Mouse Ether: Accelerating the Acquisition of Targets Across Multi-Monitor Displays

Patrick Baudisch, Edward Cutrell, Ken Hinckley, and Robert Gruen[1]
Microsoft Research & [1]Microsoft, One Microsoft Way, Redmond, WA 98052, USA
{baudisch, cutrell, kenh, robgruen}@microsoft.com

ABSTRACT

When acquiring a target located on a different screen, multi-monitor users face a challenge: differences in resolution and vertical and horizontal offsets between screens cause the mouse pointer to get warped, making the attempt to acquire the target difficult. *Mouse ether* eliminates warping effects by applying appropriate transformations to all mouse move events. In our user study, mouse ether improved participants' performance on a target acquisition task across two screens running at different resolutions by up to 28%. 7 of the 8 participants also strongly preferred using mouse ether to the control.

Author Keywords

Multi-monitor, bezel, mouse pointer, interaction technique.

ACM Classification Keywords

H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

INTRODUCTION

With dropping prices, the availability of screens with a small footprint, and research showing performance benefits for larger screens and multimon, an increasing number of computer users have switched to multi-monitor display configurations [3]. However, when accessing content on the other screen, for example, to pick a tool from the tool palette of a CAD program, multimon systems require users to move the mouse pointer across the gap between the screens. Figure 1 illustrates why this can be challenging.

Scenario 1: 1a shows an example of a multimon setup with screens of different resolution, as used for example by CAD users, who use the smaller screen for palette windows and email [5]. In this example, pixels on the right screen are larger than pixels on the left screen. When trying to move the mouse from the location labeled *start* to the location labeled *target* straight horizontally, the mouse pointer gets warped; instead of moving straight across the gap, it reappears with a vertical offset, causing the user to miss the target. 1b explains this mouse behavior by showing the operating system's perspective. The operating system does not know about different pixel sizes and thus moves the pointer along what it assumes to be a straight line. Unfortunately, in the physical world this leads to the described warping. Mouse ether is a program that eliminates warping behavior. When running mouse ether; the pointer crosses the gap in a straight line (1c).

Scenario 2: Even if the user has screens running at the same resolution, warping still takes place (2a). Whenever the user moves the mouse pointer across the gap, the system—oblivious of the gap—warps the pointer horizontally across (2b). The more the mouse is moving at an angle, however, the more the warping becomes orthogonal to the direction of motion, causing the user to miss the target. Again, running mouse ether eliminates pointer warping and thus allows users to acquire the target on the straight path.

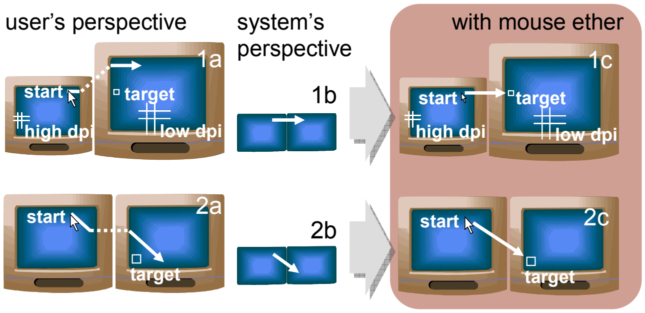

Figure 1: (a) When trying to move the mouse pointer straight to a target on the other screen of a multimon setup the pointer gets warped, because (b) the system is oblivious of resolution differences and gaps. (c) Mouse ether eliminates warping.

Without mouse ether, users continuously face an optical illusion, similar to when trying to catch a fish with a spear from above the water surface. The misleading visual feedback complicates target acquisition. Note that the "spear fishing effect" occurs across all of today's operating systems. Mouse ether eliminates this mismatch by adapting mouse behavior to the geometry that users see on their desktops. In our user study, we found that pointer warping affected user performance significantly and eliminating it using mouse ether improved user's target acquisition performance and satisfaction.

In the remainder of this paper, we look at the related work, describe how mouse ether works and what additional problem scenarios it addresses. We then report the results of our users study and close with a discussion of our findings.

RELATED WORK

Pointer warping has been studied in the context of interaction techniques that warp the mouse pointer to the target either manually (e.g., flick [4]) or based on eye gaze (e.g., [9], MAGIC pointing [11]). Visualization techniques have been proposed to help users re-acquire the mouse pointer visually, e.g., high-density cursor [2]. The drag-and-pop interaction technique avoids the need for users to cross multimon bezels by bringing potential targets to the user's current cursor location [1]. Tan and Czerwinski studied the

CHI 2004, April 24-27, 2004, Vienna, Austria.
ACM 1-58113-703-6/04/0004.

effect of monitor bezels on user's task performance [10]. The acquisition of targets under mismatching visual feedback has been studies in the context or fisheye views [6] and expanding targets [8].

HOW MOUSE ETHER WORKS: THE ALGORITHM

Mouse ether works by computing pointer locations in a device-independent coordinate system that reflects the physical size and location of each screen (addressing scenario 1) and that allows representing gaps between screens (addressing scenario 2). Figure 2 summarizes the simple algorithm. When mouse ether is launched, it reads the mouse pointer's current coordinates, translates them into device-independent real-valued *ether coordinates*, and stores them (left two dashed arrows). The ether coordinate of the pointer (e_x, e_y) is computed as the pointer's onscreen coordinates (c_x, c_y) times the screen's pixel size plus the location of the top left corner of the screen it is on.

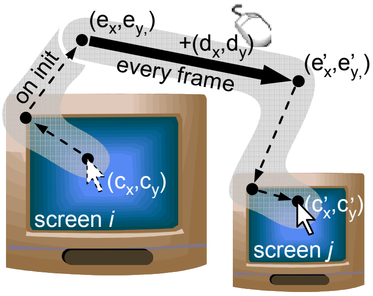

Figure 2: Mouse ether processing a mouse move

From now on mouse ether intercepts all mouse move events, so that the mouse effectively talks to mouse ether instead of to the operating system. Mouse ether adds intercepted mouse moves to its internal mouse pointer coordinates and stores the new position (e'_x, e'_y) (thick arrow). To make the mouse motion visible, the new position is translated back to screen coordinates and the pointer is displayed at that location (c'_x, c'_y) (right two dashed arrows). The back translation is done by going through the array of screens and for each screen testing whether the new mouse coordinate lies in the screen's boundaries. If so, the ether coordinates of the mouse are transformed into the screen's coordinate system using the inverse transformation to the one above. If no screen matches, the pointer is off screen and mouse ether hides the pointer.

Allowing travel through off-screen space

To make a diagonally traveling mouse pointer resurface at the correct location (scenario 2) the mouse ether algorithm makes the mouse pointer travel through off-screen space. The pointer disappears when leaving the first screen and does not reappear until the user has moved the mouse far enough to reach the other screen. By default, travel time off screen is the same as on screen, but mouse ether can be configured to use higher mouse speed while off-screen.

To prevent the mouse pointer from getting lost in off-screen space, mouse ether limits off-screen travel to the purpose of transit. Mouse ether allows entering off-screen space only at angles that (under a certain tolerance) aim for another screen unit. While off-screen, mouse motion towards open space is constrained to the closest direction aiming for a screen. Off-screen space accessible at any given time thereby forms a subset of the convex hull around screens (the translucent light areas with dashed outlines in Figure 3). Constraining space allows users to force a hidden pointer to reappear by wiggling the mouse. Whenever the user stops the mouse off-screen, mouse ether warps the pointer to the closest onscreen location.

Besides compensating for "spear-fishing" effects, off-screen travel solves another class of problems related to pointer behavior. On non-convex multimon display spaces, target acquisitions attempts that would cross an inner display border get blocked. Figure 3 shows a selection of example cases. By allowing the mouse pointer to temporarily move off-screen, mouse ether eliminates barriers and allows targets located anywhere to be acquired on the direct path.

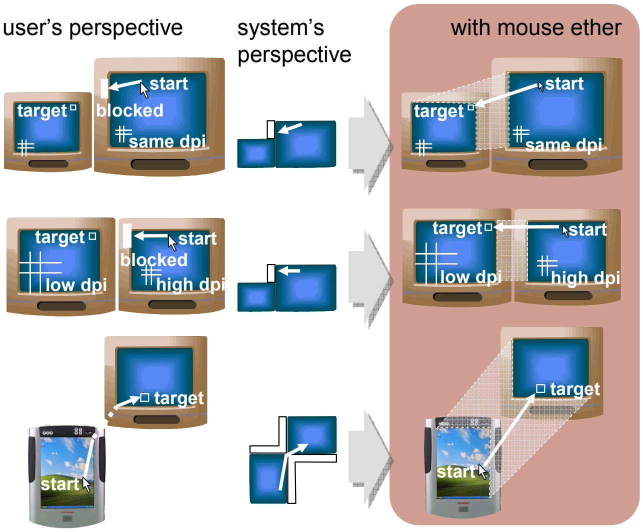

Figure 3: Without mouse ether, the mouse pointer can get blocked at screen borders on the way. Mouse ether avoids that by allowing the pointer to temporarily travel through off-screen space.

Mouse ether turns off-screen space into a medium that the mouse can travel in. The similarity to the notion of *ether*—in the 19th century physics, light waves were regarded as undulations in an all-pervading medium called *ether*—gave this technique its name.

USERS CALIBRATE USING THREE MOUSE DRAGS

When running mouse ether for the very first time, it brings up the 3-step calibration dialog shown in Figure 4. This dialog allows the user to tell mouse ether about the relative size and location of the individual screens. *Step 1:* To calibrate relative vertical location, the user drags a handle in the dependent screen (here the right screen) up or down until the bar segment lines up with the corresponding bar segment in the reference screen (here the left screen). *Step 2:* To calibrate relative pixel sizes, the user scales the right screen until the distance between the two pairs of line segments become the same; this is the case when both pairs of line segments line up. *Step 3:* To calibrate relative horizontal location, the user drags the right screen left and right until the displayed wedge appears continuous. For setups of more than two screens, the calibration is repeated for each additional screen using an already calibrated screen as the reference. For non-side-by-side setups, the calibration patterns are rotated appropriately.

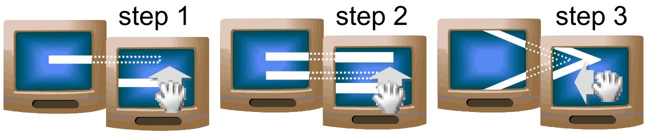

Figure 4: The mouse ether calibration procedure

USER STUDY

To objectively evaluate the performance of mouse ether, we performed a user study loosely based on a Fitts' Law target acquisition task [7]. Eight participants naïve to the technique with no experience using multiple monitors were recruited from the community for this study. Our hypothesis was that participants would acquire targets faster when using mouse ether, than when using the control interface. We expected the performance difference between mouse ether and control to be biggest in conditions where control was subject to a stronger mismatch between visual and actual pointer path.

The experiment was run on a PC running WindowsXP driving two 18" LCD monitors. The left monitor was set a resolution of 1280x1024 and the right monitor was set to 800x600, both at 60Hz refresh rate and driven by a Matrox Parhelia graphics card.

The task was administered using a modified version of WinFitts (courtesy of the Dept. of Computer & Information Science, University of Oregon). Participants read a document with general instructions for the task. For each condition (control and mouse ether), participants were allowed to play with the mouse for a short time and then performed a block of practice trials to familiarize them with the task and mouse pointer settings. They then performed a block of trials for that condition. Each block consisted of 4 trials for each of 9 different start-target combinations, and 2 directions (from high to low dpi screen and vice versa), for 72 movements per block. The starting and target locations were a set of 3 circles on each screen arranged to be symmetric about the central bezel (Figure 5a). While this arrangement is symmetric in visual space, in motor space without mouse ether this is not the case (see Figure 5b).

Starting and target circles were always on opposite sides of the screen. Each starting circle was combined with each of the 3 targets on the other screen. Because the upper and lower eccentric circles were symmetric, the 9 different movement paths were collapsed to 5 identical distances as labeled in Figure 5a. Note that while paths 2 and 4 look identical in visual space, in screen space, path 4 is longer than 2 because more of it exists in the high dpi screen (see Figure 5b). All circles were set to be the same visual size. In the high dpi (left) screen, circles had a diameter of 30 pixels, while in the low dpi (right) screen, circles had a diameter of 18 pixels. Note that because we did not systematically vary target width and distance this study cannot be considered a normal Fitts' Law task. However, our intent was to compare normal mouse control with mouse ether in a variety of screen geometries and distances.

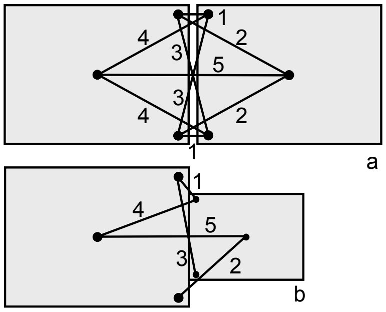

Figure 5: (a) Target layout and paths between targets as seen by participants. (b) In the control condition, the mouse moved along these actual paths. Collapsing symmetric (identically labeled) paths in (a) leads to the 5 distinct path classes shown in (b).

Note that by default, WindowsXP aligns multimon screens at the top. This warps the mouse pointer in average twice as far as when using a center-aligned setup as done in this study. Our results should therefore be considered a lower bound for the actual effect.

Results

All data analyses for movement times were performed on the median movement times for each participant and condition to normalize the typical skewing associated with response time data. Movement times were first cleaned by removing error trials and outliers (movement times greater than 4 standard deviations larger than the mean for each condition, about 0.8% of all trials). The error rate was low, 2.4% for all conditions.

We performed a 2 (Condition) x 2 (Direction) x 5 (Path) Repeated Measures ANOVA on the median movement data for each participant. There were significant main effects for all three factors (see Table 1). Mouse ether was significantly faster than the control (1311±49 ms vs. 1521±51 ms); users were faster moving from the low- to the high-dpi screen than vice versa (1369±48 ms vs. 1463±51 ms); and there was a significant effect for Path (see Figures 7 & 8). Interestingly, differences in movement times for different paths were *not* due to simple movement distance as would be typical in Fitts' tasks. In fact, the longest movement (Path 5 in Figure 5) was overall the *fastest* movement time. We look at this in more detail below.

Source	df	F	p
Condition (C)	(1,7)	69.9	<<0.001
Direction (D)	(1,7)	15.04	<0.006
Path (P)	(8,28)	3.2	<0.03
C x D	(1,7)	6.1	<0.05
C x P	(8,28)	13.2	<<0.001
D x P	(8,28)	12.2	<<0.001
C x D x P	(8,25)	4.0	<0.01

Table 1: Repeated Measures Analysis of Variance for median Movement Time.

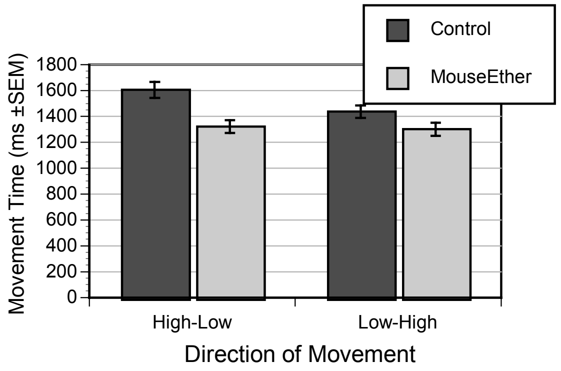

Figure 6: Interaction of Direction of movement and Condition

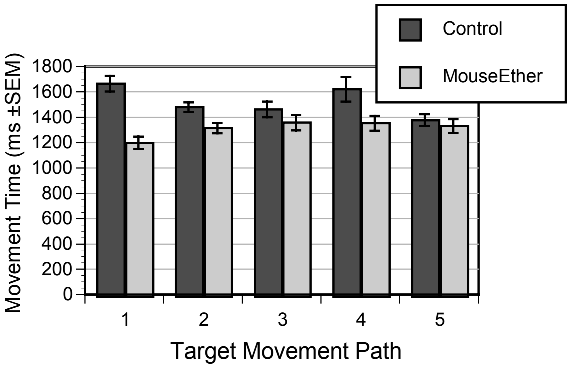

Figure 7: Interaction of Path and Condition

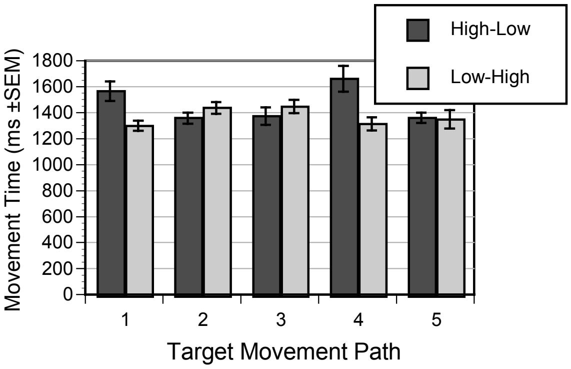

Figure 8: Interaction of Path and Direction

In addition to the significant main effects, all four interactions were significant (see Table 1). The Condition x Direction interaction reflects the fact that the difference in speed from low- to high-dpi screens is only true for the control condition; as expected, there was no difference with mouse ether (see Figure 6). The Condition x Path interaction shows that the differences in movement times for the control and mouse ether conditions varied depending on the specific movements made. From Figure 7, we see that movements for the Control were much slower than mouse ether when there was a large mismatch between the visual path the pointer should make and actual path the mouse had to traverse (Paths 1, 2, & 4). The Direction x Path interaction can be seen in Figure 8, where movements from the high- to low-dpi screen were slower than the reverse for the same reason (Paths 1 & 4). Finally, the 3-way interaction (not illustrated) appears to be due to the fact that the effect of Direction is much larger for the Control, especially for Paths 1 & 4.

In a questionnaire following this study, 7 of the 8 participants strongly preferred using mouse ether to the control for this monitor configuration.

STUDY SUMMARY AND CONCLUSIONS

Our study confirmed that mouse ether can improve user's target acquisition performance between multimon screens. As predicted, the largest effect of mouse ether was in mouse movements that were otherwise subject to a stronger mismatch between visual stimulus and actual mouse behavior. Despite their relatively modest size, our findings can be expected to have a significant impact on the daily work of multimon users. Target acquisition tasks, like those tested in the user study occur as part of many computer tasks, such as the selecting of tools from a toolbox on a second monitor. The speed up caused by mouse ether can therefore be expected to lead to small, but omnipresent time savings for multimon users.

As future work, we plan to evaluate the remaining scenarios from Figure 1 and Figure 3. Furthermore, we are working on an extended version of Fitts' law that addresses navigation tasks under mismatching visual feedback.

ACKNOWLEDGMENTS

Thanks to Mary Czerwinski, George Robertson, & Desney Tan.

REFERENCES

1. Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P, Bederson, B., and Zierlinger, Z. Drag-and-Pop and Drag-and-Pick: Techniques for Accessing Remote Screen Content on Touch- and Pen-operated Systems. In *Proc. Interact'03*, pp. 57-64.
2. Baudisch, P., Cutrell, E., Robertson, G. High-Density Cursor: A Visualization Technique that Helps Users Keep Track of Fast-Moving Mouse Cursors. In *Proc. Interact'03*, pp.236-243.
3. Czerwinski, M., Tan, D. S., & Robertson, G. G. Women Take a Wider View. In *Proc. CHI '02,* pp. 195-202.
4. Dulberg, M.S., St. Amant, R., and Zettlemoyer, L.S. An Imprecise Mouse Gesture for the Fast Activation of Controls. In *Proc. Interact '99*, pp. 375–382.
5. Grudin, J. Partitioning digital worlds: focal and peripheral awareness in multiple monitor use. In *Proc. of CHI'01*, pp. 458-465.
6. Gutwin, C. Improving focus targeting in interactive fisheye views. In *Proc. CHI'02*, pp. 267–274.
7. MacKenzie, I.S, Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction 1992.* 7: pp. 91-139.
8. McGuffin, M. and Balakrishnan, R. Acquisition of expanding targets In *Proc. CHI'02*, pages 57–64.
9. Sibert, L.E. and Jacob, R. Evaluation of Eye Gaze Interaction. In *Proc. CHI'00*, pp. 281-288.
10. Tan, D.S. and Czerwinski, M. Effects of Visual Separation and Physical Discontinuities when Distributing Information across Multiple Displays. In *Proc. Interact'03,* pp. 252-255.
11. Zhai, S., Morimoto, C., and Ihde, S. Manual and gaze input cascaded (MAGIC) pointing. In *Proc. CHI'99*, pp. 246-253.