

Enriching the Desktop Metaphor with Physics, Piles and the Pen

by

Anand Agarawala

A thesis submitted in conformity with the requirements
for the degree of Master of Science
Graduate Department of Computer Science
University of Toronto

© Copyright by Anand Agarawala 2006

Abstract

Enriching the Desktop Metaphor with Physics, Piles and the Pen

Anand Agarawala

Master of Science

Graduate Department of Computer Science

University of Toronto

2006

We explore making virtual desktops behave in a more physically realistic manner by adding physics simulation and using piling instead of filing as the fundamental organizational structure. Objects can be casually dragged and tossed around, influenced by physical characteristics such as friction and mass, much like we would manipulate objects in the real world. In addition we explore giving icons the affordances of paper supporting crumpling, creasing and pinning up to the walls for annotation. To manipulate and browse piles we integrate advances in techniques and propose our own novel set. We present a prototype, called BumpTop, that coherently integrates a variety of interaction and visualization techniques optimized for pen input we have developed to support this new style of desktop organization. Encouraging results of a user evaluation are also presented in which participants were able to complete 88% of tasks and find the interface enjoyable and satisfying.

Acknowledgements

During my time at the University of Toronto, I developed in a few ways. I moved away from home, matured and grew up a little, gained some intellectual capacity, gained some pounds and my first grey hairs, gained the affectionate nickname “crispy”, gained several friends through a multitude of good times, infamously embarrassed myself, got to meet some interesting people like Master T and be in the proximity of others (Rick Mercer, Naomi Klein, JRS-One), and had my ego destroyed and rebuilt numerous times. What a journey, what a memorable time. It could not have been possible without the following people who I truly wish to thank.

I would like to acknowledge Ravin Balakrishnan for his support, motivation, ferocious tenacity and research vision, without which this work would not be possible. Also, for encouraging me to eat body parts of shrimp I didn’t think were for eating. The valuable efforts of Karan Singh are duly acknowledged. As well, his ability to host legendary DGP house parties, where phenomenal harmonica players seem to emerge from the woodwork.

To Maegan Harrison, who has been with me from the inception of this project to its close, a beacon of love and unwavering support through the rough terrain whose dedication I will never forget. You pulled me through when I couldn’t go on and eagerly read drafts of my papers including this very document, I love you. To my mother, father and sisters, whose unlimited warmth and love made me who I am today. Mom, thanks for allowing me to fly far from the nest and become a stronger, smarter and better person, enjoying

and exploring the world and myself, and supporting me while I pursued knowledge rigorously and academically. To Pooja, for always being in my corner and making me feel like the most important person in the world, serving as a constant reminder of how to love with your whole heart. Dad, for tirelessly working and supporting me, encouraging me to follow my passions. To Annu, for taking me into her heart and home, helping me mature with the right mixture of motherly caring, freedom, learnin' and tough love! Upma, for being the big sis, looking out for me and reminding me to keep the chin up. Thanks to Tonu for letting me use her picture in my paper, video and thesis. Also, thanks to Saloni for being Saloni.

To all my friends and home-slices around the world. To the NE Crew, my base, who both shaped me and turned me out! For the unforgettable memories rooted in Mikey's basement, that grew far beyond. Mikey, Brendan, Maegan, Laura, Adam, Big up ya 'self. To Tony Gare for being my Super Nintendo buddy, Marigold for broadening my horizons, Chagani for spreading the enlightenment, Anita for letting me whomp her at chess, Gilani for showing me deeper levels of cheap and to the people I'm sure I've forgotten, Thanks.

To Saul Greenberg for enthusiastically introducing me into HCI, taking me under his wing and letting me drop a verse. To Sheelagh Carpendale for mentoring and encouraging me when I thought HCI wasn't for me. Thanks to all the members of the iLab and Graphics Jungle at my alma mater, the University of Calgary, who made it a most memorable place: Captain Kevin, Julia, Peter, RMS, Meru, Pauline and the scots. Also to Gregor McEwan, Michael Boyle and Tony Tang, for being friends and giving me relentless support, reviewing drafts and easing my doubts. Also, thanks to Tang for greasing up my application package and helping me slip into grad school.

Lastly, to all my lab mates at the DGP, you're too awesome. Jack Wang for holding it down and being my desk neighbour, remaining staunchly opposed to bread and having the funnest last name. Azeem and Patricio, my rat pack, for being the dopest, smoothest homeboys. Jacobo for helping me dilute the ratio at Yoga. To Tovi for throwing down

the big and sloppy shotguns, Noah for staying Black, Vogel for aspiring me to dress nicely, Tristan for always rocking, Gonzalo for being cool as in sweet, Anastasia for having such a warm laugh, McGuffin for being Poontang Paul's biggest supporter and being crazy, Géry for the training in French language and taste as well as timeless quotes, Wigdor for his study prowess, Khai for the sage advice and proving there was no Xerox piles paper, Hancock for making it all run smoothly and good times playing Lego Star Wars, Desi's Abhishek and Shahzad to the naan we never quite broke together at Brar Sweets, Wu for being so warm, Fono for turning up the style, Pratscher for being the nicest guy, Sammy for bringing the Sass, Laszlo for his fountain of stimulating ideas, Alex for laying down the language law on these ack's, and he and Eron for indulging in nerd chat with me, Ian with his excellent taste in music, Derek for his resume eye and all the other new kids: Todd, Christian, Simona, Mike. And lastly, the $n-1$ rule and the wonderful staff at Kom Jug Yuen for all the Crispy Beef, BBQ Pork, Salty Spicy Squid and Shrimps, General Tao's Chicken, Green Beans, Singapore Vermicelli, Shanghai Noodles and House Soup.

Previously Copyrighted Material

Much of the material included in this thesis is to be published in the proceedings of the *ACM Conference on Human Factors in Computer Systems*, and has been included herein with their permission and permission of my co-author. The ACM copyright notice applies only to those portions previously published in proceedings of the ACM (see Bibliography, Agarawala & Balakrishnan 2006):

ACM COPYRIGHT NOTICE. Copyright © 2006 by the Association for Computing Machinery, Inc. Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Publications Dept., ACM, Inc., fax +1 (212) 869-0481, or permissions@acm.org.

Table of Contents

1	Introduction.....	10
	1.1.1 Contributions.....	12
	1.1.2 Thesis Overview	13
2	Background.....	15
	2.1 Ethnographic Studies of Office and Desktop Organization.....	16
	2.1.1 How People Organize Their Desks.....	16
	2.1.2 Piling vs. Filing.....	17
	2.1.3 Finding and Reminding.....	18
	2.2 Pile Metaphor.....	19
	2.2.1 Pile Metaphor for Casual Organization	20
	2.2.2 DynaPad.....	22
	2.2.3 Other Uses of the Pile Metaphor.....	23
	2.3 Physically Inspired Interfaces	26
	2.3.1 WebBook and the Web Forager.....	26
	2.3.2 Paper-based Metaphors for Windows.....	28
	2.3.3 Higher Fidelity Physical Simulations	29
	2.3.4 Tossing Interaction.....	30
	2.3.5 Spatial Organization in 3D.....	31
3	Design Goals.....	34
4	Prototype System	44
	4.1 Interaction and Visualization Techniques.....	48
	4.1.1 LassoMenu.....	49
	4.1.2 Object Movement.....	52

4.1.3	Pile Creation.....	53
4.1.4	Pile Manipulation.....	57
4.1.5	Enhancing and Judiciously Using Realism.....	65
4.2	Implementation	68
4.2.1	Performance Concerns	68
5	User Evaluation.....	69
5.1	Findings.....	71
5.1.1	Reflection on Design Goals	71
5.1.2	Interaction Techniques.....	74
6	Conclusions and Future Directions.....	81
6.1	Contributions.....	82
6.2	Future Directions	84
	Bibliography	86
7	Appendix A: Experiment Protocol and Questionnaire	92

List of Tables

Table 1. Subjective questionnaire responses.	71
---	----

List of Figures

Figure 1. Present day virtual and physical workspaces.	11
Figure 2. Examples of typical workspace organization.....	12
Figure 3. Pile Prototype.....	20
Figure 4. Dynapad.	22
Figure 5. Piles for social navigation and security.....	23
Figure 6. Presto fluid workspace with oval piles on the desktop	24
Figure 7. Microsoft Windows Vista stacks	25
Figure 8. WebBook and Web Forager.....	26
Figure 9. Paper-based metaphor supporting rotated and peeled-back windows	28
Figure 10. 3D Paper flyer metaphor simulated with cloth and wind.....	29
Figure 11. Data Mountain prototype	31
Figure 12. Task Gallery prototype.....	32

Figure 13. Spectrum of virtual user interfaces to real desks.	36
Figure 14. Excessive use of metaphor.	39
Figure 15. Shape based gestures in previous interfaces.	42
Figure 16. BumpTop prototype.	45
Figure 17. Earlier Prototype with top-down view.	46
Figure 18. Our techniques apply to arbitrarily sized objects.	47
Figure 19. BumpTop on a commodity TabletPC.	48
Figure 20. LassoMenu. From left to right:	51
Figure 21. Lasso'n'Cross technique for pile creation.....	53
Figure 22. Transitioning from implicit to explicit piles.	57
Figure 23. Pile browsing layouts triggered by widgets.	59
Figure 24. Pressure cursor.	61
Figure 25. Dwell'n'Scrub technique for precise insertion.	63
Figure 26. Drag'n'Cross technique for precise insertion.	63
Figure 27. Real and virtual piles.....	65
Figure 28. Axis alignment enabled to enforce a tidier appearance.	67

1 Introduction

"Computers are useless. They can only give you answers."

-Pablo Picasso

Our vision for the desktop interface is to recapture the lost richness and expressiveness found on people's desks in the real world, as well as awaken our dormant capabilities as physical beings acquired over a lifetime. We note that the desktop has not significantly changed in its appearance or capabilities as a spatial organization tool since its inception in 1981 with the Xerox Star (Smith 1982). Further, despite the metaphor, current virtual desktops bear little resemblance to the look or feel of real world desktops. Chao (2001) observes that "[modern] user interfaces are usually austere, reflecting the limitations of the machines of two decades ago. Our rectilinear desktops feature a machine aesthetic, not a human one." In contrast, workspaces in the physical world typically have piles of documents, binders and other objects arranged in a way that provides considerable subtle information to the owner (*Figure 1 left, Figure 2*). For example, items are often casually placed but their spatial position and orientation are usually meaningful. Closer items can indicate urgency, and piles of items are "automatically" ordered chronologically because new items are typically placed on top. This kind of casual organization prevalent in the real world differs greatly from the GUI desktop which forces users to immediately name and file their documents into a rigid

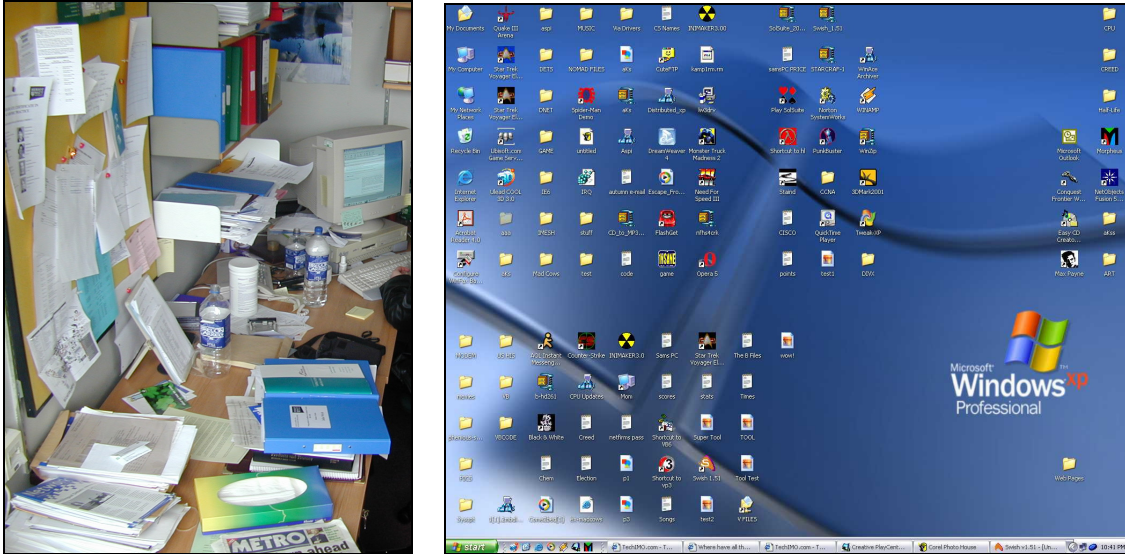


Figure 1. Present day virtual and physical workspaces.

Real desk, where items are casually organized (left). Typical virtual desktop with structured organization (right).

hierarchy (Figure 1 right). Filing typically requires more effort than piling and has been shown to have other negative effects such as encouraging premature filing of low value documents, or retaining documents with no relevance because of the effort that went into filing them (Whittaker and Hirschberg 2001).

In this thesis, we explore enriching the ubiquitous desktop metaphor computer users are comfortable with, by adding more realism to leverage some of the valuable characteristics of the real world. Our aim is not to debate the validity of the desktop metaphor (Johnson 1987; Gentner and Nielsen 1996; Tristram 2001), but to explore alternative designs to the pervasive desktop paradigm. Instead of the rigid organizational structures imposed by current virtual desktops, we use piles as a more casual organizational entity. A physics simulator allows objects to be dragged and tossed around with the feel of realistic characteristics such as friction and mass, and objects can collide and displace others. Our goal in adding physics to the desktop is to make the interaction feel more continuous and analog, rather than the discrete style imposed by digital computing. This potentially allows users to use the strategies they employ in the real world to both implicitly and explicitly convey information about their personal digital information.

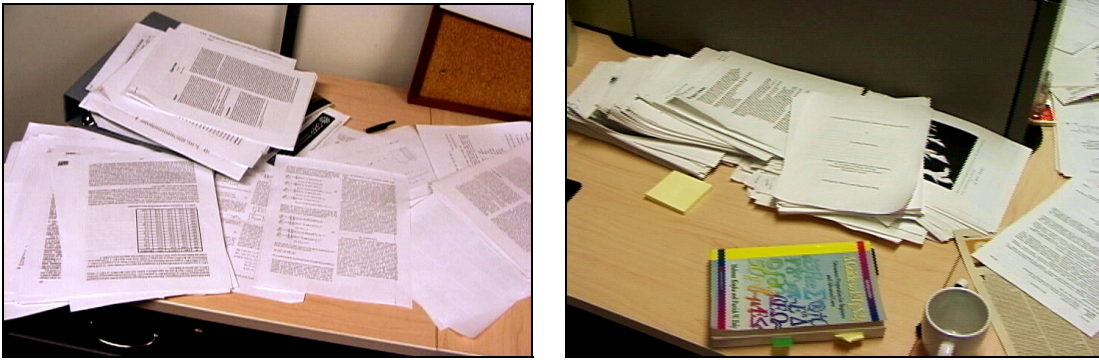


Figure 2. Examples of typical workspace organization.

Another goal of this research is to support casual organization of information in a manner where users are not forced to commit to categorization, such as the immediate naming and filing of documents. We also aim to leverage user's spatial memory and knowledge of how physical objects move in the real world. To aid in our exploration, we have developed a set of interaction and visualization techniques integrated into a prototype virtual desktop, called BumpTop, designed to be operated with direct pen input. Using a pen as the primary input mechanism can potentially enhance the feeling of realism and directness of manipulation since objects being acted upon are visible directly under the pen tip and move in response to pen motion. In addition, recent excitement and availability of pen-based TabletPC computers makes this an exciting area since there is no established standard interface for organizing personal documents, making it an area ripe for influencing with new designs.

1.1.1 Contributions

This research contributes original ideas, knowledge, interaction and visualization techniques to the field of human computer interaction. There are four major contributions detailed below.

Physical Desktop Prototype

The main contribution of this work is the BumpTop prototype which integrates a physical simulation with the standard idea of the desktop metaphor. Traditional desktop icons are given physical properties and some of the affordances of paper. We prototype a

suite of novel pen-based interaction and visualization techniques for manipulating documents to facilitate more expressive, casual desktop organization.

Interaction and Visualization Techniques for Piles

Though there have been previous investigations of piles, we integrate and significantly expand upon them. Our work investigates un-explored areas of pile-pile interaction, intra-pile interaction, pile widgets, piling of arbitrarily sized objects, implicitly created piles and hierarchical piles.

Pen-centric Interaction Techniques

We introduce novel, fluid pen-centric interaction techniques which synthesize in original ways recent research advances in pen-based techniques such as goal crossing, pressure widgets and pressure-based previews of data transformations. We re-formulate traditional interaction such as drag and drop, lasso selection and widgets and optimize for pen input. We also experiment with crossing targets that appear based on context. In addition, we emphasize interface discoverability and facilitate transition from novice to expert, which has hindered the usability of previous designs.

User Evaluation of Physics Desktop

We evaluate our designs revealing insights for future designers of physical and pen-based interfaces, shedding new light on recent pen-based interaction techniques that have rarely been used in combination with existing interaction paradigms. The method of our evaluation is qualitative observation of simulated tasks that approximate real world usage.

1.1.2 Thesis Overview

This research is based on work to be published in Agarawala and Balakrishnan (2006). This extended work provides an expanded survey of relevant literature, thorough motivation and examination of our design goals, further detail on the design and analysis of our prototype and novel techniques, detailed findings and implications from of the user evaluation and an extended discussion of future directions. A video demonstrating the

system described herein is published in the conference proceedings and is available online¹

In chapter 2, *Background*, we examine relevant prior literature including ethnographic studies of office organization, previous work on the pile metaphor, physically inspired interfaces, spatial organization in 2½D and 3D virtual environments.

In chapter 3, *Design Goals*, we establish six design goals based on strong principles of human computer interaction as well as our own experience. We draw on previous research and frame our work in the broader context of the field. These design goals enable the informed development of our prototype.

In chapter 4, *Prototype System*, we describe in detail our prototype system and suite of interaction and visualization techniques implemented to support physically based, pen-centric pile and desktop interaction. We highlight when we have disabled or tweaked the physics engine parameters and discuss performance concerns of a using a similar system in practice.

In chapter 5, *User Evaluation*, we present the findings of a qualitative user study in which participants completed simulated tasks and responded to verbal and written questionnaires on their experience. From the results we reflect on our design goals and discuss design implications.

In chapter 6, *Conclusions and Future Directions*, we summarize our work stating the contributions to the field. In addition we highlight future directions for enhancing our prototype and validating its use in a realistic setting with a longitudinal study.

¹ <http://honeybrown.ca/BumpTop>

2 Background

"If only smart people like your [stuff], it ain't that smart."

-Chris Rock

In this chapter we review prominent literature investigating relevant problems to our research. We begin by looking at ethnographic studies of how office workers process and organize paper information at their desks in the section *Ethnographic Studies of Office and Desktop Organization*. We see piling and filing emerge as paper processing strategies and review a study showing several disadvantages to filing. A study considering user behaviour using standard GUI desktops is also examined.

Next, in the section *Pile Metaphor*, we look at several different instances of research into piling in user interfaces. Our focus is on previous attempts at using piles to enhance the desktop metaphor used in today's operating systems. Other threads of pile research are also explored.

Finally, in the section *Physically Inspired Interfaces*, we look at user interfaces attempting to leverage techniques inspired by entities or behaviours with a basis in the real world. These provide excellent case studies with which to inform our research.

2.1 Ethnographic Studies of Office and Desktop Organization

There is significant literature supporting the inclusion of piles in electronic workspaces and suggesting that piling is, in many aspects, superior to filing in the real world. However, today's GUI desktops only support filing into the file systems rigid hierarchy. We now review pertinent ethnographic studies of how information is managed in the real world as well as in electronic systems.

2.1.1 How People Organize Their Desks

Malone (1983) conducted an ethnographic study of the organizational behaviors of ten office workers who do much of their work at their desks, and identified implications for designing computer-based organizational systems. He noted "subtle but useful aspects... that might inadvertently be lost in the new technology." Many of these insights are highly relevant to the development of our designs and will be carefully considered throughout. Participants ranged from secretaries, research scientists and managers. One of the principal insights provided by this work is the observation that people place items on their desk not just so they may find it later, but so it may serve as a reminder of tasks to be completed. One participant stated that their piles serve as a visual "prompter" of things to do and that "the further away from me it gets... the less often I am likely to look at it." It is estimated that 67 percent of piles on participant's desks were piles of things to do. Malone adds "since piles have no systematic order... their spatial location is often especially important in finding them."

Two basic paper organization strategies were identified in the study: 'piling' and 'filing'. Many participants found that categorizing and filing items was cognitively difficult. One participant remarked "It's interesting how hard it is to characterize these things... it's almost like leaving them out means that I don't have to characterize them... leaving them out means that I defer for now having to decide...how to use them, or decide where to put them." In addition most of the organization on people's desks consisted of untitled piles. Malone suggested that virtual desktops should provide untitled piles that support *deferred classification* as well as titled, logically arranged files. The use of 'electronic piles' on the virtual desktop should make the use of computers more natural. Schemes for further facilitating reminding in computer-based systems

include reflecting document priority using frequency (displaying higher priority items more often), size (higher priority items appear larger), location (placing important items in special places on the screen) and color. Priorities may automatically change over time. For instance, important items may increase in priority over time if left incomplete, such as important letters increasing in priority over time. Less important tasks such as reading a journal item may decrease until they disappear.

2.1.2 Piling vs. Filing

Whittaker and Hirschberg (2001) later compared real world piling and filing strategies, finding several advantages to piling. Qualitative and quantitative data was drawn from fifty office workers in a research laboratory who were about to move to offices with reduced local storage. This was opportune timing as participants would have recently handled and thought critically about most of their paper data. 15 out of 50 participants were identified as pilers, having greater than 40% of their information unfiled, though all participants used some of both piling and filing strategies.

Piling was found to be lightweight, casual, involved less overhead, and was easier to maintain than filing. Piles served as visual reminders and increased availability of recent information, concurring with Malone's (1983) previous findings. Pilers had increased availability to recent information and more frequently accessed their piles than filers accessed their file archives. It was observed that frequently accessed information, at times, percolates to the top of piles with less used information at the bottom, due to re-referencing of popular information. Pilers archives were also significantly smaller (60%) and was attributed to piled information being easier to discard.

Filing was found to have a large overhead for constructing and maintaining and some people filed only if they had secretarial assistance. Filers reluctantly discard information due to the effort put into initially filing it. Only 7% of filers found it "easy" to clean up their archives compared to 43% of pilers. Filers also *prematurely filed* documents later deemed to be of little or no value. In addition, sometimes more than one filing category applies, or an existing category is forgotten and a new one created. Some downsides of piling were also identified. Piling did not scale well and information was difficult to find once the number of piles grew large. Taken to excess piling can take over

every surface in an office, moving from tables to chairs and the floor. Despite the number of positive advantages to the piling strategy, and the fact that 30% of the participants in Whittaker's survey primarily piled their information, there remains no little technological support for piling in today's GUI desktops.

2.1.3 Finding and Reminding

Barreau and Nardi (1995) conducted a study on how people organized and searched their information within their computers. Their study revealed that people use the location of documents on their desktop to serve a critical reminder function, echoing Malone's (1983) earlier findings about people's organizational behaviours at real world desks. For instance, users would place items where they are likely to notice them as reminders of work to complete, or near the trash icon as reminders to delete them. Analogous to how office workers prominently position important items on real desks. We note that emphasis might be supported without using location to convey this information. For instance, our design supports combined increasing of size and weight to emphasize document importance or crumpling up documents to remind about deletion.

Two basic strategies were identified for finding files: location-based and logical search. Location-based search refers to the practice of users guessing and inspecting the folder in which they think the file is contained, iterating as necessary until they find the desired information. Logical search uses a text-based query to search filenames, using functionality built into the operating system. In both studies, "users overwhelmingly preferred location-based search". Logical searches were found to be slow and only effective if the filename is known. Users were hesitant to employ text-based search tools because of problems remembering filenames. In a more recent study Ravasio et al. (2004) found that expert users occasionally resorted to logical search, though only as a last-ditch effort whence files were considered lost. It is important to note however, this study was conducted a decade ago and the tools for logical search have significantly improved in that time. This fact could potentially lead to a different search preference if conducted today. Recent desktop search tools like Google Desktop (desktop.google.com) or Windows Desktop Search (www.microsoft.com/windows/desktopsearch/) make

searching almost instantaneous and search within the textual content of files as well as their filenames, reducing the requirements of remembering precise filenames. Nevertheless, the authors hypothesized that location-based search may have also been preferred since users were more engaged in investigating workspaces of their own creation. Further, location-based searches in GUI desktops remain popular today and allow users to make use of their spatial memory. Therefore, it is worthy investigating techniques for enhancing location-based search in our prototype.

Three types of information in computer systems were also identified: ephemeral, working and archived. *Ephemeral* information has a short shelf-life and includes things like “todo” lists or memos. This information was often loosely filed or left unfilled on the desktop. *Working* information had a shelf life of weeks or months and related to current projects. It was often categorized by both location and category due to its importance. Working information was frequently accessed and users had little trouble locating it. *Archived* information had a shelf life of months or years relating to often complete projects. Surprisingly, users keep little archived information in their systems. Also, every user in the study agreed that they rarely establish elaborate filing schemes for archived information because “they proved to require more time and effort than the information was worth.” This suggests an opportunity for exploring alternate organizational paradigms in virtual systems such as piling.

2.2 Pile Metaphor

We now examine prior research into using the pile metaphor in user interfaces. We look at both traditional and alternate pile visualizations that deviate from realistic representations, as well as applications of piling in novel domains aside from the desktop metaphor.

2.2.1 Pile Metaphor for Casual Organization

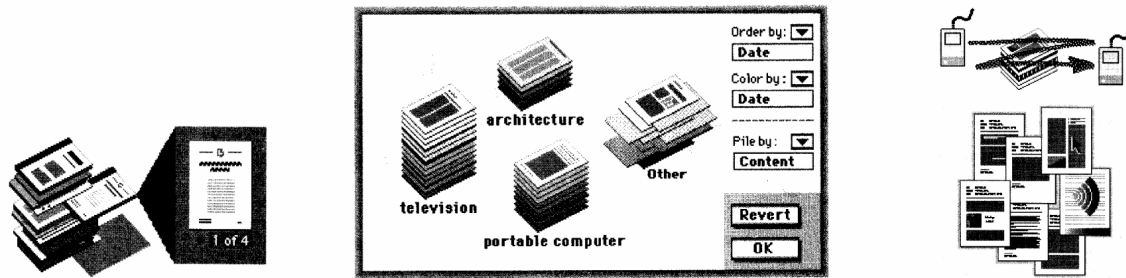


Figure 3. Pile Prototype.

From left to right: Browsing piles with 'document-pull out' viewing cone; Semantic pile visualization mode; Scrub gesture to spread out pile items. (from Mander et al. 1992)

The pile metaphor on the desktop was explored in a seminal paper by Mander et al. (1992) over a decade ago. Their prototype was based on a user-centered iterative design process involving thirteen office workers. Findings of the user interviews supported Malone's (1983) initial work mentioned previously while presenting new information. For instance, four different pile browsing methods were identified that office workers used on piles on their desks: edge, restack, hinge and spread out. In *edge* browsing people used the appearance of the outside edges of a pile to determine its contents. Common features used for pile identification were color, texture and thickness. With the *restack* method people lifted off items starting at the top of the pile, looked at them and then placed them elsewhere. With the *hinge* method, the pile was simply hinged open to view single items. Finally the *spread out* method arranged all items within a pile so the contents could be viewed in parallel. We note that any implementation of the pile metaphor should respect and potentially support these real-world pile browsing behaviours, as we aim to in our designs.

Gestures and interaction techniques (sometimes modal) were prototyped for browsing and manipulating piles and prototyping 'casual organization' on the desktop. Documents could be added to an existing pile by dragging it on top and upon mouse release the document would be animated moving to the top of the pile. Virtual pile browsing techniques were informed by the four real world methods identified previously. For instance, scrubbing the mouse horizontally over a pile resulted in a spread out representation of its contents (Figure 3, right). Restack-like functionality was also

supported by scrubbing the mouse vertically on a pile invoking a viewing cone which allowed detailed viewing of a single piled item in context (*Figure 3*, left). Three different viewing cone representations were evaluated resulting in a strong user preference for some sort of visual displacement between the target item and its pile. In addition, two piling models were explored: pile-centered and document-centered. When dragging on a visible portion of an item in a document-centered pile it is removed from the pile and becomes the active selection. In pile-centered mode however, piles behave more like traditional desktop folders where clicking on any portion of the pile moves it as a unit.

Another aim of Mander et al.'s system was to support piles that were automatically constructed and reorganized in the way a secretarial assistant might do for a manager. Pile creation rules were either defined by the user or, in theory, could be generated based on sample documents given by the user, as some information retrieval literature suggests. Automatically created piles are given a tidy, orderly appearance, while user created piles are disheveled looking with documents randomly displaced, parallel to each other (*Figure 3*, center). The follow up work to Mander et al.'s initial investigation of piling looked at piles from an information retrieval standpoint (Rose, Mander et al. 1993). It dealt with, amongst other things, techniques for automatically and manually creating piles and sub-piles based on content.

We draw from and significantly expand upon Mander et al.'s designs. Our work investigates un-explored areas of pile-pile interaction, intra-pile interaction, implicitly created piles, hierarchical piles, pile widgets, the transition between unpiled and piled information and piling of arbitrarily sized objects. In addition, we emphasize interface discoverability which has hindered the usability of previous designs. Also, some of Mander et al.'s techniques were developed in isolation. For instance, sorting piles required a special mode with its own interface inside a dialog box, and is not integrated with the main display. There was also pile-centered versus document-centered models of piles. Our implementation simultaneously supports the movement and manipulation of piles as a unit or any number of its sub-items. Our techniques are integrated into a single, largely modeless interface that can be operated with just the pen.

2.2.2 DynaPad

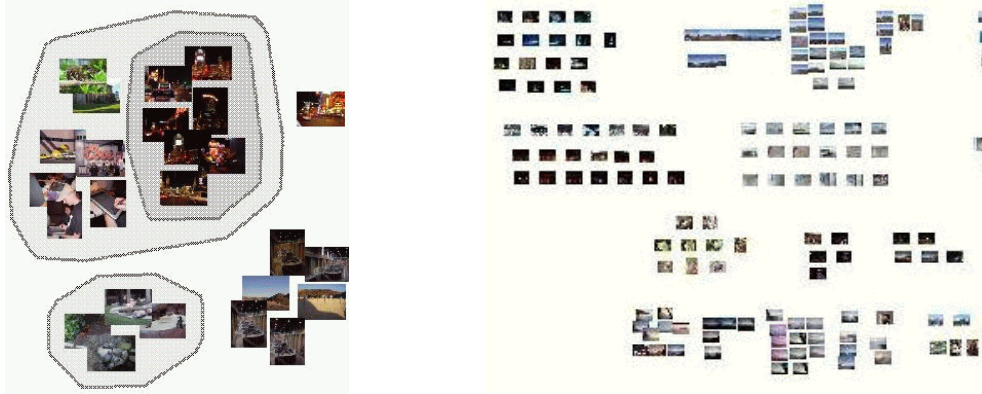


Figure 4. Dynapad.

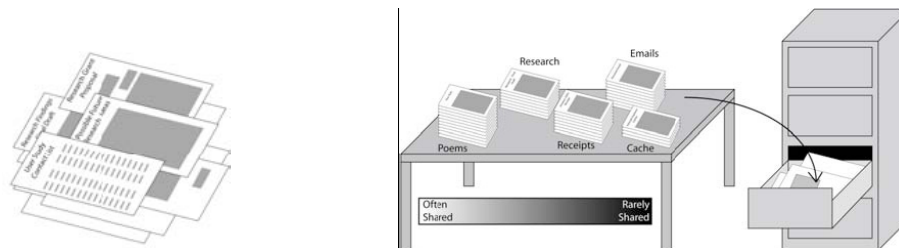
DynaPad's "open" pile representation (left) and implicit piles (right) (from Bauer, Fastrez et al. 2005)

DynaPad's (Bauer, Fastrez et al. 2004; Bauer, Fastrez et al. 2005) "open" pile representation lays out entire collections of photos side-by-side on a zoomable Pad++ (Bederson and Hollan 1994) based workspace. This representation avoids occlusion of stacked items with each other, but results in higher visual load and greater screen real-estate requirements. The latter issue is mitigated because of the infinite canvas. "Open" piles also aim to enhance remindability through visibility of all sub-objects, though this diminishes when the workspace is zoomed out and thumbnails become small. Implicit piles are supported (Figure 4 right) and can be turned into explicitly piled "clumps" by selecting the command from a contextual menu and then encircling items upon which their containing area is shaded and boundaries drawn (Figure 4 left). This shading technique in combination with presenting the entire hierarchy at once makes sub-piles easy to see. Items are added or removed from clumps by dragging them in or out of the boundary region, which adjusts accordingly. A sophisticated algorithm for membership-update of piles is presented. When zoomed out and clumps appear small they are fused together to prevent accidental item removal. For manipulating and browsing clumps tray, mutator and lens tools are provided with overlapping functionality and subtle differences that may be confusing for users. The tools all arrange source items into a grid arranged by date. The *tray* re-arranges photos dropped inside it, which are then moved when the tray is moved. The *mutator* temporarily re-arranges items beneath it but return to normal

when the mutator is moved away. The *lens* projects copies of items beneath it while leaving original items untouched.

Our alternative stack representation is in the spirit of familiar real-world piles and does not require a scalable interface, which presents its own interaction issues such as the user constantly managing the camera view. For instance, to move between an overview of the workspace to a specific pile, or from pile to pile often requires several manual pan and zoom operations by the user. Also, our linearly ordered piles support fluid sorting and re-ordering in place avoiding the need for complex external tools like trays or lenses. The DynaPad pile creation is not fluid requiring and is modal, requiring a selection from a menu. Also, the encircling used to create piles precludes lasso selections and crossing based selection of objects without introducing selection heuristics or modes.

2.2.3 Other Uses of the Pile Metaphor



*Figure 5. Piles for social navigation and security
(from DiGioia and Dourish 2005)*

Previous work has also looked at piles in different contexts. DiGioia and Dourish (2005) used a pile visualization to aid ‘social navigation’ and security. Ruffled piles were used to indicate information a group of users frequently accessed. To remove documents or piles from public access they could be moved to a ‘filing cabinet’ (Figure 5 right). This was presented as a natural and easier to understand alternative for determining folders to be shared over the Kazaa peer-to-peer network. A task that often confuses users resulting in accidental sharing of sensitive documents and violations of privacy (Good and Krekelberg 2003). Good and Krekelberg also found users were occasionally sharing much more than they thought. Using a pile visualization of shared groups of documents DiGioia and Dourish enabled users to view at a glance the magnitude of shared information, revealing discrepancies between intended and actual shared

information. This improved upon hierarchical folder listings with checkboxes indicating sharing, which users found troublesome. This work also suggests that piles may be useful for representing data beyond traditional document objects.

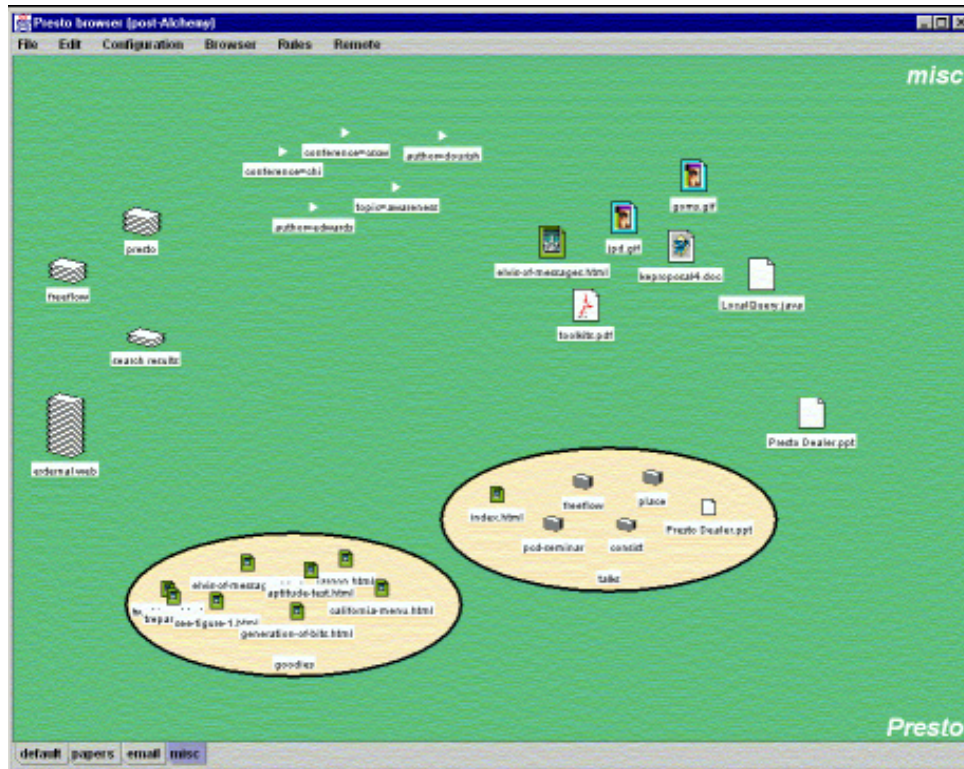
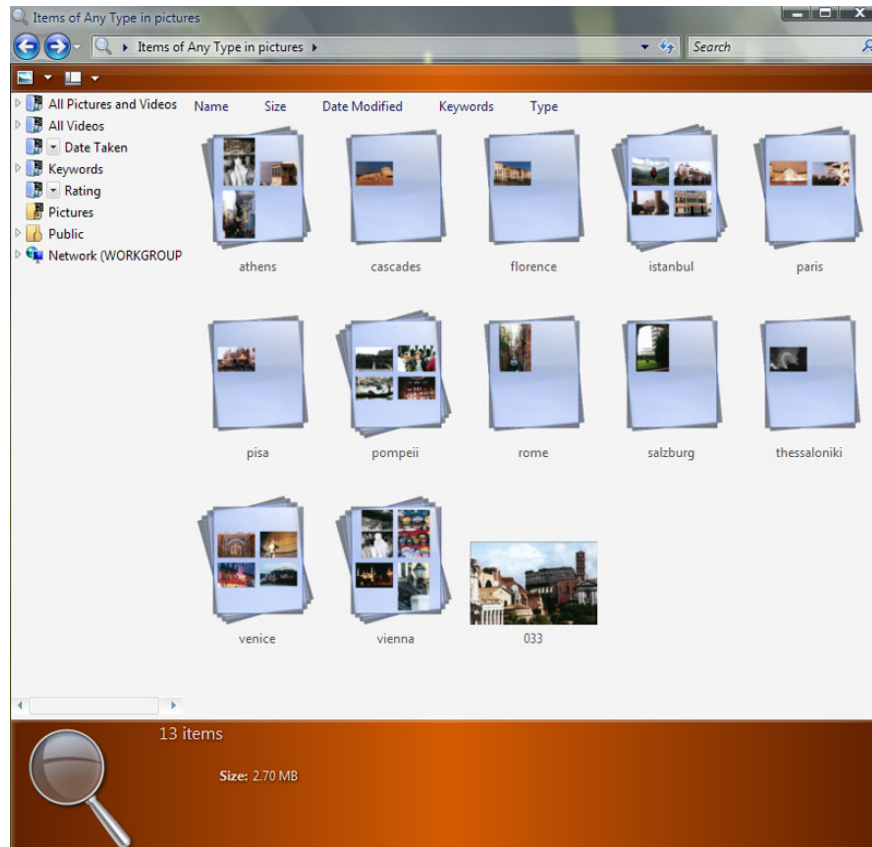


Figure 6. Presto fluid workspace with oval piles on the desktop
(from Dourish, Edwards et al. 1999)

Dourish, Edwards et al. (1999) used mixed pile representations in a single interface, drawing upon both Mander et al.'s (1992) “closed” piles and DynaPad's (Bauer, Fastrez et al. 2004; Bauer, Fastrez et al. 2005) “open” pile visualizations. In the Presto document space browser closed stacked piles appeared on the desktop representing document collections. When browsed these expanded to shaded ovals containing sub-documents as icons (*Figure 6*), similar to DynaPad piles. Sub-piles were represented as nested closed piles. Their usage of piles was preliminary and none of the richer interaction techniques of either prior pile prototypes were incorporated.

In the forthcoming Microsoft Windows Vista (www.microsoft.com/windowsvista/) piling is integrated into the standard windows explorer folder view. “Stacks” can be automatically generated as a way to visualize items by a specific attribute (*Figure 7*). For example, documents in a folder may be

stacked by author to receive a quick overview of the number of different authors and volume of their works, information not immediately apparent in a flat directory listing. According to the information available, these stacks will only be used as a visualization aid and otherwise behave as folders when double-clicked. This implementation would benefit from the addition of our pile interaction techniques and widgets for browsing, manipulation and interaction with piled and un-piled documents.



*Figure 7. Microsoft Windows Vista stacks
(from www.microsoft.com/windowsvista)*

While there have been previous investigations of the pile metaphor in user interfaces, we integrate and significantly expand upon them. Our work investigates unexplored areas of pile-pile interaction, intra-pile interaction, pile widgets, the transition between unpiled and piled information and piling of arbitrarily sized objects. In addition, we emphasize interface discoverability which has hindered the usability of previous designs. We also incorporate several advances in browsing techniques developed for other applications (Carpendale, Cowperthwaite et al. 1999; Carpendale and Montagnese

2001; McGuffin, Tancau et al. 2003). Integration of physics and the pen further opens the design space for interesting additions such as physically inspired interaction and visualization techniques, fluid pen-based interactions and experimenting with different physics-engine parameters.

2.3 Physically Inspired Interfaces

We review interfaces containing metaphors based on real world entities or behaviours. We observe cases where robust physically based algorithms provide richness resulting in a more continuous, analog feel and more “in-between” states. These physically based approaches move away from the traditional discrete nature of user interfaces and open up interaction possibilities. In addition, drawing inspiration from the real world imbues interface elements with a tangibility that may harness our capabilities as physical beings.

2.3.1 WebBook and the Web Forager



Figure 8. WebBook and Web Forager.

(Clockwise from top-left) Ruffling through pages of a WebBook; Organizing WebBooks in Web Forager; Document Lens applied to a WebBook; Bookcase for tertiary storage (from Card, Robertson et al. 1996)

A more realistic visual presentation of web browsers can be seen in the in WebBook and Web Forager (Card, Robertson et al. 1996). Users can organize groups of webpages into 3D WebBooks that embody the physicality of a book. Each webpage is represented as a single page in the book, with a traditional scroll bar(s) for the webpage. WebBooks are browsed by flipping through pages, invoked by clicking white-space on the left or right pages of the book for flipping forward or backward respectively. Pressing and holding down the mouse triggers rapid ruffling of pages to quickly and conveniently scan several web pages, like one might do with a physical book (*Figure 8*). This type of browsing behaviour is not supported in traditional web browsers. While webpages may be collected into WebBooks as an alternative to traditional text-based bookmark mechanisms, automatically generated WebBooks are also discussed. For instance, *Relative-URL books* are formed by gathering all relative links rooted in a particular webpage. The authors claim this heuristic produces “coherent and interesting books.” Another option is creating a *Search Report* book that allows one to flip and rifle through search results on a particular keyword in a physically inspired and potentially more natural way. Card et al. states another benefit of the presentation: “page turning of a book conveys information about the relationship of pages, the direction you are moving in the book, the size of the book, and the contents of the book.”

Multiple books and web pages can be stored in the Web Forager workspace supporting immediate storage on a virtual desk, or tertiary storage in a bookcase. Clicking on objects in the different locations smoothly zooms to them for focus. This leaves view management to the computer instead of the user, as in other systems like DynaPad (Bauer, Fastrez et al. 2004; Bauer, Fastrez et al. 2005) reviewed earlier. Clicking a hyperlink on a page in a WebBook will smoothly animate the flipping of pages to the desired page if it resides in the same book. If the link resides in another book the current book will be put away on the virtual desk and the new book accessed from the bookcase, or pulled from the web as appropriate. Though WebBooks open to where they were last closed when accessed, the interface misses a global “back” button as in more traditional browsers allowing the convenient retracing of steps.

Card et al. go beyond book metaphor and support browsing techniques that benefit from the underlying computer. For example a Document Lens (Robertson and

Mackinlay 1993) is used to visualize the entire WebBook contents at once. The WebBook and Web Forager interface also presents a good model for a mixed mode interface, where physically based techniques are integrated comfortably with more traditional elements, in this case web browsers, into a single seamless 3D interface.

2.3.2 Paper-based Metaphors for Windows

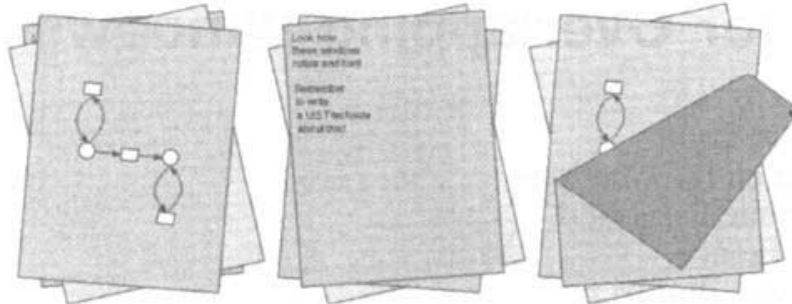


Figure 9. Paper-based metaphor supporting rotated and peeled-back windows (from Beaudouin-Lafon 2001)

Beaudouin-Lafon's (2001) physically-based designs deal with the problem of occluded, overlapping windows by employing a physical metaphor, rethinking windows as paper stacked in piles. It is observed that "although physical books and sheets of paper on a desk are rectangular, they are rarely perfectly aligned. This gives the user clues for locating them and an easy way to access them even when they are stacked." The piling concept is applied to windows and adding rotation increases visibility of otherwise occluded or overlapped items underneath. An algorithm is described for fluid integration of rotation and translation of windows with a single point of 2D input, such as that supplied by a pen or mouse. Beaudouin-Lafon's designs improve on reality since window rotation is constrained to only allow a small degree of off axis rotation. This ensures that icons are always readable and unwanted configurations impossible such as upside-down windows.

Additionally, windows can be peeled freely from any edge, like pieces of paper. A robust algorithm for peeling is introduced that influences future systems (Roussel 2003; Dragicevic 2004), including our own. Peeling back windows provides an elegant, easily understood visualization grounded in realistic behaviour and maintains spatial context as windows positions remain unchanged. This avoids problems associated with

traditional window management techniques. For instance, manually re-arranging windows to view occluded ones is time consuming and disrupts the spatial layout of the windows. Even recent solutions like the zoom-based Exposé approach built-in to MacOS X (www.apple.com/macosx/features/expose/) does not allow you to view windows in the spatial context they have been positioned in by the user. Fold'n'Drop (Dragicevic 2004) later combines the above peeling technique with a crossing-based gesture at the window edges to trigger the peeling. This gives the effect that windows can naturally be peeled by “pulling” at their edges, increasing the tangible feeling of this technique. Undo is also supported by “pushing” edges back to their original place with the same cross gesture in the reverse direction,. Fold'n'Drop lives comfortably alongside existing point-and-click techniques as demonstrated in the proof-of-concept implementation of it on the Windows XP platform called OriMado (www.kmonos.net/lib/orimado.en.html).

2.3.3 Higher Fidelity Physical Simulations



Figure 10. 3D Paper flyer metaphor simulated with cloth and wind.
(from Denoue, Nelson et al. 2003)

Denoue, Nelson et al. (2003) increase the fidelity of the paper metaphor by using real-time physical simulation of cloth (House and Breen 2000). Simulated cloth is texture-mapped with digital content (e.g. Web pages, images, video) and pinned up to a virtual bulletin board where it gently blows in the wind. The authors claim this creates a “visually interesting and inviting metaphor that is more appropriate for public bulletin boards” over traditional techniques. Dragging at cloth edges allows passers by to interact

and manipulate the digital content. The resulting system moves very realistically and we suspect the subtleties in the motion have a positive effect on the *feel* of the system to users. During AttrActive windows development coworkers approached the plasma display it was running on and commented on its appearance, providing anecdotal evidence that is successfully attracted and engaged users. An informal user study with six users confirmed the paper flyer metaphor was popular. Users easily folded and rotated the flyers suggesting transfer of real world knowledge of how physical items can be manipulated and respond to movement. One problem that arose in the study was interacting with digital content when distorted by the cloth presentation. For instance, clicking hyperlinks on cloth webpages was difficult. Though it was reportedly left for future work, the robust dynamic cloth simulation readily supported multi-point input from both multiple fingers and multiple users.

2.3.4 Tossing Interaction

Tossing is a physical action used as an interaction technique in a collection of previous work. Windows can be tossed across a workspace on DynaWall (Streitz, Geißler et al. 1999), a wall-sized interactive display. Here tossing provides the benefit reducing the physical effort required to move a window from one end of the workspace to the other, as a quick flick gesture can be made to “throw” a window as opposed to walking while dragging the window. A tossing gesture is also useful on the smaller scale. For example, nVidia’s nView extensions (www.nvidia.com) support tossing windows from one monitor across to another, saving a round trip across monitor(s) with the mouse. In addition, the casual nature of the toss allows the user to coarsely specify their actions, as situations arise when a user quickly wants a window moved out of the way or put in some general region “over there”, without worrying about specific positioning. From our own anecdotal experience, we believe there is something tangible and satisfying about users throwing items in an interface. We have noticed tossing interaction often drawing excitement and affording a level of playfulness towards interfaces.

Along with the casual nature of the toss comes its inherent imprecision. A recent attempt at improving this, by going beyond what is physically possible, can be seen in the

Push-and-Throw technique (Collomb and Hascoët 2004). Push-and-Throw draws the trajectory and end result of a toss while the mouse is down, without actually moving the target item. This gives a preview of the toss and presents the opportunity for the user to correct an inaccurate toss before the mouse is released. This gives a level of precision not provided by a traditional toss. However, the continuous visual feedback is removed reducing the beneficial tangible, physical feeling of the toss gesture and may be less satisfying or engaging than a physically based toss.

2.3.5 Spatial Organization in 3D



*Figure 11. Data Mountain prototype
(from Robertson, Czerwinski et al. 1998)*

Previous work has shown the benefits of a spatially based organization in computer systems. The Data Mountain (Robertson, Czerwinski et al. 1998) attempts to leverage spatial memory by organizing webpage thumbnails on a perspective 2½D plane. They state their results are generalizable to any sort of information organization. The design rationale is presented here:

“The primary motivation for the design of the Data Mountain came from a desire to leverage natural human capabilities, particularly cognitive and perceptual skills. In particular, 3D perception is used to allow for the representation of a large number of web page thumbnails with minimal cognitive load... In the real world, spatial memory often aids us in finding things. For example, when we place a piece of paper on a pile in our

office, we are likely to remember approximately where that paper is for a long time. Our hope is that this ability works as well in a virtual space as it does in the physical world.”
(from Robertson, Czerwinski et al. 1998)

Data Mountain supported a larger number of documents than a traditional desktop, due to its 2½D presentation where document sized diminished as it receded into the screen plane. To aid cognitive grouping passive landmarks were placed on the Data Mountain surface, in the form of colored circles on the organization plane.

A user study was conducted amongst thirty two experienced internet explorer 4 users. Data Mountain reliably facilitated rapid retrieval of web pages in comparison with internet explorer’s hierarchical text-based bookmark mechanism, using organizational structures of the user’s own creation. This suggests that spatial memory does indeed factor into 2½D virtual workspaces and provide benefits over largely text-based approaches. Amazingly, even after having not seen their organizations for four months retrieval times were not significantly slower (Czerwinski, Dantzych et al. 1999). Further, after a short period of adaptation time retrieval times were just as fast when thumbnails were replaced with entirely blank outlines.



Figure 12. Task Gallery prototype.

Art gallery metaphor used for task management (left),

Data Mountain-based “Start Palette” (from Robertson, Dantzych et al. 2000)

The Task Gallery (Robertson, Dantzych et al. 2000) used a 3D art gallery metaphor for task and window management. Windows could be arranged on the walls of the Task Gallery, and the user could virtually navigate around the space (*Figure 12* left). The traditional “Start Menu”, found in Windows operating systems, was replaced with a *Start Palette* (*Figure 12* right). The Start Palette was a Data Mountain (Robertson,

Czerwinski et al. 1998) of application icons, held in the users left virtual hand. A user study was conducted involving eleven users and the Task Gallery was shown to help with task management and was enjoyable to use, despite the sophisticated view navigation required by the environment. It also demonstrated effective transfer of spatial memory and cognition into 3D. After user's arranged windows in the virtual gallery and were removed from the computer, they were able to draw their arrangement of tasks in the 3D space on paper with some accuracy. This echoed results supporting the view that spatial memory is effective in computer environments such as the earlier Data Mountain study. Further, the study suggested transfer of other real world behaviours into the virtual space. For instance, people were more likely to place items on the walls rather than the floor or ceiling, perhaps reflecting what they were comfortable with in real world placements.

3 Design Goals

"We haven't gone beyond the surface. Let's Risk All!"

-Pablo Picasso

Early in our exploration we identified six design goals that would influence our interaction and visualization designs. These goals are based on strong principles of human computer interaction as well as our own experience. In this chapter we detail the motivations of our design goals and examine relevant previous attempts, framing our work in the broader context of prior research.

Realistic Feel

Our vision for the desktop is to recapture the lost richness and expressiveness found on people's desks in the real world, and our everyday capabilities to organize and manipulate such information. Objects on the physically enriched desktop should move realistically, allowing users to leverage their knowledge of how things move in reality. For example, after some experimentation with the system a user should be able to figure out that tossing items to a corner will cause them to collide and pile up. These ideas are similar to the spirit of the Tangible Bits vision for Tangible User Interfaces (TUIs), trying to recapture the lost benefits of physical artifacts in the transition to digital media. "GUIs fall short of embracing the richness of human senses and skills people have developed

through a lifetime of interaction with the physical world” (Ishii and Ullmer 1997). The ubiquitous computing vision (Weiser 1991) had the traditional computer disappearing in favor of several context aware devices integrated seamlessly into the environment, intended to support and co-exist within everyday life instead of transforming it. Despite their benefits there has not been a widespread adoption of TUIs or ubiquitous computing. This is potentially caused by the high implementation cost. Each application of these interfaces requires a specific hardware design. This often means a specially designed form factor as well as hardware and software development. Though these requirements have been eased by hardware prototyping toolkits like Phidgets (Greenberg and Fitchett 2001), TUIs and ubiquitous computing are not pervasive in computing today.

This contrasts with computer GUIs which have been very successful and widely prevalent in modern computing. GUIs provide a wonderful generality which can be used for wide varieties of interfaces and interaction paradigms, from 2D data analysis to 3D multimedia gaming. The desktop metaphor is the established and widespread interface on the majority of personal computers. However, the power of physicality and human capabilities for manipulating physical objects is lost. We take a middle ground approach, attempting to recapture the lost physicality while exploiting the ubiquitous and generic power of computer GUIs and leveraging comfort with the pervasive desktop metaphor. By using a high fidelity physical simulation of movement we hope to bridge the gap between impoverished computer interfaces and the rich, tangible real world environment people are accustomed to on their desks. In addition, physically-based movement is livelier and increases positional degrees of freedom for potentially more expressive layouts than a traditional desktop, which keeps icons axis-aligned in a 2D representation, where icons feel more like pixels than tangible objects.

We go beyond previous attempts at physically inspired interfaces which often introduce an isolated physically-based concept amongst a more traditional GUI (Beaudouin-Lafon 2001; Dragicevic 2004), or present interfaces with realistic 3D models but without high-fidelity physically simulated movement in response to user interaction (Card, Robertson et al. 1996). We present the following spectrum of existing HCI interfaces and paradigms (*Figure 13*), beginning with text-based command line interfaces as they approach real desks. Prior work is plotted according to how it approximates or

delivers the benefits of interaction and manipulation in the real world. For instance, augmented reality approaches that enhance real desks with a projected digital overlay such as DigitalDesk (Wellner 1993) appear on the far right of the spectrum. While purely virtual command line interfaces with no basis in the physical world appear on the far left. These placements are approximations but help frame our work in the broader context of human computer interaction.

The introduction of a realistic feel brings a sort of “physical metaphor” to interactions on the desktop. The benefits of using familiar analogies to ease the learning of a new system is also favored by others (Anderson 1980; Carroll and Thomas 1980). The use of metaphor is the first principle of Human Interface Design in the Macintosh Human Interface Guidelines. “You can take advantage of people's knowledge of the world around them by using metaphors to convey concepts and features of your application” (Apple Computer Inc. 1992). The Macintosh effectively leveraged metaphor by being the first commercially successful application of the desktop interface, invented by XeroxPARC (Smith 1982). Metaphors can also prove restrictive and their limitations are discussed below under the design goal *Disable Physics as Necessary*.

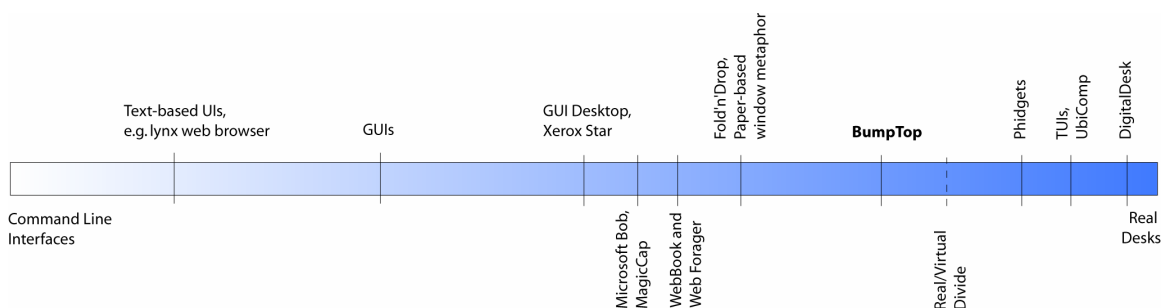


Figure 13. Spectrum of virtual user interfaces to real desks.

Tangible, Paper-like Icons

Despite the metaphor, current virtual desktops bear little resemblance to the look or feel of real world desktops. GUI desktops typically contain several low-resolution 2D bitmaps intended to represent information and applications residing in the computer. These are axis-aligned and often automatically arranged into a strict grid layout, at times negating personal organization and spatial memory. Grid layouts may be a convenient

layout algorithm but are seldom used for organizing on real desks. Their uniform presentation makes it difficult to emphasize or differentiate items subtly from the rest. Each desktop icon must be manually named with a subset of alpha-numeric characters dictated by the operating system. We note that filenames are a requirement imposed by the underlying file system implementation, but aren't necessary in the interface. The display of filenames can be truncated if they are so lengthy that they will interfere with the display of adjacent icons. Directories are similarly represented by whose contents are revealed when explicitly requested into rectilinear windows.

In contrast, a workspace in the physical world typically has layers of paper documents, binders and other objects arranged in a way that provides considerable subtle information to the owner. For example, items are often casually placed but their spatial position and orientation are usually meaningful. Closer items can indicate urgency, and layers of paper documents are “automatically” ordered chronologically because new items are typically placed on top. All arrangement is typically done by the owner of the desk over a period of time, though not always explicitly. Collections of related documents are often organized or form into lightweight piles, whose appearance exposes a rough approximation of the size and contents at a glance. In their workspaces, people will often employ sophisticated systems of rotating and pulling out items within piles for indicating separation of content or emphasizing visibility or remindability.

Our goal is to encourage real world organizational style by having documents *feel* like tangible physical objects through affordances. Affordances (Gibson 1979) are properties of physical objects that suggest how to use them. For example, a handle on a door affords opening. You might say that current icons in traditional GUIs “afford” clicking, appearing as a rectangular target of pixels, for which double-clicking is the established convention. By supporting physically realistic responses to user interaction in our interface, we hope our icons will afford the expressive organization of real desks. Beyond physical simulation of movement, we aim to increase the fidelity of the physical metaphor by embodying icons with physical properties such as those of paper. Sellen and Harper (2003) identified beneficial physical properties of paper including being thin, light, porous, opaque and flexible. Further they observed that these properties afford different human actions such as grasping, carrying, manipulating and folding.

Disable Physics as Necessary

Our intention is to leverage the beneficial properties of the physical world, but not be overly constrained by or dogmatically committed to realism. When appropriate, we intend to exploit the power of the underlying computer and turn off or alter the realistic physical simulation when it proves limiting, counter-intuitive, or where we are able to improve on reality. For instance, in the real world a binder can “pop” spreading its contents chaotically all over and without an undo. Though this could be simulated with our physics engine it is extraneous and unnecessary. We draw upon prior work in physically-inspired interfaces that limit physical simulation to provide “better than the real world” capabilities. Beaudouin-Lafon’s (2001) paper-based window metaphor restricts rotation of windows to maintain readability while reducing the ability to make a mess and the need for corrective manual re-orientation by the user. Collomb and Hascoët (2004) enhance tossing models by displaying the trajectory as the object is dragged, allow corrections to be made before the mouse is released. One instance where disabling the simulation may be beneficial in our interface is in the prevention of unwanted collisions between explicitly organized items. Further, computer simulations of physics are not perfect and suffer from numerical instability manifesting in jitter, interpenetration and unnatural movements. Mindful deviation from physics can reduce these effects.

Despite the benefits of metaphors in leveraging existing user knowledge outlined previously, overly literal application can result in awkward navigation and unnecessary complexity in order to maintain the metaphor. Gentner and Nielsen (1996) provide the following critique of metaphor driven interfaces:

“Many recent interfaces have tried to overcome the limitations of the desktop metaphor by extending it to some other room or building metaphor (e.g., [Microsoft] Bob or MagicCap) or to a village metaphor (e.g., eWorld). These 3D designs try to emulate virtual reality on a flat screen but often seem to introduce a level of clunky indirectness in achieving common user goals. They are navigationally cumbersome, asking users to go to the “other end of town” to pick up their email from the Post Office, and interactionally cumbersome, overloading users with additional windows and other interface elements necessitated by the metaphor but not by the user’s task. Although the use of metaphor may ease learning for the computer novice, it can also cripple the interface with irrelevant limitations and blind the designer to new paradigms more appropriate for a computer-

based application.”
 (Gentner and Nielsen 1996)

In our designs we will take care to not be overly constrained by maintaining physically reality as previous designs like Microsoft’s Bob (1995) (Figure 14 left), MagicCap (1994) (Figure 14 right) and eWorld (1994) have. When design opportunities are presented that go beyond the physical metaphor, they will be examined and carefully pursued as appropriate. Designs that introduce the “clunkyness” described in the passage above will be avoided or discarded as they arise.



Figure 14. Excessive use of metaphor.
 Screenshot of Microsoft Bob (left), General Magic’s MicroCap (right)

Optimize for Pen Interaction

Our work targets pen-based computers to capitalize on the opportunity presented by the recent excitement and availability of TabletPC based computers (en.wikipedia.org/wiki/TabletPC). In addition, there is currently no established standard for pen-based interfaces, making it an area ripe for influencing with new designs. TabletPCs integrate portable laptop functionality with a digitizing tablet built into the screen. The tablet provides pressure sensitive input via a pen with a barrel button as an additional input channel. Not all TabletPC configurations come with keyboards, and those that do are convertible to a “slate” mode where the screen is flipped around and folded down so the keyboard is inaccessible. The display can then be held comfortably in one hand like a pad of paper while using the pen to interact with the other free hand. Other TabletPCs are purely “slates” without a keyboard. To target the largest pen-based

audience we focus on designing for the situation where no keyboard is available for triggering interactions. Another benefit of using the pen as the primary input mechanism is the potential enhancement of the feeling of realism and directness of manipulation, since objects being acted upon are visible directly under the pen tip and move in response to pen motion.

Pen-based computers present a challenging design space as input is often limited to four multiplexed input channels: a single 2D location on the screen, pen contact/hover information above the screen, pressure and the binary pen barrel button. These channels must all be thoughtfully used as previous research has shown they each present their own design issues. For instance, even double-clicking (one of the foundations of interaction in traditional mouse interfaces) is very difficult with the pen. This is due to the difficulty in rapidly hitting the same target twice without moving the cursor position (Accot and Zhai 2002). Also, while most pens report 1024 distinct levels of pressure Ramos and Balakrishnan (2004) show that only six or less levels can be used effectively by the user. Attempting to use more degrades performance drastically. Further, any use of the pen button has been found to be troublesome since users often accidentally push it while gripping the pen to write (Fitzmaurice, Khan et al. 2003; Miller 2005). Some pen designers combat this problem by disabling the button altogether. Hardware manufacturers are mitigating false button presses with new pen designs that move the barrel button higher up on the pen. In addition to the individual issues presented by pressure, pen contact and the barrel button, it can also be difficult for users to coordinate using them all at once. Therefore, the four potential input channels offer more limited than it they may initially seem.

There has been success in working within the constraints of pen-based computers. For instance, recently Ramos and Balakrishnan (2004) introduced pressure widgets, giving users continuous visual feedback with novel visual cursor designs that indicate pressure levels and what interactions are offered. The use of goal crossing instead of double-clicking (Accot and Zhai 2002), or as the basis for an entire pen-based interface (Apitz and Guimbretière 2004) were promising. We draw inspiration from these techniques in our interface favoring crossing when applicable and exploiting the pressure sensing capabilities of the pen while avoiding designs that are problematic for pen

interaction such as small targets, and double-clicking. While the pen input is limited in some respects it affords a fluidity and elegance not found in traditional mouse or keyboard based interaction. We strive in our interface to support fluid interaction where user flow is not disrupted, but supported with non-obtrusive techniques. We follow in the spirit of others who have demonstrated fluid techniques in other pen-based domains including whiteboard interfaces (Pederson, McCall et al. 1993; Mynatt, Igarashi et al. 1999), large display interfaces (Guimbretière and Winograd 2000; Guimbretière, Stone et al. 2001) and generic pen-based techniques (Guimbretière and Winograd 2000).

Discoverable, Learnable Interface

Due to limited input possibilities in pen-based systems designers often turn to gestural interaction (Kurtenbach and Buxton 1991; Igarashi, Matsuoka et al. 1999; Hinckley, Baudisch et al. 2005). However, lack of discoverability and a mechanism for automated learning greatly hinders the usability of many previous gesture driven systems. SKETCH (Zelevnik, Herndon et al. 1996) is a pioneering gesture driven interface for design of 3D scenes. While powerful and expressive in terms of design possibilities, using the interface requires training in the large gesture vocabulary. For instance, there are 14 gestures (some with variations) for object creation (*Figure 15 left*) and 5 different gestures for camera control. In addition, different gestures must be invoked with specific mouse buttons and include pauses, which the authors admit disrupts user flow. GEdit (Kurtenbach and Buxton 1991) uses gestures like drawing the letter ‘c’ for cut functionality in the same stroke as encircling a selection to fluidly make and invoke actions on a group of objects (*Figure 15 right, bottom*). Other gestures included ending the stroke inside or outside an encircling of objects to delete or move objects respectively (*Figure 15 right, top and middle*). Even prior pile systems have gestures that do not support discovery. For instance, using a horizontal or vertical scrub gesture spreads out a pile or invokes a viewing cone for pile browsing (*Figure 3 right*) (Mander, Salomon et al. 1992).

Though shape-based gestures free users from selecting actions from distant menus in a seemingly intuitive manner, they are ultimately idiosyncratic and do not support discovery. They must be explicitly learned and their effective usage requires memorization. Shape-based gestures are also prone to being forgotten, especially when not frequently used. In addition, such gestures have inherent problems in their detection. They are based on users being able to reproduce a certain machine recognizable shape, which can be difficult for both the user and the computer. Recognition is exacerbated by having to support the subtle individual variations in different users of the system. While there have been recent advances in recognition techniques (Cao and Balakrishnan 2005) they are still not perfect and accuracy remains under 90%. The remaining false positives and negatives are frustrating to the user. In addition shape-based gestures limit inclusion of other interaction techniques. For instance, when in a gesture recognizing mode it is difficult to allow a freeform lasso selections which could accidentally trigger gestures if a gesture is accidentally drawn while creating a selection.

Our designs aim to rise above problems of discovery, learning, memory, recognition accuracy and limited compatibility in previous techniques. After learning a

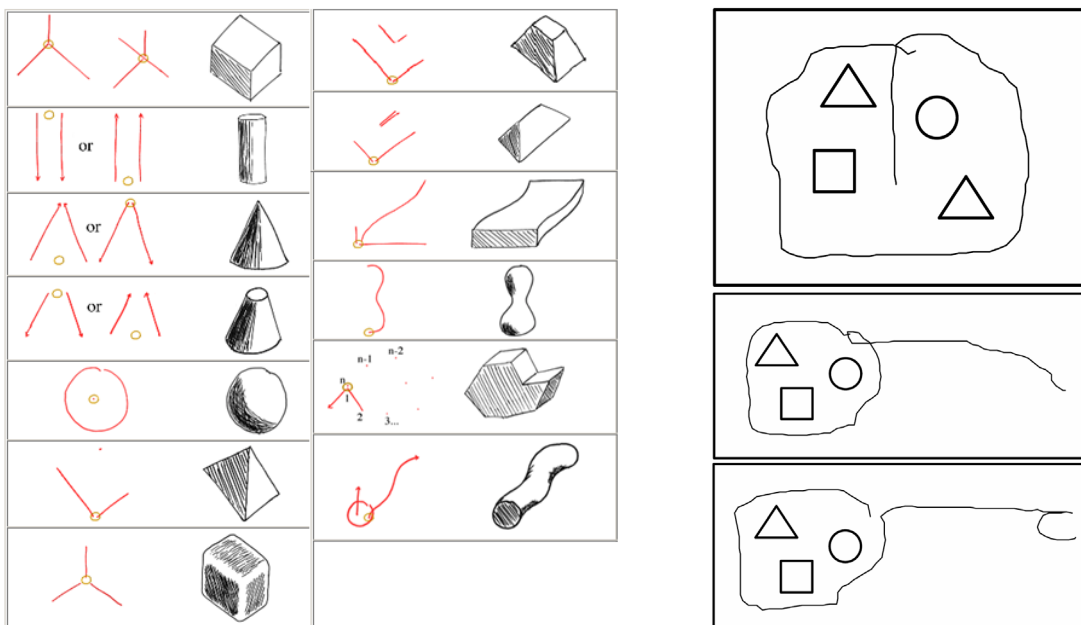


Figure 15. Shape based gestures in previous interfaces.

Left: Gestures for creating shapes in SKETCH (from Zeleznik, Herndon et al. 1996)

Right: Fluid gestures from GEdit for (from top to bottom) deleting, moving or copying objects (from Kurtenbach and Buxton 1991)

small set of basic interaction techniques, the user should be able to discover how to do more complex interactions on their own. Unlike many gestural interfaces, our goal is to avoid requiring the user to memorize a large gesture vocabulary before they can use the system effectively. We will leverage when possible existing discoverable techniques that foster smooth transitions from novice to expert behavior such as Marking Menus (Kurtenbach and Buxton 1993). In addition, we attempt to design self-revealing interaction techniques by using appropriate visual cues, and support transient easily reversible actions.

Smooth Transitions.

To avoid startling and confusing users, we employ smooth slow-in and slow-out transitions (Chang and Ungar 1993; Gonzalez 1996) for every change in data representation. It is well established (Woods 1984; Robertson, Mackinlay et al. 1991; Bartram 1997; Grossman, Balakrishnan et al. 2001) that it is easier for users to maintain a mental model of the data across smooth transitions and less time is spent comprehending the new data presentation. In addition, continuous transitions allow the user to maintain the context of their previous activities. McGuffin et al. (2003) use smooth transitions in combination with several browsing techniques of volumetric data to create easily understandable and visually appealing presentations. Transitions feel seamless for both abstract and realistic presentations. Further, this design goal is in line with our previously mentioned goal of *Realistic Feel* since there are no sudden, disjoint changes in visual presentation in the physical world. Objects are continuously moved from one position to the next.

4 Prototype System

"I've been taking some different tablets this time around."

-Peter Gabriel

The BumpTop prototype presents users with a perspective 2½D view onto a planar desktop surface tilted 25 degrees with respect to the camera (Figure 16). Our initial design had a top-down perspective view of the workspace which resembled the flat desktop users are accustomed to. However, users found it difficult to distinguish the depths of piles and confused them with single objects. To resolve this, we shifted to a 25° perspective view and added shadows to emphasize depth as in previous literature (Herndon, Zeleznik et al. 1992).

Motion is constrained to the desktop region by the walls enclosing it. The wall corners provide a natural place for documents to pile up on top of each other and act as a landscape feature that could aid in cognitive grouping of documents. As in the Data Mountain design (Robertson, Czerwinski et al. 1998), the desktop texture has a number of circles which act as “passive landmarks” that could aid in visually separating groups of items. However, users are free to place documents anywhere on the surface.



Figure 16. *BumpTop* prototype.

Our BumpTop prototype with piles as the fundamental organizational object, and physics simulation affording casual, potentially more realistic, interaction.

Documents are represented by icons as in today's popular GUI desktops. Icon geometry is a 3D cube squashed on one axis, and is texture-mapped on all sides so that when vertically stacked there is an indication of what the document type is. Being able to discern information from the edges of the icon supports a pile browsing behavior that occurs in the real world called *edge browsing* as observed by Mander et al. (1992). Also, non-zero depth is necessary for the bounding volumes used in the collision detection. It is important that objects are not too thin, as this leads to problems for the collision detection algorithms. For our prototype we use a depth of one-eighth of the length of an object, with a collision bounding geometry of an oriented box equivalent to the icons' display geometry. We have omitted textual labels on the icons as our initial focus is on the physics, pile and pen interaction issues. A benefit of unnamed documents is that users are freed from having to be committed to categorization as in current desktop interfaces that require the immediate naming and filing of documents. We note that solutions for



Figure 17. Earlier Prototype with top-down view.

dynamic labeling of arbitrary items in a 3D space and avoiding occlusion exist in the literature (Blaine, Steven et al. 2001).

It is important to observe that our techniques are scale-independent and work on any mixture of arbitrarily sized objects. This allows for interesting usage scenarios such as the organization of windows or photographs (Figure 18). This makes certain annotation techniques more relevant such as crumpling up or creasing a photo since they are clearly visible. However, our focus is on interaction with icons and not on how windowed applications would live in our interface. One can imagine a mixed-mode approach where icons and folders in our physical desktop launch standard windowed applications. An alternative would be to rethink how windowed applications could benefit from the physics paradigm, but we leave this issue for future work.

Physics-based movement of objects is simulated with rigid body dynamics, collision detection, and frictional forces. When objects collide they bump against and displace one another in a physically realistic fashion. A simulated gravitational force



Figure 18. Our techniques apply to arbitrarily sized objects. Here we see a pile of photos (bottom left) and casually arranged (top left) and crumpled up (top right) windows.

keeps objects on the ground. The introduction of physics simulation to a desktop environment makes the desktop more lively, and offers increased degrees-of-freedom for potentially more expressiveness than a traditional GUI desktop where icons are kept axis-aligned and have little resemblance to their physical counterparts. We believe that this physical simulation has a positive and subtle effect on object placement and appearance. For example, if a few documents are casually tossed to a corner they will collide and begin to accumulate. Their messy appearance subtly affords an unorganized state, without the user having to explicitly specify it.

The primary focus of the prototype is to enable casual organization of documents as one would on a real desk, using piling rather than explicit filing as the primary organizational style. We have developed a variety of novel interaction and visualization techniques for implicitly and explicitly creating, manipulating and organizing piles and items within the piles.

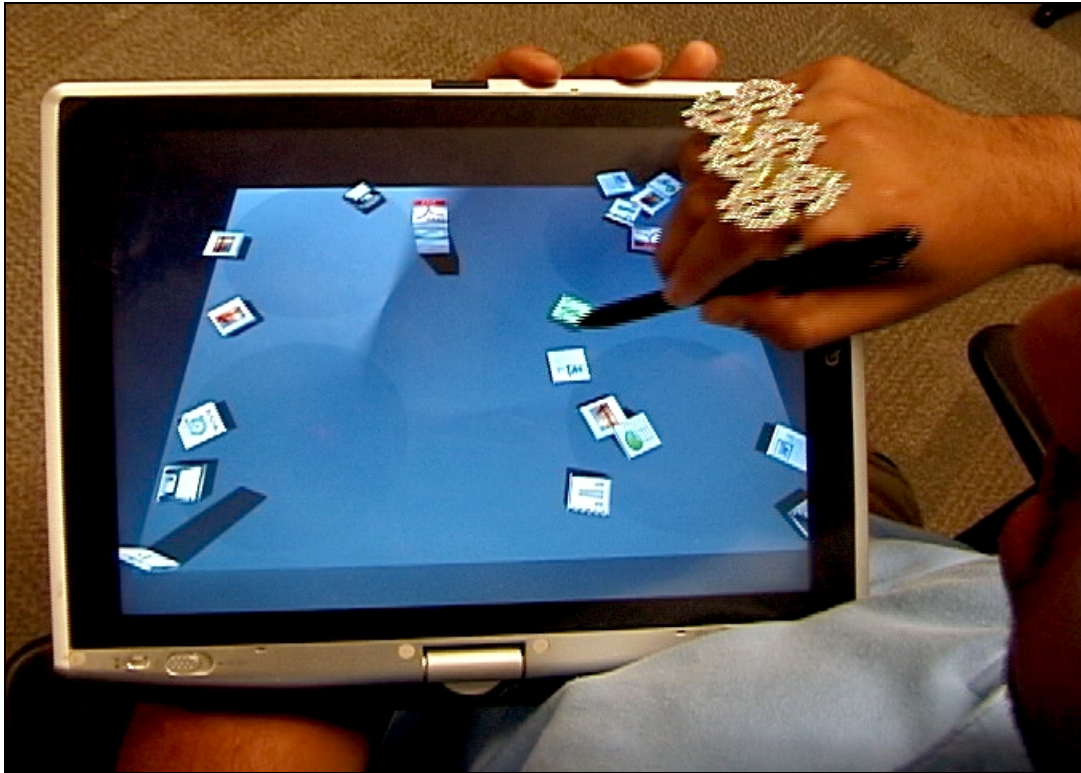


Figure 19. BumpTop on a commodity TabletPC. Manipulated items appear directly under the pen.

4.1 Interaction and Visualization Techniques

A pressure-sensitive pen with a single barrel button operating on a TabletPC is the sole input mechanism for our prototype. To facilitate very lightweight interaction for the simplest tasks, the pen by default allows users to move and or toss objects by touching and then dragging or flicking them with the pen in a manner similar to how one might use a finger to manipulate a bunch of lightweight items on a physical surface. More complex interactions, however, require additional techniques. We now discuss the design and development of the novel interaction techniques we have developed, as well as provide some critical analysis of their strengths and weaknesses.

4.1.1 LassoMenu

Motivation

Our interface required a technique for invoking commands on objects while accomplishing as many of our design goals as possible. The relevant design goals here are Optimize for Pen Interaction, Discoverable, Learnable Interface and when applicable Smooth Transitions. Ideally, we wanted something that used the single input channel of pen position, keeping interaction simple and leaving other channels available for more complex interaction. Also, since this was going to be a pervasive technique in our interface it needed to be simple and easy to learn, but powerful enough to accomplish generic actions on any selections.

There is a precedent for fluid selection and command invocation in a single stroke, found in systems pen-based interfaces such as GEdit (Kurtenbach and Buxton 1991) and Scriboli (Hinckley, Baudisch et al. 2005). However these previous approaches are limited. GEdit relies on shape-based gestures which cannot reliably be recognized and do not support discovery. In addition, they are not scalable. Each gesture can only be used for a single command, requiring a large gesture set for our usage scenario where many different kinds of interactions must to be supported. Further, gestures in GEdit do not allow specification of a parameter, required by our interface for specifying the degree of crumpling of an object, for instance. The recent pigtail and handle techniques in the state-of-the-art Scriboli do reduce a number of these issues by introducing marking menus (Kurtenbach 1993) for discoverable, scalable and generic command invocation. Though there are still drawbacks: the handle technique requires lifting the pen and returning it to the surface which breaks user flow. Also in their study, several users found the pigtail gesture difficult to create and the majority preferred the handle technique.

Initial Design and Refinements

We explored alternatives to existing techniques that potentially solved the problems of awkwardness and disrupted user flow. We initially investigated how users made selections in pen-based interfaces. Selections are commonly made with “lasso”

based encircling of desired objects, a technique which exploits the drawing capabilities of the pen. We noticed in off the shelf applications that users often naturally close the lasso, returning the pen to where the selection began. Despite that this is not required by selection algorithms which readily complete the lasso geometry. It then occurred, why not present a marking menu when the user reaches the end of the lasso, as users are likely to do anyway. So, in our first design we placed a target at the beginning of the lasso that would activate a marking menu when reached.

We quickly discovered several issues. First, the activation target showed up right away, which was problematic because it would get invoked right away. It is important not to display the invocation target until the user has left the initial region. Second, the target was initially too small and became a chore to hit. When we made the activation target bigger it became much easier to swing around and hit it with ample speed in a more natural motion, and without having to pre-calculate the stroke trajectory too much. Third, the lasso would occasionally become tedious to complete, especially when the selection area was very large and closing the lasso required covering significant screen real estate. To support “short circuiting” of the lasso selection we supported the pigtail gesture to invoke the marking menu, as in Scriboli. As our designs grew we recognized the need for 2D parameter control and evolved our marking menu to a control menu (Pook, Lecolinet et al. 2000). A control menu is a marking menu variant in which the user first selects a menu item via a simple stroke and then can choose to smoothly modify the length of the stroke to adjust the value of an associated 1D or 2D parameter. The visual design of the invocation target was improved by making the blue circle indicating the invocation target semi-transparent to reduce occlusion issues.

Final Design and Analysis

We call our novel design the LassoMenu. It combines selection, command invocation, and parameter specification in one fluid stroke (*Figure 20*). It supports the typical lasso style of drawing a path around desired objects to be selected. LassoMenu is powerful and expressive enough to trigger the majority of functionality in the system, and supports users becoming proficient in the system by learning just a single technique. LassoMenu avoids the tricky pigtail gesture and is arguably more fluid than the handle

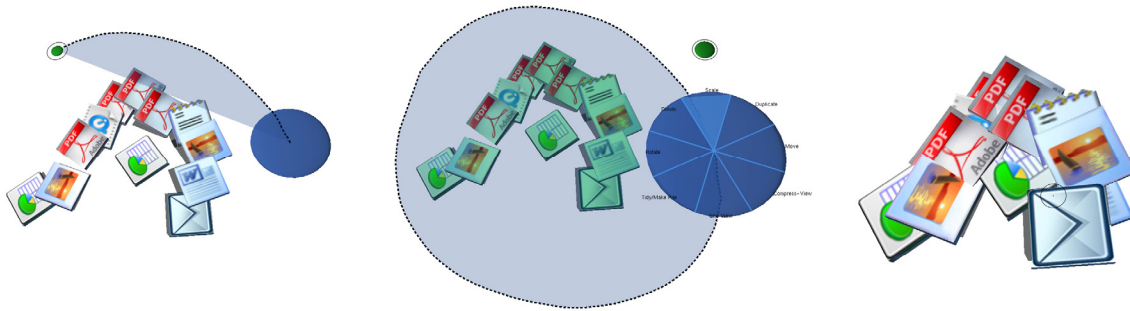


Figure 20. LassoMenu.

From left to right: Lasso selection phase; Lasso completed, pen reaches blue circle and enters control menu; Resize command selected; Remainder of pen movement adjusts resize parameter.

technique since the pen remains on the screen. Further, there is no gesture to memorize. The unobtrusive semi-transparent blue circle indicates additional functionality and the user is not penalized for simply exploring it as lifting the pen up before they leave the blue circle does nothing. The inclusion of a control menu enables the fluid transition from novice to expert functionality in that novice users can browse the menu visually to identify and select the desired items while experts who have performed the same selection numerous times in the past can simply make the stroke in the appropriate direction without visually attending to the menu itself.

Note that the LassoMenu keeps the pen down throughout interaction opening it up to concatenation with further drag interactions, for example. This feature also allows the LassoMenu to be used smoothly in combination with existing techniques like the handle and pigtail. This is demonstrated in our system with the inclusion of the pigtail technique together with the LassoMenu. However, there are some limitations of the technique. For instance, it doesn't work nicely on one item where lassoing is tedious. In addition, closing the lasso can be inconvenient or tricky if the simplest enclosing path would select unwanted items, though this is mitigated by the inclusion of pigtail. Also, persistent selections and additions/subtractions to the selection aren't supported "out of the box". However, one solution for this would be using the control menu options to facilitate this.

4.1.2 Object Movement

Design and Implementation

Objects on the desktop can be dragged around, and are attached to the pen position by a dampened spring. This is a popular method of interaction with physical simulations (Baraff, Witkin et al. 1997). Movement in the real-world is smooth, where velocities gradually rise and fall instead of the instantaneous movement found in typical GUI applications. By incorporating this spring model into our technique, it affords a subtle effect on the feel of the interaction, making it more lively and physically realistic.

Another benefit of the spring is that it allows a quick flick of an object to toss it across the screen. The item will naturally decelerate due to friction and will bump and displace objects in its path appropriately. The quicker the flick, the further and more forcefully the object will travel. Multiple objects are moved and tossed in a similar fashion. When a user lasso selects multiple documents, they are highlighted and invisible dampened springs are created between them with a complete graph topology. Selection springs allow the drag or toss of one document to tug along the other documents in the selection while maintaining their relative spatial positioning to each other. These springs are released when documents are deselected by clicking a vacant area of the desktop or starting a new selection. We note that while we used a complete graph topology for ease of implementation, it might be interesting to experiment with other topologies to reduce numerical instabilities.

The pen can also be used to ruffle through and nudge objects aside as if it had actual physical geometry in the workspace. This is accomplished by holding down the pen barrel button while moving it on or above the screen surface. We note that accidental triggering has been known to occur with use of the barrel button (Fitzmaurice, Khan et al. 2003; Miller 2005). Though, hardware manufacturers are mitigating this with new pen designs which move the barrel button higher up on the pen. Our objects behave as if they had certain physical properties. They are moveable, rigid, bouncy, and toss-able. We believe these properties enable a more physically realistic environment and afford users to organize their virtual objects in more expressive ways, as per our *Tangible, Paper-like Icons* design goal.

Analysis

While collision detection and response to object movement provides the benefit of a lively and realistic workspace, casually dragging a document from one end to another on an organized desktop can disrupt the layout. This is due to the dragged object bumping other objects in its path, which depending on the object arrangement might be difficult to avoid. This is not a problem on today's desktop and in line with our *Disable Physics as Necessary* design goal it may be beneficial to explore the user controlled disengaging of physics to prevent unwanted disruption to layout in a revised system. In addition, physics simulation algorithms deteriorate when the number of objects gets very large. Though we never experienced these problems during normal usage of our prototype, we stress tested the system by deliberately overloading the desktop with documents and allowing them to pile up in one place. Here the jitter and numerical instability was noticeable. However, since it never proved to a problem in our usage, the simulation engine's parameters were not tweaked to minimize this. Further, while the simulation has limits it is possible to increase them by disabling physics on certain objects. We adopt a similar technique to minimize instability caused by surfaces laid flat on top of each other in piles described below in the Polite Physics section.

4.1.3 Pile Creation

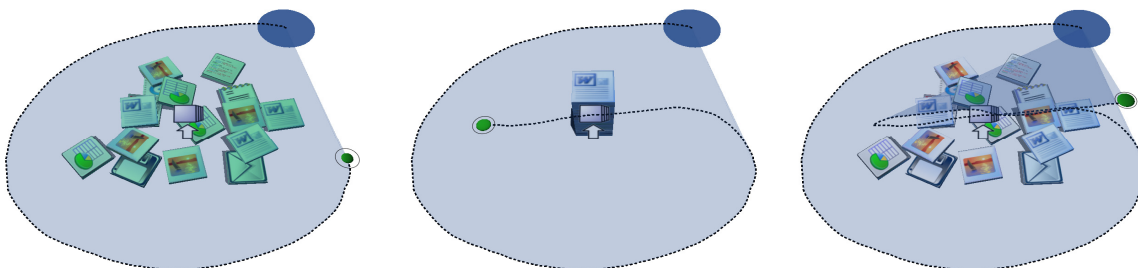


Figure 21. Lasso 'n' Cross technique for pile creation.

From left to right: Documents lasso selected; 'Create Pile' icon crossed and pile created; Undo by "un-crossing".

Lasso 'n' Cross Technique: Design

To more explicitly organize a group of objects we can create a pile out of them. Pile creation is a fundamental technique in our design and we desired a quick, easy and

elegant way to create a pile. In our design process we examined what this interaction was about and tried to reflect that in the resulting technique. Pile creation is about bundling together a group of objects to a specific location. On a real desk this usually means physically bringing together a bunch of contiguous objects, starting with the hands at the outer bounding region (specifying the scope) and then shuffling inwards to a single location. In our interface, piles are created by lassoing around a group of objects, then crossing the ‘create pile’ icon that appears at their centroid. We call this novel technique *Lasso’n’Cross* (*Figure 21*). This technique allows users to fluidly select and pile objects in one stroke and harnesses a pen friendly self-revealing semi-transparent crossing target. The interaction can be thought of as being phrased in two parts as the real world physical activity is: defining the scope and then specifying what to do with it. Since the pile is placed where the icon is displayed, stroking there draws user attention and the pen itself to where the pile will be created for easy further interaction.

Discovery is also supported as per our *Discoverable, Learnable Interface* design goal. Novice users will typically wait until they notice the icon before completing the stroke, but as they practice making the stroke over successive invocations, they transition seamlessly to expert behavior where the stroke is made without waiting for the icon to appear. Lasso’n’Cross also supports undo, allowing users to undo and redo the pile creation by consecutively re-crossing the icon. Undoing can be thought of as “uncrossing the initial cross” since the stroke is undone by making it backwards.

Lasso’n’Cross: Refinements

Our initial Lasso’n’Cross design placed an unlabelled, opaque circular activation target at the centroid of the points in the lasso. However, this presented three problems: it wasn’t clear what the unlabelled circle did, its solid opacity occluded objects below it, and basing position on the centroid of the constantly changing lasso points rapidly moved the icon making it hard to hit, sometimes requiring “chasing” it when approaching it with the pen! In addition, approaching the centroid with the lasso stroke was troublesome because it modified the selection region with objects intended for selection becoming de-selected. This was corrected by using the convex hull of the lasso stroke to indicate the selection (shown in dark blue in *Figure 21*). The convex hull avoided unwanted changes

to the selection created by attempting to cross the icon. Further, to prevent accidental crossing of the ‘create pile’ icon, the icon is only visible when the centroid is not likely to fall near the users lasso stroke. From our experience, this is typically when the stroke is not a straight line. We use a heuristic from the literature (Thorne, Burke et al. 2004) to determine if a stream of user input points is a straight line:

$$\text{Lasso arc length} / \text{Distance between lasso endpoints} > 1.2$$

Lasso’n’Cross: Analysis

Lasso’n’Cross is an improvement over similar pen-based gestures combining selection and a single action such as the delete gesture in GEdit (see Figure 4 top right, Kurtenbach and Buxton 1991), triggered if the end of the stroke is inside the closed lasso. It is advantageous because it supports undo and eases the requirement of remembering a gesture by facilitating discovery amongst novices, without being excessively obtrusive. In addition, unlike many other shape-based gestures it doesn’t need to be drawn the same way every time. It is dynamic and supports a multitude of invocation possibilities. Once comfortable with the technique it can be used very quickly and its “loopy” nature plays well to the scribbling capabilities of the pen. One limitation of Lasso’n’Cross is the inability to select items described by a concave lasso stroke. Though this hasn’t been a significant problem during our usage, it may vary in a more practical setting. One solution would be improving the heuristic. Also, in its current incarnation, the Lasso’n’Cross technique only supports the invocation of one interaction technique, activated on cross and reversed on un-cross. Future designs might experiment with invoking a marking menu once the activation target is crossed. Another interesting possibility is supporting repeated actions with crossing in the same direction, looping on a single crossing target as in the CrossY interface (Apitz and Guimbretière 2004).

Tidying, Messy Piles and Tidy Piles

When creating a pile with Lasso’n’Cross the selected object’s orientations are tidied, vertically sorted according to their original heights and stacked into a *Tidy pile* (Figure 22d). The resulting pile replaces the ‘create pile’ icon at the centroid of the selected objects. This is smoothly animated to avoid confusing the user with an

instantaneous new representation of the objects, as per our *Smooth Transitions* design goal.

Alternatively, using a single LassoMenu option on unpiled objects, the user can choose to tidy them, create a messy pile, or create a tidy pile out of them. The option is determined by the distance between the end points of the stroke drawn after “Tidy/Make Pile” has been selected in the LassoMenu. The option selected in order of shortest to longest stroke is as follows: (1) tidy documents by tightening up their poses but do not create a pile, (2) create a *Messy pile*, (3) create a *Tidy pile*, described above (*Figure 22*). These options are ranges on a continuum of stroke distances and selecting in between these ranges specifies the degree of the particular option. The document poses are updated live and the user can “scrub” to create the desired arrangement. In the initial design it was confusing as to what state the pile would be in if the pen was released. To remedy this visual feedback during the scrub is provided by icons that appear at the range edges. That is, the points during the scrub at which a messy pile or tidy pile will be created. Being able to create piles quickly was supported with the Lasso’n’Cross technique, although using the LassoMenu as an alternative allows us precise specification of a pile creation parameter. In our design we mapped this to pile tidiness, however there are other possibilities such as controlling pile location in the workspace.

A *Messy pile* integrates some of the objects’ messy pose information by interpolating between the messy and tidy arrangements of a pile. Our Messy pile improves upon the ‘disheveled pile’ in previous pile work (Mander, Salomon et al. 1992). Instead of arbitrarily displacing items in the pile to achieve a messy appearance, we incorporate meaningful spatial information from the unpiled state. The magnitude of this information is user controlled and can be reduced or emphasized by tidying or “messing up” the poses respectively.

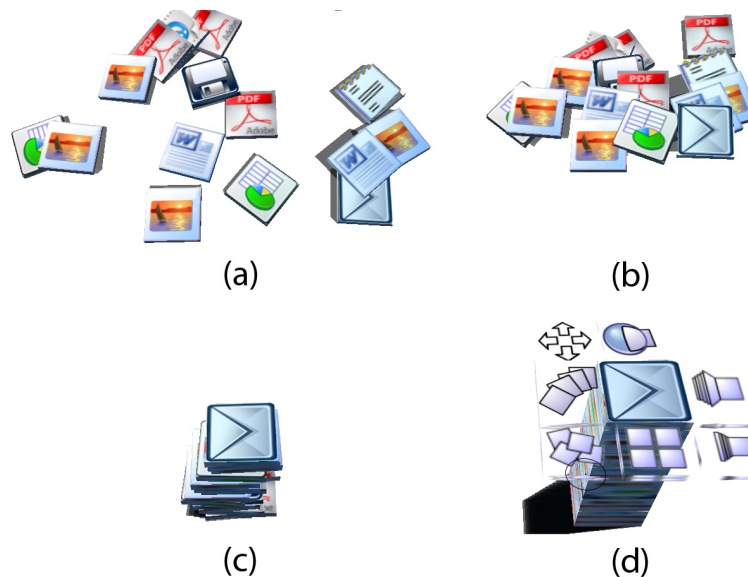


Figure 22. Transitioning from implicit to explicit piles.
 (a) Casually laid out documents. (b) Tidied unpiled documents. (c) Documents made into a Messy Pile.
 (d) A Tidy Pile with widgets revealed. Widgets in clockwise order from center-top: Fisheye, Leafer, Compression-Browse, Grid, Messy/Tidy, Fan out, Move.

4.1.4 Pile Manipulation

Pile Browsing with Widgets: Motivation and Refinements

We needed a pen-centric way to browse a pile's contents. Initially we used the LassoMenu but found lassoing a pile was tedious and too slow. We then experimented with placing a marking menu that was invoked when clicking on a pile. Though, this did not support clicking nor was it clear that a marking menu would appear when a pile was clicked upon. Our final design used self-revealing pile widgets that could be rapidly clicked or crossed without waiting for the widgets to appear, behaving and leveraging the well known benefits of marking menus (Kurtenbach 1993; Kurtenbach and Buxton 1993; Kurtenbach and Buxton 1994; Tapia and Kurtenbach 1995). We also explored a variety of pile browsing techniques. Our goal was not to provide a definitive browsing solution for pile contents, but rather explore how existing visualization techniques developed for other domains could be applied to piles. In several cases, our techniques are designed to explicitly support real-world pile browsing behaviour observed in office workers by

Mander et al. (1992). We look to the user study to see if any clear preferences arise and whether techniques should be trimmed out.

Pile Widget Design

When the pen hovers over a pile, pile widgets (*Figure 22d*) are revealed allowing the user to trigger various browsing techniques of the pile's contents (*Figure 23*). We draw upon previous work for browsing data in our different pile layouts (Carpendale, Cowperthwaite et al. 1999; Carpendale and Montagnese 2001; McGuffin, Tancau et al. 2003). The *Fan-Out* widget (*Figure 23f*) spreads pile items like a deck of cards on the user-drawn path, allowing pile contents to be viewed in parallel. This supports the spread-out behaviour observed by Mander et al. (1992). Leafing through pile contents much like one flips through pages of a book is accomplished by scrubbing the *Leafer* widget (*Figure 23b*), supporting the restack and to some extent the hinge browsing behaviour. Browsing techniques without a physical analog include the Compression-Browse widget (*Figure 23c*) which compresses items on one axis to reveal the items underneath, without moving items. The standard grid layout is also offered (*Figure 23e*). Larger piles benefit from a fisheye view (*Figure 23a*), implemented via the Elastic Presentation Framework library (Carpendale and Montagnese 2001). The *Messy/Tidy* widget (*Figure 23d*) can be viewed as the inverse of the "Tidy/Make Pile" pile creation functionality described earlier. Seeing how piled objects were originally strewn about the desktop may aid recall of pile purpose or content. Scrubbing this widget interpolates between the messy and tidy poses and at the extreme messy pose an icon appears indicating the pile will be broken. All transitions between browsing styles are animated smoothly as per our *Smooth Transitions* design goal.

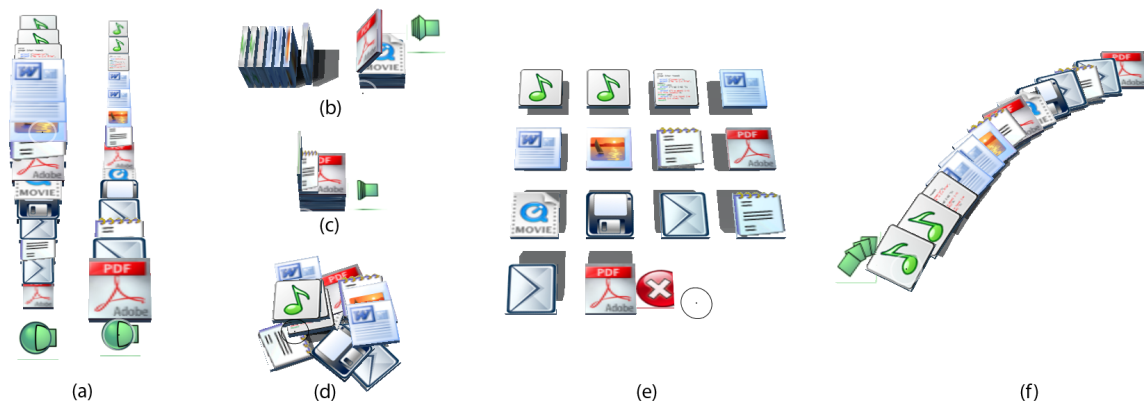


Figure 23. Pile browsing layouts triggered by widgets.

(a) Fisheye. (b) Leafing through like pages of a book. (c) Compression-Browsing higher items to view items below. (d) Interpolating between Messy and Tidy positions. (e) Grid browse, locked down for further manipulation with *PressureLock*. (f) Fan out on user drawn path.

A user clicks and drags on a widget to immediately see its impact on the layout of the pile contents. Once the pen is released the objects smoothly return to their piled state, facilitating quick, transient browsing. For more involved interactions a pile can be locked down into any of the browsing states. This is done with the *PressureLock* technique described later. Once a pile layout is locked, the widget turns into a red X (Figure 23e) and can be collapsed back to its original state with a tap. We do not need to explicitly pile objects before we can apply our browsing tools. For example, it may be useful to temporarily view casually strewn objects in a grid layout to see occluded objects. Compression-browse would similarly reveal occluded items without disturbing item positioning and marinating context. Such casual browsing of unpiled objects is triggered via the LassoMenu.

Pile Widget Analysis

The visual design of the pile widgets was iteratively refined. Initially they were boxes and looked quite like other icons in the interface, not always accurately visually describing the type of browsing they invoked. Widgets were redesigned in monochromatic colors with the assistance of a graphic designer. In addition, hovering over a widget for some time presents a tooltip with a more detailed textual description of the widget's functionality, to eliminate ambiguity about widget purpose. Widgets also act as crossing targets for our novel *Drag'n'Cross* technique for precise insertion of

objects into a pile, described later. The tradeoff of supporting crossable widgets is that piles cannot be dragged to be moved, since any drag is interpreted as a marking menu stroke. Therefore, pile movement is supported with one of the widgets. It may turn out that in practical use tapping on a widget is easy enough for users and so crossing takes away from the interface here. The user study should shed some light on this.

Enhancing Location-Based Searching

To find an item on traditional GUI desktops there are essentially two options, logical and location-based searches, defined by Barreau and Nardi (1995). Logical searches use text-based keyword searches, however this is slow and only works if you happen to know the name of your file. This is confirmed by Barreau and Nardi's study and follow up (Nardi and Barreau 1997) which found users are hesitant to employ text-based search tools because problems remembering filenames. In a more recent study Ravasio et al. (2004) found that expert-users occasionally resorted to logical search only as a last-ditch effort, when the files they were looking for were considered lost. Location-based search on the other hand, involves the user iteratively guessing and inspecting folders in which they think the file might reside. Further, in Barreau and Nardi's studies users overwhelmingly preferred location-based searches.

Since our initial implementation only supports casual and unlabelled documents logical search is not feasible. However, if we want to find a piled object but do not remember which pile it is in, we can use the browsing widgets to attempt a location-based search. Though, for large numbers of piles clicking widgets becomes tedious. For this situation we developed the *Exploding Piles* functionality, offering a way of visually searching pile contents regionally. Once Exploding Piles is globally invoked with the LassoMenu, piles are smoothly exploded into a grid view of their contents on hover. Moving the pen away collapses piles back to their original state. This technique is similar to Sonnet et al.'s (2004) work using an interactive explosion probe to browse a 3D model while maintaining context. Exploding Piles exploits the rough spatial memory a user may have about what they're looking for. For example, if it is known an item is in one of the piles in the top-right of your workspace you can inspect them by pointing at them. Using spatial memory narrowed down to a specific region is not possible with a text-based

search query. In its current incarnation, the technique is limited by requiring a separate mode. One solution to this would be using a Lens that is dragged to piles to explode them. The magnitude of benefit, if any, of the exploding piles technique over simply using the widgets, needs to be determined through a user study of performance.

Pressure Cursor and Pressure Lock Technique

We resisted using pen pressure in our interface for some time, to deliberately explore the limit of possibilities with just 2D pen position. Ultimately, we ran into input limitations when some fluid techniques already had the pen touching the screen surface in continuation of a previous interaction, but we desired to concatenate further interaction. Therefore, we turned to careful use of pen pressure. Pressure presented a unique design opportunity, since it could be used to evoke physical versions of the actions in our interface such as pressing down to “hold” a pile in place or pin up an object to the wall. When users push very hard with the pen and reach the maximum pressure level, it acts as a trigger in our interface dubbed *PressureLock*. *PressureLock* is used in our interface to lock a pile down into a specific browsing layout or pinning objects to the wall.

To provide continuous visual feedback for the *PressureLock* technique we use a circular pressure cursor (Ramos, Boulos et al. 2004) with an inner circle that increases in size with the current pressure level (*Figure 24*). When the pressure level reaches its maximum, the color intensifies and the outline turns into a bright white to indicate a *PressureLock* has occurred. When a *PressureLock* is possible it is indicated by the outer ring turning a hollow white, enabling discovery amongst novice users and in line with our design goal of a *Discoverable, Learnable Interface*.

When *PressureLock* is used for locking down a pile browsing layout, it is similar

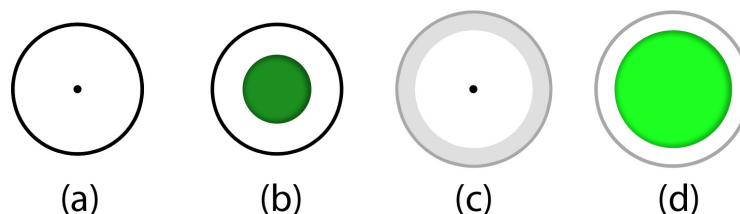


Figure 24. Pressure cursor.

(a) Normal pressure cursor with no pressure. (b) with 75% of maximum pressure. (c) Pen is in a position where PressureLock will trigger additional functionality. (d) PressureLock with 100% pressure.

to the Glimpse idea (Forlines and Shen 2005) of providing a pressure-based confirmation of a preview, with the pen-up before maximum pressure is reached being equivalent to an undo.

Adding to a Pile: Design

We explore a few techniques for precisely or casually adding object(s) to a pile. To reinforce the pile metaphor, it is important to maintain the ease with which one can add objects to the top of a pile in the real world, where one simply drops objects in the general area of a pile and they are “added” to the top. For casual and quick addition to the top of a pile we support tossing an object towards a pile. This is implemented by a threshold distance for piles that when approached by object(s) above a certain velocity inside that distance, they are smoothly added to the top of that pile. Alternatively, objects can be dragged on top of a pile that will highlight indicating that they will be added to the top on pen up, a technique seen in previous research (Mander, Salomon et al. 1992).

If the user drags an object to a pile and dwells, the pile is temporarily pushed apart allowing for precise insertion of that object into any location within the pile (Figure 25). Scrubbing the pen along the side of the pile varies the insertion point. We call this technique Dwell’n’Scrub and its visualization draws inspiration from the “side shift” pile browsing method in Mander et al.’s (1992) work. To avoid dwell which interrupts user flow, in respect of our *Optimize for Pen Interaction* design goal, we developed a more pen-centric interaction technique called *Drag’n’Cross* (Figure 26). While dragging objects, users can cross through a pile widget to use one of the browsing techniques for specific insertion. For example, if you drag an object and cross the Leafer widget, the object will be inserted at the point that you had leafed to before lifting the pen. After precise insertion, added objects slightly stick out of the pile in the direction they were added from. This indicates the recent insertion and reminds users that further organization may be necessary. To tidy the pile again, the user can simply adjust the Messy widget.

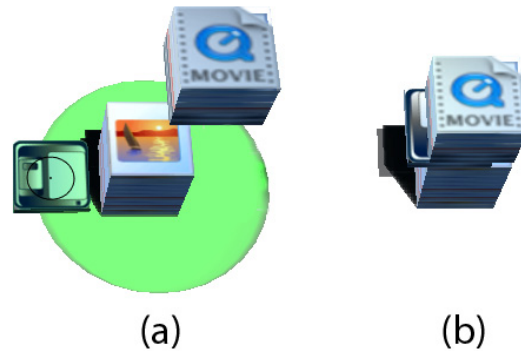


Figure 25. Dwell'n'Scrub technique for precise insertion.
 Drag and drop insertion of object into pile. (a) Drag to pile. (b) After insertion

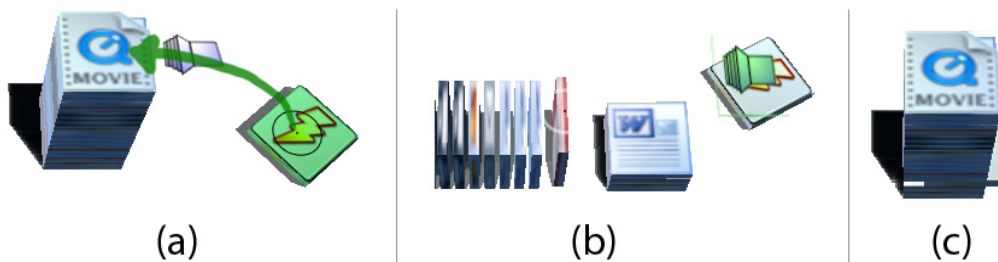


Figure 26. Drag'n'Cross technique for precise insertion.
 (a) User drags document and crosses Leafer widget, pen path shown by green arrow.
 (b) Scrub to specify insertion point. (c) Pen is released and document inserted

Adding to a Pile: Analysis

The development of several methods of insertion allows us to fill the interaction space with techniques supporting different styles of interaction, working with the tradeoffs of each technique. For instance, tossing is imprecise but quick and casual. Dwell and scrub requires more effort but is familiar to drag and drop techniques found in today's GUI desktops, though requiring user dwelling. Drag'n'Cross on the other hand is a novel technique without a common analog in today's GUIs and may prove to be too unfamiliar in the user study. However Drag'n'Cross allows fluid insertion to a specific point in the pile with a single stroke, without any dwell. In addition, it uses existing pile browsing widgets with the same visualization in another context reducing the new information that must be assimilated in order to understand the technique. With all of the insertion techniques there is the drawback that no undo is available once the action is initiated. This could be supported by including an undo option on a global control menu for instance, displayed when the LassoMenu is invoked on no selection.

Hierarchical Piles

Though we explore piling in our interface, we acknowledge the place and value of hierarchically filed information. While we could also support files in our interface, filing is already well explored in today's GUI desktops. Instead, we look to leverage the strengths of both piling and filing as we experiment with blending elements of the two paper processing strategies. In our hybrid technique, users can merge any combination of piles and objects into a new mixed mode hierarchical pile. Hierarchical piles are created using the same techniques employed to create a 'flat' pile out of unpiled objects: LassoMenu or Lasso'n'Cross. This also further generalizes the pile concept in our interface since any collection of objects, piled or unpiled, can indiscriminately be collected into a pile. The new aggregate pile stores all information regarding the sub-piles and includes items sticking out in sub-piles. If the aggregate pile is broken, sub-piles are restored in their original positions with changes, like item deletions, folded through.

A potential usage of hierarchical piles is to reduce the space occupied by any collection of items, and abstract away the visual complexity into a contained unit. Unlike storing things into a folder in traditional GUI desktops, the piling operation is reversible and goes constituent items go back to their original positions/piles when destroyed. This allows piles to be used as temporary, reversible collections, a benefit over real world piles. In a hierarchical pile users can drill down from one pile, through its sub piles until they find what they want. Though it is not implemented, when a pile is browsed, its subpiles could be clearly visible as piles with widgets and further locked down. Locking down a subpile could create the necessary layout space and perhaps clearly indicate subpiles with semi-transparent shading for each pile level. Presenting all subpiled information at once in this fashion could leverage the techniques in DynaPad which presents a similar representation (Bauer, Fastrez et al. 2004).

Manipulation of Pile Contents

While a pile is locked down into a browsing mode via the PressureLock technique we can further manipulate the pile contents with the LassoMenu. While adding physics to the desktop provides a nice realistic feel, we are not constrained to physically realistic interactions as per our *Realistic Feel* design goal. For example we can instantly sort piles

or sub-selections by type or size. Deletion and duplication is also possible. To re-arrange pile order we simply drag objects(s) to their new location within a locked down pile.

On real desks we use subtle techniques to convey information about objects in piles such as re-positioning or re-orienting certain items so they stick out (*Figure 27* top, bottom-left). We support similar functionality in BumpTop. Objects in a locked down pile can be re-oriented by the LassoMenu. Alternatively, groups of items can be re-positioned so they stick out once they are dragged outside the pile confines (*Figure 27* bottom right). Once re-positioning is initiated the pile smoothly collapses back to its piled state so it can be seen in context. Dragging moves objects parallel to the plane the pile sits in, to avoid changing pile order. If the objects are dragged so far that they no longer overlap any part of the pile, they are pulled out of the pile and become the active dragged selection. Note that the pen is still down and we may fluidly proceed with other dragging interactions from here such as insertion into a pile via Drag'n'Cross or pinning up to a wall. Dragging items out of piles could also be used to split a pile if it becomes too large and cumbersome.

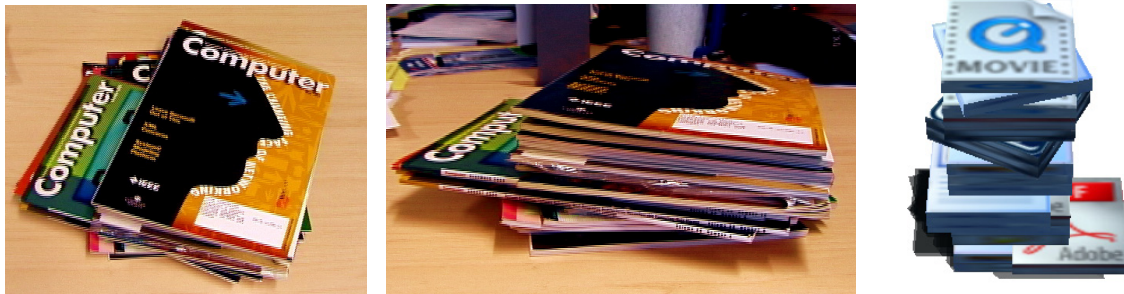


Figure 27. Real and virtual piles.

A typical pile found on real desktop with items rotated and pulled out for emphasis, top-view and side-view (left, center). Such piles are supported in our prototype (right).

4.1.5 Enhancing and Judiciously Using Realism

Whittaker and Hirschberg (2001) observed that amongst some office workers, frequently accessed items moved to the top of their piles, leaving less relevant material at the bottom due to repeated re-referencing. We aim to facilitate this behaviour by supporting easy removal and casual addition to the top of piles via tossing.

Giving Icons Affordances of Paper

Sellen and Harper (2003) identified that the physical properties of paper include being thin, light, porous, opaque and flexible. Further they observed that these properties afford different human actions including manipulating and folding. We note that sometimes people crease a corner of a document to indicate importance, or pin up documents to a cubicle wall to enhance its visibility and ability to remind. Objects in BumpTop support several manipulations that physical paper does, in line with our design goal of *Tangible, Paper-like Icons*.

We support freeform creasing of a corner or folding of a document with the robust paper peeling algorithms developed in Beaudoin-Lafon's (2001) work (Figure 18). To escalate a document to even greater importance we can pin it up to a wall by dragging it to vacant wall space (Figure 16) and using PressureLock to create a springy joint. To remove it we simply pull the document off the wall. This also makes use of the rarely used screen space occupied by the walls, necessary for our perspective view. Another emphasizing technique we explore is locking the rotation of objects to make them standup on end (Figure 28), despite collisions. This is accomplished with a LassoMenu triggered rotation and PressureLock once the desired rotation is specified. Further, and in combination with all of the previous emphasisation techniques, objects may be resized. Object density remains constant so bigger icons have more mass while reducing size reduces mass. Bigger documents are not only more visually noticeable but behave as though they are more important, displacing smaller items when moved and being difficult to move when bumped by smaller, lighter items. These techniques implement suggestions by Malone (1983) for increasing the size of an icon or putting it in a special screen location to increase its priority and ability to remind. Alternately, to de-emphasize items we can crumple them up (Figure 18). This provides an in-between state for items whose utility is questionable but are not quite ready for deletion. Crumpling is done from the LassoMenu which specifies a 2D parameter for distorting the object mesh. For example, a long quick stroke results in a tightly crumpled document.

Polite Physics

There is a danger of taking any metaphor too literally, as we discuss in detail under the *Disable Physics as Necessary* design goal in chapter 3. In addition, the

physical simulation can sometimes prove disruptive or counter productive to the interface. Therefore, in line with our design goal of *Disable Physics as Necessary*, there are instances where we disable it. For example, when items are in a messy or tidy pile and if physics are enabled, the collision detection of perfectly touching items causes numerical instability, jittering and unnecessary movement by piled items. Also, dragged icons knock over piles. Further, un-restricted physics allows six degrees of freedom in the potential positions of objects. This added freedom and expressiveness also affords more ways to “make a mess” versus a standard 2D grid desktop. Therefore, we have experimented with a mode where all objects remain axis-aligned. Collisions are no longer physically accurate but objects are easily readable remaining properly oriented, more closely resembling modern GUI desktops. In this mode the desktop seems aesthetically tidier but looks and feels mechanical (*Figure 28*). It may also be interesting to experiment with allowing a small amount of off-axis rotation to keep objects easily readable, as in Beaudoin-Lafon’s (2001) work.

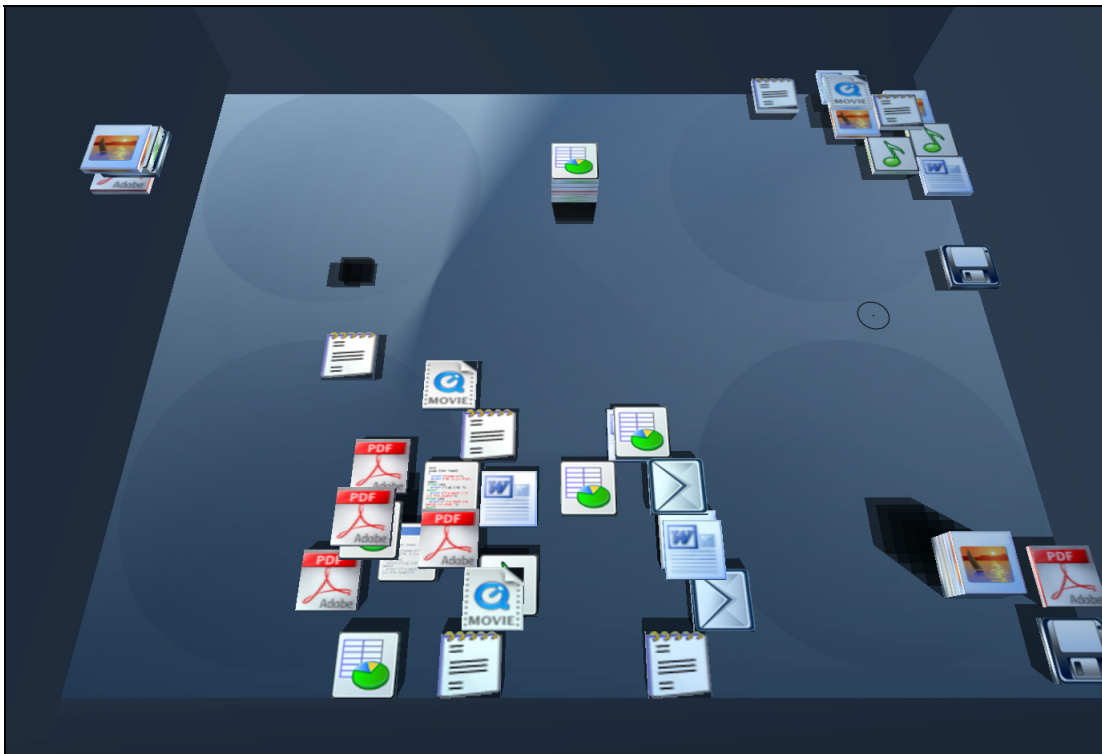


Figure 28. Axis alignment enabled to enforce a tidier appearance. A side effect of locking rotation: the ‘shelf’ on the left-hand side of the screen was created by pinning up rotation locked items to the wall.

4.2 Implementation

BumpTop was developed on a Toshiba Portége M200 TabletPC with a 1.6 Ghz Intel processor, 1 GB RAM and a GeForce FX Go 5200 graphics card. The BumpTop prototype is written with C++, OpenGL and GLUT. Rigid Body Dynamics and collision detection are provided by the NovodeX Physics SDK (www.novodex.com) which is a high-performance physics engine popular amongst game developers and is used to power the marquee first-person shooter Unreal Tournament. Some desktop item icons were designed by David Vignoni from the Nuvola icon theme.

4.2.1 Performance Concerns

While our prototype runs in realtime on commodity hardware, one may consider it to be a waste of CPU resources to have a physics simulation engine constantly running. However, it is reasonable to assume that future CPU speeds will be sufficiently high that the additional cost of running such a simulation will be negligible. Further, slated for release in 2006 is a dedicated Physics Processing Unit (PPU) that would provide hardware acceleration for all the physics calculations, freeing the CPU for other tasks. The PhysX PPU is to be released by Ageia (www.ageia.com), the maker of the NovodeX Physics SDK we use. As such, our prototype could readily be ported to receive hardware acceleration.

5 User Evaluation

*"Life can only be understood backward,
but it must be lived forward."*

-Søren Kierkegaard

To evaluate our prototype we ran an observational user study with six participants (two female, four male) with computer skills ranging from novice to pen-interface experts. Users participated in one-hour think-aloud sessions on a TabletPC that were videotaped with two cameras: one capturing the screen and the other an overall shot of the workspace, including the facial expressions of the participant. Subtle refinements to the protocol were made as the experiment progressed and is as follows (also see Appendix

A: Experiment Protocol and Questionnaire). First participants were introduced to basic functionality with a three minute introduction to the following techniques: moving items, the LassoMenu, using the LassoMenu to create a pile, using widgets to browse a pile and Pressure Lock for locking down a pile. The introduction ended with a pile on the screen that had just been closed from a pressure lock so it had the users focus. Following the introduction there was a ten minute “discovery” period where participants simply explored the system. After this period participants were instructed on the remaining functionality that they hadn’t discovered. For the next approximately fifty minutes users

completed twenty-nine tasks that could be completed using multiple strategies. Tasks were designed to simulate the usage of the prototype in a realistic setting and walk a subset of the functionality of the system, requiring the more important techniques several times. The functionality not evaluated includes our more experiment techniques, particularly organization of photos and windows, exploding piles technique and axis-alignment technique. The order of tasks was occasionally changed to accommodate the flow of conversation and slight adjustments to the protocol were made when technical difficulties were encountered. Finally, a retrospective interview and written questionnaire on their experience was completed.

Results are drawn from the analysis of the interviews, questionnaires and detailed notes made during the experiment. Think-aloud experiments were first described by Karl Duncker (1945) and are a traditional psychological research method (Ericsson and Simon 1984). They are a very popular method for usability evaluation of computer interfaces in industry. Clemmensen (2002) found think-aloud was the single most frequently used technique amongst HCI practitioners in Denmark. Nielsen (1993) notes that this type of method is effective for achieving “a wealth of qualitative data [collected] from a fairly small number of users. Also, the users’ comments often contain vivid and explicit quotes...”. Analysis of video was not undertaken as there was ample information in the raw data from the study, though synchronized recordings of the two video streams are available for further analysis as appropriate.

The results are encouraging. Participants were able to discover functionality on their own and became comfortable and proficient accomplishing most tasks. Subjects successfully completed 88% of tasks (standard deviation: 0.07), disregarding ones where technical difficulties were encountered. It is encouraging to see with just a few minutes of introduction to the system participants were able to become proficient enough to complete a large majority of tasks.

Questionnaire responses confirmed that: techniques were easy to learn (4.67/5), users liked the software (4.67/5), and techniques felt familiar (4.50/5). A novice participant who has difficulty operating a standard computer commented “for me who doesn’t know how to use a computer, this interface is good... can it replace my current one?” Techniques like tossing were found empowering as they allowed leveraging of

real-world knowledge. Many participants found the interface playful, fun and satisfying. Comments included “its got a good feel to it”, “its fun actually”, “it’s a cool interface”, “its very smart” and “can I have one of these now?”

The evaluation also revealed some issues with our interface. In the remainder of this chapter we discuss in detail these issues as they relate to our novel interaction techniques, pen-based and physics-based interfaces. We begin with a reflection upon our initial design goals drawing upon data from the user study.

5.1 Findings

Questionnaire Item	Mean (Std. Dev.)
I [Liked/Disliked] the software.	4.7 (0.6)
The software is [Efficient/Inefficient].	3.8 (1.1)
The software is [Easy/Difficult] to use.	4.4 (0.6)
The interaction techniques are [Easy/Difficult] to learn.	4.7 (0.6)
The software feels [Familiar/Unfamiliar].	4.5 (0.9)
The software [Matched/Did not Match] the way I expect objects to behave in the real world.	3.2 (1.4)
It is [Easy/Difficult] to find the items I am looking for	3.5 (0.9)
I feel [In Control/Not In Control] of the software	4.2 (0.8)
I am [Able/Unable] to accomplish what I am trying to do	4.4 (0.6)
Organizing icons is [Easy/Difficult] with the software	3.7 (1.6)
Organizing icons with this software [Is/Is Not] akin to organizing them on my desk	3.2 (1.8)

*Table 1. Subjective questionnaire responses.
5=Positive, 1=Negative. Higher scores are more favorable to our prototype.*

5.1.1 Reflection on Design Goals

Initially we established some design goals based on our own experience as well as previous usability research. We now reflect on how our designs met some of these goals informed by data from the user evaluation.

Discoverable, Learnable Interface

Users were able to complete approximately 88% of tasks without extensive training. Participants used the ten minute “discovery” portion of the experiment in different ways. Some experimented with the movement of objects and spatial layouts.

All experimented with the pile widgets to invoke the various browsing methods and discovered interaction techniques not explicitly instructed to them in the introduction. This indicates that the interface did indeed achieve the *Discoverable, Learnable Interface* design goal. Users explored the techniques offered by the LassoMenu, which indicates that the discoverability of Marking Menus is also present in the LassoMenu. Our aim to not penalize users for exploring the system was also achieved. When participants were unsure which LassoMenu command was required for a particular task they routinely tried out several different options, releasing the pen to undo the action as appropriate. One participant commented that the responsiveness of the system encouraged him to explore it further.

In addition, all participants felt they got better with the techniques as time went on. Some emphatically stated that if they were given the tasks again they could effortlessly complete them. Further, when technical issues required tasks to be attempted again, they were often done with much greater ease and proficiency.

Realistic Feel

During the discovery period and idle time between tasks users could be found playfully tossing or re-arranging items or eagerly watching the results of collisions. One participant commented “I can make as much of a mess as I want without my mom getting mad at me... It’s like being in a sandbox all over again”. This playfulness also translated to users to become very proficient at arranging items with subtle and precise movements. For example, within minutes a participant was delicately balancing a document on top of another document pinned up to the wall. One participant carefully arranged icons on edge and toppled them over like dominoes. This behaviour suggests successful leveraging of real world knowledge of movement, the aim of our *Realistic Feel* design goal. Further users felt the software felt familiar (4.5/5) in the post study questionnaire and one commented how the interface felt “like a room, but less dusty”.

Sometimes users had “expectations of physicality” which were not always supported by the interface. For example, when a pile was locked down in a fanout layout most users expected to be able to drag unstacked items into or out from it. However, this functionality was not implemented in this way due to the inability to drag crossing targets

mentioned in the *Crossing-based Interaction* section below. The affordance of wanting to pull out pile items sticking out from a pile is also very strong and Mander et al's (1992) pile work also encountered users often trying to drag items out, though it wasn't supported. Conversely, the physics engine enabled dynamic behaviour and since it provided a complete set of rules to respond to varied user input the interface supported functionality that wasn't always explicitly designed. For instance one participant successfully tried to toss a document onto a pile by banking it off the wall. One user commented that he preferred techniques which "maintained the illusion of reality in the interface". This feeling may be linked too other users preferences for browsing techniques with physical analog's (Leafer and Fanout) discussed in detail below under the heading *Pile Browsing Techniques*.

Smooth Transitions

Two users explicitly praised the smoothness of the interface. In addition, there was no noticeable confusion or bewilderment when piles were being browsed, even though browsing quickly and drastically changed pile representation. In fact, five out of six users said some sort of pile browsing was amongst their favorite techniques in the entire system. More discussion on pile browsing results is below in the section *Pile Browsing Techniques*. There were however two instances where smooth transitions caused issues. First, the smooth fade-in of the Lasso'n'Cross icon may have hindered its discoverability since its appearance often went unnoticed and no one discovered Lasso'n'Cross. Second, for precise Drag'n'Cross insertion via Leafer or Compression widgets, the smooth animation of one item transitioning to the next caused ambiguity as to where exactly the item would be placed. Both of these issues and corrective strategies are discussed further below

Disable Physics as Necessary

When asked if things moved as they expected them to one user responded that it did most of the time but sometimes he wanted the ability to drag things across the screen without affecting other objects. This would also help one of the shortcomings of the tossing technique mentioned below, namely that you would not need to be mindful not to disturb other objects when tossing. Users felt less strongly (3.14/5) when asked if

“Organizing icons with this software [Is/Is Not] akin to organizing them on my desk”. One self-proclaimed “neat freak” participant said that her desk was more clean and tidy. We anticipated the increased degrees of freedom in positioning the icons in our prototype and therefore designed the axis-align mode discussed in the *Polite Physics* section. However, we did not enable this technique in the study.

5.1.2 Interaction Techniques

LassoMenu

While the LassoMenu was effective in enabling user’s discovery and transition to expert behaviour, there were some areas for improvement identified during our user evaluation. For instance, the LassoMenu was found to have a large footprint and was difficult to operate at the edges of the screen where encircling objects with a closed lasso, displaying a menu and moving the pen to specify a parameter became difficult. Also, when the pen was moved off the trackable screen and onto the bezel, the pen input stream is broken as if the pen was released, frustrating the user. Leaving the trackable area was common even when not hovering due to the smooth and continuous surface. One potential solution to the problems encountered at the edge of the screen would be adopting a FaST slider approach (McGuffin, Burtnyk et al. 2002) where once an action is selected from the LassoMenu a slider is then “glued” to the cursor and can be repositioned to a more convenient location before the parameter is specified. Also, the LassoMenu was often occluded by the user’s hand requiring physical adjustments to view occluded parts (this occlusion problem is inherent in most pen-based direct input systems). Encircling a single item was cumbersome and tedious. A crossing based selection may be appropriate here and would help the problem of selecting objects at the edges. Alternately, a click or pressure-based invocation of the LassoMenu for a single item could help.

When asked to increase the size of an icon by about two-fold one user was annoyed that he was unable to precisely set the size to two times that of the original. To accommodate this a snapping mechanism could be incorporated to bias the control menu toward rounder values. In addition some participants had trouble with the control display gain of certain parameter specifications on the LassoMenu. While this could be manually

tweaked for better results it may also be helpful to provide visual feedback with a slider during the parameter specification portion of the LassoMenu such as those used in FaST Sliders mentioned above. In addition, using FaST sliders for our purposes would be a novel 2D application of the concept, and may be useful in indicating that there are in fact two degrees of freedom for the parameter. Some participants only took advantage of one degree of freedom for certain LassoMenu options. For instance, the crumple option was placed South on the LassoMenu and users simply continued their stroke in the South direction, limiting their crumple to a single dimension. A 2D widget would indicate that there is a greater range of interaction possible.

Many users didn't remember to drag out of the LassoMenu circle thinking that clicking or just releasing the mouse in the highlighted menu region should activate functionality. One participant commented that the menu looked too much like a pie menu and that was a result of his confusion. Adding arrows or using a more typical marking menu visual representation may alleviate this problem.

Lasso'n'Cross

Lasso'n'Cross was the preferred technique for pile creation. Two users preferred Lasso'n'Cross, one user preferred LassoMenu, one user had no preference and two users didn't comment. Users quickly became comfortable with the technique, gracefully making a circular version of the gesture. Three users stated that making piles was amongst their favourite interaction techniques in the system. However, the evaluation did illuminate some areas for improvement to Lasso'n'Cross, particularly in its discoverability and support of more complex selections.

In attempt to make the Lasso'n'Cross gesture discoverable we displayed a monochrome icon with an arrow to indicate that it could be crossed, while the user was drawing their lasso. This turned out to not be strong enough in affording discovery of the Lasso'n'Cross technique. Users largely did not notice the icon or invoke Lasso'n'Cross during the discovery portion of the experiment, when not introduced to the technique before hand. This could be because the crossable icon blended in with the document icons and the smooth fade-in of the icon increased the problem. To make the icon more noticeable the fade-in could be removed and a high-contrast outline added. Animation

could be added for further emphasis, for instance a dwell-triggered animated extension of the current stroke across the crossable icon could be helpful, akin to the technique found in the Scriboli work (Hinckley, Baudisch et al. 2005). A similar strategy could be used to indicate that uncrossing is also possible, which sometimes went unnoticed by users.

Lasso'n'Cross selects items within the convex hull of the lasso stroke drawn by the user. Though the convex hull is not always precisely equal to the lasso stroke it was not significantly limiting in our study where many tasks required various combinations of items to be selected. In fact, users often kept their lassos simple and roughly circular in shape where the convex hull is equivalent to the lasso stroke. Further, users avoided intricate lassoing required for more complex selections by moving relevant items together so a simple circular lasso could be made to select them. One user was frustrated by not being able to make persistent selections, which would make selecting multiple items from different areas of the screen easier. A revised prototype should investigate techniques for selecting multiple non-contiguous items and fluidly committing actions on them. The ability to arbitrarily add or subtract from selections would also be useful.

Pile Browsing Techniques

Five out of six users stated that some sort of pile browsing amongst their favorite interaction techniques in the system. The most popular pile browsing techniques were Grid, Fanout and the Leafer. Three out of six participants mentioned the grid widget when asked “Which Interaction Technique(s) did you like the best?” in the post-study interview. Fanout was also popular, explicitly stated by two users, one who excitedly remarked about the technique “this is cool”. One user suggested it would be useful to allow the fanout technique to retract when dragged backwards along the path. In the current implementation the layout only grows with the stroke and the user must re-initiate the fanout if they want a shorter spread of icons. It was also noted that when the fanout path becomes excessively large icon placement along the path becomes unpredictable. Though, this is not a serious issue since releasing the mouse returns the pile to its previous state. One user remarked he liked the support for edge browsing and being able to get an idea of tidy pile contents by looking at the coloured edges of the icons.

Fisheye and Compression browse were the least popular techniques. Fisheye was rarely used as piles seldom reached a size where it offered any benefit, though this may not be the true in everyday use of the system. For piles in certain positions, items compressed with the compression browse technique occluded items below because of the perspective of the workspace. In addition, one user was confused when asked to re-orient items in a locked down pile under compression browse. They interpreted the compressed items as already rotated upwards to take up less space. One user stated that he preferred pile browsing techniques that allowed him to main his “illusion of reality” in the interface. We note that fisheye and compression browse were the least popular and do not physically plausible unlike the other browsing techniques. Though it could be that fisheye and compression browse techniques are not very effective in the domain of piles, we speculate that their lack of popularity may have been due to a desire for consistency with the physics metaphor in the interface.

Pressure Based Techniques

We observed that our pressure widget visual suggestions were not strong enough. When the cursor went hollow to indicate a pressure lock was possible (Figure 24c), it was too subtle and largely went un-noticed during the discovery portion of the experiment. However, when pressure techniques were learned they were quite successful. Users quickly learned the pressure lock for locking down a pile and pinning up items and accomplished these actions with little difficulty. Users enjoyed pinning and unpinning objects to the wall during idle time. The tasks involving pinning were found “easy” and done quickly with little error.

Crossing-based Interaction

Goal crossing as an interaction technique has recently received much attention in the use of pen-based interfaces as an alternative to the point-and-click paradigm which can be difficult with a pen (Accot and Zhai 2002; Apitz and Guimbretièrre 2004; Sachi and Michiaki 2004). Our interface combined crossable elements with techniques such as lasso selections and tossing and found that these are not always compatible. For instance, the pile widgets support activation from crossing-based activation in the direction of the pile outwards. Our initial design supported crossing widgets from outside of the pile

towards the center. However, this was removed because we observed in our pilot study that user's would accidentally activate a widget by crossing it while trying, often quickly, to make a lasso selection or use the Lasso'n'Cross technique to create a pile. Additionally, to support outward crossing for pile widgets we were unable to support dragging to move piles or dragging pile items outside of piles. This was problematic as almost all users initially expected piles and their contained items to be moveable by dragging them like unpiled items and attempted to move them as such. Mander et al. (1992) encountered the same problem of often trying to click and drag on piled items to move them. Further, crossing was difficult and awkward when the widgets were at the borders of the display because of the lack of screen for the swiping motion required of the pen. The concerns above raise the more general issue of how crossing-based techniques can be made to live comfortably with other interaction techniques in a single interface.

Crossing also provided positive benefits in our interface. Crossing was preferred to clicking on widgets two to one amongst the three users who commented or had a preference. One user exclaimed he preferred crossing "big time", while the other commented how it felt "smoother" than clicking. Of the users without a clear preference, one stated that he liked crossing on widgets when he felt it continued the direction and "maintained the illusion of reality" of the interaction, specifically citing the Fan-out and Leafer widgets as examples. One experienced pen user felt that the visual design of the widgets evoked a click and not a cross. Some of the visualizations for crossable targets mentioned in previous work (Accot and Zhai 2002; Apitz and Guimbretière 2004) could be use to mitigate this.

Self-Revealing Widgets

In addition to the widget concerns mentioned below in the Drag'n'Cross section, our user evaluation also raised issues regarding our use of self-revealing pile widgets in pen-based systems. Widgets that revealed themselves based on pen position led to accidental invocations since they appeared surprisingly when the pen was lifted from and returned to the trackable area to interact with objects nearby the suddenly appearing widgets. For example, a user would enter the hover zone and click down too quickly to notice the widget under their pen while trying to lasso select some objects. In addition,

some pile widgets became inaccessible when their parent pile was moved to the edges of the display since they were re-located off screen. Further, when piles got too close to each other their widgets would occlude and intersect each other. To prevent these situations pile movement can be restricted so that they never occur. Widgets also appeared when not necessary, such as during a lasso selection when they cannot be used and revealing them causes confusion. While the self-revealing widgets provided the benefit of an uncluttered workspace, they can be refined to maximize effectiveness.

Drag'n'Cross

Three out of six users encountered difficulty with the Drag'n'Cross technique for precise insertion of items into a pile, making some mention of precise insertion when asked in the post-study questionnaire “Which Interaction Technique(s) gave you the most trouble today?” Note that Dwell'n'Scrub was not included for comparison due to technical issues. There were several accidental invocations since the Drag'n'Cross widgets appeared only when pen was very close to a pile and went unnoticed since the pen was often being moved quickly and casually. To reduce surprising the user it may be helpful to remove the self-revealing nature of the Drag'n'Cross widgets, having them appear immediately after item dragging is initiated. The frustration of a false invocation was exacerbated due to the lack of easy undo once the Drag'n'Cross widget was initially crossed.

Since Drag'n'Cross was intended for precise placement inside a pile, our design overlooked a subtle feature that made it suboptimal for placing items on top of a pile. Namely, when Drag'n'Cross is invoked it starts animating the transition with the insertion point after the first item. While it is possible to insert on the top it requires backtracking and caused confusion. This problem could be rectified by placing the initial insertion point at the top of the pile and delaying animation of the leafing or compressing, depending on the widget. In addition, some participants had trouble with precise insertion as it was not always clear where the item would be placed upon pen release. The smooth animation of these techniques often had items appear in ambiguous intermediate poses as they animated. The insertion point could be clarified by adding a visual indicator.

When asked for their preference amongst techniques for precise Drag'n'Cross insertion into a pile three chose the Leafer technique while one stated no preference between Leafer and the Compression Browse widget. A clear preference is surprising since the only difference in the techniques was in the visualization when scrolling through documents in the pile. One user commented that he preferred the Leafer widget because it helped him maintain his illusion reality in the interface, as discussed earlier.

Tossing

There was a great deal of user preference for tossing to add items to a pile versus drag and drop. Four users preferred tossing, one preferred dragging and one had no preference. Participants emphatically stated tossing “[is] just so fun” and “I like to throw it!” One user felt that tossing wasn’t precise, such as in the case of two piles close together, it could be difficult to toss an item at a specific one. In addition tossing to a pile beyond another pile was impossible in our current implementation. Our current algorithm simply added items tossed towards a pile above a certain velocity and within a certain radius of the closest pile. This could be improved by allowing an “escape velocity” for tossed items, enabling items to be tossed beyond piles if they’re going fast enough or not in the direction of the pile. Tossing was sometimes difficult without “runway space” created by first moving items to an area where there was enough space to toss it without disturbing other objects.

Pile Movement

One user complained that the when the piles were dragged they didn’t feel like they moved “right”. In our implementation springs were created with complete graph connectivity when a pile was dragged to keep items together. This user described the resulting sluggish movement “jello piles”. In addition, piles didn’t support tossing like document icons. In a revised system piles should support tossing and pile movement tweaked and sped up, closer to the feel of moving document icons in the interface which was found effective and enjoyable.

6 Conclusions and Future Directions

*"Do the best you can, and after that Funk it,
and everything will be alright by Thursday"*

-George Clinton

In this thesis we have presented a novel take on the ubiquitous desktop metaphor for personal computer information management. Our aim was to enhance the desktop in support of more casual, expressive and powerful techniques found on real world desks. We developed a suite of interaction and visualization techniques that explore the use of piles as the primary organizational entity for desktop objects. A wealth of pen-centric, discoverable, fluid techniques were designed for our specific application that are also applicable in general pen interfaces. Apart from the specific techniques, our contributions include a detailed exploration of the piling metaphor, and an integration of the techniques into a coherent prototype that ensures that the techniques operate well as a whole. A qualitative user study was conducted involving simulated tasks. Results are encouraging and illuminate implications for future physics or pen based interface designs. The evaluation confirmed that our techniques were not only useable and discoverable but also highlighted areas for improvement and revealed directions for future research. Our primary contributions are summarized in detail below.

6.1 Contributions

Our goal was not to create a definitive system to solve the problems of computer organization; rather we wanted to explore alternative designs that attempt to regain some of the lost benefits in the transition to digital desktops. This research contributes original ideas, knowledge and interaction techniques to the field of human computer interaction. We now discuss the contributions presented in the preceding chapters.

Physical Desktop Prototype

The main contribution of this work is the BumpTop prototype which integrates a physical simulation with the standard idea of the desktop metaphor. The prototype runs in realtime on commodity TabletPC hardware and leverages the timing of a new pen-based platform without a successful desktop interface. Traditional desktop icons now inhabit a 2½D desktop. Icons are represented with 3D geometry and given some of the affordances of paper. The increased range of motion translates to subtle and more expressive arrangements of documents. Rigid body dynamics, collision detection and frictional forces allow documents to move realistically, bumping and displacing each other when they collide. The potential for disorder with the increased freedom is restricted with judicious use of physics in techniques such as enforcing axis alignment. Techniques for manipulation, organization, annotation, emphasis and search of document objects were presented. The techniques apply to objects of arbitrary sizes such as photos, or windows as demonstrated. In chapter 3 we established design goals to guide our development based on prior research as well as our own experience.

Interaction and Visualization Techniques for Piles

While there have been previous investigations of piles we integrate and significantly expand upon them. Our work investigates un-explored areas of pile-pile interaction, intra-pile interaction, pile widgets, the transition between unpiled and piled information and piling of arbitrarily sized objects. In addition, we emphasize interface discoverability which has hindered the usability of previous designs. We also incorporate several advances in browsing techniques developed for other applications (Carpendale, Cowperthwaite et al. 1999; Carpendale and Montagnese 2001; McGuffin, Tancau et al.

2003). A messy pile representation is investigated whose “messiness” has meaning derived from its initial pose before being added to the pile, unlike previous approaches whose messiness is created by random displacement.

Pen-centric Interaction Techniques

We introduced novel, fluid pen-centric interaction techniques which synthesize in original ways recent research advances in pen-based techniques. For instance, the LassoMenu successfully synthesized lasso selections and control menus in a technique that allows fluid selection and command invocation in a discoverable way, and supports transitioning of novice to expert behaviour. Other novel techniques include crossable widgets which were popular but are susceptible to the pen interaction issues discussed in chapter 5. Lasso’n’Cross is a simple, discoverable gesture that supported undo and multiple forms of invocation. Drag’n’Cross is a crossing based technique akin to drag and drop for precise insertion into a pile.

User Evaluation of Physics Desktop

We conducted a qualitative user evaluation with six users whose computer skills ranged from novice to pen-interface expert. Participants were able to discover functionality after only 3 minutes of basic introduction to the system. Users completed approximately 88% of simulated tasks after additional but brief training. This confirms that the interface was indeed discoverable and learnable. In addition, many users found the physical interface playful, fun and satisfying. Our evaluation revealed insights on our own designs as well for designers of physical and pen-based interfaces. User feedback confirmed the LassoMenu was an effective, discoverable technique but that it could be improved. Specifically, its usage at the edges of the screen or invocation on a single item were identified for improvement. In addition its visual representation could benefit from a traditional marking menu design. Insights to pen interface designers from our study include the mindful use of crossing with other interaction paradigms, awareness of technique deterioration at the screen edges and discouraging the use of pen position to indicate user focus and automatically hide/reveal information. Drag’n’Cross was susceptible to false invocations. Lasso’n’Cross was found to be a popular, enjoyable gesture that was quickly learned, though its discoverability could be improved.

6.2 Future Directions

Extending the use of Physics

One of the most common suggestions from those who have tried our prototype is that we should make increased use of document meta-data. For example, we could map file size to mass or volume or friction index of the object's representation. Also, other physical properties could be used as cues for conveying content information. For example, larger files might move slower because they feel 'heavier' or older files appear dog-eared to show their wear. We could also explore using other physical phenomena to guide the designs. For example, we could model objects as sheets of paper that can be folded in interesting ways to convey information or draped over other objects. One user suggested that it would be useful to be able to pin something to the desktop like you could to the walls. The same pressure lock and tear away techniques could be used to pin and unpin items. Another possibility would be to use gravitational fields that surround piles and attract other similar files. For instance a pile full of photos would have a greater pull on a photo being tossed by it than a neighbouring pile of spreadsheets. A gravitational field implementation would also support the escape velocity described in the *User Evaluation* section to allow items to be tossed beyond piles. Another popular suggestion is the extension of the physical metaphor to include a magnet to gather like icons from the desktop. For instance, a video magnet could be used to attract video files from a large unorganized heap of documents while maintaining context. Multiple magnets could be used in conjunction to see spatial relationships— similar to the Dust and Magnet work (Yi, Ponder et al. 2005).

Longitudinal Study

To evaluate BumpTop's validity in practice as an alternative to the standard desktop we would like to conduct a more longitudinal study where our prototype is used in situ as a desktop replacement for an extended period of time. This would require a few changes to the current interface. First, it would need to pull data from the user's current desktop including textual labels for existing named icons. Second, it is not clear how today's windowed applications would live in our interface. One approach involves

rethinking windows through the physics paradigm. Another alternative would be a mixed mode approach where icons and folders existing on the physical desktop launch the windows we are used to today. Finally, at present full time pen-users are extremely rare and so a longitudinal study of our designs would likely be used by more typical mouse users. While our techniques were designed to be pen-centric, the mouse can be used to drive the current prototype. Although, many of the interactions such as crossing and tossing do not feel as fluid when used with a mouse, and the pressure-based techniques do not transfer well. To support a longitudinal study and broaden the influence of our techniques a version of the prototype re-designed for the mouse could be explored.

Another concern in a more realistic setting is scalability of our techniques. Like the standard GUI desktop, our prototype also runs into problems when the number of items gets large. As Whittaker and Hirschberg (2001) found, “the main limitation of [piling] was that it did not scale well: pilers found difficulties accessing information once piles had begun to multiply”. It would also be interesting to explore extensions that might deviate somewhat from the physical piling metaphor but benefit from leveraging the underlying computer.

Bibliography

(1994). eWorld. Cupertino, California, Apple Computer.

(1994). MagicCap. Sunnyvale, California, General Magic.

(1995). Bob. Seattle, Washington, Microsoft Corporation.

Accot, J. and S. Zhai (2002). More than dotting the i's - foundations for crossing-based interfaces. ACM CHI Conference on Human Factors in Computing Systems, New York, NY, ACM.

Agarawala, A. and R. Balakrishnan (2006). Keepin' It Real: Pushing the Desktop Metaphor with Physics, Piles, and the Pen. ACM CHI Conference on Human Factors in Computing Systems, Montreal, Quebec, Canada, ACM.

Anderson, J. R. (1980). Cognitive skills and their acquisition. Hillsdale, N.J., Lawrence Erlbaum.

Apitz, G. and F. Guimbretière (2004). CrossY: a crossing-based drawing application. ACM UIST Symposium on User Interface Software and Technology.

Apple Computer Inc. (1992). MacIntosh Human Interface Guidelines, Addison-Wesley Longman Publishing Co., Inc.

Baraff, D., A. Witkin, et al. (1997). "An introduction to physically based modeling: particle system dynamics." ACM SIGGRAPH Course Notes.

Barreau, D. and B. A. Nardi (1995). "Finding and reminding: file organization from the desktop." SIGCHI Bull. 27(3): 39-43.

Bartram, L. (1997). Can motion increase user interface bandwidth. IEEE Systems, Man and Cybernetics.

Bauer, D., P. Fastrez, et al. (2004). Computationally-enriched "piles" for managing digital photo collections. IEEE VLHCC.

Bauer, D., P. Fastrez, et al. (2005). Spatial Tools for Managing Personal Information Collections. Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 4 - Volume 04, IEEE Computer Society.

Beaudouin-Lafon, M. (2001). Novel interaction techniques for overlapping windows. ACM UIST Symposium on User Interface Software and Technology, New York, NY, ACM.

Bederson, B. and J. Hollan (1994). Pad++: a zooming graphical interface for exploring alternate interface physics. ACM UIST Symposium on User Interface Software and Technology.

Blaine, B., F. Steven, et al. (2001). View management for virtual and augmented reality. Proceedings of the 14th annual ACM symposium on User interface software and technology. Orlando, Florida, ACM Press.

Cao, X. and R. Balakrishnan (2005). Evaluation of an online adaptive gesture interface with command prediction. GI 2005 – the Graphics Interface Conference.

Card, S. K., G. G. Robertson, et al. (1996). The WebBook and the Web Forager: an information workspace for the World-Wide Web. Proceedings of the SIGCHI conference on Human factors in computing systems: common ground. Vancouver, British Columbia, Canada, ACM Press.

Carpendale, M. S. T. and C. A. Montagnese (2001). A framework for unifying presentation space. ACM UIST Symposium on User Interface Software and Technology, New York, NY, ACM.

Carpendale, S., D. Cowperthwaite, et al. (1999). The Tardis: A visual exploration environment for landscape dynamics. SPIE Conference on Visual Data Exploration and Analysis VI.

Carroll, J. M. and J. C. Thomas (1980). Metaphor and the cognitive representation of computing systems. Report RC 8302, IBM Watson Research Center.

Chang, B. and D. Ungar (1993). Animation: From cartoons to the user interface. ACM UIST Symposium on User Interface Software and Technology.

Chao, D. (2001). Doom as an interface for process management. Proceedings of the SIGCHI conference on Human factors in computing systems. Seattle, Washington, United States, ACM Press.

Clemmensen, T. (2002). Interest in theory and method in a Danish community of usability specialists (in Danish). Working paper, Copenhagen Business School, Department of Informatics, HCI research group.: 1-30.

Collomb, M. and M. Hascoët (2004). Speed and accuracy in throwing models. HCI'04, Leeds, UK, British HCI Group.

Czerwinski, M., M. v. Dantzich, et al. (1999). The contribution of thumbnail image, mouse-over text, and spatial location memory to web page retrieval in 3D. Interact.

Denoue, L., L. Nelson, et al. (2003). A fast, interactive 3D paper-flier metaphor for digital bulletin boards. ACM UIST Symposium on User Interface Software and Technology.

DiGioia, P. and P. Dourish (2005). Social navigation as a model for usable security. ACM SOUPS.

- Dourish, P., W. K. Edwards, et al. (1999). "Presto: an experimental architecture for fluid interactive document spaces." ACM Trans. Comput.-Hum. Interact. **6**(2): 133-161.
- Dragicevic, P. (2004). Combining crossing-based and paper-based interaction paradigms for dragging and dropping between overlapping windows. ACM UIST Symposium on User Interface Software and Technology.
- Duncker, K. (1945). On Problem Solving, American Psychological Association.
- Ericsson, K. A. and H. A. Simon (1984). Protocol analysis: Verbal reports as data. Cambridge, MA, MIT Press.
- Fitzmaurice, G., A. Khan, et al. (2003). Tracking menus. Proceedings of the 16th annual ACM symposium on User interface software and technology. Vancouver, Canada, ACM Press.
- Fitzmaurice, G., A. Khan, et al. (2003). Tracking menus. ACM UIST Symposium on User Interface Software and Technology, New York, NY, ACM.
- Forlines, C. and C. Shen (2005). Glimpse: a novel input model for multi-level devices. CHI '05 extended abstracts on Human factors in computing systems. Portland, OR, USA, ACM Press.
- Gentner, D. and J. Nielsen (1996). "The Anti-Mac interface." Commun. ACM **39**(8): 70-82.
- Gibson, J. J. (1979). The ecological approach to visual perception. Boston, Houghton Mifflin.
- Gonzalez, C. (1996). Does animation in user interfaces improve decision making. ACM CHI Conference on Human Factors in Computing Systems.
- Good, N. S. and A. Krekelberg (2003). Usability and privacy: a study of Kazaa P2P file-sharing. Proceedings of the SIGCHI conference on Human factors in computing systems. Ft. Lauderdale, Florida, USA, ACM Press.
- Greenberg, S. and C. Fitchett (2001). Phidgets: easy development of physical interfaces through physical widgets. Proceedings of the 14th annual ACM symposium on User interface software and technology. Orlando, Florida, ACM Press.
- Grossman, T., R. Balakrishnan, et al. (2001). Interaction techniques for 3D modeling on large displays. ACM I3DG 1999 Symposium on Interactive 3D Graphics, New York, NY, ACM.
- Guimbretière, F., M. Stone, et al. (2001). Fluid interaction with high-resolution wall-size displays. ACM UIST Symposium on User Interface Software and Technology, New York, ACM.
- Guimbretière, F. and T. Winograd (2000). FlowMenu: combining command, text, and data entry. ACM UIST Symposium on User Interface Software and Technology, ACM.
- Herndon, K., R. Zeleznik, et al. (1992). Interactive shadows. ACM UIST 1992 Symposium on User Interface Software and Technology, New York, NY, ACM.
- Hinckley, K., P. Baudisch, et al. (2005). "Design and analysis of delimiters for selection-action pen gesture phrases in Scriboli." ACM CHI Conference on Human Factors in Computing Systems: 451-460.

- House, D. H. and D. E. Breen, Eds. (2000). Cloth modeling and animation, A. K. Peters, Ltd.
- Igarashi, T., S. Matsuoka, et al. (1999). Teddy: a sketching interface for 3D freeform design. ACM SIGGRAPH.
- Ishii, H. and B. Ullmer (1997). Tangible bits: towards seamless interfaces between people, bits and atoms. ACM CHI Conference on Human Factors in Computing Systems, New York, NY, ACM.
- Johnson, J. (1987). "How faithfully should the electronic office simulate real one?" SIGCHI Bull. **19**(2): 21-25.
- Kurtenbach, G. (1993). The design and evaluation of marking menus. Department of Computer Science. Toronto, Canada, University of Toronto.
- Kurtenbach, G. and W. Buxton (1991). "Issues in combining marking and direct manipulation techniques." ACM UIST Symposium on User Interface Software and Technology: 137-144.
- Kurtenbach, G. and W. Buxton (1993). The limits of expert performance using hierarchical marking menus. ACM CHI Conference on Human Factors in Computing Systems, New York, NY, ACM.
- Kurtenbach, G. and W. Buxton (1994). User learning and performance with marking menus. ACM CHI Conference on Human Factors in Computing Systems, New York, NY, ACM.
- Malone, T. (1983). "How do people organize their desks?: Implications for the design of office information systems." ACM Transactions on Information Systems **1**(1): 99-112.
- Mander, R., G. Salomon, et al. (1992). A "pile" metaphor for supporting casual organization of information. ACM CHI Conference on Human Factors in Computing Systems.
- McGuffin, M., N. Burtnyk, et al. (2002). FaST Sliders: Integrating marking menus and the adjustment of continuous values. Graphics Interface, Calgary, Alberta, Canadian Human Machine Communications Society.
- McGuffin, M., L. Tancau, et al. (2003). Using deformations for browsing volumetric data. Proceedings of IEEE Visualization.
- Miller, L. (2005). Case Study of Customer Input for a Successful Product. Agile 2005.
- Mynatt, E., T. Igarashi, et al. (1999). Flatland: New dimensions in office whiteboards. ACM CHI Conference on Human Factors in Computing Systems, New York, ACM.
- Nardi, B. and D. Barreau (1997). "'Finding and reminding" revisited: appropriate metaphors for file organization at the desktop." SIGCHI Bulletin **29**(1): 76-78.
- Nielsen, J. (1993). In Usability Engineering, Academic Press.
- Pederson, E., K. McCall, et al. (1993). Tivoli: An electronic whiteboard for informal workgroup meetings. ACM CHI Conference on Human Factors in Computing Systems, New York, ACM.

Pook, S., E. Lecolinet, et al. (2000). Control menus: Execution and control in a single interactor. ACM CHI Conference on Human Factors in Computing Systems (Extended Abstracts), New York, NY, ACM.

Ramos, G., M. Boulos, et al. (2004). Pressure widgets. ACM CHI Conference on Human Factors in Computing Systems.

Ravasio, P., Sissel Guttormsen Schar, et al. (2004). "In pursuit of desktop evolution: User problems and practices with modern desktop systems." ACM Trans. Comput.-Hum. Interact. **11**(2): 156-180.

Robertson, G., M. Czerwinski, et al. (1998). Data mountain: Using spatial memory for document management. ACM UIST 1998 Symposium on User Interface Software and Technology, New York, NY, ACM.

Robertson, G., M. v. Dantzich, et al. (2000). The Task Gallery: A 3D Window Manager. ACM CHI 2000 Conference on Human Factors in Computing Systems, New York, NY, ACM.

Robertson, G., J. Mackinlay, et al. (1991). Cone trees: Animated 3D visualization of hierarchical information. ACM CHI Conference on Human Factors in Computing Systems.

Robertson, G. G. and J. D. Mackinlay (1993). The document lens. Proceedings of the 6th annual ACM symposium on User interface software and technology. Atlanta, Georgia, United States, ACM Press.

Rose, D. E., R. Mander, et al. (1993). Content awareness in a file system interface: implementing the "pile" metaphor for organizing information. Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval. Pittsburgh, Pennsylvania, United States, ACM Press.

Roussel, N. (2003). Ametista: a mini-toolkit for exploring new window management techniques. Proceedings of the Latin American conference on Human-computer interaction. Rio de Janeiro, Brazil, ACM Press.

Sachi, M. and Y. Michiaki (2004). Tapping vs. circling selections on pen-based devices: evidence for different performance-shaping factors. Proceedings of the SIGCHI conference on Human factors in computing systems. Vienna, Austria, ACM Press.

Sellen, A. and R. Harper (2003). The myth of the paperless office, MIT Press, Cambridge, MA.

Smith, D. (1982). Designing the Star User Interface. Byte: pp. 242-282.

Sonnet, H., S. Carpendale, et al. (2004). Integrating expanding annotations with a 3D explosion probe. AVI Conference on Advanced Visual Interfaces.

Streitz, N., J. Geißler, et al. (1999). i-LAND: an interactive landscape for creativity and innovation. ACM CHI Conference on Human Factors in Computing Systems, New York, NY, ACM.

- Tapia, M. and G. Kurtenbach (1995). Some design refinements and principles on the appearance and behavior of marking menus. ACM UIST Symposium on User Interface Software and Technology, New York, NY, ACM.
- Thorne, M., D. Burke, et al. (2004). "Motion doodles: an interface for sketching character motion." ACM Transactions on Graphics **23**(3): 424-431.
- Tristram, C. (2001). The Next Computer Interface. MIT Technology Review: December.
- Weiser, M. (1991). "The computer for the 21st century." Scientific American **265**(3): 94-104.
- Wellner, P. (1993). "Interacting with paper on the digital desk." Communications of the ACM **36**(7): 87-96.
- Whittaker, S. and J. Hirschberg (2001). "The character, value, and management of personal paper archives." ACM Transactions on Computer-Human Interaction **8**(2): 150-170.
- Woods, D. (1984). "Visual momentum: a concept to improve the cognitive coupling of person and computer." International Journal of Man-Machine Studies **21**: 229-244.
- Yi, J. S., R. M. Ponder, et al. (2005). "Dust & Magnet: Multivariate Information Visualization using a Magnet Metaphor." Information Visualization **4**(4): 239-256.
- Zelevnik, R. C., K. Herndon, et al. (1996). SKETCH: An interface for sketching 3D scenes. ACM SIGGRAPH Conference on Computer Graphics and Interactive Techniques, New York, NY, ACM.

7 Appendix A:

Experiment Protocol and Questionnaire

BumpTop Prototype Evaluation (PI: A. Agarawala)

Protocol:

- Present and obtain signed consent form
- Read instructions to subject:

- Thank you for agreeing to participate in this study. The session will last approximately 60 minutes. You are free to take a break at any time, or cease participation without penalty. Throughout the session, a video camera will record your actions, and the microphone in front of you will record your voice and I will be taking notes.
- You are encouraged to **talk aloud** about what you are doing. Feel free to express confusion or happiness with the system, and if the system is not doing what you wanted, etc.
- Please remember: we are here today to evaluate a system, not to evaluate you: please relax, take your time, and feel free to ask any questions you may have.

- You are now seated at a tablet computer, all your input to this system will be accomplished by this pressure-sensitive pen that operates when hovering above or touching the screen surface.

- The first part of the experiment is exploratory: after some basic instructions you will be given 10 minutes to try the system, and see how much of its functionality you can *discover*. During this time I will be observing your actions and taking notes. Please feel free to make comments and ask questions at any time, though most questions will be recorded and answered *after* this phase of the experiment is complete.

BUMPTOP and INTERACTIONS INTRODUCTION

- The system you will be using today is designed to organize and manipulate icons as you would papers on a desk. Motion is simulated with physics to make movements more realistic. You drag items to move them, and toss.
 - Many basic interactions on this desk, such as creating a piles are performed by the *LassoMenu*. The LassoMenu is invoked when circling to select objects and closing the lasso by entering the blue circle that appears where you started and pulling out in the direction of the desired interaction.
 - After you select your interaction the rest of the pen movement will let you adjust it [show how this works, with pile creating example]. The LassoMenu works on single, multiple items, piles or empty selections.
 - Piled items can be browsed or manipulated by clicking or crossing on one of their *Widgets* and keeping the pen touching the surface. In all but the moving and pile-breaking widget, when you release the pen things go back to normal.
 - To keep things laid out you can use the *PressureLock* technique where you push hard with the pen until the green circle fills the cursor and turns white. [show locking of pile layout]. Click the X to go back to how things were. PressureLock can be used to lock other things, the cursor will turn a faded white when you can PressureLock something.
-
- I will now give you 10 minutes to discover the functionality of the system. Please feel free to try anything you like, make comments, and ask questions. Do you have any questions before we begin?
-
- Allow the participant to begin, make notes of their actions.
 - I will now ask you to perform various specific tasks using the system. Please feel free to ask questions as you perform these tasks. Are you ready to proceed? (read each of the task descriptions as it appears on the notes sheets).
 - Thank you for your help today! We are eager to hear any thoughts, comments, suggestions, or questions you may have about our system. It is a research prototype, and is open to extensive modification.

Feature Checklist & Notes: (PI: A.Agarawala)

Participant Number: _____

Date: _____

Feature Discovery/Use Checklist

<i>Feature</i>	<i>Discovered</i>	<i>Explained</i>	<i>via LM</i>	<i>via Widget</i>
LassoMenu				
Lasso Completion		X		
Self-Intersection				
Pressure Lock (h.widget indicates possible)		X		
Single/Multiple Item				
Move by Drag				
Toss				
Nudge items around (push button)				
Delete				
Duplicate				
Re-Orient				
Re-Orient (lock)				
Resize				
Crease				
Crumple				
Pin up to wall (lock)				
Lasso Selection				
Multiple Items				
Move (springs)				
Browse: Grid (piles too)				
Browse: Compression				
Tidy				
Turn into pile				(CentCross, undo)
Pile				
Move				
Duplicate				

Delete				
<i>Widget Click or Cross [ToolTips]</i>				
Browse: Grid (lock)				
Browse: Fisheye (lock)				
Browse: Fanout (lock)				
Browse: Leaf-through (lock)				
Browse: Compression browse (lock)				
Browse: Messy State				
Break Pile (back to original positions)				
Sort by Type				
Sort by Size				
Insert item: Drag ontop				
Insert item: Toss ontop				
Insert item: dwell and scrub				
Insert item: cross Leafer				
Insert item: cross Compressor				
Merge Piles				CentCross
Break Merged Piles				
<i>Intra-Pile (Pile Laid Out)</i>				
Sort by Type				
Sort by Size				
Delete				
Re-arrange item order (click or LM)				
Emphsize: stick out of pile				
Emphasize: rotate				
Pull item(s) out of pile				

Experiment Tasks

1. Casually move the music file from the bottom right corner to the top of the music pile, moving your pen as little as possible.
2. Delete a PDF from the unorganized items in the bottom-left corner.
3. Gather all video files in the unorganized items in the bottom-left corner and create a pile out of them. This pile should be located in the vacant bottom-middle portion of the desktop.
4. Duplicate video pile and merge it with music pile
5. Precisely place a diskette from the unorganized items into the Annual Report above the topmost PDF.
6. In the Annual Report Pile: sort all items below the diskette by Type.
7. Delete the 2nd last item in the Annual Report pile.
8. Move the top item in the Annual Report pile to the right of the diskette you previously inserted.
9. Rotate the spreadsheets in the Annual Report pile so they stick out.
10. One of the piles contains a star icon with a number in it. Locate this star icon, what is the number on the Star? Answer: _____.
11. Find all PDFs in the central pile and increase their size by about 2-fold. When you are done all icons/piles should be in the same positions as when you started.
12. Count how many diskette icons the entire desktop contains. When you are done all icons/piles should be in the same positions as when you started. Number of Diskette Icons: _____.
13. Break apart the music and video pile you previously merged. Then delete the video pile.
14. Alter the appearance of one the enlarged PDF icons so that you remember it for a later task. Do not change its size.
15. Crumple up a different enlarged PDF and toss it into a vacant corner.
16. Pin up one of the un-altered enlarged PDFs onto the wall.
17. Take one of the word documents from the unorganized items and carefully place it on top of the pile of movies.
18. Place the PDF whose appearance you altered in step X at the bottom of the of the pile of Movies.
19. Make a copy of one of the un-altered enlarged PDFs
20. Delete the crumpled up PDF.
21. Use the positioning of of the Star icon in the pile of videos to emphasize it so it is visible when you leave the pile, without effecting the order of items in the pile.
22. Pull the star out of the movies pile and pin it up to the wall.
23. Pull the Pinned up PDF off the wall and casually put it on top of the pile of movies. Quickly order the pile so that the largest items are at the bottom.
24. Make 3 of the unorganized items permanently stand up on an edge.
25. Count how many word documents the central pile contains. When you are done all icons should be in the same positions as when you started. Number of word documents: _____.
26. Where in the central pile is the lightning bolt icon? (It is buried and not visible at first glance). Determine this without altering the positions of the icon piles. Circle one:
Top-left Top -right Bottom-left Bottom-right CENTER

27. Cleanup the unorganized pile a bit and make it take up less desktop space, but still try and maintain some of its spatial positioning.
28. Turn the unorganized items into a pile.
29. Quickly determine the 2nd and 3rd item in the pile you just created. What are they:
2nd Item: _____ 3rd Item: _____.

Post-Study Questionnaire

1. **Gender:** Male Female
2. **Age:** 18-24 25-34 35-44 45-60
3. **Highest Level of Education Completed/Enrolled In:**
 High School Post-Secondary Graduate Post-Graduate
4. **Handedness:** Right Left
5. **Have you ever used a Tablet-based PC (including any kind of external wacom tablet)?** Yes No
6. **Have you ever used a PocketPC, Palm or other pen-based device?**
 Yes No
7. **Do you have experience in pen/brush-based art?**
 None A Little A Lot
8. **How much time do you spend on a computer a day?**
 1-2 hours 3-5 hours 5+ hours

Today you were exposed to a number of interaction techniques for manipulating items and piles. Please read the following sentences and rate your preferred technique by putting a checkmark in their respective box.

9. **Which technique did you prefer for creating a pile (check one):**
 Lasso'n'Cross LassoMenu No preference
10. **Which technique did you prefer for Browsing items? (check one):**
 Clicking the Widget Crossing the Widget
 Using the LassoMenu No preference
11. **Which technique did you prefer for adding an item to the top of a pile (check one):**
 Tossing towards the pile Dragging on to the pile No preference
12. **Which technique did you prefer for inserting items to a specific place within a pile (check one):**
 Dwell and Scrub Drag-and-Cross on Leaf-Through Widget
 Drag-and-Cross on Compression-Browse Widget No preference

Please rate the following statements by checking one of the choices:

13. I [Liked/Disliked] the software.

Disliked	Somewhat Disliked	Neither Liked/Disliked	Somewhat Liked	Liked

14. The software is [Efficient/Inefficient].

Inefficient	Somewhat Inefficient	Neither Efficient/ Inefficient	Somewhat Efficient	Efficient

15. The software is [Easy/Difficult] to use.

Difficult	Somewhat Difficult	Neither Easy/Difficult	Somewhat Easy	Easy

16. The interaction techniques are [Easy/Difficult] to learn.

Difficult	Somewhat Difficult	Neither Easy/Difficult	Somewhat Easy	Easy

17. The software feels [Familiar/Unfamiliar].

Unfamiliar	Somewhat Unfamiliar	Neither Familiar / Unfamiliar	Somewhat Familiar	Familiar

18. The software [Matched/Did not Match] the way I expect objects to behave in the real world.

Did Not Match	Somewhat Did Not Match	Neither Matched/ Did Not Match	Somewhat Matched	Matched

19. It is [Easy/Difficult] to find the items I am looking for

Difficult	Somewhat Difficult	Neither Easy/Difficult	Somewhat Easy	Easy

20. I feel [In Control/Not In Control] of the software

Inefficient	Somewhat Inefficient	Neither In Control/ Not In Control	Somewhat In Control	In Control

21. I am [Able/Unable] to accomplish what I am trying to do

Unable	Somewhat unable	Neither Able/Unable	Somewhat Able	Able

22. Organizing icons is [Easy/Difficult] with the software

Difficult	Somewhat Difficult	Neither Easy/Difficult	Somewhat Easy	Easy

23. Organizing icons with this software [Is/Is Not] akin to organizing them on my desk

Is Not	Somewhat Is Not	Neither Is/Is Not	Somewhat Is	Is

Verbal Questionnaire

24. **For browsing files, which of the following do you primarily use (check one)?**

Graphical Desktop/Icon view folders

Hierarchical Tree Browser

Command Line

Other, please specify _____

25. **Which Interaction Technique(s) gave you the most trouble today?**

26. **Which Interaction Technique(s) did you like the best?**

27. **Did things move on the desk as you expected them to? If not, when did they not.**

28. **Do you feel like you got better with the techniques as time went on?**

29. **Do you have any suggestions for ways to improve the software?**

Please tell us your final thoughts and/or impressions you may have about the software or your experience in this study.

Thank you! This completes the session.