QuickSelect: History-Based Selection Expansion

Sara L. Su
Massachusetts Institute of Technology
Computer Science & Artificial Intelligence Lab.
sarasu@csail.mit.edu

Sylvain Paris
Adobe Systems, Inc.
Creative Technologies Lab.
sparis@adobe.com

Frédo Durand
Massachusetts Institute of Technology
Computer Science & Artificial Intelligence Lab.
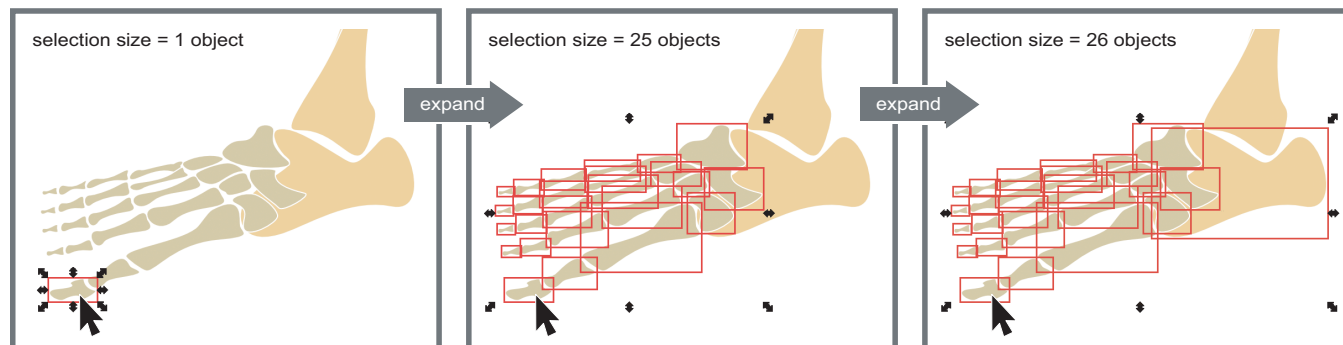fredo@csail.mit.edu

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

ABSTRACT

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. This approach 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.

Keywords: Selection, grouping, 2D drawing, operation history

Index Terms: H5.2 [Information interfaces and presentation]: User Interfaces—Graphical user interfaces. I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction Techniques

1 INTRODUCTION

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. They support multiple-item selections by several methods, typically selection by region (lasso or marquee tool) or shared attribute, or 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 paper, we describe a lightweight method for simplifying the reuse of complex selections. The core idea of our approach is to take the document 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 (§3.1). We demonstrate our approach in the context of 2D graphical editing (§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 conducted a pilot study. Feedback was generally positive and indicated that participants found this feature useful and easy to learn. When editing drawings with existing operation histories, users were faster at completing fixed tasks using the QuickSelect tool (§4.1).

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 (§4.2). We will discuss the strengths and limitations of selection expansion, as well as the potential for broader use of history to improve interaction (§5).

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.

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. For window management, Beaudouin-Lafon [2] proposes interaction techniques that extend a paper metaphor. Magic Lens filters [4] and content-aware free-space transparency [9] reveal hidden content in a portion of the screen.

Addressing 2D drawing, the Tumbler and Splatter tools [14] help users access occluded content by, respectively, cycling through layers of a document and "splatting" objects into an exploded view. Schwarz *et al.* [17] 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.

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 [3] and Flexcel [19]). Greenberg and Witten [6] and Mitchell and Shneiderman [11] have investigated the effects of dynamically organizing menu items based on frequency and recency of use. Sears and Shneiderman [18] 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 [5] 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.

2.3 Generalizing Selections

In their LAPIS text-editing system [10], Miller and Myers demonstrated "selection guessing", a method of inferring a multiple selection user-provided positive and negative examples. Ritter and Basu [15] 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.* [7] 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 [16]. However, while ScanScribe seeks to find perceptually significant structure in documents, our equivalent of structure is determined by user selection patterns.

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 first present the general algorithm before describing its implementation in a 2D graphical editor.

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 now consider an example expansion. We analyze the operation history of the document, the beginning of which is visualized below (Figure 2).

Figure 2: Visualization of the document's operation history.

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.

Suppose the query is (e). Below is an excerpt containing only operations affecting (e) (Figure 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.

Figure 3: 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 candidate objects to add to the set: d, f, and g (Figure 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.

Figure 4: Query `(e)` yields three candidate objects: `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 5). In both cases, the frequency is 2.

Figure 5: Query `(e,d)` yields two candidate objects: `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 6), so we have `(e,d,g,f)`.

Figure 6: 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 7).
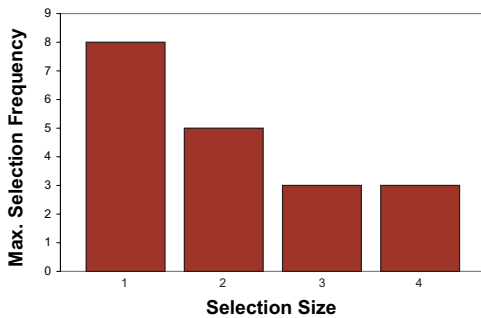

Figure 7: Query selection size vs. Maximum selection frequency.

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 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.

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 [8]. 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 paragraph. 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.

4 EVALUATION

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. 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 (§4.1). 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 (§4.2). The session concluded with an interview.

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 research participation mailing lists and received small
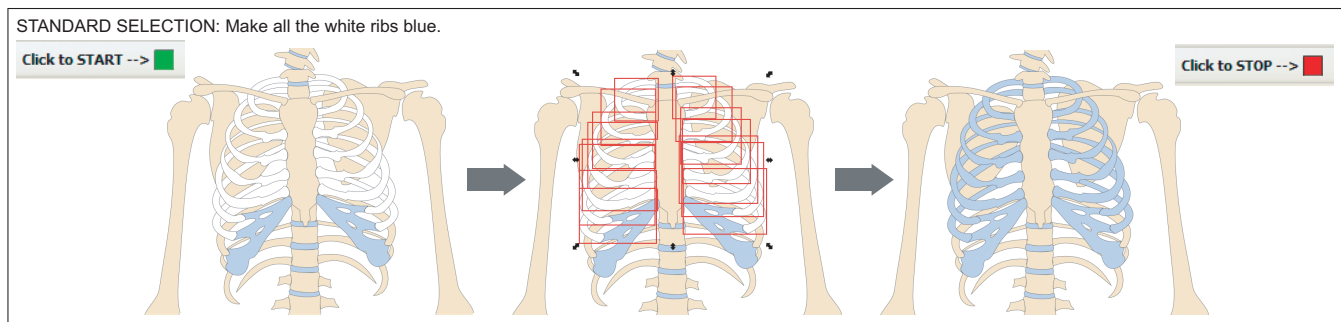
STANDARD SELECTION: Make all the white ribs blue.

Click to START -->

Click to STOP -->

Figure 8: Selection reuse with existing histories (Section 4.1). 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 article, they had the conventional appearance in the software used in the study.

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.

Apparatus

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.

4.1 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.

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.)

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 [12] and were edited by the authors. Examples are shown in Figures 8, 11, and 12. 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 9.

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).

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. 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. Completion time was automatically recorded for each trial, and editing errors were recorded by the observer. Each trial proceeded in the following steps, also illustrated in Figure 8:

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.

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. The observer recorded mistrials, *i.e.* when the subject clicked "Stop" before making all of the changes given in the objective. Mistrials were not included in the result timings.

Hypotheses

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

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 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.
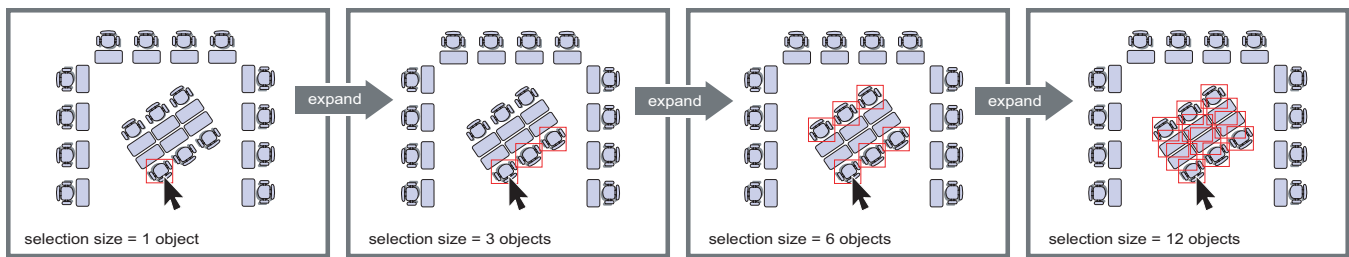
Figure 9: The `seats` task: "Delete only the chairs at the center of the room." We show the existing selections that can be reused for a particular object. Starting with one chair, the user can expand the selection to view previous selections. In this case, the given task could be completed with the set given after the second expansion. If the user expands too far, unwanted objects can be removed from the set by standard deselection.

4.2 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.

Design and 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, we introduced the mouse wheel interface to QuickSelect and asked participants 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. We 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 (§4.1).

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.

Results

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 informative comments 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 Flash."

5 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.

5.1 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 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

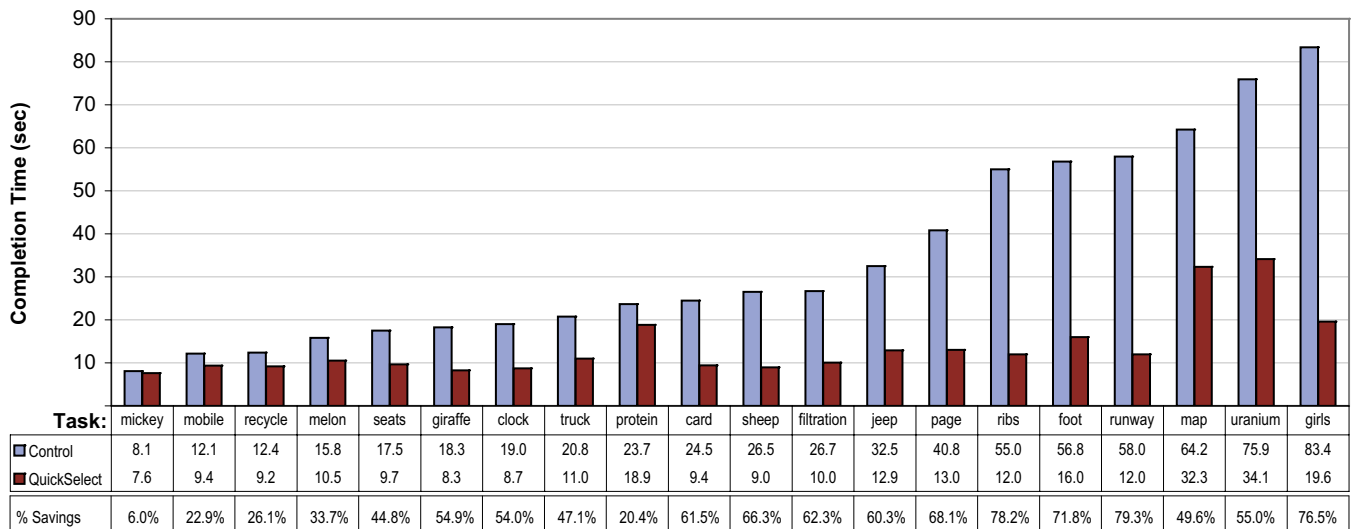| Task:	mickey	mobile	recycle	melon	seats	giraffe	clock	truck	protein	card	sheep	filtration	jeep	page	ribs	foot	runway	map	uranium	girls
☐ Control	8.1	12.1	12.4	15.8	17.5	18.3	19.0	20.8	23.7	24.5	26.5	26.7	32.5	40.8	55.0	56.8	58.0	64.2	75.9	83.4
☐ QuickSelect	7.6	9.4	9.2	10.5	9.7	8.3	8.7	11.0	18.9	9.4	9.0	10.0	12.9	13.0	12.0	16.0	12.0	32.3	34.1	19.6
% Savings	6.0%	22.9%	26.1%	33.7%	44.8%	54.9%	54.0%	47.1%	20.4%	61.5%	66.3%	62.3%	60.3%	68.1%	78.2%	71.8%	79.3%	49.6%	55.0%	76.5%

Figure 10: Performance comparison for selection reuse using existing histories (Section 4.1). Shown are the median task completion times for two interface conditions. Participants completed the selection authoring tasks faster using the QuickSelect interface.

measure of complexity. Figure 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 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.

5.2 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 12. In addition, some compositions are simple enough that multiple object selections are unnecessary.

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.

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" [1]). In the Inkscape

(a) uranium task

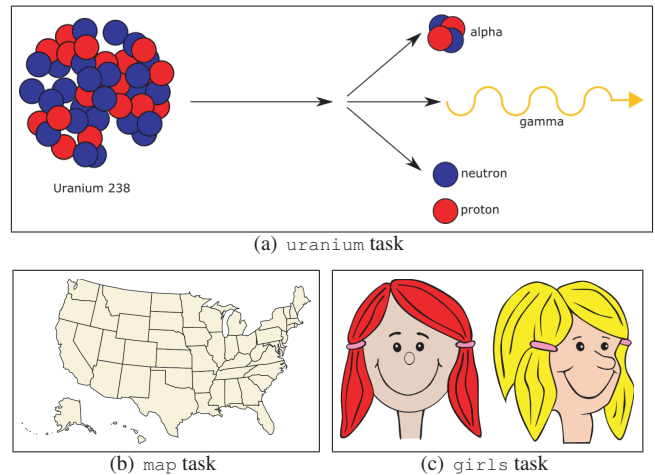
(b) map task

(c) girls task

Figure 11: Tasks for which QuickSelect noticeably increased efficiency. (a) "Delete all of the red balls." QuickSelect enabled users to re-select occluded balls. (b) "Color all states on the East coast green." Standard selection required more zooming to access differently-sized states. (c) "Change the hair of both girls to brown." The overlapping hair was easier to manipulate with QuickSelect.

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 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.

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 [13]). 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.

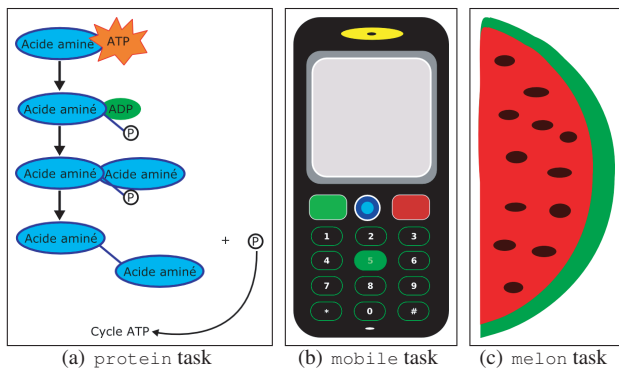(a) `protein` task   (b) `mobile` task   (c) `melon` task

Figure 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. (a) "Delete the blue ovals, including their text." (b) "Delete the first row of buttons (the un-numbered ones)." (c) "Make the black seeds white."

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.

Additional expansion heuristics: Another interesting direction for future research is incorporating additional user feedback into the selection expansion algorithm. The algorithm described in this paper 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.

6 CONCLUSIONS

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 [10] and file managers [15], 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 searching it becomes too expensive.

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 paper, 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.

REFERENCES

All of the URLs listed below are valid as March 13, 2009.

[1] Adobe Systems. Design Center Tutorials: How do I select just one kind of object from a design? http://adobe.com/designcenter/illustrator/articles/illcs2qaarselect.html.

[2] M. Beaudouin-Lafon. Novel interaction techniques for overlapping windows. In *UIST '01: Proceedings of ACM Symposium on User Interface Software and Technology*, pages 153–154, 2001.

[3] B. B. Bederson. Fisheye menus. In *UIST '00: Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 217–226, 2000.

[4] E. A. Bier, M. C. Stone, K. Pier, W. Buxton, and T. D. DeRose. Toolglass and magic lenses: the see-through interface. In *SIGGRAPH '93: Proceedings of the Conference on Computer Graphics and Interactive Techniques*, pages 73–80, 1993.

[5] L. Findlater and J. McGrenere. A comparison of static, adaptive, and adaptable menus. In *CHI '04: Proceedings of SIGCHI Conference on Human Factors in Computing Systems*, 2004.

[6] S. Greenberg and I. Witten. Adaptive personalized interfaces: A question of viability. *Behaviour & Information Technology*, 4(1):31–45, 1985.

[7] J. Heer, M. Agrawala, and W. Willett. Generalized selection via interactive query relaxation. In *CHI '08: Proceedings of SIGCHI Conference on Human Factors in Computing Systems*, pages 959–968, 2008.

[8] Inkscape. http://www.inkscape.org.

[9] E. W. Ishak and S. K. Feiner. Interacting with hidden content using content-aware free-space transparency. In *UIST '04: Proceedings of ACM Symposium on User Interface Software and Technology*, pages 189–192, 2004.

[10] R. C. Miller and B. A. Myers. Multiple selections in smart text editing. In *IUI '02: Proceedings of the International Conference on Intelligent User Interfaces*, pages 103–110, 2002.

[11] J. Mitchell and B. Shneiderman. Dynamic versus static menus: An exploratory comparison. *SIGCHI Bulletin*, 20(4):33–37, 1989.

[12] Open Clip Art Library. http://www.openclipart.org.

[13] P. Pirolli and S. K. Card. Information foraging. *Psychological Review*, 106:643–675, 1999.

[14] G. Ramos, G. Robertson, M. Czerwinski, D. Tan, P. Baudisch, K. Hinckley, and M. Agrawala. Tumble! Splat! Helping users access and manipulate occluded content in 2D drawings. In *Proceedings of the Conference on Advanced Visual Interfaces*, pages 428–435, 2006.

[15] A. Ritter and S. Basu. Learning to generalize for complex selection tasks. In *IUI '09: Proceedings of the Conference on Intelligent User Interfaces*, 2009.

[16] E. Saund, D. Fleet, D. Larner, and J. Mahoney. Perceptually-supported image editing of text and graphics. In *UIST '03: Proceedings of ACM Symposium on User Interface Software and Technology*, pages 183–192, 2003.

[17] M. Schwarz, T. Isenberg, K. Mason, and S. Carpendale. Modeling with rendering primitives: an interactive non-photorealistic canvas. In *NPAR '07: Proceedings of the International Symposium on Non-Photorealistic Animation and Rendering*, pages 15–22, 2007.

[18] A. Sears and B. Shneiderman. Split Menus: Effectively using selection frequency to organize menus. *ACM Transactions on Computer-Human Interaction*, 1(1):27–51, 1994.

[19] C. G. Thomas and M. Krogsæter. An adaptive environment for the user interface of Excel. In *Proceedings of the International Conference on Intelligent User Interfaces*, pages 123–130, 1993.