

Evaluating Visual Cues for Window Switching on Large Screens

First Author Name (Blank if Blind Review)

Affiliation (Blank if Blind Review)

Address (Blank if Blind Review)

Second Author Name (Blank if Blind Review)

Affiliation (Blank if Blind Review)

Address (Blank if Blind Review)

ABSTRACT

An increasing number of users are adopting large, multi-monitor displays. The resulting setups cover such a broad viewing angle that users can no longer simultaneously perceive all parts of the screen. Changes outside the user's visual field often go unnoticed. As a result, users often have trouble locating the active window, especially after switching focus using a keyboard. This paper surveys graphical cues designed to direct visual attention and adapts them to window switching. Visual cues include five types of frames and mask around the target window and four trails leading to the target window. We report the results of two user studies. The first evaluates each cue in isolation. The second study evaluates hybrid techniques created by combining the most successful candidates from the first study. The best cues were visually sparse – combinations of curved frames which use color to pop-out and tapered trails with predictable origin.

Author Keywords

Large displays, attention, window switching, user study.

ACM Classification Keywords

H5.2. [User Interfaces]: Graphical User Interfaces (GUI), Windowing Systems.

INTRODUCTION

Users of desktop computers are increasingly turning to larger displays and multiple-monitor setups; indeed, usability studies show significant productivity benefits and higher satisfaction [21, 9, 22]. A key advantage of additional screen space is the ability to keep more windows open simultaneously, reducing the amount of resizing, repositioning, and other window-management activity [18, 12].

Unfortunately, however, additional windows also mean that more windows are competing for the users' attention. This can make it harder for users to locate desired contents. In this paper, we specifically look at the task of locating a win-

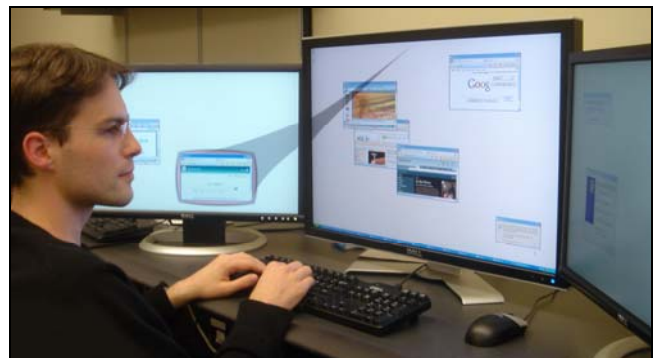


Figure 1: As this user is switching windows by hitting `<alt><tab>` a colored frame and a trail from the center of the screen are directing his attention to the window receiving focus. This particular visualization style performed among the best during our user studies.

row receiving focus (Figure 1). On large displays, window switching takes place on average once every 20.9 seconds [12], thus several hundred times per day. When using a keyboard shortcut to switch (e.g. `<alt><tab>` in Microsoft Windows), users need to locate the window receiving focus.

Two visual cues help draw the user's attention to that window. First, in case the window was partially occluded, that area is now repainted. Second, window managers typically highlight the active window by altering the color of its title bar.

On large screens and multi-monitor systems, however, these cues typically fail, because they are localized to the region of the newly active window, which may be outside the user's field of view [6]. Once a user has missed activation, the new window provides no further signals likely to catch the user's eye. Only the second cue, the difference in color, remains, which is a comparably weak cue [2, 11]. Given this lack of a pop-out effect, finding the active window now requires a time-consuming visual scan of the display. In the two user studies reported in this paper, this scan often took several seconds.

A variety of visual effects have been proposed for directing a user's attention on the screen. *Spotlight* [13] enables speakers to guide an audience to a display region by darkening the entire area around this region. *Phosphor* [3] shows diagrammatic depictions of interface transitions. The Apple *Leopard* operating system bounces popup windows as they appear, and Windows *Vista* highlights the active window using a shadow effect.

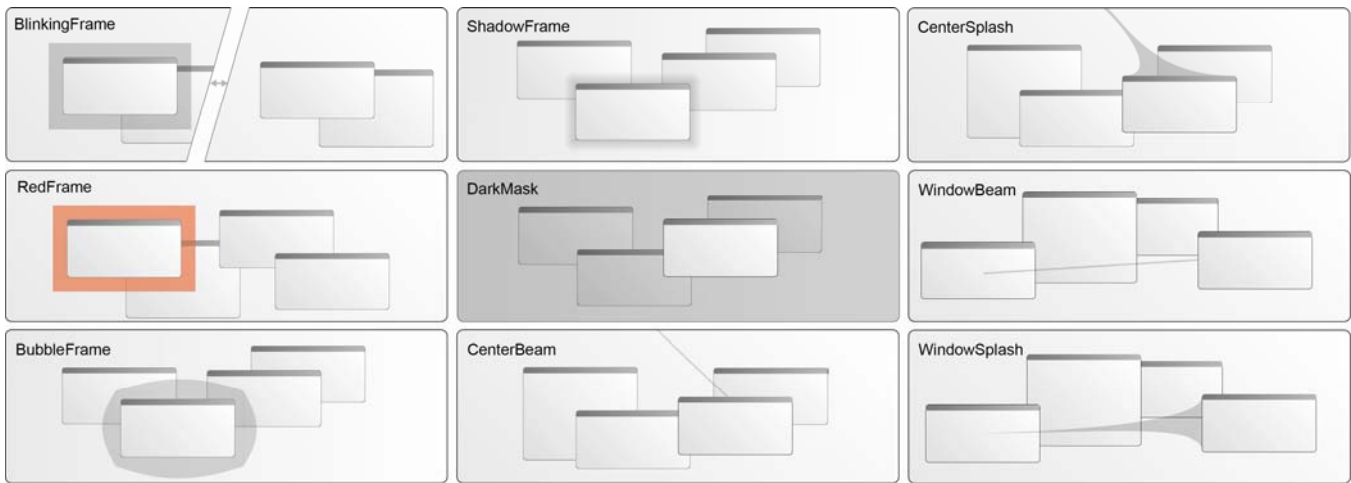


Figure 2: Four cues highlight the target with a frame, one cue masks background, and four show graphic trails.

In this paper, we ask whether these techniques can be used to speed up window switching and if so, which is most effective. In two user studies we investigate nine methods and combinations for guiding users through focus switches (Figure 2). We find that hybrids of trails and frames are highly effective, and conclude with implications for design.

RELATED WORK

Although we are not aware of work evaluating visual effects for switching windows, there is a large literature on directing attention and visualizing transitions in interfaces. Much of this work is based on perceptual foundations.

Perceptual Underpinnings

Humans see only 2 degrees with high acuity, but low-acuity peripheral vision may extend to 100-110 degrees from the point of fixation. Furthermore, the visual field is not constant but shrinks with mental workload and with age [16]. In comparison, a typical three-monitor configuration, such as that used in our study, covers a visual angle of 143 degrees.

Feature integration theory [23] assumes that search proceeds in two stages: an initial stage of *pre-attentive* search, where low-level features (such as color) are efficiently extracted in parallel across the full field of view, and a stage of *attentive* search, where interesting points identified in the first stage are visited serially. If an item visually pops-out because a feature pre-attentively guides attention, the time required to find it is independent of the number of distracters. Numerous studies have thus tried to identify features that can guide attention. The most certain features are color, motion, orientation, and size [24]. Other likely features are shape and shading. Shading can enable the perception of depth which pops out [14]. Not all guiding features have the same visual span, however. For example, motion can be detected at greater distances from the point of fixation than color [2] or shape. In general, targets far from the point of fixation are detected slower and less accurately [27]. Furthermore it has been found that guiding features are not truly independent: Random variations of color interfere with the ability to find a target based on orientation [7] or

shape [15]. The converse may also be true but to a lesser degree, suggesting that there exists a hierarchy with some features, such as color, dominating others, such as orientation or shape [7].

Complex backgrounds may also make it harder to separate items and thus also reduce search efficiency [26]. In the presence of distracters, search becomes more efficient when target and distracters are less similar. Heterogeneity among distracters reduces search efficiency, but if distracters can be grouped by some criteria efficiency is improved [10].

In contrast to common belief, motion does not necessarily draw attention, although the appearance of a new perceptual item does [11]. Yet, a short interruption in what is being seen, such as a blink, can prevent a person from noticing visual changes. Larger targets and fewer distracters reduce the likelihood of change blindness [1].

Pop-out effects

Researchers have created a variety interface techniques which seek to exploit the human per-attentive capabilities.

Emphasizing the target

The *Popout Prism* [20] enlarges occurrences of search terms in a document and places them on a semi-transparent bright background, to speed location. In addition to color and size, the appearance of the cue itself draws attention. *Halo* [3] is a visualization technique for objects that are off-screen, such as relevant places on a map. Halo makes use of shape and color; its red arcs easily pop-out from the rectangular structure of typical road maps. Bartram et al. [2] used motion in a notification technique, finding it 6-25% more effective than color or shape in drawing attention. Motion, however, may also be found distracting, and Bartram showed that moving animations were more distracting than anchored animations; the least distraction came from slow blinking. This is consistent with Plaue et al.'s finding that slow fade-in/fade-out transitions are less distracting [10].

Deemphasizing background content

Spotlight [13] is a technique for directing users' attention by darkening the majority of a display, leaving a focal area brightly visible. A variation of the technique, which com-

bines the spotlight with a beam descending from the center top of the display, performs equally well. Zhai et.al. [28] compared four masking effects, investigating the trade-off between effectiveness of directing attention and readability of background material. For their tasks on a desktop-sized screen, they found there to exist a range of mask intensities such that highlighted objects visually pop-out without hindering the performance reading background. Nevertheless, subjects consistently reported difficulty reading background material when masking effects were displayed. Apple's *Safari* browser uses a mask to direct attention to keywords in a page-level searches.

In our work, we evaluated the use of color, shape, and shading to draw attention to window activations. We also created a mask that darkens the screen around the target. Given the high frequency of window switches per day, we felt that distraction and fatigue would likely be an issue with motion-based techniques. We therefore follow recommendations in the literature [2, 10] and only consider local fade-in and fade-out animations.

Graphical Depictions

To avoid the drawbacks of motion-based methods, several researchers have proposed innovative static depictions. *Drag-and-pop* [3] is an interaction technique for accessing objects which would otherwise be hard to reach. Users can interact with translated copies of icons and “rubber bands” show where these originate. *Phosphor* [3] alerts users to potential mistakes (e.g., accidentally deleting a desktop icon) by showing diagrammatic depictions of interface transitions, after they have occurred. Bezerianos et. al. describe a related paradigm, mnemonic rendering [6].

Inspired by these ideas, we created four graphical trails to guide users to the target window (Figure 2).

NINE VISUAL CUES FOR WINDOW SWITCHING

Our main design goal is to successfully direct the user's attention to the window receiving focus. A successful technique, however, also needs to minimize side effects. More concretely, the technique should (1) not distract users who already know where the target window is located, (2) minimize its impact on other screen content (even if a window is inactive, users might want to scan its contents). A cue should therefore affect only a small area of screen space. And (3) the technique should use only sparse visual effects to minimize fatigue.

Based on these requirements, we created nine basic cues, each representing a unique feature. We use a factorial design, meaning that these cues can be combined in numerous ways to create more complex signals, including those found in the literature. In our study we evaluate the basic cues as well as selected combinations.

The basic cues fall into two groups: those that attempt to pop attention to the location of the target window, and graphical trails which lead to the target window. The former have the potential benefit of harnessing the pre-attentive visual system, making the target stand out from remaining

content; the latter have the potential to work even when the target is not within the user's field of view and in the presence of substantial visual noise, in which case they may reduce search in two dimensions to a one dimensional scan.

Highlighting the target window

All highlighting conditions attempt to enable quick visual search by leveraging guiding features in vision. Since we do not want to affect the user's ability to work with the target window, the window area itself is taboo. Instead we draw visual effects around the target.

Our goal is to obtain a pop-out effect. In order to achieve this, the stimulus has to be different from window contents, including standardized user interface toolkits, but also web page contents, such as flash animations designed to draw the user's attention. A priori it is not clear which guiding features would stand out from such content. We therefore consider several features that have been identified in the perceptual literature, including motion, color, shape, shading and luminance. Rather than combining all features into a single cue, we investigate each feature individually, so that we can systematically search for better cues by combining promising candidates. In total, we created four frames, each utilizing the same screen area, and a mask:

BlinkingFrame is a black rectangular 20-pixel-wide frame. Its opacity fades back-and-forth between 50% and 0% (fully transparent) every second. This cue is motivated by work suggesting that fade-in and fade-out animations are less distracting than others [2, 10].

RedFrame is a red rectangular 20-pixel-wide, 25%-opacity frame. We chose pure because it is bright and distinct from the blue, black, and gray frequently used in current window managers.

BubbleFrame is a black 25%-opacity “inflated” rectangle, 10 pixels wide around the window's corners and 28 pixels along edge centers. We chose a rounded shape, because it separates nicely from recto-linear screen contents [24].

ShadowFrame is similar to a cue used Windows Vista—a black rectangular 20-pixels-wide frame with a transparency gradient ranging from 50% along the window edge to full transparency at its outside edge. This shading resulted in rounded corners. We expected the cue's gradient-nature to minimize impact on the readability of background content.

DarkMask is motivated by Spotlight [13] – a 75%-opacity black mask covering the entire screen except for the target window.

Trails leading to the target window

Before users can follow a trail, they must first visually acquire a part of it and then determine the correct direction to follow. To investigate these aspects, we used a factorial design with two dimensions: *trail origin* and *trail shape*. All four conditions create a visual path, 25% opacity black, from a known screen location to the newly active window. See Figure 2.

CenterBeam draws a 2-pixel-wide line from the center top of the center screen to an edge of the target window, such

that an extension of the line would intersect the center of the window. The use of a fixed location facilitates location of the trail's origin, although eye travel may be required. The location at the top center almost always leads to an angled line, which contrasts with recto-linear screen contents. Pilot testing had revealed that a 2-pixel line suffices for recognition, while minimizing interference.

CenterSplash, which is motivated by *Beamlight* [13], also uses a trail from the top center of the screen to the target, but the trail has a concave, tapered shape that grows in width from one pixel at the display's center top of the display to the full width of the target window. The asymmetry provides a cue for directionality, the taper's concavity serves as a cue for distance, ala drag-and-pop [3], and a curved outline distinguishes the trail from the background.

WindowBeam uses the same line as CenterBeam; however, the line starts at the center of the previously active window to approximate the user's current focus of visual attention and minimize eye travel. To convey directionality, the beam is rendered on top of the previous window and "below" the target window. If the new target is located on top of the previous window, the beam is drawn as a curve leaving the window and looping back in.

WindowSplash combines the tapered trail of CenterSplash with the previous-window origin of WindowBeam. Directionality is again conveyed by the tapered path.

Combining cues

Combining cue effects defines a space of 160 possible styles: 2 (blinking) x 2 (red) x 2 (bubble) x 2 (shadow) x 2 (mask) x 5 trails (2 shape x 2 origins + noTrail). Figure 1 shows one of these styles, a combination of RedFrame, BubbleFrame, ShadowFrame and CenterSplash.

Trail-based cues require users to trace the trail, while target-window highlighting avoids that additional step. In that sense, a trail may be seen as a fallback in case users miss the highlighting. In terms of design goals, this suggests that trails should not interfere with target highlights. During a pilot study, however, we noticed precisely this interference with some designs. The individual cues just presented take this into account – the trail lines are sufficiently distinct from the surface-oriented target cues. In addition, the thinness and colors minimize interference with the background.

IMPLEMENTATION

We implemented all nine cues in a program which activates whenever the user hits <alt><tab>. Any of the 160 combinations may be selected through a configuration menu, which is invoked by clicking on a systray icon. The program is written in C# using .NET 3.5 and the Windows Presentation Foundation API. It uses C++ and the Win32 API for communication with the operating system. Each effect is rendered onto a DrawingVisual pane using basic shapes, such as Bezier curves or polygons, and gradient or solid fills. The DrawingVisual is then placed into a transparent window which spans the entire desktop and is positioned on top of all other windows.

In our original implementation we detected the activation of new windows by installing a global hook on the HShell_WINDOWACTIVATED message. Unfortunately, rendering of certain cues and communication with the operating system introduced a noticeable lag between window activation and cue appearance. Thus, we programmed our system to automatically take screenshots of all activated desktop windows prior to the study, and to display the frame of an activated window together with the cue. This ensured that the cue appeared simultaneously with the window's title-bar color change.

USER STUDIES

To evaluate the relative effectiveness of the visual cues we performed two user studies. Our goals were to measure the effectiveness of each cue at directing the user's attention to the target window and also to measure side effects. As previously mentioned a successful technique should not distract users who already know where the target window is, should minimize impact on other screen content, and should minimize fatigue. The first requirement is captured by analyzing response time when switching windows. To estimate distraction, we occasionally introduced changes to screen content in non-active windows at the time when cues were displayed. Since objective measures of fatigue are costly to obtain, vary widely and are highly sensitive to many factors such as task period [8], we instead rely on subjective user feedback.

Since it is infeasible to evaluate the full factorial design with 160 combinations, we restricted our first study to the nine basic cues. Based on these findings, we then selected promising combinations to evaluate in the second study.

STUDY 1 – NINE BASIC CUES

In our first study, we analyzed the effectiveness of nine basic cues plus control in directing the users' attention to a target window. We also measured the impact of the individual cues on the participants' ability to perceive competing events on the screen, as well as subjective preferences.

Interfaces

There were 10 interface conditions: Control plus the nine cues BlinkingFrame, RedFrame, BubbleFrame, ShadowFrame, DarkMask, CenterBeam, CenterSplash, WindowBeam, and WindowSplash. All conditions including control provided the regular Windows XP target highlighting by changing the window color from light blue <RGB #7b99e1> to dark blue <RGB #0055e5>, as well as the "flashing" effect in cases where an occluded window was brought to the front.

Task

During each trial, the participants' task was to switch window focus by hitting the space bar, and then to locate the newly active window. To verify that participants had located the right window, they were required to enter the number of the new target window. The title bar of each window read "Window - #", where # was a number from 0-

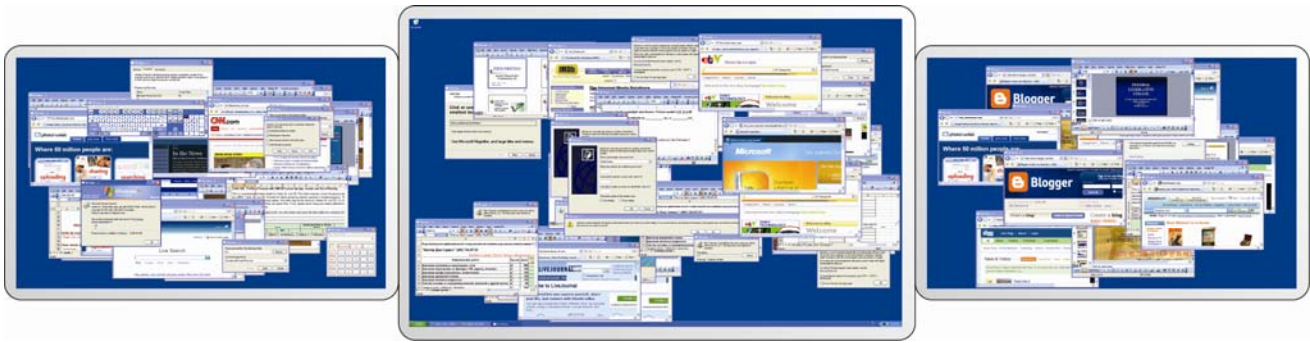


Figure 3: Eighty randomly positioned windows showing content of Windows XP applications, Office applications, and popular web pages.

9. Participants read the number and entered it by pressing the corresponding number key on the keyboard.

If an incorrect number was hit, the system provided no feedback. If the correct number was hit, the number was removed from the window's title bar as were all displayed cues. When the user next pressed the space bar, a new (potentially different) number was placed back into that window's title bar and the system activated a new window.

In 20% of the cases when participants switched windows, a popup window with a capital letter appeared for 350 ms at another location on the screen. In addition to locating the newly active window, participants were asked to enter the letter on the keyboard if they saw the popup and recognized its letter.

Procedure

The study design was within subjects with 10 conditions (interface) and 62 repetitions for each condition. For each trial, we recorded task completion time and error in finding the target window when no popup window was displayed, and error in recognizing the letter on the popup window when a popup window was displayed.

We pre-computed 10 different experimental configurations, each with a different desktop layout, sequence of window activations, choice of numbers in window title bars, as well as pop-up windows and their letters. Participants performed the study with each configuration and in the same order. However, each such configuration was combined with an interface condition and the order of interface conditions was balanced following a Latin square of order 10.

Prior to the study, users filled out a background questionnaire and trained on all interface conditions, each time performing at least 3 window activations and seeing 1 popup window. Trials were grouped into blocks of 62 repetitions and each block was performed using a new interface condition. Grouping was necessary, because frequent switching between the visual styles might prevent users from getting used to any of them and developing expertise in interpreting feedback. After each block, they were asked to provide subjective feedback on the condition just used. After the study, participants filled in another questionnaire about their subjective preference with regards to all conditions.

Desktop content and window activation sequence

In order to create a more realistic desktop environment, we sampled representative window content as follows. We created screenshots of 30 windows from the Windows XP accessories and control panel, 30 windows with browsers showing the 30 most-popular websites in the United States¹, and 30 windows with the highest ranked Word, PowerPoint, and Excel documents on Google². From this set of 90 windows we uniformly sampled 80 windows for each configuration of desktop content (see e.g. Figure 3). The appearance of the desktop and windows (colors and schemes) was set to Windows XP defaults. While some of the original websites contained flashing animations, these animations were reproduced in our study system.

Windows were positioned uniformly at random across the whole display, but only positions where a window was entirely within monitor boundaries were considered. The sequence of window activations was random, but not uniform. Since we separately consider different distances between subsequently-activated windows in our analysis, we wanted a roughly uniform distribution on these distances. We obtained this by sampling the next window with a weight exponential in the distance to the previous window. Since we believed that the difficulty of finding a target window also depends on the amount of overlap between windows (bringing a window to the front can create a flashing effect), we ensured that a balanced number of trials each did and did not involve a flashing effect due to a change in z order. Popup windows were placed uniformly at locations in the screen area spanned by a vertical line through the screen center and the outermost vertical line that tangents the target window. Numbers and letters were drawn uniformly.

Apparatus

The experiment was run on a PC with an extended desktop setup that spanned three monitors: at the center a 30" LCD monitor with a resolution of 2560 x 1600 pixels, and on each side a 24" LCD monitor with 1920 x 1200 pixels (Figure 5). Two monitors were driven by an nVidia Quadro FX 4600 card, and one by an nVidia Quadro NVS 285, all at a refresh rate of 60Hz. The operating system used was

¹ According to Alexa.com traffic rankings.

² Obtained using queries 'filetype:doc', 'filetype:ppt', 'filetype:xls'.

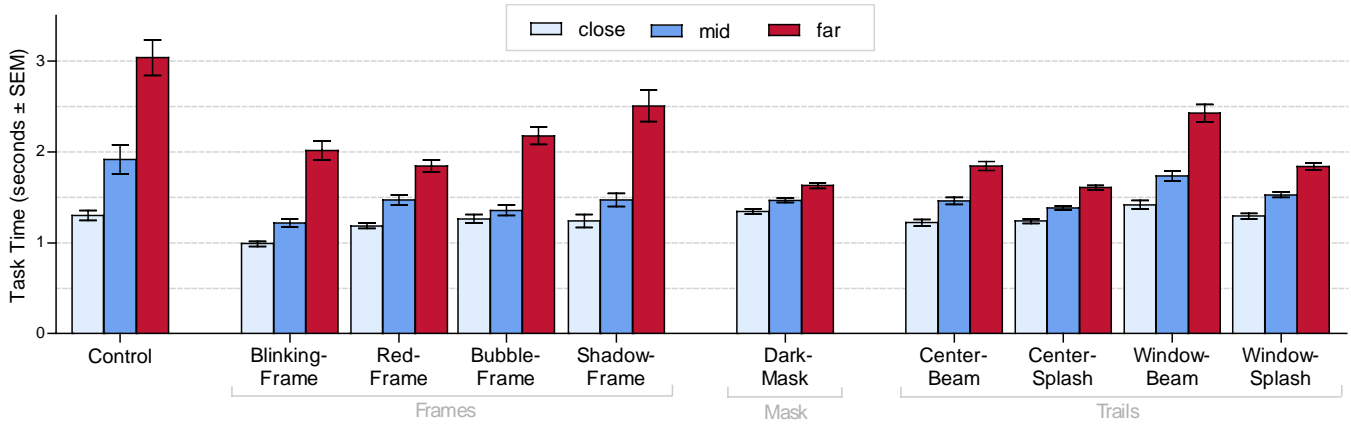


Figure 4: Task times for basic cues at varying target distances: All cues except WindowBeam outperform Control.

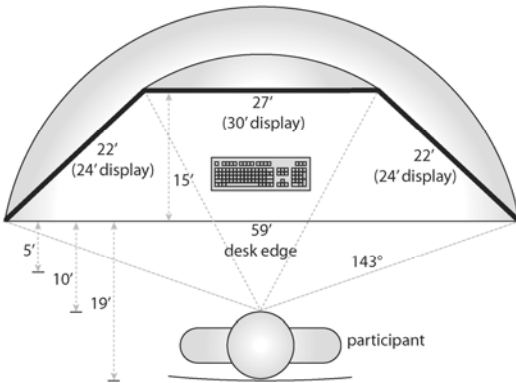


Figure 5: Setup of our three-monitor display. The field-of-view angle was 143°.

Windows XP. The angle between monitors was 43°, following the guidelines by Su and Bailey [19] suggesting an angle of 45° or less.

Participants sat on a chair which was firmly attached to the desk using a leash and whose back had a distance of 34" from the center monitor. In a pilot experiment, we noticed that some participants tried to move backwards as the study proceeded; this movement reduces the visual angle covered by the displays, making visual changes more easily detectable. We adjusted the distance of the chair such that an average person was looking at the center screen from 25" distance.

Participants

We recruited 10 participants. Eight were graduate students; one was an undergraduate, and one was employed. All participants reported at least 5 hours of computer usage daily, and half reported at least 5 hours of daily usage of dual monitors. Only one participant did not use a large display (21'-30') or dual monitors on a regular basis, and no participant regularly used three monitors. All participants had normal or corrected-to-normal vision; none were colorblind.

Hypotheses

We grouped trials by target distance from the previous focus window into three categories *close* ($d < 2000px$), *mid*

($2000 \leq d \leq 4000px$), and *far* ($> 4000px$), and had eight hypotheses. Those in **bold** were (largely) confirmed.

[H1] For target windows located far away from the previous focus window each of the eight interface conditions will result in a shorter task time than Control.

[H2] For the frame conditions the dominant features (motion and color) will perform better than those using shape and shading. Thus, BlinkingFrame and RedFrame will outperform BubbleFrame and ShadowFrame.

[H3] For the trail conditions a predictable trail origin will be more effective than a window-based trail origin. Therefore, CenterBeam will perform better than WindowBeam, and CenterSplash performs better than WindowSplash.

[H4a] For target windows far from the previous window trails will be more effective than frames, but **[H4b]** for target windows close to the previous window frames will outperform trails.

[H5] The popup windows will be more frequently missed by subjects in the DarkMask condition than in any other.

[H6] The popup windows will be more frequently missed when frames with dominant features (motion and color) are used than with shape or shading features.

[H7] Trail conditions will lead to more misses than frame conditions. Among trails conditions, the popup window will be missed more frequently when splash effects are used than when beams are used.

[H8] Users will find BlinkingFrame more annoying than other frames, and they find the mask (DarkMask) more annoying than frames or than trails.

[H9] Users will prefer to use CenterSplash than any other condition.

Results

Task Completion Time

We first report on task completion time (Figure 4), measured from the time when a cue was presented to the time when the participant hit the correct number on the keyboard. We cleaned our dataset by removing the first two trials in each condition, all trials at which a popup was displayed, and all trials at which the user entered a wrong

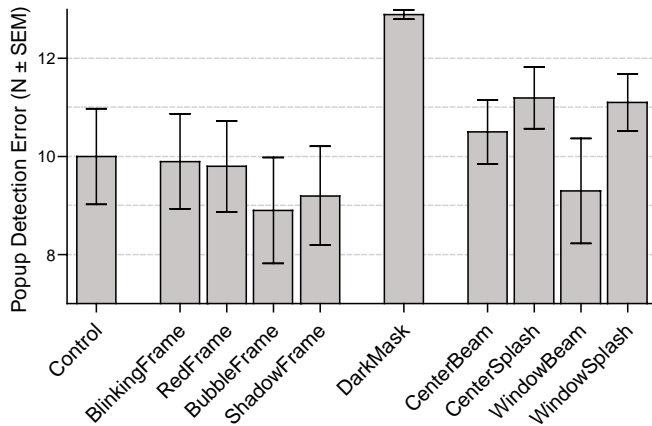


Figure 6: Pop-up detection errors of basic cues.

number. We then applied a logarithmic transformation to task time and computed the mean for each combination of *Participant*, *Condition*, and *Distance*. We performed a Mixed Model analysis, modeling *Participant* as a random effect and *Condition* and *Distance* as model effects.

Condition was significant ($F(9,261)=3.1, p<.01$), leading us to investigate pairwise differences. Comparing each cue to Control we got:

<i>Condition</i> vs. Control	F(1,279)	significance
BlinkingFrame	48.6	$p < .01 *$
RedFrame	18.1	$p < .01 *$
BubbleFrame	15.7	$p < .01 *$
ShadowFrame	9.2	$p < .01 *$
DarkMask	13.5	$p < .01 *$
CenterBeam	17.0	$p < .01 *$
CenterSplash	20.1	$p < .01 *$
WindowBeam	.04	$p > .85$
WindowSplash	7.0	$p < .01 *$

All cues except for WindowBeam thus performed significantly better than Control and hypothesis H1 was largely confirmed.

Analyzing the frame conditions, we found that cues using dominant features, BlinkingFrame and RedFrame, had lower task times than cues using less dominant features, BubbleFrame and ShadowFrame ($F(1,279)=8.9, p<.01$). This confirms H2.

Next, we compared trails with predictable locations to trails with window-based locations. CenterBeam performed significantly better than WindowBeam ($F(1,279)=8.9, p<.01$), but no difference was found between CenterSplash and WindowSplash ($F(1,279)=3.3, p>.06$). H2 therefore is partially confirmed. We later also compared each trail to the remaining three and found that WindowBeam performed worse than the other trails ($F(1,279)=19.0, p<.01$) and CenterSplash performed significantly better than the other trails ($F(1,279)=7.0, p<.01$).

We next tested for an interaction between *Condition* and *Distance*, and found an effect ($F(18,261)=3.0, p<.01$). When targets were far, trails performed better than frames ($F(1,261)=12.6, p<.01$), but when targets were close there

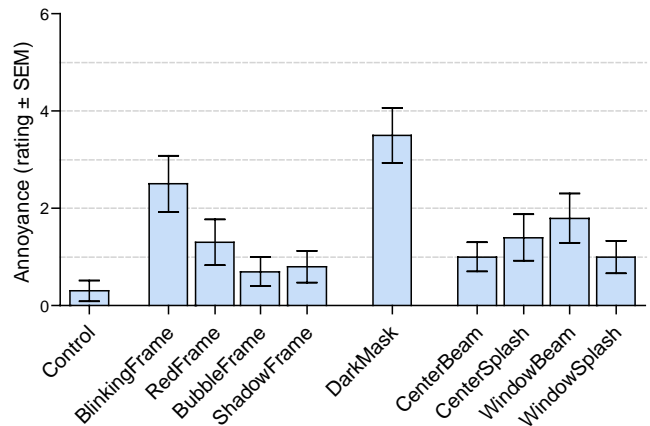


Figure 7: Visual annoyance of basic cues.

was no significant difference ($F(1,261)=2.3, p>.13$). This shows that the trails were not so distracting as to slow down users, even when the target was within close proximity. H4a is confirmed, but not H4b.

Pop-up Detection Error

We next report on participant's detection of letters in the popup windows when they switched to a newly active window (Figure 6). Popup windows were displayed in 13 of the 62 trials for each condition. We performed a Mixed Model analysis of the error, modeling *Participant* as a random effect and *Condition* as a model effect.

Condition was again significant ($F(9,81)=4.3, p<.01$), and so we looked at pairwise differences. Consistent with H5, DarkMask had a significantly higher number of errors than any other condition, as the following table shows:

<i>Condition</i> vs. DarkMask	F(1,279)	significance
Control	12.7	$p < .01 *$
BlinkingFrame	13.6	$p < .01 *$
RedFrame	14.5	$p < .01 *$
BubbleFrame	24.2	$p < .01 *$
ShadowFrame	20.7	$p < .01 *$
CenterBeam	8.7	$p < .01 *$
CenterSplash	4.4	$p < .04 *$
WindowBeam	19.6	$p < .01 *$
WindowSplash	4.9	$p < .03 *$

We also tested for a difference of frames with dominant features (BlinkingFrame and RedFrame) vs. other features (BubbleFrame and ShadowFrame) but found no significant effect ($F(1,81)=1.9, p>.16$). H6 can thus not be confirmed.

Trail conditions led to more target misses than frame conditions ($F(1,81)=7.0, p<.01$), and among the trail conditions, splash effects had more misses than beam effects ($F(1,81)=4.7, p<.04$). This confirms H7 and may imply that more visual change occurring at different parts of the screen impact user's awareness of other screen content.

Subjective Feedback

Finally, we report on subjective feedback. Using a 7-point Likert scale, participants rated visual annoyance of the cues (Figure 7) and indicated if they would use them in their

daily work. To find statistically relevant differences in the ratings, we computed an ANOVA on ranks and applied the Wilcoxon test. DarkMask was found more annoying than any other condition except for BlinkingFrame:

<i>Condition vs. DarkMask</i>	χ^2	significance
Control	13.9	$p < .01$ *
BlinkingFrame	2.8	$p > .09$
RedFrame	8.8	$p < .01$ *
BubbleFrame	12.8	$p < .01$ *
ShadowFrame	13.3	$p < .01$ *
CenterBeam	8.2	$p < .01$ *
CenterSplash	9.5	$p < .01$ *
WindowBeam	7.9	$p < .01$ *
WindowSplash	12.2	$p < .01$ *

As expected, BlinkingFrame was found more annoying than BubbleFrame ($\chi^2 = 10.5$, $p < .01$), RedFrame ($\chi^2 = 4.2$, $p = .04$) and ShadowFrame ($\chi^2 = 11.0$, $p < .01$), which confirms H8.

When asked what they would actually use, participants favor CenterSplash over many conditions:

<i>Condition vs. CenterSplash</i>	χ^2	significance
Control	9.2	$p < .01$ *
BlinkingFrame	9.9	$p < .01$ *
RedFrame	2.9	$p > .08$
BubbleFrame	7.9	$p < .01$ *
ShadowFrame	8.7	$p < .01$ *
DarkMask	3.5	$p > .06$
CenterBeam	2.1	$p > .14$
WindowBeam	7.7	$p < .01$ *
WindowSplash	0.25	$p > .61$

Yet, CenterSplash was not significantly preferred over RedFrame, CenterBeam, WindowSplash and surprisingly also DarkMask. Our data revealed that the majority of participants strongly disliked DarkMask but that two participants strongly favored it for its effectiveness. Although CenterSplash was clearly one of the most popular cues, our data did not provide enough support for H9.

After the study, we also asked participants to rank all conditions. Ordering the conditions by their average ranks we obtained CenterSplash, WindowSplash, CenterBeam, RedFrame, BubbleFrame, WindowBeam, DarkMask, ShadowFrame, BlinkingFrame, and Control.

Discussion

Our analysis revealed major differences among conditions. All frames were more effective than Control, but several participants remarked that they did not notice much difference between ShadowFrame and Control. The shadow used was very subtle and did not stand out much from other screen content. It is remarkable, however, that even with just 25%-opacity and given the relatively noisy background used in the study, basic features such as shape, color, and motion easily pop-out. As we expected, the animated BlinkingFrame was found more annoying than other frames.

DarkMask was very effective, but participants had greater difficulty in detecting popup windows with this cue. This

was not surprising, since DarkMask caused a flashing effect across the entire display. One participant reported that this flashing effect also made it impossible for him to detect the target window based on its flashing (although visual search based on other features was easy after the cue was displayed). Several participants complained that the change in luminance when displaying and removing the mask was “painful to the eye”.

WindowBeam was the only cue that did not perform better than Control, and qualitative feedback from participants provided further insights. The beam was often rendered over a large amount of screen content, reducing its visibility. Due to the horizontal configuration of monitors, WindowBeam’s 2-pixel line was frequently nearly horizontal and thus difficult to follow given the cluttered background. CenterBeam, in contrast, did not suffer from this problem, since the angle was typically much farther from rectilinear. Participants also reported that they sometimes were inclined to “chase the tail” of the line, because WindowBeam’s symmetry provided no affordance of the direction to the target window.

CenterSplash was very effective, allowed detection of popup windows, and was barely annoying. It was the participants favorite cue.

STUDY 2 – CUE COMBINATIONS

In our second study we analyzed the effectiveness, impact on other screen content, and user preferences of selected hybrid techniques, created as combinations of the basic cues tested in our first study.

Interfaces

We used 8 interface conditions: Control, CenterSplash, CenterSplash+R, CenterSplash+R+B, CenterSplash+R+B+S, WindowSplash+R+B+S, CenterBeam+R+B+S, DarkMask+B.

We included CenterSplash because of its good performance in the first study. There were 15 possible combinations with the individual frame conditions ($2^4 - 1$), but to keep the number of conditions manageable, we instead created a linear number of conditions by successively adding frames in order of descending performance: RedFrame (R), BubbleFrame (B), and ShadowFrame (S). We left out BlinkingFrame, because of its low satisfaction rating. We also included WindowSplash and CenterBeam, combined with the same three frames to match the CenterSplash condition. We did not include BlinkingFrame and WindowBeam, because of their low satisfaction ratings.

Finally, we added another variation of DarkMask to the set of interfaces. It had performed poorly on the popup detection task, but we were wondering if a different opacity might alleviate this. We reduced opacity from 75% to 25%, but preliminary experiments showed that this made it more difficult to find the target window. In addition to reducing opacity, we combined the mask with BubbleFrame, which meant highlighting a more rounded area around the rectangular target, similar to the effect in Spotlight [13].

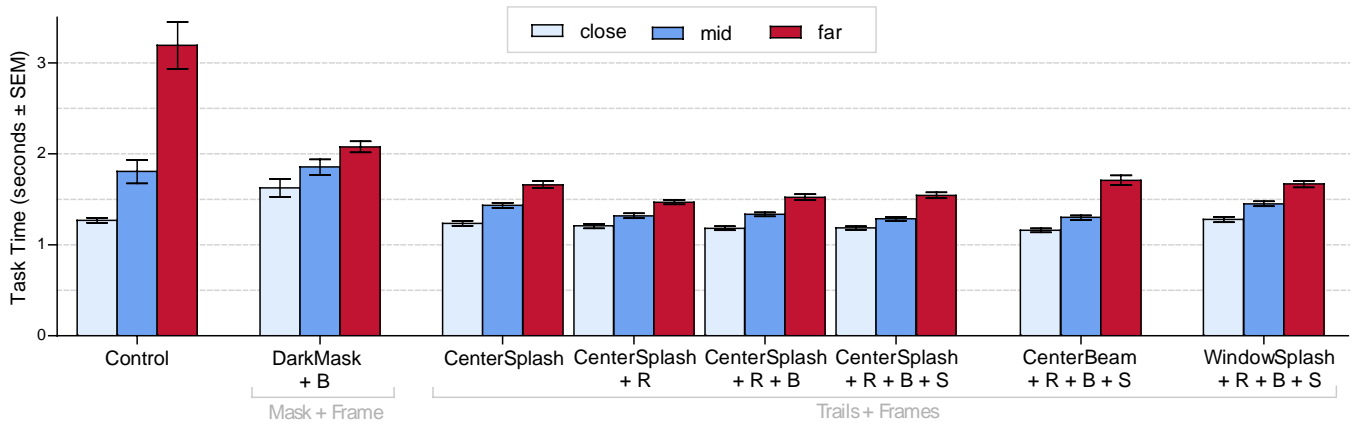


Figure 9: Task times for combined cues at varying target distances: Adding a frame to CenterSplash reduces time.

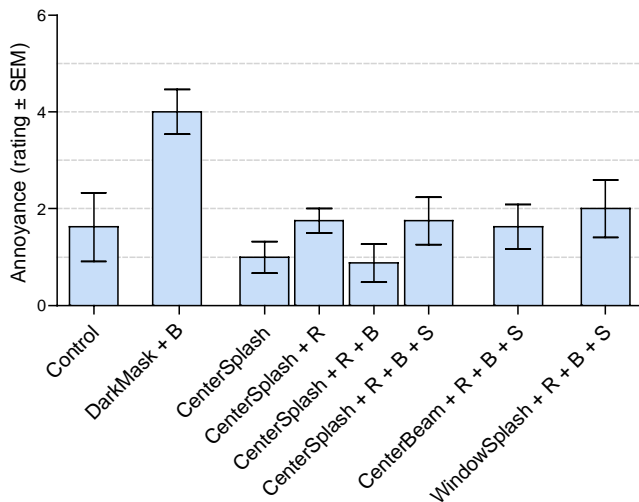


Figure 8: Visual Annoyance of combined cues.

Task, Procedure, Desktop Content, Apparatus

The procedure was identical to that of the previous study, except that conditions were counterbalanced following a Latin square of order 8 rather than 10. Task and apparatus were identical; desktop content and window activation sequences were generated using the same techniques.

Participants

We recruited 8 participants, none of whom had participated in Study 1. Five were graduate students, one was an undergraduate student and two were employed. All participants reported regular computer usage of at least 5 hours per day, and half reported using dual monitors for at least 5 hours per day. All participants had normal or corrected-to-normal vision; none were color blind.

Hypotheses

We had four hypotheses (those in **bold** were confirmed):

[H10] Adding a frame to CenterSplash will reduce task completion time. (We did not expect that adding more frames would further reduce time.)

[H11] DarkMask+B will perform better than Control.

[H12] Adding frames to the trails will increase the popup detection error.

[H13] Participants will find DarkMask+B more annoying than any other condition in this experiment.

Results

Task Completion Time

Figure 9 shows the timing results. We performed the same analysis as in our first study, again finding *Condition* to be significant ($F(7,175)=21.0, p<.01$). Consistent with our first study, CenterSplash performed better than Control ($F(1,179)=29.0, p<.01$). Adding RedFrame to CenterSplash led to another significant improvement ($F(1,179)=3.9, p<.05$), confirming H10. There were no significant differences when further adding BubbleFrame ($F(1,179)=.01, p>.91$) and ShadowFrame ($F(1,179)=.01, p>.93$). Surprisingly, the evidence was not strong enough to conclude that DarkMask+B performs better than Control ($F(1,179)=.19, p>.66$), and so we reject H11.

Popup Detection Error

With respect to popup detection error, *Condition* was not significant in this experiment ($F(7,49)=1.3, p>.28$) and so we also have to reject H12.

Subjective Feedback

Figure 8 shows the results. To evaluate participant feedback, we again computed an ANOVA on ranks and applied the Wilcoxon test, which confirmed H13 – DarkMask+B was found more annoying than any other cue. We again asked users to rank all conditions and computed an order by average ranks: CenterSplash+R+B+S, CenterSplash+R+B, CenterSplash+R, CenterSplash, WindowSplash+R+B+S, CenterBeam+R+B+S, DarkMask+B, and Control.

Discussion

Our second study shows that combining basic cues can indeed be useful. Taking one of the best performing trails from the first study, CenterSplash, and adding RedFrame led to a further reduction in task time. Furthermore, rankings of users seem to suggest that adding frames to trails does not make them less popular.

Another interesting result of this study is the poor performance of DarkMask+B. After reducing opacity to 25% but adding BubbleFrame, the cue no longer performs better

than Control, despite still being more annoying than other ones tested. We conclude that mask effects are not suitable for window switching.

CONCLUSION

The increasing use of large displays demands new visual cues for guiding attention; when using a multi-monitor system, users have greater difficulty finding the target when switching windows. This paper shows that highlighting the target window and presenting graphical trails alleviates this problem.

The results of our two studies have important implications for the design of visual cues to guide attention: (1) Visual sparseness is important. In general, more visual change means more annoyance, even when opacity is low (25%), and visual change, which is not local to the target, interferes with the user's ability to detect the target based on its onset. We therefore recommend against masks. (2) Both frames and trails are effective in reducing task time, but when targets are far, trails perform better than frames. This suggests that one may wish to adapt the cue based on target distance. (3) Trails originating at a predictable location and those originating at the last focused window are both effective, but it is crucial for trails to stand out from rectilinear screen content. We therefore suggest using asymmetric trails such as splash effects. (4) Among the frames, a variety of features easily pop-out on usual desktop content, even when shown with low opacity. The dominant features, motion and color, are more effective than shape or shadows. Unsurprisingly, however, motion was also found more annoying than any of the other three features, and so we favor the use of color, shape, and shadows. (5) Combinations of trails and frames are more effective than the cues in isolation and resulted in high subjective satisfaction.

REFERENCES

1. Austen, E. and Enns, J. T. Change detection: Paying attention to detail. *Psyche*, 6, 11.
2. Bartram, L., Ware, C., and Calvert, T. W. Moticons: detection, distraction and task. *Int. J. Hum.-Comput. Stud.*, 58, 5: 515–545, (2003).
3. Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P., Bederson, B., and Zierlinger, A. Drag-and-Pop and Drag-and-Pick: Techniques for Accessing Remote Screen Content on Touch- and Pen-operated Systems. In *Proc. Interact 2003*, Zurich Switzerland, August 2003, pp. 57-64.
4. Baudisch, P. and Rosenholtz, R. Halo: a technique for visualizing off-screen objects. In *Proc CHI 2003*, 481-488.
5. Baudisch, P., Tan, D. S., Collomb, M., Robbins, D. C., Hinckley, K., Agrawala, M., Zhao, S., and Ramos, G. Phosphor: explaining transitions in the user interface using after-glow effects. In *Proc UIST*. 2006, 169–178.
6. Bezerianos, A., Dragicevic, P., and Balakrishnan, R. Mnemonic rendering: an image-based approach for exposing hidden changes in dynamic displays. In *Proc. UIST*. 2006, 159–168.
7. Callaghan, T. C. Interference and dominance in texture segregation. *Perception & Psychophysics*, 46, 4 (1989), 299–311.
8. Chi, C.-F. and Lin, F.-T. A comparison of seven visual fatigue assessment techniques in three data-acquisition vdt tasks. *Human Factors*, 40, 4 (1998), 577-590.
9. Czerwinski, M., Smith, G., Regan, T., Meyers, B., Robertson, G. G., and Starkweather, G. Toward characterizing the productivity benefits of very large displays. In *Proc. INTERACT*. 2003, 9-16.
10. Duncan, J. and Humphreys, G. W. Visual search and stimulus similarity. *Psychological Review*, 96, 3 (1989), 433–458.
11. Hillstrom, A. P. and Yantis, S. Visual motion and attentional capture. *Perception and Psychophysics*, 55, 4 (1994), 399–411.
12. Hutchings, D. R., Smith, G., Meyers, B., Czerwinski, M., and Robertson, G. G. Display space usage and window management operation comparisons between single monitor and multiple monitor users. In *Proc. AVI*. 2004, 32–39.
13. Khan, A., Matejka, J., Fitzmaurice, G. W., and Kurtenbach, G. Spotlight: directing users' attention on large displays. In *Proc. CHI*. 2005, 791–798.
14. Kleffner, D. A. and Ramachandran, V. S. On the perception of shape from shading. *Perception and Psychophysics*, 52, 1 (1992), 18–36.
15. Pashler, H. Cross-dimensional interaction and texture segregation. *Perception and Psychophysics*, 43, 4 (1988), 307–318.
16. Plaue, C. and Stasko, J. T. Animation in a peripheral display: distraction, appeal, and information conveyance in varying display configurations. In *Graphics Interface*. 2007, 135–142.
17. Rantanen, E. M. and Goldberg, J. H. The effect of mental workload on the visual field size and shape. *Ergonomics*, 42, 6 (1999), 816–834.
18. Robertson, G. G., Czerwinski, M., Baudisch, P., Meyers, B., Robbins, D. C., Smith, G., and Tan, D. S. The large-display user experience. *IEEE Computer Graphics and Applications*, 25, 4 (2005), 44–51.
19. Su, R. E. and Bailey, B. P. Put them where? Towards guidelines for positioning large displays in interactive workspaces. In *Proc. INTERACT 2005*, 337–349.
20. Suh, B., Woodruff, A., Rosenholtz, R., and Glass, A. Popout Prism: Adding Perceptual Principles to Overview+Detail Document Interfaces. In *Proc CHI 2002*, 251-258.
21. Tan, D. S., Gergle, D., Scupelli, P., and Pausch, R. With similar visual angles, larger displays improve spatial performance. In *Proc CHI 2003*, 217–224.
22. Tan, D. S., Gergle, D., Scupelli, P., and Pausch, R. Physically large displays improve performance on spatial tasks. *ACM Trans. Comput.-Hum. Interact.*, 13, 1 (2006), 71–99.
23. Treisman, A. and Gelade, G. A feature-integration theory of attention. *Cognitive Psychology*, 12 (1980), 97–136.
24. Vogel, D. and Baudisch, P. Shift: A Technique for Operating Pen-Based Interfaces Using Touch. In *Proc. CHI 2007*, pp. 657-666.
25. Wolfe, J.M. and Horowitz, T. S. What attributes guide the deployment of visual attention and how do they do it? *Nature Reviews Neuro-science*, 5 (2004), 1–7.
26. Wolfe, J. M., Oliva, A., Horowitz, T. S., Butcher, S. J., and Bompas, A. Segmentation of objects from backgrounds in visual search tasks. *Vision Research*, 42 (2002), 2985–3004.
27. Wolfe, J. M. and O'Neill, P. Why are there eccentricity effects in visual search? visual and attentional hypotheses. *Perception and Psychophysics*, 60, 1 (1998), 140–156.
28. Zhai, S., Wright, J., Selker, T., and Kelin, S.-A. Graphical means of directing user's attention in the visual interface. In *Proc. INTERACT 1997*, 59–66.