# Gismo: Graphical Interface for Searching with Multi-touch Objects

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, giving the user some 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, we have 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 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

Databases store vast amounts of data that people must interact with every day. However, the interfaces that control databases have not changed significantly in the last thirty years. Most are still command line-like interfaces with a list of results that match the user's query. Good interfaces will have a bar on the side to let you add additional filters to your search results. There is no way to easily compare queries side by side, and it can be impossible to see trends within the list of results.

Desktop document management on the other hand has evolved considerably in the same amount of time. Instead of working with a command line as seen in computers from the 70's and 80's, users now have files, folders, desktops, icons, thumbnails, shortcuts, and a myriad of other metaphors that improve usability and performance. Gismo is a comprehensive graphical user interface for databases that pulls from many of the strengths of a desktop user interface. Instead of a desktop metaphor, Gismo uses a digital tool metaphor.

Tools are an integral piece of how humans interact with their physical environment. When the analogy has been translated into a metaphor for software interfaces, it has been shown to have good usability, and users have adopted the metaphor with little difficulty. However, these metaphors are only shallow representations of the full experience of using a physical tool. Our program tries to incorporate as many aspects of tool use as possible. We have chosen a technology that allows full movement of arms with a touch screen interface, and our digital representations of objects have naive physics that more closely mimic real world interactions.

In addition to a tool metaphor, Gismo draws on many strengths of the desktop user interface, such as using spatial memory to improve recall and organizational practices. Databases are read dynamically into Gismo where the entries are translated into life size index cards that can then be moved around the screen. The two and a half dimensional interface allows things to be focused, piled, pinned, and grouped. Multi-touch interactions allow users to zoom, drag, and rotate as appropriate for the objects in the interface.

The goal of this project was to create a graphical user interface that would transform database searches into a visual experience that leverages many of the cognitive benefits of tool use.

Our experiment was designed to test if a user could gain some understanding of the content of their data set without ever having seen it before. We hypothesized that our system would allow users to more accurately determine the relatedness of an entry to a given subject, and that the user could do it faster than a traditional list-based interfaces that are common today.

To do this we created a program that leveraged several different interactions that have shown to be beneficial to user performance:

- Tools: Affordances of tools shape the users approach to problems put before them

- Metaphors: simplify problems by framing them in a familiar context

- Data visualizations: Provide context and can show patterns within the data set as a whole

- Visual memory: improves recall of items

In this paper, we describe the technology that we chose to use for this project (SUR40), the tool metaphor and how a user would utilize the tools, and the user study that we did to compare it to Mendeley. We conclude with some discussion of why we felt our user trials did not yield positive results, and future work that we plan on doing for the next iteration of Gismo.

# 3 Related Work

## 3.1 Tool Metaphor

GUI metaphors became popular because of their ability to reduce complicated tasks to a simpler structure. The desktop metaphor, originally introduced by Xerox in the '70's, has become ubiquitous for personal computers. Command line usability is notoriously difficult for beginners and non-technical users. On the other hand, being able to interact with visual icons gives users an underlying model of the effect of their actions. Giving users this mental mapping opened up computing to an entirely new demographic of people, and subsequently, the adoption of home computers skyrocketed.

We explore a novel user interface with a tool metaphor for database searches. Tool metaphors already exist in some conventional software environments. Most notably, visual editing programs like Photoshop and Paint use analogs to real world objects like paint brushes and erasers to communicate functionality. 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 [13]. When we factor in the affordance of a tool, we realize that tool metaphors may have the ability to positively impact usability in ways that other metaphors may not.

Direct manipulation interfaces provide a good environment to explore affordances in software [7] [16]. Users interact directly with objects on the screen instead of with menus or commands [9] [23], which are continuously represented with animation instead of appearing and disappearing as needed [23]. Gentner argues that deeper metaphors that are more consistent with the physical world can improve learnability and usability of the system [8] and can be a good opportunity for the user to become familiar with the system [3].

## 3.2 Spatial Cognition

Previous research has shown the cognitive benefits of being able to organize items spatially [1] [20]. Robertson et. al made a two and a half dimensional interface called DataMountain for bookmarking websites that showed dramatic improvement in efficiency and error over the built in text-based system. DataMountain's design is very similar to ours so we will discuss their work in depth throughout the paper.

Numerous studies have compared graphical environment and textual environments. Sebrechts et. al most notably did a formal evaluation of text-based, 2-D, and 3-D interfaces [21]. They found that the 2-D interface had the best performance in both user satisfaction and performance with modern input devices. They varied the input devices, and found that they made a significant impact on the usability of the interface. The more closely the input method matched the visualization on the screen, the more usable the system. This result lends support that touch may be the best interaction technique for direct manipulation interfaces such as ours.

## 3.3 Touch

Touch screens have become popular in recent years, but they are often a step backwards in usability [17]. Gesture-based interfaces 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 [18], but adoption of standards continues to be a problem with few exceptions like pinch to zoom.

Other work suggests that multi-touch interfaces may have usability gains similar to command-line interfaces for expert users [6] [24] [18]. The biggest efficiency gain comes in target selection, which is the majority of screen interaction [6]. The nearly unlimited number of available gestures allows a lot more to be done in a smaller amount of space and time.

Tabletop touch screens have a significantly different interaction experience than smaller screens. Consequently, some of the conventions used on other devices have not translated well. On larger touch screens, users benefited from direct touch input over mouse which is a more active experience [11]. This may seem counter intuitive because of the larger amount of movement required to reach the entire screen. However, the kinesthetic signals (muscle memory) that are associated with embodied movements have been shown to improve spatial recall by 19% in an experiment in document retrieval [24]. This may also have impacts on overall the overall usability of large screens.

## 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 for quick retrieval with positive results [1] [5] [4]. However, scaling the interfaces to be able to handle large amounts of data continues to be an unsolved problem [25].

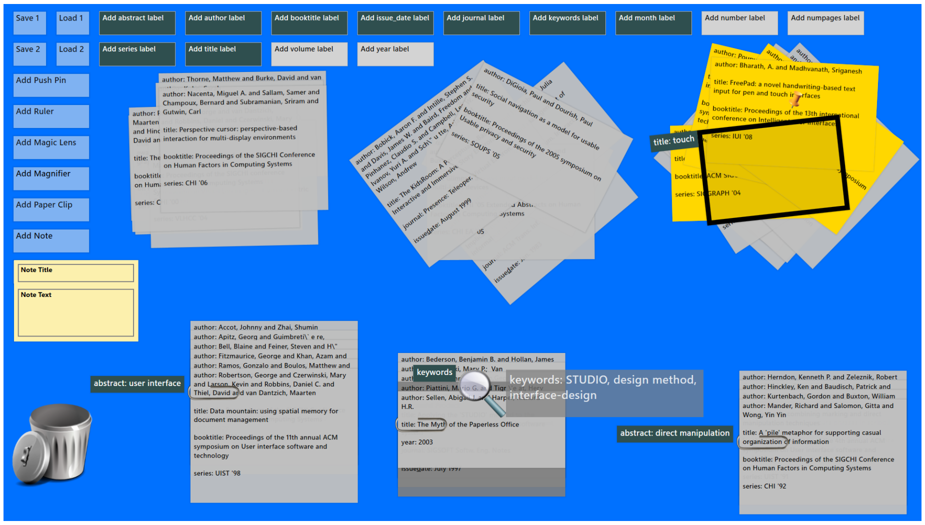
Some interfaces have tried different visualizations to put as much information on the screen as possible [15] [19] [22] [26]. These types of interfaces try to find some way to group things together into topics [15], but a lot of information is lost when this is done [19]. Because our understanding of natural language processing is imperfect, the algorithms doing the modeling can make imperfect decisions. This leads to documents being miscategorized (and therefore overlooked), or making decisions that are not obvious to the user. GUI's benefits are significantly reduced when the user's mental model breaks down, which causes confusion.

Trying to use spatial relationships to display different kinds of information seems to be a better solution [2] [5] [4] [14]. Bier et. al used the x,y position of a document to communicate the value different attributes, allowing multiple metadata to be easily seen at the same time [2]. Similarly, INQUERY was a system that grouped results into row and column clusters based on related data [14]. Ultimately, all of these still only provide surface-level information. This can make it more difficult to look at different aspects of the data, since these interfaces cannot be modified to organize information based on search criteria.

## 4 Gismo

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 have extended 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. Users interact with objects on the screen by direct manipulation with an established set of interaction mechanism.

This program would be used to search and organize information stored in



author: Thorne, Matthew and Burke, David and van

author: Nacenta, Miguel A. and Sallam, Samer and Champoux, Bernard and Subramanian, Sriram and Gutwin, Carl

title: Perspective cursor: perspective-based interaction for multi-display environments

booktitle: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems

series: CHI '06

author: Bobick, Aaron F. and Intille, Stephen S. and Davis, James W. and Baird, Freedom and Pinhanez, Claudio S. and Campbell, L... Ivanov, Yuri A. and Sch\" u tte, A... Wilson, Andrew

title: The KidsRoom: A P... Interactive and Immersive...

journal: Presence: Teleoper.

issuedate: August 1999

author: DiGioia, Paul and Dourish, Paul

title: Social navigation as a model for usable security

booktitle: Proceedings of the 2005 symposium on Usable privacy and security

series: SOUPS '05

author: Bharath, A. and Madhvanath, Sriganesh

title: FreePad: a novel handwriting-based text input for pen and touch interfaces

booktitle: Proceedings of the 13th international conference on Intelligent user interfaces

series: IUI '08

booktitle: ACM SIG...

series: SIGGRAPH '04

author: Accot, Johnny and Zhai, Shumin
author: Apitz, Georg and Guimbreti\` e re,
author: Bell, Blaine and Feiner, Steven and H\"
author: Fitzmaurice, George and Khan, Azam and
author: Ramos, Gonzalo and Boulos, Matthew and
author: Robertson, George and Czerwinski, Mary and Larson, Kevin and Robbins, Daniel C. and Thiel, David and van Dantzich, Maarten

title: Data mountain: using spatial memory for document management

booktitle: Proceedings of the 11th annual ACM symposium on User interface software and technology

series: UIST '98

author: Bederson, Benjamin B. and Hollan, James
author: Piattini, Mario G. and Tigr...
author: Sellen, Abigail J. and Harper, Richard H.R.

title: The Myth of the Paperless Office

year: 2003

issuedate: July 1997

keywords: STUDIO, design method, interface-design

author: Herndon, Kenneth P. and Zeleznik, Robert
author: Hinckley, Ken and Baudisch, Patrick and
author: Kurtenbach, Gordon and Buxton, William
author: Mander, Richard and Salomon, Gitta and Wong, Yin Yin

title: A 'pile' metaphor for supporting casual organization of information

booktitle: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems

series: CHI '92

Figure 1: Screenshot of Gismo using various tools.

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.

## 4.1 Software Design Decisions

We felt that it was important that Gismo's use case was not too narrowly focused. We wanted it to be flexible enough to be able to be data independent and be used by both beginners and experts. It's backend has three layers. The lowest layer is where the data is stored independently in a relational database, bibTex, or other format. The middle layer consists of a parser that converts that data from it's original form into the internal representation of Entries and Filters. The highest layer is the interface where the user interacts with the data.

### Data

Since we were pulling from a pool of graduate students for our user study, we decided to have our data focus on research publications, pulled mostly from the ACM, IEEE, and Springer digital libraries. The easiest way to store our data was in bibTex. The data was copied directly from the website, and then the abstracts were added to the entries manually for our purposes. This was the only time that a human modified the data in any way.

When the user saves his or her session, it is saved into a bibTex file, with the added attributes of x, y, orientation, and z-index. When the session is reloaded, the parser recognizes these attributes and sets the corresponding values in the internal representation rather than reading them into the list of attributes.

### Parser

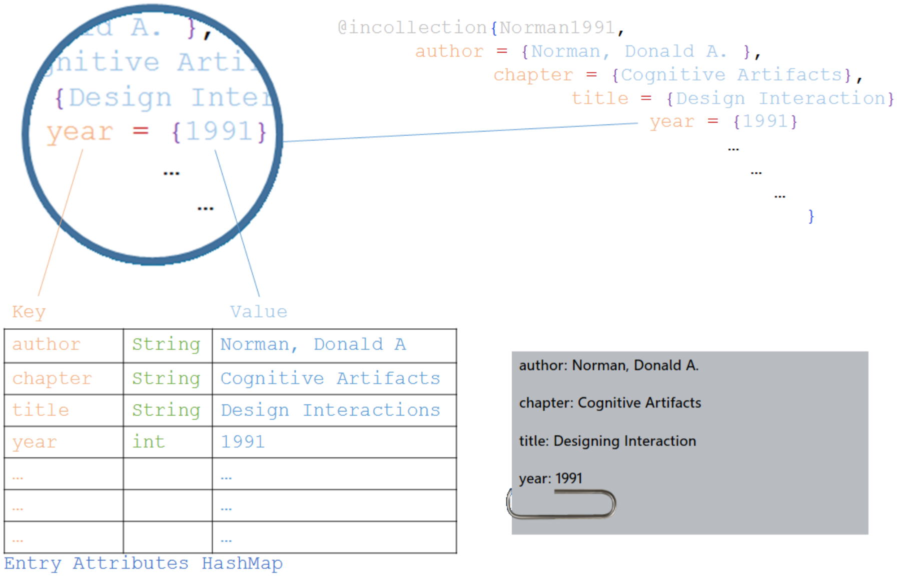
The middle layer needs to be written specifically for whatever kind of database you are using to store your data. We realize that this is a major trade-off to not have the program working out of the box, but abstracting the data out of the interface was the best choice for both performance and flexibility.

The middle layer performs two steps:

1. Create a list of all possible attributes and their data type.
2. Read in the data and create a card for each item in the database.

The current parser is for bibTex and is optimized slightly for our data. Even though, almost every entry had a URL where the full text could be found, we had to have the parser ignore it, because C# was stalling on the slashes. After loading it in once, we decided to have it ignore a couple of other attributes ("location", "articleno") that only appeared in one entry and would therefore not be useful for searching.

Parsing out bibTex is easy for a computer to do.



```
@incollection{Norman1991,
        author = {Norman, Donald A. },
                chapter = {Cognitive Artifacts},
                        title = {Design Interaction}
                                year = {1991}
                                   ...
                                      ...
                                         ...
                                            }
```

| Key | | Value |
|---|---|---|
| author | String | Norman, Donald A |
| chapter | String | Cognitive Artifacts |
| title | String | Design Interactions |
| year | int | 1991 |
| ... | | ... |
| ... | | ... |
| ... | | ... |

author: Norman, Donald A.

chapter: Cognitive Artifacts

title: Designing Interaction

year: 1991

Figure 2: This figure goes through how the data looks in each later. The data layer consists of plain text in bibTex form. The middle layer creates an attribute HashMap that is used as the backend for the Entry that appears on the screen.

The starting screen of the interface has buttons at the top of the screen for each unique attribute that appears in the database. The background of the button is determined by the attributes data type, so the parser has to determine these things before loading. The database is read into the program dynamically at each start-up and

## 4.2 Save States

Gismo includes 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 done to maintain object permanence and to facilitate using the same queries on either configuration.

## 4.3 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 can specify a query string that will be used in a “contains” search within that attribute. If the attribute is a number, the user can use any of the comparison functions available such as greater than, less than, etc.

Example: Assume that your dataset is a collection of research papers with the attributes:

- (String) Title
- (String) Author
- (Date) DatePublished
- (List<String>) Keywords
- (int) pages

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

- Pages: > 5
- Pages: < 10
- Date: > 2009
- Keywords: "Computer Vision"

Once created, the filter tile can be attached to tools by intersecting the two objects. To detach, a long touch will remove the filter tile from the screen.

## 4.4 Gismo Tools

The focus of this project was searching, so the tools that were useful for that task can 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. Therefore, filters were disabled for the following tools.

### 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 with casual organization.

### 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 [1] as it does not afford browsing. However, it unmistakably shows you entries that are categorized together.

### 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 the user can tap it once to switch to the body and title of the note.

### Trash Can

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

## 4.5 Gismo 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 results are hidden. The main functionality lets you see the results of queries without modifying the entries. As a result, you can explore several queries side by side or consecutively before making decisions about your next action.

### Magnifier

The magnifier lets the user view any attribute of an entry. To activate the magnifier, you must drag a filter tile to the magnifier. Instead of creating a query, the corresponding data of that attribute will be displayed in a pop-up next to the magnifier. The magnifier will accept any number of filter tiles, and the attribute will be displayed below the last one.

### Paper Clip

Paper clips 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 match the query. With this tool, the user can easily separate the entries in which you are interested from those that you are not.

## 4.6 Gismo Interaction Design

The buttons that are 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.

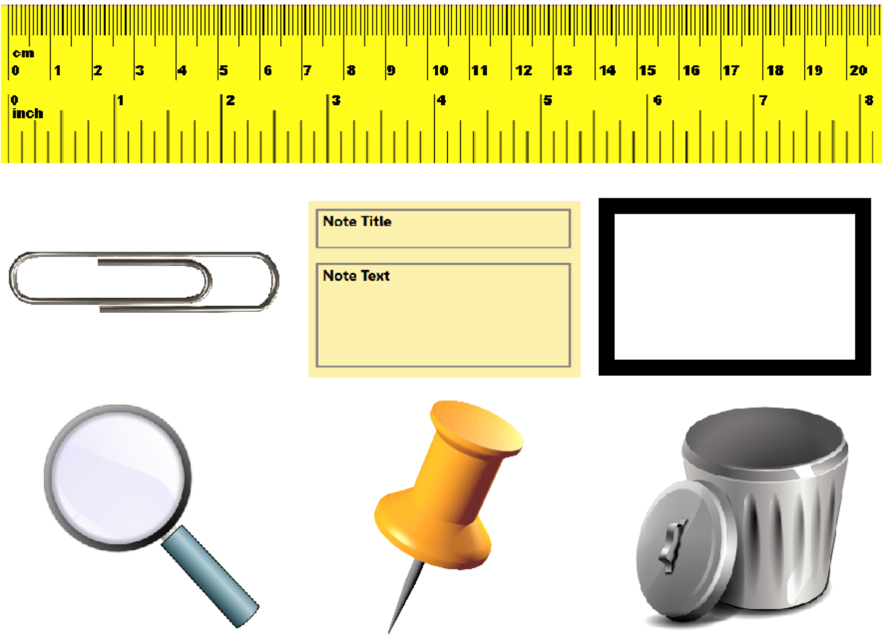


Figure 3: Images used for digital tools. From Top to bottom, and left to right: Ruler, paperclip, sticky note, magic lens, magnifier, push pin, and trash can.

The other UI Elements are sub-classed from a ScatterViewItem, a PixelSense 2.0 specific object class, and all have the same interaction properties. All elements 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 dragging with two fingers disables the tool's interactions so that you could drag it to where you would like to use it. Additionally, you can do a two finger rotation as you would expect, and a one finger rotation around the center of mass. Objects can be flicked across the screen, bouncing when it reaches the edge of the screen (a fullscreen ScatterView object). All objects, including filter tiles and notes, are activated by intersecting the borders of the items. Once a filter tile or note has been attached to its target, it can be removed with a long touch (1.5 seconds).

## 4.7 Hardware and Developer Tools

The PixelSense 2.0 developer packages for the SUR40 have a lot of positives that made it attractive for this project. The developer tools have 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,

Figure 4: SUR40 Touchscreen Table

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, its rotation and placement is random so to encourage users to stand on any side of the table. These objects, combined with the C# wpf framework, minimized the amount of physics that we had to program by hand.

The touch screen table was designed in a way that allows a single user could reach any part of the screen without barring access to multiple users simultaneously. The touchscreen is not capacitive. Each pixel also contains an IR Sensor and an RGB Sensor. 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 third interaction state. However, it also reduces its usability since the IR sensor has a lot more noise than a capacitive touch screen.

# 5 User Study

Our work with Gismo pulls heavily from the concepts established in Robertson et. al [20]. Their system, Data Mountain, focuses on organizing bookmarks from the world wide web instead of database entries, but the tasks are similar enough cognitively that we decided to structure our user study similarly to theirs.

Robertson used a between users study where each participant was placed in one of three groups. Everyone in each group used either Internet Explorer 4 (IE4), Data Mountain 1, or Data Mountain 2 (a revised version of Data Mountain 1 based on user feedback). Participants were given 100 web pages to store in whatever organizational structure that they wanted. In the second half of the experiment they had to retrieve the web pages within a given amount of time. They measured four variables: retrieval time, incorrect retrievals before finding the correct page, failed trials, and a user evaluation.

We decided that we could do a similar study to evaluate Gismo. Both programs use spatial cognition to organize a data set and both programs are designed to give users an understanding of content based on attributes. Instead of simply retrieving a given entry, we asked users to make objective categorizations to test if they actually understood what the data set contained. Although there are many similarities between Data Mountain and Gismo, there are some significant differences between our user studies that are more suited to our domain tasks.

The goal of our experiment was to see if a user could gain some understanding of the content of their dataset without ever having seen it before. We designed our experiment to test two hypotheses:

1. Users will be able to more accurately determine relatedness of an entry to a given subject using Gismo than with Mendeley.

2. Performance time using Gismo will be lower than the performance time of using Mendeley.

We were originally planning for fifteen to twenty participants, but our preliminary feedback showed definitively negative results. As a result, we decided to cut the user trials short after eight participants. If we had gone forward, we would have completed a full power analysis to ensure that we had enough participants for an effective sample size [12].

## 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 Gismo 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

Gismo 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 Gismo 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 has forty-one sources that encompass six subjects: advantages of a pile metaphor, stylus interactions, spatial organization, realism, browsing techniques, naive physics, and animation. Dataset one encompassed twenty-one papers covering pile metaphors, stylus interactions, and spatial organization. The remaining twenty papers went into dataset two. 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 Gismo and Mendeley to overcome any performance bias of the data. The trials were timed, and at the end they were asked to answer a short survey.

## 6 Results

Unfortunately, the IR technology that this table used to determine touch did not work as well as we had hoped. In fact, this model was discontinued within a year and a half of its release because of significant hardware performance problems. When sitting in front of the table, it is 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 hand was because it saw a multi-touch gesture. 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 Gismo, but the on screen keyboard continued to be a major source of frustration for the users. Five of the eight participants mentioned that they had shied away from using the tools because the keyboard registered so many incorrect events, making it impossible to type. Beyond the keyboard, sometimes an event triggered by a wrist would make something fly across the screen which would break the user's concentration. The hardware faults caused the usability of Gismo to suffer greatly, but the magnitude of the problem was not apparent until user trials had begun.

## Performance Time

The performance time of Gismo ended up being slower than Mendeley. This can be explained by a few things. First, two 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 had fired. Third, none of the users had any familiarity with Gismo before the trials. Since they were learning the system while using it, there is the possibility that performance time would improve over time.

Robertson et. al [20] also looked at this metric. The completion times are used directly in their analysis of variance, without adjustments for non-normality. Therefore, we also assume a normal distribution for performance time.

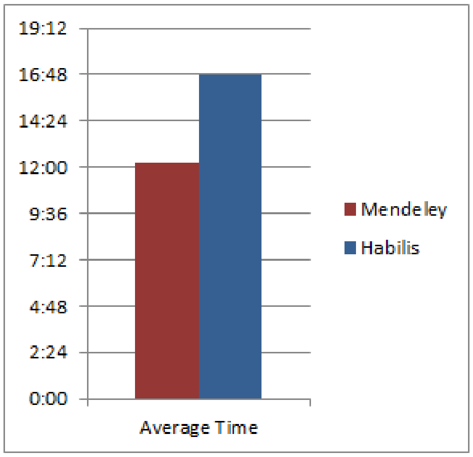


Figure 5: Average Time for completion

A 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 Gismo 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

We had hypothesized that users could more accurately categorize the data with Gismo than with Mendeley. However, our trials showed that the opposite was true. We think that this was a combination of an unfamiliar system combined with an under utilization of the tools available. No one used the sticky note, six 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 Gismo performed equally twice, but otherwise Mendeley provided better results. A Single Factor ANOVA was performed on the number of correctly categorized papers with a statistically reliable main effect where $F = 6.4$ and $p < 0.03$

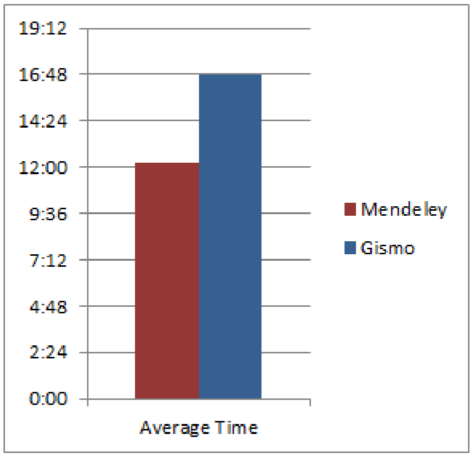


Figure 6: SUR40 Touchscreen Table

## Questionnaire

Once finished, the users answered three questions about their experience.

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 Gismo?

The first question asks about user preference, the second asks about cognitive benefits, and the third is about usability and feature visibility.

Three users said that they preferred Gismo to Mendeley, and a fourth said that he would if the ghost events of the table could be fixed. With nearly half enjoying the experience despite the system bugs, there is potential to raise this number if all of the hardware problems can be fixed.

Many of the design principles that went into this project were to help visualize overall trends in the data. All of the users identified a mobile theme in dataset two, while two users identified touch based themes while on the table for dataset one. These are the distractor themes. Six of the eight users identified a physics theme, but the rest of the subjects that were recorded were not listed by more than one user.

The last question yielded the most interesting results. By asking this question, we were able to determine which features were not immediately visible to the user. For example, three users said that they would have liked a way to read the abstract of the papers. The magnifier will bring up the abstract to read for just this purpose, but these individuals never found this functionality, even if they had used the magnifier for other purposes. Another person mentioned that they wished that there was a way to organize the entries into a sorted list. All paperclips automatically alphabetize the pile based on the first attribute, in this case author. The other suggestions were minor: have the papers snap to the middle of the push pin; change the active area of an entry; cut down on the clutter of the buttons at the top. These suggestions imply that the users are already thinking around the affordances of the interface. After only a small amount of time with Gismo, no one asked for anything that would be difficult to integrate into the existing interface.

## 7 Discussion & Future Work

Unfortunately, our user study did not demonstrate that Gismo was an effective alternative to Mendeley. We felt that this result was due in large part to the hardware rather than the design. A more accurate touch screen would decrease user frustration and may encourage users to explore the interface more.

One of the most common feedback that we received was that the "messy" initial state was a little overwhelming. If we could somehow group the papers to begin with, it would give the user some place to start. The next iteration of this program will use intelligent topic modeling (Latent Dirichlet Allocation) to provide the user with piles at start up. There will be a simple interface that will allow the user to decide which attributes they want to use for the topic modeling and to vary the parameters to their needs.

Another point of improvements will be on the scalability of the interface. Like most GUI's there is a tradeoff of visual benefits vs screen space. Although performance doesn't degrade very quickly, after about 200 entries, Gismo becomes unwieldy simply because there is not enough space in which to work with the entries efficiently.

In summary, we have created a novel touch screen database interface for exploring datasets visually through a tool metaphor. Our contribution derives from utilizing technology to create an interface that has benefits over traditional computing. We believe that Gismo has been a successful cognitive tool that reduces large problems into a simpler task, and with some more fine tuning, we are confident that it will show concrete benefits over list based interfaces.

# 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] Eric A. Bier and Adam Perer. Icon abacus: Positional display of document attributes. In *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '05, pages 289–290, New York, NY, USA, 2005. ACM.

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

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

[9] 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.

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

[11] 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.

[12] Russell V. Lenth. Some practical guidelines for effective sample-size determination, 2001.

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

[14] Josiane Mothe and Taoufiq Dkaki. Interactive multidimensional document visualization. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 363–364, New York, NY, USA, 1998. ACM.

[15] David Newman, Timothy Baldwin, Lawrence Cavedon, Eric Huang, Sarvnaz Karimi, David Martinez, Falk Scholer, and Justin Zobel. Invited paper: Visualizing search results and document collections using topic maps. *Web Semant.*, 8(2-3):169–175, July 2010.

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

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

[18] 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.

[19] Lucy Terry Nowell, Robert K. France, Deborah Hix, Lenwood S. Heath, and Edward A. Fox. Visualizing search results: Some alternatives to query-document similarity. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '96, pages 67–75, New York, NY, USA, 1996. ACM.

[20] 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.

[21] 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.

[22] Ben Shneiderman, David Feldman, Anne Rose, and Xavier Ferre Grau. Visualizing digital library search results with categorical and hierarchical axes. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, DL '00, pages 57–66, New York, NY, USA, 2000. ACM.

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

[24] 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.

[25] Steve Whittaker and Julia Hirschberg. The character, value, and management of personal paper archives. *ACM Trans. Comput.-Hum. Interact.*, 8(2):150–170, June 2001.

[26] B. Yost and C. North. The perceptual scalability of visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 12(5):837–844, Sept 2006.