation: can it facilitate? *International Journal of Human-Computer Studies* 57, 4, 247–262.
- TVERSKY, B., AGRAWALA, M., HEISER, J., LEE, P., HANRAHAN, P., PHAN, D., STOLTE, C., AND DANIEL, M.-P. 2006. *Applied Spatial Cognition From Research to Cognitive Technology*. Lawrence Erlbaum, ch. Cognitive Design Principles for Automated Generation of Visualizations.
- WINKEL, G. H. 1987. Implications of environmental context for validity assessments. In *Handbook of Environmental Psychology*, D. Stokols and I. Altman, Eds. Wiley, 71–97.
- ZACKS, J., TVERSKY, B., AND IYER, G. 2001. Perceiving, remembering and communicating structure in events. *Journal of Experimental Psychology: General* 136, 29–58.