Regional Undo/Redo Techniques for Large Interactive Surfaces

Thomas Seifried[1]**, Christian Rendl**[1]**, Michael Haller**[1]**, Stacey D. Scott**[2]

[1]Media Interaction Lab
University of Applied Sciences Upper Austria
Hagenberg, Austria
mi-lab@fh-hagenberg.at

[2]Collaborative Systems Lab
University of Waterloo
Waterloo, Canada
s9scott@uwaterloo.ca

ABSTRACT

When multiple users are simultaneously sharing a workspace, it is not always clear what should happen when a user invokes an undo action. In this paper we explore different user interfaces for undo/redo for co-located collaborative workspaces, such as large interactive whiteboards. A preliminary study revealed that users expect neither a global nor personal undo, but rather a regional undo. We propose and evaluate three automatic regional undo/redo techniques (clustering, workspace, field of view) designed for a large interactive whiteboard. The results of the evaluation showed that an undo technique based on users' field of view was most preferred, while the content-based clustering technique produced most errors. We conclude with potential improvements to the developed techniques, and propose a set of design recommendations for implementing regional undo/redo on large interactive surfaces.

Author Keywords

Undo; co-located collaboration; interactive surfaces; interactive whiteboard; large displays; multi-user input;

ACM Classification Keywords

H.5.2 [Information Interfaces & Presentation]: User Interfaces - Graphical user interfaces; H.5.3 [Information Interfaces & Presentation]: Group and Organization Interfaces - Computer Supported Collaborative Work;

General Terms

Design, Human Factors.

INTRODUCTION

Undo/redo is a very common application operation familiar to most users. It provides a simple method to correct errors, to recover a former document state and it is also a safe way to learn software features in a trial-and-error approach [7]. Therefore, undo/redo is seen as an essential functionality [29], which virtually every desktop application provides. On such single-user systems undo is quite simple to implement and it is quite clear what users expect when they select the undo functionality. In contrast, the undo action on a multi-user interface is more complex – not only from a technical [20], but also from a user interaction perspective (e.g., awareness problems, feedback).

When multiple users are working on a document at the same time, it is not always clear what happens if one user invokes an undo action. Historically, this problem has been mainly addressed in the field of *distributed* collaborative environments. Due to the fact that interactive surfaces are becoming larger and larger, the same problem is occurring in *co-located* collaborative setups. Moreover due to advances of input technology, many of these large interactive surfaces also provide a *multi-user* experience, which again makes it more complex to address the undo/redo problem.

Figure 1: Collaborative *(left)* and individual *(right)* work on a large interactive display.

Figure 1 depicts a scenario on a large interactive wall-display, where multiple users can either work close together *(left)* and share the same region, or work individually *(right)*. In both cases, the undo/redo action should make sense to each person. A *global undo* [27] introduces significant issues for solving the undo/redo problem on a large screen, because the person on the right side would not be aware of what is happening if the pair on the left side triggered an undo action. A *personal undo*[1] [1], on the other hand, may also make undo highly cumbersome. If, for example, the pair on the left shares the same sketch on the board, they may consider it as their collective creation and may perform an undo action not just for their own personal strokes (Figure 2), but for the entire sketch. In this case, the system should allow them to also undo strokes that have been produced by their collaborator.

---

[1] *Personal* undo is a synonym to *local* or *user* undo.
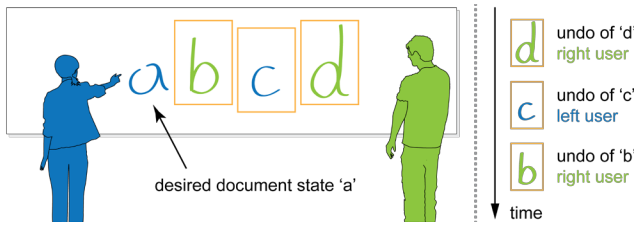
Figure 2: A personal undo action requires a close collaboration of all users to get back to the desired status.

In this paper, we address these challenges. To understand user expectations of undo/redo actions performed on a large interactive display, we conducted a background study. Based on the results of this study, we present the design and implementation of several *regional undo techniques* that provide different solutions to meet the users' expectations. Finally, we discuss the results of the user evaluation of the presented techniques.

RELATED WORK

Undo/Redo has been a focus of research for many years and has primarily been studied for single-user [27] and distributed multi-user [1] systems. However, a classification scheme has yet to be established. To reduce confusion about the terms in this paper we propose a new classification scheme, and structured the discussion of the related work using this scheme. Undo/redo concepts can be described using three main categories: undo/redo *model*, *method* and *interface*.

The *undo model* refers to algorithms and data structures behind undo/redo actions and is most relevant for the technical implementation. In general, undo models can be split into two groups: models that only allow actions to be undone sequentially, namely *linear* undo models, and undo models which can undo arbitrary actions from the past, namely *non-linear* undo models. *Non-linear* undo models can be further divided into models for single-user [2, 5, 28, 29, 30] and multi-user applications [3, 20, 21, 22, 25, 24, 26]. While the data structure used to store "undoable" actions is different for many undo models, most models facilitate a so-called *history list,* which holds a linear, chronologically sorted list of all undoable actions. In order to support synchronous collaborative work a non-linear multi-user undo model is necessary as described by Sun [25, 24]. However, the undo model is not the focus of this work and more detailed information about algorithms and data structures can be found in the referenced work. In our software prototype, we implemented a *non-linear (selective)* undo model based on Sun's *AnyUndo* [25].

The second category constitutes the *undo method*, which defines the general behavior of undo in an application, e.g. *global* [1, 27], *personal* [1], *selective* [3, 20] or *regional undo* [20]. Generally speaking, the undo method defines the way actions are selected for undo. For example in personal undo, users only undo actions that have been invoked by themselves, or in regional undo, users only undo actions in

a certain area on the screen. The offering of *personal*, *selective* or *regional undo* in an application requires *a non-linear* undo model that can selectively undo actions from history.

Finally, the *undo/redo interface* describes the concrete user interface for an undo method. This includes both interaction as well as visualization. Research in this area mainly focused on history visualizations as proposed in [8, 13, 14]. Nevertheless the invocation of undo and feedback can also pose an issue, especially in a co-located multi-user environment. Furthermore, Choudhary and Dewan identified general requirements for multi-user undo [7] which should be considered in the design of an undo interface. In this paper we focus on different *undo interfaces* within the *regional undo* method.

Regional undo has thus far received little attention. Prakash and Knister [20] introduced *regional undo/redo* as a possible side-effect of their *selective* undo approach for a distributed text editor. They pointed out that designing a user interface for *regional undo* might be much more complicated than for *global* or *personal undo*, because the location of text regions may change over time. Li et al. [15] approached this problem by using arbitrary regions in text documents to choose actions for *selective undo*. Moreover, a technique called "region transformation" modifies regions according to later executed actions. Nevertheless, in this paper we focus on sketched content on large interactive surfaces which is quite different to word processing in text editors. Meng et al. [17] and Nakamura et al. [19] presented methods to filter actions in a textual history visualization based on specific regions in a document. In both papers, additional user interface dialogs were introduced which considerably limit the usability for most undo use-cases. In contrast, we aim to simplify the user interaction by embedding *regional undo* directly into the document without additional dialogs.

As Abowd et al. [1] have claimed, undo is highly related to user expectations. There has been some previous work that has examined user expectations related to undo functionality, but these have focused on single-user [6] or distributed [10] settings. Yet, no one has investigated user expectations in a co-located, collaborative work setting, which we believe may introduce key differences in user expectations of undo functionality. Therefore, we conducted a background study to capture user expectations in a co-located whiteboard setting.

UNDERSTANDING USERS' EXPECTATIONS

To gain a better understanding of how users perceive undo/redo actions, we conducted a background study. The primary goal of this study was to find out which of the undo/redo models (global-, personal- or regional undo) matches user expectations. 23 unpaid volunteers (17 males, 6 females) were recruited from the local university. The participants' ages ranged from 20 to 32 years (M=24.3, SD=3.4). All participants were daily users of computers, with an average daily usage of 8.5 hours (SD=2.5). Only

13% of them had worked with an interactive whiteboard before; none had used it for synchronous co-located collaboration.

Apparatus

The study was conducted on a large (4.5m×1.2m) interactive whiteboard (IWB), operated by three Hitachi CP-A100 projectors with a total resolution of 3072×768. The system was capable of handling simultaneous multi-user input by using Anoto digital pens (ADP-301) (Figure 3) [9].
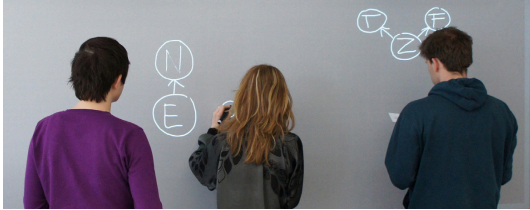

Figure 3: The first study was conducted to determine expectations of undo in co-located collaborative work.

A very simple (freehand) drawing application was used on the IWB, whereby all graphical user interface elements were hidden to reduce clutter.

Procedure

Each participant was accompanied by two study confederates [16] to provide a multi-user experience with three people per group. The confederates were not allowed to express any opinions and were also not interviewed to minimize influence on the participants' answers. The same people acted as confederates for each trial.
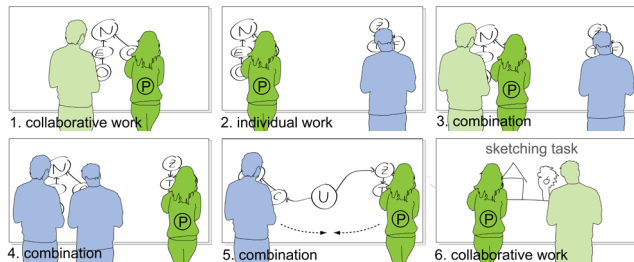
Figure 4: Six scenarios were conducted in the background-study. The participant is highlighted with "P".

At the beginning, each participant was asked to complete a pre-test questionnaire soliciting demographic data. Prior to data collection the goal and procedure of the experiment was introduced to the participants. Each participant had to perform six tasks; every task was designed to imitate a different scenario of co-located work on the IWB: individual work, collaborative work, and combinations of both (Figure 4).

After each task, participants were asked to explain verbally what they would expect to change on the IWB, if they invoke a first and a second undo action. Additionally, each participant concluded a task by completing a short post-trial questionnaire. For each task the whole screen was cleared. Five tasks were to draw a simple directed graph (Figure 3)

and one task to draw a sketch. Each participant and confederate was provided with instructions printed on a sheet of paper, containing the steps to construct the graph or sketch (e.g., *node "E" depends on node "N"*).

For individual working scenarios, the participants' instructions contained all necessary information to finish the graph, but for collaborative ones they contained only half of the information. Due undo granularity was not subject of this study and to keep the results comparable, the participant were told that one step on the instruction sheet will be undone by a single undo action. According to the recorded verbal answers the experimenter recreated a resulting graph, representing the expected result of each undo step, which was used for analysis.

Design

The study was a within-subjects design and the order of the tasks was counterbalanced. The resulting (partly undone) graphs were matched with the anticipated results of *global* undo, *personal* undo and *regional* undo. To highlight differences between the undo methods, the tasks were designed so that the anticipated result of each method was different if possible. The results of the sketching task were not included into the accumulated results because the same collaborative setting (group work) was also represented in a graph drawing task and the results were the same.

Results

The study indicates that regional undo meets the expectations of the users in most cases (cf. Figure 5). In 86% of all cases it produced the results the participants were expecting. Personal undo (47%) and global undo (22%) would have been much less expected. In 5% of all cases the expectation could not be applied to any known undo method.
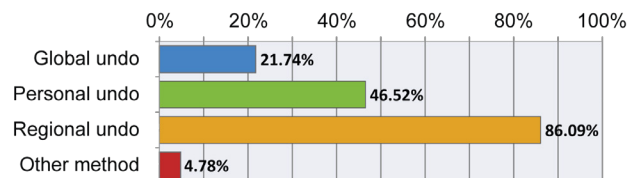

Figure 5: Regional undo meets the expectations most.

The observation of the participants also showed that the expected behavior of undo depends on the type of work (Figure 6). When working on a common task, users expect undo to affect the collective work of the group. When working on an individual task, users expect undo to be applied to their own work. These results also apply even if others are working on the same interactive space. No participant expected that an undo action would have any effect on other graphs except their own graphs as long as they do not change the focus of their current work. Also, awareness of the other people's actions in the interactive surface seems to be an important factor. Users reported that they were not aware of the other people's actions when they were working on another task, even when they were quite close to each other like in Figure 3. Therefore, it was im-

possible for them to anticipate the effects of their undo actions on other people's work. Furthermore, most participants reported that such behavior would be socially inacceptable.
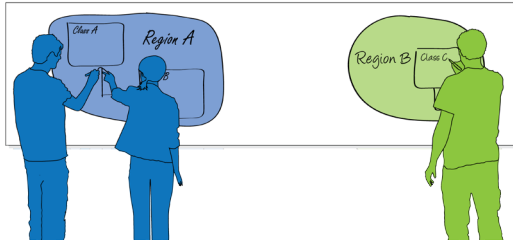
Figure 6: Users expect that undo only affects their current workspace, which might either be a collaborative one or not.

The controlled nature of the study and the strong focus on collaborative work might have exaggerated the results. So, one might argue that personal undo would meet the expectations well enough in a real world scenario. While this might be true for correcting errors in most cases, for other uses of undo (trial-and-error, restore document state) it causes problems (as described in Figure 2). Moreover in some cases personal undo does not work at all. If users leave the collaborative session access to their undo-steps will be lost. Also technical limitations of the current system can trigger this problem, for example if the system loses track of the user or if users switch input devices which are used to identify users.
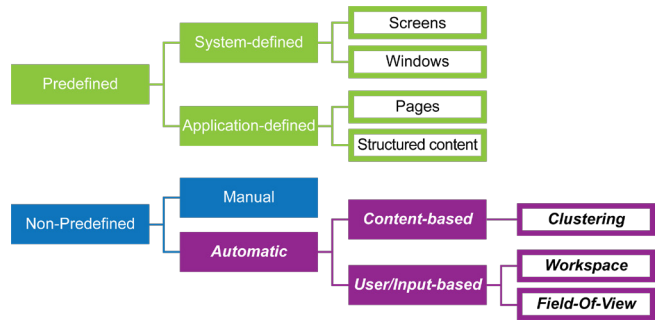
REGIONAL UNDO/REDO
The results of the background study suggest that most users expect *regional undo* once they are working on large interactive surfaces. This means, the spatial "reach" of undo-redo actions should be limited to a certain area of the surface. This may be due to the user's mental separation of the whole display into separate workspaces, but also due to the lack of awareness of what other users are doing elsewhere on the large surface.

But regional undo/redo can be implemented in several different ways. In general, a regional undo/redo technique consists of the following three steps: First, the definition of a region, second, the selection of a region, and finally the undo/redo action itself within that region. Although each of those steps needs to be considered, the way a region is defined is a strong distinction for regional undo techniques.

Defining Regions
In the context of regional undo, we use the term "region" as a two-dimensional area with distinct boundaries of arbitrary shape, which is ultimately limited by the boundaries of the display. Therefore, a region can be defined in many ways. Figure 7 depicts a conceptual framework of how we can define regions for the undo/redo actions. The simplest approach is to use already visible boundaries. Those predefined regions could be system-related, e.g., boundaries of the screen, or defined by the application, e.g., by showing pages or other elements that visually separate the screen. If

there are no pre-defined regions, a user could create them on demand by explicitly defining regions.

Figure 7: A conceptual framework for defining regions.

In the following sections, we will briefly describe different regional undo/redo techniques based on the described framework, but we will focus on *automatic techniques* due to their higher complexity.

Pre-defined Regional Undo/Redo
In many cases, the display or the application already provides a physical or virtual region. For example in a multi-projector setup, each projection screen may be visually separated from each other (if not edge-blended). Furthermore, the application itself can provide pre-defined virtual regions. Editors, for example, often mimic the real world by providing pages. Such physical and virtual boundaries are already good candidates for an undo/redo region as they are easy to understand. We also have a similar experience if we work with multiple applications on the desktop PC, where every application has its own undo/redo history.

Manual Regional Undo/Redo
A very simple way of defining an undo/redo region is to define it manually as described by Li et al. [15]. In this case, users can create any desired size of a region on which they would like apply an undo/redo action. An explicit visual feedback of the undo/redo region is not necessary since it is defined by the users themselves. This technique can be easily implemented in many applications. If the application features some kind of selection (e.g., a lasso or marquee tool), manual regional undo can utilize those features and can be applied to the selected region.

The disadvantage of this technique is that it requires a two-step approach, namely *selecting the region* and *triggering the undo/redo action*. This might be okay if a user wants to restore a certain document state on a certain area, but it will be very cumbersome and time-consuming for small corrections. An alternative implementation of a manual regional undo would be a visual overlay or magic lens that provides quicker access. However, this would introduce additional visual clutter.

AUTOMATIC UNDO/REDO TECHNIQUES
For applications with no explicit visual regions (e.g., a large sketching surface), regional undo techniques are more complex. In this case, we have to find a way how to detect re-

gions automatically. In the following sections, we explore three different techniques that facilitate the spatial organization of content in the application and/or the users' location.

Automated Clustering Undo/Redo Technique

Instead of defining the region manually, a clustering technique groups currently visible items into clusters defining an undo/redo region. The idea behind this content-based regional undo/redo technique is to mimic the human's perception that items belong together if they are in spatial proximity [12]. With this technique, regional undo is executed in the region next to the undo/redo button the user pressed (Figure 8).
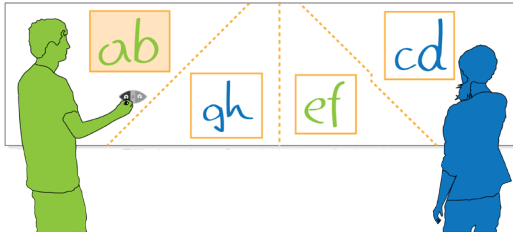
Figure 8: In the clustering technique, items in close proximity are grouped together and define an undo/redo region. Once a user selects an undo action, the cluster closest to the invoked tool palette (containing the undo button) will be chosen.

As depicted in Figure 9, our implementation of the clustering results in rectangular boxes containing associated content. We used an axis-aligned bounding box approach for the automated clustering technique. Once the distance between two items is under a certain threshold (in our case 11.7 cm or 80 px) they are grouped automatically into a new cluster. These clusters are calculated for every item on the screen. If a user selects an undo action, the cluster closest to the invoked user interface element (containing the undo button, see Figure 8) is chosen.
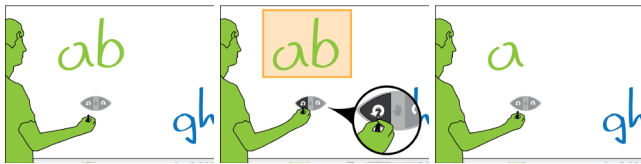

Figure 9: Upon selection of the undo-button on the digital tool palette (left), users get visual feedback of the undo/redo region (middle). The feedback disappears once the undo action has occurred (right).

By automatically generating undo/redo regions, the required effort to perform an undo action is reduced. In contrast to the later described techniques, this only provides temporary visible feedback about the region (see Figure 9). If this feedback would be permanent, the visual structure of an application would change and may also put too much emphasis on undo/redo functionality.

Workspace Undo/Redo Technique

This technique is based on the idea that users occupy a certain workspace on a large interactive surface (Figure 10). The background study showed that users would not expect that their actions have an effect outside their current work area. Therefore in this technique the undo/redo region is defined by the users' workspace. Although each user occupies a single workspace, collaborative work is still supported as such workspaces may overlap if two or more users are working in the same area.
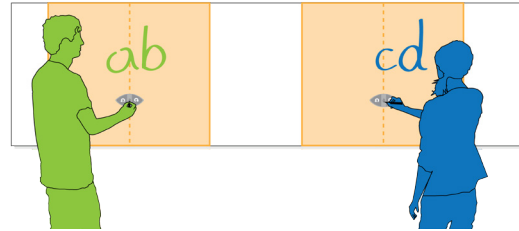
Figure 10: The workspace technique facilitates the users' workspace for regional undo.

We also simplified the workspace region by expanding it to the full height of the surface, because people usually do not work individually on top of each other on a vertical display. This simplification also allows placement of unobtrusive feedback to the top or bottom of the workspace and hence a way to inform the user about the extents of the workspace. In our implementation a bar (Figure 11) is displayed on top of the screen indicating the width of the workspace. The initial size is 75 cm which has been derived from guidelines for an ergonomic workspace [11]. The location of the workspace is controlled by the user interface element that contains the undo/redo buttons. This element is always at the center of the workspace. In our implementation the undo/redo buttons are incorporated to the users' digital tool palette which can be freely moved to any location on the display. Hence the workspace is relocated accordingly.

Figure 11: In our implementation, workspaces are visualized by color-coded bars on top of the screen. Inward arrows clarify overlapping workspaces. Positioning the bars at the top of the workspace limits the visual clutter during work.

Alternatively, the whole workspace can be manually repositioned by dragging the workspace bar to a different location. Moreover they can be manually resized by using the arrow-shaped handles (see Figure 11). In both cases the bar will show a lock symbol to indicate that the position of the digital tool palette will not affect the location of the workspace anymore.

In contrast to the cluster regions in the automated clustering technique, this technique provides areas for permanent

visual feedback. Yet it introduces minimal clutter, because it is usually located in the periphery visual field of a user and users rarely interact near the boundaries of such large displays [9]. Despite this it still provides good feedback for changes of the workspace - since the human eye is highly sensitive to motion in the peripheral visual field.

Field-Of-View Undo/Redo Technique

The field of view (FOV) technique is based on the area a user physically sees. In the background study, we observed that the awareness is an important factor in setting users' expectations of undo/redo actions. Furthermore, users did not expect that an undo/redo action would affect an area far from their current location. Therefore, the undo/redo region is formed around the user's field of view (Figure 12).
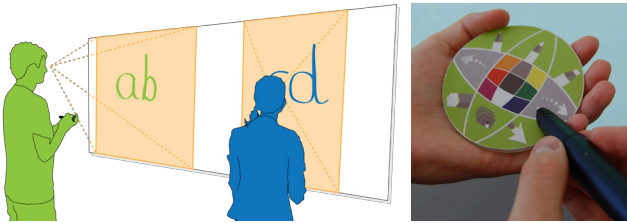
Figure 12: The personal field of view defines the undo region. A tangible tool palette facilitates triggering of undo/redo actions and allows the user to move freely.

This also gives users the ability to simply change the undo/redo region by moving their body. To support "free" movement of users, undo/redo-actions are triggered by a mobile interface (Figure 12, *right*). This enables users to step back from the whiteboard to enlarge their field of view and still be able to trigger undo/redo. Hence, no direct input is necessary on the interactive surface.
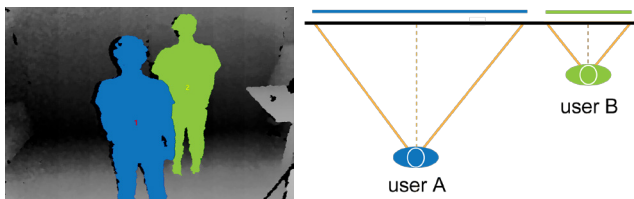
Figure 13: Users in front of the interactive surface are identified and their position tracked using Microsoft Kinect (left). Depending on the distance to the surface the width of their visual field is estimated (right).

Our implementation of this technique uses an approximation of the field of view by tracking users in front of the large interactive surface. As depicted in Figure 13, users are identified and their position tracked by using a Microsoft Kinect; one Kinect is able to capture the entire 4.5 meters in front of the IWB. Depending on the distance between the user and interactive surface we estimate the area visible to the user (we assume a field of view of 125°). An input device is mapped to a tracked user upon the first interaction on the IWB by comparing the users' location to the received input coordinates. Similarly to the workspace technique, the approximated visual field is only applied in the

horizontal direction. Moreover, the extents of the region are visualized by a bar on the top of the display. Users can either use a digital or tangible tool palette to trigger an undo action (our tangible tool palette is shown in Figure 12, *right*).

In theory, visual feedback should not be necessary in an optimal implementation of this technique. But accurate eye-tracking would be necessary to calculate the precise visual field, which would require additional hardware and calibration for each user.

EVALUATION

We conducted an initial experiment to evaluate the proposed automatic regional undo/redo techniques. In the study, we mainly focused on qualitative data to gain insights on the expectations for each technique. Overall the objectives of this study were:

1. to measure the users preference,
2. to understand the users preference,
3. to observe which technique is less prone to errors (errors in the sense of undo/redo-actions leading to unexpected results), and
4. to find triggers for such unexpected undo results in our regional undo techniques.

Participants

In total, 18 unpaid volunteers (15 males, 3 females) participated from local software engineering companies (14 employees) and the university (4 employees). Participants performed the study in pairs (nine pairs in total). Their age ranged from 22 to 44 years (M=29.2, SD=5.7). All participants were daily computer users, with an average daily usage of 8.6 hours (SD=2.0). No participant reported using an interactive whiteboard frequently, but 56% had used one before, and 17% had used one for synchronous co-located collaboration at least once. 17 participants reported having a technical background. Finally, 95% of participants used undo/redo regularly, with 50% of them using it frequently.

Apparatus

The study was conducted in the same room and used the same hardware as the background study. A similar sketching application was used, with the addition of the regional undo/redo techniques described above. It also provided a digital and tangible tool palette to trigger undo and redo actions.

Tasks

All participants performed three different tasks that were designed to represent typical use cases for co-located work on an IWB. Each task consisted of two parts covering major undo use cases: error-correction and recovering a former document state. In the first part participants had to write text on the IWB. After each word we analyzed the written strokes with handwriting recognition software using Microsoft's built-in Ink Analyzer. Participants could only continue with the task when a word was recognized correct-

ly. In the second part the participants were told to undo actions until they reach a specific (former) document state.

The quality of the recognition software was intentionally reduced with an average error probability of 60% per word to ensure that every participant had to call undo multiple times during one trial. At the beginning, users were informed about the error-proneness of the handwriting recognition software, so that participants did not feel discouraged or embarrassed during the study.

Task 1 – Individual work, change of location: Each participant had to write a short sentence at a pre-defined location on the IWB (four words with 35 letters on average). Upon completion, participants then had to switch places and undo their partner's entire sentence. This sequence was then repeated with a different sentence (Figure 14).
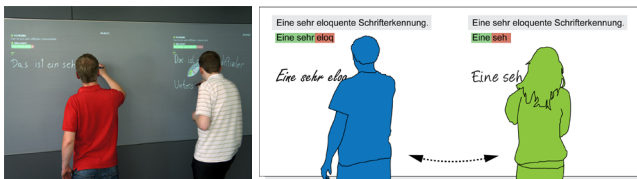

Figure 14: Individual work, change of location after completion of sentence.

The objective of this task was to test the regional undo techniques focusing on an individual working scenario. We also wanted to find out how well the techniques performed when participants changed locations.

Task 2 – Collaborative work, turn-taking: Similar to the first task, participants had to transcribe a sentence. In contrast to task 1, the participants had to complete this task together, each taking turns to write successive words in the sentence.
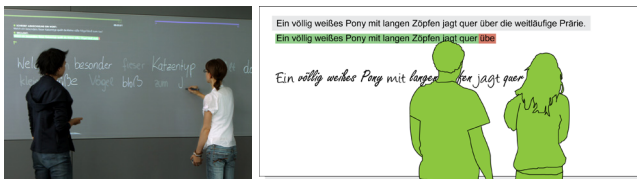

Figure 15: Collaborative work, turn-taking after each successive word.

The sentences were also longer in this task (11-13 words with 66 letters on average). Once the sentence was completed, the participants had to clear the whole sentence by using undo (Figure 15).

This task was designed to mimic a turn-taking collaborative working scenario. Such a scenario is common in collaborative work, especially during discussions (e.g., when creating a to-do list). Moreover, the length of the chosen sentences in this task ensured usage of a large area on IWB.

Task 3 – Collaborative work, synchronous work: In the final task, participants had to collaboratively recreate a single tag-cloud consisting of 14 words. The words in the tag-cloud were presented in three steps, with each step revealing more words. Participants were instructed to maintain the predefined spatial arrangement of the tag-cloud. Once completed, participants were asked to clear the whole tag-cloud (Figure 16).

The objective of this task was to encourage synchronous collaborative work on a single artifact. This scenario emulates common collaborative work activities, where the group members divide the task into sub-tasks. This usually appears when content needs to be created quickly under time pressure.
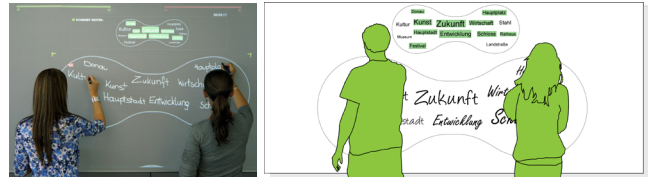

Figure 16: Collaborative and synchronous work.

The tasks were designed to challenge the limits of the different regional undo techniques. Therefore, participants experienced pros and cons of each technique to give them an impression how well each technique performs.

Design
The study used a 3 (task) × 3 (technique) within-subject design. In each session, a pair of participants performed three tasks (individual, collaborative turn-taking, collaborative synchronous), with each undo/redo technique (clustering, workspace, FOV). To minimize learning effects, the presentation order of the tasks was counterbalanced using a 3×3 Latin-square design. Similarly, the order of techniques within each task was also counterbalanced. Sentences (tasks 1 and 2) and tag-cloud words (task 3) were different for every trial in a session.

Procedure
Participants were first welcomed and provided an overview of the study and its goals. Participants then completed a pre-test questionnaire gathering demographic data. The sketching application, the study tasks, and the three different undo techniques were then explained to the participants. They were then given a 5-minute training session to familiarize themselves with the system, to practice writing in the system, and to practice the three undo techniques. Following this training session, they performed the three tasks, each with every undo technique. After completing a task with each technique participants were asked to complete a post-trial questionnaire. After completing all tasks, participants completed a post-study questionnaire regarding all executed tasks and techniques. The study was concluded with an informal interview to capture general experiences and feedback from participants. In total, each session lasted for approximately one hour.

Data Collection
Relevant data was captured through software logging, observation, questionnaires, and interviews. Participants'

system interactions were captured in computer log files, including all undo/redo actions. In all sessions, observational field notes were taken by two researchers present in the room, focused on counting and classifying unexpected behavior of undo (i.e., undo errors). Undo errors were counted if an undo action was followed by an instant correction, or were clearly not expected (e.g. undone action was not part of the incorrect word). Furthermore the observers also took short notes what might have caused this undo error. The undo error rate was calculated by dividing the number of counted undo errors with the total amount of undo sequences executed by the participants. The sequence of subsequent undo actions was captured by software logging. Ratings and feedback were captured with questionnaires using 7-point Likert-scale questions, and general feedback and experiences with the informal interview.

Results

Overall, the participants rated the FOV technique highest (M=6.16, SD=0.98). As depicted Figure 17 (top), half of the participants even gave the highest rating. The workspace techniques' rating was lower (M=5.33, SD=1.37), whereas the clustering technique got the lowest rating (M=4.16, SD=1.33) with almost 40% of the participants rating the clustering technique negatively (1-3).
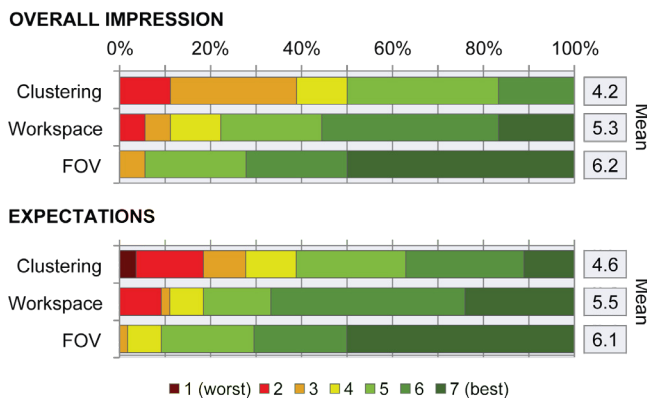
OVERALL IMPRESSION

EXPECTATIONS

Figure 17: Participants' ratings on overall impression (top) and how well each technique met the expected behavior (bottom).

This also reflects the accumulated ratings of how well the undo technique met the expectations of the participants (Figure 17, bottom). Participants reported that the FOV technique matched their expectations the most, while the clustering technique matched their expectations the least.

Unexpected Undo Behavior - Undo Errors

Log file analysis revealed that, on average, 20.96 (SD=8.29) undo sequences occurred per trial resulting in an average error rate of 3.2% (SD=5.2%). The undo error measurements were analyzed using a repeated-measures ANOVA test with Bonferroni corrections for pair-wise comparisons. For all tests an alpha level of 0.05 was used. The ANOVA test showed significant main effects for both the *Task* ($F_{2,16}$ = 6.399, $p < 0.01$) and the *Technique* ($F_{2,16}$ = 6.275, $p < 0.01$). The assumption of sphericity was not

violated. Post-hoc analysis showed that the clustering technique produced a significantly higher error rate than all other techniques (*Workspace*: $t(8)$ = 3.33, $p < 0.05$; *FOV*: $t(8)$ = 2.39, $p < 0.05$). The c*lustering* technique's error rate of 5.1% (SD=2.1%) was twice as high as from the *workspace* (M=2.2%, SD=1.6%) or *FOV* technique (M=2.2%, SD=2.8%). No significant difference was found between workspace and *FOV* technique.

These results are consistent with participants' feedback on the error-proneness of the techniques (Figure 18). Interestingly, even though the actual and perceived error-proneness of the workspace and FOV technique were similar, participants' clearly favored the FOV technique, as indicated by their high ratings of this technique when ask how well each technique met their expectations (Figure 17).
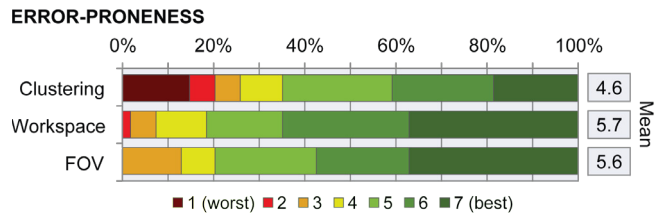
ERROR-PRONENESS

Figure 18: Ratings on the error-proneness of each evaluated regional undo technique (overall).

Causes of Errors

There were several reasons why the different undo techniques occasionally led to undesirable results. Causes of error with the clustering technique were (in total 26 errors):

1. *Ambiguous clusters / too large clusters (occurred 23 times):* Sometimes the calculated clusters did not match the expectations of the participants. For example, this happened during task 3 when participant were simultaneously working on the tag-cloud. Due to the close proximity between the words the whole tag-cloud was recognized as a single cluster, but participants expected separate undo regions.
2. *Cluster selection error (3 times):* If there were multiple clusters recognized, sometimes participants triggered an undo/redo action on a different cluster then they intended. For example, this often happened when they kept the digital tool palette static and reached for it to trigger undo, but the tool palette was closer to another cluster.

Causes of error with the workspace technique were (in total 11 errors):

1. *Incorrect workspace position (5 times):* Sometimes, the position of the workspace was unexpected because the participants placed their digital tool palette not in the center of their actual work area and hence inappropriate actions were undone.
2. *Locked workspace (3 times):* The participants' workspace was manually set to a specific position on IWB. Occasionally, they forgot to unlock it before triggering undo from a new position.

3. *Overlapping workspaces (3 times):* The workspaces of both participants sometimes overlapped when they were working individually but in close proximity. This occasionally resulted in contents of the other user being erroneously undone.

The main causes for errors with the FOV technique were (in total 10 errors):

1. *Incorrect position (5 times):* Sometimes, the position of the calculated field of view was unexpected, for example, when a participant changed their viewing angle without changing their body position.
2. *Unexpected size (4 times):* For some participants the calculated field of view was too large when standing close to the board.
3. *Overlapping field of view (1 time):* Similarly to the "overlapping workspaces" error described above, the field of view of participants overlapped when participants were working in close proximity, which occasionally led to content of the other user being erroneously undone.

DISCUSSION

Although the FOV is equally prone to errors as the workspace technique, participants preferred the FOV technique. Post-study interviews suggest that participants felt to control the FOV technique took less effort. One participant described the main advantage of the FOV over the other techniques: "The region is always there where I need it". Moreover, participants reported that they were able to more quickly become aware of the results of an undo action because, as another participant described, "it must be somewhere close to me". Despite the users' clear preference toward the FOV technique, all techniques produced some undesirable results. Therefore, we propose following improvements to reduce undo errors.

Improving Automatic Regional Undo/Redo Techniques

Our implementation of the FOV technique only tracked the position of a body. This implementation provided reasonable control over the undo region, however, tracking head [4] and, in particular, eye movement, would improve this technique. This may reduce errors of incorrect position and possibly also those of size. But FOV tracking depends on additional hardware that may not be available or feasible for every setup. In such cases improved clustering or workspace based techniques are also reasonable.

Clustering: The results of the study indicated that participants expected more fine-grained clustering, like in [18]. One way to address this would be to use lattice grouping to provide different levels of clusters [23]. Such approach, however, would also increase the complexity of cluster selection. Furthermore, an ideal clustering technique would need to comply with the way people structured content in their mind, like described in Gestalt laws of grouping [12]. Despite this, the lack of persistent visual feedback is clearly an open problem in this technique. Also it could be easily done by showing the boundaries of clusters all the time, it

would change the visual structure of an application and may also put too much emphasis on undo/redo functionality.

Workspace: Another potential improvement would be to combine clustering and workspace techniques. Visualization of the workspace was highly appreciated by participants. The main sources of undesirable undo behavior resulted from the workspace size not matching participants' expectations and the need to manually modifying its location. Locking its position was especially problematic. Combining the workspace and clustering techniques would enable the size of a workspace to be automatically adjusted, based on nearby clusters.

Regional Undo/Redo Design Recommendations

Summarizing we propose the following design recommendations which are based on the lessons learned from development of our regional undo/redo techniques and user feedback from both user studies:

- **Visualize extents of regions:** The position and extents of an undo region must be clear to the user. Therefore, it is essential to visualize them.
- **Minimize users' effort:** Regional undo/redo should require as little user effort as possible (in comparison to global or personal undo), especially when a user wants to correct a mistake.
- **Minimize visual overhead:** Though undo is a valuable feature, it is not the main application functionality and, thus, visual overhead should be minimized.
- **Provide additional manual control:** Automatic regional undo techniques can handle most undo cases. When users experience limits, however, manual control of the undo region should be provided. Automatic and manual regional undo can easily be combined, for example by enabling undo/redo on a user selection. But manual controls must be clear and simple to use.

LIMITATIONS

This paper is a first step exploring the area of undo/redo on large interactive surfaces. Hence, the tasks included in the evaluation were intentionally contrived to present opportunities to test the limits of the techniques under study in an efficient manner. However, these tasks were designed to emulate aspects of common collaborative work scenarios on an IWB. While we recognize that different types of collaborative work activities may occur, especially during long work processes, we feel the included tasks provide a good starting point for understanding at least the common usage scenarios of undo/redo on an IWB. However, further study is warranted in a more naturalistic setting with real task groups performing real-world tasks. We also limited our focus to text input on an IWB, because this activity often produces errors. Nevertheless, IWBs are frequently used for digital drawing. We believe, however, that study results will also apply to those use cases. Furthermore only a limited set of commands (*add* and *remove*) were used in this study. Further study is necessary for commands with a larger impact on the visual structure such as movements or

automatic sorting. Moreover, as this work is mostly focused on vertical displays further study is warrant for tabletop displays.

CONCLUSION AND FUTURE WORK

In this paper, we investigated undo and redo for co-located collaborative work on large vertical surfaces, supporting both collaborative and individual work. We first conducted a background study, which showed that regional undo, which only applied undo/redo actions to localized content, was most consistent with users' expectations in such an environment. Thus, we developed several regional undo techniques (clustering, workspace, and FOV) that each provide different potential regional undo solution. We then evaluated the developed techniques in an initial user study. The study revealed that users strongly preferred the FOV technique, while making significantly more errors with the clustering technique compared to either the workspace or FOV technique (which produced similar error rates). Based on the study results, and lessons learned in developing the evaluated techniques, we suggest improvements to these techniques, and recommendations regarding the design of regional undo techniques on large interactive surfaces.

In the future, we are interested in exploring regional undo techniques in mixed co-located and distributed cooperative work scenarios. We also plan to investigate the use of large-scaled eye tracking for refining the FOV technique. There are also additional topics related to undo/redo that were not discussed in this paper. For example, undo granularity is an important design consideration to providing effective undo functionality [1]. Finally, we plan to investigate the question, if regional undo, in general, can be substituted by a different method of undo, such as an "intelligent" group undo where the systems recognizes group or individual work regardless of regions on the display.

REFERENCES

1. Abowd, G. D. and Dix, A. J. Giving undo attention. *Interacting with Computers*, 4 (1992), 317-342.

2. Archer, Jr., J. E., Conway, R. and Schneider, F. B. User Recovery and Reversal in Interactive Systems. *ACM Trans. on Programming Languages and Systems*, 6 (1984), 1-19.

3. Berlage, T. A selective undo mechanism for graphical user interfaces based on command objects. *ACM Trans. on Computer-Human Interaction*, 1 (1994), 269-294.

4. Bezerianos, A., Dragicevic P. and Balakrishnan, R. Mnemonic Rendering: An Image-Based Approach for Exposing Hidden Changes in Dynamic Displays. In *Proc. UIST 2006*, 159-168.

5. Cass, A. and Fernandes, C. Using Task Models for Cascading Selective Undo. In *Proc. TAMODIA 2006*, Springer (2007).

6. Cass, A. G., Fernandes, C. S. T. and Polidore, A. An empirical evaluation of undo mechanisms. In *Proc. NordiCHI 2006*, ACM (2006), 19-27.

7. Choudhary, R. and Dewan, P. A general multi-user undo/redo model. In *Proc. ESCSW 1995*, Kluwer Academic Publishers (1995), 231-246.

8. Grossman, T., Matejka, J. and Fitzmaurice, G. Chronicle: capture, exploration, and playback of document workflow histories. In *Proc. UIST 2010*, ACM (2010), 143-152.

9. Haller, M., Leitner, J., Seifried T., Wallace, J., Scott, S., Richter, C., Brandl, P., and Gokcezade, A., The NiCE Discussion Room: Integrating Paper and Digital Media to Support Co-Located Group Meetings. In *Proc. CHI 2010*, ACM (2010), 609-618.

10. Hazemi, R. and Macaulay, L. User requirements for undo support in CSCW. In *Proc. of the HCI'95*, Cambridge University Press (1995), 181-193.

11. ISO, EN ISO 14738, Anthropometric requirements for the design of workstation machinery

12. Koffka, K. *Principles of Gestalt Psychology*, Harcourt, 1935.

13. Kurlander, D. and Feiner, S. *A history of editable graphical histories*, MIT Press, Cambridge, MA, USA, 1993, 405-413.

14. Kurlander, D. and Feiner, S. A Visual Language for Browsing, Undoing, and Redoing Graphical Interface Commands. *Visual Languages and Visual Programming*, Plenum Press (1990).

15. Li, R. and Li, D. A Regional Undo Mechanism for Text Editing. In *Proc.IWCES 2003*.

16. McGrath, J. *Groups: Interaction and Performance*, Prentice-Hall, Englewood, NJ, 1984

17. Meng, C., Yasue, M., Imamiya, A. and Mao, X. Visualizing Histories for Selective Undo and Redo. In *Proc. APCHI 1998*, IEEE (1998), 459-464.

18. Mynatt, E. D., Igarashi, T., Edwards, W. K. and LaMarca, A. Flatland: new dimensions in office whiteboards. In *Proc. CHI 1999*, ACM (1999), 346-353.

19. Nakamura, T. and Igarashi, T. An application-independent system for visualizing user operation history. In *Proc. UIST 2008*, ACM (2008), 23-32.

20. Prakash, A. and Knister, M. J. Undoing actions in collaborative work. In *Proc. CSCS 1992*, 273-280.

21. Ressel, M. and Gunzenhäuser, R. Reducing the problems of group undo. In *Proc. SIGGROUP* 1999, 131-139.

22. Ressel, M., Nitsche-Ruhland, D. and Gunzenhäuser, R. An integrating, transformation-oriented approach to concurrency control and undo in group editors. In *Proc. CSCW 1996*.

23. Saund, E., Fleet, D., Larner, D. and Mahoney, J. Perceptually-supported image editing of text and graphics. In *Proc. UIST 2003*, 183-192.

24. Sun, C. and Chen, D. Consistency maintenance in real-time collaborative graphics editing systems. *ACM Trans. on Computer-Human Interaction*, 9 (2002), 1-41.

25. Sun, C. Undo any operation at any time in group editors. In *Proc. CSCW 2000*, 191-200.

26. Sun, D. and Sun, C. Operation context and context-based operational transformation. In *Proc. CSCW 2006*, 279-288.

27. Teitelman, W., Goodwin, J. and Bobrow, D. Interlisp reference manual. *Xerox Palo Alto Research Centers*, 1978.

28. Vitter, J. US & R: A new framework for redoing. *IEEE Software*, 1 (1984), 39-52.

29. Yang, Y. Undo support models. *International Journal of Man-Machine Studies*, 28 (1988), 457-481.

30. Zhou, C. and Imamiya, A. Object-based nonlinear undo model. In *Proc. COMPSAC 1997*, IEEE (1997), 50-55.