HabilisX: Document Exploration Through a Tool-Based Touch Screen Environment

Prairie Rose Goodwin

Abstract

This project explores a new touch screen interface for interacting with data sets through a digital tool metaphor. Research in psychology has shown that when we hold a tool, such as a hammer, our minds frame potential solutions around the affordances of the tool. Additionally, metaphors have proven to be a powerful way to positively affect user experience, in part because the user has expectations about how different objects behave. Our system attempts to recreate the experience of using physical tools. Touch screens provide the ideal interaction style for this metaphor because they allow the user to be more physically engaged than a traditional point-and-click interaction. With the support of the James B. Hunt Library at North Carolina State University, we created a work environment with life-size tools using a 40-inch PixelSense 2.0 touch screen table. Our interface consists of two and a half dimensional representations of paperclips, pushpins, rulers, magnifying glasses, "magic lenses", and sticky notes that can be used to manipulate database entries that appear as index cards scattered randomly on the table. All objects on the screen have physical properties like center of mass, momentum, and friction to mimic the behavior of real objects on a table. The tools are designed to let the user search and organize the data efficiently by taking advantage of spatial cognition and information visualizations. Moreover, the user can leverage basic filtering queries by attaching "filter tiles" to the tools that dictate the entries with which the tools will interact. Our goal is to show that our system is user-friendly and allows users to categorize data faster and more accurately than a comparable tabular interface.

1 keywords

Information Visualization, Spatial Cognition, Spatial Memory, Document Management, Database Interface

2 Introduction

3 Related Work

3.1 Tool Metaphor

GUI Metaphors allow users to represent problems in a more easily solvable format. As cognitive tools defined by Hutchins [9] and Norman [12], they have shown that they can be extremely powerful aids. Since the introduction of the desktop metaphor by Xerox in the '70's it has become ubiquitous for personal computers. Command line usability is notoriously difficult for beginners and non-technical users, whereas being able to interact with visual icons gives users an underlying model of what their actions are doing. This allowed a whole new demographic to use computers, and subsequently the adoption of home computers skyrocketed.

This project explores a novel user interface with a tool metaphor for database searches. Tool metaphors have been introduced at a surface level in some conventional software environments. Most notably visual programs like photoshop or paint have a tools that have analogs to real world objects. However, the metaphor is usually quite shallow, and beyond icon and task, bear little resemblance to the experience of using that tool. This is important because there is a cognitive shift that occurs when using a physical tool in which the user starts to think of the tool as an extension of his or her body. [11] When we factor in the affordance of a tool, we realized that tool metaphors have the ability to remold the experience into something familiar.

Direct manipulation interfaces provide a good environment to explore affordances in software [6], [12]. Users interact directly with objects on the screen instead of with menus or commands [8] [17], which are continuously represented with animation instead of appearing and disappearing as needed.[17] Getner argued that deeper metaphors that are more consistent with the physical world can improve learnability and usability of the system [7] and can be a good opportunity for the user to become familiar with the system [2].

3.2 Spatial Cognition

There is a large body of work that has shown the cognitive benefits of being able to organize items spatially. [1] [15] Robertson et al. made a 2-D interface for bookmarking website that showed dramatic improvement in efficiency and error over the built in text-based system.

Most work has been done between a graphical environment versus a textual environment, but Sebrechts et al took this concept one step further and did a formal evaluation of a text-based, 2-D, and 3-D interfaces.[16] They found that the 2-D interface had the best performance in both user satisfaction and performance. Additionally, they tried multiple input devices, which made a significant impact on the usability of the interface.

3.3 Touch

Touch screens have become popular in recent years, but many feel that they are a step backwards in usability [13]. Gestures have many of the drawbacks of a command line interface: commands can be hard to remember, impossible to explore, and hard to debug. There are no visual cues for what gestures the system will recognize, and there is not a standard set of gestures for all functionality. Some work has been done to standardize gestures to make them more intuitive [14], but adoption of standards continues to be a problem with a few exceptions like pinch to zoom.

New work suggests that multi-touch interfaces may also have a usability gain for expert users when used appropriately. [5] [18][14]. It has less chance of causing carpal tunnel, and is more efficient in the amount of time to select targets than the mouse, which is the majority of screen interaction [5]. Tabletop touch screen interactions showed that users benefited from direct touch input over mouse. However, for traditional computers, some tasks were more efficiently done with a mouse. [10]. Moreover, using a table is a more active form or interaction because of the larger amount of movement required to reach the entire screen. This full embodied interaction has shown even more cognitive benefits. Kinesthetic signals (muscle memory) has been shown to improve spatial recall by 19% in an experiment document retrieval setting.[18]

3.4 Document Management

So much of what we do with computers is dependent on being able to find pertinent information. Many systems have focused on how to keep personal documents organized in a desktop interface with positive results [1] [4] [3].

4 Habilis

Our system has a well established set of interaction mechanisms, and clear parallels to physical tools in the real world.

We propose a novel way to display and organize information visually through the metaphor of physical tools. Tool-use in the physical world generally includes an object in the operator's hand and broad motions to perform an action. We propose to extend these concepts to digital tools by implementing a program on the Microsoft Pixelsense 2.0 that lets the user select (pick-up) tools on the screen and perform motions that mimic physical tool usage. Program description:

This program would be used to search and organize information stored in a database. After getting an initial body of entries to search, each entry will appear as individual tiles with properties of naive physics that can be moved around the screen by touch.

Figure 1: Screenshot of Habilis after having used all tools.

4.1 Save States

Habilis included two save states. You can switch between two configurations of data for comparative purposes. The tools do not change when a save state is loaded. This is in part to maintain object permanence, but also so that the same queries can be used on either configuration.

4.2 Filter Tiles

The name of each attribute or column header is shown on buttons that generate a label when pressed. These labels specify search criteria and can be attached to the tools to determine the items with which the tool will interact. For example, if the attribute is a string, the user should be able to specify a particular string to search for. If the attribute is an int, the user should be able to use any of the compare functions available such as greater than, less than, etc.

Example: Let's say that your data set is a collection of research papers with the attributes:

(String) Title (String) Author (Date object) DatePublished (List¡String¿) Keywords (int) pages

If you are looking for a paper on computer vision published in the last 5 years that is between 5 and 10 pages, you could create the query labels:

- Pages: > 5
- Pages: < 10
- Date: < 2000

- Keywords: "Computer Vision"

A single tool that has all of these queries would match to a paper that is 5-10 pages long, published before the year 2000, with the author- supplied keyword "Computer Vision".

Once created, you can attach and detach the query labels to tools by intersecting the two objects to quickly interact with your data.

Sticky Note

Sticky Notes attach directly to entries and look similar to a filter with a different background color. Once attached, the note will display the title, but tap it once to switch to the body of the note, and again to see the title again.

4.3 Habilis Tools

Not all tools can accept filters. The focus of this project was searching, so the tools that were useful for that task cab accept filters. However, for other tasks, we found it was more useful to have tools that were consistent for all entries. These tools were used for modifying entries based on their location rather than their content.

Push Pin

The pushpin is used to "save" entries. Once pinned, the entry is immobilized and no other tool can modify that entry. Entries can be thrown under a pin to create a messy pile.

Ruler

The ruler can push entries around the screen and organize them visually by aligning them horizontally or vertically. This state is somewhere in between a tidy pile and a messy pile as it does not afford browsing. However, it unmistakably shows you entries that are categorized together.

Trash Can

The trashcan removes any unwanted entries or tools. Just like in a Desktop GUI, just drag the item to the trash can, and it will disappear from the screen.

4.4 Habilis Filter Tools

The following tools can be modified by attaching filter tiles. Once attached, the tool will only interact with the entries that match the tile's query.

Magic Lens

The magic lens is a window that highlights entries that match your query. Additionally, the lens will pop the results to the front of the interface so that no result is hidden.

Magnifier

The magnifier lets you view any attribute of an entry. Simply drag the filter tile to the magnifier and instead of creating a query, the corresponding data of that attribute will be displayed in a pop-up next to the tool. Drag any number of filter tiles, and the attribute will be displayed below the last one.

Paper Clip

Paper clips are used to create organized piles, or "tidy piles". If no filters are attached, the paper clip will pick up everything. If a filter tile is added, it will drop any entries that do not mach the query.

4.5 Habilis Interaction Design

The buttons that were initially placed on the screen to populate the tools are not movable, scalable, or rotatable. They can be activated by a quick or long touch and have no other interactions.

All other UI Elements were sub-classed from a ScatterViewItem and have the same interaction properties. All elements have have physical properties like center of mass, momentum, and friction to mimic the behavior of real objects on a table. You can drag them around using one finger, but having two fingers on the tool disabled its use so that you could drag it around the screen without interactions with other UI elements.

Once a filter tile or note has been attached to its target, it can be removed with a long touch (1.5 seconds).

4.6 Hardware and Developer Tools

The Pixelsense 2.0 developer packages for the SUR40 had a lot of positives that made it attractive for this project. The developer tools had a parent classes (ScatterView) and (ScatterViewItem) that gave objects physical properties. When touched, a shadow was activated to give the illusion of the item being picked up. ScatterViewItems have built in multi-touch gestures, such as pinch to scale, dragging, and rotations. Additionally, they had inertia, momentum, and a center of mass that allows users to do a one finger rotation as well. The ScatterView is optimized for ScatterViewItems creating an environment in which they can exist. ScatterViews have well defined edges that ScatterViewItems cannot cross, instead bouncing off the edge when thrown. When a ScatterViewItem is places on it, it's rotation and placement is random as to encourage users to stand on any side of the table. These objects, combined

Figure 2: SUR40 Touchscreen Table

with the C# wpf framework, minimized the amount of physics that we had to program by hand.

As a stand alone touch screen table, it was designed in such a way as to allow a single user could reach any part of the screen without moving, while still allowing for multiple users simultaneously. The touchscreen is not capacitive. Each pixel has an IR Sensor, an RGB Sensor, as well as the pixel. The table relies on the distance of your hand from the screen as read by the IR sensor to determine whether or not you are touching it. Additionally, the RGB sensor allows the table can recognize "Microsoft Tags" which are simpler QR codes that can be used for object recognition.

The fact that the table uses IR sensors expands the realm of interactions. For example, digital art could be painted with an actual paint brush. For traditional computing, it gives the user the "hover" state, where the user targets a UI element without activating it, for a 3rd interaction state.

5 User Study

5.1 Methods

Subjects

Eight Computer Science graduate students between the ages of 22 to 26 from various research specialties participated in this study. No one had previous experience using Habilis or the SUR40, while six out of eight had no previous experience with Mendeley. All subjects were male with normal or corrected-to-normal vision with no other known disabilities.

Equipment

Habilis was run on the Samsung SUR40 40-inch touch screen table with no external hardware. Users were forced to use only touch interactions with a virtual onscreen keyboard. Mendeley was run on a Sony Flip-PC touch screen convertible laptop. However, no user chose to utilize the touch interactions of the laptop, and instead used a wireless mouse and build in keyboard.

Procedure

This experiment was a comparative study between Habilis and Mendeley. To test which system was better for organizing documents into categories, users were asked to identify citations of a source paper. We chose to use Bumptop [1] as our source paper because it has many sources that neatly divide into several subjects. The authors also released a seven minute video that gives a detailed overview of all of the features and novel designs without mentioning their sources.

Bumptop had forty-one sources that encompassed six subjects: advantages of a pile metaphor, stylus interactions, spacial organization, realism, browsing techniques, naive physics, and animation. Dataset 1 encompassed twenty-one papers covering pile metaphors, stylus interactions, and spacial organization. The remaining twenty papers went into dataset 2. Distractor papers were added from mobile and touchscreen research until each dataset had a total of thirty-seven papers. The subjects were then asked to identify which papers were related to bumptop. The datasets were alternated between Habilis and Mendeley to overcome any performance bias of the data. The trials were timed, and at the end they were asked 3 questions:

1. Which system did you prefer? why?
2. Did you see any topic clusters? If so, what topics?
3. What features were you looking for but didn't see in Habilis?

6 Results

Hardware Problems

Unfortunately, the novel IR technology that this table did not work as well as we had hoped. In fact, this model was discontinued within a year and a half of it being released because of significant hardware problems. When sitting in front of the table, it was as likely to read the users wrist as a touch as his or her finger. When it recognized a "blob" where your palm could be seen, it would register circa 50 events in and around that area. If there were significant smudges on the table, that could throw off what was happening where your had was. The most significant problem however, was that the calibration was so sensitive that if the lighting changed at all from when you calibrated it, the table would not function correctly. This meant that it could not be placed in any room with natural light. It took us several months to figure out this problem, not realizing that the time of day or the weather could have such a significant impact on the table's usability.

Once we got the table into a windowless room, we were able to improve many of the problems programmatically within Habilis, but the on screen keyboard continued to be a major problem. 5 of the 8 participants mentioned that they had shied away from using the tools because the keyboard was so hard to use. Occasionally, a ghost event would make something fly across the screen which would break the user's concentration. There is no question that the hardware failures contributed to the results of the user study.

Performance Time

Despite the advantages of the GUI system, the performance time was slower than Mendeley. This can be explained by a few things. First, 2 users did not utilize the tools at all and instead just read the attributes off of the card. Second, the hardware event misfires as described above had a tendency to break the user's train of thought, and they had to compensate by taking time to get back to where they were before the event fire. Third, none of the users had any familiarity with Habilis before the trials. Since they were learning the system while using it, there is the possibility that performance time would improve over time.

An single factor analysis of variance (ANOVA) was performed on the performance time of the data grouped within subject. Mendeley takes less time to complete the task than Habilis with $F = 3.25$ and $p = 0.06$. Doing a t-test on the same data found $t = -3.18$ and $p < 0.01$.

Number of Incorrect Categorizations

Users did not more accurately categorize the data with Habilis than Mendeley as we had hoped. We think that this was a combination of an unfamiliar system combined with a lack of adoption of the tools available. No one used the sticky

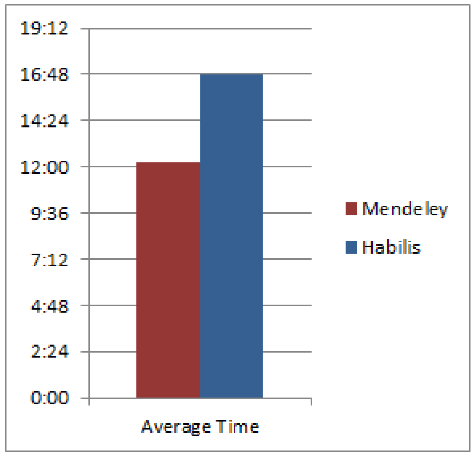

Figure 3: Average Time for completion

note, 6 did not use the ruler, and some tools were used once and then never again. Most interactions were with the paperclip, push pin, and magic lens.

Mendeley and Habilis performed equally twice, but otherwise Mendeley provided better results. A Single Factor ANOVA was performed on the number of correctly categorized papers with $F = 6.4$ and $p < 0.03$

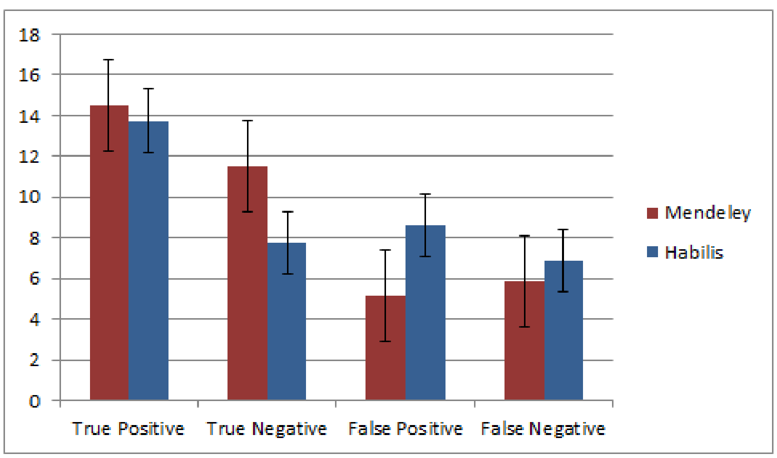

Figure 4: SUR40 Touchscreen Table

Questionnaire

7 Discussion & Future Work

References

[1] Anand Agarawala and Ravin Balakrishnan. Keepin' it real: Pushing the desktop metaphor with physics, piles and the pen. In Rebecca E. Grinter, Tom Rodden, Paul M. Aoki, Edward Cutrell, Robin Jeffries, and Gary M. Olson, editors, *Proceedings of the 2006 Conference on Human Factors in Computing Systems (CHI 2006)*, pages 1283–1292, Montreal, Canada, April 2006. ACM Press.

[2] Gerhard Fischer. Turning breakdowns into opportunities for creativity. *Knowl.-Based Syst.*, 7(4):221–232, 1994.

[3] Schubert Foo and Douglas Hendry. Desktop search engine visualisation and evaluation. In *Proceedings of the 10th International Conference on Asian Digital Libraries: Looking Back 10 Years and Forging New Frontiers*, ICADL'07, pages 372–382, Berlin, Heidelberg, 2007. Springer-Verlag.

[4] Schubert Foo and Douglas Hendry. Evaluation of visual aid suite for desktop searching. In *Proceedings of the 11th European Conference on Research and Advanced Technology for Digital Libraries*, ECDL'07, pages 333–344, Berlin, Heidelberg, 2007. Springer-Verlag.

[5] Clifton Forlines, Daniel Wigdor, Chia Shen, and Ravin Balakrishnan. Direct-touch vs. mouse input for tabletop displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 647–656, New York, NY, USA, 2007. ACM.

[6] William W. Gaver. Technology affordances. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '91, pages 79–84, New York, NY, USA, 1991. ACM.

[7] Don Gentner and Jakob Nielsen. The anti-mac interface. *Commun. ACM*, 39(8):70–82, August 1996.

[8] E. Hutchins. Metaphors for interface design. In M. M. Taylor, F. Neel, and D. G. Bouwhuis, editors, *The Structure of Multimodal Dialogue*, pages 11–28. North-Holland, Amsterdam, 1989.

[9] E. Hutchins. *Cognition in the Wild*. A Bradford book. MIT Press, 1995.

[10] Per Ola Kristensson, Olof Arnell, Annelie Bjork, Nils Dahlback, Joackim Pennerup, Erik Prytz, Johan Wikman, and Niclas strom. Infotouch: An explorative multi-touch visualization interface for tagged photo collections. In *Proceedings of the 5th Nordic Conference on Human-computer Interaction: Building Bridges*, NordiCHI '08, pages 491–494, New York, NY, USA, 2008. ACM.

[11] Angelo Maravita and Atsushi Iriki. Tools for the body (schema). *Trends in Cognitive Sciences*, 8(2):79 – 86, 2004.

[12] Donald A. Norman. Designing interaction. chapter Cognitive Artifacts, pages 17–38. Cambridge University Press, New York, NY, USA, 1991.

[13] Donald A. Norman and Jakob Nielsen. Gestural interfaces: A step backward in usability. *interactions*, 17(5):46–49, September 2010.

[14] Chris North, Tim Dwyer, Bongshin Lee, Danyel Fisher, Petra Isenberg, George Robertson, and Kori Inkpen. Understanding multi-touch manipulation for surface computing. In Tom Gross, Jan Gulliksen, Paula Kotz, Lars Oestreicher, Philippe Palanque, RaquelOliveira Prates, and Marco Winckler, editors, *Human-Computer Interaction INTERACT 2009*, volume 5727 of *Lecture Notes in Computer Science*, pages 236–249. Springer Berlin Heidelberg, 2009.

[15] George Robertson, Mary Czerwinski, Kevin Larson, Daniel C. Robbins, David Thiel, and Maarten van Dantzich. Data mountain: Using spatial memory for document management. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology*, UIST '98, pages 153–162, New York, NY, USA, 1998. ACM.

[16] Marc M. Sebrechts, John V. Cugini, Sharon J. Laskowski, Joanna Vasilakis, and Michael S. Miller. Visualization of search results: A comparative evaluation of text, 2d, and 3d interfaces. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 3–10, New York, NY, USA, 1999. ACM.

[17] Ben Shneiderman and Catherine Plaisant. *Designing the User Interface: Strategies for Effective Human-Computer Interaction (4th Edition)*. Pearson Addison Wesley, 1992.

[18] Desney S. Tan, Randy Pausch, Jeanine K. Stefanucci, and Dennis R. Proffitt. Kinesthetic cues aid spatial memory. In *CHI '02 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '02, pages 806–807, New York, NY, USA, 2002. ACM.