

Magic Desk: Bringing Multi-Touch Surfaces into Desktop Work

Xiaojun Bi^{1,2}, Tovi Grossman¹, Justin Matejka¹, George Fitzmaurice¹

¹Autodesk Research, Toronto, ON, Canada
 {firstname.lastname}@autodesk.com

²University of Toronto, Toronto, ON, Canada
 xiaojun@dgp.toronto.edu

ABSTRACT

Despite the prominence of multi-touch technologies, there has been little work investigating its integration into the desktop environment. Bringing multi-touch into desktop computing would give users an additional input channel to leverage, enriching the current interaction paradigm dominated by a mouse and keyboard. We provide two main contributions in this domain. First, we describe the results from a study we performed, which systematically evaluates the various potential regions within the traditional desktop configuration that could become multi-touch enabled. The study sheds light on good or bad regions for multi-touch, and also the type of input most appropriate for each of these regions. Second, guided by the results from our study, we explore the design space of multi-touch-integrated desktop experiences. A set of new interaction techniques are coherently integrated into a desktop prototype, called Magic Desk, demonstrating potential uses for multi-touch enabled desktop configurations.

Author Keywords

Multi-touch, TableTop, Desktop Work

ACM Classification Keywords

H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

General Terms

Human Factors, Design

INTRODUCTION

In recent years, multi-touch displays [8, 10, 37] have received a great deal of attention, both in the research community, and in consumer devices. The research literature has shown numerous benefits of multi-touch input, such as increasing the bandwidth of communication between human and computers [16] and its compatibility to control multiple degrees-of-freedom [23]. Because of its unique affordances, research in multi-touch applications generally involves a standalone touch sensitive device, sometimes with peripheral displays [36], with a custom designed UI optimized for touch [39]. Less explored is how multi-touch could be integrated into our current desktop experience.

Unfortunately, along with its advantages, touch input suf-

fers from certain known problems. Text entry is cumbersome [14], and the “fat finger” problem limits the precision of touch input [9]. With the existence of such challenges, it is hard to imagine that our mouse and keyboard devices, which provide precision input, could be completely replaced by multi-touch surfaces. Instead, we foresee that *future computing environments will be a blend of keyboards, mice and touch devices.*

With the release of Microsoft Windows 7, which supports multi-touch [24], and the commercial availability of multi-touch monitors [7] and laptop displays [6], the industry has already moved in this direction. But, this begs the question: is a vertical display monitor the right way to integrate multi-touch into the desktop experience? Other planar regions for touch input include the areas on the desk surrounding the mouse and keyboard. To integrate touch into the desktop experience successfully, it is crucial to understand the properties of different touch regions and their relationships with the devices we already use.

In this paper, we provide two main contributions, to advance our understanding of the integration of multi-touch and desktop configurations. First, we systematically investigate users’ single and multi-touch input abilities on the potential touch regions in a desktop computing environment, including the vertical display monitor. The vertical display performed poorly in both one- and two-hand touch tasks, showing that the main option commercially available today might in fact be the worst one. Second, guided by the study results, we explore the design space of multi-touch integrated desktop experiences, with the design and implementation of a set of interaction techniques, such as an enhanced task bar and multi-functional touch pad. All of the techniques were coherently integrated into a desktop prototype called Magic Desk (Figure 1).

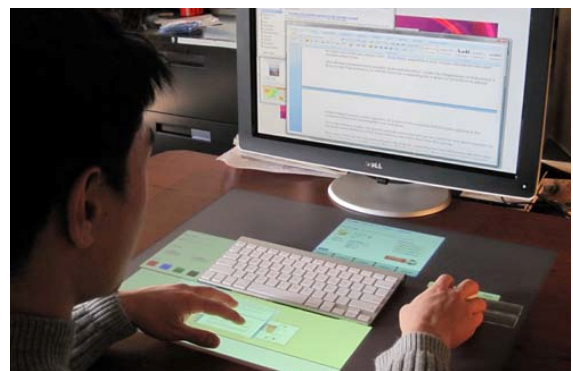


Figure 1. Working on the Magic Desk.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2011, May 7–12, 2011, Vancouver, BC, Canada.

Copyright 2011 ACM 978-1-4503-0267-8/11/05...\$10.00.

RELATED WORK**Ergonomic Studies of Physical Desktop Tables**

Constrained by the lengths of human arms and rotation angles of joints, a user's "reach" heavily impacts how a table can be used: it dictates the space available for interaction. Anthropometrical research has determined where the user is able to reach when sitting in front of a horizontal table [13]. Hedge [13] also proposed a model predicting maximum comfortable reach (i.e., the Zone of Comfortable Reach or ZCR) Scott et al. [29] further revealed different types of usage within the reachable area: space near the body was usually used for working while space further away was used for storage. In a desktop computing environment, the existence of mice and keyboards will probably affect the accessibilities of various touch regions. Specific studies investigating the interplay between keyboard/mouse and various touch regions are required.

Studies of Digital Tabletop Usage

Numerous researchers have explored how tabletop devices could facilitate daily desktop work. Morris et al.'s study [22] showed that though an additional stylus-enabled display provided extra screen real estate, shortcomings included complicating window management, and overhead related to input device switching. Wigdor et al. [35] reported on the experience of an individual who exclusively used a DiamondTouch table for office work. Hancock and Booth [11] investigated menu selection on horizontal and vertical display surfaces, leading to a menu placement strategy for a tabletop display. We complement these high-level studies by systematically assessing the human's interaction abilities in the different touchable regions.

Touch-enhanced Devices and Environments

We classify this research into the following three categories:

1) Using an interactive tabletop

A sizeable amount of research has explored interacting on a digital tabletop. Some important examples include Krueger et al. [17], who explored interacting with virtual objects using users' live images; the Digital Desk [15], which enabled users to interact with paper documents; the InteractiveDESK [2], which responded to users' operations on real objects (e.g., keyboards, digital pens) on a desktop to reduce workload; Rekimoto et al. [26, 27], who implemented an augmented surface allowing users to interchange digital information among various objects; and, Wu et al. [39], who studied using various multifinger and whole hand gestures. More recent work has looked at bringing physical simulation into touch-screen interaction [38]. Common among these foundational research projects is that they utilize the interactive tabletop as a standalone input platform, and do not consider the integration of multi-touch with a desktop configuration.

2) Touch enabled desktop devices

Recently, there has been substantial work augmenting common desktop devices with touch functionality. Multi-touch desktop screens are now commercially available and

the Window 7 OS supports multi-touch input. Block et al. [3] introduced the Touch-Display Keyboard, in which graphical output and input are extended across the keyboard's surface. The Mouse 2.0 [32] and the Apple Magic Mouse [1] allow users to perform multi-finger gestures on the surface of the mouse. Yang et al. [40] further augmented a mouse with an interactive touch sensitive display. While these explorations are all promising, the surfaces they provide for touch are non-planar and have limited space. In contrast, we will be exploring combinations of planar multi-touch surfaces with traditional input devices, to support traditional, larger-scale, multi-touch interactions.

3) Combinations of desktop devices and touch surfaces.

The Pebbles project [21] explored an enhanced PC computing environment with a touch-screen mobile device, but the device had limited input space and only provided single point input. The Bonfire system [15] enhanced mobile laptop interaction by projecting interactive displays on both sides of a laptop keyboard, however the implementations were focused more on augmenting the surrounding environment, rather than supporting desktop activities. Hartmann et al. explored a combination of mice and keyboard input with a multi-touch surface [12]. They focused on interactions among co-located groups around a large table. We focus on enhancing a single person's desktop work.

In summary, there has been a large amount of research on touch enhanced tables and environments, but we are unaware of any which systematically investigates the integration of multi-touch technologies into a desktop environment.

EXPERIMENT

In a traditional desktop environment, where a user sits in front of a desk and input is performed with a mouse and keyboard, the planar regions available for touch input include the entire space surrounding the keyboard and mouse. However, in terms of commercial availability, multi-touch is typically constrained to the vertical display monitor.

With little current understanding of the benefits or drawbacks of using the planar regions surrounding the keyboard, and how these regions compare to a multi-touch vertical monitor, we are motivated to investigate the effect that touch region has on interaction capabilities. In addition, we investigate the transition cost when a user changes input channels from a keyboard or mouse to any of these regions, and effects on fatigue.

Touch Regions

The main independent variable for the study is the touch region: top (*t*), bottom (*b*), left (*l*), and right (*r*) regions of the desk surface, and the vertical screen (*s*) (Figure 2).

To determine reasonable sizes for these touch regions, we surveyed 20 daily computer users on their normal seating and keyboard positions. Results showed that 95% of users put the keyboard centered in front of their body. The mean distance between the bottom edge of the keyboard and center of their bodies was 17cm with a standard deviation of

5.3cm. The mean distance between the user's eyes and the screen was 68cm with a standard deviation of 10.5cm.

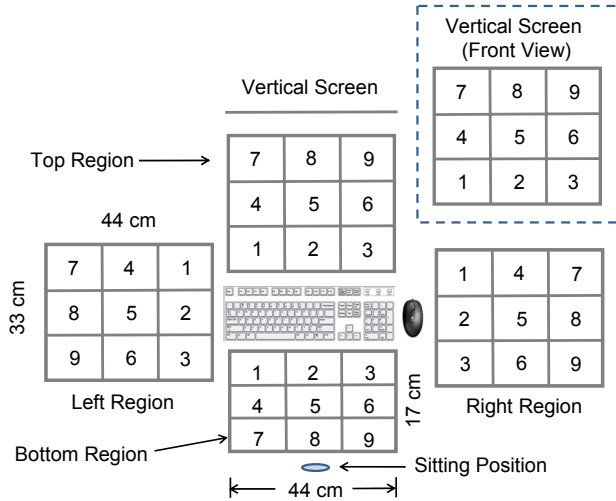


Figure 2. Experiment touch regions (top view). A front view of the vertical screen is illustrated in the top-right corner.

According to Hedge et al's [13] model, the Zone of Comfortable Reach area (ZCR) on a table is a spherical shell centered on each shoulder, the radius of which is the asromion to grip distance. Guided by the size of a regular keyboard (45 X 16cm) and a normal human arm's length (75cm), we used a 44 x 33 cm rectangle for the top, left and right regions, which would cover more than 90% of the ZCR area in these regions. Constrained by the sitting distance, we set the size of the bottom region to 44cm x 17cm.

The size of the vertical screen was also 44 x 33 cm, which is a reasonable size for a monitor. The distance between the screen and the center of the body was fixed at 68cm, which was the average eye-screen distance from the survey. Given the average human arm length of 75 cm, most users can comfortably touch the screen from such a position. On the right side we offset the screen from the keyboard by 8cm to leave room for the mouse, since we felt it would be impractical to have a multi-touch device right beside the keyboard, where it would get occluded by the mouse.



Figure 3. A participant performing the experiment in left (a), bottom (b), and screen (c) conditions.

Apparatus

We used a 21" multi-touch enabled screen with a resolution of 1600 x 1200 to simulate each of the touch regions. For the *l*, *r*, *t*, and *b* regions, the screen was placed horizontally on the table, and the keyboard and mouse were raised to the same plane as the screen (Figure 3). In the *s* condition, the monitor was tilted 10 degrees backwards (Figure

3). A standard 101-key keyboard, 44cm wide, and 16cm deep, was used.

Task Positions

We further divided each region into a 3x3 grid of cells (Figure 2). The height and width of each cell was one third of its region's height and width. We numbered the cells in the row/column closest to the keyboard with #1, 2 and 3, while cells in the furthest row/column with #7, 8, and 9.

Tasks

We designed three tasks, as abstractions of the type of gestures that might be performed with multi-touch.

Gesture Task. This task represents a simple single finger gesture. Initially, a start circle, gesture direction line, and objective line (80 pixels wide) appeared on the touch screen (Figure 4a). The center of the starting circle was in the center of one of the 9 cells, and the direction of the gesture line was either up, down, left or right. The distance between the center of the starting circle and the objective line was 125 pixels. A participant had to touch the starting circle with one finger and move to cross the objective line. The widget turned white when the circle was touched (Figure 4b), and gold when the finger crossed the objective line, indicating completion of the task (Figure 4c). If the user failed to cross the target line, the gesture would need to be repeated.

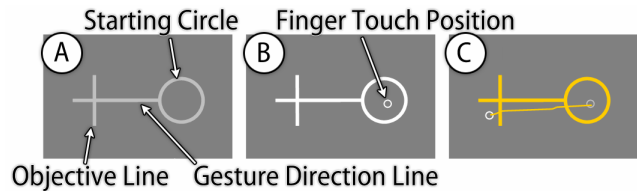


Figure 4. Gesture Task.

One-Hand Docking. This task was designed to represent a one-handed, multi-finger task. Initially, one small green square (150 by 150 pixels) and one large yellow square (250 by 250 pixels), appeared on the screen (Figure 5a). The participants were asked to dock the green square by moving, rotating, and scaling it to cover the yellow square (Figure 5b). The borders of both squares turned gold when the green square was successfully docked (Figure 5c). The participant could manipulate the green squares with commonly used manipulation gestures: translate by dragging it with one or more fingers, scale by moving two fingers apart/together, and rotate by rotating the fingers. Participants were only allowed to use one hand, either left or right, during this task.

The task position was controlled by placing the yellow square in the center of one of the 9 cells. The initial distance between the centers of the green and yellow squares was 500 pixels. The relative offset angle of these two squares was randomized.

Two-Hand Docking. This task was designed to investigate performance of two-handed tasks. Participants performed

the same docking task (Figure 5) but had to use one finger from each hand.

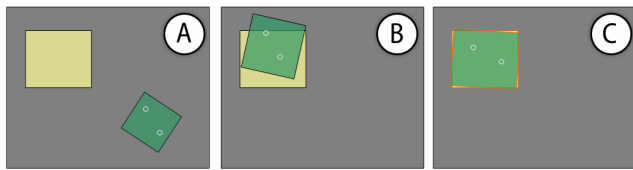


Figure 5. Docking Task. The small white circles in pictures show finger positions.

Hand Positioning for Start and End of Each Task

To test the transition costs between devices, we considered two common desktop configurations of our hands:

Keyboard+Keyboard: In this mode, the trial begins and ends with both hands on the keyboard, when the user simultaneously presses the F and J keys with their left and right hands respectively.

Keyboard+Mouse: In this mode, the participant begins and ends the trial with one hand on the keyboard and one hand on the mouse. The participant would simultaneously press the F key with their left hand and the mouse left button with their right hand to start and end a trial.

For both the *gesture* and *one-hand docking* conditions, users could use either hand to complete the task. An experimenter recorded which hand was used for each task.

Participants

Ten subjects (4 female, ages 18~35) participated in the study, three of whom were left-handed. All worked with computers more than 5 hours per day and naturally operated the mouse with the right hand.

Design

We used a within-subject, full-factorial repeated measure design for all the experiments. Each participant first performed all trials for the *gesture* task, followed by the *one-hand docking*, and finally the *two-hand docking* task.

For each task, the independent variables were *touch region* (*l, r, b, t, s*), *grid cell* within a region (1~9), and *start-end position* (keyboard+keyboard and keyboard+mouse). The orders of the touch regions were counterbalanced using a Latin Square. Half of the participants performed the tasks with keyboard+keyboard mode first, followed by keyboard+mouse mode. For each start-end position within a region, the participant performed tasks in three blocks with each block having 9 trials. Within each block, each grid cell value appeared exactly once, in random order. The design resulted in a total of 270 trials per task, for each participant. Prior to formally starting each task, participants performed three warm-up trials to become familiar with the task. After completing each of the three tasks, each participant rated the five regions according to their overall feelings.

Measures

Completion Time. The *completion time* consisted of *switch forward*, *execution*, and *switch back* times. *Switch forward*

time is the elapsed time between the trial start action and when the first finger contacted the touch region. *Execution* time is the elapsed time between the first finger contact and the moment the participant's fingers finally leaves the experiment area. The *switch back* time is the time elapsed between removing the hand from the touch surface and performing the end trial action.

Number of Clutches. A “clutch” occurs when a participant lifts all fingers off of the surface, and then proceeds to touch the surface again. Since a trial does not end until it has been successfully completed, this measure might provide indication of the difficulty level of the task.

Fatigue Level. To measure muscle fatigue, participants were asked to rate fatigue level after each block (from 0-no fatigue to 7-very fatigued). A 5-minute break was enforced between regions to increase the likelihood that each region began with a 0 fatigue rating. Since our goal was to measure how fatigue level changed as time progressed in a touch region, no break was taken within each region.

Results

We performed a factorial repeated measure ANOVA on each task independently.

Gesture Task

Completion Time. The mean *completion time* (all in ms) for the 5 regions were 2009.3 (*b*), 2162.8 (*l*), 2428.2 (*r*), 2286.0 (*s*), and 2094.6 (*t*) (Figure 6). ANOVA showed that region had a significant main effect on *Completion Time* ($F_{4, 36} = 6.107, p < 0.05$). Pairwise mean comparison were significant for *b X r*, *b X s*, *t X r*, and *t X s* ($p < 0.05$).

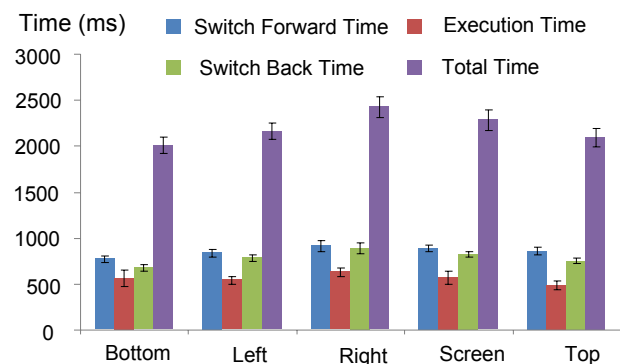


Figure 6. Completion time for Gesture Tasks.

ANOVA showed that region had significant main effects on both *switch forward* ($F_{4, 36} = 4.910, p < 0.05$) and *switch back* Time ($F_{4, 36} = 7.586, p < 0.05$), but not *execution* time ($F_{4, 36} = 1.041, p = 0.399$), indicating that the differences in completion time were mainly from the switching procedures. *Grid Cell* also had a significant main effect on completion time ($F_{8, 72} = 5.75, p < 0.05$). No significant main effect was observed for *start-end position* on completion time. ANOVA showed a significant *Region × Grid Cell* interaction on completion time ($F_{32, 288} = 18.9, p < 0.05$), but not for *Region × Start-end Position* or *Start-end Position × Grid Cell*.

Figure 8a visualizes the mean *completion time* of each grid cell. In the *t*, *l* and *r* regions, the completion times were shortest in the cells closest to the keyboard/mouse (i.e., *grid cells* #1~#3), and increased as the distance from the keyboard/mouse grew. In the *b* region, completion times were more uniform.

Number of Clutches. No significant main effect of region on *number of clutches* was observed. The means were 0.233(*b*), 0.126(*l*), 0.239(*r*), 0.157(*s*), and 0.122(*t*), indicating that most of the time users could successfully perform the gesture task on the first stroke. Similarly, ANOVA did not show significant main effects of *start-end position*, or *grid cell* on *number of clutches*.

Fatigue Levels. We specifically investigated the average fatigue level across all the three blocks. ANOVA did not show a significant main effect of *region*, *start-end position*, or *grid cell* on average fatigue, with means of 0.6(*b*), 0.8(*l*), 0.7(*r*), 1.0(*s*), and 0.5(*t*). All participants reported that *gesture* tasks were easy to perform and they felt little fatigue.

Hand Usage. All the participants performed *gesture* tasks with their left hands on the *l* region, and right hands on the *r* region. Participants performed *gesture* tasks with their right hands for 67%, 71% and 82% of trials in *b*, *t* and *s* regions respectively. They commented that they preferred to perform tasks with their dominant hands.

One-Hand Docking

Completion Time. As shown in Figure 7, the means of *completion time* were 3138.9 (*b*), 3453.5 (*l*), 3860.8 (*r*), 3929.3 (*s*), 3553.2(*t*). ANOVA showed that region had a significant main effect on *Completion Time* ($F_{4, 36} = 7.390, p < 0.05$). Pairwise mean comparison showed significant differences for *bXl*, *bXr*, *bXs*, *bXt*, and *lXs* ($p < 0.05$). A significant *Region* × *Grid Cell* interaction ($F_{32, 288} = 23.1, p < 0.05$) was observed, but not for *Region* × *Start-end Position* or *Start-end Position* × *Grid Cell*. In contrast to the *gesture* task, significant main effects were found for region for *Switch Forward* ($F_{4, 36} = 3.433, p < 0.05$), *Execution* ($F_{4, 36} = 5.025, p < 0.05$), and *Switch Back* time ($F_{4, 36} = 4.585, p < 0.05$). ANOVA also showed a significant main effect of *grid cells* on completion time ($F_{8, 72} = 4.63, p < 0.05$). As shown in Figure 8b, the effect of region on the *one-hand docking* task was similar to the *gesture* task. Users performed uniformly well across the *b* region, and

completion times in the *t*, *l* and *r* regions were shorter in the cells closest to the keyboard or mouse (i.e., *cells* 1-3), and increased as the cells became further away. The mean completion time in the *s* condition was the longest among the five tested regions (Figure 7). ANOVA did not show a significant main effect for *start-end position*

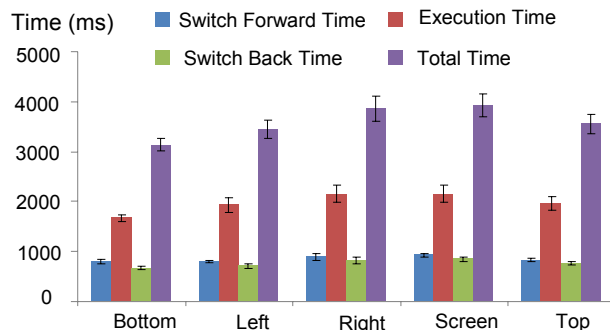


Figure 7. Completion time in one-hand docking.

Number of Clutches. *Region* had a significant main effect on the *number of clutches* ($F_{4, 36} = 5.197, p < 0.05$), with means of 0.31(*b*), 0.50(*l*), 0.48(*r*), 0.65(*s*), and 0.52(*t*). Pairwise mean comparison showed significant differences for *bXl*, *bXr*, *bXs*, *bXt*, *lXs*, and *rXs*, indicating that users clutched least often in the bottom region. ANOVA did not show significant main effects for either *start-end position* or *grid cell* on *number of clutches*. A Significant *Region* × *Grid* interaction ($F_{32, 288} = 12.1, p < 0.05$) was observed, but not for *Region* × *Start-end Position* or *Start-end Position* × *Grid Cell*.

Fatigue Levels. No significant main effect of *region*, *start-end position* or *grid cell* on average fatigue level was observed. Means were 1.3(*b*), 1.8(*l*), 2.4(*r*), 2.0(*s*), 1.6(*t*), and 8 of the 10 participants commented that it was simple and easy to perform these tasks and did not feel fatigued.

Handedness. All participants performed *one-hand docking* tasks with left hand on the *l* region, and right hand on the *r* region. They performed one-handed docking tasks with right hand for 75%, 77% and 82% of trials in *b*, *t* and *s* regions respectively.

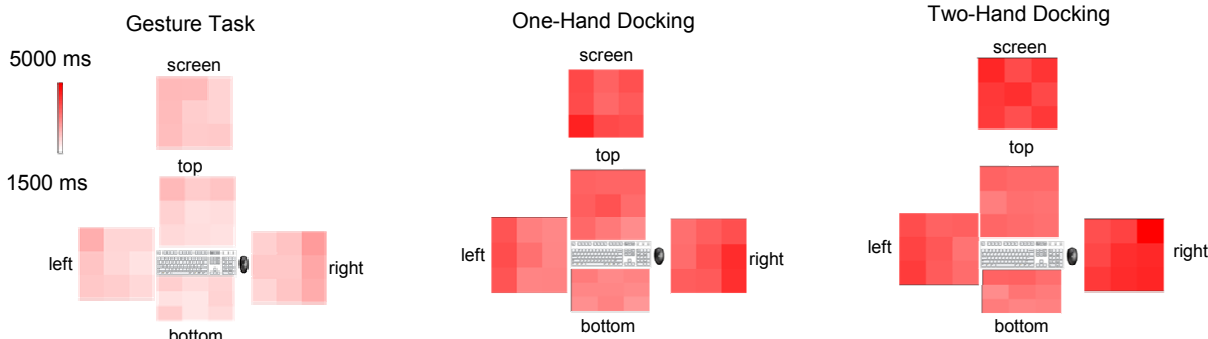


Figure 8. Mean Completion Time per cell in a region.

Two-Hand Docking

Completion Time. The means of *completion time* (ms) were 3538.5(*b*), 4004.8(*l*), 4590.2(*r*), 4434.6(*s*), 3687.4(*t*) (Figure 9). ANOVA showed that region had significant main effects on *completion time* ($F_{4,36} = 16.191, p < 0.01$), *switch forward* ($F_{4,36} = 21.389, p < 0.01$), *execution* ($F_{4,36} = 5.685, p < 0.05$), and *switch back time* ($F_{4,36} = 17.992, p < 0.05$). For completion time, pairwise mean comparison showed significant differences ($p < 0.05$) between every pair of regions except *bXt*, *lXs*, and *rXs*. ANOVA also showed a significant main effect of *grid cells* on completion time ($F_{8,72} = 3.27, p < 0.005$). No significant main effect was observed for *start-end position* on completion time. ANOVA showed a significant *Region* × *Grid Cell* interaction on completion time ($F_{32,288} = 23.6, p < 0.05$), but not for *Region* × *Start-end Position* or *Start-end Position* × *Grid Cell*.

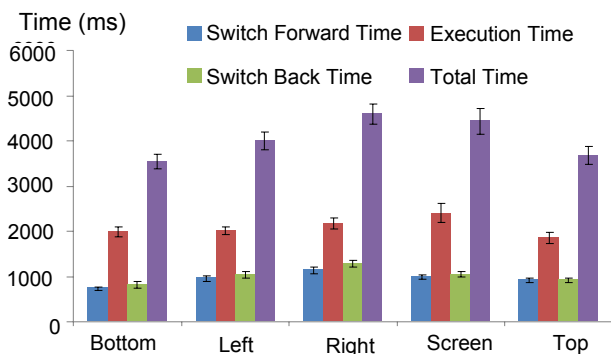


Figure 9. Completion time in two-hand docking.

Number of Clutches. No significant main effect of *region*, *start-end position* or *grid cell* on the number of clutches was observed ($F_{4,36} = 1.744, p = 0.162$), with means of 0.39(*b*), 0.31(*l*), 0.38(*r*), 0.55(*s*) and 0.29(*t*).

Fatigue Levels. Different from both *gesture* and *one-hand docking* tasks, touch region had a significant main effect on average fatigue level ($F_{4,36} = 3.18, p < 0.05$), with the *t* being the least fatiguing and right being the most fatiguing region. The means of average fatigue level were 2.2(*b*), 2.6(*l*), 3.2(*r*), 2.7(*s*), and 2.0(*t*). Pairwise mean comparison showed significant differences for *bXr*, and *tXr*. 6 out of the 10 participants reported that they disliked performing the two-handed docking task in the right region because they had to rotate their waists significantly to complete the task. No significant main effect was observed for either *start-end position* or *grid cell* for fatigue levels. A Significant *Region* × *Grid* interaction ($F_{32,288} = 10.4, p < 0.05$) was observed, but not for *Region* × *Start-end Position* or *Start-end Position* × *Grid Cell*.

Overall Subjective Opinions

Participants rated each region according to their overall satisfaction after completing all the tasks. The three tasks were classified into two categories: one-handed tasks (*gesture* and *one-hand docking*) and two-handed tasks (*two-hand docking*). For the question, “*Are the tested tasks easy to perform in each region (0:very difficult, 4:very easy)?*”,

ANOVA showed that there was a significant main effect of region on rates in the two-handed tasks ($F_{4,36} = 7.075, p < 0.05$), with the *b* and *t* regions being the easiest and the right region being the worst. The mean rates were 3.2(*b*), 1.9(*l*), 1.4(*r*), 2.7(*s*) and 3.2(*t*). No significant main effect was observed for regions on one-handed tasks. These results are consistent with *completion time* results: it is easier to perform two-handed tasks in the *b* and *t* regions.

IMPLICATIONS OF RESULTS

One-Handed Tasks

As expected, users performed one-handed tasks the fastest in zones close to the keyboard or mouse (i.e., *grid cells* 1-3) due to the short travel distance. The entire *Bottom* region performed particularly well in both the *gesture* and *one-hand docking* task: the mean completion time of most *grid cells* in bottom regions is faster than the average completion time of every other region. Some users commented that it was easier and more comfortable to touch in the *bottom* region by just withdrawing a hand back than reaching it out to make contact in the other regions. Additionally, since the reachable area in the bottom region is smaller than those in other regions, the average hand traveling distance is shorter, which also contributes to the faster completion times. One problem with the bottom region is occlusion of the display caused by the user’s hands. Two participants reported this occurring. These problems could be alleviated by designing occlusion-aware interfaces [33].

Given the relative prevalence of touch sensitive monitors, we believe it is a very important result that users performed one-handed tasks poorly on the screen, where mean completion time was the longest in one-hand docking and second longest in the *gesture* task. We argue that lifting arms up from the desk surface to the monitor leads to a greater switching cost, and operating in the air could lead to poor performance. Six out of ten participants reported that performing tasks on the vertical surface was more difficult than on the horizontal surface because they could not rest their arms while performing the tasks.

In summary, the study reveals the users’ capabilities of performing one-handed tasks:

- Users perform one-handed tasks efficiently in zones close to keyboard or mouse.
- Users perform one-handed tasks generally well across the entire *bottom* region.
- The vertical *screen* is a poor region for performing one-handed tasks.

Two-handed Tasks

Overall, the results show that users performed two-handed docking tasks quickly, and felt less fatigue in the bottom and top regions than in the other three regions. We argue that this may be due to ergonomic issues. In the right and left regions, users had to rotate their torso for two-handed tasks to get both hands over to one side of the keyboard. This body rotation might lead to muscle fatigue and poor performance. Similar to one-handed tasks, users com-

mented that operating on the vertical screen required holding their hands in the air, which caused fatigue.

Based on these results, we draw the following conclusions about the users' abilities to perform two-handed tasks:

- The best zones for performing two-handed tasks are the *bottom* and *top* regions.
- Users perform two-handed tasks poorly in the *left*, *right* and *screen* regions, and these regions also caused increased levels of fatigue.

Summary

In both one- and two-handed tasks, some of the results are within our expectation. For example, zones close to the keyboard and mouse are good for one-handed tasks, and the top and bottom suit two-handed tasks well. In doing this study, we validated such expectations, and in addition, provided a quantitative analysis and in-depth understanding of each zone. Specifically, we have captured the precise magnitude of effects by each region and its 9 grid cells.

In addition, the study also reveals some interesting findings. First, the bottom region suits both one- and two-handed interaction very well. Second, the vertical screen is less efficient for touch interaction. This is a particularly important finding, given that touch screen computers are becoming more prevalent [6, 7].

COMBINING MULTITOUCH AND DESKTOP WORK Implementation

Guided by the study results, we designed and implemented a set of interaction techniques integrating multi-touch input with a mouse and keyboard to facilitate desktop work. Our purpose is to demonstrate example interactions and usages, and in particular, demonstrate how different regions within the desktop environment can be used for touch, and how such interactions can be guided by our study results.

The interaction techniques are coherently integrated into a desktop prototype, called Magic Desk (Figure 1, Figure 10). We demonstrate our new techniques in an environment which has all five planar touch surfaces available. The current system was implemented on a Microsoft Surface with a Dell Multi-touch display. A QWERTY keyboard and wireless mouse are used, and have tags so that their position and orientation can be recognized by the surface.

Multi-Functional Touch Pad Abstract Version of a Window

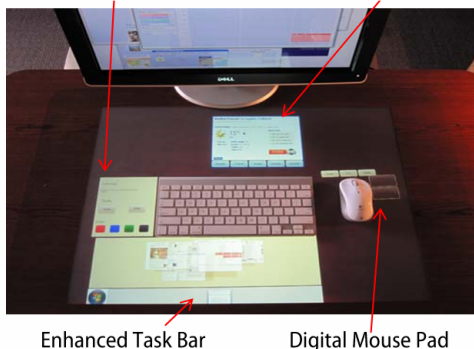


Figure 10. The Magic Desk components.

Enhanced Window Management and Task Bar

As users process increasing amount of digital information, they desire more flexibility of managing windows, such as moving/resizing multiple windows simultaneously, and arranging multiple windows to form a semantic layout [4]. To enhance the flexibility and increase the input bandwidth of managing windows, we designed an enhanced task bar (Figure 10), allowing users to simultaneously manage multiple windows directly with two hands.

Thumbnails of open windows are displayed in the enhanced task bar and the location and sizes of these thumbnails conveys the spatial location and sizes of open windows on the monitor. Since the Enhanced Task bar has a wider aspect ratio than the vertical computer screen, overlapping windows are spread out more horizontally and thus are more accessible for manipulation. Moreover, the following operations are enabled:

Resize. Moving fingers apart (or together) on the thumbnails enlarge (or shrink) the corresponding windows.

Maximize/Restore. Double tapping on the thumbnail maximize/restore the corresponding window.

Minimize/Restore. Flicking the thumbnail down minimizes the corresponding window and sends the thumbnail to a bottom strip. Flicking the thumbnail up from the bottom region restores the window.

Implications from the Study. As the enhanced task bar technique involves a rich set of two-handed operations, we suggest placing this component at either bottom or top region. In the current Magic Desk system, the enhanced task bar is coupled to the bottom edge of the keyboard.

Multi-Functional Touch Pad

Two-handed interaction has been shown to be beneficial in certain interaction tasks, such as controlling multiple-degrees-of-freedom [18]. By designing a multi-functional touch pad on the left side of the keyboard, we enable such interaction paradigm in a desktop work environment: the right hand interacts with the mouse while the left hand uses the touch pad. We implemented the following functions:

Controlling multiple degrees of freedom. The mouse is used to select a target, while the left hand fingers control additional degrees of freedom (e.g., rotating and scaling a geometric object) (Figure 11a).

Adjusting Control-Distance Gain of a mouse. Through the mouse speed region on the touch pad (Figure 11b), users can move their fingers apart to increase the CD gain and together to reduce it. This can be done in parallel to a mouse operation task with the right hand.

Controlling a secondary cursor. A secondary cursor controlled by the left hand on the touch pad is introduced to work in parallel to the primary cursor operations. Using a relative mapping, the user can move the secondary cursor, that is constrained within a tool palette, to select different tools, while controlling the main cursor on the canvas to draw graphics (Figure 11c).

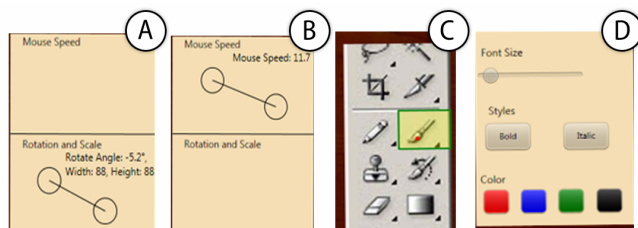


Figure 11. The content on Multi-functional touch pad for (a) rotating and scaling an object, (b) controlling mouse speed, (c) a secondary cursor for selecting drawing tool, and (d) a customized tool palette. The circles in (a, b) show finger positions.

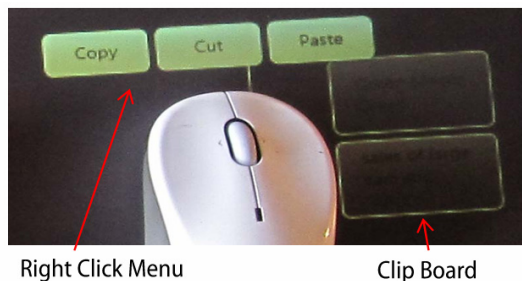


Figure 12. Digital Mouse Pad.

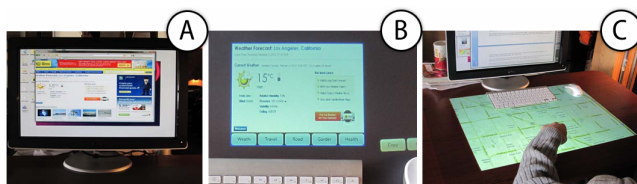


Figure 13. (a) A weather forecast window in full-version on the screen. (b) The abstract version of the same window on the table. (c) After a keyboard and mouse were pushed away, a map application automatically expanded to fill the entire desk.

Customized tool palette. The multi-Functional touch pad can also serve as a repository for storing commonly used UI elements. For example, Figure 11d shows a touch pad with touch buttons and sliders for a text editing program. To add a new element to the palette, a user duplicates it from the monitor by flicking it down with a single finger. The flicked element then animates to the touch pad. Dragging an undesired element out of the palette removes it.

Implication from the Study. All of the interactions on the multi-functional touch pad are one-handed. More specifically, most interactions are performed with the left hand while the right hand is operating the mouse. According to the experiment results, the optimal region for left-hand operations is the rightmost area of the left region. Therefore, the touch pad is coupled with the left edge of the keyboard.

Digital Mouse Pad

The digital mouse pad (Figure 12) is designed to augment mouse operations. The right-click mouse menu is persistently visualized on the digital mouse pad, and the user can trigger commands by directly touching the corresponding button. A multi-item clipboard is also visualized next to the

mouse. Users can directly tap the content on the clipboard to paste it at the location of the cursor. Bringing the common commands on the digital mouse pad allows users to quickly access them, albeit it might require users switch eye-gazing position from the screen to the table. However, as users are familiar with the locations of menus on the digital mouse pad, this switch cost might be reduced. An alternative is to display virtual representations of users' hands on the screen, so that they do not have to look at the table during the interaction.

Implication from the Study. The experiment results indicated that the region close to the right side of the keyboard is one of the high-performing zones for one-handed tasks. UI elements (e.g., right-click menu and clipboard) on this region can be easily accessed.

Continuous Workspace

The touch regions on a desk can be combined with the touch screen to provide a continuous work space. The continuous workspace supports the following operations:

Adapted window content. Users can freely drag windows between displays using fingers or a cursor to take advantage of the extra display surfaces. Since interaction focus is usually located on the monitor, applications on the tabletop mostly play a supportive role in displaying peripheral information [10]. Thus, windows shift from full versions on the screen (Figure 13a) to abstract versions on the interactive table (Figure 13b) to allow users to absorb the most useful information with a simple glance.

Adapted UI layouts. UI elements within a window are rearranged to be close to the keyboard (Figure 13b), because these areas are best for performing one-handed direct touch. The UI elements are also enlarged to suit touch interaction.

Full Tabletop Interaction. Using the entire desk for interaction may be well-suited for specific tasks such as previewing images, navigating maps, and annotating documents. When the keyboard is moved out of the way, the window on the horizontal table will automatically expand to fill the entire desk. The horizontal table now becomes a full multi-touch display, on which users can freely pan and zoom displayed pictures with fingers (Figure 13c). Placing the keyboard back to the center of the desk returns the table to the standard mode.

Implications from the Study. The minimal use of touch on the main monitor was driven by its poor results from the study. Instead, touch for the vertical screen was only used to send content to the horizontal surface. In addition, the adapted UI layouts were guided by our finding that touch regions should be made as close to the keyboard as possible.

Informal User Feedback

We asked each of six users to freely and extensively try the interaction techniques on the table for 40 to 50 minutes. In general, they commented that interaction techniques were easy to learn and use. The most popular features were the enhanced task bar technique for managing windows, the continuous workspace for dragging windows continuously

across regions, and detecting the keyboard position to enable full tabletop interaction. No major problems were observed. Although these sessions were meant for initial feedback, they did give us a sense that the integration of multi-touch into the desktop environment may be welcomed by users.

POSSIBLE ENABLING IMPLEMENTATIONS

Our implementation of Magic Desk was carried out on a Microsoft Surface, which allowed us to prototype interactions within each region surrounding the keyboard. However, one could imagine numerous other configurations supporting one or multiple regions of multi-touch interaction (Figure 14). Figure 14a shows the scenario where the entire tabletop is both display and touch capable. Figure 14b, c and d illustrate how a subset of the touch and display regions could be reproduced by using auxiliary devices. For example, a multi-touch tablet, such as an iPad, could be placed next to the keyboard (Figure 14b). This would support interactions such as our Multi-Functional Touch Pad. Additionally, an ultra-thin, multi-touch display pad, possibly implemented by layering a transparent UnMousePad [28] on top of an e-ink display, could be positioned below the keyboard (Figure 14c), thus enabling enhanced task bar techniques. An additional touch-tablet device could be positioned underneath the mouse to support digital mouse pad operations (Figure 14d). Since the mouse would sit on top of the display, the display could be positioned next to the keyboard, possibly mitigating some of the negative effects associated with the right region in our study, which was displaced from the keyboard to leave room for the mouse.

DISCUSSION AND FUTURE WORK

Some desktop users tend to clutter desk space with various physical objects such as paper documents. To cope with cluttered desk space, we suggest using automatic occlusion reduction [33], adaptive layout [33], freeform display representation [5], or customized tabletop widget [19] techniques. These physical artifacts [2, 15, 30] could also be virtually augmented with multi-touch surfaces.

These cluttered desks can be considered as subsets of the complete multi-touch enabled desktop (Figure 14a): only parts of the entire desk would be available for touch interaction. Our observation study indicates that most users have some spare real-estate in at least one of the regions we studied. (e.g., many users seldom cluttered the bottom region). Touch interaction techniques could be implemented in these areas. In addition, technology development could in turn affect users' behaviors. It is possible that users may adapt their workplaces to create room for supplementary multi-touch surfaces, thus benefiting from the proposed interaction techniques.

A related issue is the potential problem of false touch activations, potentially from the hands resting on a touch enabled surface. Many multi-touch systems, such as the Microsoft Surface, already have finger detection libraries, and can ignore non-finger input. This worked sufficiently in our implementation, but warrants further investigation.

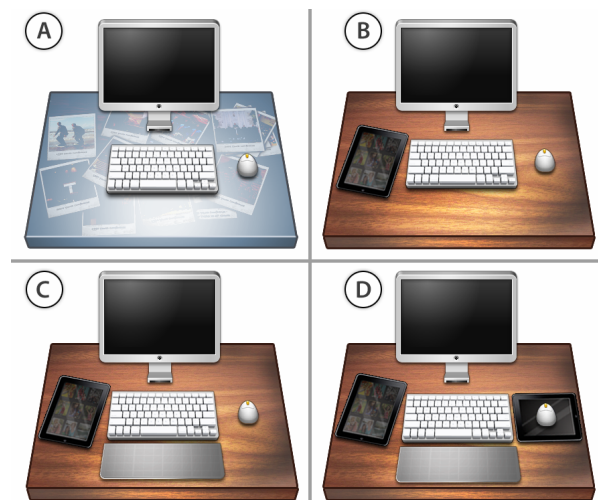


Figure 14. Potential configurations for multi-touch desktop computing. a) The entire table is a multi-touch display surface. b) A multi-touch tablet is placed next to the keyboard to be used as an additional input device. c) The addition of a multi-touch display pad below the keyboard. d) An additional touch display is placed under the mouse.

CONCLUSION

In this paper, we explored both theoretical and practical issues related to integrating planar multi-touch surfaces into a desktop computing environment. We systematically studied user's touch input abilities and transition costs between keyboards/mice and the five planar touch regions via controlled experiments. Guided by the study results, we explored the design space of a multi-touch integrated desktop environment, by designing and implementing a set of interaction techniques utilizing planar touch regions with a mouse and keyboard to facilitate desktop work. All the interaction techniques were coherently integrated into a desktop prototype called Magic Desk. The system demonstrates various possibilities of integrating multi-touch with a mouse and keyboard in desktop work.

REFERENCES

1. Apple Magic Mouse. <http://www.apple.com/>
2. Arai, T., Machii, K., Kuzunuki, S. and Shojima, H. (1995). InteractiveDESK: a computer-augmented desk which responds to operations on real objects. *ACM CHI Poster* 141-142.
3. Block, F., Gellersen, H., and Villar, N. (2010) Touch-Display Keyboards: Transforming Keyboards into Interactive Surfaces. *ACM CHI*. 1145-1154
4. Bi, X., Balakrishnan, R. (2009) "Comparing Usage of a Large High-Resolution Display to Single or Dual Desktop Displays for Daily Work". *ACM CHI*. 1005~1014
5. Cotting, D. and Gross, M. (2006) Interactive environment-aware display bubbles. *ACM UIST* 245-254.
6. Dell Latitude XT2 Tablet PC Touch Screen <http://www.dell.com/tablet?s=biz&cs=555>

7. Dell SX2210T Multi-Touch Monitor
<http://support.dell.com/support/edocs/monitors/sx2210t/>
8. Dietz, P. and Leigh, D. (2001) DiamondTouch: a multi-user touch technology. *ACM UIST*. 219-226.
9. Forlines, C, Wigdor, D., Shen, C., and Balakrishnan, R. (2007) Direct-touch vs. Mouse Input for Tabletop Displays. *ACM CHI*. 647-656
10. Han, J. Y. (2005). Low-cost multi-touch sensing through frustrated total internal reflection. *ACM UIST*. 115-118.
11. Hancock, M. and Booth, K. (2004). Improving menu placement strategies for pen input. *Graphics Interface* 221-230.
12. Hartmann, B., Morris, M. R., Benko, H., and Wilson, A. D. (2009). Augmenting interactive tables with mice & keyboards. *ACM UIST*. 149-152.
13. Hedge, A., Anthropometry and Workspace Design, in *DEA 325/651*. 2002, Cornell.
14. Hinrichs, U., Hancock, M., Collins, C., and Carpendale, S. (2007) Examination of Text-Entry Methods for Tabletop Displays. *Tabletop*. 105-112.
15. Kane, S. K., Avrahami, D., Wobbrock, J. O., Harrison, B., Rea, A. D., Philipose, M., and LaMarca, A. (2009). Bonfire: a nomadic system for hybrid laptop-tabletop interaction. *ACM UIST*. 129-138.
16. Kin, K., Agrawala, M., and DeRose, T. (2009). Determining the benefits of direct-touch, bimanual, and multifinger input on a multitouch workstation. *GI*. 119-124
17. Krueger, M. W., Gionfriddo, T., and Hinrichsen, K. (1985). VIDEOPLACE—an artificial reality. *SIGCHI Bull.* 35-40.
18. Kurtenbach, G., Fitzmaurice, G., Baudel, T., and Buxton, B. (1997). The design of a GUI paradigm based on tablets, two-hands, and transparency. *ACM CHI*. 35-42.
19. Leithinger, D. Haller, M. (2007) Improving Menu Interaction for Cluttered Tabletop Setups with User-Drawn Path Menus. *IEEE TableTop*, 121-128
20. Matthews, T., Czerwinski, M., Robertson, G., and Tan, D. (2006). Clipping lists and change borders: improving multitasking efficiency with peripheral information design. *ACM CHI*. 989-998
21. Myers, B. A. (2001). Using handhelds and PCs together. *Commun. ACM* 44, 11, 34-41
22. Morris, M. Brush, A. J., and Meyers, B. (2008) A Field Study of Knowledge Workers' Use of Interactive Horizontal Displays. *TableTop*. 105-112
23. Moscovich, T. and Hughes, J. F. (2008). Indirect mappings of multi-touch input using one and two hands. *ACM CHI*. 1275-1284.
24. Multi-Touch in Windows 7. microsoft.com/windows
25. Pinhanez, C., Kjeldsen, R., Tang, L., Levas, A., Podlaseck, M., Sukaviriya, N. and Pingali, G. (2003) Creating touch-screens anywhere with interactive projected displays. *ACM MULTIMEDIA*. 460-461.
26. Rekimoto, J. (2002) SmartSkin: an infrastructure for freehand manipulation on interactive surfaces. *ACM CHI*. 113-120.
27. Rekimoto, J. and Saitoh, M. (1999) Augmented surfaces: a spatially continuous work space for hybrid computing environments. *ACM CHI*. 378-385.
28. Rosenberg, I. and Perlin, K. (2009). The UnMousePad: an interpolating multi-touch force-sensing input pad. *SIGGRAPH*. 1-9.
29. Scott, S.D. (2003) Territory-Based Interaction Techniques for Tabletop Collaboration. *ACM UIST Companion*. 17-20.
30. Steimle, J., Khalilbeigi, M., and Mühlhäuser, M. (2010). Hybrid groups of printed and digital documents on tabletops: a study. *ACM CHI EA*. 3271-3276.
31. Ullmer, B. and Ishii, H. (1997) The metaDESK: Models and prototypes for tangible user interfaces. *ACM UIST* 223-232.
32. Villar, N., Izadi, S., Rosenfeld, D., Benko, H., Helmes, J., Westhues, J., Hodges, S., Ofek, E., Butler, A., Cao, X., and Chen, B. (2009). Mouse 2.0: multi-touch meets the mouse. *ACM UIST*. 33-42
33. Vogel, D. and Balakrishnan, R. (2010). Occlusion-Aware Interfaces. *ACM CHI*. 263~272
34. Wellner, P. (1993) Interacting with paper on the DigitalDesk. *Communications of the ACM*, 36 (7), 87-96.
35. Wigdor, D., Penn, G., Ryall, K., Esenther, A., Shen, C., (2007). Living with a Tabletop: Analysis and Observations of Long Term Office Use of a Multi-Touch Table. *TableTop*. 60-67.
36. Wigdor, D., Shen, C., Forlines, C., and Balakrishnan, R. (2006). Table-centric interactive spaces for real-time collaboration. *AVI*. 103-107.
37. Wilson, A. D. (2005) PlayAnywhere: a compact interactive tabletop projection-vision system. *ACM UIST*. 83-92.
38. Wilson, A. D., Izadi, S., Hilliges, O., Garcia-Mendoza, A. and Kirk, D. (2008) Bringing physics to the surface. *ACM UIST*. 67-76.
39. Wu, M. and Balakrishnan, R. (2003) Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. *ACM UIST*. 193-202.
40. Yang, X., Mak, E., McCallum, D., Irani, P., Cao, X. Izadi, S. (2010). LensMouse: Augmenting the mouse with an interactive touch display. *ACM CHI*. 2431-2440.