

Pins 'n' Touches: An Interface for Tagging and Editing Complex Groups

Sven Strothoff*, Wolfgang Stuerzlinger†, Klaus Hinrichs*

*Visualization and Computer Graphics Research Group, University of Münster, Germany

†School of Interactive Arts + Technology, Simon Fraser University, Vancouver, Canada

ABSTRACT

Influenced by mouse/pen-based user interfaces, most touch-based object tagging techniques rely mostly on a single interaction point. Once objects are tagged, typically only individual object inclusions/exclusions are possible. Yet, for tagging larger groups with complex spatial layouts, more refinement may be necessary to achieve the desired result. We apply visual tag markers to objects to visualize their group association. Through a new multi-touch pin gesture that “pins” one or more objects “down” while tagging, our new interface is capable of efficiently grouping objects, as identified in our user studies.

ACM Classification Keywords

H.5.2. Information Interfaces and Presentation (e.g. HCI): Input devices and strategies; Interaction styles.

Author Keywords

Multi-touch; Pin gesture; Complex group selection; Selection editing; Tagging.

INTRODUCTION

Selection of one or more objects is a fundamental operation in user interfaces, and usually done with the mouse or pen. Such interfaces rely on a single interaction point (cursor). Touch interfaces have (largely) adopted the single-touch convention, especially for tagging. Yet, this design does not leverage the unique capabilities of multi-touch, such as multiple simultaneous touches, which enable the user to effectively control multiple cursors. Here we explore the idea that using multiple touches enables more efficient tag editing by adding or removing *multiple* objects to/from existing groups. We first examine existing selection techniques, discuss the relationship between tagging and grouping, and present use cases for tagging of complex groups. Then we introduce our new multi-touch tagging technique and how it edits tags of complex groups. To evaluate our interaction technique we present two user studies comparing different aspects of our method to identify its performance characteristics.

Motivation

While examining existing multi-touch input methods, we noticed a lack of techniques that enable the user to tag complex

groups. Looking at the design space for such a technique, we developed the following questions:

- How can we support tagging of more than a single object at a time, which resembles a grouping task?
- For tagging with multiple concurrent tags, how is overlap, *i.e.*, objects belonging to more than one group, handled?
- How do we show that an object belongs to multiple groups?
- How can we use multiple touches to support efficient editing of multiple concurrent tags?

Related Work

We first compiled a list of references [2–5, 7, 10, 12, 14, 16, 17, 19, 21–23, 26] to survey existing work for group selection. Then, we categorized them by assigning classifiers based on input hardware, mode of operation and capabilities. For brevity, we list the classifiers directly with the references at the end. We used the following criteria:

MOUSE, PEN, TOUCH For which input method was the selection technique designed?

MULTI-TOUCH This classifier is only assigned if more than one simultaneous touch is used.

HOLD A hold gesture, *i.e.*, a stationary touch for a prolonged period of time, is used in addition to the normal tap gesture.

PIN A hold gesture, that is used in combination with other user actions.

RECTANGLE, LASSO If a rectangular region can be specified RECTANGLE was assigned. LASSO was assigned if an arbitrary region can be selected by lassoing around it.

IN/OUT Is there a difference between interaction that is performed/started inside or outside of an existing selection?

REGION, OBJECTS Is the selection technique used to select objects or a region of interest, such as a region in an image?

AUTO MODE Is there an automatic switch between selection and de-selection, *i.e.*, does the system support more than tapping objects to toggle selection.

ON/OFF OBJECTS Is there a differentiation between interactions/gestures that are performed/started on objects or in the surrounding space?

DIRECT, INDIRECT Are objects selected directly, *i.e.*, by tapping on them, or indirectly, *e.g.*, by surrounding them?

SURROUND, CROSSING Is the selection performed by surrounding the desired objects or by crossing them?

PATH Is the selection performed along a path? Or are only selection areas supported?

The overview of all pair-wise combinations of classifiers is shown in Figure 1. The numbers at the intersections indicate

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ITS 2015, November 15–18, 2015, Funchal/Madeira, Portugal.

Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM 978-1-4503-3899-8/15/11 ...\$15.00.

DOI: <http://dx.doi.org/10.1145/2817721.2817731>

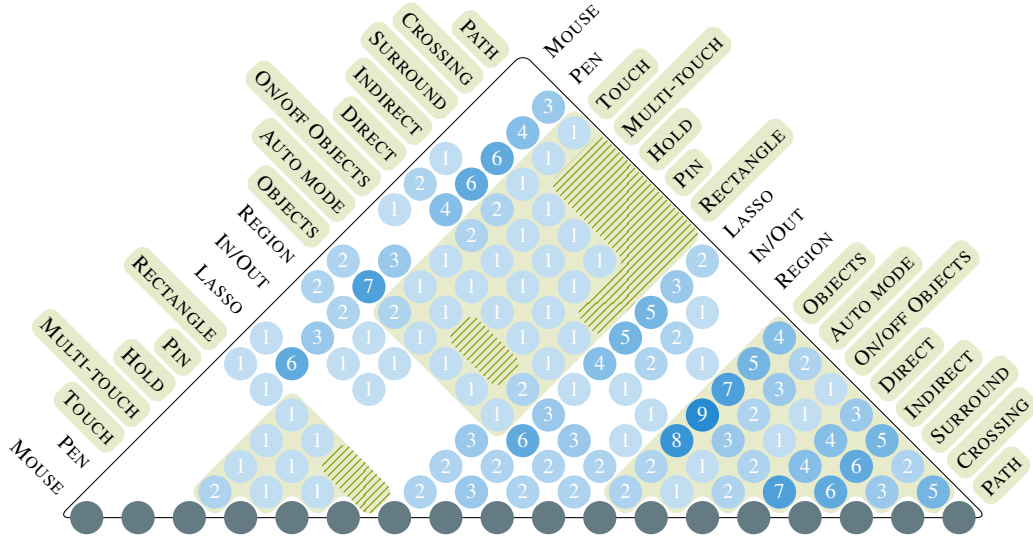


Figure 1. All pair-wise combinations of classifiers with their number of occurrences provide an overview of the examined body of previous work on group selection. The classification of our novel tagging interface (shaded in green) and the new contributions (hatched area) are also shown.

how often the corresponding combination is present in previous work. The classification of our novel tagging interface is shaded in green. Hatched areas represent new contributions.

Multi-touch interaction is often constrained to a single touch. There are two reasons for that. Most multi-touch systems are small hand-held devices. When one hand holds the device, only a single hand is available for interaction. Second, most user interface development was influenced by mouse- or pen-based interfaces, which implies a single cursor/touch. The only multi-touch interaction that is truly “multi” is common gestures, such as two finger scrolling or pinching. As large tablets and notebooks and desktop computers with multi-touch capable displays become available, true multi-touch input becomes more important. Moscovich et al. [18] argued that controlling two independent points of interaction using a single hand is difficult and have shown that two hands perform better than one for these tasks. Furthermore, a single moving touch can be augmented by additional stationary ones. These additional touches can be used to define contextual information for the action performed by the primary one. Strothoff et al. [23] used this idea to select rectangular regions.

Looking at the gaps in Figure 1 one can identify that true *multi-touch* interaction together with common group selection techniques like path or rectangular region has not been explored. The results of previous works suggest that there is a potential benefit in combining existing elements to close the gap with a multi-touch group selection technique. The combination of pin and hold gestures with other selection techniques yields new possibilities for editing of tags or selections. Wobbrock et al. [25] identified a gesture similar to our *pin* gesture, but did not explore further.

Contribution

We propose a novel combination of interaction techniques based on pinning touches that facilitates the definition of complex groups through tagging. With the classifiers used for

structuring related work we classify our technique as: TOUCH, MULTI-TOUCH, HOLD, PIN, RECTANGLE, OBJECTS, AUTO MODE, ON/OFF OBJECTS, DIRECT, INDIRECT, SURROUND, CROSSING, PATH and introduce the following contributions:

- A classification and visual summary of previous work on group selection;
- a multi-touch user interface technique for tagging of complex groups;
- support for multiple concurrent and overlapping groups;
- pin gestures for efficient editing of existing groups;
- an evaluation of our new techniques in two user studies.

Pinning Touches

In contrast to a tap, a hold gesture is a stationary touch for a prolonged period of time. We call our extension of the touch input vocabulary *pinning touches*, i.e., stationary touches that are held during *other* touch actions. These pinning touches *pin down* objects or user interface elements. Through pinning touches we can associate additional contextual information with standard touch actions, such as tapping, swiping, tracing paths, flicking and two finger swipe or pinch gestures. Thus, operations that typically require a mode switch can be performed directly with pinning touches.

Use Cases for Pinning

Currently, deleting images in a photo application usually requires a temporary switch to a delete mode and then tapping on individual images. With pinning touches the explicit mode switch is unnecessary. Images can be pinned down to specify them as context for the delete action: The user can touch and hold, i.e., pin, several images and then tap a button to delete them. Alternatively, the delete button can be pinned to implicitly activate a delete mode. All images that are then selected, by tapping or some other method, will be deleted. As two simultaneous touches (one of them on the delete button) are required, accidental deletion is unlikely. Similarly, text attributes can be applied by pinning down an attribute in text

processing, for instance to *paint* a bold font attribute over a passage. Multiple pinning touches can even apply multiple attributes concurrently. Also, pinning touches can copy styles. Then, pinning a word and painting over text changes the style to match that of the pinned word. Pinning touches also extend multi-touch gestures. Thus a scaling gesture (pinch) can be constrained to horizontal, vertical or uniform scaling by pinning down respective modifier buttons.

Moreover, pinning touches can also enable functionality similar to a context menu. When a context menu is invoked, the properties of the object at the cursor determine what actions to display in the menu. Similarly, pinning touches can also supply context to actions performed with the primary touch. Gutwin et al. [6] presented a command selection technique that combined a pinning touch for menu access with taps to select commands from this menu. Yet, no context was derived from the pinning touch.

Li et al. [13] identified that pressing a button with the non-dominant hand is effective for switching between modes in pen-based interfaces. But this is limited to a single pair of modes. Hinckley et al. [8] also used a button as a mode switch. They extended the number of accessible modes through a pop-up menu. With multi-touch, the menu itself can act as a collection of switches that can be held down (*pinned*) to select the desired mode. This combines the semantics of holding a button to temporarily switch modes with the choices a menu offers. Hinckley et al. [9] presented a combination of touch and pen-based input. They used touches similar to our pinning touches to augment pen-based actions.

Using Pinning and Tags to Define Groups

Combining objects into groups is a common user interface operation in graphic design or layout software. Usually objects are first selected and then a widget or a shortcut is used to group these objects. Through pinning touches, groups can be defined without a menu or shortcut. For this, a single object is first pinned down and then other objects to be grouped are selected. If the pinned object is already part of a group, the additional objects are added to its group. Note that a pinning touch *implicitly* selects the target group as part of the interaction. In photo management software grouping is also frequently used to structure and sort photos, into events or by subject. For example, a user wants to group all photos taken while visiting Madeira, while all photos of the user's dog are collected in another group. When the user brought his dog along to a trip to Madeira, these photos belong to more than one group. To represent such group associations, tags can be added to the objects. Then, all photos from Madeira share one tag and those of the user's dog another one.

In this paper we present an interaction technique that combines pinning touches and tags to enable the definition and editing of complex group associations.

Use Cases for Tag-based Groups

In recent versions of OS X multiple tags can be assigned to files to assist with categorization. As touch-sensitive laptops are becoming more common, there will be an increased need to select, tag and group files using touch gestures. Modern

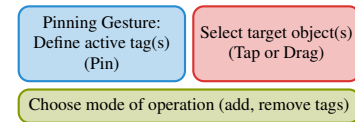


Figure 2. The three components of our interface: One technique to define a set of active tags and one to select target objects. We require another technique to choose the mode of operation, *i.e.*, adding or removing objects to the group.

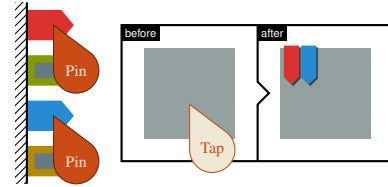


Figure 3. A set of tag widgets is displayed at the screen border. Pinning down widgets defines a set of tags for selection operations. Alternatively, the user can pin existing objects with desired tags, see Figure 4.

photo applications permit users to assign tags based on faces in a picture or the location the photo was taken at, supported by (semi-)automatic classification. Many e-mail clients support tagging of messages with different colors, and Gmail permits multiple tags per e-mail. All these cases represent multiple overlapping tag groups, such as an e-mail that is both *work related* and *important* or a photo of *my dog* taken on *Madeira*.

OUR PIN-BASED INTERFACE FOR TAGGING

We propose a new interaction technique, which enables tagging of objects. With it multiple tags can be assigned concurrently to each object, permitting objects to belong to multiple groups at the same time. The association is visualized through different colored tags. Once tags have been assigned, our technique enables easy modification of these tags. Three main components are necessary in our interface: A way to define a set of active tags, a technique for selecting target objects, and a way to specify the mode of operation: active tags can either be added to or removed from target objects (see Figure 2).

Defining a Set of Tags

Our tagging technique uses *pinning* touches. If the user touches and holds tagging widgets or already tagged objects while performing other actions, we call those *pinned* down. Pinned widgets/objects are used in our technique to define the set of active tags for subsequent operations.

Tag Widgets. We show a set of widgets, one for each available tag, at the border of the screen. Pinned widgets activate the corresponding tags, which are applied on selection. Through pinning multiple widgets the user can compose any combination of tags, as illustrated in Figure 3.

Pinning Already Tagged Objects. When the user pins down an object, this adds the tags assigned to this object to the set of active tags (cf. Figure 4). Pinning down multiple objects simultaneously activates the union of the objects' individual tags (cf. Figure 5). The tag widgets provide visual feedback of the active set. Assuming that similar objects are in close proximity, such as sequences of photos in a camera roll, applying the same set of tags to similar objects is then a *local* operation:

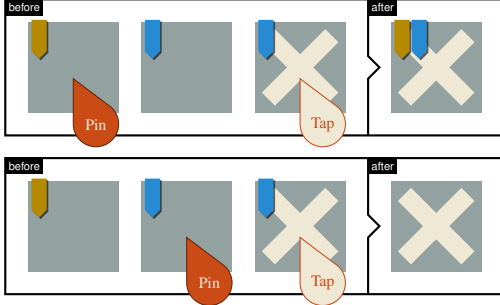
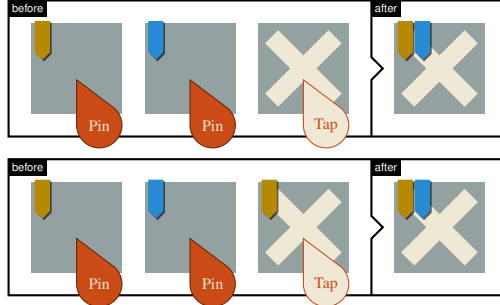
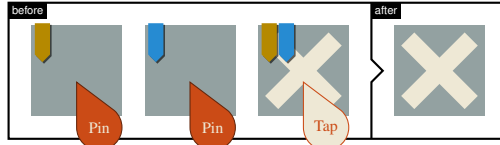


Figure 4. Pinning down a single tag: tapping an object toggles the tag on that object.



(a) If multiple objects are pinned down, the union of their tags is applied when tapping.



(b) Tags are removed if the tapped object already has all pinned tags.

Figure 5. Semantics for multiple pinned tags.

First, pin an already tagged object down and then select the other (close by) objects. For example, the user can then easily specify that *this is also* a photo of my dog taken on Madeira.

Selecting Target Objects

With a set of active tags, one or more target objects have to be chosen. Our tagging technique supports tapping for single and path-based operations for multiple objects.

Selecting Single Objects Through Tapping. Selecting single objects via tapping is common in touch interfaces, e.g., [12, 26]. If a single tag is pinned down with our technique, this tag is toggled on any (other) tapped object. In the top sequence in Figure 4, the yellow tag is pinned down. As the tapped target object does not yet have the yellow tag, it is added to it. In the bottom sequence the pinned blue tag is already assigned to the target object, so it is removed. When multiple tags are pinned, the result depends on the current tags of the tapped object. Figure 5(a) shows two sequences with a pinned set of yellow and blue tags. In both cases the tapped object has not been assigned both tags, so the pinned tags are added. If, on the other hand, the target object has already all active tags, as shown in Figure 5(b), the pinned tags are removed.

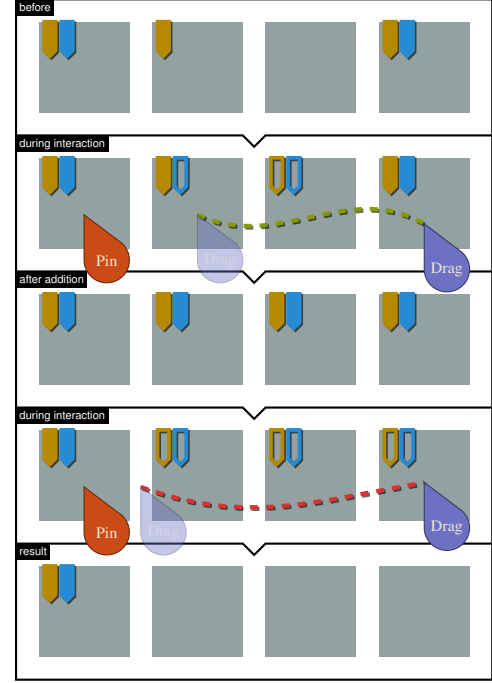


Figure 6. The starting point of the path decides the operation for the intersected objects. If the path begins inside an object tags are added (green path). If the path begins in the void (i.e., on the background, outside of all objects) tags are removed (red path). Outlined markers preview the changes that will be applied.

Pinning the void, i.e., the background area between objects, and tapping on an object removes all tags from said object.

Tagging Multiple Objects. The user can trace a path to operate on, i.e., tag, multiple targets. While interacting with single objects the user’s intent to add or remove a tag can be deduced automatically, using the approach outlined above. Yet, for multiple objects it is in general necessary to explicitly define the operation as either adding or removing tags. The operation is then applied to all objects intersecting the path. In our technique, the position of the first touch starting the path determines the mode. If the touch is inside an object, the current set of pinned tags will be added. Else, if the first touch is in the void, i.e., on the background, the pinned set of tags will be removed from the objects. See Figure 6 for illustrations. This extends previous work [23], which only supported selection of a single rectangular region.

Visual Feedback and Order of Operations

The color of the traced path shows the mode of operation: green for adding tags, red for removing them. While the path is traced, the system shows a preview of the resulting tag changes. Solid tags remain unchanged by the operation, while outlined tags illustrate changes. See Figure 6. The order of the three interaction components, shown in Figure 2, is not predefined and our technique supports all combinations. For instance, if the preview does not show the desired result, it is possible to change the set of pinned tags while tracing a path and see an updated preview. Even tagging target objects by first tracing a path and afterwards pinning tags down is possible, as long as the finger defining the path is not lifted.

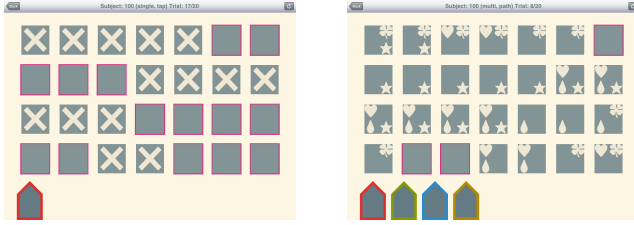


Figure 7. Screenshots of the two phases of the first study. (left) In the first phase all objects marked with an X had to be tagged. (right) In the second phase four different tags had to be assigned: heart → red, clover → green, drop → blue, star → yellow.

As long as pinning touches exist, they continue to define the set of active tags, which can be used for future operations.

USER STUDY 1: INITIALLY ASSIGNING TAGS

Initially, we contemplated an evaluation of pinning in isolation. Yet, the results of directly related work in pen-based interfaces [8, 9] convinced us that the outcome of such an experiment would be predictable. Therefore we decided to validate and evaluate our new multi-touch tagging technique directly in two user studies. In the first one we compared a variant of our technique that permits path gestures to tag multiple objects (PATH) with a more traditional single-touch tapping approach (TAP), where objects are tagged by tapping them one by one, as the control condition. We presented participants with a task similar to tagging photos in a photo application.

Participants and Setup. 21 participants (20 male, 1 female), ages 20 to 34 (median 28), participated in the study. All but two were students from the local university. All participants reported to have at least some experience with multi-touch devices. Most of them use smartphones or tablets regularly. The experiment, including instructions, training and debriefing, took about 45 minutes per participant. Breaks were permitted at any time. One participant had to be dropped due to a series of interruptions. The experiment was performed using an Apple iPad attached to a desk in front of the sitting participant.

Procedure. With each technique participants were asked to assign tags to split the presented objects into groups, based on their content. The order of techniques was counterbalanced using a 2×2 Latin square. After the experiment participants completed a short questionnaire.

Experimental Task. The study was divided into two phases. The first one used a single tag and participants were asked to tag all objects marked by an X (see Figure 7, left). The second phase used four tag categories: red, green, blue and yellow. Objects were marked with zero to four symbols representing the target tag set to be applied to each object, see Figure 7 (right). Tags had to be assigned as follows: heart → red, clover → green, drop → blue, star → yellow.

Initially, we contemplated using photos. However, the lack of a suitable photo set in the public domain led us to choose a different approach. Moreover, we did not want to deal with issues of users not being able to differentiate between similar images. Thus, we chose to simplify the task to a symbol

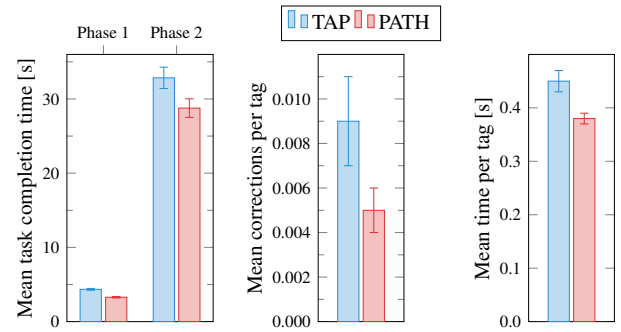


Figure 8. Comparison of TAP and PATH for: (left) mean task completion time in both phases of the first study, (center) mean number of corrections for each placed tag, (right) mean time for each placed tag.

recognition task, also because we are only interested in analyzing pinning and tagging performance. We first randomized the symbol positions on the objects to more closely resemble features in a photo. However, a pilot study revealed that the cognitive load for this was too high and resulted in both very high selection variability and overall slow performance. As we are looking for reliable measurements of selection task performance we needed to eliminate the potential confound of image recognition. Therefore, we simplified the task and assigned symbols to fixed locations on the objects.

In both phases no tags were assigned initially. When all tags were correctly assigned, the trial was automatically completed. After a notification, the experiment automatically advanced to the next trial. A colored border was added to objects that had all tags correctly assigned. This helped participants identify single missing tags towards the end of each trial.

Experiment Design. In each trial a sequence of 28 objects (7×4) was displayed. For each phase 10 object sequences were generated and shown twice to the participants, resulting in a total of 80 trials per participant (2 phases, 2 techniques, 10 sequences, 2 repetitions). The order of sequences was randomized for each user and technique. To minimize the effect of switching selection techniques, four training trials were added before each trial set, but discarded from analysis.

Photo organization applications usually display photos in the order they were taken, e.g., in a camera roll. This results in sequences of similar photos that would require similar tags. Therefore, we generated symbol sequences using an exponential distribution with an expected sequence length of 3 ($\lambda = 1/3$) for the trials in our study. For the second phase, sequences of each symbol were generated independently of each other, resulting in zero to four symbols per object.

Results

To analyze the data of the experiment we performed a 2×2 within-subjects, repeated measures ANOVA ($\alpha = .05$) for the used selection technique and the presented sequence. The dependent measure was the task completion time, measured from the first touch in each trial until all tags were correctly assigned. The results are summarized in Figure 8.

Task Completion Time. For the first phase (single tag) a significant main effect of technique, $F(1, 19) = 100.79, p < .001$, was found, with mean times of 4.35 s (SE = 0.12) for TAP and 3.28 s for PATH (SE = 0.10). A significant main effect of the presented sequence, $F(9, 171) = 70.56, p < .001$, was found, with mean times ranging from 2.25 s to 5.12 s. There was a significant interaction of technique and sequence, $F(4.53, 86.24) = 3.62, p = .007$. For the second phase (up to four tags) a significant main effect of technique, $F(1, 19) = 6.93, p = .016$, was found, with mean times of 32.84 s (SE = 1.44) for TAP and 28.77 s for PATH (SE = 1.25). A significant main effect of the presented sequence, $F(9, 171) = 20.42, p < .001$, was found, with mean times ranging from 24.40 s to 36.17 s. There was no interaction of technique and sequence, $F(4.57, 86.78) < 1$. Post-hoc analysis for both phases showed that the differences in completion time were caused by the sequences that required the least/most tags to be completed. We observed a linear trend between the number of target tags and the completion time: task completion time = $-3.41 + (0.59 \times \text{number of target tags})$, $R^2 = 0.8$.

Error Rate. To measure the error rate, we looked at the number of corrections a user performed in relation to the total number of tags to be placed. There are two types of such *errors*: Assigning a wrong tag to an object and removing a tag that was correctly assigned. We combined the error rates for both phases and performed a 2×2 within-subjects, repeated measures ANOVA with experiment phase and technique as factors. A significant main effect of phase, $F(1, 19) = 48.33, p < .001$ was found, with mean error rates of 0.002 (SE = 0.001) for the first and 0.012 (SE = 0.002) for the second phase. Overall, a significant main effect of selection technique, $F(1, 19) = 5.97, p = .024$ was found, with mean error rates of 0.009 (SE = 0.002) for TAP and 0.005 (SE = 0.001) for PATH. There was no interaction of phase and technique, $F(1, 19) = 1.74, p = .203$.

Time per Tag. To further investigate how the two techniques compare to each other and to illuminate the difference between the two phases, we looked at the mean time to place a tag. We calculated the time per tag for each trial by dividing the task completion time by the number of tags that had to be assigned to complete the sequence. Any wrong tags that had to be removed again were not counted, thus yielding the mean time it took to correctly place a tag. To analyze the time per tag we performed an ANOVA of the combined values of both phases. A significant main effect of phase, $F(1, 19) = 261.77, p < .001$, was found, with mean times of 0.29 s (SE = 0.01) for the first phase and 0.55 s (SE = 0.02) for the second phase. A significant main effect of selection technique, $F(1, 19) = 22.41, p < .001$, was found, with mean times of 0.45 s (SE = 0.02) for TAP and 0.38 s (SE = 0.01) for PATH. There was no interaction of phase and technique, $F(1, 19) < 1$.

Observations. Overall, users had no problems understanding and using the pinning touches in the tagging interface. While it was possible to assign multiple tags at the same time and participants were shown how to do so, we noticed that after the first few operations during the training virtually all tags were placed one after another. We also observed a change in

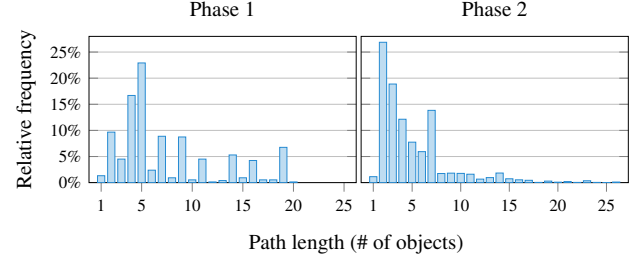


Figure 9. Histogram of path lengths, i.e., the number of affected objects, for both phases of the experiment. A clear shift towards shorter paths is visible for the second, more complex, phase.

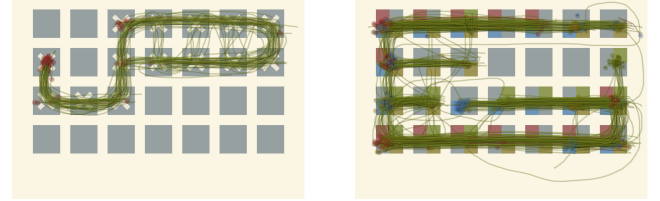


Figure 10. Illustrations overlaying paths of representative tag operations for each phase of the experiment. (left) In the first phase longer, curved paths were used. (right) The paths of the second phase were shorter and show a distinct *scanline* pattern.

the usage of the path gesture between the two phases of the experiment. Thus we took a closer look at the recorded paths. Figure 9 shows the distribution of path lengths, measured as the number of affected objects. In the first phase a length of five was most frequent, but considerably longer paths occurred as well. For the second phase the peak shifted towards a path length of 2, followed by a steep fall-off. An overlay of all recorded paths for a representative sequence of each phase is shown in Figure 10. In the first phase, many long winding paths were used, in most cases to tag each connected component of marked objects via a single path gesture. During the second phase, most tags were applied in a *scanline* pattern. Rows of objects were considered one after another, resulting in predominantly horizontal linear paths, as in Figure 10 (right). Users selecting a whole row caused the secondary peak for paths of length 7 in phase two.

Subjective Ratings. After the experiment participants were asked which tagging technique they perceived as faster. All but two thought they were faster with the path technique. The remaining rated both techniques as equal. Additionally, participants were asked for their preferred technique. All but one preferred the path technique to tapping. Most participants found individually tapping objects one by one to be tedious. One participant preferred tapping and stated that path had a higher risk of selecting more objects than intended. However, that participant also stated that the path technique would still be faster overall. Some participants that started the study with tapping inquired if there was a way to affect multiple objects at once—some even proposing tracing a path through them.

USER STUDY 2: MODIFYING TAGS

The first user study showed that the path-based selection technique is well suited for assigning tags to an untagged set of objects. Although the participants were introduced to the

mechanism for addition or removal of tags, they rarely used it, likely because it was only needed for the removal of individual accidentally placed tags. Thus, the first study yielded no insight on how users respond to the mechanism for adding or removing tags. Our second study forced participants to use both add and removal operations.

Participants and Setup. 16 participants (15 male, 1 female), ages 26 to 38 (median 29), participated in the study. All but one were students recruited from the local university department. All participants reported to have at least some experience with multi-touch devices. Most of them use smartphones or tablets on a regular basis. Participants took about 20 minutes to complete the study and were permitted breaks. The setup was identical to the first user study.

Procedure. Sequences of objects with already assigned tags were presented to the users. With each technique participants were asked to correct the tag assignment, so that only the marked objects were tagged—adding and removing tags as necessary. The order of techniques was counterbalanced using a 2×2 Latin square. After the experiment participants were asked to complete a short questionnaire to provide their subjective ratings and preference of both techniques.

Experimental Task. The experimental task was similar to phase one of the first study. However, this time there were already some tags assigned. Only a single tag category was available. Participants were asked to edit the presented tags, so that only the objects marked by an X were tagged. In contrast to the first study, users not only had to add but also to remove wrongly assigned tags. When all tags were correct, the trial was automatically completed and the experiment advanced to the next trial. To help locating the final missing or misplaced tags a colored border was added to correctly tagged objects.

Experiment Design. In each trial a sequence of 28 objects (7×4) was displayed. Ten object sequences were generated and shown twice to each participant, for a total of 40 trials (2 techniques, 10 sequences, 2 repetitions). The order of sequences was randomized for each user. Four training trials were added before the trials and discarded from analysis. The sequences were generated as in the first study. For each object sequence a sequence of tags was independently generated using the same distribution. Layering the tag and object sequences resulted in groups of correct and wrong tag assignments. In a first pilot for this experiment tags were assigned completely randomly. While randomly distributed errors may occur, we aimed to simulate a pre-tagged data set with some kind of systematic issue. For instance, if a face detection algorithm fails for one image, it is likely that it will also fail for very similar images in a sequence.

Results

The collected data was analyzed using a within-subjects, repeated measures ANOVA ($\alpha = .05$) using technique and the presented sequence as factors. If the assumption of sphericity was violated degrees of freedom were adjusted via Greenhouse-Geisser. The results are summarized in Figure 11.

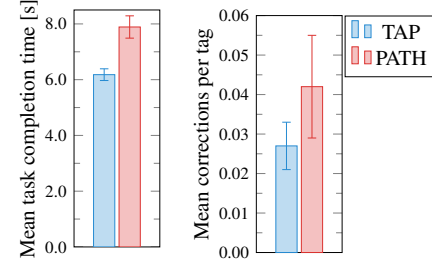


Figure 11. Comparison of TAP and PATH for: (left) mean task completion time, (right) mean number of corrections for each changed tag.

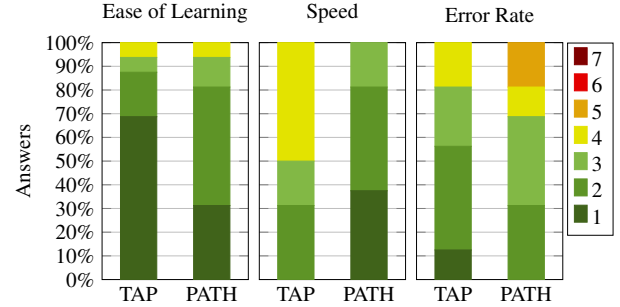


Figure 12. Ratings on ease of learning, speed and error rate based on a 7-point Likert scale (1 = very good).

Task Completion Time. The task completion time for each trial was measured from the first touch until the trial was completed successfully. We were able to find significant main effects for technique, $F(1, 15) = 40.19, p < .001$, and presented sequence, $F(9, 135) = 29.41, p < .001$. The mean times were 6.18 s (SE = 0.21) for TAP and 7.89 s (SE = 0.40) for PATH. As in the first study, post-hoc analysis showed that the significant differences in completion time were caused by the sequences that required the least/most tags to be changed. There was a significant interaction of technique and sequence $F(9, 135) = 6.59, p < .001$.

Number of Corrections per Tag. We analyzed the number of superfluous tag changes a user performed in relation to the minimum number of tag changes required to complete a trial, *i.e.*, the sum of tags that had to be added and removed. There was no effect for technique, $F(1, 15) = 1.334, p = .266$, nor for sequence, $F(3.76, 56.35) = 1.64, p = .180$ and no interaction of technique and sequence $F(3.29, 48.72) = 1.40, p = .252$.

Subjective Ratings. After each set of trials participants were asked to rate each technique concerning ease of learning, speed and error rate. Each category was rated using a 7-point Likert scale (1 = very good to 7 = poor). The results are summarized in Figure 12. After the trials, participants were asked which technique they preferred for each category and their overall preference. The results are shown in Figure 13. The collected data suggests that the participants' overall preference was influenced mostly by the perceived speed. All participants that preferred PATH for its speed also preferred it overall.

DISCUSSION

While the technique was not universally rated as *very* easy to learn, users had no problems understanding and using pinning

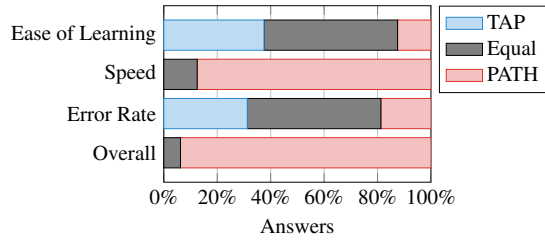


Figure 13. After completing the experiment participants were asked to state their preferred techniques regarding ease of learning, speed, error rate and their overall preference.

touches in the experiments, which matched our expectations and previous work [8, 9]. This supports the utility of pinning touches in multi-touch interfaces.

The results of the first study demonstrate that tracing a path through groups of objects was faster than tapping each object individually. This is not unexpected [1, 15], but current tagging techniques rarely support this. While the second phase of the first study required four times more tags on average, users took nearly ten times longer due to the increased task complexity. Although there is a significant difference in completion times between both techniques, the time is also dependent on the number of tags that needed to be assigned to complete a trial. To understand this interaction, we analyzed the time and corrections required for each correctly placed tag in both phases. PATH was significantly faster than TAP, by about 15%. Using a path gesture to tag multiple objects is more complex than tapping each single object individually, as multiple objects have to be considered while planning the gesture. Our results show that the benefit of affecting multiple objects outweighs the seemingly higher complexity of the path gesture. This is very promising as it opens the way for more efficient touch interaction compared to tapping.

In the second phase of the first study it took users nearly twice as long to correctly assign a tag. The mean error rate was six times higher than in the first one, which is expected due to the higher task difficulty. Still, the error rate remained surprisingly low for the large amount of fast tagging operations: one error per 83 modifications. Although the time for tagging and the error rate increased when dealing with multiple tags, we believe that the results show that our interface still works well. Some users modified multiple tags in final corrections at the end of a trial. We believe that this is due to the nature of our experimental task, which started with an *empty* object sequence. Consequently, it was somewhat easier to pin one category and finish setting all tags before moving to another tag, compared to constantly switching the set of active tags. The increased complexity of the second phase likely caused users to consider one row after another, which was easier than identifying larger connected components. The resulting *scan-line* pattern is illustrated in Figure 10 and also explains the shorter paths observed during the second phase (cf. Figure 9).

The first study provided little insight into our method for changing the mode of operation, *i.e.*, if tags are to be added or removed, which motivated our second study. After a short explanation of the principle and a bit of experimentation during training, participants quickly grasped the concept and were

able to complete the task successfully. Yet, we were surprised to find that, in contrast to the first study, individual tapping was faster. After a closer look at the sequences generated for the experiment, these results were not unexpected. Baudisch [1] identified that several objects need to be manipulated in one mouse drag to see a speed-up over single mouse clicks. The same insight holds true here. In our task, the sequences of consecutive objects with missing or misplaced tags were often no longer than three objects. For such short paths there is no speed benefit for path selection. Even though the quantitative measurements show that tapping was significantly faster, all but two users *rated* the path technique as faster. Users reported that they perceived tapping objects individually as tedious. Thus tapping many objects may seem to take longer than a few path gestures. The qualitative results further show that tapping was rated as both easier to learn and less error-prone. Yet, the perceived speed appears to dominate the overall rating.

The measured number of corrections should not be interpreted as an error rate. It is a combination of accidental errors and deliberately superfluous actions. With the path technique a few users added more tags than necessary and later removed them again. During debriefing three participants highlighted that they did this to chain smaller actions together, by cleverly combining add and remove operations. This behavior was not observed during the first study.

Kin et al. [11] have shown that experienced users can perform better by using more than one finger and tapping multiple objects at once. While our implementation supported this, during the experiment all users performed taps and pins with a single finger. Some users discovered the option to tap multiple objects at once, but reverted back to single tapping for the study. Although we only tested our interface on objects in a regular grid layout, previous work [3] suggests that path-based selection should also work for node networks with non-rectangular layouts. In our studies, users relied primarily on the tagging widgets to complete the task. Although pinning down (multiple) objects was always enabled, it was not used much. We plan to evaluate the object pinning aspect and the rectangle selection variant in more detail in the future.

DESIGN CONSIDERATIONS

Here we mention several design considerations for integrating our new technique into applications. It is not possible to present a single best variation of our technique, which fits all scenarios and devices. Thus, we present variations for different components of our interaction technique. However, all variants use the same underlying principle. The two main components, *i.e.*, defining a set of tags and selecting target objects, are independent of each other. This provides the flexibility to pick the best match to the target application and device combination. Moreover, object layout, *e.g.*, grid vs. random, also influences the choice of the “best” method to select multiple objects as well. We also discuss the need to adapt to different devices: not only for different screen sizes but also uni- vs. bi-manual input.

Necessity of Tagging Widgets

Pinning objects that already have the desired set of tags works best if related objects are close to each other. However, if there

is no object nearby that can be used to *gather* the desired tags by pinning, an appropriately tagged object has to be found first. Assuming such an object exists, such a search could take time. But, if a tag has not been used yet, finding said tag among the objects is impossible. Another example is the removal of a single specific tag. The user would have to find and pin first an object that has only the desired tag(s). Thus another mechanism is necessary and the user can always use the tagging widgets at the side of the screen. when there is no object with the desired set of tags available (or it would simply take too long to find it).

Copying vs. Toggling vs. Combining Tags

There are several possibilities for applying a set of tags to one or more target objects. The simplest possibility is to copy the tags to all target objects, overwriting any previous tags—effectively *painting* the tags on the objects. However, for complex tag assignments this may require more steps than ideal. Adding another tag to differently tagged objects could become tedious, as existing tags would have to be re-selected, to avoid overwriting. We decided to permit a more expressive combination of tags, as presented. During early development we played with toggling all selected tags on target objects. Yet, toggling all tags simultaneously results in hard to predict changes. Thus we decided that the explicit choice of either adding or removing tags is a better alternative.

Adaptation to Smaller Devices

Depending on the used device, the presented tagging widgets might not be the best possible interface for accessing single tags. The two main factors are display size and if the device is operated using a single or both hands. While pinning objects and performing selections is possible with a single hand, it is much more limiting than bi-manual input. Yet, the behavior of the tagging widgets at the display border can be adjusted for uni-manual input by making them *also* toggle-able. On larger devices, such as tabletop displays, there is enough space for tag widgets at the border of the screen. This way any desired combination of tags can be activated at all times, independent of the tags currently assigned to the visible objects. Pinning tagged objects that are close to the objects being modified also remains useful. Large screens benefit from this *local* operation, as attention then does not need to be divided between the target area and the tagging widgets. For smaller multi-touch devices, such as smartphones or tablets, most often a single hand is used to interact, while the other hand is holding the device. Wagner et al. [24] provided an interesting related discussion. On small devices, a pop-up menu with selectable tags, activated with a single hold gesture, is likely a good alternative—it can be operated using a single hand and does not consume screen real estate, unless it is invoked. To resolve the ambiguity with a pin gesture, we cancel the menu if a second touch is registered.

Selecting Rectangular Groups of Objects

In addition to selecting objects crossed by a path, our technique can also support the selection of objects inside rectangular regions. Naturally, this depends on the usage scenario, object layout and application. To select a rectangular group of objects an area can be defined by dragging the diagonal of the rectangle. All objects intersecting or completely inside the

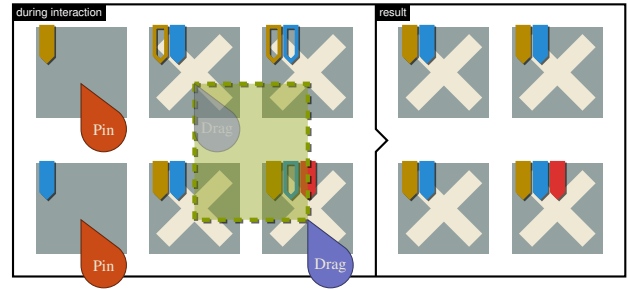


Figure 14. Dragging a rectangle selects all objects that are contained or intersecting it. As with the path technique the initial touch position of the dragging operation defines if the pinned tags are added (as shown here) or removed. Outlined tags preview impending changes.

rectangle are then selected. Figure 14 illustrates this idea. The mechanism for choosing the mode of operation, *i.e.*, if tags are to be added or removed from the selected objects, remains the same, *i.e.*, is based on the position of the first touch. Depending on the spatial layout and application context, rectangular or path selection is more useful. For cases when both are required, Saund et al.’s overloaded version of lasso and rectangle selection [21] is compatible with our work.

Simplification to a Single Group

Many existing applications support only selection of a single group. These can still benefit from the selection *editing* capabilities of our interface. In this scenario the interaction technique of pinning a selected object is still very useful for adding *additional* objects to an existing selection with a rectangle or path gesture. Yet, highlighting is likely more appropriate than a tag interface. This idea effectively enhances existing multi-touch group selection approaches, which typically do not support editing of a selection.

Selecting Overlapping Objects

If objects overlap each other it might not be possible to select the desired objects using a single selection operation. One of the benefits of our technique is its capability to easily extend and modify selections. The user can compose the desired set of objects through multiple selection steps. Harpoon [12] already demonstrated that *partially* overlapping objects can often still be selected using path-based techniques. For objects that completely overlap, a technique such as *Tumble! Splat!* [20] could be added to access occluded objects. Intelligent selection assistance systems such as *Suggero* [14] might also be helpful. If the background is completely covered by an object, *e.g.*, while the view is zoomed in, it is impossible to signal a subtraction by starting a path gesture on the background. In this case the editing gesture could be started on the bezel.

CONCLUSION AND FUTURE WORK

We introduced our new multi-touch tagging interface. It employs tag widgets to define multiple overlapping tag groups. One of its key features is the ability to easily edit existing tags by adding and removing items, even just for a single group. Our interface relies heavily on pin gestures, *i.e.*, holding touches, to provide a context for other actions performed at the same time. We validated our design in user studies and demonstrated that our tagging interface performs faster than alternative interfaces that rely on tapping of individual objects.

The work of Dehmeshki et al. [3] suggests that our path-based technique directly generalizes to other, non-rectangular layouts and we will explore this. We also plan to investigate what other interaction tasks could benefit from pinning.

REFERENCES WITH CLASSIFIER TAGS

1. Baudisch, P. Don't Click, Paint! Using Toggle Maps to Manipulate Sets of Toggle Switches. In *Proc. UIST '98*, ACM (1998), 65–66.
2. Dehmeshki, H. and Stuerzlinger, W. Design and Evaluation of a Perceptual-based Object Group Selection Technique. In *Proc. BCS '10*, (2010), 365–373. INDIRECT, CROSSING, ON/OFF OBJECTS, PATH, AUTO MODE.
3. Dehmeshki, H. and Stuerzlinger, W. GPSel: A Gestural Perceptual-Based Path Selection Technique. In *Proc. SG '09*, (2009), 243–252. PEN, PATH, CROSSING, SURROUND, INDIRECT, OBJECTS, LASSO, AUTO MODE.
4. Dehmeshki, H. and Stuerzlinger, W. ICE-Lasso: An enhanced form of Lasso selection. In *Proc. TIC-STH '09*, IEEE, (2009), 630–635. PEN, LASSO, OBJECTS, IN/OUT, INDIRECT, SURROUND.
5. Dehmeshki, H. and Stuerzlinger, W. Intelligent Mouse-Based Object Group Selection. In *Proc. SG '08*, Springer (2008), 33–44. MOUSE, OBJECTS, DIRECT, INDIRECT.
6. Gutwin, C., Cockburn, A., Scarr, J., Malacria, S., and Olson, S. C. Faster Command Selection on Tablets with FastTap. In *Proc. CHI '14*, ACM (2014), 2617–2626.
7. Hinckley, K., Guimbretiere, F., Agrawala, M., Apitz, G., and Chen, N. Phrasing Techniques for Multi-stroke Selection Gestures. In *Proc. GI '06*, CIPS (2006), 147–154. PEN, CROSSING, LASSO, SURROUND, INDIRECT, DIRECT, OBJECTS, IN/OUT, AUTO MODE.
8. Hinckley, K., Guimbretiere, F., Baudisch, P., Sarin, R., Agrawala, M., and Cutrell, E. The Springboard: Multiple Modes in One Spring-loaded Control. In *Proc. CHI '06*, ACM (2006), 181–190.
9. Hinckley, K., Yatani, K., Pahud, M., Coddington, N., Rodenhouse, J., Wilson, A., Benko, H., and Buxton, B. Pen + Touch = New Tools. In *Proc. UIST '10*, ACM (2010), 27–36.
10. Kawasaki, Y. and Igarashi, T. Regional undo for spreadsheets. In *Adjunct Proceedings of UIST '04*, ACM (2004), 2 pages. MOUSE, REGION.
11. Kin, K., Agrawala, M., and DeRose, T. Determining the Benefits of Direct-touch, Bimanual, and Multifinger Input on a Multitouch Workstation. In *Proc. GI '09*, CIPS (2009), 119–124.
12. Leitner, J. and Haller, M. Harpoon Selection: Efficient Selections for Ungrouped Content on Large Pen-based Surfaces. In *Proc. UIST '11*, ACM (2011), 593–602. PEN, CROSSING, SURROUND, INDIRECT, DIRECT, LASSO, OBJECTS, PATH.
13. Li, Y., Hinckley, K., Guan, Z., and Landay, J. A. Experimental Analysis of Mode Switching Techniques in Pen-based User Interfaces. In *Proc. CHI '05*, ACM (2005), 461–470.
14. Lindlbauer, D., Haller, M., Hancock, M., Scott, S. D., and Stuerzlinger, W. Perceptual Grouping: Selection Assistance for Digital Sketching. In *Proc. ITS '13*, ACM (2013), 51–60. PEN, CROSSING, INDIRECT, DIRECT, OBJECTS, PATH.
15. Luo, Y. and Vogel, D. Crossing-based Selection with Direct Touch Input. In *Proc. CHI '14*, ACM (2014), 2627–2636.
16. Mizobuchi, S. and Yasumura, M. Tapping vs. Circling Selections on Pen-based Devices: Evidence for Different Performance-shaping Factors. In *Proc. CHI '04*, ACM (2004), 607–614. PEN, OBJECTS, LASSO, DIRECT, INDIRECT, SURROUND.
17. Moran, T. P., Chiu, P., and Melle, W. van. Pen-based Interaction Techniques for Organizing Material on an Electronic Whiteboard. In *Proc. UIST '97*, ACM (1997), 45–54. PEN, IN/OUT, LASSO, RECTANGLE, REGION, AUTO MODE.
18. Moscovitch, T. and Hughes, J. F. Indirect Mappings of Multi-touch Input Using One and Two Hands. In *Proc. CHI 2008*, ACM (2008), 1275–1284.
19. North, C., Dwyer, T., Lee, B., Fisher, D., Isenberg, P., Robertson, G., and Inkpen, K. “Understanding Multi-touch Manipulation for Surface Computing”. In *Human-Computer Interaction INTERACT 2009*, vol. 5727. LNCS. Springer Berlin Heidelberg (2009), 236–249. TOUCH, OBJECTS, DIRECT, INDIRECT, SURROUND, ON/OFF OBJECTS, HOLD, PIN, MULTI-TOUCH.
20. Ramos, G., Robertson, G., Czerwinski, M., Tan, D., Baudisch, P., Hinckley, K., and Agrawala, M. Tumble! Splat! Helping Users Access and Manipulate Occluded Content in 2D Drawings. In *Proc. AVI '06*, ACM (2006), 428–435.
21. Saund, E., Fleet, D., Lerner, D., and Mahoney, J. Perceptually-supported Image Editing of Text and Graphics. In *Proc. UIST '03*, ACM (2003), 183–192. MOUSE, RECTANGLE, LASSO, ON/OFF OBJECTS, REGION, OBJECTS, DIRECT.
22. Seifried, T., Rendl, C., Haller, M., and Scott, S. Regional Undo/Redo Techniques for Large Interactive Surfaces. In *Proc. CHI '12*, ACM (2012), 2855–2864. PEN, SURROUND, OBJECTS, REGION.
23. Strothoff, S. and Hinrichs, K. Adding Context to Multi-touch Region Selections. In *Proc. MUM '13*, ACM (2013), 15:1–15:8. TOUCH, IN/OUT, RECTANGLE, REGION, MULTI-TOUCH, AUTO MODE.
24. Wagner, J., Huot, S., and Mackay, W. BiTouch and BiPad: Designing Bimanual Interaction for Hand-held Tablets. In *Proc. CHI '12*, ACM (2012), 2317–2326.
25. Wobbrock, J. O., Morris, M. R., and Wilson, A. D. User-defined Gestures for Surface Computing. In *Proc. CHI '09*, ACM (2009), 1083–1092.
26. Xu, P., Fu, H., Au, O. K.-C., and Tai, C.-L. Lazy Selection: A Scribble-based Tool for Smart Shape Elements Selection. *ACM Trans. Graph.* 31, 6 (Nov. 2012), 142:1–142:9. TOUCH, CROSSING, OBJECTS, CROSSING, DIRECT, INDIRECT, PATH.