Combining gestures and graphical elements for collaboration using multi-touch surfaces

Yazmin Magallanes,
Ariel Molina,
J. Alfredo Sánchez
Interactive and Cooperative Technologies Lab
Universidad de las Américas Puebla, UDLAP
Cholula, Puebla, México
{yazmin.magallanesvz, ariel.moliniara, alfredo.sanchez}@udlap.mx

Abstract — In this paper we report on early findings on how a set of gestures and graphical elements can be used as a means to introduce and manipulate ideas represented by images, text, and freehand drawings around a horizontal multi-touch surface in the context of collaborative processes. A fan-like menu that can be obtained from any point on the interactive surface is one of the key proposed graphical elements. We also propose a knowledge management system for this scenario, which allows for the organization of ideas into hierarchical conceptual structures. We also present two innovative solutions to challenging issues related with the undo/redo stack and the clipboard mechanism for interactive surfaces.

Keywords—gestures; graphical interface; collaboration; innovation; interactive; multi-user

I. INTRODUCTION

Innovation is the introduction of a new or significantly improved product (good or service), a new marketing method or a new organizational method in the internal practices of the company, workplace organization or external relations [1]. Tactile surfaces, such as interactive whiteboards or tables may be useful to support innovation processes. This is because they enable natural user interfaces and collaboration environments in which participants have a general view of what is being done. Collaboration within an innovation process is a complex activity involving idea generation, people and knowledge organization. A useful environment for idea generation, annotation and organization is still a challenge for Human-Computer Interaction (HCI) experts as well as software developers. Designing the ultimate multi-touch interface for collaboration and innovation is a difficult goal to accomplish if avoiding interference of the innovation process is also a must. This is because innovation processes are usually very dynamic and fast paced.

Design tasks often require careful control of the ways in which objects are manipulated [2]. By creating user interfaces for interactive surfaces, developers and researchers are faced with the question of which design processes can be used to build an efficient and usable system. Existing techniques are not taking into account the special needs and limitations of interactive surfaces [3]. We have been exploring ways to introduce natural user interfaces to support innovation processes. We have been particularly interested in designing interfaces for helping users to capture, represent and manipulate their ideas on interactive surfaces. We have been investigating the use of voice [4] and how it is used in conjunction with specialized gestures [5]. Through this work, we have realized the need to provide interface alternatives for interactive surfaces that involve flexible combinations of gestures and graphical elements. We have developed a model for this sort of interface and we have addressed issues that arise in actual usage settings. This paper reports on our findings in this regard.

II. RELATED WORK

Eight challenges that need to be considered when designing a user experience for multi-touch interfaces are discussed in [6] and how motivated our work: Affordances of screens, tactile user feedback, ergonomics, individual differences, accessibility, gestures and patterns, supporting data input and multi-user support.

Nacenta et al. [7] in their investigation found that interaction in a multi-touch surface can be very quick and natural, but also presents the problem of separability: one dimension (e.g. orientation) becomes difficult to control without affecting others (e.g. size). We are partic interested in exploring how this applies to horizontal surfaces.

Studies by Benko et al. [8] present a set of five techniques that help users perform a selection of very small elements using multi-touch devices. Vogel and Baudisch [9] propose a technique called “Shift”. This technique addresses the occlusion problems caused by using one’s fingers to interact with a multi-touch surface.

Nielsen et al. [10] described the algorithmic complexity of gesture recognition, learning curve, intuitiveness and ergonomics. Some gesture models are discussed by [11] and [12]. They refer to the inclusion of graphical user interface (GUI) elements such as menus or other widgets as a complement or alternative methods. Also, some software libraries provide recognition for hundreds of gestures. Findings show that many of them are not helpful for real world usage, e.g., GestureWorks (www.gestureworks.com).

III. THE INOVIMM MODEL

The most common approach utilized for interface elements and document edition on multi-touch interfaces still relies on the legacy desktop metaphor. While this works well for single user workflow, it has disadvantages for multi-user collaboration. The desktop metaphor typically assumes a single user and relies on non-concurrent access to user interface objects. We have derived our proposals for multi-touch applications from observing innovation groups during early stages of generation and organization of ideas. We found there is a void to be filled for multi-touch interfaces to
support collaboration and idea generation for innovation processes. Graphical interface elements and user gestures for these applications were carefully implemented to fulfill user needs in this scenario.

In order to test concepts, we have designed InnovIMM, a collaboration environment in which ideas can be captured and organized. InnovIMM is based on the use of three main components: cards, floating fan menus and toolbars.

A. Cards

In InnovIMM, a card is a workspace used to organize knowledge. Cards may contain various types of objects, for example: text, images, and freehand drawings. New cards can be linked so as to create a hierarchy of cards. Objects act as ancestor nodes in a pattern similar to other knowledge management systems such as KMS [13].

The card implementation has an extensible architecture and allows for additional plugins providing more object types in the future (even created by third parties). Some ideas for useful objects to implement include web search, video insertion, and even in-place calculators or other small applications.

B. Fan Menu

Fig. 1 illustrates the proposed fan menu. Users can pop up a fan menu by touching any empty space of a card, although some contiguous space is ignored in order not to overlap with existent menus. The pie menu is in fact a semi-pie and fan-like animations are added for cosmetics and for hinting the user about navigation to sub-menus. Menu and submenu elements are given different font sizes to distinguish between them. Also, as shown in Fig. 1, the user can go back to the main menu by selecting the top entry (“Insert”). Currently, we only use text for menu options, but, as we found in usability studies, additional graphical elements are desirable and will be added as future work.

Once the user selects an action, the menu is closed and hidden away. Currently available menu options are briefly discussed next.

1) Text

The submenu "Insert", contains the item "Text", which is useful for editing text using a virtual keyboard, as displayed in Fig. 2.

2) Images

At any time, any user can display the menu and select the menu item "Insert" followed by “Picture”. An image selector will be displayed as shown in Fig. 3 (left) so the user is able to insert any image from a database. Additional images can be inserted by repetition of these actions.

At that time the menu will hide and display a range image repository. The image can be inserted by a touch on it. This process can be performed as many times as necessary to insert images.

3) Freehand

If the user selects the option "Draw" - "Freehand", then the fan menu is hidden and a canvas to draw is displayed. The user can now draw their ideas.

C. Toolbar

In our model, the user can add and interact with several graphical elements, which may be images, text, drawings or canvas areas. When selected, each of these elements is enhanced with an additional interface element (a toolbar) that allows for the execution of common tasks such as “delete” (Fig. 4 bottom) and some object-specific actions, such as “lock” for the painting canvas. The toolbar is positioned at the bottom right of every object. Fig. 4 (top) illustrates the proposed toolbar and provides further details on the iconography and its meaning.

Establishing relationships between ideas and organizing them is very important in the process of innovation. In order to organize the information generated during the process we developed a method to create new cards and create links to them. This functionality was implemented as an element for the toolbar, and the corresponding action can also be performed with a gesture. The mechanism for link creation and navigating between links is similar to the KMS hypertext and knowledge management system [13]. In formal terms, any object can be the ancestor of another object via a link.
D. Gestures

We propose a companion set of gestures for performing some of the actions contained in the menu and some activities for the application we developed. The proposed gestures were targeted at innovation processes performed over a multi-touch surface. Fig. 5 summarizes our proposed set of gestures.

The following is a list of gestures proposed and tested during usability studies.

- **Make a link**: Tap twice to on an item to generate a new card linked to the current item.
- **Show menu**: Tap on any empty space of the card to pop up a fan menu.
- **Minimize card**: Perform a five finger pinch to minimize a card and go to the previous card.
- **Zoom**: Perform a two finger pinch. This gesture commonly used to increase or decrease the size of the digital elements.
- **Search**: Slide a finger to mimic a magnifying glass to look for elements of a website or ideas within the cards.
- **Delete**: Slide a finger in a zigzag pattern to delete items embedded in the cards.
- **Help**: Slide a finger as a question mark to find information about the set of gestures.
- **Move**: Slide two fingers to move digital elements (Fig. 5).

E. Prototypical implementation

Most of the graphical user interface was implemented using the Qt Modeling Language (QML) with exported elements designed in Adobe Photoshop. The internal architecture and a minimal amount of GUI elements were developed in C++ using Qt 4.7 (http://qt.nokia.com/). Although the proposed gestures were fully tested, they were implemented only partially. We used the Qt gesture framework for some gestures (like pinch-zoom and two-finger drag), the rest we carefully simulated using Wizard of Oz [14]. The application receives input events via the TUIO protocol with 60Hz latency. Final experiments were conducted on a Sony Vaio Laptop running Ubuntu 10.04 over an Intel Corei7 (8 cores) CPU and a multi-touch horizontal surface prototype from EDIS Interactive. The table has 80x60cm multi-touch area but is user-blind.

IV. OPEN ISSUES IN MULTI-TOUCH INTERFACES

Derived from our observations and formal evaluation with actual users (discussed in the following section), we found two issues that merit serious attention and worked on solutions that need to be included in multi-touch interfaces. These two issues have to do with undo/redo and copy/paste functionality, as described in this section.

A. Multi-user undo framework issues and a solution

In a single-user environment, the clipboard is for the exclusive use of the user and can never be overwritten or spontaneously modified; the unique user that is working on the system has complete control over it. Most current multi-touch displays make it possible for multiple users to interact with objects on the interface, but cannot distinguish the specific users who touch the interface or produce a given gesture. We refer to these multi-touch surfaces as user-blind, as they are tactile but remain unaware of the specific users who act upon them. Consider, for example, the scenario depicted in Fig. 6 (top). In this scenario a timeline is shown and users Anna and Bob are interacting with a multi-touch surface (which might be user-blind). Everything works fine during the first four time steps; in the example user Anna is creating a painting, erases something and begins to think about undoing this action in order to recover the painting. But before Anna executes the undo action, user Bob already painted something else. If the legacy undo/redo model is used when Anna executes the “undo” action, it would undo the action performed by Bob, as it was the last global action (Fig. 6 middle). Then Bob would be perplexed to see how his creation has suddenly disappeared. Anna, on the other hand, might think that her action was ignored and might even be tempted to repeat it until her work appears again, potentially destroying the work of several other members. This is a side effect of the legacy undo model being tied to the global timeline.

In order to address this problem, one approach is to make the surface user-aware such as [15]. It should be noted, though, that this approach requires additional devices attached to the user. These devices might require pre-calibration and may be cumbersome to wear. On the other hand, most of the current multi-touch surfaces discussed in scientific literature [16], even commercial offers are user-blind.
Some advances have been made on user identification for undoing actions in [19], but they tend to force the user to have work areas that must be dragged and to restrain undo actions to that area. Also, social protocols must be enforced in order to keep the user working over their own work area. Per-object undo has the potential to allow for greater freedom in object manipulation and history replaying without enforcing either work areas or social protocols. This makes us believe that our proposed solution is practical and substantially beneficial for the majority of current multi-touch surfaces and might even be applied for future surfaces.

B. Workaround for the clipboard copy/paste issues

The current metaphor for the clipboard and the traditional Copy, Cut and Paste actions is attached to the idea of a global buffer that retains an amount of temporal data for the purpose of creating copies of it or to move it around. Some extensions have been made for creating clipboard stacks that have a finite amount of slots to retain several buffers in a FIFO structure. But in general, the clipboard mechanism is the same as popularized at the Xerox PARC circa 1975 and then by Apple in the early eighties, which is strongly attached to the idea of a pointing device. This pointing device assumption is not valid in multi-touch devices, which do not have the notion of a pointer.

The scenario shown in Fig. 7 presents a multiuser surface where user Annie and Bob both create content on the surface (Fig. 7 top-left). If we follow the current clipboard mechanism and both Annie and Bob want to copy they must perform the “copy” action. The problem arises when multiple “copy” actions are performed sequentially because the latest “copy” overrides any current content in the clipboard. Say Annie copies first, then Bob copies afterwards, then Annie “pastes” the content. Annie was expecting her content to be pasted, but some random content was pasted, she might not be aware that Bob made a copy and the content she pastes seems random. Bob might paste and his content might or might not be the correct content he expects, depending if his “copy” was the last one or not. In a multi-user interactive surface there is simply no guarantee to have the clipboard behave as expected unless some social rules are enforced and strictly followed. However these social rules are easily broken and users can wreak havoc as discovered and reported by Fiebrink, et al. [20] when testing collaborative musical interfaces for multi-touch surfaces. They report “Users sometimes subverted the cooperative process by hitting ‘Copy’ […] without announcing this to the group. When more than one user acted independently in this way, without others being aware of their actions, this resulted in confusion”. They empirically discovered important aspects of the legacy clipboard that have to be solved by multi-touch interactive surfaces: concurrency and user-blindness. Further investigation on this matter is important in order to overcome current limitations.

In order to address this issue, illustrated in in Fig. 7 (top-right), we propose the use of an automatic duplication mechanism instead of the traditional copy/paste. If Annie wants to “copy” and then “paste” her content, and we want to avoid interruption and corruption of her data, the operation must be atomic. But atomicity means the action is now just one atomic “duplicate” action. The same is valid for Bob. The duplication mechanism uses an atomic duplicate and this ensures their content is correctly replicated and it even simplifies the copy/paste combo into a single action.
V. Evaluation

Usability studies were performed for testing user interaction and experiences with the proposed model. Our subjects, five female and nine male, were between 18 to 36 years old. Their education level ranged from undergraduate to graduate students. The Think Aloud [21] method and video analysis of recordings were used to study user behavior. Participants were instructed to perform 14 tasks to encourage object manipulation. Insertion of various types of content (ideas), edition, scaling, manipulation and card creation and navigation were among the activities on the tests. Users were encouraged to try both using the UI elements such as the menu and toolbars and direct gestural interaction.

A. Results

Fig. 8 and 9 summarize our results for our GUI and the proposed gesture model. The three best ranked GUI elements were (1) menu design, (2) freehand drawing and (3) image insertion. The application’s usefulness for idea generation and innovation activities also received positive comments, and so did the toolbar’s usefulness. On the other hand, some elements ranked less positively were some icons in the toolbar, specifically the undo/redo icons, which were some-times mistakenly identified as previous/next actions. When told about the actual actions performed by these elements, users were pleased with the per-object undo/redo framework although it was still not implemented or functional (it was not part of the study).

Some icons in the toolbar that led to some confusion were the “author tagging” and “link creation,” but we think these issues can be linked to the lack of familiarity of some of the subjects with the innovation methodology for which our application is targeted. The proposed gesture for “move” caused confusion as many users attempted to move objects by using only one finger. Some other minor technical errors in the menu were detected after detailed video analysis.

As noted in Fig. 9, the subjects evaluated most of the gestures as intuitive, including: help, delete, zoom, rotate, link card and show menu. We observed that multiple users were able to work smoothly on the per-object toolbar associated with every object. These are very encouraging results.

VI. Conclusion

In this paper we have introduced a set of graphical elements and a model of gestures to enable multi-user collaboration for innovation scenarios. Together, these tools represent a model to allow users to represent manipulate and generate ideas.

We developed a prototype for this proposal and we conducted several tests with potential users over the graphical elements and the gestures in order to know whether they are usable and if they have practical application on the proposed scenario.

So far, users have evaluated the proposed model as an intuitive and straightforward mechanism for collaboration and innovation processes, especially for the idea generation process. Positive user comments were confirmed with the evaluation results. We have demonstrated that our proposal of UI elements and gestures are useful for innovation processes on multi-touch surfaces, furthermore, most of the proposed gestures are easy to learn and remember as well as intuitive.

Only two-finger drag gesture for moving objects was evaluated as counterintuitive. Preference in general was using one-finger drag, as popularized by many touch interfaces like those in mobile devices. This is not difficult to solve and our current version now has single-finger drag.
Additionally, we encountered some difficulties for implementing the undo/redo stack as currently used by desktop computers. The same happened to the clipboard mechanisms. To address these two issues, we propose two solutions which are already part of the ongoing work. These solutions, which still need to be formally evaluated, are very promising in solving the undo/redo and the clipboard mechanism challenges. User experience studies also are part of the work we plan to undertake in the near future.

ACKNOWLEDGMENT

The authors would like to thank Efecto Digital Interactivo S.A. de C.V. (EDIS Interactive) for providing the hardware used for our studies. Support also has been provided by the CUDI Program.

REFERENCES