Squeeze Me, Hold Me, Tilt Me! An Exploration of Manipulative User Interfaces

Beverly L. Harrison, Kenneth P. Fishkin, Anuj Gujar, Carlos Mochon*, Roy Want

Xerox Palo Alto Research Center
3333 Coyote Hill Road, Palo Alto, California, USA 94304
{beverly, fishkin, agujar, want}@parc.xerox.com, carlosm@mit.edu

ABSTRACT
This paper reports on the design and use of tactile user interfaces embedded within or wrapped around the devices that they control. We discuss three different interaction prototypes that we built. These interfaces were embedded onto two handheld devices of dramatically different form factors. We describe the design and implementation challenges, and user feedback and reactions to these prototypes. Implications for future design in the area of manipulative or haptic user interfaces are highlighted.

KEYWORDS: Physical, tactile, and haptic UIs, pressure and tilt sensors, UI design, interaction technology.

INTRODUCTION
Over the past 5 years there has been increasing interest in augmented reality and physically–based user interfaces [4, 6, 7, 8, 10, 12, 15, 16, 17]. A goal of these emerging projects is to seamlessly blend the affordances and strengths of physically manipulatable objects with virtual environments or artifacts, thereby leveraging the particular strengths of each. Typically, this integration exists in the form of physical input devices (e.g., "phicons" [7], "bricks" [4]) virtually linked to electronic graphical objects. Manipulation of the physical objects signals a related operation on the associated electronic objects. This mapping is further reinforced by tightly coupling the placement of the physical objects relative to the electronic objects on a flat table-like display surface.

Another approach has been to use standard monitors or even stereoscopic monitors with more realistic input devices [6, 8]. In these cases, unique physical input devices are cleverly matched to the requirements of the specific application domain (e.g., MRIs, remote telerobotic control). The affordances of the input devices are well matched to the virtual representation of the object that they represent. Designers of commercial video games have been taking such an approach to user interface manipulation since the invention of the Data Glove™ and, more recently with such games as flight simulators and car racing, where the UI is controlled by steering throttles or steering wheels. Again, in these examples a specialized input device controls a separate electronic display.

These extensions to graphical user interfaces seem logical in view of the widespread support and acceptance of direct manipulation interfaces [11] and of real-world metaphors, such as trash cans and file folders [12]. We believe that such physical user interface manipulators are a natural step towards making the next UI metaphor the real world itself: real objects having real properties that are linked to or embedded in the virtual artifacts that they control. Furthermore, we conjecture that this metaphor reflects a powerful, largely unexplored user interface paradigm.

We set out to further explore this new area. We have been influenced by several previous research prototypes that reflect elements of an "embedded physicality" approach. Fitzmaurice [3], Rekimoto [9], and Small & Ishii [12] attached sensors to small handheld displays and subsequently used these displays to "scroll" through or view a larger virtual space. Movement of the display is mapped to corresponding movements in the virtual space, such as changing the view perspective [9] or to the degree of magnification [12]. These prototypes demonstrated the intuitiveness of this embedded physicality approach. The work we report here incorporates manipulations different from these previous examples to further improve our understanding of the breadth and potential of these new kinds of interactions.

Our work differs from the previous work on "physical handles" in one particularly interesting way. We are investigating situations in which the physical manipulations are directly integrated with the device or artifact, such as a small PDA, that is being controlled. We are not exploring separate input devices, but rather making the physical artifact itself become the input device by means of embedded sensor technologies.

The goal of this paper is to share our experiences in designing, building, and using several prototypes of such user interface techniques and devices and in reflecting on new ways for thinking about this new class of user interface.

Permission to make digital/hard copies of all or part of this material for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication and its date appear, and notice is given that copyright is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires specific permission and/or fee.
CHI 98 Los Angeles CA USA
Copyright 1998 0-89791-975-0/98/4..$5.00

* Intern at Xerox PARC from the MIT Department of Physics.

Petitioner Samsung Ex-1038, 0001

CHOOSING TASKS

We chose three diverse and previously unexamined user tasks. This allows us to explore alternative kinds of manipulations, test different sensing technologies, and more thoroughly probe the research space. Additionally, we selected tasks that represented two different types of interaction: active user interaction via explicit physical manipulations or via passive user interaction, sensed implicitly. Finally, we selected tasks that were relevant for other PARC research groups who were implementing applications for portable document devices [10]. For this reason, we focused on portable pen-based systems.

By implementing new user tasks, we hope to contribute to the general body of knowledge about physically manipulatable interfaces. We believe that this experience will assist us in formulating a more general framework, design principles, and theoretical foundations for physically afforded interfaces [2].

We chose several simple tasks: navigation within a book or document, navigation through long sequential lists, and document annotation. In the next section, we describe manipulation of real world, traditional artifacts and outline the task representation, user actions, and feedback for each of our selected tasks. Following this, we describe our three task UI designs in terms of how these real world manipulations were mapped to the devices we selected. Again we discuss our designs in terms of task representation, user actions required, and feedback. We then highlight some of the implementation decisions and tradeoffs that impacted the interaction design. Finally, we discuss feedback from the informal evaluations and interviews conducted thus far and the implications for future work.

Navigation within a Book or Document

The task representation assumes that the book or document has a number of predictable properties. These include physically manipulatable page units, a sequential organization, a thickness or "extent", and obvious start/end points. These properties afford page-by-page manipulation and movement through blocks of pages relative to the start/end points of the whole book or document. The user actions we looked at were flicking corners of pages (for page-by-page navigation) and thumbing into a location of the book/document by moving blocks of pages relative to the beginning or ending. Manipulation of these traditional artifacts provides feedback in the form of visual cues (pages move or "animate", new destination page shows, new relative location shows), auditory cues (the sound of pages turning), and kinesthetic cues (tactile pressure on finger or thumb, tactile feedback of pages moving or sliding).

Navigation through Sequential Lists

Generally, users conceptualize lists in different ways than books or documents (though similar navigation techniques could be used). We decided to use a Rolodex listing of index cards for this list navigation task. In this case, the task representation assumes physically manipulatable items or cards, a circular sequential organization, and a knob that controls the Rolodex. User actions are manipulation via turning the knob (with a rate parameter) and stopping at a desired location. Visual feedback includes the flipping of items or cards, the rate of flipping, and a new destination item or card. Auditory feedback is the sound of the cards flipping. Kinesthetic cues include finger pressure, extent of rotational movement, and direction of rotation.

Annotation and Handedness Detection

We defined this task as hand written annotation on a page-by-page basis (i.e., one page at a time), where the original page contains margins and white space within which the annotations are made. User actions are typically bi-manual: the non-dominant hand anchors the page while the dominant hand writes the annotations wherever there is room. Visual feedback is the appearance of the annotations. There is minimal auditory feedback. Kinesthetic cues are the pen pressure in the hand, anchoring pressure in the non-dominant hand, and the pen/writing movement and friction.

This task is of particular interest to us in that we introduced new capabilities not available in the real world annotation task. As such, it represents an opportunity for computationally enhanced interaction. In traditional page annotation users must fit the annotations into existing, limited white space based on the static layout of the page. During annotation, their writing hand also often obstructs the text that they are commenting on. We decided to optimize annotation by maximizing the amount of white space and its position within the page. We detect the handedness of the user and then dynamically shift the text "away" from the writing hand thereby maximizing white space directly under that hand (see Figure 4 bottom). We describe this design and the implementation of it in subsequent sections.

SELECTION OF DEVICES

Our design criteria were that the devices chosen would be handheld, support pen-based input, allow serial port input (for sensor communication), have a development environment for custom applications, and be cost effective (since we anticipated embedded hardware). Ideally, we wanted several devices with different form factors.

We decided to use two different portable devices to test our manipulations. We chose to use a handheld computer for the page turning and handedness detection manipulations (a Casio Cassiopeia™). For the list navigation manipulations, we chose a Palm Pilot™. Clearly, a number of other devices could have been selected – these two were chosen for their ubiquity.

DESIGNING THE INTERACTION

Navigation within a Book or Document

This task was divided into several simple, common subtasks: turning to the next page, turning to the previous page, and moving forward or backwards in large "chunks" relative to the beginning and ending of the document.

Page-by-page navigation

We now had to decide on how these tasks would be accomplished by a user – we tried for manipulations similar to those used in traditional artifacts. As the Cassio™ dis-

Petitioner Samsung Ex-1038, 0002

played individual pages with a sequential ordering, we decided to use a flick on the upper right corner from right to left to indicate "forward one page". A flick on the upper left corner from left to right would indicate "back one page". These actions were highly similar to the real world actions. Visual feedback was similar; pages changed (without animation), and the new destination page and page number became visible after the user action. After a page turning manipulation, both the page number and the contents change to reflect either the preceding or next page, depending upon the direction of the stroke. However, we did not implement sound effects in this iteration and some kinesthetic feedback was lost (notably the friction of pages sliding). Figure 1 shows a "real-world" page-turning gesture (top), and the implemented interface to the page-turning command (bottom) on the Cassio™.

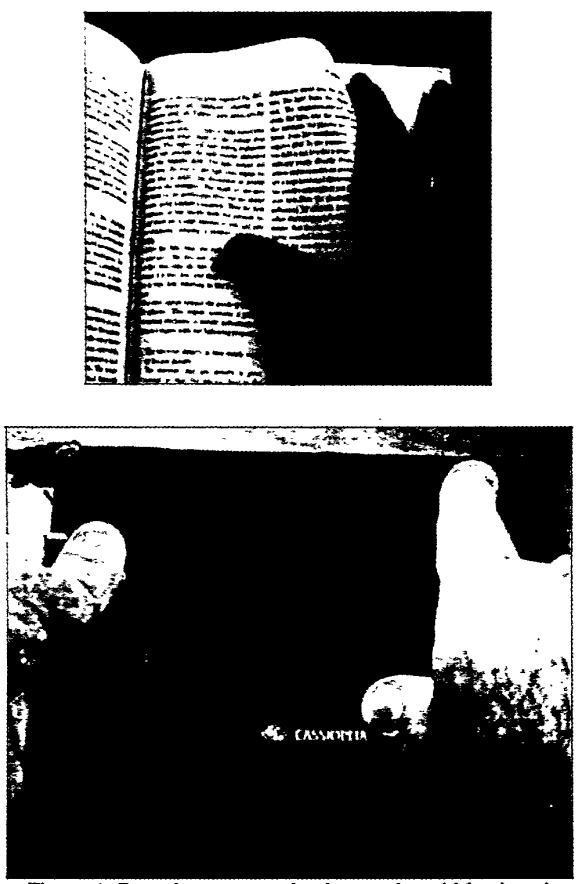

Figure 1. Page-by-page navigation, real-world (top) and with the prototype (bottom)

This interaction requires that the left and right upper corner detect a finger press, the direction of a stroke, and a release of pressure. Several implementation options are possible. Within each application where document reading occurs, a touch sensitive display can detect pressure points and their origin, determine if this aligns with a document upper corner, track the path of pressure to determine the stroke direction, and execute the appropriate page turn. Alternatively, the surface of the device can have pressure sensors attached to it, which detect when they are pressed, detect the direction of pressure from a stroke, and have the active application respond appropriately. We decided to try this approach since this allowed us to "retro-fit" pressure-sensing technology onto a normally pressure-insensitive device. Also, we did not need to use valuable screen real estate for the large area graphics needed to display a finger operated button. Finally, this would provide us with opportunities to later use the sensor technology in other application contexts and across applications.

Navigation by relative position

The extent and start/end points were not obviously represented (the thickness of the Cassio™ was invariant and too narrow for relative positioning). Hence moving forward or backward by chunks relative to the beginning or ending of a document was more difficult to represent for virtual documents. We decided to use a grasping manipulation at the top of the device, where the relative position of the grasp determined the relative position within the document. Far left corresponded to page 1 and far right corresponded to the last page. While this was not tightly analogous to known real world metaphors, it appealed to the well-known GUI metaphor of the scroll bar. A grasp gesture will move to a new location in the document and display the new location's page number and contents.

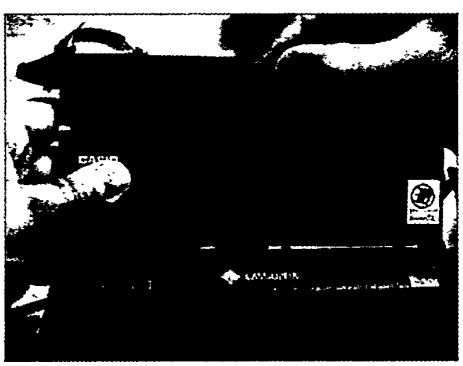

Figure 2. Navigation by relative position

Based on our chosen representation for moving by "chunks" and the corresponding user action, we again decided to use pressure sensors. To detect a grasp with the thumb and a finger, pressure strips were attached along the front top edge of the device. Grasping any portion of the strip moves to a document position relative to the beginning or ending of the document, where the beginning maps to the far left of the strip and the end maps to the far right of the strip. For example, Figure 2 shows a "grasp" gesture moving the display to roughly 2/3 of the way through of the document.

Navigation through Sequential Lists

We used a Rolodex metaphor–based technique for list navigation (see Figure 3, top). The circular list is manipulated by turning the Rolodex, while the direction of the turn determines whether the cards flip from A to Z or from Z to A. Our device-embodied representation was similar to the

Petitioner Samsung Ex-1038, 0003

real world artifact in that we used items with visual tabs arranged in a sequence. Turning the circular list towards the user would begin flipping through from A towards Z (assuming an alphabetized list) and vice-versa. On a physical Rolodex, users turn the knob rotationally (at some rate of speed) (Figure 3, top). On the Palm Pilot™, the user action was in fact a tilt movement away from a neutral resting position and not a rotational turn of a knob (this would be more akin to rotation of the entire Rolodex). Instead of having a rate or speed of turning we used the extent or degree of tilt (Figure 3, bottom). Turning "harder" (i.e., to a larger extreme) moves faster through the list, similar to Rekimoto [9]. To stop at or select a particular item, the user either ceases to tilt (i.e., maintains the list container in a neutral or vertical position relative to that item), or squeezes the device, mimicking a grasping gesture (akin to grasping the Rolodex card).

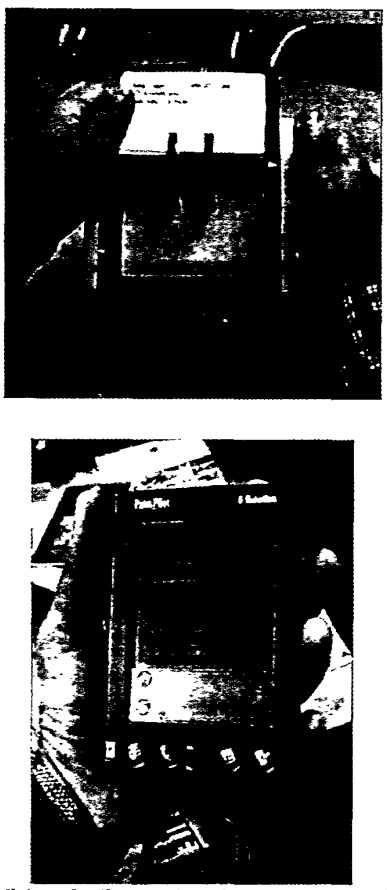

Figure 3. list navigation, real-world, knob rotation (top), and with the prototype, device tilt (bottom)

Annotation and Handedness Detection

Finally, we consider the task of optimizing annotation through maximizing "appropriate" white space and maximizing text visibility by detecting handedness. Sensing handedness means that text or graphics would be moved towards the non-dominant hand while screen space would be maximized on the opposite side (next to the hand holding the stylus). This strategy is appropriate for maximizing legibility of the text while holding the stylus and annotating adjacent to the text (Figure 4, top). In general, we were also interested in exploring different unobtrusive mechanisms for determining handedness.

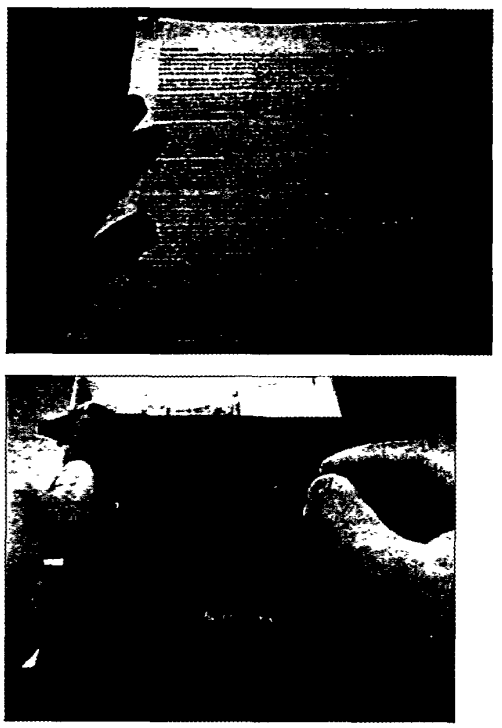

Figure 4. Annotation, real-world (top) and with the prototype (bottom)

For the handedness detection task we needed to understand something about how users hold and operate the intended device. We designed this such that no special manipulation was needed other than picking up the device and/or stylus. (i.e., "passive user interaction"). The handedness detection is immediately visible when the user invokes any application that wants to be "handedness-aware". In the specific task we implemented, for a right-handed user, text is immediately left justified and annotation space remains at the right side of the text (Figure 4, bottom). The reverse is true for left handed users. When both hands are used or the device is set down (i.e., no hands), the text appears centered. Feedback from the annotation remains consistent with the real world case; ink trails appear as the pen writes.

Finally, we examined sensor options for unobtrusively determining handedness. Several implementation paths were considered. Heat sensors on either side of the device could potentially detect whether contact with a hand occurred on the right or left side of the device (based on heat from the user's hand). However, this detection would be complex since physical extremities such as hands and feet generate heat levels comparable to many other environmental factors, including that of the device itself. Another alternative

Petitioner Samsung Ex-1038, 0004

was to detect properties of the writing which are unique to left handed or right handed writing styles. This is somewhat problematic since these algorithms are complex and the system can only take effect after the user has started writing. We decided to use pressure sensors again, this time to determine the points of contact and detect how the device was being held (if at all). Pressure sensing pads were attached to the back of the device, on the left and right sides, in alignment with positions used for holding the device.

IMPLEMENTATION

We now focus on the implementation details and issues that directly impacted the user interface and interaction design.

Navigation within a Book or Document

The Casio device was augmented with a network of pressure sensors. Two overlaid strips on the top edge detect the page turning manipulations (Figures 1 and 2). The pressure sensor network reports its current values through an interface connected to the RS232 port on the device. A simple communications protocol was devised, where each packet indicates the ID of the reporting sensor, and the current value of the sensor. Packets are only sent when the value changes. Absolute values, rather than deltas, are reported, so that we can recover from dropped/damaged packets. The document reading application runs as a multi-threaded application under Windows CE: one thread performs user I/O, while the other monitors the sensor stream.

To implement the page turning manipulations, two pressure sensors are overlaid at the top edge of the device. One type of sensor strip reports pressure, but not spatial location. The second type reports spatial location, but not pressure. Unfortunately, the spatial sensor tended to have a great deal of jitter. In order to compensate for this, two measurements were made from the spatial sensor: the first measuring the distance from the left end, the second measuring the distance from the right end. The sum of these two values should be a constant - if they differ too much from this constant, the values are rejected. Otherwise, the average of the two values is used. The {location, pressure} values are stored from the moment of pressure-down to pressure-up. If the sum of the inter-location differences is negative, the user is deemed to be stroking from right-to-left. If the sum is positive, the user is deemed to be stroking from left-to-right. If, regardless of this sum, the range of spatial locations is in a narrow range, the user is deemed to be pressing at a certain spot (i.e., the "grasp" gesture).

Navigation through Sequential Lists

In order to implement a tilt detection mechanism for continuous list scrolling on a handheld computer, we investigated a number of sensors. The commercially available tilt-sensor design we chose is based on an electrolyte bordered on two sides by a pair of conductive plates. As the device is angled towards or away from either plate, the amount of electrolyte in contact with the plate varies. The area of fluid in contact with each plate will affect the impedance presented by the contacts of the sensor. By monitoring this impedance and converting its change into a voltage, a simple ADC interface to a microcontroller can capture the data and then process it. In our system the tilt angle is converted into a 4-bit value and transmitted to the Palm Pilot™ across an RS232 link after being prefixed with the 4-bit sensor-ID, a total of 8 bits for each tilt sample.

By mounting a tilt sensor of this type to the case of a Palm Pilot™, with the sensor plates parallel to the plane of the display, we were able to use the sensor readings as a crude measure of the computer's orientation relative to gravity. We arranged it so that the Pilot generated a neutral reading at the 45 degree point and produced 8 readings forward and backwards from that position: 45 degrees being close to the angle that most people use to read from the display of the Pilot. Even though the range of angles detectable is thus very coarsely defined, we found that it has been adequate to implement and support the Rolodex–like metaphor.

In addition to sensing tilt, the system must differentiate between inadvertent tilting, such as when walking with it, and intentional tilting, when the user wishes to navigate. There are two possible ways of addressing this issue. The first method is to apply higher threshold values to the tilt sensing itself, thereby removing manipulations which are not of extremes and hence presumably retaining only deliberate user requests. This was infeasible in our desired application since we wished to use ranges of tilt to indicate the rate of list movement. Another possible solution is to create a second specific manipulation that indicates user intention. In our case, we decided to use an initial squeeze of the device to indicate the desire to navigate through the list, followed by a second squeeze to "grasp" the desired item, thereby ending the navigation task. To avoid muscle stress, users did not have to maintain the squeezing pressure during navigation. The device was padded with foam to further suggest squeezing capability.

To achieve the squeeze feature, we attached pressure sensors along both sides of the Palm Pilot™ in positions that aligned with the users' fingers and thumb (independent of which hand was holding the device). To differentiate squeezing from holding the device, we tested 10 users and derived an appropriate threshold value for the pressure sensors. (In this case, using higher pressure threshold values to differentiate inadvertent from intentional action was appropriate). The "squeezing" gesture has several advantages for users of a hand-held device. It doesn't require that the user reposition either hand, or that the users alter their viewing angle or viewing distance to the device, requiring only a momentary increase in the pressure used to hold the device. While the sensors report which finger(s) are exerting this pressure, at present our algorithms make no use of this additional information.

The list navigation task provides two levels of user feedback. Since the device is often moved about, the "tilt feature" is initially disabled. When users wish to navigate through lists they commence movement by squeezing the device. At this point the device is tilt-enabled. At present we have a message displayed indicating this to the users (it

Petitioner Samsung Ex-1038, 0005

says "Go Tilt!"). Clearly, a different message and a different means of conveying tilt-enabled would be better. Independent of this or any message, it is visually obvious when tilt-based navigation is enabled. Tilting works as described and users can see the list flipping through entries at varying rates of speed in the appropriate direction, depending upon the direction and magnitude of the tilt. The display ceases moving when the user either holds the device in the neutral position or again squeezes the device, thereby disabling tilt. (This is akin to grabbing the currently displayed item).

Annotation and Handedness Detection

Since these handheld devices are typically gripped by the edge with one hand, while they are used with the other hand, we detected handedness by mounting two pressure-sensitive pads on the back surface. If the pressure pad on the back left is pressed, the sensor thread of the program concludes that the user is holding the device with (at least) the left hand. If the pad on the back right is pressed, the program concludes that the user is holding the device with (at least) the right hand.

USAGE

A number of design issues arose as we iterated through the design and development of the prototypes. We did a number of in-laboratory, informal user tests to estimate threshold values for sensors. Typically this was done with our immediate research project group and a few other interested people (n=7). Once the sensor values were initially determined, we then carried out informal user testing and interviews on 15 different people outside our research project group. These users were fellow research staff who had little experience with physically manipulatable interfaces. Users were not instructed on how to hold the devices. They were given only brief descriptions of what the sensors would do (e.g., turns pages like you would in a paper document, tilting this moves through the Rolodex list). Following this, we observed them and recorded their comments. We asked them specific questions about what they expected to occur, what problems they encountered, and what they liked most and least.

General Comments and Impressions

In general, our test users found the manipulations "intuitive", "cool", and "pretty obvious in terms of what was going on." Some users needed quick demonstrations to understand that their manipulations would actually be interpreted. Our users had little or no exposure to physically embedded user interfaces and therefore often did not expect interaction with the device to be understood. Undoubtedly, conveying the basic paradigm will be necessary in the same way that users needed to understand the conceptual foundation for direct manipulation interfaces and mice. Once users understood the basic paradigm, they immediately began to explore the range of interaction. Just as GUIs users try to find out what is "clickable" by moving around the screen with the cursor and clicking, our test users tried a variety of manipulations on the prototypes to see what the range of detectable manipulations was. For example, to turn pages they tried long and short strokes, fast and slow strokes, light and hard strokes, and starting the stroke at different points on the device surface.

While the user explicit actions were quickly understood, the passive interaction (handedness) was perceived as "magical." Since no explicit commands or manipulations were needed, users seemed amazed that the device recognized and optimized for handedness. They were unable to tell how this was accomplished without us explaining it. This suggests not only that passive manipulations can be powerful, but that they greatly impact a user's interaction experience when well integrated with the form factor of the device. We clearly need to explore more passive manipulations to see if this is a general property. Additionally, this illustrates an opportunity for computationally augmented task representations that provide more than the real world analogy (in experientially positive ways).

Navigation within a Book or Document

Several interesting usage observations were made in these tasks. Because of our need to overlay pressure sensors, users now had to exert greater pressure than they expected for the page-turning manipulations. Users try out manipulations based on their expectations from the real world. A page turn in a paperback book, for example, takes very little pressure. All of our users initially attempted exactly the same manipulation on the device, which was too light to be sensed. However, they were able to quickly adjust with practice.

In general, we believe that users will attempt to exactly replicate the analogous real–world manipulation, when those metaphors are used; and they will expect them to work. If we are striving for enriched interaction experiences, the more exactly we can support or match these expectations the better. Making our sensors more sensitive to detect lighter page turning strokes would clearly be an improvement.

Users had no problem in discovering the manipulation needed for "previous page" once they had tried the "next page" manipulation. Despite slight differences in the pressure required over that of real–world interaction, users relied on extending their understanding of the real–world metaphor to guide their further assumptions about what was possible with the device–embodied interface. As in GUI design, small inconsistencies in metaphor seem to be forgiven.

Users needed to have the "navigation by chunks" mechanism described to them. Almost certainly this was because the device did not resemble a book, nor did the manipulation map directly to the manipulation in the real world. Grasping along a strip to indicate relative position is unique to this interaction. Once described or briefly demonstrated, users had no trouble remembering this manipulation or applying it.

One difficulty arose as a consequence of our implementation strategy. Since page turning strokes and grasping are both done on the same region and using the same pressure sensors, it was sometimes difficult to differentiate between

Petitioner Samsung Ex-1038, 0006

a very short "stroke" and a wide "grasp." Our disambiguating algorithm would sometimes compute this incorrectly, which would surprise the users. Since users' finger widths vary and we also want to support short strokes, this problem is not easily solved. We need to re-examine (a) whether there are better sensing technologies available or a different sensor configuration that would solve this, (b) whether minor alterations to the manipulations used would help differentiate the two, or (c) whether there is a better metaphor for navigation by chunks. (We have already implemented another version of page-by-page navigation on a tablet computer using 2 independent but adjacent small pressure sensors in the corners of the device. This seems highly reliable but makes integration with a stroke gesture (relative positioning) less seamless since pressure strips are not extended over these pressure pads to the edges of the device).

In general, the "navigation by chunks" task illustrates the tradeoff between intuitive real–world mappings which try to stay true to the real–world (and hence may be difficult to implement) versus learned mappings (which may be more easily integrated into the device). At this point, it is unclear how much learning is reasonable given that the overall goal is enriched interaction experience and intuitive interfaces.

Navigation through Sequential Lists

The list navigation task revealed some additional design issues (thus supporting our strategy of prototyping different device form factors, different manipulations, and different sensing technologies).

One issue was determining the range of angles for the tilt operation and the value for the neutral angle where the device remains in a resting state. We determined the initial neutral angle by in-laboratory testing. The range of tilt angles was partly based on just noticeable differences, both in terms of discernable tilt angles and in terms of discernable list scrolling speeds. The range of perceptible tilt is clearly an important determinant in setting and assigning values for the tilt manipulation's parameters. At present the 16 sensed tilt angles map to 6 different rates of scrolling.

One result of broader user testing identified the difficulty in stopping at a particular item within a list. Users would scroll quickly to the correct general area, then attempt to scroll slowly to the desired item. We now believe that our slow scrolling speed is still set too fast, as users tend to overshoot the target item. We believe that it is generally necessary to fine tune continuously issued gestural commands that control rate and/or direction of a corresponding action. We are investigating this issue further to determine how much individual difference amongst users effects their ability to precisely control list manipulation. This suggests that some "layered manipulations" may be useful, with one type of manipulation for coarsely specified actions, followed by a second manipulation for finely specified actions (as Guiard's bi-manual research suggests for handedness [5]). Trained parameterization or user-customized settings may also help alleviate this problem.

Finally, as a consequence of using tilt to control list navigation, display visibility was an issue. In particular, we avoided use of extreme angles of tilt, since the Palm Pilot™ display was not readable at these angles. Different devices and/or displays have different viewing angle restrictions with must be taken into account if the display is the primary feedback mechanism or if the display plays a central role in the task.

Annotation and Handedness Detection

The passive sensing used in the detection of handedness worked amazingly well. It detected and responded correctly, and users did not need to alter their usage of the device in any way from what seemed natural. All users remarked on the "magical" nature of this feature. We believe that the success of this feature is partly due to the in-laboratory, pre-testing we carried out. We tested 15 different users to fine-tune the placement of the pressure pads to accommodate different sized hands, slight differences in method for holding the device, and whether the left and right hand were used in exactly the same positions.

A possibility for the strong positive reactions to the seamingly "magical" feature was the augmentation of real-world capabilities. By optimizing annotation space, we have created a function that does not exist in the corresponding real-world scenario. In order to create computational *enhancements* (as opposed to unexpected and bizarre system performance), the system must accurately "know" what the user wants to do. These underlying assumptions and the subsequent matching of systems responses to user expectation are crucial. In this case, annotation optimization worked well because our assumptions accurately predicted user goals.

One aspect of this interaction that still requires adjustment is the changeover from one hand to another. If users momentarily shift the device from one hand to the other, the contents of the screen immediately move as well. Some users commented that seeing the screen contents jump around was disconcerting. However, it is unclear how often this scenario would arise in actual day-to-day usage. One improvement would be to briefly delay the screen change to determine that the user is not merely rapidly shifting the device to be more comfortable. Determining this time duration for detecting a "resting state" versus a "transient and temporary change" might improve the current interface.

FUTURE DIRECTIONS

In this paper we have explored the design, implementation and testing of three specific cases of physical manipulation prototypes. We discussed issues raised in designing and implementing these prototypes. And we have briefly outlined some of the results of user testing of these prototypes. Much interesting research remains to be done.

Our strategy is to implement other novel manipulations, new device-embodied task representations, and test new sensor technologies. We are interested in further exploring explicit manipulations as well as seamlessly sensed passive manipulations, with a goal of better understanding how this new paradigm can enrich the user's interaction experience.

Petitioner Samsung Ex-1038, 0007

This new type of interaction can be very dependent on the form factor of the device being augmented. We are interested in incorporating new manipulations into a number of different devices including tablet computers, conventional scanners, copiers, and monitors.

User expectation is often based on real–world experience and feedback, particularly if strongly analogous situations are represented. We would like to augment the simple visual feedback of the current prototypes to also include auditory feedback, animation, and potentially increased tactile feedback (though not necessarily forced-feedback).

We would also like to prototype some devices without displays and determine what kinds of manipulations and status indicators are possible and what types of computational power can be enhanced by such devices.

These all represent interesting research avenues for further systematic investigation. These should lead us to a better understanding of physically embedded user interfaces as a new paradigm, its limitations and strengths, and design principles to guide others exploring this area.

ACKNOWLEDGMENTS

We wish to particularly thank Tom Moran of Xerox PARC for his many substantive suggestions on the structure of this paper and for his extensive collaborative efforts in defining possible theoretical frameworks for this work [2]. We also thank the CHI reviewers for their helpful comments and suggestions. Our research additionally benefited from input given by members of the Portable Document Reader group at Xerox PARC, DpiX, and FXPAL. Finally, thanks to Rob Burtzlaff, patent attorney extraordinaire, without whom this paper might not have been released for publication.

REFERENCES

1. Card, S. K., Mackinlay, J. D., and Robertson, G. A. A Morphological Analysis of the Design Space of Input Devices. *ACM Transactions on Information Systems*, (2), April 1991, pp. 99-122.
2. Fishkin, K. P., Moran, T., and Harrison, B. L. Design Principles for Manipulative User Interfaces. Working Paper, December 1997. Xerox Palo Alto Research Center. Palo Alto, CA.
3. Fitzmaurice, G. Situated Information Spaces and Spatially Aware Palmtop Computers, *CACM*, *36*(7), July 1993, pp.38-49.
4. Fitzmaurice, G., Ishii, H., and Buxton, W. A. S. Laying the Foundations for Graspable User Interfaces. *Proceedings of CHI'95*, pp. 422-449.
5. Guiard, Y. and Ferrand, T. Asymmetry in Bimannual Skills. In *Manual Asymmetries in Motor Performance*, Elliot & Roy (eds), 1995, pp. 176-195. CRC Press, Boca Raton FL.
6. Hinckley, K., Pausch, R., Goble, J. and Kassel, N. Passive Real-World Interface Props for Neurosurgical Visualization, *Proceedings of CHI'94*, pp. 452-458.
7. Ishii, H. and Ullmer, B. Tangible Bits: Towards Seamless Interfaces between People, Bits, and Atoms. *Proceedings of CHI'97*, pp. 234-241.
8. Milgram, P. Rastogi, A., and Grodski, J. J. Telerobotic Control Using Augmented Reality. *IEEE Robot and Human Communication (RO-MAN)'95*, Japan, July 1995.
9. Rekimoto, J. Tilting Operations for Small Screen Interfaces. *Proceedings of UIST '96*, pp.167-168.
10. Schilit B. N., Golovchinsky, G and Price M. Beyond Paper: Supporting Active Reading with free-form digital ink annotations. *Proceedings of CHI'98.*
11. Schneiderman, B. The Future of Interactive Systems and the Emergence of Direct Manipulation. *Behaviour and Information Technology*, *1*, 1982, pp. 237-256.
12. Small, D., and Ishii, H. Design of Spatially Aware Graspable Displays. *Extended Abstracts of CHI'97*, pp. 367-368.
13. Smith, D. C., Irby, C. H., Kimball, R., Verplank, W., and Harslem, E. Deisgning the Star User Interface. *Byte* 7(4), April 1982, pp. 242-282.
14. Taylor, A. R., Nonverbal Communications Systems in Native North America. *Semiotica*, *13*(4), 1975, pp. 329-374.
15. Want, R., Schilit, B. N., Adams, N. I., Gold, R., Petersen, K., Goldberg, D., Ellis, J. R., and Weiser, M. An Overview of the ParcTab Ubiquitous Computing Experiment. *IEEE Personal Communications*, December 1995, pp. 28-43.
16. Weiser, M. The Computer for the 21st Century. *Scientific America*, *265*(3), 1991, pp. 94-104.
17. Wellner, P. Mackay, W., and Gold, R. Computer Augmented Environments: Back to the Real World. *CACM*, *36*(7), July 1993.

Petitioner Samsung Ex-1038, 0008