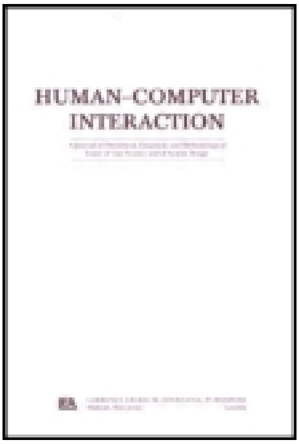


Human-Computer Interaction
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/hhci20

WallTop: Managing Overflowing Windows on a Large Display

Xiaojun Bi [a] , Seok-Hyung Bae [b] & Ravin Balakrishnan [a]
[a] University of Toronto , Canada
[b] Korea Advanced Institute of Science and Technology , Republic of Korea

Accepted author version posted online: 14 Jun 2013.Published online: 27 Dec 2013.

To cite this article: Xiaojun Bi , Seok-Hyung Bae & Ravin Balakrishnan (2014) WallTop: Managing Overflowing Windows on a Large Display, Human-Computer Interaction, 29:2, 153-203, DOI: 10.1080/07370024.2013.812411

To link to this article: http://dx.doi.org/10.1080/07370024.2013.812411

HUMAN–COMPUTER INTERACTION, 2014, Volume 29, pp. 153–203
Copyright © 2014 Taylor & Francis Group, LLC
ISSN: 0737-0024 print / 1532-7051 online
DOI: 10.1080/07370024.2013.812411

WallTop: Managing Overflowing Windows on a Large Display

Xiaojun Bi,[1] Seok-Hyung Bae,[2] and Ravin Balakrishnan[1]
[1]University of Toronto, Canada
[2]Korea Advanced Institute of Science and Technology, Republic of Korea

With the ever increasing amount of digital information, users desire more screen real estate to process daily desktop computing work, and might well benefit from using a large high-resolution display for information management. Unfortunately, we know very little about users' behaviors when using such a display for daily computing, and current user interfaces are mainly designed for normal-sized desktop monitors, which might not well suit a large display. In this article, we first present a longitudinal study that investigates how users manage overflowing digital information on a wall-sized display in a personal desktop computing context by comparing it with single and dual desktop monitors. Results showed users' unanimous preferences of working on a large display and revealed large-display users' unique activity patterns of managing windows. Guided by the study results, we designed a set of interaction techniques that provide greater flexibility in managing multiple windows. They include facile methods for selecting, moving, and resizing multiple windows using an active window boundary called *Fringe*, rearranging selected windows using multi- and single-window marking menus, packing/unpacking the selected windows using easily activated icons, and freely adjusting the order of overlapping windows with a *Jab-to-Lift* operation. We coherently integrated these techniques with traditional operations in a large-display window management prototype called *WallTop*. Two rounds of usability testing showed that users can quickly and easily learn the new interaction techniques and apply them to realistic window management tasks on a large display with increased efficiency.

Xiaojun Bi (xjunbi@gmail.com) is a computer science researcher with an interest in human–computer interaction; he is now at Google. **Seok-Hyung Bae** (seokhyung.bae@kaist.ac.kr) is an Assistant Professor in the Department of Industrial Design, Korea Advanced Institute of Science and Technology; his research interests include computer-aided industrial design, human-computer interaction, and computer graphics. **Ravin Balakrishnan** (ravin@dgp.toronto.edu) is a Professor of Computer Science and Canada Research Chair in Human-Centered Interfaces at the Department of Computer Science, University of Toronto.

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/hhci.

CONTENTS

1. INTRODUCTION

The rapid development of information technology has been boosting information growth over the past decade. The total amount of digital information on earth reached 281 billion GB in 2007, around 45 GB per person. This explosively growing "digital universe" enriches people's daily digital lives while giving desktop users heavy workload; it is common that they juggle multiple applications processing massive information on a desktop computer.

To effectively process overflowing information, desktop users tend to work on a big monitor simply because it can display a large amount of information. Research literature reveals overwhelming benefits of working on a big desktop monitor. Simmons's (2001) study showed that a 21-in. monitor outperformed 15-in., 17-in., and 19-in. ones in spreadsheeting tasks and multitasking. Czerwinski et al.'s (2003) work showed that a 42-in. display outperformed a 15-in. monitor in multiple-step, cognitively loaded tasks. A recently released Pfeiffer Report (Pfeiffer Consulting, 2005) revealed that a 30-in. monitor led to increased productivity not only when using professional design, digital imaging, and digital video authoring programs but also when using office applications such as word processing and spreadsheet programs, in comparison to 20-in. and 17-in. monitors.

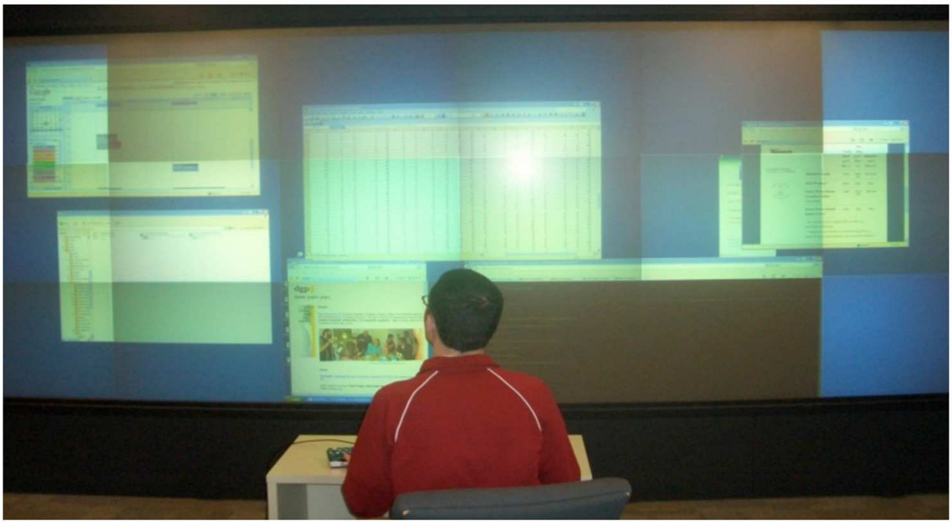
Thanks to the advances of display technology, big desktop monitors are increasingly affordable and have been gradually adopted. Over the past decade, the size of commonly used desktop monitor has increased from 15 in. to 25 in. Thirty-inch screens are now available at commodity prices. Many manufactures have released high-resolution LCDs bigger than 100 in. As this trend develops, we believe that it is just a matter of time before desktop users have a very large display for daily work; they may purchase digital wallpaper and attach it on a wall to obtain a large display, which thereafter serves as the primary screen for desktop work. Investigating the effects of such large displays and designing appropriate interaction techniques become timely issues.

A wall-sized display is able to visualize a great amount of information, which seems beneficial in processing overflowing digital information. On the other hand, the excessive amount of digital information might exceed human perception capability. It could become distracting, thus hindering users' information-processing performance. In desktop computing environments, digital information is visualized and presented through windows. The large amount of digital information will probably lead to increased workload for managing windows thus offsetting the benefits of the large working space. Thus, before we step into large-display-enhanced desktop environments, it is important to disentangle their pros and cons.

In the physical world, office workers' activities change drastically when they switch from a pilot tray table to a large office desk, which offers a greater working space, thus providing more flexibility for arranging paper documents and making sense of different files. Likewise, will desktop users' information management behaviors also undergo drastic changes when they replace a normal-sized monitor with a wall-sized display in virtue of its greater amount of screen real estate? Current window management systems, such as Windows XP or Mac OS, are designed for normal-sized monitors, which allow users to operate only on a single window at a time. If users' window management behaviors change drastically, how should we refine them to fit large-display users' demands? These questions are fundamental but their answers still remain unknown.

In this article, we first present a longitudinal diary study that investigates users' window management behaviors when they switch from standard computing envi-

FIGURE 1. Processing desktop work on a wall-sized high-resolution display (size: 4.88 m wide × 1.83 m high, resolution: 6144 × 2034).



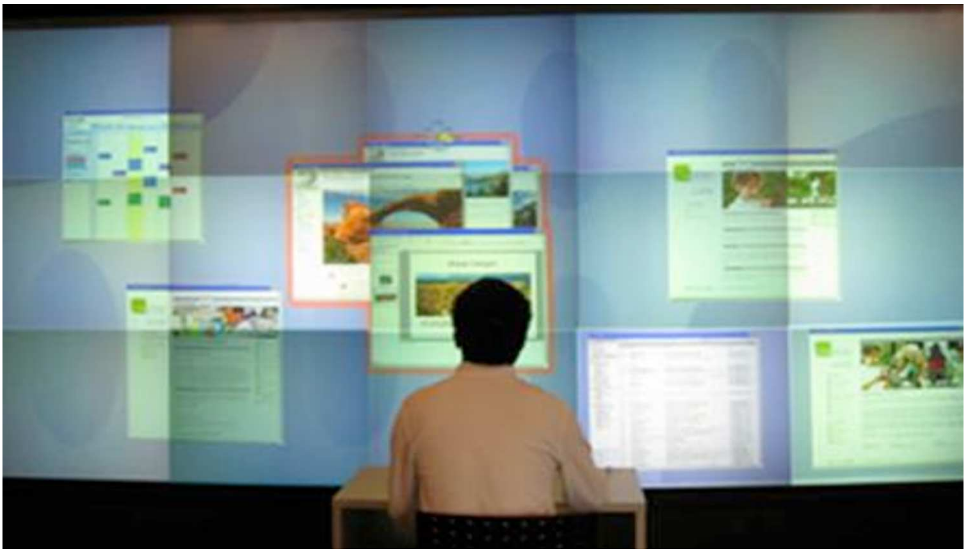
ronments (single or dual monitor) to a wall-sized high-resolution display (Figure 1) a 5-day period. By recording participants' daily activities, window events, and mouse operations, we investigate how users manage overflowing digital information in different display configurations. In addition, we discuss the pros and cons of working on a wall-sized display.

Based on the study result, we then design and implement a set of new interaction techniques addressing the challenge of managing windows on a large display. One of them is the active window boundary widget called *Fringe* that enables users to freely operate on an arbitrary number of windows and that facilitates window moving/resizing operations. Coupled with marking menus (Kurtenbach, 1993), it provides users with a variety of multi- and single-window operations, such as spreading, piling, and quick centering/moving aside/closing. In addition, we design the *Jab-to-Lift* operation to offer flexibility in rearranging the overlapping order of windows. All these new techniques are coherently integrated with traditional window operations into a large-display-oriented window management prototype, called *WallTop* (Figure 2). Two rounds of usability testing show that these new techniques significantly improve the efficiency of managing windows on a large display.

2. RELATED WORK

We first discuss research studying effects of normal-sized desktop monitors in Section 2.1, followed by a review of research about utilizing multiple monitors of

FIGURE 2. A user managing windows on WallTop.



desktop work in Section 2.2. Section 2.3 summarizes work about using a large display for desktop work. In Section 2.4, we review interaction techniques addressing the challenge of managing windows on a large display.

2.1. Using a Normal-Sized Monitor to Process Desktop Work

A sizeable amount of research has been carried out investigating how commonly used desktop monitors affect users' performance and behaviors for desktop work. Simmons (2001) conducted a study comparing user's performance on 15-in., 17-in., 19-in., and 21-in. monitors in word-processing, spreadsheeting, multitasking, web-browsing, slide-making, and information-collecting tasks. The results revealed that users performed spreadsheeting and multitasking in significantly less time on the 21-in. display than on the others. Statistical analysis also indicated that the 19-in. and 21-in. displays exhibited preference scores significantly higher than the 17 in. and 15 in. Moreover, in a computer-based verbal test, empirical studies (Fridgeman, Lennon, & Jackenthal, 2003) showed that users achieved higher scores on a 17-in. (1024 × 768) monitor than on a 15-in. (640 × 480) or 17-in. (640 × 480) one, indicating that a larger high-resolution monitor could improve performance. For a text reading task, a 15-in. screen generated a more efficient integration process in constructing semantic representation than a 12-in. one, leading to less learning time (Brujin, Mul, & Oostendorp, 1992). A Pfeiffer Report (Pfeiffer Consulting, 2005) showed that a 30-in. monitor led to increased productivity in professional design, digital imaging, digital video authoring, word processing, and spreadsheeting, in comparison to 20-in. and 17-in. monitors.

Aside from performance increases, a bigger monitor also provides ergonomic benefits. Sommerich, Joines, and Psihogios's (1998) work revealed that a 19-in. monitor generated less muscle activity than a 13.6-in. one for screen-intensive work.

Previous research revealed the overwhelming benefits of a bigger, normal-sized monitor over a smaller one. However, a normal-sized monitor occupies only a small portion of a user's visual field, roughly less than 10%. When switching to a large display that can take more than 70% of the visual field (i.e., a wall-sized display), the findings for normal-sized monitors might not hold, and users' desktop behaviors might change drastically. Thus, substantial pieces of research are needed to investigate for large displays.

2.2. Using Dual or Triple Monitors to Process Desktop Work

Besides using a large display, another approach to expanding screen real estate is to use multiple monitors. Current technology allows users to easily and cheaply get additional monitors. Recent video cards can support more than four independent monitors; many PCs allow users to use multiple video cards simultaneously. Extra monitors provide a user with additional screen real estate, offering more flexibility for daily work.

Grudin (2001) documented the usage of dual monitors, revealing that second monitors are generally used for secondary activities related to principal tasks, for peripheral awareness of information outside the main focus, and for easy access to useful resources. Hutchings, Smith, Meyers, Czerwinski, and Robertson (2004) deployed a tool, called VibeLog, to allow a group of single-, dual-, and triple-monitor users to log window management activities. Analysis of VibeLog data showed differences in windows management activities between different display configurations. Due to the long distance a user had to traverse to click on the taskbar, dual- and triple-monitor users tended to access windows directly more and used the taskbar less than single-monitor users. As dual and triple monitors provided a greater amount of screen real estate than a single one, there were more visible windows on dual- and triple-monitor configurations. Moreover, as the larger visualization surface could display more information, users preferred to work with dual or triple monitors.

Dual or triple monitors allow users to visualize more information simultaneously; however, they can also cause some usability problems. For example, a large number of visible windows pose an extra burden for window management. Small windows such as dialogues and toolbars may appear at unexpected positions when invoked (Hutchings et al., 2004). To ease the window management on dual or triple monitors, Hutchings and Stasko (2007b) developed the Snip technique, which allows a user to constrict the view onto any window, thus reducing the space needed to display information and resulting in fast information access. The mudibo system (Hutchings & Stasko, 2005, 2007a) alleviates the problem of dialogues and toolbars appearing at unexpected locations by placing the invoked small windows in multiple locations simultaneously. This technique allows users to easily interact with dialogues/toolbars in any desired location.

Both multiple monitors and a large display offer users plentiful screen real estate. The major difference between them is that multimonitor configurations introduce physical bezels or even gaps within the visualization surfaces, which cause visual discontinuity and affect users' performances and behaviours. Bi et al.'s (2010) studies showed that the bezels were detrimental to visual search performance and the tunnel-steering operations. Ball and North (2005) showed that some users took advantage of the bezels and utilize them to divide different work space. Because of the effects introduced by the bezels, prior research on multiple monitors can at best serve as a very rough guide to single, large high-resolution display usage; research explicitly focused on the latter is clearly required, thus this article takes one step in this direction.

2.3. Using Large Displays in Desktop Computing Environments

To facilitate processing the increasing amount of digital information, some researchers initially investigated using a very large, even wall-sized display as a desktop monitor.

Andrews, Endert, and North (2010) conducted studies investigating effects of a large, 32-megapixel (10240 × 32000) display on analytic tasks. The display consisted of a 4 × 2 grid of 30-in. LCD panels, each with a maximum resolution of 2560 × 1600. Their studies showed that analysts use the large screen real estate both as a form of rapid access external memory and as an added semantic layer in which meaning was encoded in the spatial relationships between data, documents, display, and analyst.

Ball and North (2005) reported an observational study of the use of a large tiled (3 × 3) display composed of nine 17-in. monitors. The entire large display was approximately 37-in. tall, with the total resolution of 3840 × 3072. In their study, five participants used this nine-monitor tiled display for at least 3 months in a form of time sharing. All of the participants were graduate students in computer science and used it to perform normal daily activities such as reading/writing e-mails, browsing the Internet, writing papers, programming, viewing images, and running experiments. A behavioral analysis showed that the large high-resolution display afforded a number of advantages, including improving task switching and viewing large documents, increasing users' collaboration abilities, and enhancing awareness for secondary tasks. One interesting finding was that the physical bezels surrounding each monitor affected the large display usage. Users avoided positioning a picture across the bezels not to result in visual discontinuity, and on the other hand, they utilized the bezels to segregate work space.

Bishop and Welch (2000) reported their 1-year experience of using projector displays as primary workplaces. Two projectors were mounted overhead to display abutted images on a large, flat display surface. Each projector displayed 1024 × 768 pixels for a composite display of 2048 × 768 pixels. Numerous advantages were reported throughout the study, including improved social and technical interaction, better ergonomics, and high information content. Aside from the benefits, the large display system introduced some problems, such as generating extra heat and noise and occasionally compromising privacy. Despite these shortcomings, overall, the authors concluded that they felt better, got more work done on the large display, and never wanted to go back to conventional displays.

Czerwinski et al. (2003) carried out a study to characterize productivity benefits of using a large display for complex, multiapplication office work. Participants performed multiple-step, cognitively loaded tasks in two conditions: a normal 15-in. monitor, and a 42-in. wide surface, called DSharp, created by using three projectors with a resolution of 1024 × 768 onto a curved Plexiglas panel. The result showed a significant performance advantage of using the large display, as well as positive user preference and satisfaction with its use over the smaller one.

Andrews et al. (2010), Ball and North (2005), and Bishop and Welch (2000) conducted qualitative studies investigating a large display usage in a realistic setting. However, all of them studied the effects of a large display solely. They did not explicitly compare the use of a large display with regular desktop settings. Distinct from the prior work, our longitudinal diary study investigated the large display effects in comparison with traditional single- and dual-monitor environments. Furthermore, we specifically investigated window management behaviors through analyzing basic mouse and windows events.

2.4. Window Management

Given a large visualization surface, desktop users tend to open numerous windows to process increasing amount of digital information. How to effectively manage such a large number of windows becomes a big challenge for large-display users.

Designs addressing this challenge include WinCuts (Tan, Meyers, & Czerwinski, 2004), GroupBar (Smith et al., 2003), Scalable Fabric (Robertson et al., 2004), proxy-based & cursor sweeping actions (Ahlström, Großmann, Tak, & Hitz, 2009), Taskposé (Bernstein, Shrager, & Winograd, 2008), and importance-driven compositing window management (Waldner, Steinberger, Grasset, & Schmalstieg, 2011). WinCuts enables users to replicate arbitrary regions of existing windows into independent windows. The users can fill the information as much or as little space as they like. GroupBar provides task management features by extending the current desktop taskbar metaphor. It facilitates task management by letting users drag and drop tiles representing open windows into high-level tasks called groups. Scalable Fabric manages multiwindow tasks on the Windows desktop using a focus-plus-context display to allocate screen real estate in accordance with users' attention. Users can define a central focus area as well as a peripheral region on the display surface. Focus-area windows behave like normal desktop windows, whereas windows in peripheral areas are scaled down. Different from standard title bar and border dragging techniques, Ahlström et al. (2009) implemented techniques for resizing and moving windows. These two actions are performed by manipulating proxy objects close to the cursor and by sweeping cursor motions across window borders. Taskposé is a window visualization tool that lays out related windows close to one another. Waldner et al. (2011) proposed optimizing the spatial window layout for maximum visibility and interactivity of occluded contents. Other techniques to improve window management include desk pop & stack leafing (Faure, Chapuis, & Roussel, 2009) and Push-and-Pull Switching (Xu & Casiez, 2010).

One challenge of managing windows as well as other objects (e.g., icons, shortcuts) on a large display is to access them when they are located far away. Techniques addressing this challenge include drag-and-drop (Collomb, Hasco, Baudisch, & Lee, 2005), drag-and-pop (Baudisch et al., 2003), and tablecloth (Roberson et al., 2005).

There also have been many other window management systems, most of which were designed for typical desktop displays. Virtual desktop managers expand the space available for application windows by allowing users to switch between different workspaces (Henderson & Card, 1986). This technique virtually increases the limited screen real estate of desktop monitors. Task Gallery (Robertson et al., 2000) and Data Mountain (Robertson et al., 1998) use 3D virtual environments to help users organize windows and documents, leveraging human spatial cognition and perception. Pad++ (Bederson & Hollan, 1994) provides users with a large, zoomable virtual space, and focus-plus-context screens (Baudisch, Good, & Stewart, 2001) extend users' desktop with additional low-resolution large screen space. Kandogan and Shneiderman (1997) proposed the concept of elastic windows, which are organized in a hierarchical manner and arranged to fill the screen to effectively utilize the available space. Chapuis and Roussel (2005) proposed Metisse, a window management system in which windows can be rotated, scaled, and peeled back (Beaudouin-Lafon, 2001). Stuerzlinger, Chapuis, Phillips, and Roussel (2005) proposed Façades, allowing users to customize user interface elements to create adaptable and adaptive interfaces.

In this paper, we incorporate some existing concepts as is or in a modified manner, such as grouping (Robertson et al., 2004), folding windows (Chapuis & Roussel, 2007), and marking menus (Kurtenbach, 1993) to propose a framework of Fringe+marking menu, which allows users to freely operate on either single or multiple windows, as opposed to other systems designed only for single-window management. Furthermore, we integrate a set of new interaction techniques into WallTop, which address the challenges that have been neglected in previous research (refer to Figure 31 that details the differences between WallTop and other window management systems).

Other related work involves managing the spatial aggregation of objects. Mander, Salomon, and Wong (1992) utilized piling to organize documents on a desktop and implemented gesture-based piling operations. Agarawala and Balakrishnan (2006) developed BumpTop, a virtual desktop based on physical simulation, which uses piling as the fundamental structure to organize desktop icons. Watanabe, Washida, and Igarashi (2007) designed bubble clusters for manipulating desktop icons and digital ink; aggregated objects are automatically recognized as a group so that the user can drag, copy, and delete them effectively. Distinct from this work organizing desktop icons and digital ink, our system manages windows, thus leading to unique visualization design and multiobject operations.

3. A LONGUITIDNAL DIARY STUDY: LARGE DISPLAY VERSUS SINGLE/DUAL MONITOR(S)

To understand how users process desktop work on a wall-sized high-resolution display, we conducted a longitudinal diary study.[1] Especially, we aimed to disentangle the pros and cons of a large display and understand how users manage windows in comparison to traditional desktop computing environments (i.e., single or dual monitor). Rather than conduct a controlled experiment to examine individual aspects in isolation, we carried out a diary study in a more realistic context, allowing us to explore usage in a broad range of computing activities over a 5-day period.

Future offices will be equipped with displays in various shapes and forms, including wall displays, tabletop displays, and monitors with different sizes. Constrained by the structure of a personal office, the largest single continuous display that an information worker can work with is at most wall sized (e.g., 4 m wide × 1.5 m high). Our study focuses on this extreme situation. Investigating such a setting would reveal how the large visualization capacity affects users' performance and behaviors. Also, on its own it represents a category of large-display usage—using the large display mounted on the wall as the primary work space (e.g., Bishop & Welch, 2000).

3.1. Method

Participants

We recruited eight participants by posting online advertisements: four who used a single-monitor configuration and four who used a dual-monitor configuration for their daily computing. Two single-monitor users (S1, S2) usually worked on a 17-in. LCD, and the other two (S3, S4) used a 21-in. LCD in their daily work. Two of the dual-monitor users (D1, D2) worked on two 18-in. LCDs, one (D3) used a 21-in. in conjunction with an 18-in. LCD, and one (D4) used two 21-in. LCDs. S4 and D2 were female and the others were male. All participants used computers more than 5 hr per day. S3, D2, and D3 were graduate students from Computer Science Department; S1, D1, and D4 were graduate students from Electronic and Computer Engineering Department. According to our preexperiment questionnaire, their daily computing activities were programming, reading/writing documents, web browsing, and e-mailing. S2 was a graduate student from Chemical Engineering Department. His major activity during the 2 weeks of the study was writing up his Ph.D. thesis. S4 was an unemployed person, and her major activity was seeking job information online. Each participant ensured that he or she performed similar computing tasks during the 2 weeks of the study. All of the participants were Windows XP users, and a variety of software was installed on the large-display computer to ensure that they could carry out their daily work. The study was conducted in a locked office

[1] Part of the study results were reported in the paper "Comparing Usage of a Large High-Resolution Display to Single or Dual Desktop Displays for Daily Work" (Bi & Balakrishnan, 2009).

to ensure privacy. Each participant received $50 compensation upon the completion of the study. We chose experienced computer users as we wanted to see how their interaction and visualization strategies might change when they moved to a single, large high-resolution display; in contrast, inexperienced users would not have a baseline of the strategies that they would need to adapt when moving to our large-display setup.

Apparatus

The highest resolution of a current display on the market is 2560 × 1600 (e.g., Apple Cinema 30-in. HD display, http://www.apple.com). Tiling multiple projectors is a common approach to obtaining a seamless, high-resolution display beyond this limit. In our study, we used a 4.88 m wide × 1.83 m high display (Figure 1), composed of 18 projectors in a 6 × 3 tiling, to create a geometrically seamless image. We did only basic color and brightness calibration; hence, there was some variation in brightness and color between each two projectors' part of the overall image. Each projector had a resolution of 1024 × 768 pixels, for a total display resolution of 6144 × 2304 pixels. The size of a single pixel was 0.079 cm wide × 0.079 cm high, which subtended a visual angle of 0.023° with a sitting distance of 2 m. This visual effect is equivalent to viewing a 0.0316 cm × 0.0316 cm pixel (80 dpi) with a sitting distance of 80 cm.

As there was no operating system specific to large displays, we ran standard Windows XP on a computer with several multiheaded graphics cards that could drive multiple projectors concurrently. Various software applications were also installed to ensure that participants could perform their daily work. A movable desk with four wheels was initially positioned on the southwest corner of the large-display room, and participants were asked to freely move it according to their preferences. The input devices were a traditional mouse device and keyboard.

Design

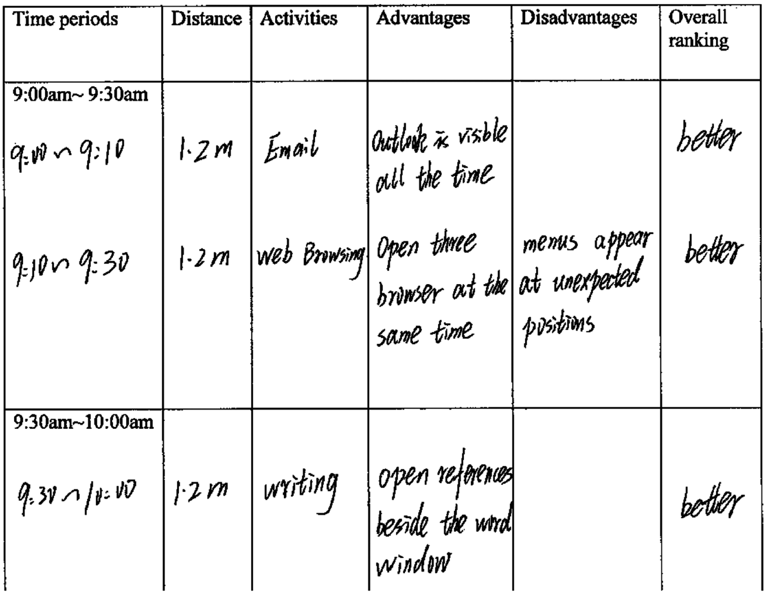
Participants were categorized into two groups according to their normal computing environments: single-monitor or dual-monitor users. Participants switched to exclusively use the large display as their daily computing environment for 5 consecutive days, with 5 hr per day. Thus, each group had two working conditions: their normal working environments (single- or dual-monitor) and the large-display condition. The study ran over an 8-week period, with one participant using the display each week.

Participants were asked to perform their daily routine work on the large display. Throughout the study, each participant was the only person in the large-display room, ensuring a personal office scenario and keeping privacy. Because the optimal position and distance of using the large display was unclear, we let participants freely adjust the sitting position and distance in the study.

To understand how users work on the large display, we employed the following observation methods.

Activity Log. Each participant maintained a daily activity log when working on the large display, writing down what the participant did every half an hour. Figure 3

FIGURE 3. **An example activity log.**



Time periods	Distance	Activities	Advantages	Disadvantages	Overall ranking
9:00am~ 9:30am					
9:00 ~ 9:10	1.2m	Email	Outlook is visible all the time		better
9:10 ~ 9:30	1.2m	web Browsing	Open three browser at the same time	menus appear at unexpected positions	better
9:30am~10:00am					
9:30 ~ 10:00	1.2m	writing	open references beside the word window		better

shows part of a daily activity log. The participant briefly described the activities in the last half hour; recorded the approximate sitting distance to the display; listed the advantages/disadvantages of undertaking these activities on the large display; and finally ranked the large display as “Better,” “Worse,” or “Equal” in comparison to performing the same activities in the normal computing environment.

Daily Interview. To gain further insight of users’ activities and examine the issues not covered by the daily activity log, a follow-up structured interview was carried out at the end of each day. The interview occurred right after the participant finished daily work on the large display. The whole process usually took 1 hr and was taped for later analysis. In the interview, participants were asked to elaborate on the events recorded on the activity log, describe screen real estate usage, and talk about the pros and cons of using the large display. We asked the users for more detail if their explanations were unclear. However, in order not to bias the outcome, we paid special attention to ensure that the additional questions were nonsuggestive and only clarifying in nature.

Recording Windows and Mouse Events. We ran the VibeLog application (Hutchings et al., 2004) on the large display to record every window event that occurred and detailed information of each running window. It maintained two kinds of logs: *Event Log* and *Window Log*. Event Log had an entry for every window management event that occurred. These events included closing, activating, moving, resizing, minimizing, and maximizing a window. Each log entry had a time stamp and the title of the window operated. Complementarily, Window Log created a series of entries each minute, one entry for each opened window on the system. The entry contained detailed information about the corresponding window, including the spatial coordinates, size, and status of the window, and the window's overlapping order on desktop.

We also recorded every mouse event using an in-house application called *MouseLog*. It created an entry for every mouse activity, which contained the spatial coordinates of the mouse cursor, and event type (i.e., left button down/up, right button down/up, and dragging with left/right button down).

Both VibeLog and MouseLog worked together by programmatically hooking into the public window system events made available by Windows XP. They occupied little system resources and did not interfere with normal computer use.

These four recording means are complementary to one another. The daily activity log provides data about a participant's real-life activities in the workplace, whereas the follow-up interview extends those results with more qualitative descriptions. Because both the activity log and interview require that participants self-report their activities after the fact, they may fail to recall all activities of interest, or their beliefs about the study's purpose may bias their selection of events to report. Complementarily, VibeLog and MouseLog objectively record user activity to corroborate the results drawn from the activity logs and interviews and to reveal other interesting activities that might be missed by self-reporting. Logging low-level events has been gradually adopted as a means of understanding users' interaction activities. For example, Tak and Cockburn (2009) developed Window Watcher, a tool for visualizing windowing activities.

Note that before a participant switched to using the large display, VibeLog and MouseLog were deployed in the participant's normal computing environment for 5 working days to record window and mouse events. This data served as a baseline. When working on the large display, all four recording methods were employed to gather richer data. All participants reported that they were doing similar work during these 2 weeks (1 week in the normal computing environment, and the other on the large display). We did not do the activity log or daily interview during the 5 days of standard computing environment usage, as we felt it would impose undue demands on our participants who were already committing significant time to this study, whereas any new insights gained would likely be minimal because standard computing environments have already been well studied.

3.2. Results

Rather than report the results gleaned from each of our four logging methods separately, we instead present a more holistic analysis that extracts the most interesting findings from all our data.

Overall Activities

Figure 4 shows the categories of activities and the time logged on those activities across all the participants on the large display. The categories that accounted for most

FIGURE 4. Hours logged per major activity on the large display.

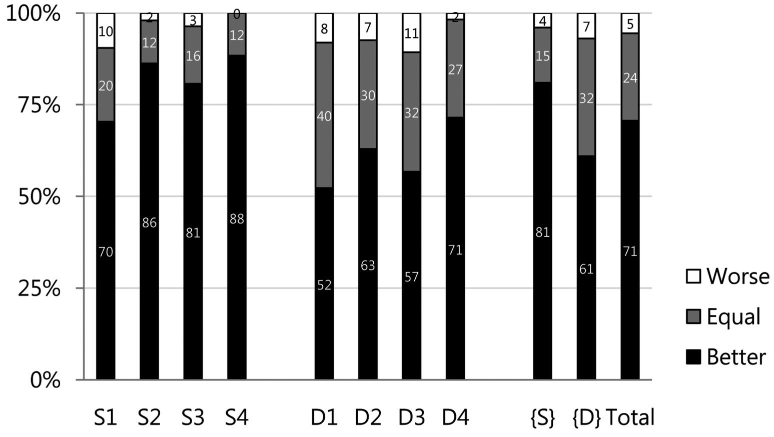
Activity	Hours	Ratio (%)	No. of Participants
Web browsing (e.g., Internet Explorer)	54.5	25.9	8
Word processing (e.g., Word)	48.3	23.0	8
Paper reading (e.g., Acrobat Reader)	42.0	20.0	8
E-mailing (e.g., Outlook)	21.8	10.4	8
Programming (e.g., NetBeans)	12.4	5.9	5
Spreadsheeting (e.g., Excel)	10.5	5.0	3
Slide-making (e.g., PowerPoint)	6.5	3.1	3
Online chatting (e.g., MSN Messenger)	5.0	2.4	6
3D modeling (e.g., Maya)	5.0	2.4	1
Graphic drawing (e.g., Photoshop)	4.5	2.1	2
Total	210.5	100	

of the hours were web browsing, word processing, and paper reading, with each of them constituting more than 20% of the total hours logged. Moreover, all eight participants performed web browsing, word processing, paper reading, and e-mailing tasks. These four tasks took up nearly 80% of the total hours logged.

Subjective Opinions

Figure 5 shows each participant's subjective opinion comparing the large display with their normal computing environments. Surprisingly, all participants overwhelmingly preferred the large display; for each participant, "Better" and "Equal" constituted more than 90% of the rankings, and more than 50% of the rankings were "Better."

FIGURE 5. Participants' subjective opinions of the large display compared to the traditional single- or dual-monitor configuration (S1, S2, S3, S4: four individual single-monitor participants; D1, D2, D3, D4: four individual dual-monitor participants; {S}: single-monitor participant group; {D}: dual-monitor participant group; Total: all eight participants).



All participants reported hoping to obtain more screen space for their computers. One dual-monitor user (D1) commented:

> When I was working on a 15-in. monitor, I thought a 19-in. LCD would be much better. When I switched to a 19-in. one, I found it was still too small. Now I am using two 18-in. monitors, but I still hope to get more screen space.

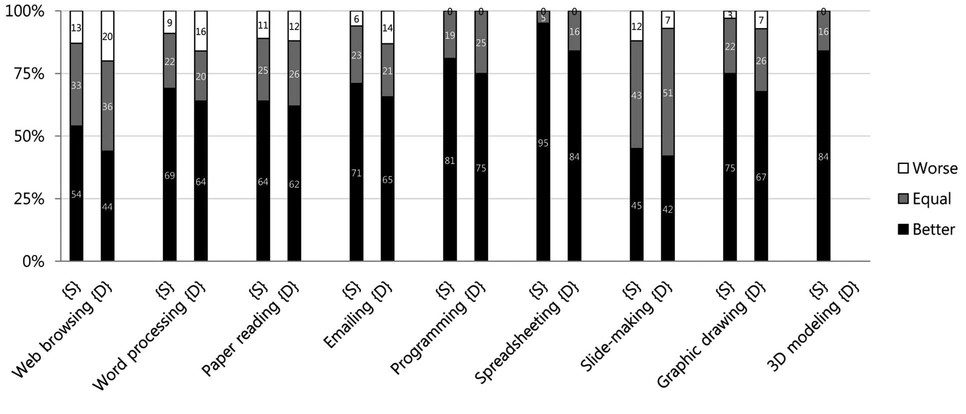
In particular, one single-monitor user (S4) always ranked the large display "Better" or "Equal." Her activities on the large display included web browsing, word processing, e-mailing, online chatting, and 3D modeling. She expressed a strong preference for working on the large display, commenting:

> The larger one is pretty better. It [the large display] offered me much flexibility of doing daily work. I can choose the optimal amount of screen space according to the task at hand.

Figure 5 also demonstrates that single-monitor users {S} have a stronger preference for using the large display than dual-monitor users {D}. Single-monitor participants rated the large display "Better" for 81% of hours, whereas dual-monitor participants did so only for 61% of the working time. The relatively smaller screen space on a single monitor may make single-monitor users feel like they are benefiting more from the large display than dual-monitor users.

Figure 6 breaks down the rankings by task. Both single- and dual-monitor users unanimously preferred the large display to their normal computing environments across all the tasks; for each task, "Better" and "Equal" constituted more than 80% of the rankings. In particular, the large display was always rated "Better" or "Equal" for programming, spreadsheeting, and 3D modeling.

FIGURE 6. Participants' task-wise subjective opinions of the large display compared to the single- and dual-monitor configurations ({S}: single-monitor participant group; {D}: dual-monitor participant group).



Note. **No dual-monitor participants did 3D modeling.**

Three single-monitor users (S1, S2, S3) and two dual-monitor users (D1, D3) performed programming tasks on the large display. All reported preferring the large display, because it allowed them to simultaneously view multiple windows, which they felt improved their workflow. A typical example was D1, who coded in Java on the large display. He kept a Java NetBeans window, an online help document, a dictionary application, and a Google search web page visible concurrently, where the Java NetBeans was the primary coding window, the online help document and the dictionary helped him to check unfamiliar Java functions, and the search page was used to find other information from the Internet. Keeping all these four windows visible eased access to any of them. In contrast, in his normal dual-monitor setup, he could keep just two of these applications visible, thus requiring frequent interchanging actions.

Data analysis using spreadsheet programs received the highest "Better" ranking (91%). Two dual-monitor users (D2, D4) and one single-monitor user (S2) analyzed data using Excel on the large display. All of them reported benefiting significantly from the larger display, because of the huge number of spreadsheet cells that could be kept visible simultaneously.

One single-monitor user (S4) used Maya to build 3D models on the large display. She rated the large display "Better" for more than 80% of her time. The large display could visualize several subwindows she had to use concurrently while working in Maya, thus significantly improving the 3D modeling workflow. The large display received the highest "Worse" ranking (20%) for web browsing tasks. Participants reported that rendering a large image was slow and there were some problems with the web browser (Internet Explorer), as some interface elements did not scale correctly to the large display. For example, when a web browser window was enlarged, the content inside was not enlarged as expected. Also, all the participants complained that it was difficult to minimize a window when it was close to the left or right side of the large display, because the "Minimize" icon on the window title bar was too small to accurately click.

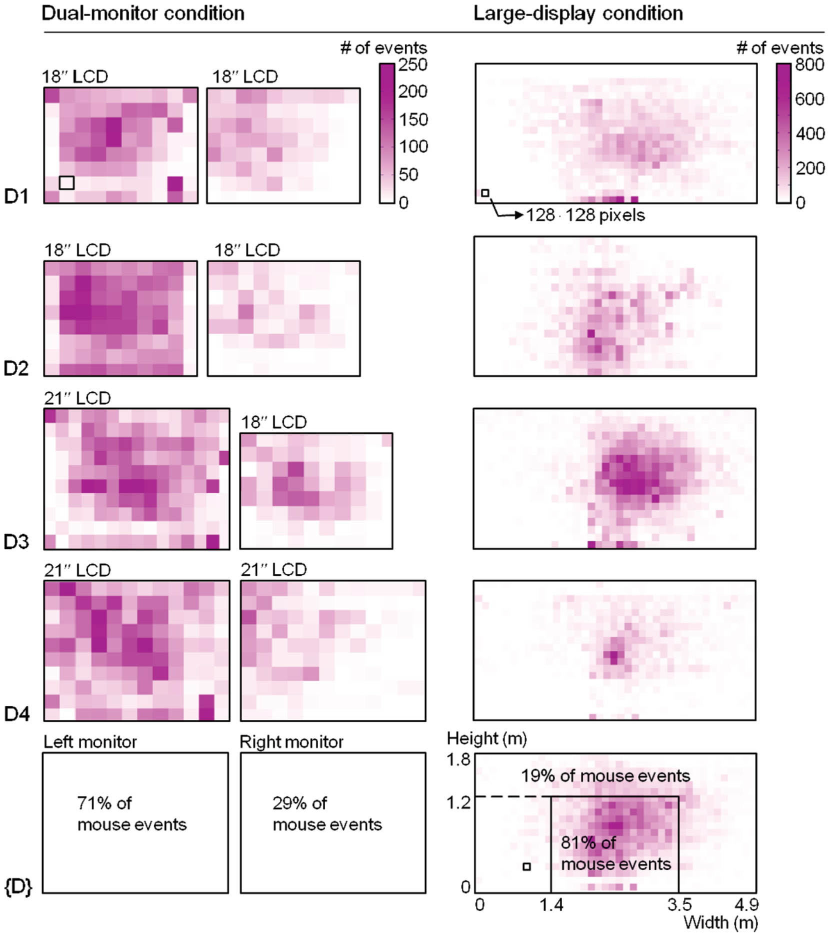
Window Management

Screen Real Estate Partition

All the dual-monitor participants reported mentally partitioning the screen real estate into *focal* and *peripheral* regions in both the dual monitors and large display when managing various windows, whereas single-monitor participants did so only on the large display. Generally, the focal region was used for primary windows, to which most interaction activities (e.g., programming, word processing, graphic drawing) were devoted over time, whereas the peripheral region was used for secondary windows, such as e-mail clients, instant messaging clients, and personal "to-do" lists.

Although the dual-monitor participants partitioned screen space in both the dual-monitor and large-display conditions, they did so differently in these two circumstances. The distribution of mouse events shows differences in location of both focal and peripheral regions (Figure 7). On dual monitors, participants performed

FIGURE 7. Mouse event distribution on dual monitors and the large display for dual-monitor participants (D1, D2, D3 and D4).



Note. **The bottom figure shows the overall percentage of mouse events for all the dual-monitor participants ({D}).**

more activities in one monitor than in the other: 71% of mouse events occurred within one monitor, and 29% occurred in the other. According to dual-monitor participants' reports, one monitor was usually used as the focal region and the other as the peripheral region. Due to monitors' physical bezels, spanning a window across two monitors suffered from visual discontinuity, so the primary activities were usually restricted to one monitor. However, when dual-monitor participants worked on the large display, 81% of mouse events occurred in the center region of the large display, and 19% occurred in the remaining "inverted-U" shaped area. The center region was used as the focal region, whereas the rest of the display was used as the peripheral region. This distribution might be explained by users' sitting positions. As all the participants sat in front of the horizontal center of the large display, the center part is the closest region to the users, making it convenient to view and interact with. Consequently, users were more likely to position focus-required work in this part and peripheral applications requiring less attention slightly farther away.

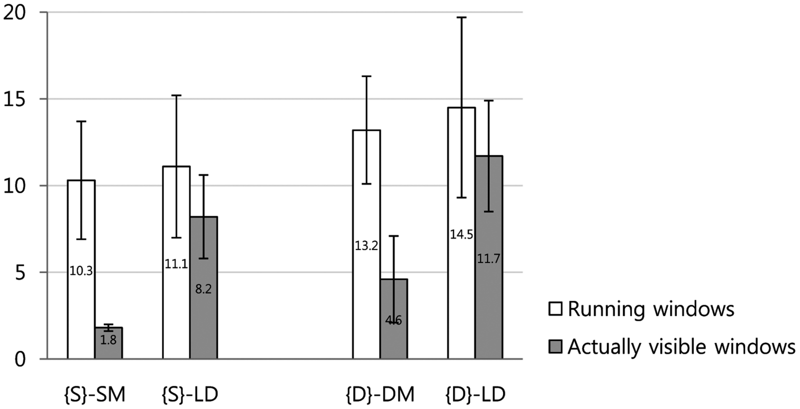
Window Visibility

To understand how users visualize windows, we defined *the number of running windows* as the number of windows that are open at a particular time, and *the number of actually visible windows* as the sum of the ratios of visible part of each opened window. For example, if there are three windows opened at a particular time: the first window is 100% visible, the second one is 50% visible (the other 50% is obscured by other windows), and the last one is 0% visible (completely obscured by others); the number of running windows and actually visible windows are 3 and 1.5, respectively.

Figure 8 shows the means of these two measures (each with its standard deviation) for single- and dual-monitor users, respectively. These numbers were measured every minute. As shown, the number of running windows did not undergo drastic changes for either single- or dual-monitor users when they switched to the large display. As expected, participants were doing similar work in both single/dual and large display environments.

In contrast, the number of actually visible windows increased drastically as users switched to the large display. The mean numbers of actually visible windows were 1.8 (single-monitor users) and 4.6 (dual-monitor users) in normal computing environments, respectively, with 8.2 (single-monitor users) and 11.7 (dual-monitor users) on the large display, respectively. All the participants reported that they tended to visualize more windows on the large display because it improved their workflow for complex, multiwindow tasks, such as programming and graphic drawing. They

FIGURE 8. Mean number (with standard deviation) of running windows and actually visible windows across single- and dual-monitor users ({S}: single-monitor participant group; {D}: dual-monitor participant group; SM: single-monitor condition; DM: dual-monitor condition; LD: large-display condition).



also pointed out that being able to visualize a large number of windows was a big advantage of working on the large display. It allowed them to conveniently process a large amount of information.

Long-Term Visible Windows

What kind of windows did users keep visible for a long time? To understand this question, we looked into the long-term visible windows in different conditions. We defined windows that remained 100% visible over 50% of the participant's total working time as long-term visible windows. As shown in Figure 9, all the participants used more long-term visible windows on the large display than in their normal computing environments: The mean numbers of long-term visible windows are 1.75 (large-display condition) and 0 (single-monitor condition) for single-monitor participants, and 3 (large-display condition) and 0.75 (dual-monitor) for dual-monitor participants.

According to Figure 9, long-term visible windows could fall into three categories: communication tools (e.g., e-mail client, online messenger), frequently used applications (e.g., file manager, web browser), and frequently checked information (e.g., weather forecast application, personal calendar). Most of the participants re-

FIGURE 9. Long–term visible windows.

	Single-Monitor Condition		Large-Display Condition	
Participant	No.	Applications	No.	Applications
S1	0	N/A	2	E-mail client (78%), web browser (55%)
S2	0	N/A	1	Online messenger (55%)
S3	0	N/A	2	Online messenger (63%), web browser (84%)
S4	0	N/A	2	E-mail client (87%), file manager (57%)
	Dual-Monitor Condition		Large-Display Condition	
Participant	No.	Applications	No.	Applications
D1	1	E-mail client (56%)	2	E-mail client (69%), file manager (77%)
D2	1	Online messenger (54%)	3	E-mail client (73%), online messenger (68%), word processor (63%)
D3	0	N/A	3	E-mail client (87%), web browser (57%), weather forecast application (95%)
D4	1	File manager (62%)	4	E-mail client (91%), online messenger (67%), file manager (61%), personal calendar (58%)

Note. The numbers in parentheses show the percentages of visible time over the entire work time (S1, S2, S3, S4: four individual single-monitor participants; D1, D2, D3, D4: four individual dual-monitor participants).

ported that these long-term visible windows eased communication with other people, facilitated the access to frequently used applications, and enabled them to be aware of peripheral information. Among all the participants, D4 kept most long-term visible windows, which included an e-mail client, an online messenger, a file manager, and a personal calendar. D4 reported that the e-mail client and online messenger were live communication channels, through which he kept in touch with his friends; keeping a file manager visible enabled him to quickly access desired files, which also facilitated copying and pasting file operations; the personal calendar served as a reminder of his daily work.

Window Arrangement

One major distinction on the large display compared with dual monitors was related to how users arranged their application windows. Due to the relatively limited and divided screen space on the dual monitors, participants usually invested little effort in arranging application windows in this condition. One dual-monitor participant (D2) reported:

> I just throw all the non-primary applications to the secondary monitor, and do not care much about the layout. If I need to use one of them, I maximize it by a simple click.

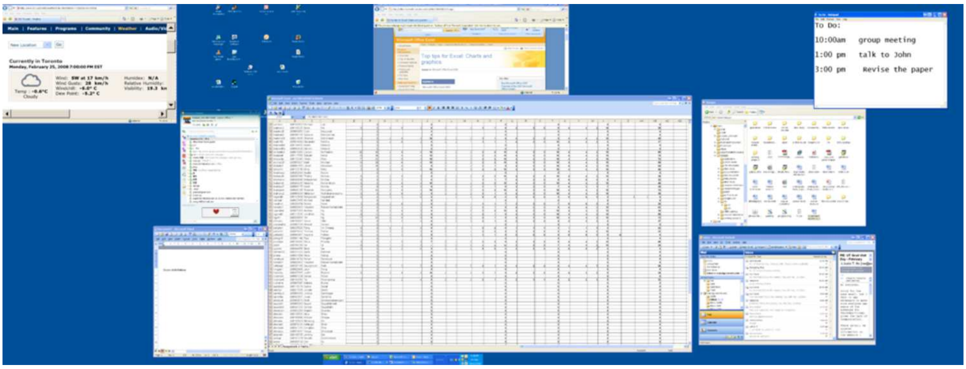
However, on the large display, participants tend to expend more effort to optimize the layout of application windows to improve workflow. One participant reported that at the beginning of each day, he spent nearly 5 min arranging the applications. He commented:

> It takes a certain amount of time at the beginning of each day, but worthwhile, because this optimal layout helps me greatly afterwards.

In particular, our postinterviews revealed that there are two common strategies for arranging windows.

- The first is to position the applications requiring interaction activities, such as word-processing programs, close to the center of the screen, whereas the applications only passively displaying information, such as a weather forecast window or a calendar, on the side or corner of the screen. Figure 10 shows a layout using this strategy captured during one participant's usage session. This strategy aims to facilitate the interaction with the applications beside the focal region.
- The second strategy is to arrange windows according to their relevance to the primary task. The more relevant, the closer the application is to the primary task window. For example, when using a drawing application, the canvas window was positioned in the center and surrounded by applications displaying reference materials. Other applications less relevant to the drawing process were placed farther away.

FIGURE 10. A screen capture of the large display from one participant's session, illustrating how the screen real estate is implicitly partitioned into focal and peripheral regions.



To Do:
10:00am group meeting
1:00 pm talk to John
3:00 pm Revise the paper

Interacting With Windows in the Peripheral Region

Another difference is how users interact with peripheral windows. On dual monitors, users just slightly turned their heads and bodies to work on applications on the secondary monitor, such as e-mail and online messaging clients. These applications were dragged into the primary monitor only when they would occupy interaction focus for a long time. This behavior occurred because two monitors were usually placed close to each other, requiring only a slight head or body movement to shift attention.

In contrast, when interacting with peripheral windows on the large display, users often dragged them from the peripheral into the focal region and seldom worked with them with their heads and bodies being turned. This action likely resulted from these peripheral windows usually residing on the side or corner of the large display, thus requiring plenty of head and body movements to shift attention. Also, concentrating on distant applications is difficult, which discourages working with them in the peripheral region. For example, e-mail client programs usually resided on the secondary monitor in the dual-monitor configuration. A user (D3) slightly rotated his head and body to respond to his e-mail client whenever a new message arrived. However, when working on the large display, he frequently moved the e-mail client back and forth between the focal and peripheral regions—dragging the e-mail client into the focal region to reply and then sending the application back to the previous location afterward. He commented:

> Although dragging the application back and forth between side and center regions is time consuming, it is still more comfortable than rotating my head and body to interact with it.

On the large display, because the peripheral region was distant from seating positions and viewed from more acute angles, pictures and text in this region were usually made magnified to ensure clarity and facilitate viewing. As seen in the screen shot (Figure 10), the picture within the weather forecast window was magnified by

200%, and the font size for the "to-do" list was enlarged to 25 pt from 10 pt, which was usually used on a normal desktop screen. A similar enlargement of peripheral contents did not occur in the dual-monitor condition.

Window Operations

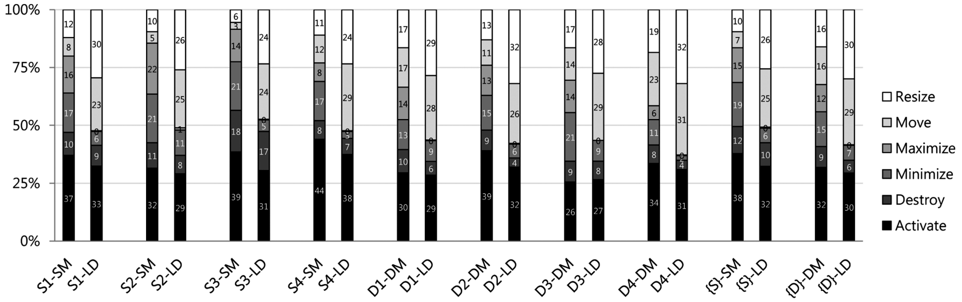
All the window management activities are executed through basic window operations. To gain a deeper insight into unique window management behavior on the large display, we compared basic window operations across different conditions.

Figure 11 shows the distribution of different window management operations for each person. One common characteristic across all the participants is that moving and resizing operations constitute much higher percentages of total operations on the large display than in their normal computing environments. For single-monitor users, the mean percentage of moving plus resizing operations was 51.5% ($\sigma = 2.4$) on the large display and 16.7% ($\sigma = 6.1$) on the single monitor; for dual-monitor users, it was 58.5% ($\sigma = 2.4$) on the large display and 32% ($\sigma = 9.7$) on the dual monitors. Whenever a new application was opened, the following operation was to optimize its position and size. Moreover, frequently switching applications back and forth between the peripheral and focal regions also causes more moving operations on the large display.

Another interesting finding is that dual-monitor participants performed more window moving and resizing operations than single-monitor participants in their normal computing environments. However, when using the large display, both single- and dual-monitor participants performed similar percentages of moving and resizing operations, which were much higher than those in their normal environments.

In contrast to moving and resizing, maximizing and minimizing operations constituted a much lower percentage of actions on the large display. Only one (S2) out of eight participants ever maximized windows on the large display. S2 did so to show a Google map. Other participants reported never spanning a window across the entire large-display surface. When they needed to visualize windows containing rich

FIGURE 11. Distribution of window-management operations per participant (S1, S2, S3, S4: four individual single-monitor participants; D1, D2, D3, D4: four individual dual-monitor participants; {S}: single-monitor participant group; {D}: dual-monitor participant group; SM: single-monitor condition; DM: dual-monitor condition; LD: large-display condition).



information, roughly 60% to 70% of the entire display surface was typically sufficient. This is a particularly interesting finding in that it may indicate the upper bound of window size that users are comfortable working with. Minimizing operations were often performed in single- or dual-monitor conditions to save screen space, but it was rarely used on the large display due to the ample available space.

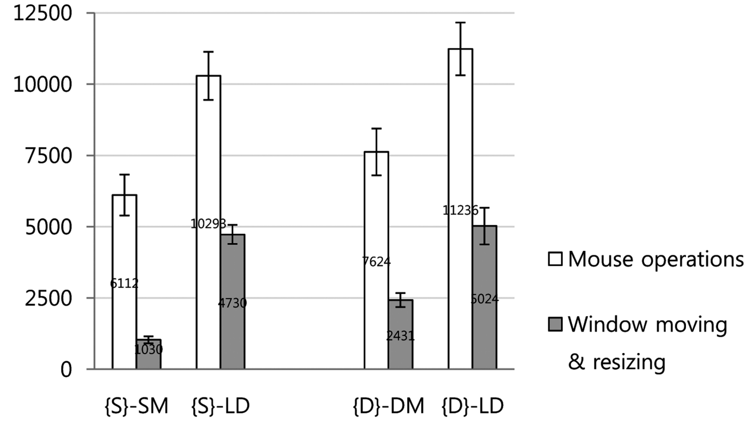
Mouse Operations for Moving and Resizing Windows

On current window management systems (e.g., Windows XP or Mac OS), users move and resize windows by clicking and dragging their title bars, borders, or corners. As users moved and resized windows more frequently on large displays, we might conjecture that they also performed more mouse clicking and dragging operations. Figure 12 confirms it. A general trend illustrated is that users tended to perform more mouse operations when switched from single/dual monitors to the large display. For single-monitor users, the mean number of mouse operations on the large display was 68% higher than that on single-monitor conditions, whereas for dual-monitor users, the number of mouse operations on the large display was 47% higher than in dual-monitor conditions.

Echoing the finding shown in Figure 11 that window moving and resizing operations took higher percentages than other window management operations, Figure 12 shows that the actual number of window moving and resizing operations in the large-display situations was also higher than that in single/dual-monitor environments.

The drastic increase of window moving and resizing operations leads the rise of the number of mouse operations. For single-monitor users, the mean number of mouse operations increased by 4,181 from single-monitor to large-display condition, whereas the mean number of window resizing and moving operations increased by 3,700, indicating that around 88% of mouse operation increase was caused by window

FIGURE 12. Mean number (with standard deviation) of mouse operations and window moving & resizing operations across all the participants for 5 days ({S}: single-monitor participant group; {D}: dual-monitor participant group; SM: single-monitor condition; DM: dual-monitor condition; LD: large-display condition).



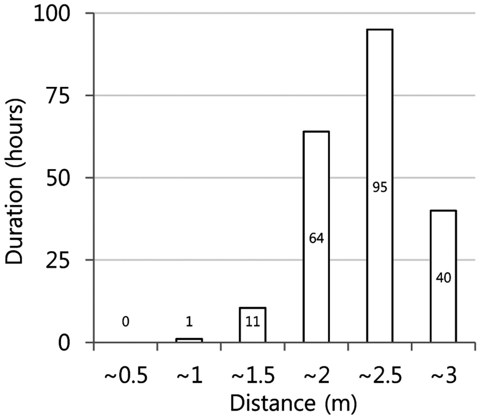
moving and resizing operations. Similarly, around 72% of mouse operation increase was caused by window moving and resizing operations for dual-monitor users.

Seating Location

As the optimal seating location when working with a large display is unclear, participants were allowed to freely adjust their seating positions during the study. Wheels were mounted on the table and chair to enable mobility. All of the participants reported sitting in front of the horizontal center of the display because they felt that their visual fields covered most of the screen space from this position. Figure 13 shows the durations of all the sitting distances from the display, which were self-reported by reading markers on the floor and filling the activity log. As shown, the distances ranged from 0.9 m to 3.0 m, with more than 90% of data between 1.5 m to 3.0 m. Most relevant is that participants sat within 2.0 m to 2.5 m of the display nearly 50% of the time. We also investigated the sitting distances across days. The mean sitting distances across participants for Days 1–2 and Days 3–5 were 1.8 m ($\sigma = 1.2$) and 2.5 m ($\sigma = 0.2$), respectively. Of interest, sitting distance varied substantially during the first 2 days of the study but kept relatively constant in the last 3 days. As reported by users, they were experimenting different sitting locations at the beginning of the study. However, they less frequently changed the location during the last 3 days, indicating that they had settled on preferred positions.

When the participants worked in their normal computing environments, the distances to the screen ranged from 0.75 m to 0.85 m. All participants reported sitting slightly farther away when switching to the large display because they wanted to view more of the screen at once. One single- and one dual-monitor participants (S2 & D3) reported that sitting too close (less than 1.0 m) to the large display made them feel as if they were sitting up close to and facing a wall, which was very uncomfortable. On the other hand, sitting too far away could also make viewing details on the display difficult. It seemed that a distance of 1.5 m to 2.5 m was the preferred range for

FIGURE 13. Time spent at different distances to the large display.



most of the participants, at which they could clearly perceive the content on the large display and their visual field covered sufficient screen real estate.

3.3. Discussion

Benefits of Large Displays

As shown in Figure 6, despite the fact that the Windows XP operating system was designed for a normal desktop screen, participants still overwhelmingly preferred a large display to their normal computing environment (i.e., single- or dual-monitor). Our results indicate that a large display can provide several key benefits:

Scattered Information-Processing Tasks

A large display benefits scattered information-processing tasks by enabling the simultaneous display of multiple windows. Scattered information-processing tasks are tasks in which information is scatted across multiple windows. Information workers need to collect, search, and compare information across multiple windows. Examples include programming and slide making. In a programming task, although the primary working window is the coding application (e.g., Java NetBeans), the user needs to search for and check relevant information for coding in other windows (e.g., a function name in a Java document window). In a slide-making task, the user needs to collect materials from different windows such as web pages and picture folders, and edit/organize them within the slide-making application (e.g., Microsoft PowerPoint). As most of the relevant windows are visualized on the large display, information scattered in different windows are visible simultaneously. Our study results (Figure 8) confirmed this trend: Users tend to visualize and interact with multiple windows when switching to use a large display.

As multiple windows are visualized, information workers can easily collect, search, and compare the information without interchanging windows. In contrast, they need to frequently switch back and forth among multiple windows in single- and dual-monitor conditions because only one or two windows are totally visualized due to the small screen real estate. By reducing the number of window interchanging operations, a large display benefits the scattered-information processing tasks.

Previous research (Czerwinski et al., 2003; Simmons, 2001) has shown that a large display (albeit theirs was of lower resolution and much smaller than the display used in our study) outperforms a small display for complex, multiapplication office work. Complementarily, our study shows that users intentionally optimize window layout to facilitate scattered information processing to improve their workflows.

Multiscale Navigation Tasks

Another obvious benefit is related to applications that contain rich digital information and require users to perform frequent navigation operations in both the lateral and depth directions to search for desired information. For example, a user pans and zooms in/out on a large map to locate a target; a data analyst scrolls up/down and zooms in/out on a big spreadsheet to search for desired data. This type of task is referred as a multiscale navigation task. As a large amount of information is visualized simultaneously on a large display, users tend to search for desired data via physical navigation (e.g., changing eye positions, rotating heads), as opposed to using virtual navigation (e.g., panning and zooming in/out). Reducing the workload of virtual navigation leads to better performance on a large display for such tasks. For example, one participant (D2) expressed a strong preference for analyzing data in Excel files on the large display, because it could show all the columns. Unlike working on the large display, he had to frequently scroll left and right, up and down in his usual dual-monitor setup because only partial information was visible.

Awareness of Peripheral Applications

Being aware of peripheral applications could improve work productivity (MacIntyre et al., 2001). A large display can fully take advantage of users' peripheral vision by showing numerous peripheral applications. For example, as shown in Figure 10, a user opened a weather report, to-do list, e-mail client, and online messaging client in the peripheral region. He could maintain awareness of these four peripheral applications while working on the primary task. In contrast, in his usual single-monitor environment, due to the limited screen real estate, only the major application window was fully visible. Peripheral applications were usually obscured. The study results (Figure 9) further indicate that large display users tend to keep three kinds of windows visible most of the time: communication tools (e.g., e-mail client, online messenger), frequently used applications (e.g., file manager, web browser), and frequently checked information (e.g., weather forecast application, personal calendar). Designing applications or widgets that are visible permanently and display these three kinds of information could benefit large-display users.

Immersive Experience

Tan et al.'s (2004) work shows that a large display outperforms a normal monitor in 3D navigation tasks because of the immersive experience generated by the large display. Although users' personal desktop work is not typically composed of 3D navigation tasks, participants reported that a large display engaged them more in the daily work than single or dual monitors. When sitting in front of the large display, they felt "surrounded" by the task. This feeling helped them focus attention on the task, especially when they were performing attentive work such as proofreading and coding. In addition, the large display might provide some ergonomic benefits. In the single-monitor condition, the user's head and eyes are restricted to the limited screen space, which might easily cause fatigue when working for a long time. However, users felt more relaxed using a large display because they could freely adjust hand and body positions while maintaining a good view of the display.

Challenges of Managing Windows on a Large Display

Despite the overwhelming preferences of working on a large display, challenges arise when users switch from a traditional working environment to a large display. As plentiful of digital information and numerous windows are visualized, large-display users exhibit distinct patterns in managing windows. However, current window management systems (e.g., Windows XP) are designed for a normal-sized monitor, which seem too rigid for a large display. Based on our study results, we would like to suggest enabling functions to suit large-display users' unique window management behaviors as follows:

Multiwindow Operations

Large-display users tend to spatially arrange multiple windows into various layouts to convey semantic meanings. For example, software developers spread multiple help documents around the code-editing window; windows not being used are stacked at the side for later reference. However, current window management systems are not sufficiently flexible to support these multiwindow operations. They typically allow users to operate on only a single window at a time. Thus, arranging multiple windows on a large display becomes tedious and time-consuming. To alleviate this problem, a window manager for a large display should offer appropriate multiwindow operations.

Flexible Window Arrangement

Office workers enjoy greater flexibility in organizing paper documents with two hands in the physical world than with a single cursor in the digital counterpart. They can quickly move documents in and out of the center of a desk for information processing. Relevant files are interleaved for cross-referencing, and documents not being used are roughly stacked to convey subtle information. To mimic these paper-organizing behaviors, new window management systems might offer a more natural and flexible approach for arranging windows. For example, a large-display oriented window management system should allow users to freely adjust the order of overlapping windows or to quickly spread/pack windows.

Facilitation of Moving and Resizing Windows

Our basic window operation analysis showed that users moved and resized windows more frequently on our large display than on single or dual monitors—moving and resizing accounted for more than 50% of window operations on the large display and less than 30% on single or dual monitors (Figure 11). The numbers of moving and resizing operations increased by 359% (for single-monitor users) and 162% (for dual-monitor users) when users switched to the large display, which accounted for 88% (for single-monitor users) and 72% (for dual-monitor users) of the moue operation increases (Figure 12). Unfortunately, many large-display users who participated in our study were not satisfied with current window moving/resizing operations; they complained that the thin title bar and the tiny corner of a window was difficult to point and drag, especially when windows were located on the right and left sides, far from their focus area. Holding the mouse button and dragging a long distance was tended to be prone to fatigue. A lighter-weight moving and resizing approach might significantly benefit users.

One distinct pattern of large-display window management activities was that users frequently switched windows between center and side regions for changing work focus. Easing this switching process could also significantly improve window management efficiency.

In-Place Command Execution

In the traditional window management systems such as Windows XP or Mac OS, some window operations are executed by clicking buttons on a window, such as Close and Minimize buttons on the title bar. However, accurately clicking a small button distant to a user's sitting position is error-prone on a large display, as reported by participants (Section 3.2). Moreover, because these buttons are usually fixed on the corners of a window, clicking them may require extra cursor movements. To alleviate these problems, new systems should allow users to execute commands in-place and avoid high-precision clicking.

Applicability to Other Display Settings

The large display in this study consisted of 18 projected screens. Another approach to creating a large high-resolution display is to tile multiple LCDs. However, it introduces physical bezels between display panels, which affect users' behaviors and performance (Ball & North, 2005; Bi, Bae, & Balakrishnan, 2010). In this article, we focus on seamless displays.

As display technologies develop, it is possible that in the near future users might work on small-sized but very high resolution displays. Such a display will also provide a large visualization capacity. For example, a 1.9 m wide × 0.7 m high display with 80 dpi has the same resolution (6144 × 2304) to the 4.88 m wide × 1.83 m high display used in our study. Are the findings obtained in our study applicable to other settings?

The visual effect of a display depends on sitting distance. With different sitting distances, displays with different sizes might have similar or even identical visual effects. For example, given the preferred sitting distance of 2 m for our large display, the visual angle of viewing a single pixel (0.079 cm wide × 0.079 cm high) is 0.023°. It is equivalent to view a 0.0316 cm wide × 0.0316 cm high pixel (80 dpi) with the sitting distance of 80 cm, which is a common sitting distance for current desktop monitors, according to our prestudy survey. Therefore, viewing the large display in our study at the sitting distance of 2 m has the same visual effect with viewing a 1.9 m wide × 0.7 m high display (80 dpi, 6144 × 2304) at the sitting distance of 80 cm; objects on both displays would be identical to users because of the same viewing angles. Thus, the findings obtained on the large display in our study are likely to be applicable to such a 1.9 m wide × 0.7 m high display.

From another point of view, the main findings in our study, including the benefits of large displays and challenges of managing windows, are caused by two factors: the large visualization capacity of the display and the inconvenience of single-window-based window management operations provided by current operating systems (e.g., Windows XP or Mac OS). Regardless of the physical size of a display, these two factors persist for high-resolution displays. It is likely that the findings obtained in this study are still applicable even if displays are physically smaller. We would like to further investigate this issue in the future.

4. WALLTOP: FLEXIBLY MANAGING WINDOWS ON A LARGE DISPLAY

To meet the aforementioned challenges, we design and implement a set of new interaction techniques to facilitate managing overflowing windows on a wall-sized display. These techniques are also carefully designed to ensure high compatibility with traditional window operations, which have existed for more than two decades with little change. Replacing traditional operations with an entirely new metaphor might require excessive learning efforts. We therefore argue that new window operations should coherently co-exist with existing ones. Accordingly, our new techniques are designed for augmenting, rather than replacing, the traditional operations.

4.1. WallTop Interaction Techniques

Fringe

The Fringe is the virtual boundary enclosing selected windows (Figure 14). It was designed to ease multiwindow operations. The longitudinal study revealed that large-display users visualized more windows and arranged them in different layouts according to the task at hand. For example, when making a PowerPoint slides, the

FIGURE 14. (a) Overlapped windows; the Fringe is activated when the cursor approaches the border of the windows (b), and deactivated when the cursor moves away (c).

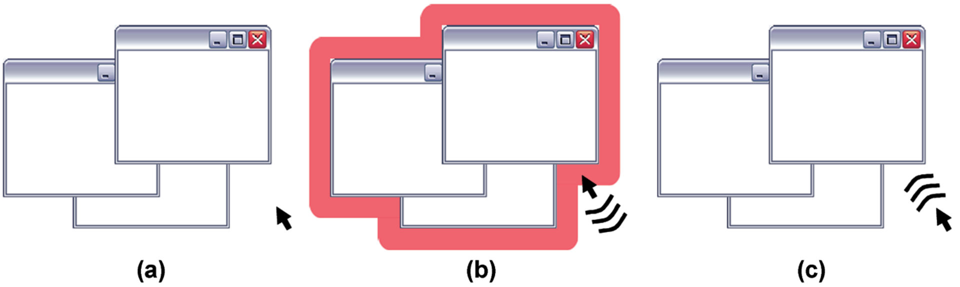
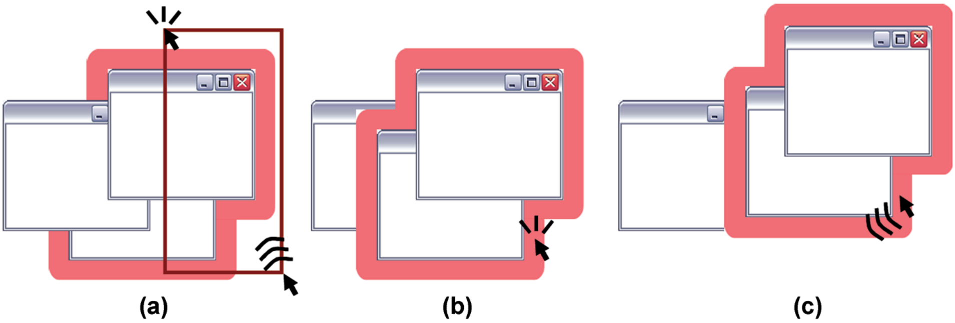


FIGURE 15. (a) Drawing a rubber band on multiple windows, (b) clicking on the Fringe of the selected windows, (c) moving them by dragging the Fringe.



PowerPoint application window was placed at the center, surrounded by other material windows. When working on tasks involving multiple windows, users tended to place them close to or overlapped with one another. It allowed the users to easily access relevant information across different windows. Inspired by these findings, overlapped windows by default are recognized as a group. The Fringe around them fades in when the cursor is approaching the border of the group (within 3 cm from any window of the group; Figure 14a and 14b), and fades out when the cursor is moving away (7 cm away from every window in the group; Figure 14c). The user can also manually select multiple windows by drawing a rectangular rubber band (Figure 15)—windows intersected by the rubber band rectangle are selected. Selecting one window is achieved by directly clicking anywhere inside it (Figure 16). In any situation, the Fringe is adjusted to exclusively enclose the selected windows. In the current prototype, the Fringe is set to 3 cm wide to ensure easy selection.

Once multiple windows are selected, users can simultaneously move and resize them: Dragging a convex corner on the Fringe resizes every window enclosed (Figure 17), whereas dragging any other position on the Fringe moves windows simultaneously (Figure 15c and Figure 16c). Users can freely operate on an arbitrary

FIGURE 16. (a) The Fringe activates for multiple windows, (b) selecting a window by clicking inside it, (c) moving it by dragging the Fringe (or the title bar).

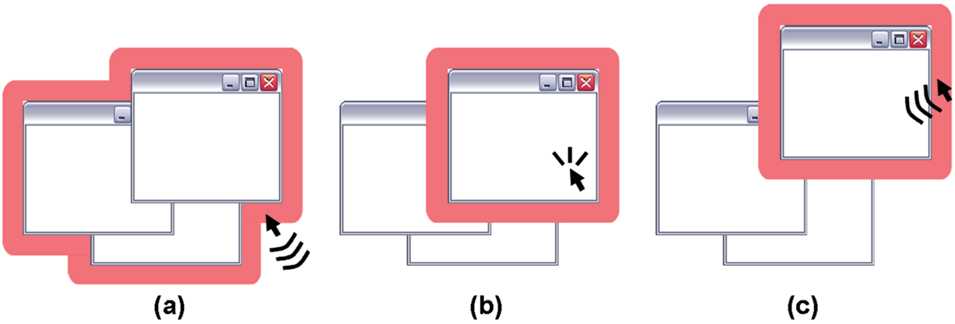
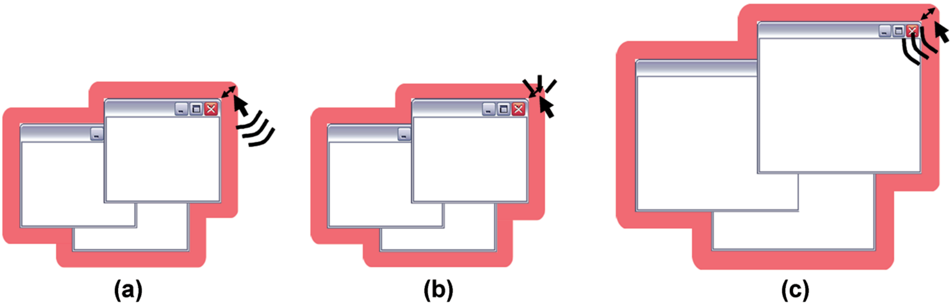


FIGURE 17. Dragging one of the convex corners of the Fringe that encloses selected windows resizes all of them at once.



number of windows with the Fringe, no matter whether they are spatially aggregated or not. Because the Fringe is much wider than the border of each window as well as its title bar, selecting, moving, and resizing windows with the Fringe are much easier than with the window border or title bar.

Jab-to-Lift

Office workers easily arrange physical paper documents with two hands. They often use the dominant hand to insert documents underneath others while lifting up the corner of those other documents with the nondominant hand. Inspired by this lifting and inserting behavior, we implemented the Jab-to-Lift operation as a more flexible supplement to the normal window dragging and dropping activity, which always places a selected window on top of the other windows, thus enabling some of the power of the real-world two-handed interaction but with just a single cursor.

Jabbing—performing a scratch motion by quickly and repeatedly moving the selected windows vertically, horizontally, or diagonally (Figure 18a) across the boundary of other windows animatedly shifts (folds) the closest corner of the static ones (Figure 18b). The selected windows can be inserted beneath folded ones (Figure 18b). The folded windows spring back to their original shape when the mouse button is

FIGURE 18. (a) Jabbing a selected window at the boundary of other window to fold its corner, (b) inserting the selected window under the folded window, (c) unfolding the window with the mouse button released.

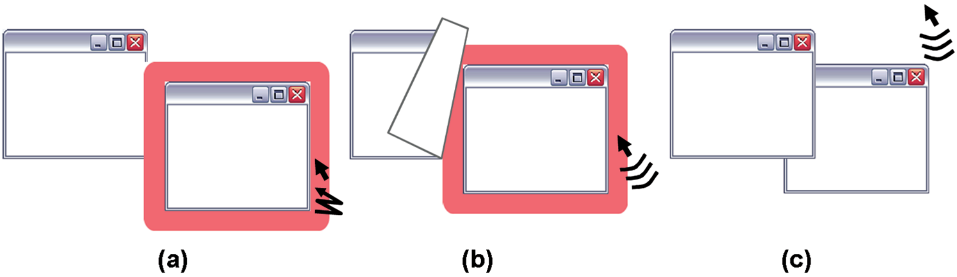
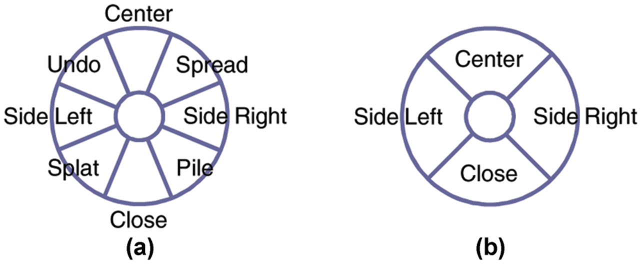


FIGURE 19. (a) Multi–, and (b) single–window marking menus.



released (Figure 18c), or the selected windows are moved away (at least 6 cm away from folded windows). Both folding and unfolding are visualized with smooth animation to avoid confusion.

Folding windows have been used to reveal underlying contents (Beaudouin-Lafon, 2001) or to facilitate copy/paste operations (Chapuis & Roussel, 2007; Dragicevic, 2004). We employ this concept in conjunction with the jab gesture that explicitly conveys the user's intention of interleaving a selected window(s) with other window layers. Traditional window operations only allow users to drag a window over others, which is too rigid to appropriately support arranging windows on a large display. Jab-to-Lift offers more flexibility in changing the overlapping order of windows on a large display using just one cursor with a natural and easily understandable metaphor of peeling back and inserting physical paper documents.

Multi- and Single-Window Marking Menus

Many of the interactions on WallTop are performed via right-click marking menus (Kurtenbach, 1993), which are activated when the user right clicks on the Fringe (Figure 19). Dragging the cursor with the right-button down beyond the inner circle highlights a color-coded command wedge (Figure 20a, Figure 21a, Figure 22a, Figure 23a). Subsequently releasing the right-button executes the corresponding command. We implemented two types of marking menus on WallTop. The multiwindow

FIGURE 20. Choosing Spread item on the marking menu (a) arranges the selected windows (b) in a tiled layout where the window positioned on top is centered (c).

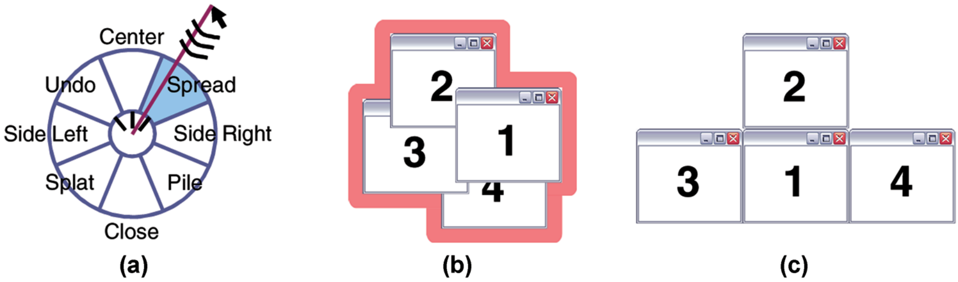


FIGURE 21. Choosing Splat item on the marking menu (a) arranges the selected windows (b) in a radiated configuration from the group center (c).

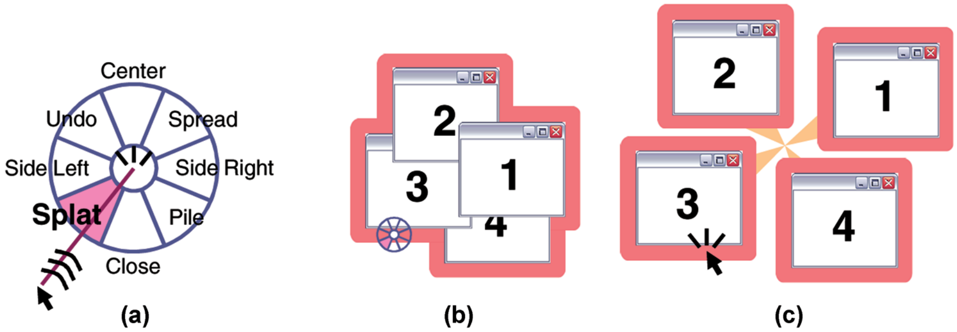


FIGURE 22. Choosing Pile item on the marking menu (a) arranges the selected windows (b) in a tidy pile layout (c).

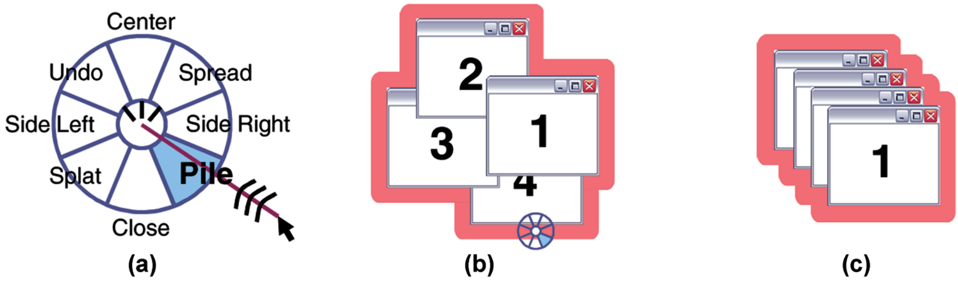
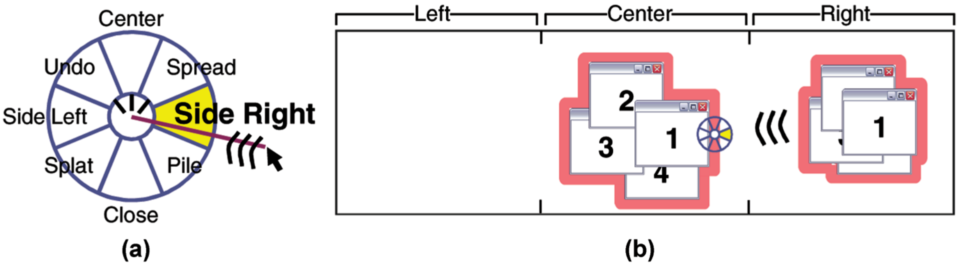


FIGURE 23. Choosing Side Right item on the marking menu (a) shifts the selected windows to the right (b). Windows are automatically packed at the side regions.



marking menu (Figure 19a) is invoked if the Fringe encloses more than one window, otherwise the single-window marking menu (Figure 19b) is activated.

Marking menus are designed to address our fourth design goal in Section 3.3—executing commands in-place and avoiding precise selection. Users can activate marking menus anywhere on the Fringe without moving the cursor to a specific small area (e.g., the corners or the title bar of a window[s]). Because menu items are selected by cursor movement direction, precise pointing is not necessary. We use the Fringe area only for menu activation because right clicking on other regions of a window might conflict with the traditional window functions. For example, in the Windows XP OS, right clicking on the title bar invokes a window menu and right clicking inside a window invokes an application menu.

We did not employ left-click marking menus to avoid conflicting with moving a window(s): Left-button clicking and dragging on the Fringe moves or resizes selected windows. We considered left-button double-click marking menus as another alternative but only kept it as an optional setting based on the results of our first-round usability test (see Section 4.3).

Multiwindow Operations

The marking menu on the Fringe establishes a framework for performing a variety of multiwindow operations. This is used to make typical selection-and-action phrases: The former (the Fringe) visually indicates the selected targets and the latter (the marking menu) provides command options. A set of multiwindow operations are designed to mimic how office workers organize paper, such as piling, packing, and spreading (Sellen & Harper, 2003).

Spread. Office workers often spread documents out during active reading or comparison reading. The primary document is placed at the center and surrounded by supporting ones. Large display studies also show such spreading behaviors are common when a task involves multiple windows (e.g., programming or active reading). Invoking *Spread* command in the marking menu quickly arranges selected windows in a tiled layout (Figure 20). Once executed, the top window of the selected windows is placed at the center and surrounded by other windows.

Splat. Splatting is an effective tool for selecting an object from a group of occluding ones (Apple Expose; Baudisch et al., 2003; Ramos et al., 2006). Invoking the *Splat* command in the marking menu temporarily separates windows apart (Figure 21). Each separated window is connected to the group center with its ray fan that indicates the splat status. The subsequent mouse left-button click (Figure 21c) brings separated windows back to original positions with the clicked one on top. Splat uses a force-directed layout algorithm (Ramos et al., 2006) where each participating window is slid along the ray starting from the group center to its center by the distance proportional to its size. This algorithm roughly preserves the spatial relationship of the selected windows.

Pile. Document piles are commonly seen on office workers' desks (Sellen & Harper, 2003). Piling has been used as an effective way of organizing files and desktop icons (Agarawala & Balakrishnan, 2006; Mander et al., 1992). Executing the Pile command via the marking menu piles up selected windows with their title bars uncovered to show their names (Figure 22). Once piled, leafing through the windows by moving up or down the mouse wheel brings a window from bottom to top or top to bottom, enabling quick window browsing.

Center/Side. Users frequently switch windows back and forth between center and side areas of a large display. The former is the focal region where most mouse and keyboard events are triggered, whereas the latter is the peripheral region where windows not being used currently are usually positioned. For example, an e-mail client program is dragged into the focal region from the periphery when the user is to check messages and is later on repositioned to the peripheral region to make space in the center area for other windows used for the next task to occupy. Large-display users reported that dragging windows back and forth between the focal and peripheral regions was annoying, mainly due to the fatigue caused by holding the mouse button for a long time and traveling a long distance. Center and Side Left/Right commands are designed to ease this frequent switching process. Invoking the Center command on the marking menu brings selected windows to the center area of the display, and restores the previous window layout, whereas Side Left and Side Right shift windows to empty space at the left and right areas of a large display, respectively (Figure 23). On the marking menu, Side Left and Side Right commands are placed at west and east wedges to be consistent with window moving directions.

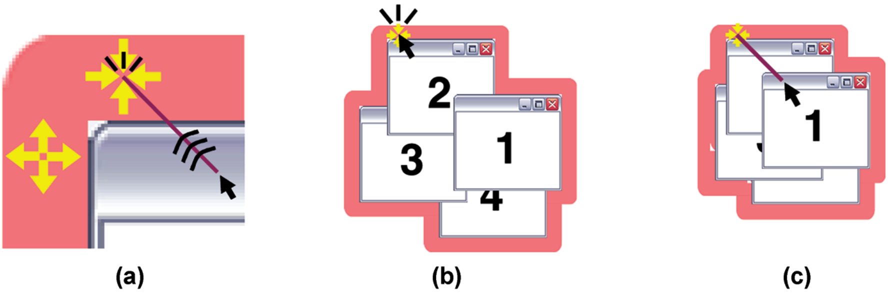
Scalable Fabric (Robertson et al., 2004) allows users to quickly move windows from center to side by clicking the window's minimize button and from side to center by clicking anywhere inside the window. On a large display, however, users still click the minimize button to minimize windows and click on windows placed in the peripheral regions to operate on the applications. Thus, Scalable Fabric's position switching convention can cause operation conflicts. Our use of Center and Side Left/Right actions with marking menus solves this conflict and coherently integrates with existing operations while providing users with more flexibility—users can choose which side (left or right) to place windows.

Undo/Close. Any window arrangement operations executed previously can be reversed by using the Undo command; the Close command closes the selected windows.

Pack/Unpack. Packing files is another common behavior of office workers. After reading some paper documents, they lazily pack documents into a messy pile. Different from a tidy pile, a messy pile still subtly conveys the initial spatial information of each document, which might ease later retrieving action (Agarawala & Balakrishnan, 2006). We design Pack and Unpack icons on the upper left corner of the Fringe for continuously packing and unpacking windows (Figure 24a). These two operations are designed as icons because they require subsequent continuous mouse movement to determine operation parameters. It is hard to fit such a combination operation into a marking menu.

Clicking the Pack icon and dragging in any direction tightens the selected windows (Figure 24b and c). The dragging distance determines how tightly windows are

FIGURE 24. (a) Pack (up) and Unpack (left) icons; clicking Pack icon (b) and dragging in any direction continuously tighten the selected windows (c).



packed. Unpack works in reverse—clicking it and dragging separates windows apart. Automatic Pack and Unpack functions are also enabled to support large-display users' common behaviors of packing documents on the side region and unpacking windows in the focal region. When moved from center to side, windows are automatically packed (Figure 23b), and unpacked when moved from side to center.

Single-Window Operation

Right click on the Fringe enclosing only one window triggers the single-window marking menu, which contains four commonly used commands (Figure 19b). Similar to those on the multiwindow marking menu, these options are used to center, side, or close the selected window.

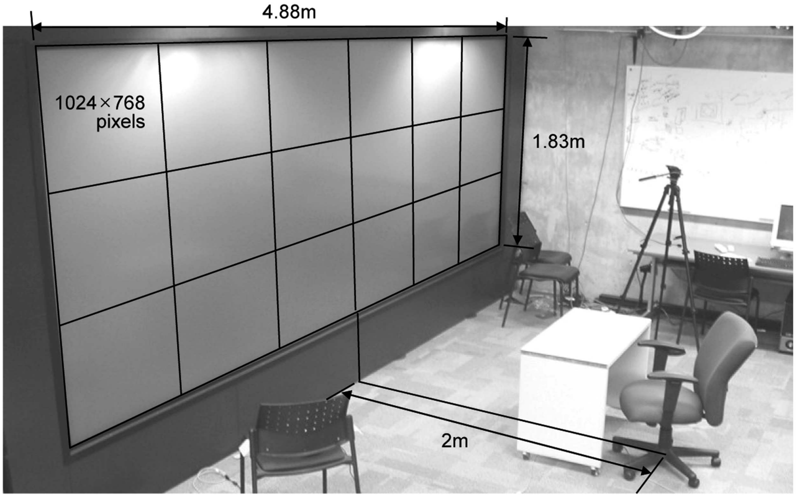
4.2. Implementation

We integrated the new interaction techniques with traditional window operations (Windows XP) into a prototype called *WallTop*. The current WallTop is a proof-of-concept system (using images of windows instead of real applications) written in C++, OpenGL, and GLUT and running on a 4.88 m wide × 1.83 m high large display via the Chromium system (chromium.sourceforge.net). The entire display consists of 18 projectors arranged in a 6 × 3 configuration, with a total resolution of 6144 × 2304 pixels (1024 × 768 pixels each; Figure 25). WallTop is designed to facilitate window management on a large display that is big enough to simultaneously visualize multiple windows. We demonstrate the interaction techniques on this wall-size display as a proof-of-concept. They could be implemented on large displays of various sizes with at most minor changes.

4.3. First-Round Usability Test

We conducted two rounds of usability tests of WallTop on a wall-sized display. The first-round test was to discover potential usability problems and refine interaction techniques.

FIGURE 25. Specification of WallTop and usability test setup.



Test Design

Six participants (two female, four male; 18 to 34 years old), participated in videotaped think-aloud sessions lasting an hour each. After a 20-min introduction session, participants tried out the new interaction techniques. In particular, they were instructed to compare the right click and left double click for activating marking menus. Participants sat 2 m away from the center of the display (Figure 25), which is a normal sitting distance for the large display reported in the longitudinal study (Section 3.2).

Results and Observations

Subjective data show that most users prefer right click to invoke marking menus to left-button double click. The average scores of right-button click and left-button double click are 4.2/5 and 3.3/5, respectively. Five of six participants reported a preference for the right-click marking menu. Three users explained that right-click marking menu required only one click action, whereas double click required two. One participant said he was familiar with right-click marking menu because of the familiarity with the Maya 3D modeling and animation system. The only participant who preferred double click explained that he disliked pressing and holding the right button with the middle finger. Because the majority of the participants preferred the right-click marking menus, we chose it as the default for WallTop. Double-click marking menu is kept as an option that users can enable via a systems setting dialog box.

We initially designed only one icon for both packing and unpacking to distinguish them by cursor dragging directions—clicking the icon and dragging down packs the selected windows while dragging up unpacks them. However, we observed usability problems with this approach across all the participants. All six participants reported that they disliked using one icon for both packing and unpacking. Also, five participants said they were confused with using direction to distinguish two opposite functions, even after 1 hr of practice. To alleviate this problem, we finally designed two separate icons for these two operations.

We also determined several design parameters via this usability test. For example, the Fringe is 3 cm wide and fades out when the cursor is 7 cm away, and folded windows spring back when the cursor is 6 cm away. We iteratively found that these values ensure easy selection of windows on their Fringe and prevent them from being deselected unexpectedly in the middle of operations.

4.4. Second-Round Usability Test

The second round test was to formally evaluate the new interaction techniques.

Test Design

After refining the techniques according to the first-round feedback, we conducted a user study to evaluate WallTop. In particular, we aimed to determine how quickly and correctly users learn our new interaction techniques and how much these techniques improve the efficiency of managing windows on a large display.

Eight participants (two female, six male; 18 to 34 years old) who did not participate in the first-round test were recruited. Similar to the first-round test, they sat 2 m from the center of the display (Figure 25). After a 20-min introduction session, participants performed 29 window management tasks ranging from simple tasks to compound ones.

Simple Tasks

Tasks labelled ST1 24 were simple ones that required users to roughly reposition or resize windows with one or two specific operations described on the instruction sheets that we provided (Figure 26). The series of simple tasks were designed to test if users could correctly understand and use the new interaction techniques.

Compound Tasks

We extracted window management behaviors from the large display user's realistic activities reported in the longitudinal study to create five compound tasks (Figure 27). Although we used images of windows instead of real application windows, each of the compound tasks were carefully chosen to emulate a set of common daily window management activities.

- Information checking (CT1): Move an application (e.g., personal information program such as a calendar) from side to center, enlarge it to reveal the contents, and then return it back to roughly the original position and size.

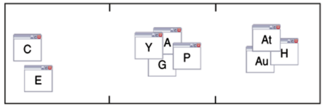
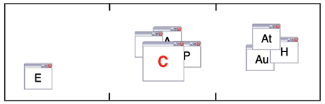
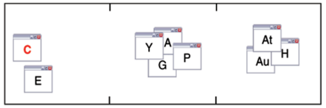
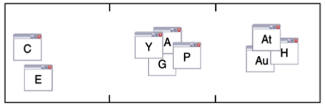
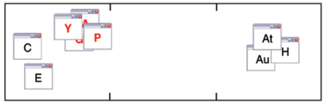
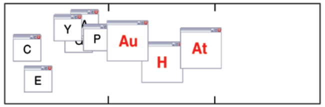
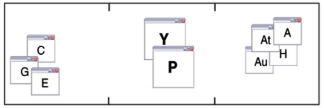
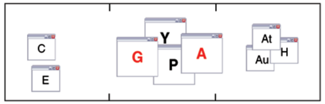
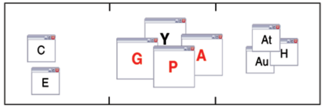
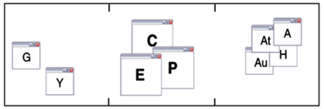
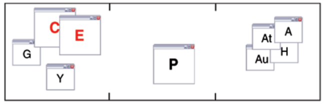
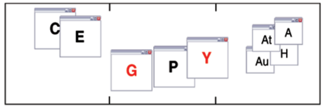
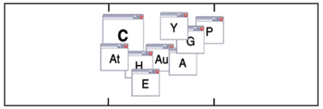
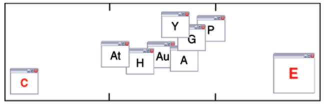
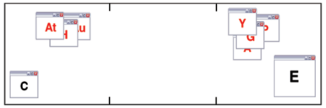
FIGURE 26. Simple tasks (summary).

ST	Description
Move single or multiple windows by dragging the Fringe.	
1	Move A/G/Y/P (Center) simultaneously to Right.
2	Move A (Center) to Right.
3	Move A/Y (Center) simultaneously to Right.
4	Move C/E (Left) simultaneously to Center.
Resize single or multiple windows by dragging a corner of the Fringe.	
5	Enlarge and then shrink A/G/Y/P (Center) simultaneously.
6	Enlarge and then shrink C/E (Left) simultaneously.
7	Enlarge and then shrink C (Left).
Change the overlapping order of windows by using the Jab-to-Lift operation.	
8	Insert C (Left) under E (Left).
9	Insert C (Left) under A/G/Y/P (Center).
10	Insert A (Center) between G and Y (Center).
Pack or unpack windows by using the Pack/Unpack icons.	
11	Pack and then unpack A/G/Y/P (Center).
12	Pack and then unpack C/E (Left).
Perform multiwindow operations with multiwindow marking menus.	
13	Spread A/Y/G/P (Center).
14	Undo ST13.
15	Pile up A/G/Y/P (Center) and review them with the mouse wheel.
16	Splat A/G/Y/P (Center) and select G as the top window.
17	Side A/G/Y/P (Center) to Left.
18	Side A/G/Y/P (Center) to Right.
19	Center C/E (Left).
20	Close A/G/Y/P (Center).
Perform single-window operations with single–window marking menus.	
21	Center C (Left).
22	Side P (Center) to Right.
23	Side P (Center) to Left.
24	Close E (Left).

Note. A = Park–Arches; G = Park–Glacier; Y = Park–Yosemite; P = PowerPoint; C = Calendar; E = E-mail; Au = CHI–Author; At = CHI–Attending; H = CHI–Home.

- Task switching (CT2): Move windows that have been used for the current task from center to side, pack them, and move other windows that are to be used for the next task from side to center, and spread them.
- Window arrangement (CT3, CT4): For focused tasks, find and move relevant windows scattered across the large display next to the main application window, then resize and reposition them to form a spread layout. These tasks were intended to mimic programming and/or active reading activities.
- Tidying (CT5): Clean the workspace by grouping relevant windows together, and move them aside.

FIGURE 27. Compound tasks (summary).

CT	Description		
1	***Information checking*** (Calendar) 	Move C (Left) to Center and enlarge it. 	Move C back to Left and make it small. 
2	***Task switching*** (from PowerPoint to CHI) 	Move A/G/Y/P to Left and tighten them. 	Move Au/At/H to Center and enlarge them. 
3	***Window arrangement*** (PowerPoint) 	Move A/G to Center next to P. 	Re–order windows (A/G under P). 
4	***Task switching and window arrangement*** 	Move E (and C) to Left. 	Move G/Y to Center next to P. 
5	***Tidying*** 	Move C/E to Side; shrink C and enlarge E. 	Move/tighten At/Au/H, A/G/Y/P, respectively. 

Note. A = Park–Arches; G = Park–Glacier; Y = Park–Yosemite; P = PowerPoint; C = Calendar; E = E-mail; Au = CHI–Author; At = CHI–Attending; H = CHI–Home.

Here is the description of a compound task written on a piece of paper handed to the participants in the study.

> e.g., CT2. Move "Park-Arches", "Park-Yosemite", "Park-Glacier", and "Power-Point-Park" windows from the center region to the left side of the display, and tighten them to save space. Move windows "CHI-Author", "CHI-Attending", and "CHI-Home" from the right side to the center region. Surround "CHI-Home" window with "CHI-Attending" and "CHI-Author".

Different from simple tasks where every operation is specified, users were allowed to perform compound tasks using any available operation. Another difference is that compound tasks were performed under two conditions:

- *With-condition*: Participants can use both new and traditional window interaction techniques (i.e., dragging the title bar to move a window, dragging its border or corner to resize a window, clicking buttons on its upper right corner to close or maximize a window).
- *Without-condition*: Participants are not allowed to use new interaction techniques; they can use only traditional window operations.

We set these two conditions to test how much new interaction techniques improve the efficiency of managing windows on a large wall display. Four of eight participants performed the compound tasks under the with-condition first and then under the without-conditions. The order was reversed for the other four participants.

For both the simple and compound tasks, participants were provided with sheets of paper on which the detailed description of each task was written. They were instructed not to start each task until they completely understood it and to start each task by clicking the Start button and end by clicking the Finish button (both buttons were located at the bottom of the screen).

Dependent Variables

Simple Tasks

We focused on two dependent variables for each simple task.

- *Completion time*: The time lapse between the user clicking the Start and Finish buttons.
- *Number of failed trials*: If windows are repositioned or resized differently from the description of each task, it is a failed trial and the participant has to redo it. An experimenter monitors the experiment and judges each trial to be successful or failed immediately after the Finish button is clicked. Simple tasks must be performed with operations specified on the script, otherwise it is a failure. Take ST19, for example. It is a failed trial if a participant moves the window "Calendar" to the center region by dragging the Fringes. The system logs every operation and signals if wrong commands are used.

Compound Tasks

Similar to simple tasks, we record both completion time and the number of failed trials for each compound task. The only difference is that the logged operations are not used to judge a trial to be successful or failed, because participants are free to use any available operations for completing compound tasks.

Results

Simple Tasks

Figure 28 shows the mean completion time and total number of failed trials for each task. The total number of failed trials sums failed trials of all the participants for each task. The mean completion time of 24 tasks is a reasonably quick 7.9 s ($\sigma = 1.3$), implying that users executed these operations in a timely manner. The total number of failed trials is zero for 18 of 24 tasks, indicating that users could learn most of the interaction techniques without any trouble. For ST2, 4, 9, 13, and 20, this dependent variable is less than or equal to 2, showing that most users correctly performed these tasks the first time. ST10 was found to be the most difficult task that asks users to insert one window between two overlapped windows. Four users encountered difficulties when performing it for the first time, but they succeeded in the second trial.

In general, all the participants were able to learn our new interaction techniques quickly and easily. The poststudy questionnaire confirms that the new techniques were easy to learn (4.5/5), that users were able to accomplish the tasks (4.4/5), and that users liked the new techniques (4.6/5).

FIGURE 28. Mean completion time and total number of failed trials for simple tasks.

ST (Simple Task)	1	2	3	4	5	6	
Completion time (s)	7.1	6.5	6.3	7.2	6.1	7.4	
Standard deviation (σ)	1.4	1.3	1.5	1.8	2.3	2.8	
No. of failed trials	0	1	0	1	0	0	
ST (Simple Task)	7	8	9	10	11	12	
Completion time (s)	6.8	7.6	8.1	11.4	7.7	10.7	
Standard deviation (σ)	3.9	3.2	1.9	4.5	2.2	2.0	
No. of failed trials	0	0	1	4	0	0	
ST (Simple Task)	13	14	15	16	17	18	
Completion time (s)	8.2	7.2	7.8	9.3	8.1	8.2	
Standard deviation (σ)	2.4	1.2	3.8	4.2	2.3	3.2	
No. of failed trials	2	0	0	0	0	0	
ST (Simple Task)	19	20	21	22	23	24	Total
Completion time (s)	9.1	9.5	7.9	6.9	7.3	7.6	7.9
Standard deviation (σ)	3.7	4.7	3.5	3.3	3.5	3.2	1.3
No. of failed trials	0	1	0	0	0	0	10

Compound Tasks

Figure 29 shows the mean time for performing CT1 to CT5. The participants completed each task much faster with new interaction techniques. The average mean completion times are 59.54 s ($\sigma = 16.4$) and 77.86 s ($\sigma = 21.4$) under the with- and without-condition, respectively. A dependent *t* test shows a significant difference in completion time for every task between the two conditions ($p < .01$ for all the tasks). Only two failed trials were observed across all participants and tasks—one for CT1 under the with-condition and the other for CT5 under the without-condition. Participants who committed the failed trials explained that they misunderstood the descriptions of the tasks.

To answer why participants performed given compound tasks much faster with new interaction techniques, we analyzed the logged operations. The average numbers of operations for each task are 8.7 ($\sigma = 3.5$) and 14.3 ($\sigma = 5.6$), with and without the new techniques, respectively. It indicates that the participants used fewer operations to complete tasks when they were allowed to use the new window operation techniques. Furthermore, the proportions of used operations under the with-condition (Figure 30) show that more than 85% of used operations were the new interaction techniques for each task despite the fact that participants were free to use both the new techniques and traditional operations. In particular, the Fringe usage accounted for nearly 50% on average. Thus, we can say that the performance gain in the tests mainly comes from using our new interaction techniques and that applying them would likely significantly improve window management efficiency on a large display because these compound tasks represent realistic window management activities.

Five participants reported that the most helpful technique was the Fringe because it helped save time and effort, enabling them to move or resize multiple windows simultaneously. Four noted that Center and Side Left/Right commands were very helpful. One explained:

> I use the menu to move windows because I am just too lazy to move the cursor a long distance.

FIGURE 29. Mean completion time (with standard deviation) for compound tasks.

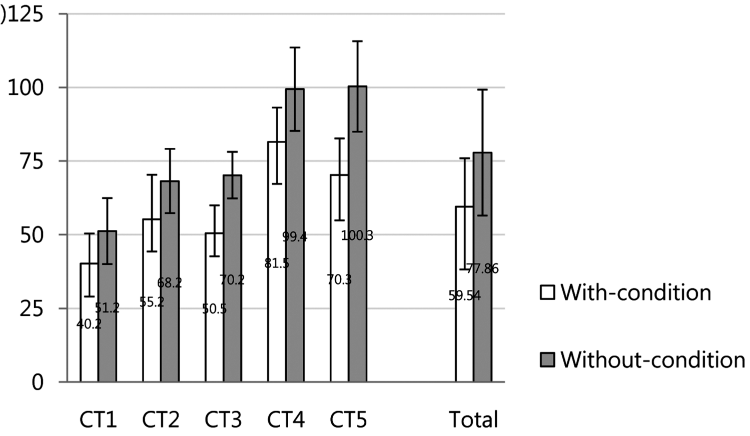
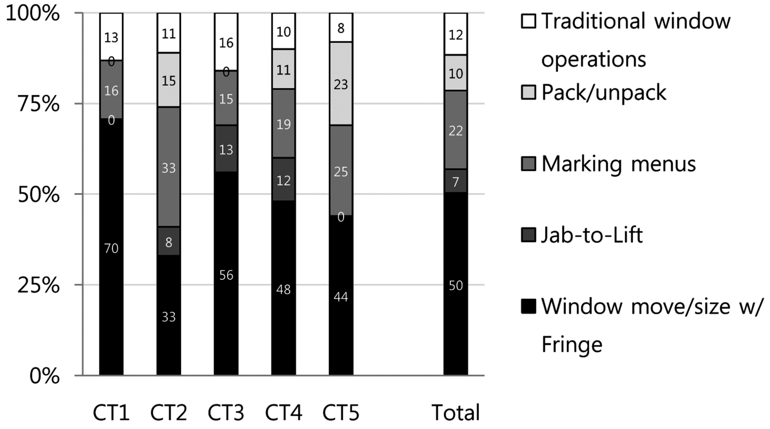


FIGURE 30. Proportions of used operations under the with new techniques condition.



In the poststudy questionnaire, all participants agreed to the usefulness of WallTop: Do the additional window operations make window management better or worse? (4.6/5) (1 = *much worse*, 5 = *much better*).

4.5. Discussion

Reflection of Design Objectives

Our new interaction techniques were designed to meet the challenges of managing overflowing windows on a large display discussed in Section 3.3. Informed by the usability results, we now reflect on how the proposed interaction techniques meet our design objectives.

Multiwindow Operations

The Fringe helps users select multiple windows easily, and the multiwindow marking menu offers a variety of operations including quickly centering/siding windows and arranging them in different layouts. The results of the simple task test show that users can quickly and easily learn these operations, and those of the compound task test show that using these techniques can significantly shorten the completion time of realistic window management activities.

Flexible Window Arrangement

The Jab-to-Lift technique adds flexibility in arranging windows. It allows users to insert a window(s) under others, complementing traditional window dragging and dropping only on top of others. The usability tests showed that users could quickly learn this operation after a short period of practice. One participant reported that the Jab-to-Lift seemed a bit slower than traditional ways, but she still liked it because it was more natural and flexible, with a dose of fun.

The multiwindow operations via marking menus also allow users to quickly spread, pile up, pack, and unpack multiple windows. Together with the Fringe, they offer greater flexibility of managing windows than traditional window management operations.

Facilitation of Moving and Resizing Windows

Because the Fringe is wider than traditional window borders or title bars, and it appears on four sides of a window instead of only on top, moving and resizing windows become much easier from a basic Fitts's law perspective. The second usability test, given compound tasks involving many moving and resizing operations, confirmed this with the Fringe usage accounting for 50% of the number of all the operations done and that the participants performed the tasks much faster with the Fringe than without it. Furthermore, the Center, Side Left/Right operations on the marking menus meet the demands of quickly switching windows between center and side regions.

In-Place Command Execution

We achieved this goal by using marking menus. Users can invoke the menus anywhere on the Fringe instead of a fixed corner. In addition, the marking menus utilize cursor moving direction instead of precise clicking for command execution, avoiding the error-prone clicking operations on a large display. One exception is the Pack and Unpack icons, which are located on the upper-left corner of the Fringe. Because they require subsequent continuous mouse movement, it is hard to fit them into a marking menu. But to ease clicking them, these icons are bigger than traditional Minimize and Close buttons. One participant suggested placing Pack/Unpack icons on every convex corner of the Fringe. However, we did not incorporate this suggestion as we felt it would result in visual clutter.

Compatibility With Traditional Window Operations

In addition to meeting the challenges of managing windows, our new techniques were also designed to ensure high compatibility with traditional window operations. In the WallTop system, none of the new interaction techniques conflicts with the existing window operations. All the traditional operations are also maintained in our prototype and can be used in conjunction with new ones. In addition, we note that some of the new interaction techniques are judiciously designed to leverage the user's familiarity with existing interaction conventions, such as dragging the Fringe to move windows, and dragging its corner to resize them.

Comparison With Other Window Management Systems

WallTop includes a set of new interaction techniques addressing challenges neglected in previous research. Figure 31 summarizes the novelty of WallTop in comparison with other window management systems.

FIGURE 31. New functions in WallTop compared with other systems.

		Scalable Fabric	Metisse
	WallTop	(Robertson et al., 2004)	(Chapuis & Roussel, 2005)
Multiwindow operations	Multiwindow resizing and moving via the Fringe Multiwindow rearranging, quick moving, packing and unpacking via the Fringe+marking menu	Multiwindow grouping/ungrouping	Not enabled
Flexible window arrangement	Adjusting the order of overlapping windows via the Job-to-Lift operation Several window layouts via multiwindow marking menu	Traditional operations[a]	Rotating and peeling back windows
Facilitation of moving and resizing windows	Easy window moving and resizing via the Fringe Center, Side Left/Right commands on marking menus to quickly switch windows between sides and center	Minimize/Maximize buttons to quickly move windows between sides and center	Traditional operations[a]
In-place command execution	Marking menus on the Fringe	Not enabled	Not enabled

[a] Traditional operations on Window XP or Mac OS.

5. CONCLUSIONS

In this article, we first presented a longitudinal diary study comparing usage of a large display to those of single and dual monitors in a desktop work environment. The results indicated users' unanimous preferences for working on a large display and revealed large-display users' distinct behavior patterns in managing windows. In particular, the following major findings were discovered through the study:

- Users unanimously prefer working on a large display over a single or dual monitor(s). A large display could benefit scattered information-processing tasks and multiscale navigation tasks, enhance users' awareness of peripheral applications, and offer immersive working experiences.
- Large-display users exhibit distinct behaviour patterns in managing windows. Users tend to utilize the center part as the focal region where primary windows are located. The remaining space is used as the peripheral region, where peripheral applications reside. Compared to normal single- or dual-monitor users, large-display users invest more effort to spatially arrange windows in various layouts and have special interaction and visualization demands of windows in the peripheral region. Users tend to perform more mouse operations and window management operations on a large display. The increasing mouse operations are caused by the increased number of window moving and resizing operations.
- Traditional window management systems are designed for normal-sized monitors, which are not well suited to large displays. We identified four challenging functions that large-display users will benefit from: multiwindow operations, flexible window arrangement, facilitation of moving and resizing windows, and in-place command execution.

Based on the study results, we then designed a set of new interaction techniques to meet the challenges of managing numerous windows on a large display, including the Fringe, Jab-to-Lift, and various window arranging functions such as Spread, Pile, Center, and Side. All these techniques were coherently integrated with traditional operations in our proof-of-concept window management prototype, WallTop. The usability tests showed that these techniques are easy to learn and can significantly improve the efficiency of managing windows on a large display.

Aside from studying users' behaviors and designing appropriate interaction techniques, we also made contributions for quantitative study design and creating benchmarks for evaluating window management systems.

- To understand how users perform desktop work on a large display, we adopted an approach of enhancing self-reported data (activity logs and interviews) with objectively recorded data (mouse and window event logs). These methods are complementary; self-reported data provide detailed qualitative data describing a participant's real-life activities in workplace, whereas objectively recorded data provide cross-validation for the findings obtained from qualitative data.

This approach could be adopted to analyze user behaviors in other computing scenarios with minor changes.

- In the second-round usability test (Section 4.4), we designed a set of window management tasks ranging from basic operations on a single window to complex arrangement tasks involving multiple windows. These tasks are extracted from the longitudinal diary study to reflect the commonly performed window management tasks for desktop work. They could also serve as benchmarks to evaluate the efficiencies of other window management systems.

We left some possible window operations not fully explored, such as the taskbar functions. Smith et al. (2003) showed that the taskbar needs to be augmented to suit a large display, such as providing thumbnail views of windows and group functions. Designing a large-display-oriented taskbar will be an interesting topic for our future work. We note that the interaction techniques suggested in this article are independent of taskbar functions. They can be easily incorporated with any taskbar with minor changes.

NOTES

Background. This article is based on the Ph.D. thesis of Xiaojun Bi at the University of Toronto Computer Science Department.

Acknowledgments. We thank the support of many DGP colleagues. We are particularly grateful to the action editor and the three exceedingly insightful reviewers whose comments and suggestions have improved the article.

HCI Editorial Record. First manuscript received October 18, 2010. Revisions received February 24, 2012 and February 11, 2013. Accepted by Rob Jacob. Final manuscript received June 2, 2013. — *Editor*

REFERENCES

Agarawala, A. & Balakrishnan, R. (2006). Keepin' it real: pushing the desktop metaphor with physics, piles and the pen. *Proceedings of the CHI 2006 Conference on Human Factors in Computer Systems*. New York: ACM.

Ahlström, D., Großmann, J., Tak, S. & Hitz, M. (2009). Exploring new window manipulation techniques. *OZCHI*, 177–183.

Andrews, C., Endert, A., & North, C. (2010). Space to think: large high-resolution displays for sensemaking. *Proceedings of the CHI 2010 Conference on Human Factors in Computer Systems.* New York, ACM.

Ball, R., & North, C. (2005). An analysis of user behavior on high-resolution tiled displays. *IFIP International Conference on Human–Computer Interaction*, 350–364.

Baudisch P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P., Bederson, B., & Zierlinger, A. (2003). A drag-and-pop and drag-and-pick: techniques for accessing remote screen content on touch- and pen-operated systems. *Interact*, 57–64.

Baudisch, P., Good, N., & Stewart, P. (2001). Focus plus context screens: Combining display technology with visualization techniques. *Proceedings of the UIST 2003 Symposium on User Interface Software and Technology*. New York: ACM.

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

Bederson, B., & Hollan, J. (1994). Pad++: A zooming graphical interface for exploring alternate interface physics. *Proceedings of the UIST 1994 Symposium on User Interface Software and Technology*. New York: ACM.

Bernstein, M., Shrager, J., & Winograd, T. (2008). Taskposé: Exploring fluid boundaries in an associative window visualization. *Proceedings of the UIST 2008 Symposium on User Interface Software and Technology*. New York: ACM.

Bi, X., Bae, S., & Balakrishnan. R. (2010). Effects of interior bezels of tiled-monitor large displays on visual search, tunnel steering, and target selection. *Proceedings of the CHI 2010 Conference on Human Factors in Computer Systems*. New York: ACM.

Bi, X., & Balakrishnan, R. (2009). Comparing usage of a large high-resolution display to single or dual desktop displays for daily work. *Proceedings of the CHI 2009 Conference on Human Factors in Computer Systems*. New York: ACM.

Bishop, G., & Welch, G. (2000). Working in the office of real soon now". *IEEE Computer Graphics and Applications, 20*(4), 76–78.

Brujin, D. D., Mul, S. D., & Oostendorp, H. V. (1992). The influence of screen size and text layout on the study of text. *Behaviour & Information Technology, 11*, 71–78.

Chapuis, O., & Roussel, N. (2005). Metisse is not a 3D desktop! *Proceedings of the UIST 2005 Symposium on User Interface Software and Technology*. New York: ACM.

Chapuis, O., & Roussel, N. (2007). Copy-and-paste between overlapping windows. *Proceedings of the CHI 2007 Conference on Human Factors in Computer Systems*. New York: ACM.

Collomb, M., Hasco, M., Baudisch, P., & Lee, B. (2005). Improving drag-and-drop on wall-size displays. *GI*, 25–32.

Czerwinski, M., Smith, G., Regan, T., Meyers, B., Robertson, G., & Starkweather, G. (2003). Toward characterizing the productivity benefits of very large displays. *Interact*, 9–16.

Dragicevic, P. (2004). Combining crossing-based and paper-based interaction paradigms for dragging and dropping between overlapping windows. *Proceedings of the UIST 2004 Symposium on User Interface Software and Technology*. New York: ACM.

Faure, G., Chapuis, O., & Roussel, N. (2009). Power tools for copying and moving: useful stuff for your desktop. *Proceedings of the CHI 2009 Conference on Human Factors in Computer Systems*. New York: ACM.

Fridgeman, B., Lennon, M. L., & Jackenthal, A. (2003). Effect of screen size, screen resolution, and display rate on computer-based test performance. *Applied Measurement in Education, 16*, 191–205.

Grudin, J. (2001). Partitioning digital worlds: Focal and peripheral awareness in multiple monitor use. *Proceedings of the CHI 2001 Conference on Human Factors in Computer Systems*. New York: ACM.

Henderson, D. A., & Card, S. (1986). Rooms: The use of multiple virtual workspaces to reduce space contention in a window-based graphical user interface. *ACM Transactions on Graphics, 5*, 211–243.

Hutchings, D., Smith, G., Meyers, B., Czerwinski, M., & Robertson, G. (2004). Display space usage and window management operation comparisons between single monitor and multiple monitor users. *Advanced Visual Interfaces*, 32–39.

Hutchings, D., & Stasko, J. (2005). mudibo: Multiple dialog boxes for multiple monitors. *CHI Extended Abstracts*, 1471–1474.

Hutchings, D., & Stasko, J. (2007a). Consistency, multiple monitors, and multiple windows. *Proceedings of the CHI 2007 Conference on Human Factors in Computer Systems*. New York: ACM.

Hutchings, D. & Stasko, J. (2007b). Quantifying the performance effect of window snipping in multiple-monitor environments. *Interact*, 461–474.

Kandogan, E., & Shneiderman, B. (1997). Elastic Windows: Evaluation of multi-window operations. *Proceedings of the CHI 1997 Conference on Human Factors in Computer Systems*. New York: ACM.

Kurtenbach, G. (1993). *The design and evaluation of marking menus* (Unpublished doctoral dissertation). University of Toronto, Toronto, Canada.

MacIntyre, B., Mynatt, E., Voida, S., Hansen, K., Tullio, J., & Corso, G. (2001). Support for multitasking and background awareness using interactive peripheral displays. *Proceedings of the UIST 2001 Symposium on User Interface Software and Technology*. New York: ACM.

Mander, R., Salomon, G., & Wong, Y. Y. (1992). A "pile" metaphor for supporting casual organization of information. *Proceedings of the CHI 1992 Conference on Human Factors in Computer Systems*. New York: ACM.

Pfeiffer Consulting. (2005). *The 30-inch Apple Cinema HD display productivity benchmark* (Pfeiffer Report). Naples. FL: Author.

Ramos, G., Robertson, G., Czerwinski, M., Tan, D., Baudisch, P., Hinckley, K., & Agrawala, M. (2006). Tumble! Splat! Helping users access and manipulate occluded content in 2D drawings. *AVI*, 428–435.

Robertson, G., Czerwinski, M., Baudisch, P., Meyers, B., Robbins, D., Smith, G., & Tan, D. (2005). The large-display user experience. *IEEE Computer Graphics & Applications, 25*, 44–51.

Robertson, G., Czerwinski, M., Larson, K., Robbins, D. C., Thiel, D., & van Dantzich, M. (1998). Data mountain: Using spatial memory for document management. *Proceedings of the UIST 1998 Symposium on User Interface Software and Technology*. New York: ACM.

Robertson, G., Horvitz, E., Czerwinski, M., Baudisch, P., Hutchings, D., Meyers, B., . . . Smith, G. (2004). Scalable fabric: Flexible task management. *AVI*, 85–89.

Robertson, G., van Dantzich, M., Robbins, D., Czerwinski, M., Hinckley, K., Risden, K., . . . Gorokhovsky, V. (2000). The Task Gallery: A 3D window manager. *Proceedings of the CHI 2000 Conference on Human Factors in Computer Systems*. New York: ACM.

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

Simmons, T. (2001). What's the optimum computer display size? *Ergonomics in Design, 9*(4), 19–25.

Smith, G., Baudisch, P., Robertson, G., Czerwinski, M., Meyers, B., Robbins, D., & Andrews, D. (2003). GroupBar: The TaskBar evolved. *OZCHI*, 34–43.

Sommerich, C. M., Joines, S. M. B., & Psihogios, J. P. (1998). Effect of VDT viewing angle on user biomechanics, comfort, and preference. *Human Factors and Ergonomics Society 42nd Annual Meeting*, 861–865.

Stuerzlinger, W., Chapuis, O., Phillips, D., & Roussel, R. (2006). User interface façades: towards fully adaptable user interfaces. *Proceedings of the UIST 2006 Symposium on User Interface Software and Technology*. New York: ACM.

Tak, S., & Cockburn, A. (2009). Window Watcher: A visualisation tool for understanding windowing activities. *OZCHI*, 241–248.

Tan, D. S., Meyers, B., & Czerwinski, M. (2004). WinCuts: Manipulating arbitrary window regions for more effective use of screen space. *Proceedings of the CHI 2004 Conference on Human Factors in Computer Systems*. New York: ACM.

Waldner, M., Steinberger, M., Grasset, R., & Schmalstieg, D. (2011). Importance-driven compositing window management. *Proceedings of the CHI 2011 Conference on Human Factors in Computer Systems*. New York: ACM.

Watanabe, N., Washida, M., & Igarashi, T. (2007). Bubble clusters: An interface for manipulating spatial aggregation of graphical objects. *Proceedings of the UIST 2007 Symposium on User Interface Software and Technology*. New York: ACM.

Xu, Q., & Casiez, G. (2010). Push-and-pull switching: Window switching based on window overlapping. *Proceedings of the CHI 2010 Conference on Human Factors in Computer Systems*. New York: ACM.