

Managing Ambiguity in Mode Inferencing

Keith Deming and Edward Lank

Computer Science Department

San Francisco State University, 1600 Holloway Avenue, San Francisco, CA, 94132

kademing@sfsu.edu, lank@cs.sfsu.edu

Abstract

The inferred mode protocol is designed to determine a user's intention in stylus interfaces with the goal of eliminating the mode problem. To accomplish this task, the system reasons using characteristics of a user's action and features extracted from image material to determine whether a user's gesture constitutes selection or inking. However, in many situations, uni-modal pen interfaces do not provide sufficient information to determine, unambiguously, a user's intention. In this work, we describe our on-going research in the use of additional modalities to eliminate ambiguities that exist in the inferred mode protocol. We describe systems that incorporate stylus orientation and stylus pressure into the determination of user intention. We outline the results of a series of pilot studies analyzing the usability of these techniques. As well, we describe prototype applications developed as a result of these user trials.

Introduction

Sketch applications are software programs which allow computer users, particularly tablet computer users, to create informal content by inking on a computer display. On-going work seeks to design sketch applications that are more usable.

To address one specific usability issue with sketch applications, Saund and Lank developed an *inferred mode protocol* [Saund and Lank 2003]. The goal of the inferred mode protocol was the elimination of mode errors in sketch interfaces. Mode errors in sketch applications are an artifact of the need to overload the limited input modality of a stylus to perform multiple tasks, specifically creating information (inking) and rearranging existing sketch material (editing). The inferred mode protocol examines a user's action and the underlying sketch content to infer a user's intention without the requirement to set a software mode. However, in some situations, the user's action is ambiguous. As well, initial user trials indicate that preference is evenly split between users who prefer a sketch application with manually set software modes, and our interaction mechanism which eliminates the need to set software modes.

Our on-going work seeks to analyze and understand mode errors and their prevention more fully. As one step

toward this, we describe a number of pilot studies that extend the inferred mode protocol. We have designed a series of interfaces that take advantage of the additional input modalities provided by the current generation of tablet computers. These modalities include the use of stylus orientation information and the use of pressure. We have integrated these modalities into the Inkscribe application developed by Saund and Lank [Saund and Lank 2003].

In this paper, we describe pilot studies on two interaction techniques designed to limit the ambiguity inherent in the inferred mode protocol. The first interaction mechanism is designed to contrast the use of mediators to handle ambiguity with the use of stylus orientation as an explicit mode switching operation. The second interface is designed to take advantage of pressure regions as a mechanism for extending the click-to-select interaction in Inkscribe.

In this paper, we first describe the Inferred Mode Protocol. We then describe our two variants of the Inkscribe application, including a set of pilot studies that examines the potential benefits and limitations of these techniques. Finally, we conclude by outlining our on-going research agenda.

Background

Several researchers have recognized the problems inherent in mode-based interaction. For example, Tesler, in his work on the Smalltalk system, describes a campaign to eliminate, or severely curtail, the use of modes in applications due to the need, on the part of users, to manipulate both content and the software system state [Tesler 1981]. Norman studied mode errors with the goal of identifying and mitigating their cause [Norman 1982]. Finally, researchers have worked on mechanisms to enable monitoring of software state by users, including the use of kinesthetic feedback [Raskin 1997, Mohamed and Fells 2002, Sellen et al. 1990], clear depiction of modes [Ryu 2002], the use of physical objects/tangible UIs as in the Smartboard system, and the use of specialized gestures [Kurtenbach and Buxton 1993, Pook et al. 2000, and the Apple Newton].

In the inferred mode protocol, Saund and Lank seek to prevent mode errors by eliminating the requirement that users first set software application mode before interacting with the system [Saund and Lank 2003]. To accomplish this, they analyze user interaction to determine likely user

intention. Only in the case of ambiguity is the user prompted for additional information to clarify his or her intention.

The Inferred Mode Protocol

Figure 1 depicts the reasoning diagram in Saund and Lank’s inferred mode protocol. In this diagram, we see that, given a user gesture, the reasoning process incorporates characteristics of the gesture, information about the underlying image material, and information based on the current state of the application to analyze the action. In many situations, intention can be inferred from this information. In others, ambiguity exists.

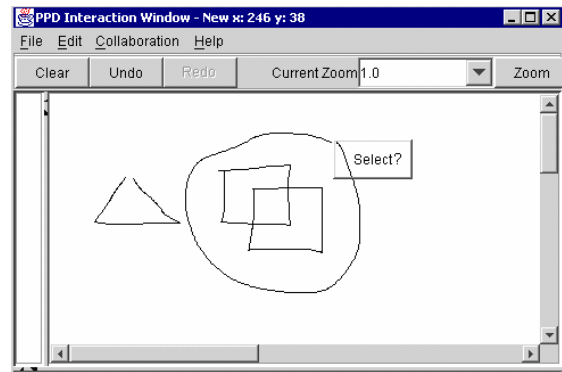


Figure 2: A mediator manages ambiguity in Inkscribe.

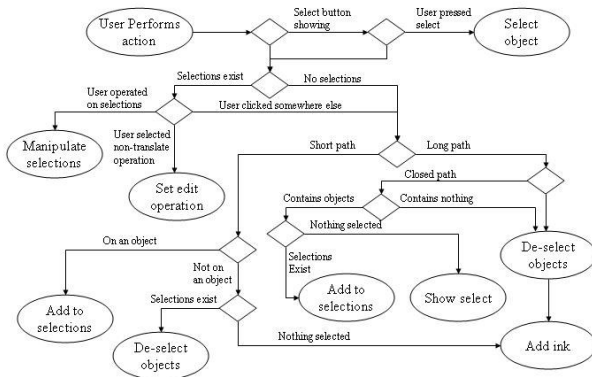


Figure 1: Inkscribe reasoning diagram.

Where ambiguity exists, the inferred mode protocol prompts the user for additional information. It does this through the use of a mediator [Mankoff et al. 2000] in the form of a pop-up button labeled “Select?”, as shown in Figure 2.

In their study of the inferred mode protocol, Saund and Lank conducted a pilot study where user preference was approximately evenly split between users who preferred the inferred mode protocol and those that preferred a traditional moded interface. Comments from those preferring the moded interface included a preference for the predictability of the moded interface (stated by an HCI researcher and test subject), and a dislike for software that tried to be too “clever” during interaction (stated by a former AI researcher, also a test subject).

While we might question the representative nature of the user population in the initial study, there seems to be clear opportunity for additional research into the inferred mode protocol. Specific goals of this on-going research include eliminating the ambiguity in interaction and increasing the predictability of the system for end users.

Prototype Interaction Techniques

In this section, we describe two systems for limiting ambiguity in selection in our interface. The first technique involves the use of stylus orientation as a mode indicator, while the second involves the use of pressure to determine the size of a selection region.

Stylus Orientation

In our first application, we introduce a method for determining users’ intent by using the two distinct ends of the stylus to indicate which mode is desired at any one time. We bind the point end of the stylus to draw mode, and the eraser end of the stylus to selection mode.

Application Design. Our system consists of a tablet PC that has a dual ended stylus running custom sketch software we have created. Our custom sketch software, hereafter referred to as StylScribe, takes advantage of the fact that a tablet PC can distinguish between the point end of a stylus and the eraser end of a stylus. Whenever the user brings the stylus in contact with the surface of the tablet screen, we poll the tablet for stylus orientation information and then set the mode accordingly.

This simple method for switching modes has at least two benefits over other methods. The first benefit of using stylus orientation to determine mode is that the average computer user is already familiar with dual moded qualities of a pencil (one can easily think of the graphite end as a drawing mode and the rubber end as erasing mode). By using this pre-existing familiarity, we reduce the cognitive load placed on the user that would otherwise be introduced by wholly new methods of interacting with pen based systems.

The second benefit of using stylus orientation to determine mode is the tactile feedback the user receives from the stylus. Texture, weight and position of buttons on the stylus all serve to remind the user of the current orientation of the stylus and thus allow the user to avoid making mode errors. In addition to how the stylus feels while simply being held, it also transmits additional tactile feedback to the user while gestures are being made. Where

the point end of the stylus glides rather easily over the surface of the tablet, the eraser end provides more resistance due to friction. This form of feedback allows the user to minimize the amount of time spent in the incorrect mode given that pen orientation was not noticed at the beginning of a gesture. This tactile sensation is most similar to kinesthetic feedback approaches that have been explored elsewhere in the literature [Sellen et al. 1990].

User Testing. We conducted a preliminary trial of a pen-base application that incorporates stylus flipping. Our prototype application, StylScribe, is built on the same platform as the inferred mode protocol application InkScribe [Saund and Lank 2003]. A reference version of the application using traditional modes was also created.

In this user trial, we test user preference for pen-flipping as a form of mode selection, contrasting it with both the inferred mode protocol and a fully moded system. The fully moded version used a conventional prior mode-switching paradigm in the form of buttons at the top of the screen. Apart from how users switch from one mode to another, all three systems were identical in interface and function.

Our user trial involved six users performing a predetermined set of tasks over a period of fifteen minutes. Each user was given a short demonstration of how each system works and given some sample tasks to perform in order to familiarize them with how the systems work. The order in which the systems were presented was varied to avoid biasing the results. During the user trial, the users were asked to recreate some content we had provided for them. Once the provided content had been copied to the screen, we asked the users to rearrange it and add more content. A sample of what was asked of the users follows:

1. Transcribe a series of numbers given.
2. Rearrange the numbers on the screen in to ascending sorted order, circling the even numbers as you go.

As the users completed their tasks on each system, observations were made to determine where users were having difficulty. After the users completed all tasks on all systems, they were given a short survey asking them to rate their experiences with each individual system and then to choose which system they preferred over the others.

There were four questions on the survey, and the set of four questions was repeated three times, once for each system. All the questions were to be answered on a scale of one to five. Following are the questions as written in the survey:

1. How easy to use did you find the pen input system?
2. How intuitive did you find switching from one mode to another?
3. Would you say switching modes caused errors?
4. How likely would you be to continue using this type of pen input system?

Users were asked to indicate agreement with the above statements for each of the systems (5 = most positive to 1 = most negative). Finally, a question at the end of the survey which asks the users to indicate which of the three systems they preferred.

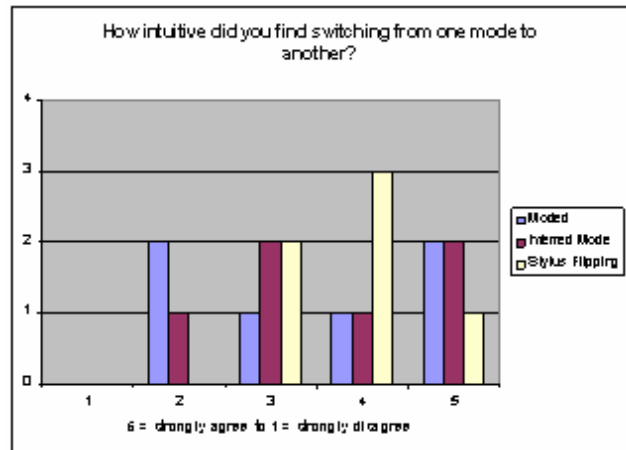


Figure 3: Results of survey answers for question two.

Results. The strongest result of our survey is that not one of our participants preferred the moded system to the other two systems. Three users preferred the stylus flipping system, two preferred the inferred mode protocol, and one user did not respond to the question of which system was preferred. However, the non-responsive participant did rank both stylus flipping and inferred mode higher than the moded system on every other question, which leads us to believe that, at the very least, the moded system is not preferred to the other two systems.

The histograms in Figure 3 summarize the responses for question 2 and in Figure 4 for question 4 for our six users.

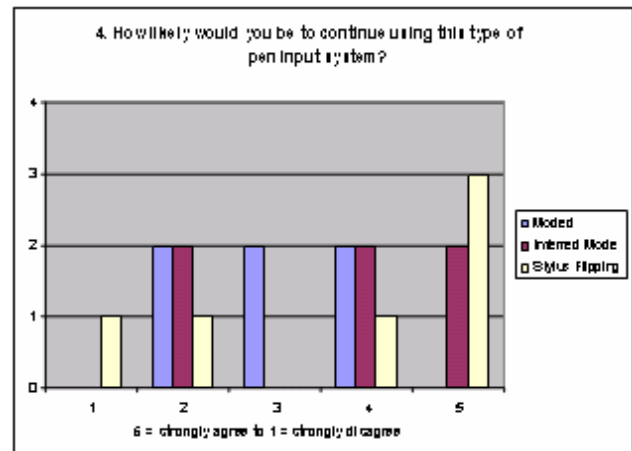


Figure 4: Responses to question four.

Qualitative Observations. While using the moded system, it is easy to see the frustration that arises as users find themselves in an unexpected mode. In order to avoid this frustration, many users will take a little extra time when they remember to switch modes and click the button for the desired mode several times. Saund and Lank noted similar behaviour in their earlier user trial [Saund and Lank 2003].

Both the inferred mode protocol and the stylus flipping paradigm avoid this frustration.

Pressure-Based Selection

One challenge in the design of inferred mode protocols is the overloaded looping gesture. This gesture can indicate either an encircling or a selection operation. In contrast, a user’s intention to select an object when he or she clicks on an object can be unambiguously recognized.

This fact motivated our interest in pressure as an input modality in stylus based systems. One challenge with click-based selection is the need to determine a selection region. One option is to select an individual stroke; another option is a group of strokes. However, systems typically recognize that many possibilities exist in selection region size. Systems commonly use multiple-clicks to select successively larger regions, but multiple clicks are more challenging for users on a tablet interface [Jarrett and Su 2003]. For example, for small objects retargeting can be challenging. As well, performing multiple clicks within a sufficiently short time frame can be more challenging with a stylus than with a mouse. We propose using pressure as a modality for selecting regions of varying size.

Our work is similar in nature to work by Kurtenbach and Buxton on the GEdit system [Kurtenbach and Buxton 1991]. They describe an interaction technique where leaving the stylus depressed allowed fine-tuning of the positioning of objects during translation. Both this work and our interaction technique were motivated by the observation that “continuity of motion and physical tension” can increase user performance [Sellen et al. 1990].

Pressure as a Modality. Before designing an application that made use of pressure as an input modality, we first ran a pilot study to gauge computer users’ comfort with varying pressure as a mechanism for controlling behavior in an interface. To that end, we constructed a software application that monitors how well a user interacts with a tablet PC using pressure.

The specific goal of our user trial was to determine the level of control users could exert over an application. Specific questions we wished to answer included:

- How many levels of pressure could individual users distinguish in an interface?
- Could users maintain pressure levels over longer periods of time?
- Could users maintain consistent pressure levels during stylus motion?

Our test application is pictured in Figure 5. In this application, three to ten different pressure levels could be pictured on the screen. A series of sliders allowed users to adjust thresholds between pressure levels themselves. Users started with three pressure levels and began a series of tasks. Users were asked to work upward in number of pressure levels from three to the highest possible point they could reach while still completing the tasks.

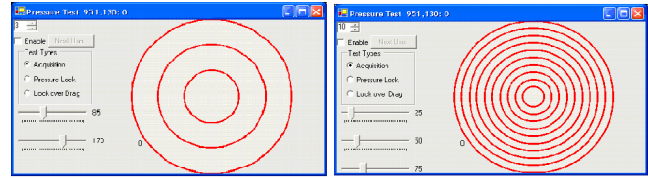


Figure 5: Our test application with three and ten levels of pressure.

To answer the above questions, we designed three tasks: acquisition, lock, and lock over drag. In the acquisition test, we tested whether a user could reach and maintain stably a particular pressure level with the specified number of thresholds for five seconds. The system indicates to the user that he or she is in the desired pressure level by coloring one, two, or three of the rings interiors in red. After five seconds the color switches to green, indicating successful completion of this test.

The second task, pressure lock, is designed to test a user’s ability to maintain an acquired pressure level over an extended period of time. We asked the user to hold pressure for an additional five seconds to determine how stable an individuals sense of applied pressure is, as well as how much fatigue and strain is involved in using pressure as a modality.

The final task asked users to drag while maintaining a specified pressure level. We measure how far the user is able to move before falling out of the specified pressure region.

User Trial Results. We conducted a pilot study on the ease of use of pressure as a modality in pen systems for ten users. All users could accomplish the tasks with three pressure levels, and three users could actually perform tasks with all ten possible pressure levels. The number of users for each pressure level is summarized in Figure 6.

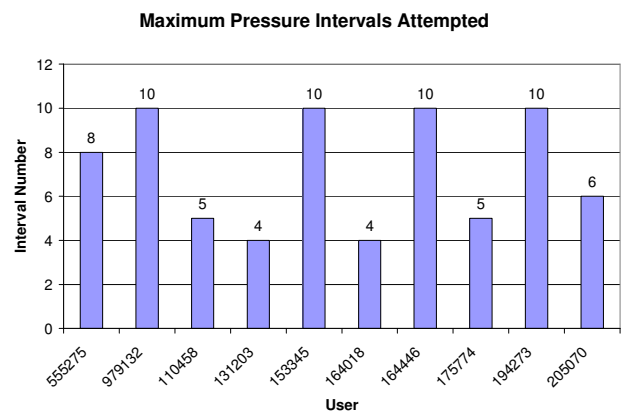


Figure 6: Number of intervals attempted. Three of the users completed the tasks with 10 intervals.

The number of intervals attempted represents the point at which a user stopped. One of the users who stopped with ten pressure levels could not acquire or lock on the pressure; the other three actually completed all tasks with ten pressure levels. It may be interesting to note that those

users who did continue with the experiment until all tasks had been completed seemed to be spurred on by the game aspect of the application. Fatigue and frustration could not overcome some users' desire to "complete" the software trial. It is possible that this response to the software application skewed the results toward more people completing more trials at higher levels of division. By examining the graph in Figure 4, one can easily see a skew toward the higher number of pressure intervals, with four out of ten participants continuing until they reached the maximum number allowed by the application.

We also measured the distance users could drag while locked. As users began dragging, the system would monitor the pressure level. When it moved outside the upper or lower threshold, it would capture and store the distance traveled. We note that users could not hold pressure stable while moving, and that some users only moved an average of four or five pixels before losing their pressure lock. Users could drag up to a total of 1024 pixels, but, as shown in Figure 7, users typically performed poorly.

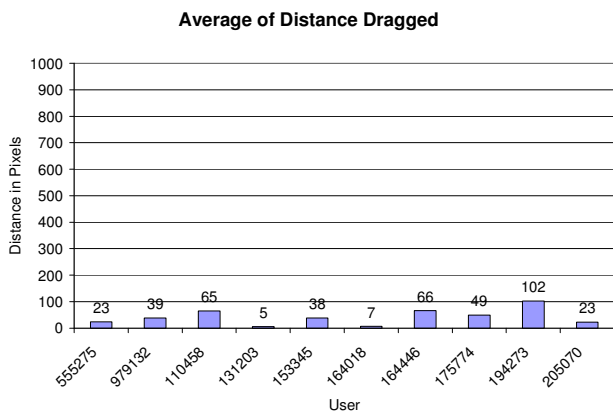


Figure 7: Results of dragging while maintaining pressure. Users were instructed to drag the stylus in a straight line, from one side of the screen to the other. Maximum value is the approximate screen width in pixels.

To measure user performance, we also examined number of attempts it took users to acquire a pressure level at various levels of granularity. This metric measures the number of times users attempted to acquire and then missed the pressure level they were targeting. We see, in Figure 6, our measure for users at three levels of pressure. Figure 7 depicts number of attempts for five levels of pressure, and Figure 8 depicts number of attempts for ten levels of pressure. Note that the outlier in Figure 8 was unable to complete the task.

Qualitative Observations. Initially all users had trouble finding an maintaining pressure levels in the interface. However, after three to four minutes of experience with the system, users developed the ability to acquire, lock, and maintain the lock for pressure levels.

One challenging aspect of the task was the tendency for the stylus devices to "stick" slightly at high or low pressure readings and then to adjust suddenly. The lag was an aspect of the system users had to learn to account for, and undoubtedly effected the time required to develop expertise with the interaction technique. This was true of both Acer and Wacom stylus devices used in the user trail.

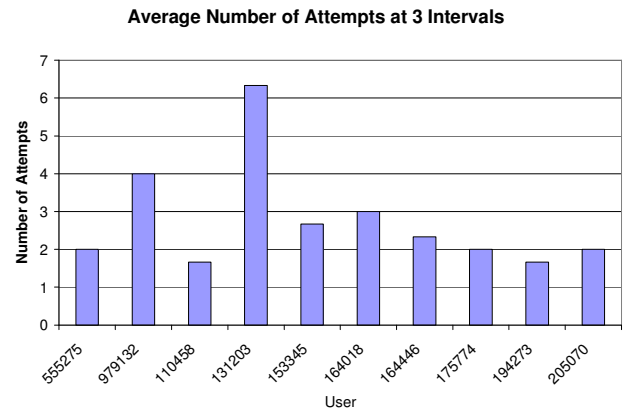


Figure 8: Average number of attempts per user for three pressure levels.

Users found the stylus tip much more difficult to use as compared to the eraser end on dual ended stylus devices. There appear to be two reasons for this. First, the eraser end on a dual ended stylus permits a significant amount of compression (more than 0.25mm). This allowed users to judge their pressure based on eraser compression. In contrast, the stylus tip has much less travel, on the order of 1mm. The limited travel prevented the use of compression as a judge of pressure for stylus tips. Second, using both Acer and Wacom styluses, the eraser tip permitted pressure readings of any value between 0 and 255 (the minimum and maximum respectively), whereas the stylus tip only allowed pressure readings of between 0 and approximately 110. This limited range made it slightly harder for users to work with the different pressure levels.

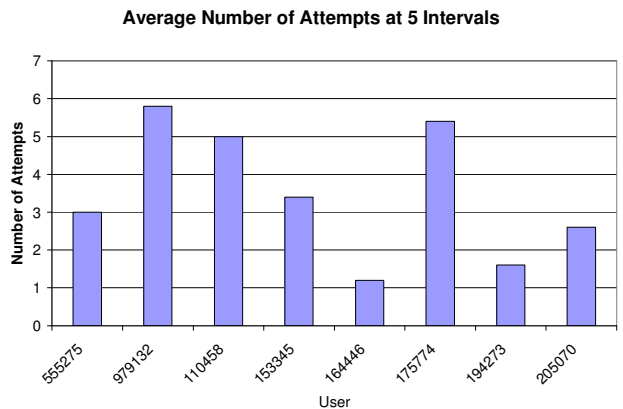


Figure 9: Average number of attempts per user when working with five levels of pressure.

System Design. Our final interface for pressure-based selection incorporates three pressure levels into selection. These pressure levels correspond to stroke, character, and paragraph/line level selection regions.

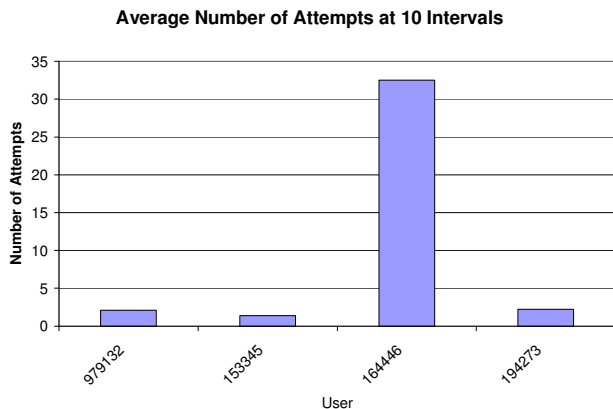


Figure 10: Average number of attempts with ten levels of pressure (for those users reaching ten). The outlier could not acquire a specific pressure level.

Conclusions and Future Work

In this paper, we describe our on-going work on the inferred mode protocol. We have designed two variants of the inferred mode protocol where we incorporate additional modalities provided by dual ended tablet PC styluses into systems that allow fluid switching between inking and editing modes.

Our first technique eliminates intelligent discrimination between inking and editing. In this application, stylus orientation is used to determine whether a user wishes to edit or draw. When the stylus tip is in contact with the surface, we assume an inking mode. When, instead, the eraser tip is in contact with the surface, we assume an edit mode. Our current application uses gesture recognition techniques to discriminate between selection (based on a recognition of encircling and click-on actions) and erase actions (based on the recognition of scratch-out gestures performed with the eraser). Our early trials indicate that this mechanism for interaction may be marginally more intuitive for users. However, both the inferred mode protocol and stylus orientation are preferred to traditional mode-based interaction.

Our second technique incorporates the use of pressure into the determination of the size of a selection. Higher pressure is used to indicate larger selection gestures. Based on our pilot studies, users distinguish between three levels of pressure (soft, medium and hard) relatively easily. However, performance degrades above three levels of pressure for most users. We have incorporated the use of pressure-based selection into the stylus orientation technique.

Our current work is directed to two main goals. First, we are conducting laboratory studies of our prototype

applications, with the goal of tuning our inferred mode protocol variants. However, laboratory studies are a relatively impoverished mechanism for testing the usability of sketch-based systems. With this in mind, we are designing sketch-based drawing applications for Tablet PC platforms that incorporate our inferred mode protocol variants. We are also experimenting with an implementation of the inferred mode protocol in a Pocket PC drawing application, Fish-i-Pad, designed by Lank and Phan [Lank and Phan 2004].

Acknowledgements

The authors would like to acknowledge the support of PARC, Inc., which has made available the Inkscribe application to SFSU for non-commercial academic research purposes.

References

- Jarrett, R. and Su, P. *Building Tablet PC Applications*, Microsoft Press, 2003, ISBN 0-7356-1723-6.
- Kurtenbach, G. and Buxton, W., "Issues in combining marking and direct manipulation techniques", Symposium on User Interface Systems and Technology, UIST 1991, pp. 137 – 144.
- Kurtenbach G. and Buxton W., "The Limits of Expert Performance Using Hierarchic Marking Menus", INTERCHI '93, 482-487.
- Lank, E. and Phan, S., "Focus+Context Sketching on a Pocket PC", Extended Abstracts of the ACM Conference on Human Factors in Computing Systems, CHI 2004, pp. 1275 – 1278.
- Mankoff, J., Hudson, S. E., and Abowd, G. D., "Interaction techniques for ambiguity resolution in recognition-based interfaces", Symposium on User Interface Systems and Technology, UIST 2000, San Diego, pp. 11 - 20.
- Mohamed F. and Fels S., "LMNKui: Overlaying computer controls on a piano controller keyboard", CHI 2002, 638-639.
- Pook S., Lecolinet E., Vaysseix G., and Barillot E., "Control menus: execution and control in a single interactor", CHI 2000 Extended Abstracts, pp. 263 - 264.
- Raskin J., "Looking for a Humane Interface", CACM, 40:2, 1997, 98-101.
- Ryu H., "Will it be upper-case or will it be lower-case: can a prompt for text be a mode signal", CHI 2002, 836-837.
- Saund E., Lank E., "Stylus Input and Editing Without Prior Selection of Mode", UIST 2003, November 2003.
- Sellen A., Kurtenbach G., and W. Buxton, "The Role of Visual and Kinesthetic Feedback in the Prevention of Mode Errors", Proceedings of Human-Computer Interaction, 1990, 667-673.
- Tesler L., "The Smalltalk Environment", Byte, August 1981, 90-147.
- Engelmore, R., and Morgan, A. eds. 1986. *Blackboard Systems*. Reading, Mass.: Addison-Wesley.