Keepin' It Real: Pushing the Desktop Metaphor with Physics, Piles and the Pen

Anand Agarawala and Ravin Balakrishnan
University of Toronto
www.dgp.toronto.edu
anand | ravin @dgp.toronto.edu

ABSTRACT
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 lightweight objects in the real world. 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.

Author Keywords
Piles, physics-based desktop, pen-based interfaces

ACM Classification Keywords
H.5.2 [User Interfaces]: Interaction styles

INTRODUCTION

Despite the metaphor, current virtual desktops (Figure 1a) bear little resemblance to the look or feel of real world desktops (Figure 1b). A workspace in the physical world typically has piles of 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 piles of items are "automatically" ordered chronologically because new items are typically placed on top. This casual organization, prevalent in the real world, differs greatly from the GUI desktop which forces users to immediately file their documents into a rigid hierarchy. Filing typically requires more effort than piling and has been shown [39] to have other negative effects such as encouraging premature storage of low value documents, or retaining useless documents because of the effort that went into filing them.

In this paper, we explore adding more realism to the virtual desktop to leverage some of the valuable characteristics of the real world. Our aim is not to debate the validity of the desktop metaphor [38], but to explore alternative designs to the ubiquitous desktop paradigm. Instead of the rigid organizational structures imposed by current virtual desktops, we use piles as a more casual organizational entity (Figure 1c). 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 the objects they own. Another goal 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 aim to leverage user's spatial memory and knowledge of how things move physically in the real world. To aid in our exploration, we have developed a set of pen-based interaction and visualization techniques, integrated into a prototype virtual desktop, called BumpTop. Using a pen as the primary input device can potentially enhance the feeling of realism and directness of manipulation since objects being acted upon are visible directly under the pen tip. In addition, there is currently no established standard for pen-based interfaces, making it an area ripe for influencing with new designs.

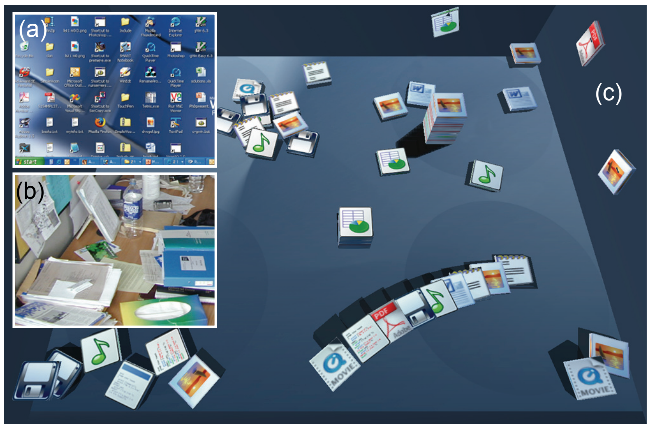

Figure 1. (a) Typical virtual desktop with structured organization. (b) Real desk, where items are casually organized. (c) Our BumpTop prototype with piles as the fundamental organizational object, and physics simulation affording casual, potentially more realistic interaction.

RELATED WORK

Malone's [24] early study of office worker organizational behaviors identified two basic paper organization strategies: 'piling' and 'filing'. It was also found that categorizing and filing items was cognitively difficult. It was noted that virtual desktops should provide untitled piles that support *deferred classification* as well as titled, logically arranged files. Further, it was postulated that 'electronic piles' should make the use of computers more natural.

Whittaker et al. [39] later compared the two strategies and found that piling offered a several advantages. Piling was lightweight, casual, involved less overhead, and was easier to maintain than filing. Piles served as visual reminders and increased availability of recent information. Pilers more frequently accessed their piles than filers accessed their file archives. Pilers archives were also smaller (60%), attributed to piled information being easier to discard. Filers reluctantly discard information due to the effort put into initially filing it. 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. On the other hand, pilling 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. Despite the advantages of piling, there remains little technological support for piling in today's GUI desktops.

The pile metaphor was explored in a prototype developed by Mander et al. [25] over a decade ago. Their prototype was based on a user-centered iterative design process. They introduced gestures and interaction techniques (sometimes modal) for browsing and manipulating piles and facilitating 'casual organization' on the desktop. We draw from and significantly expand upon their designs. In addition, some of their previous techniques were developed in isolation; we integrate them into a single, largely modeless interface.

Previous work has also looked at piles in different contexts. DiGioia et al. [14] 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'. DynaPad's [5] "open" pile representation laid out entire collections of photos side-by-side on a zoomable Pad++ [7] 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 infinite canvas. "Open" piles also aim to enhance remindability through visibility of all sub-objects, although this diminishes when the workspace is zoomed out and thumbnails become small. Our alternative stack representation is in the spirit of familiar real-world piles and does not require a zoomable interface. Our linearly ordered piles support fluid sorting and re-ordering in place without the need for additional tools.

In the forthcoming Microsoft Windows Vista (www.microsoft.com/windowsvista/) 'stacks' are provided as a way to visualize items by a specific attribute. For example, documents in a folder may be stacked by author. Here a pile representation is used only as a visualization aid and would benefit from the addition of our pile interaction techniques for browsing, manipulation and interaction with piled and un-piled documents.

Recent physically-inspired GUI designs such as Beaudouin-Lafon's work [6] rethinks windows as paper stacked in piles. Windows can be freeform peeled like real pieces of paper with a robust algorithm, which we use in our prototype. Peeling and re-orientation allows viewing of occluded windows below. Fold'n'Drop [15] used the above peeling technique and triggered it with a crossing-based gesture. Denoue et al. [13] use real-time simulated cloth texture-mapped as fliers and pinned up to virtual bulletin board that blow in the wind. Tossing as a window moving technique can be seen in a collection of work including Yatani et al. [41], DynaWall [36], and nVidia's nView extensions.

Previous work has also shown the benefits of a spatially based organization. The Data Mountain [31] leverages spatial memory in organizing webpage thumbnails on a perspective 2½D plane and showed improved user performance against text-based bookmarks. Amazingly, even after having not seen their organizations for four months their retrieval times were not significantly slower [12]. The TaskGallery [32] used a 3D art gallery metaphor for arrangement of application windows and was shown to help with task management and was enjoyable to use. It also demonstrated effective transfer of spatial memory and cognition into 3D.

Recent investigations into pen-based computing [1, 2, 21] have broken away from traditional point-and-click interfaces to techniques that are easier accomplished with the pen such as goal crossing [1]. We add to this set of pen-centric research by synthesizing crossing with other techniques such drag and drop, lasso selection and widgets. We also experiment with crossing targets that appear based on context.

In short, 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 [9, 10, 27]. Integration of physics and the pen also 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.

DESIGN GOALS

Early in our exploration, we identified several goals that would influence our interaction and visualization designs:

Realistic Feel. Objects on the physically enhanced desktop should move realistically, allowing users to leverage their knowledge of how things move in the real world. 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.

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 example, we can turn off physics to prevent unwanted collisions between explicitly organized items.

Tangible, Paper-like Icons. Our goal is to make documents *feel* like tangible physical objects. Sellen et al. [34] identified that the beneficial physical properties of paper include being thin, light, porous, opaque and flexible. Further they observed that these properties afford different human actions such as grasping, carrying, manipulating and folding. By embodying icons with added physical properties such as those of paper, we believe that users might be empowered to organize their virtual desktops in more casual, subtle and expressive ways as they do in their real workspaces.

Optimize for Pen Interaction. Our work focuses on designing for the new generation of pen-based computers, where no keyboard is available for triggering interactions. We strive for fluid interaction including favoring crossing [1, 2] when applicable and exploiting the pressure sensing capabilities of the pen [30], while avoiding designs that are problematic for pen interaction such as small targets, and double clicking.

Discoverable, Learnable Interface. After learning a small set of initial 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 leverage existing discoverable techniques that foster smooth transitions from novice to expert behavior such as Marking Menus [23]. 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 [11, 18] for every visual change in data representation. It is well established [4, 19, 33, 40] that 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.

BUMPTOP PROTOTYPE OVERVIEW

The BumpTop prototype presents users with a perspective 2½D view onto a planar desktop surface tilted 25 degrees with respect to the camera (Figure 1c). 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 [20].

Motion is constrained to the desktop by the walls enclosing it. The wall corners provide a 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 [31], 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.

Documents are represented by icons whose 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 its type. Being able to discern information from icon edges supports a pile browsing behavior that occurs in the real world called *edge browsing* [25]. Also, non-zero depth is necessary for the bounding volumes used in the collision detection. We do not place textual labels on the icons as our focus is on the physics, pile and pen interaction issues. However, techniques for dynamic labeling of items in a 3D space while avoiding occlusion exist in the literature [8].

It is important to note 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 2). However, we focus 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.

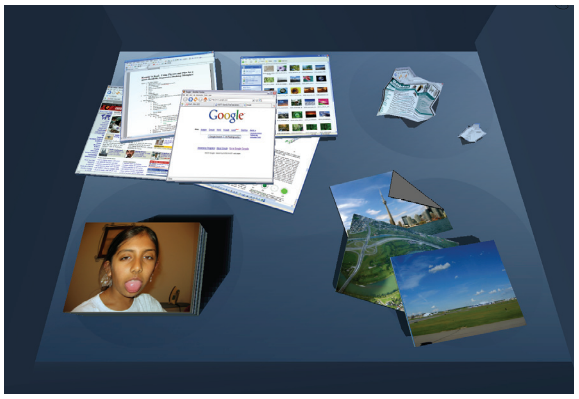
Figure 2. Our techniques apply to arbitrarily sized objects. Here we see a pile of photos (bottom left) and casually arranged (top left) and crumpled up (top right) windows.

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 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.

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.

LassoMenu

Much of the interaction in the system is triggered by a technique we call the LassoMenu that combines selection, command invocation, and parameter adjustment in one fluid stroke (Figure 3). This is a design alternative to the pigtail and handle techniques for triggering combined selection and action described by Hinckley et al. [21]. Users select items in the typical lasso fashion of drawing a path that encloses them. Once the lasso stroke has begun a semi-transparent blue circle is placed at the beginning of the lasso stroke. If the stroke is closed by the pen re-entering the blue circle users are presented with a control menu [29], a marking menu [23] 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 parameter.

The LassoMenu avoids the pigtail gesture that some users found difficult and was less preferred than the handle technique in Hinckley et al.'s study [21]. In addition, the LassoMenu is arguably more fluid than the handle technique which interrupts the stroke by requiring the pen to be lifted for the marking menu to appear. 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 can, if desired, be used smoothly in combination with existing techniques like the handle and pigtail [21]. Using the pigtail with the LassoMenu allows for command invocation without closing the lasso stroke.

Object Movement

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 [3]. 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

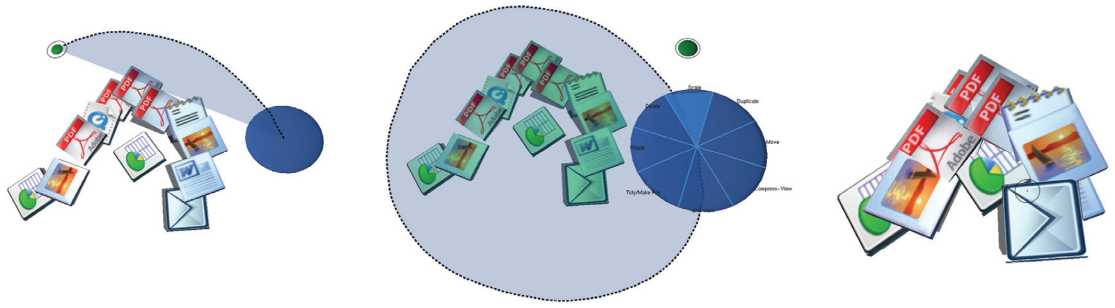

Figure 3. LassoMenu. (left) Lasso selection phase. (center) Lasso completed, pen reaches blue circle and enters control menu. Resize command selected. (right) Remainder of pen movement adjusts resize parameter.

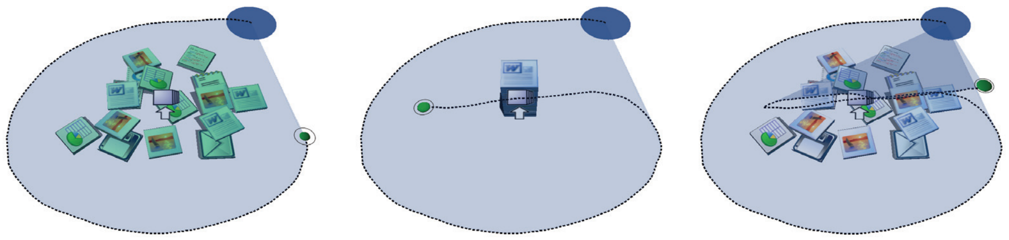
Figure 4. Lasso'n'Cross technique for pile creation. (left) Documents lasso selected. (center) 'Create Pile' icon crossed and pile created. (right) Undo by "un-crossing".

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 [16, 28] though new pen designs which move the button further up on the pen minimize this. 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.

Pile Creation

Lasso'n'Cross Technique

To more explicitly organize a group of objects we can create a pile out of them. Piles are created by lassoing around a group of objects, then crossing the 'create pile' icon that appears at their centroid. We call this technique *Lasso'n'Cross* (Figure 4). This novel technique allows users to fluidly select and pile objects in one stroke. 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 "un-crossing the initial cross" since the stroke is undone by making it backwards.

Lasso'n'Cross is an improvement over similar pen-based gestures combining selection and a single action such as the delete gesture in GEdit [22] 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.

By using the convex hull of the lasso stroke to indicate selected items (shown in dark blue in Figure 4) we avoid unwanted changes to the selection from the stroke portion that approaches and crosses the Lasso'n'Cross 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 [37] to determine if a stream of user input points is a straight line:

lasso arc length / distance between lasso endpoints > 1.2

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 heights and stacked into a Tidy pile (Figure 5d). 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 5). These options are ranges on a continuum of stroke

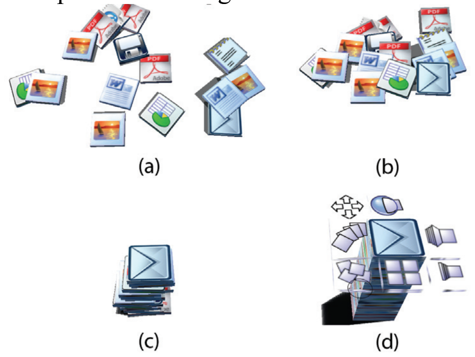

Figure 5. (a) Casually laid out documents. (b) Tidied unpiled documents. (c) Documents made into a *Messy Pile.* (d) A *Tidy Pile* with widgets revealed. Widgets in clockwise order from center-top: Fisheye, Leafer, Compression-Browse, Grid, Messy/Tidy, Fan out, Move.

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. Visual feedback during the scrub is provided by icons that appear at the range edges. That is, the points at which a messy pile or tidy pile will be created.

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' [25]. Instead of arbitrarily displacing items in the pile to achieve a messy appearance, we incorporate meaningful spatial information from the unpiled state.

Pile Manipulation

Supporting Pile Browsing with Widgets

When the pen hovers over a pile, pile widgets (Figure 5d) are revealed allowing the user to trigger various browsing techniques of the pile's contents (Figure 6). We draw on previous work on browsing data for our different pile layouts [9, 10, 27]. In several cases, our techniques are designed explicitly to support real-world pile browsing behaviour observed in office workers by Mander et al. [25]. The *Fan-Out* widget (Figure 6f) 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. [25]. Leafing through pile contents much like one flips through pages of a book is accomplished by scrubbing the *Leafer* widget (Figure 6b). The Compression-Browse widget (Figure 6c) compresses items on one axis to reveal the items underneath, without moving items. The standard grid layout is also offered (Figure 6e). Larger piles benefit from a fisheye view (Figure 6a), implemented via the Elastic Presentation Framework library [9]. The *Messy/Tidy* widget (Figure 6d) is like 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.

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 6e) and can be collapsed back to its original state with a tap.

Hovering over a widget for some time presents a tooltip with a more detailed description of the widget's functionality. Widgets also act as crossing targets for our novel *Drag'n'Cross* technique for precise insertion of objects into a pile, as described later. All transitions between browsing styles are animated smoothly as per our *Smooth Transitions* design goal.

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 items. Such casual browsing of unpiled objects is triggered via the LassoMenu.

Regional Visual Search

If we want to find a piled object but do not remember which pile it is in, we can use the browsing widgets to try and find it. However, for a large number 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 invoked with the LassoMenu, piles are smoothly exploded into a grid view on hover. Moving the pen away collapses piles back to their original state. This technique is similar to Sonnet et al.'s [35] work using an interactive explosion probe to browse a 3D model while maintaining context. Exploding Piles exploits the rough spatial memory a user might 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.

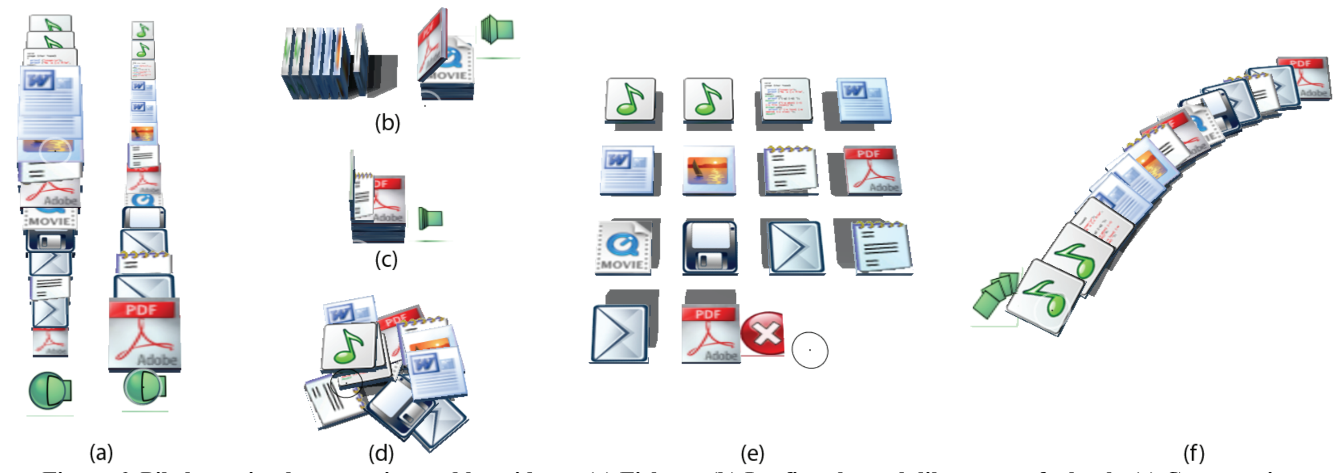

Figure 6. Pile browsing layouts triggered by widgets. (a) Fisheye. (b) Leafing through like pages of a book. (c) Compression-Browsing higher items to view items below. (d) Interpolating between Messy and Tidy positions. (e) Grid browse, locked down for further manipulation with PressureLock. (f) Fan out on user drawn path.

Pressure Cursor and Pressure Lock Technique

When users push very hard with the pen and reach the maximum pressure level, it acts as a trigger dubbed *PressureLock* which is used, for example, to lock a pile down into a specific browsing layout or pinning objects to the wall. Pushing the pen hard on the screen surface for pinning evokes similar actions in the real world.

To provide continuous visual feedback for the PressureLock technique we use a circular pressure cursor [30] with an inner circle that increases in size with the current pressure level (Figure 7). 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 to the Glimpse idea [17] of providing a pressure-based confirmation of a preview, with the pen-up before maximum pressure is reached being equivalent to an undo.

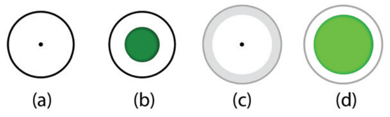

Figure 7. Pressure cursor. (a) Normal pressure cursor with no pressure. (b) with 75% of maximum pressure. (c) Pen is in a position where PressureLock will trigger additional functionality. (d) PressureLock with 100% pressure.

Adding to a Pile

In the real world one simply drops objects onto the top of a pile. Similarly, 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 previously [25]. 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 8). Scrubbing the pen along the side of the pile varies the insertion point. To avoid dwell which interrupts user flow, in respect of our *Optimize for Pen Interaction* design goal, we have developed a more pen-centric interaction technique called *Drag'n'Cross* (Figure 9). 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.

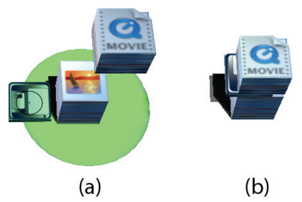

Figure 8. Drag and drop insertion of object into pile. (a) Drag to pile. (b) After insertion

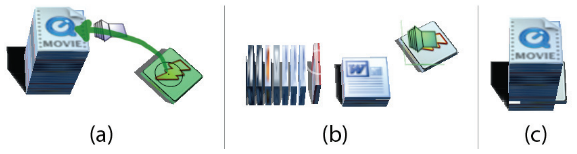

Figure 9. Drag'n'Cross technique for precise insertion. (a) User drags document and crosses Leafer widget, pen path shown by green arrow. (b) Scrub to specify insertion point. (c) Pen is released and document inserted

Hierarchical Piles

We experiment with blending elements of the two paper processing strategies: piling and hierarchical filing. In our hybrid technique, users can merge any combination of piles and objects into a new pile, using the same techniques employed to create a 'flat' pile out of just objects: LassoMenu or Lasso'n'Cross. 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.

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 1). We support similar functionality. Objects in a locked down pile can be re-oriented from the LassoMenu. Alternatively, groups of items can be re-positioned so they stick out (Figure 10). 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 is too large or cumbersome.

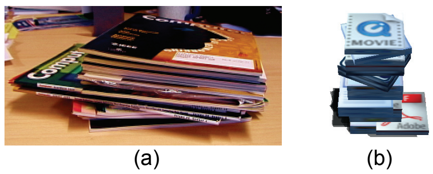

Figure 10. Pile with items rotated and pulled out for emphasis. (a) In the real world (b) In our prototype.

Enhancing and Judiciously Using Realism

Whittaker et al. [39] observed that frequently accessed items moved to the top of their piles, leaving less relevant material at the bottom due to repeated re-referencing. We facilitate this by supporting easy removal and casual addition to the top of piles via tossing.

Giving Icons Affordances of Paper

Sellen et al [34] identified that the physical properties of paper include being thin, light, porous, opaque and flexible. Further, these properties afford different human actions including folding, crumpling, creasing, and pinning.

BumpTop supports several manipulations that physical paper affords, 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 work [6] (Figure 2). To escalate a document to even greater importance we can pin it up to a wall (Figure 1c) by using PressureLock to create a springy joint. To remove it we simply pull the document off the wall. Another emphasizing technique we explore is locking the rotation of objects to make them standup on end (Figure 11), 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 emphasization 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 [24] for increasing the size of an icon or putting it in a special screen location to increase its priority and ability to remind.

To de-emphasize items we can crumple them up (Figure 2). 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

The physical simulation can sometimes prove disruptive, and we occasionally disable it, in line with our design goal of *Disable Physics as Necessary*. 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". 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 11). 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 work [6].

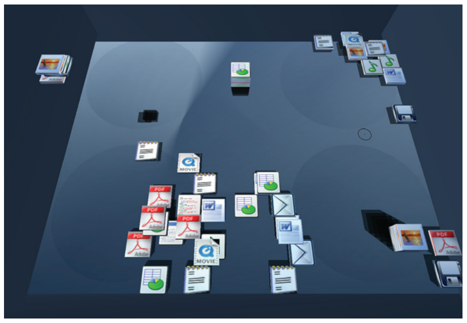

Figure 11. Axis alignment to enforce a tidier appearance. The 'shelf' (left) was made by pinning up a rotation-locked item.

IMPLEMENTATION

BumpTop runs in real-time on a Toshiba M200 TabletPC with a 1.6 Ghz CPU, 1 GB RAM and a GeForce FX Go 5200 graphics card. The prototype is written with C++, OpenGL and GLUT. Rigid body dynamics and collision detection are provided by the NovodeX Physics SDK (www.novodex.com). Some desktop item icons are by David Vignoni from the Nuvola icon theme.

INITIAL USER EVALUATION

To evaluate our designs we conducted a qualitative user study. Six participants (2 female, 4 male with computer skills ranging from novice to pen-interface experts) participated in videotaped think-aloud sessions lasting an hour each. This consisted of a 3 min introduction, 10 min "discovery" period where users explored the system, followed by instruction on the remaining functionality they didn't discover. Finally 29 tasks were performed that could be completed using multiple strategies. Post-study written and verbal questionnaires were also completed.

The results are encouraging. Participants were able to discover functionality on their own and became comfortable and proficient accomplishing most tasks. Questionnaire responses confirmed that: techniques were easy to learn (4.7/5), users were able to accomplish what they trying to do (4.4/5), users liked the software (4.7/5), and software felt familiar (4.5/5). 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.

The evaluation also revealed some issues with our interface that are applicable to future designs of pen-based interfaces. Some techniques deteriorated at the screen boundaries. Crossable widgets were awkward at the screen borders and not always compatible with other interaction techniques. Crossable objects could accidentally be triggered by sloppy lasso selections and prevent crossing targets from being dragged. In one case, this violated user's expectations of physicality, since our crossable pile widgets result in users not being able to directly drag a pile or its sub-items as they might expect. However, crossing was preferred by many users who felt it was "smoother" than clicking and contributed to the realism of some techniques like the Fanout or Leafer widgets. Widgets that revealed themselves based on pen position led to accidental invocations since they appeared unexpectedly when the pen was lifted from and returned to the trackable area. Drag'n'Cross also suffered from false invocations and precise insertion was found troublesome by 3 of 6 participants.

For single items the LassoMenu was inefficient. Also, the LassoMenu was often occluded by the user's hand, an inherent problem in most pen-based direct input systems. Specifying a parameter was difficult at the screen edges. These problems could be mitigated by using crossing based selections, click invoked marking menus on single items, and tear-off FaST sliders [26].

Lasso'n'Cross was the preferred technique for pile creation. Most users quickly became comfortable with it and several stated that creating and browsing piles with Grid, Leafer or Fanout were amongst their favourite interactions. Unfortunately the discoverability of Lasso'n'Cross was hindered by users not noticing the crossable monochrome icon that smoothly faded in. While participants stated they enjoyed the smoothness of the interface, this was perhaps an over-application of our *Smooth Transitions* design goal.

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 10 min "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 mentioned in the introduction. In addition, some participants emphatically stated that if they were given the tasks again they could effortlessly complete them.

Realistic Feel. During the discovery period and idle time between tasks users were seen playfully tossing or re-arranging items or watching the results of collisions. This playfulness also translated to users becoming proficient at arranging items with subtle and precise movements. For example, within minutes one 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 behavior suggests successful leveraging of real world knowledge of movement. Users said 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". Tossing was preferred to dragging for insertion by 4 of 6 users. Pressure based techniques were learned and used with little error.

Smooth Transitions. Users liked the smoothness of the interface but, as in the example discussed above, this can be detrimental in cases where actions need to be noticeable.

DISCUSSION AND CONCLUSIONS

One of the most common suggestions from those who have tried our prototype is that we should make more extensive 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 modeling objects as sheets of paper that can be folded in interesting ways to convey information or draped over other objects.

Future work includes a longitudinal usability study to verify the interface's validity in practice, as well as improvements to the scalability of the techniques. Like the GUI desktop, our prototype runs into problems when the number of items gets large. As Whittaker et al. [39] found, "the main limitation of [piling] was that it did not scale well: pilers found difficulties accessing information once piles had begun to multiply". We intend to explore extensions that might deviate somewhat from the physical piling metaphor but benefit from leveraging the underlying computer.

In summary, we have presented and evaluated interaction and visualization techniques that explore the use of piles as the primary organizational entity for desktop objects. 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.

ACKNOWLEDGMENTS

We thank Joe Laszlo, Gonzalo Ramos, Daniel Vogel, Azam Khan, and members of the Dynamic Graphics Project lab.

REFERENCES

1. Accot, J. & Zhai, S. (2002). More than dotting the i's - foundations for crossing-based interfaces. *CHI*. p. 73-80.
2. Apitz, G. & Guimbretière, F. (2004). CrossY: a crossing-based drawing application. *UIST*. p. 3-12.
3. Baraff, D., Witkin, A., & Kass, M. (1997). An introduction to physically based modeling: particle system dynamics. *SIGGRAPH Course Notes*.
4. Bartram, L. (1997). Can motion increase user interface bandwidth. *IEEE Systems, Man and Cybernetics*. p. 1686-1692.

5. Bauer, D., Fastrez, P., & Hollan, J. (2004). Computationally-enriched "piles" for managing digital photo collections. *IEEE VLHCC*. p. 193-195.
6. Beaudouin-Lafon, M. (2001). Novel interaction techniques for overlapping windows. *UIST*. p. 152-154.
7. Bederson, B. & Hollan, J. (1994). Pad++: a zooming graphical interface for exploring alternate interface physics. *UIST*. p. 17-26.
8. Bell, B., Feiner, S., & Höllerer, T. (2001). View management for virtual and augmented reality. *UIST*. p. 101-110.
9. Carpendale, M.S.T. & Montagnese, C.A. (2001). A framework for unifying presentation space. *UIST*. p. 61-70.
10. Carpendale, S., Cowperthwaite, D., Tigges, M., Fall, A., & Fracchia, D. (1999). The Tardis: A visual exploration environment for landscape dynamics. *SPIE Conf. on Visual Data Exploration & Analysis VI*. p. 110-119.
11. Chang, B. & Ungar, D. (1993). Animation: From cartoons to the user interface. *UIST*. p. 45-55.
12. Czerwinski, M., van Dantzich, M., Robertson, G., & Hoffmann, H. (1999). The contribution of thumbnail image, mouse-over text, and spatial location memory to web page retrival in 3D. *Interact*. p. 163-170.
13. Denoue, L., Nelson, L., & Churchill, E. (2003). A fast, interactive 3D paper-flier metaphor for digital bulletin boards. *UIST*. p. 169-172.
14. DiGioia, P. & Dourish, P. (2005). Social navigation as a model for usable security. *ACM SOUPS*. p. 101-108.
15. Dragicevic, P. (2004). Combining crossing-based and paper-based interaction paradigms for dragging and dropping between overlapping windows. *UIST*. p. 193-196.
16. Fitzmaurice, G., Khan, A., Pieke, R., Buxton, B., & Kurtenbach, G. (2003). Tracking menus. *UIST*. p. 71-79.
17. Forlines, C. & Shen, C. (2005). Glimpse: a novel input model for multi-level devices. *CHI Ext. Abs.* p. 1375-1378.
18. Gonzalez, C. (1996). Does animation in user interfaces improve decision making. *CHI*. p. 27-34.
19. Grossman, T., Balakrishnan, R., Kurtenbach, G., Fitzmaurice, G., Khan, A., & Buxton, B. (2001). Interaction techniques for 3D modeling on large displays. *ACM Symp. on Int. 3D Graphics*. p. 17-23.
20. Herndon, K., Zeleznik, R., Robbins, D., Conner, B., Snibbe, S., & van Dam, A. (1992). Interactive shadows. *UIST*. p. 1-6.
21. Hinckley, K., Baudisch, P., Ramos, G., & Guimbretiere, F. (2005). Design and analysis of delimiters for selection-action pen gesture phrases in Scriboli. *CHI*. p. 451-460.
22. Kurtenbach, G. & Buxton, W. (1991). Issues in combining marking and direct manipulation techniques. *UIST*. p. 137-144.
23. Kurtenbach, G. & Buxton, W. (1993). The limits of expert performance using hierarchical marking menus. *CHI*. p. 35-42.
24. Malone, T. (1983). How do people organize their desks?: Implications for the design of office information systems. *ACM Trans. on Info. Systems*, *1(1)*. p. 99-112.
25. Mander, R., Salomon, G., & Wong, Y. (1992). A "pile" metaphor for supporting casual organization of information. *CHI*. p. 260-269.
26. McGuffin, M., Burtnyk, N., & Kurtenbach, G. (2002). FaST Sliders: Integrating marking menus and the adjustment of continuous values. *Graphics Interface*. p. 35-42.
27. McGuffin, M., Tancau, L., & Balakrishnan, R. (2003). Using deformations for browsing volumetric data. *IEEE Visualization*. p. 401-408.
28. Miller, L. (2005). Case study of customer input for a successful product. *Agile2005*.
29. Pook, S., Lecolinet, E., Vaysseix, G., & Barillot, E. (2000). Control menus: Execution and control in a single interactor. *CHI Ext. Abs*. p. 263-264.
30. Ramos, G., Boulos, M., & Balakrishnan, R. (2004). Pressure widgets. *CHI*. p. 487-494.
31. Robertson, G., Czerwinski, M., Larson, K., Robbins, D., Thiel, D., & van Dantzich, M. (1998). Data mountain: Using spatial memory for document management. *UIST*. p. 153-162.
32. Robertson, G., van Dantzich, M.., Robbins, D., Czerwinski, M., Hinckley, K., Risden, K., Gorokhovsky, V., & Thiel, D. (2000). The Task Gallery: A 3D Window Manager. *CHI*. p. 494-501.
33. Robertson, G., Mackinlay, J., & Card, S. (1991). Cone trees: Animated 3D visualization of hierarchical information. *CHI*. p. 189-194.
34. Sellen, A. & Harper, R. (2003). The myth of the paperless office: MIT Press, Cambridge, MA.
35. Sonnet, H., Carpendale, S., & Strothotte, T. (2004). Integrating expanding annotations with a 3D explosion probe. *AVI*. p. 63-70.
36. Streitz, N., Geißler, J., Holmer, T., Konomi, S.i., Müller-Tomfelde, C., Reischl, W., Rexroth, P., Seitz, P., & Steinmetz, R. (1999). i-LAND: an interactive landscape for creativity and innovation. *CHI*. p. 120-127.
37. Thorne, M., Burke, D., & van de Panne, M. (2004). Motion doodles: an interface for sketching character motion. *ACM Trns. on Graphics*, *23(3)*. p. 424-431.
38. Tristram, C. (2001). The next computer interface. *MIT Technology Review*, *December*.
39. Whittaker, S. & Hirschberg, J. (2001). The character, value, and management of personal paper archives. *ACM Trans on CHI*, *8(2)*. p. 150-170.
40. Woods, D. (1984). Visual momentum: a concept to improve the cognitive coupling of person and computer. *Intl. Journal of Man-Machine Studies*, *21*. p. 229-244.
41. Yatani, K., Tamura, K., Hiroki, K., Sugimoto, M., & Hasizume, H. (2005). Toss-it: intuitive information transfer techniques for mobile devices. *CHI Ext. Abs*. p. 1881-1884.