DeskJockey: Extending peripheral digital information into the physical workspace

by

Ryder Ziola

Submitted in partial fulfillment of the
requirements for the degree of
Bachelor of Computer Science, Honours

at

Dalhousie University
Halifax, Nova Scotia
May, 2006

DALHOUSIE UNIVERSITY

FACULTY OF COMPUTER SCIENCE

The undersigned hereby certify that they have read and recommend to the Faculty of Graduate Studies for acceptance a thesis entitled "**DeskJockey: Extending peripheral digital information into the physical workspace**" by **Ryder Ziola** in partial fulfillment of the requirements for the degree of **Bachelor of Computer Science, Honours**.

Dated: April 7, 2006

Supervisor: ______________________________
Dr. Kori Inkpen

Reader: ______________________________
Dr. Stephen Brooks

DALHOUSIE UNIVERSITY

Date: **April 7, 2006**

Author: **Ryder Ziola**

Title: **DeskJockey: Extending peripheral digital information into the physical workspace**

Department: **Computer Science**

Degree: **B.C.Sc. (Honours)** Convocation: **May** Year: **2006**

Table of Contents

List of Figures

Abstract

This thesis describes the design of DeskJockey, a system that aims to extend a users workspace by projecting peripheral and ambient information on physical surfaces in the environment. We implemented a prototype, using a world-in-miniature metaphor to easily locate and manipulate content from the extended workspace (i.e. the projected display). Informal feedback of this system showed potential for this type of interaction, particularly to support monitoring information.

Acknowledgements

I would like to thank Dr. Kori Inkpen for all the guidance and help she has provided; the members of the EdgeLab for their advice, specifically Ritchie Argue for his technical direction; and my family and Vicki for their support.

Chapter 1

Introduction

1.1 Motivation

Principally, enhancements to the display capabilities of the single-user single-monitor workstation are achieved through the addition of more monitors, increasing the available screen size and resolution. As multi-monitor configurations have become increasingly prevalent, it has become evident that, as more monitors are added, their benefits are characterised by diminishing returns. Study of the use of extra monitors [8] has revealed that a large part of their use is for peripherally monitoring information. Peripheral monitoring benefits from the ability to put information in a fixed, dedicated location where it can be seen and referenced easily. Given that continuity of screen spaces is not important to this goal, the opportunity to make these dedicated locations resident on a different medium is appealing.

1.2 Goal

Toward this end, we have created the DeskJockey system (Figure 1.1) that uses a tabletop projected display to house peripheral and ambient information in support of a traditional desktop environment. The ability to use the table to casually display digital information would enhance both the abilities of the computer and those of the table. The steadily decreasing price of projectors makes it feasible to accomplish this without a large investment in infrastructure.

In this process however, it is important that the desk not lose the qualities that define it. Currently, the division between desk and computer monitor is defined by whether the content is digital or physical. We intend to redefine this as the division between the active and the passive. The goal is not to turn the desk into an active display; a myriad of properties — resolution, orientation, lack of privacy — would make it a poor substitute for a monitor for the majority of tasks. Its passive nature

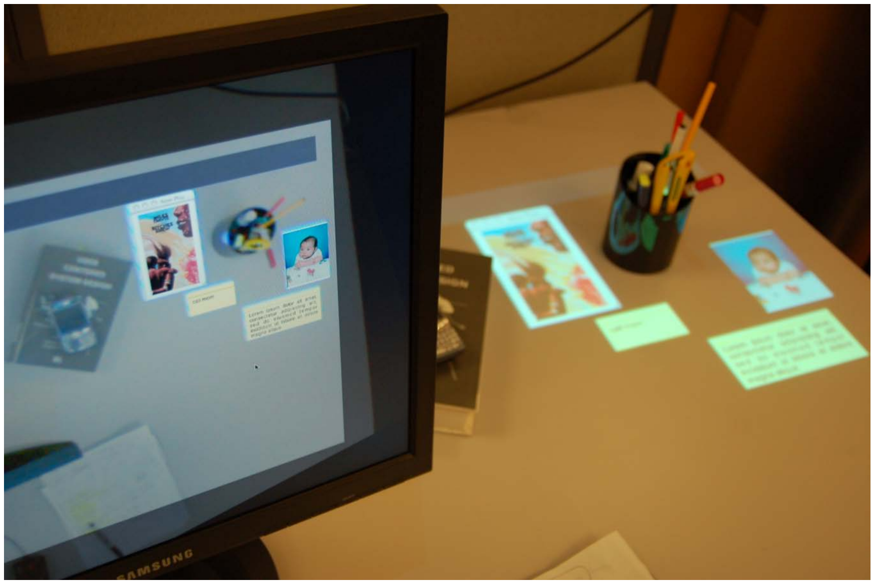


Figure 1.1: The DeskJockey system

is a strength, and makes it the natural place to display passive digital information.

The goal of this research is to design a table which can house real paper next to digital calendar information, digital sticky notes next to the (physical) telephone, pictures of the kids that are not necessarily static, while keeping the users subtly informed about their world with ambient information. While explicitly not making an attempt at augmented reality, this system would at least allow the virtual to bump up against the physical more casually.

Ambient information is one area where the system could have a strong use. Rather than requiring a dedicated display or specifically designed embodied devices, by supplying a large amount of peripheral pixels, ambient information can be presented in a casual, organic way. Exterior daylight or network activity, for example can be tracked implicitly in physical spaces that are already available.

The single-user windowed workstation is an entrenched reality. The momentum of this reality hinders the adoption of more task-specific, innovative interaction methods. Conversely, this also means that everyone has a staggering amount of computing power, information, and connectivity under their desk. DeskJockey attempts to easily allow this information to reside naturally in the wider workspace, enhancing the enjoyment and productivity of the user.

1.3 Outline

Chapter 2 explores the relevant work that has been done that will aid in determining the capabilities such a system would need to possess to support peripheral and ambient information. In Chapter 3, the design of a system as well as several possible interaction techniques are outlined. Chapter 4 details the implementation and the specifics of our interaction technique. The results of our informal evaluation are presented in Chapter 5. Chapter 6 covers conclusions that can be drawn and future work in this direction.

Chapter 2

Related Work

2.1 Multi-Monitor Use

Grudin's study of the usage patterns in multi-monitor environments offers useful insight [8]. By interviewing users of multi-monitor configurations at their workplaces, Grudin examined how they were dividing tasks among their monitors. Without fail, the logical organisation of the monitors – how the cursor travels between the screens – matched the physical organisation of the monitors. It was found that each monitor was treated as a discrete place to view output, rather than being used simply as an extension of existing space. The bezel that surrounds most monitors physically separates two adjacent monitors and prevents them from being used contiguously and treated as one display. Most users considered one monitor, often the larger one, to be primary and the others to be secondary. The primary monitor was used for the majority of interaction, and the main task occupying the user would be placed on it. Though used often as direct support for the primary task, a secondary monitor was also used as a fixed location for monitoring live information channels. For this purpose, the value of a secondary monitor is in its ability to provide "instant access to a resource in a known location in peripheral vision"([8], p.461), rather than providing additional space for interacting with and supporting the primary task. Examples of tasks which fall into this category include monitoring an e-mail in-box, a to-do list, calendar, or contact list.

A more quantitative study of multi-monitor usage was performed by Hutchings *et al.* [9]. The window visibility and management activities of single-monitor and multi-monitor users on their own workstations were logged and analysed. They found that single-monitor users have less unused screen space, but not significantly fewer visible windows than multi-monitor users. The study compared e-mail activity and visibility on the various configurations. Amongst the single-monitor users, the e-mail program was being actively used 90.0% of the time it was visible. For multi-monitor configurations, e-mail was only active 44.6% of the time it was visible. When it is visible without being active, we can assume it is being used for information monitoring purposes.

2.2 Desks

The area of enhanced and/or digital desks is an active one. Wellner's DigitalDesk [17] was an early attempt to replace the metaphorical desktop with a literal, physical desk. Paper documents placed on the desk were augmented with digital information by being projected upon. A vision-based system allowed for direct input with video-based finger-tracking. Cameras captured an image of the document being worked on and optical character recognition algorithms were run to digitise any text on the document. This made the desk more like a workstation by endowing plain paper with computer-like functions, such as copy and paste, or by allowing paper to exhibit complex behaviour, like acting as a calculator.

InteractiveDesk [1] aimed to endow physical objects with the properties of digital storage. A pen-based projected display was integrated with a standard upright display workstation, giving one the option of using either the keyboard and mouse to interact with the upright display or the pen to interact with the table. Digital files are associated with physical objects, allowing the user to retrieve digital files by presenting the physical object they are connected to. For example, putting a physical folder on the table would trigger the associated digital files to be opened as well.

As one example of the "Tangible Bits" approach [10], the metaDESK [16] intended to control digital information with physical objects. The "Tangible Bits" philosophy was critical of previous desks for extending the GUI's limitations to the real world, rather than bring the real world's strengths to the digital world. In response, the metaDESK has a tangible user interface consisting of physically instantiated interface elements that can be manipulated directly. It was used to implement "Tangible Geospace", a large geographical map with an arm-mounted LCD panel to act as a "active lens", providing close-up views of the content on the desk.

Research that has been performed into the affordances offered by digital desks provides insight into the limitations of the medium. Elliot and Hearst [6] had architecture students perform sketching and image-sorting tasks using a monitor and mouse, a tablet-sized screen with a pen, and a large digital desk with a pen. Despite the advantage offered by the digital desk — an abundance of display space to use for forming piles — the participants preferred to perform the image-sorting task on the monitor using a mouse. Moreover, the desk produced less refined sketches and was less popular than the tablet. Participants felt that the desk surface was too large to use and found it difficult to interact with its peripheral areas.

2.3 Peripheral Information

Information can be considered peripheral if it is not necessary for the current task, but may be of interest to the user nonetheless. Information monitoring tasks form a large subset of peripheral activities and centre on the problem of notification: how to minimise distraction while informing the user of potentially important or interesting changes in the system's state. This category includes tasks such as being notified of incoming e-mails, being aware of an upcoming meeting, or seeing which friends are logged into their instant messaging application. As we are interested in providing support for peripheral information monitoring, the mechanisms necessary for performing it should be examined.

Investigations into the requirements for a successful peripheral interface focus on the goal of notifying the user of a change in the information being monitored without allowing it to hinder the current task. Maglio & Campbell [12] performed three studies to compare various methods of peripherally presenting text, including a number of variations on a scrolling ticker, fading text, and visual and accompanying auditory feedback. They had participants perform a text-editing task while peripheral information was intermittently presented and tested retention of the peripheral information. They found that continuous motion, in the form of a scrolling ticker, was both detrimental to performance of the primary task and unreliable at notifying users of the peripheral information. Discrete motion, active only when new information is presented, was found to be most effective. It should be noted, however, that the peripheral information was displayed on the same monitor as the primary task, limiting our ability to extrapolate to a situation involving discrete, physically separated displays.

Exploring this idea further, Bartram *et al.* [2] studied the ability of *Moticons* —

icons with simple motion — to notify users of changes. A variety of experiments were performed to discern the effects of motion, colour, shape and distance on the ability of an icon to attract a user's attention. It was found that motion is detected better than changes in colour or shape, especially in the periphery of the user's vision, and is well suited to notification. The time required to detect motion in the periphery is affected by the level of engagement with the primary activity. As a result of the study, a set of guidelines were developed for motion-based notification techniques. In general, slow linear motion was found to elicit the best response times, while not distracting the user. Conversely, icons that zoom toward the user or travel large distances were found to distract and annoy. Most significant for our purposes, it was found that a repeating motion with an amplitude as small as 1° of the visual angle is still reliably detectable in peripheral vision.

2.4 Ambient Information

A subset of peripheral information, ambient information can be identified by its intended subtlety. It is meant to implicitly communicate information not necessarily relevant to the task at hand through environmental cues and is not intended as a means of explicit notification. The ambientROOM [11] is another product of the Tangible Bits project discussed above. It aims to create an environment that interfaces with digital information, attempting to engage the human ability to process background information. The current activity-level of a loved-one is displayed as ripples projected on the ceiling of the room. The volume of a background soundtrack of nature sounds can roughly signal quantities (such as incoming e-mail). The hands of the clock in the room can be moved, shifting the room's displays to the corresponding past or future state. These fully-fledged ambient environments require dedicated physical objects or displays and naturally come with a high infrastructure requirements.

As we are interested in supporting ambient uses, the design requirements for ambient information prove useful. Heuristics developed by Mankoff *et al.* [14] for the evaluation of ambient displays help to inform us of the affordances a system must offer to be compatible with ambient use. Two ambient displays were constructed, for simulating daylight and the tracking the bus-schedule. Experts in the field evaluated the ambient displays to develop a list of known design issues. A list of potential heuristics were compiled based on Nielsen & Molich's user-interface heuristics [15]. The inapplicable heuristics were removed and new, ambient-specific ones were added. Using these heuristics, non-experts were told to evaluate the ambient displays. Based on their ability to address the known design issues, heuristics were selected. Of the heuristics they developed, two stand out as being relevant specifically to the design of a system capable of supporting ambient uses, rather than the design of the ambient application itself:

- *perpipherality of display:* It can be easily monitored while being unobtrusive.
- *easy transition to more in-depth information:* It is necessary to be able to easily engage the system, converting the interaction from passive to active.

A system intended for use in part as an ambient display should accommodate these properties.

Chapter 3

DeskJockey

3.1 Overview

To support both peripheral and ambient information dispersed throughout the workspace, we developed DeskJockey. The workspace retains a typical desktop workstation with an upright monitor, keyboard and mouse, where the majority of work is performed. Secondary displays are projected on the surface of tables and walls in the workspace. Digital objects can be projected on these displays, providing quick reference to information and a richer environment. Potential digital objects could include nearly anything, though information monitoring and notification tasks would be best suited to the large spaces and fixed locations. The workstation controls projectors mounted overhead, allowing digital objects, generated by the workstation, to be displayed on any surface in the workspace (see Figure 3.1). This makes every surface a potential repository for digital information, with virtual objects placed amongst physical objects. For organic integration with the physical desk, it is important that the objects appear to be objects on the table, rather than objects on a computer generated display. This requires masking the inactive parts of the projected screen. For our purposes, this can be accomplished simply by projecting a black background.

3.2 Input Methods

In an ideal world, every surface in our workspace would be capable of acting as both an input and an output device. The ubiquity of projectors has made the latter possible; any surface is now capable of acting as a display. The promise of ubiquitous input, though, is still unfulfilled.

While a variety of tabletop input methods exist, they come with a high overhead in cost and complexity. The goal of this project is not to provide a fully functional tabletop, but rather to provide a simple method to augment the existing environment.

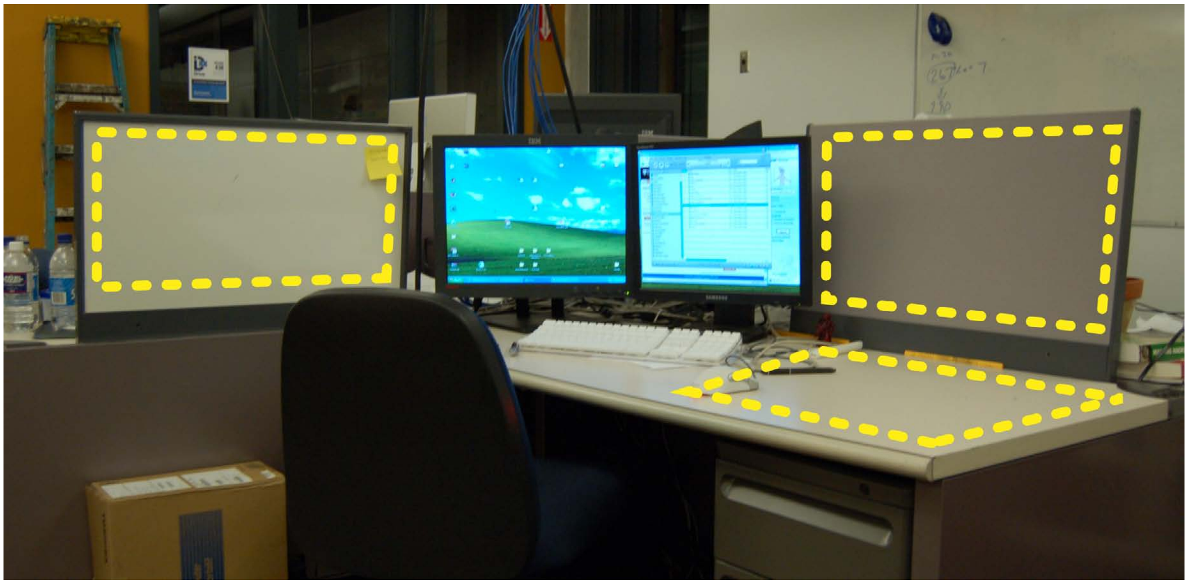


Figure 3.1: In an ideal situation, all of a workspace's physical surfaces could display passive digital information.

For this reason, the decision was made to rely on no source of input aside from the keyboard and mouse associated with the primary display.

3.3 Proposed Interaction Methods

Because the projected display surface itself is passive, it was necessary to develop an interaction technique to manage the projected surfaces. The content displayed on the projected surface is an extension of the user's workspace therefore, it would make sense that the interaction mechanism be a natural extension of the windowed GUI. In addition, the majority of monitoring tasks serve as a gateway into an active task (for example, an instant messaging contact list serves as a gateway to a chat), therefore it is necessary to accommodate a passive desk as the starting point for an active task. Ideally, the active task would take place on the primary display, which is better suited to prolonged detailed interaction, however, there are a host of problems inherent in switching from a vertical to a horizontal surface mid-task. Toward this end, four methods of interacting with the virtual desk objects were proposed and explored, employing varying degrees of abstraction with respect to the interaction with the table.

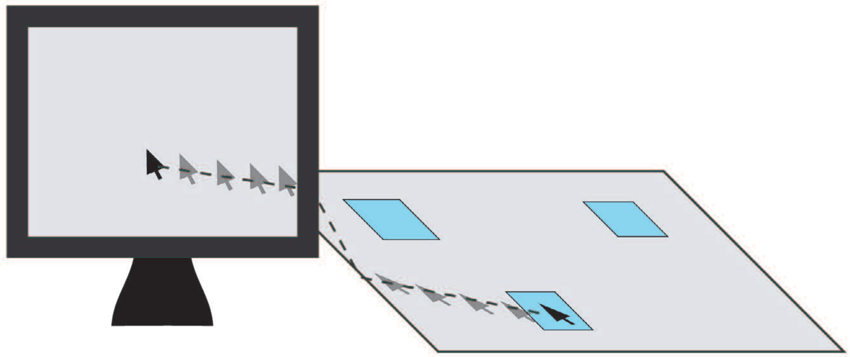


Figure 3.2: The Direct Traversal Technique

3.3.1 Direct Traversal

The first — and most naïve — technique proposed is to simply treat the additional projected surfaces as an extension of the workstation's primary display (see Figure 3.2). In the fashion that a traditional multi-monitor configuration extends the system's desktop across multiple geometrically contiguous monitors, our projectors could be mapped into this space and treated like any other monitor.

Some of the more obvious difficulties with this idea could be addressed by introducing some basic enhancements to the system's understanding of the layout of the monitors and the projectors. Specifically, the change of orientation, lack of geographical contiguity, and varying pixel sizes could lead to poor target acquisition and frustration. This could be mitigated by the introduction of a technique similar to the Mouse Ether [3] approach. By allowing for the existence of interstitial space between monitors, the cursor's movement as it changes screens could be made to correspond more intuitively with the real geographical layout of the projectors and the monitors. Similarly, a rotation in the cursor's orientation could be added to the projected surfaces to ensure that the movement between screens follows expectations.

3.3.2 Cursor Jump

The second technique proposed is the cursor jump (see Figure 3.3), inspired by Hrvoje Benko and Steven Feiner's "Multi-Monitor Mouse" [4], which allowed a cursor to jump between displays by pressing a mouse button or key-combination. Similarly, in our use of this technique, either a mouse-button combination (for example, the side-button on

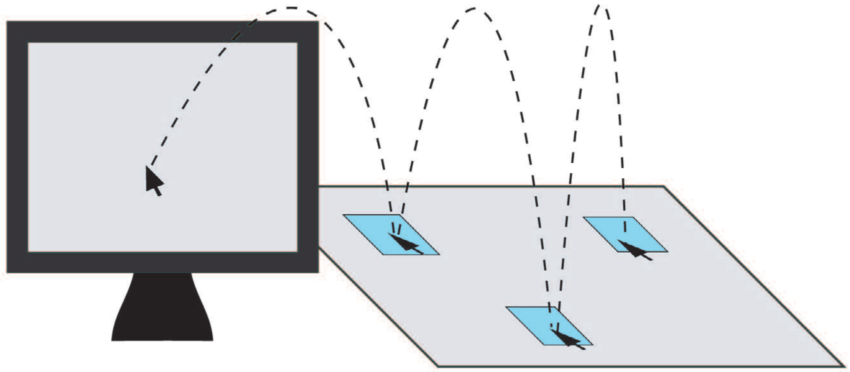


Figure 3.3: The Cursor Jump Technique

a 5-button mouse) or a key-combination jumps the cursor between the various objects on the table in a logical order, accompanied by some attention-getting visual effect to help locate the destination of the jump. The objects can then be manipulated directly.

As the projected spaces are not contiguous with the workstation's space, adding items to the desk would be more complicated than under the direct traversal technique. In order to add an object, while dragging an object, the mouse-button or key combination is pushed. The object being dragged at the time jumps to the desk, where it can be positioned. Naturally, as the table's objects are being iterated to reach the desired object, this solution does not scale well as the number of objects on the table increases.

3.3.3 The Swift Return Technique

The third technique proposed is the Swift Return technique (Figure 3.4). Reminiscent of Mac OS X's Exposé feature [7], navigating objects on the desk occurs on the primary display rather than on the desk. The system is invoked by pressing a hot-key or moving the cursor over a hotspot — a corner of the screen, for example. Upon invocation, all of the items on the table are zoomed from the table to the primary screen in a graphically self-explanatory manner. All of the objects are displayed on the primary screen and scaled (if necessary) so they can be viewed simultaneously. Once on the screen, the desired object is selected. All objects then return to the table, with the cursor following the selected object. The object can then be manipulated

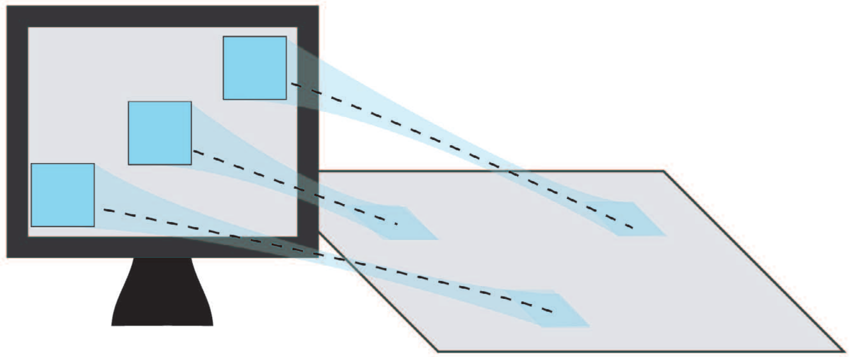


Figure 3.4: The Swift Return Technique

directly. As with the cursor jump, the disjoint spaces require a special mechanism for adding an object to the desk. This is accomplished by dragging an object to the hotspot, at which point it is sent to an arbitrary location on the desk. The cursor moves to the desk with the object, allowing the user to position it.

3.3.4 World-in-Miniature

In the final proposed technique, the user performs navigation and manipulation on the primary display rather than the desk (see Figure 3.5). Upon triggering the system (again through hot-key or cursor hotspot), the user is presented with a view representing the table's projectable area. Video of the table is used to provide information about the real objects on the desk, making it easier to position objects in the world-in-miniature. This view displays a small, scaled representation of all the objects that are on the desk. Objects can be added to the desk, organised and removed from the desk by manipulating the representative icons. Any object can also be selected for detailed inspection and manipulation.

In contrast to the Swift Return method described above, the icons representing the items on the desk are meant to be comprehensive stand-ins for the real objects on the desk, rather than simply a short-cut to the desk's objects. Moving an icon results in similar movement of the real object and dragging the icon off the representation of the desk results in the object it represents returning to the primary display. Naturally, the use of a camera adds to the complexity and expense of the system.

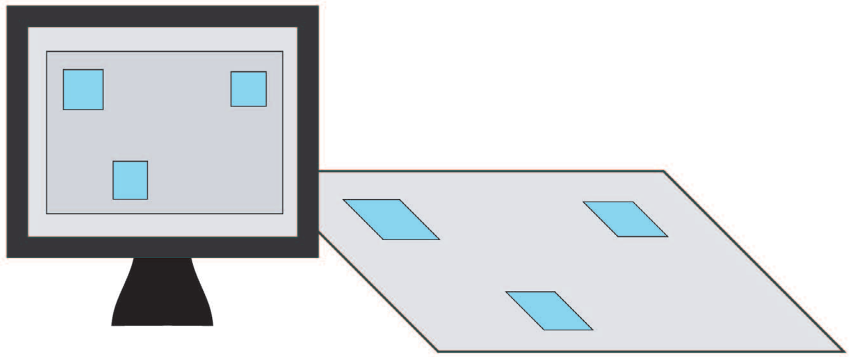


Figure 3.5: The World-in-Miniature Technique

3.4 Decision Criteria

Three main criteria were used to choose which of the above interaction techniques to develop for the initial prototype of this system:

- The technique should provide quick access to the table and its contents. It should take no more than a few discrete actions to uniquely identify the item you wish to manipulate.
- The technique should provide an easy and intuitive method for moving an object from the primary display to the table and back off the table when it is no longer needed. Manipulation of the object should also be easy to perform.
- As the table is designed to provide peripheral support, methods for interacting with it should not intrude on the regular operation of the primary screen any more than is minimally necessary. Without deliberately triggering the system, use of the primary display should be unaffected by the presence of the system. Similarly, it should be difficult to accidentally trigger the system.

3.5 Decision

- The World-in-Miniature and Swift Return methods can provide access to any object on the table with two actions: one to trigger the system, the second to select the desired object. The Cursor Jump method clearly fails this criterion if there are more than a small number of objects on the table. The Direct

Traversal method requires only one action, but the distance and change in orientation from the primary display to the table would make the acquisition of a target difficult.

- The world-in-miniature metaphor supports additions and removals quite naturally. Adding an object to the on-screen metaphorical table implies adding it to the real table. Repositioning an object on the table is performed by manipulating the virtual representation of the object in the world-in-miniature. The Swift Return method, by comparison, lacks a robust metaphorical representation of an object on the table, necessarily making the method of addition and removal more arbitrary.

- Direct Traversal would create the greatest intrusion on the regular use of the primary display. Overshooting targets, such as scroll-bars, on the edge of the screen, would result in the cursor moving mysteriously to the table and being lost, a known problem in multi-monitor configurations [13]. By this criterion, there is no real difference between the other three methods, as they all require the use of a hotspot, keyboard or mouse combination.

Interpreted in the light of these criteria, the decision was made to implement the world-in-miniature technique.

Chapter 4

Implementation

The world-in-miniature design for DeskJockey was elaborated upon and implemented. DeskJockey was written for Mac OS X in a combination of C and Objective-C.

4.1 Physical Setup

The current physical set-up of DeskJockey (see Figure 4.1) is comprised of a single monitor workstation with a single projector. The workstation has a single 19" monitor with 1280 x 960 resolution on a large 100cm x 150cm table. The projector is mounted above the table, projecting to the right of the keyboard. The projector is treated by the operating system as a display, allowing us to use the operating system's windowing system to manage our content and to use any windowed application as on object on our table. The projector supports a native resolution of 1024 x 768, but was increased to 1280 x 1024 to prevent objects on the table from being considerably larger than they appear on the screen. The projected display was given a black background, resulting in the objects on the table appearing cropped, with nothing being projected on the unused parts of the table.

4.2 Triggering the System

Manipulation of virtual items on the physical desk occurs through the DeskJockey world-in-miniature system. The world-in-miniature can be viewed by moving the cursor to a hotspot that extends across the bottom of the primary display. When the cursor is moved down, off the bottom of the screen, a transition to a view of the world-in-miniature is triggered. The bottom of the display was chosen because the table sits below the bottom of the monitor and therefore moving "down" off the monitor onto the desk suggests itself as a natural mapping. To minimise accidental invocations, there is a speed threshold that must be satisfied. If the cursor reaches

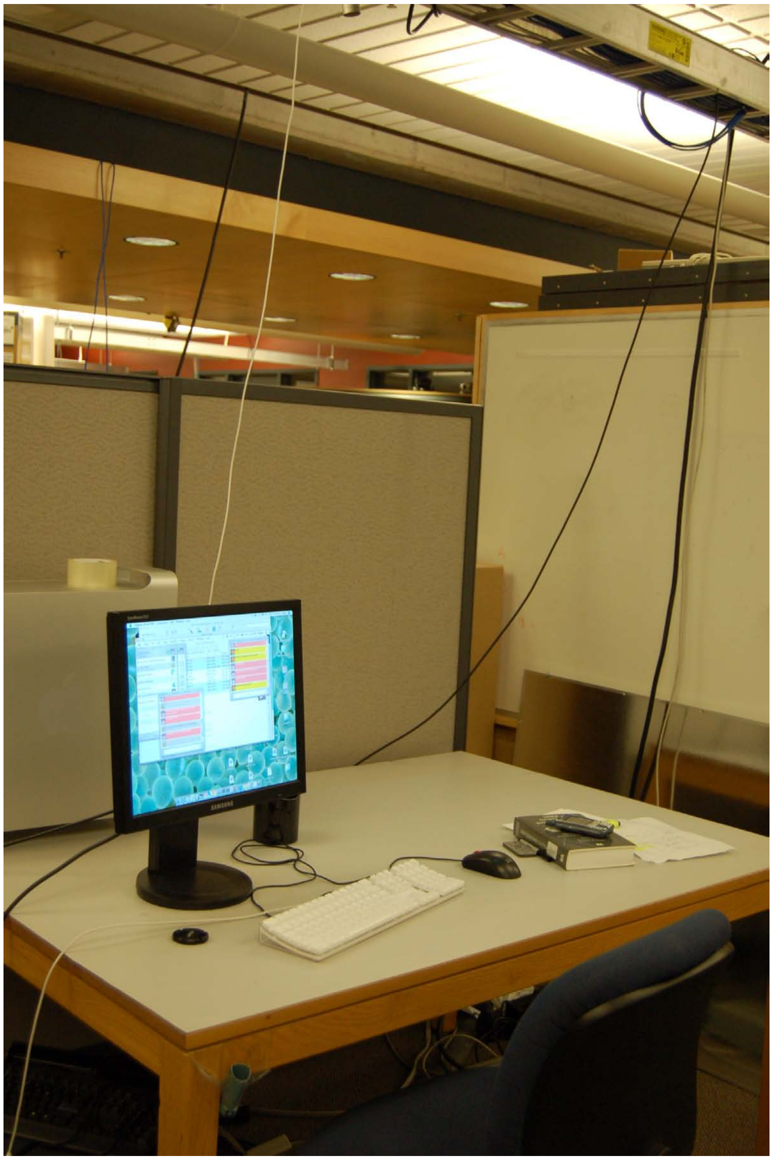


Figure 4.1: The current physical set-up of DeskJockey employs a projector and camera mounted above a large table.

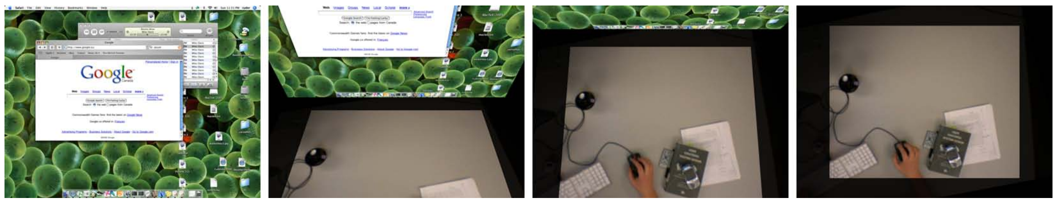


Figure 4.2: The visual transition to the world-in-miniature.

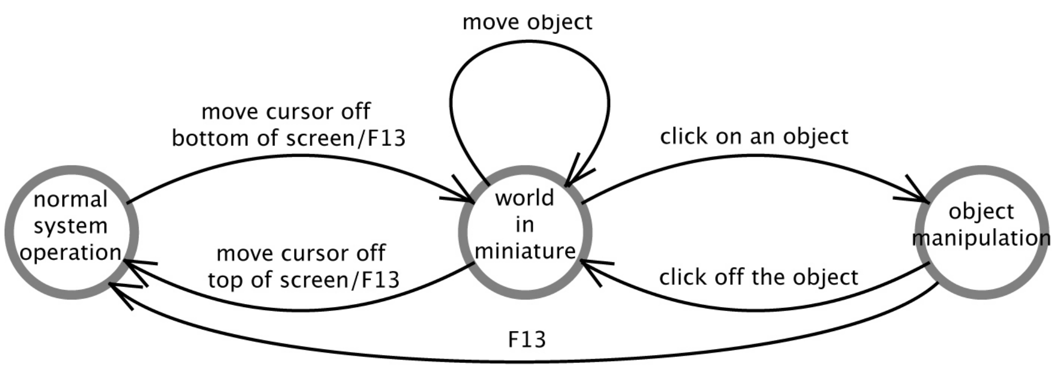


Figure 4.3: The progression from normal operation to the world-in-miniature to interaction.

the bottom of the screen, but is not moving with a speed greater than the threshold, it is not triggered and the workstation functions normally. After iterative testing, a Δy value of 15 pixels per mouse-event was chosen. Alternately, a hotkey (F13) can be pressed to achieve the same effect. Figure 4.3 shows the state transition diagram for triggering and interacting with the world-in-miniature system.

Upon triggering the system, a visual transition occurs from the monitor's current contents to a representation of the table (see Figure 4.2). In imitation of the physical set-up, the screen simulates a 90° rotation effect, moving the screens current contents up and off the screen, replacing it with a view in miniature of the objects that are on the table. Conversely, once viewing the representation of the table, moving off the top of the screen or pressing F13 again triggers the reverse transition and the workstation returns to its normal operation. The transition was created using the Quartz Composer visual programming language.

Figure 4.4: The World-in-Miniature offers a real-time video view of the table.

4.3 View of the Table

After transitioning to the world-in-miniature, the user is presented with a representational view of the table as shown in Figure 4.4. The view is centred on the projectable area of the table, in the same orientation as the projector. Naturally, with multiple projectors this would have to be expanded to accommodate their multiple orientations. Areas outside the projector's range are given semitransparent shading to indicate that items placed in this area won't be displayed to the table.

One consequence of working with metaphors rather than the objects themselves is the loss of context provided by the real table. Without a mechanism for directly seeing how the metaphorical table corresponds to the real table, orientation is impossible to divine and avoiding collisions with physical artifacts on the table becomes difficult. This problem is prevented through the addition of a simple web-cam. Real-time video of the table provides the background for the representation of the table and virtual objects can be placed with respect to real objects. The camera we used provided 640 x 480 VGA resolution.

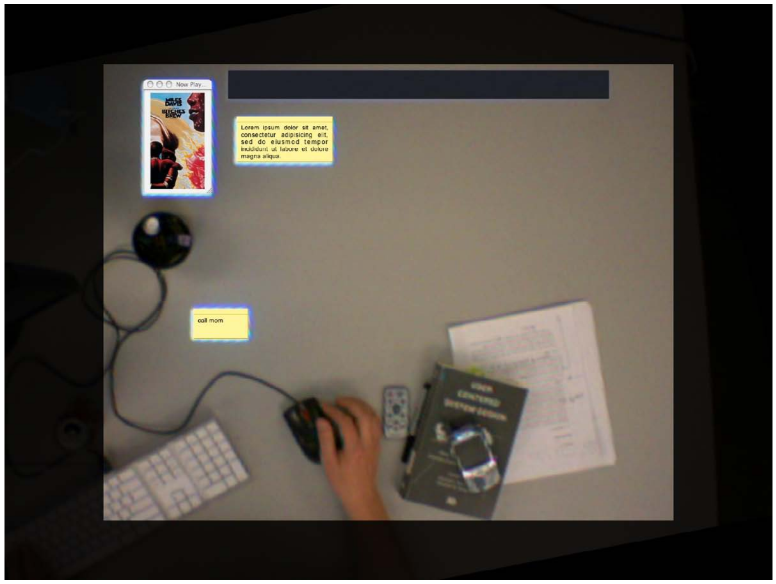


Figure 4.5: An object on the table is represented by its screen-shot.

4.4 View of the Objects

Each object on the table is represented by an icon in the world-in-miniature (see Figure 4.5). In an attempt to make the metaphor as robust as possible, icons are shown as a screenshot of the object they represent, scaled to correct proportions. The icons can be dragged on the world-in-miniature to move the virtual object on the table. To aid in making a correspondence between the icons and the objects they represent, when the cursor moves over an icon, it glows yellow, along with the object on the table. Any window in the operating system can be added to the table by dragging the window down, to the hotspot on the bottom of the screen, adding the object to the centre of the table. Similarly, an icon is removed from the table and returned to the primary display by dragging it up, off the table.

The image for the icon is updated on a timer, copied from the window server's buffered bitmap of the window *before* the window server composites the windows to the screen. This allows us to keep the icon of each window updating in near-real-time, regardless of whether the window is obscured or even if it has been moved partly off the operating system's display.

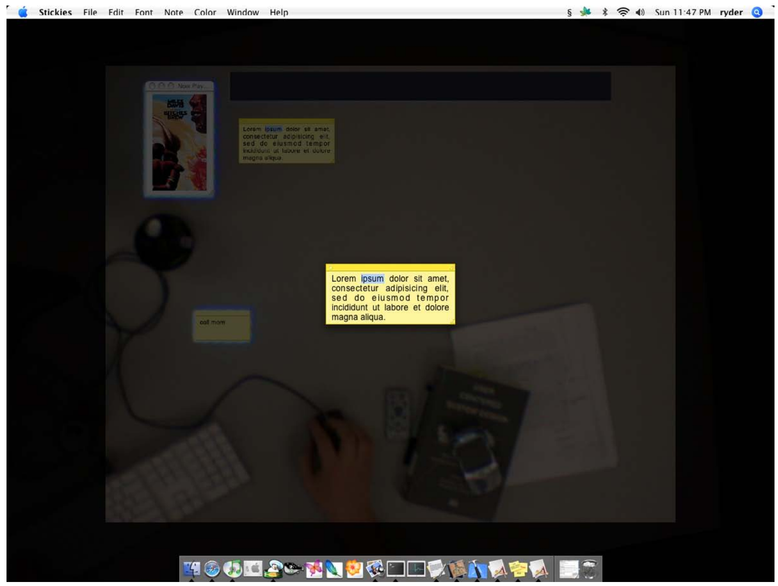


Figure 4.6: An object on the table is brought to the primary display for manipulation.

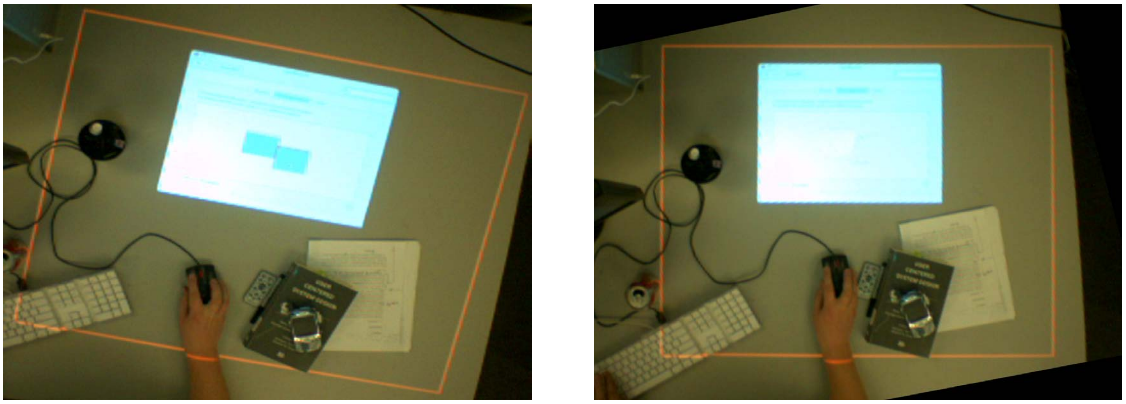
4.5 Working with the Objects

Although the main purpose of the system is for passive information monitoring, interaction with items on the table is nonetheless a requirement. Selecting an object on the screen then interacting with it on the table is awkward and requires a modal shift. To avoid this, the decision was made to make all interaction take place on the screen. When the user clicks on an icon in the world-in-miniature, the object it represents is moved to the primary display for interaction (see Figure 4.6). The representation of the table is shaded and the object is brought to the foreground, where it can be interacted with directly. To return to the world-in-miniature, the user clicks anywhere off of the window or presses escape, and the object is returned to the table and the world-in-miniature returns to the foreground of the primary display.

4.6 Some Technical Issues

4.6.1 Video Calibration

In the on-screen representation of the table, the icons are being drawn over the video of the table, which of course contains the projected objects themselves. This required that the icons be aligned perfectly with the video beneath them. There are too



(a) Video Before Correction (b) Video After Correction

Figure 4.7: (a) the video before calibration (b) after calibration, the video is aligned with the world-in-miniature.

many factors — camera, projector, and table orientation — to achieve the precision necessary purely through mechanical alignment. To compensate for these effects, a perspective transform was applied to the incoming video (see Figure 4.7).

If the physical set-up has changed, the system requires a brief calibration to be run. The user is presented with raw video from the camera. Sequentially, each corner of the projector's display area is hilighted with a green dot. The user identifies the dot in the video of the projection. This allows us to build a 3 x 3 matrix that maps the projector's four corners to the corresponding four dots on the video. This matrix is inverted to map from the video's co-ordinate system to the projector's and applied to the raw video as a perspective transform, correcting for rotation, scaling, shearing, and perspective distortions.

Once calibrated, the video is transformed in real-time. The icons can be drawn over the resulting video and are aligned with remarkable precision. Ideally, the process of calibration could be improved to be performed automatically using a vision system rather than relying on the user to recognise the projected area in the video.

4.6.2 Moving Windows

There was initially some difficulty giving DeskJockey the ability to control all of the operating system's windows. The window server, responsible for positioning all windows on the system, restricts a process to manipulating the windows it has created. Only one process, the Dock, is permitted to manipulate arbitrary windows. This problem was circumvented by injecting our code into the Dock's process space and having it control windows on our behalf, to create and move objects on the table.

4.7 Supplemental Applications

Little has been said so far about the actual objects being placed on the desk. By designing DeskJockey to work with any window in the operating system, it leaves this question deliberately unaddressed. That said, some applications are better suited to being placed on a table than others. In addition, title bars and widgets are not very useful and go against the spirit of DeskJockey. Toward this end, several applications were selected, modified, or created to illustrate specific uses.

4.7.1 Stickies

A simple program for placing Post-It style notes on your desktop. With its simple notes designed for passive viewing, it is ideal for our purposes.

4.7.2 Pictures

A small program was written for displaying a image or pdf without a border or titlebar, at an arbitrary size. It serves as a picture frame or view of a document.

4.7.3 Clutter

Clutter is an open-source program for displaying cover art for the .mp3 currently playing [5]. As well, it allows you to make casual piles of album covers that can be selected to play the album. This allows us to have the cover art for our current song sitting on the table and to have digital albums stacked and scattered on the table much like their physical counterparts. Minor changes were made to the GUI to suit our needs.

4.7.4 Widget Viewer

A generic wrapper was written to display a Mac OS X Dashboard widget as a window that can be placed on the table. Many of these small programs are designed for purposes similar to ours, to monitor information or status. Relevant examples include

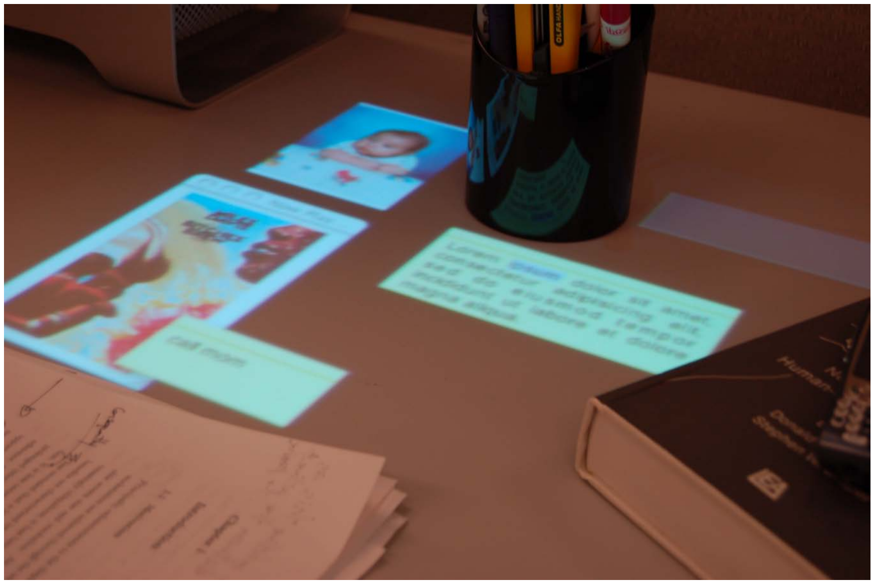


Figure 4.8: Supplemental applications, clockwise from the top: Pictures, Sky, Stickies, and Clutter.

a calendar widget, a weather widget, and a widget that represents new emails as flowers in a vase.

4.7.5 Sky

A daylight monitor was created as a small example of the ambient information potential of DeskJockey. It loads an image from a local outdoor webcam, calculates the average colour of the sky and displays a large block of that colour with the texture of the sky from the camera. During the day, it roughly communicates if the sky is blue or grey and becomes dark as the sun sets.

Chapter 5

Informal Evaluation

Following the implementation of the system, preliminary, informal demonstrations were held to solicit feedback from several members of the Edgelab. In this process, they were introduced to the system and shown examples of its use with the supplemental applications discussed above. After being allowed to become acquainted with it, their impressions were sought. Specifically, the following issues were of the greatest interest to us:

- whether a projected peripheral display would be compatible with the physical workspaces they currently use;
- whether they perform enough monitoring tasks (or other tasks compatible with a semi-interactive display) to find such a set-up useful and if DeskJockey would accommodate their monitoring tasks;
- how they felt about interacting with objects on the primary display, rather than on the desk;
- if they found the world-in-miniature approach valuable, especially in comparison with a direct-manipulation approach, which they were given the opportunity to experience.

5.1 Feedback

Most people commented that the primary obstacle to using DeskJockey in their actual workspaces was the clutter they have on their desks. This was mitigated somewhat by the presence of the video background, allowing the placement of virtual objects to be informed by the physical clutter. Some foresaw no problems projecting on top of the clutter on their desks and said that they might flip over an offending piece of paper to get a white surface. One common response to the problem of clutter was to ask about the possibility of projecting on a wall or bulletin-board in their workspace rather than on the desk. Others suggested a combination of the table and the wall, as was originally envisioned in the generalised system, with some elements — photographs, calendars, ambient displays — on the wall and other more active elements on the desk. Overall, there was enthusiasm for the possibility of adding digital content to areas in the workspace that are not necessarily part of the workstation's active monitors.

The participants interviewed generally said they performed multiple monitoring tasks that would be well suited to use with DeskJockey. Monitoring instant-messaging buddy lists was one example that was brought up as an example for the pattern of desired interaction. The desk would be useful for monitoring the online status of one's friends, while the interaction with the friends in the form of a chat would take place on the primary display. Several participants enjoyed the idea of being able to monitor information passively, to not be forced to switch tasks and interact with an application to respond to an alert.

The supplemental applications presented were specifically selected or designed to have minimal GUI widgets such as title-bars. This decision was well received, as the visual distinction between the standard system windows and objects on the table reinforced the idea that the table was augmented, rather than feeling (and consequently being treated) as a fully-capable tabletop display. There was concern that in general use, without the specifically chosen applications, this would be lost and the desk would seem like a crippled monitor rather than an augmented table.

Concerning the decision to use the primary display for interacting with the table's objects, the consensus was that it was a necessary decision and probably better than switching to the table to interact. Nonetheless, there was an interest in seeing if this assumption would be borne out by a study. Similarly, there was interest in seeing if users would employ DeskJockey for its intended use with passive information. If, in practice, this isn't true, many of the design decisions would need to be re-evaluated.

When given the opportunity to interact with objects on the table directly, by traversing the table with a cursor rather than using DeskJockey's world-in-miniature, most people reacted negatively. One participant commented that the objects near the back of the desk were harder to manipulate. Several others found that the act of moving between the primary monitor's vertical orientation and the table's horizontal orientation was disconcerting. Overall, the impression was that something was lost by making the desk an area for active work, like a display, rather than retaining the passive qualities of a real desk.

Many of the visual aspects of the implementation proved popular: the transition to and from the desk view, the real-time video in the background, and the live screen-shots of the desk's contents. The relatively low resolution of the projector was perceived as less of a problem than expected. One participant even enjoyed reading a digital document with relatively small text placed on the table.

One problem encountered by several participants was that the transition to and from the world-in-miniature was often triggered accidentally. When the cursor is moved off the bottom of the screen to transition to the world-in-miniature, the cursor is deliberately re-located to the top of the screen, maintaining its position relative to the desktop as it rotates off the top of the screen. Most of the accidental transitioning occurred at this time, due to the cursor being warped too near to the top of the screen following the transition to the world-in-miniature. This made it quite easy to transition back unintentionally by moving the cursor up only slightly. This could be avoided if the software positioned the cursor further from the top of the screen.

Many of the participants mentioned potential privacy concerns associated with displaying their content on a desk. Although none of them said it would prevent them outright from using DeskJockey, the consensus was, unsurprisingly, that they would treat the desk space as being more public than a primary display, where private information would likely remain. Further discussion with those interested in projecting on the walls around their desk revealed a continuum of perceived privacy, decreasing from monitor to desk to wall. In general, these privacy concern were equated with existing concerns about physical objects, such as paper, in the same spaces.

Chapter 6

Conclusions

We have designed DeskJockey, a system for augmenting a workspace by displaying peripheral and ambient information on the surfaces of the physical space. Using readily availible equipment and little infrastructure, we have employed the large size and relatively static layout of a table to address the tasks of peripheral monitoring and ambient awareness. Virtual objects — such as calendars, to-do lists, reminder notes or incoming e-mails — are projected alongside physical passive objects on the desk or walls.

DeskJockey was implemented using a single projector displaying content on a table. We developed the first iteration of an interaction technique for managing the new digital content. Based on a world-in-miniature metaphor, scaled views of the virtual objects on the desk are shown over video of the physical objects on the desk. Manipulations of the scaled views of the objects results in similar manipulation of the corresponding virtual object on the desk. Interaction with the objects takes place on the primary display, where prolonged interaction is most easily performed. Our implementation of DeskJockey was met with positive feedback in an informal evaluation.

DeskJockey represents a novel way of seeing a digital table, as a simple augmentation to a traditional workstation, rather than as the primary locus of interaction. The world-in-miniature technique employed by DeskJockey addresses the unique interface problems that arise from having screens that can display content, but are not suitable for interaction. Our implementation has demonstrated that it is feasible to enhance an environment without necessarily adding to the active workspace area.

6.1 Future Work

Before conducting any studies, there are several technical enhancements to DeskJockey to be made. The informal evaluation made it clear that some aspects of the system should be adjusted, such as the sensitivity of the hotspot. In addition, the system would benefit greatly from the the ability to arbitrarily scale and rotate objects on the table. Currently, the objects placed on the table are obliged to keep the orientation of the projector and have a scale determined by the resolution of the projector. The technical reasons for this limitation could be surmounted fairly easily, and the world-in-miniature could provide the ability to modify the display properties of an object. Similarly, several changes in the interaction technique would be necessary to support multiple projected surfaces, rather than the single projector on the table.

Future studies could then focus on evaluating and refining the interaction technique that has been developed. A longitudinal study of the system could establish if users generally find the system to be useful. More specifically, it would be interesting to see if users would be content with using the table space for passive tasks, rather than using the surface as an active table-top display.

Additionally, some questions were raised in the process that would be worthy of exploration. As mentioned above, our implementation assumed that it was less distracting to bring an object to the monitor from the table for interaction, rather than bring the cursor to the table from the monitor. The generalised question of the relative severity of disruptions caused by content displacement as opposed to disruptions caused by a switch in input mode would be interesting to pursue.

Bibliography

[1] Toshifumi Arai, Kimiyoshi Machii, Soshiro Kuzunuki, and Hiroshi Shojima. Interactivedesk: a computer-augmented desk which responds to operations on real objects. In *CHI '95: Conference companion on Human factors in computing systems*, pages 141–142, New York, NY, USA, 1995. ACM Press.

[2] Lyn Bartram, Colin Ware, and Tom Calvert. Moticons: detection, distraction and task. *Int. J. Hum.-Comput. Stud.*, 58(5):515–545, 2003.

[3] Patrick Baudisch, Edward Cutrell, Ken Hinckley, and Robert Gruen. Mouse ether: accelerating the acquisition of targets across multi-monitor displays. In *CHI '04: CHI '04 extended abstracts on Human factors in computing systems*, pages 1379–1382, New York, NY, USA, 2004. ACM Press.

[4] Hrvoje Benko and Steven Feiner. Multi-monitor mouse. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 1208–1211, New York, NY, USA, 2005. ACM Press.

[5] Clutter. http://http://www.sprote.com/clutter/.

[6] Ame Elliott and Marti A. Hearst. A comparison of the affordances of a digital desk and tablet for architectural image tasks. *Int. J. Hum.-Comput. Stud.*, 56(2):173–197, 2002.

[7] Exposé. http://www.apple.com/macosx/features/expose/.

[8] Jonathan Grudin. Partitioning digital worlds: focal and peripheral awareness in multiple monitor use. In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 458–465, New York, NY, USA, 2001. ACM Press.

[9] Dugald Ralph Hutchings, Greg Smith, Brian Meyers, Mary Czerwinski, and George Robertson. Display space usage and window management operation comparisons between single monitor and multiple monitor users. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 32–39, New York, NY, USA, 2004. ACM Press.

[10] Hiroshi Ishii and Brygg Ullmer. Tangible bits: towards seamless interfaces between people, bits and atoms. In *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 234–241, New York, NY, USA, 1997. ACM Press.

[11] Hiroshi Ishii, Craig Wisneski, Scott Brave, Andrew Dahley, Matt Gorbet, Brygg Ullmer, and Paul Yarin. ambientroom: integrating ambient media with architectural space. In *CHI '98: CHI 98 conference summary on Human factors in computing systems*, pages 173–174, New York, NY, USA, 1998. ACM Press.

[12] Paul P. Maglio and Christopher S. Campbell. Tradeoffs in displaying peripheral information. In *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 241–248, New York, NY, USA, 2000. ACM Press.

[13] Regan L. Mandryk, Malcolm E. Rodgers, and Kori M. Inkpen. Sticky widgets: pseudo-haptic widget enhancements for multi-monitor displays. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 1621–1624, New York, NY, USA, 2005. ACM Press.

[14] Jennifer Mankoff, Anind K. Dey, Gary Hsieh, Julie Kientz, Scott Lederer, and Morgan Ames. Heuristic evaluation of ambient displays. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 169–176, New York, NY, USA, 2003. ACM Press.

[15] Jakob Nielsen and Rolf Molich. Heuristic evaluation of user interfaces. In *CHI '90: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 249–256, New York, NY, USA, 1990. ACM Press.

[16] Brygg Ullmer and Hiroshi Ishii. The metadesk: models and prototypes for tangible user interfaces. In *UIST '97: Proceedings of the 10th annual ACM symposium on User interface software and technology*, pages 223–232, New York, NY, USA, 1997. ACM Press.

[17] Pierre Wellner. Interacting with paper on the digitaldesk. *Commun. ACM*, 36(7):87–96, 1993.