# Bridging Gaps with Pointer Warping in Multi-Display Environments

**Manuela Waldner, Ernst Kruijff, Dieter Schmalstieg**
Institute for Computer Graphics and Vision, Graz University of Technology, Austria
{waldner | kruijff | schmalstieg }@icg.tugraz.at

## ABSTRACT

Pointer warping can be an effective alternative to relocate the mouse pointer to a remote display in multi-display environments. It minimizes the mouse pointer travel and does not require the user to search for a path to the target display. However, little is known about the factors that influence the performance of pointer warping. In this paper we explore the characteristics of pointer warping compared to standard mouse behavior on a dual-monitor setup with varying physical distance. Our results show that the performance of pointer warping is hardly affected by the distance of the pointer warp, but is influenced by the direction of the warp.

## Author Keywords

Pointer warping, multi-display environments

## ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: User Interfaces—*Input devices and strategies (e.g. mouse, touchscreen)*

## INTRODUCTION

Multi-display environments (MDE) can combine displays of varying form factors and physical arrangements. Working across these displays requires that users can move the mouse pointer from one display to another. What may seem as a trivial issue has been shown to cause noticeable interaction problems. Certain display factors in MDEs – such as depth offsets between displays, non-optimal seating arrangements [10], monitor bezels and size-resolution mismatches [2] – may negatively influence targeting performance. Pointer warping (like $M^3$ [3]) can help to overcome these limitations as it lets the user "jump" between displays by pressing a button on mouse or keyboard. It is not affected by physical discontinuities in the same extent as conventional mouse pointer navigation, as no seamless mouse pointer path across displays is required. Previous investigations have shown that pointer warping is beneficial when crossing multiple homogeneous monitors [3], accessing heterogeneous displays with strong size-resolution mismatches [4], for overcoming subjectively complex display crossings [9], and when sitting at an inconvenient location towards the displays [10]. However, by warping the mouse pointer to a remote display location, the visually perceived path between start and target location is disrupted. This disruption is known as the visual-device space mismatch [4]. Reports on pointer warping have not yet explored this problem in full extent. This is surprising, as pointer warping is usually seen as the remedy in a large, heterogeneous MDE, where direct mouse pointer navigation cannot be employed to reach all displays.

This paper explores the effects of bridging gaps between physically discontinuous displays with standard mouse and pointer warping techniques in MDEs. We compared performance and movement characteristics of pointer warping and standard mouse pointer behavior in a homogeneous dual-monitor setup with varying distance between the monitors and different movement paths. Our results show that pointer warping performance is hardly affected by the distance being warped – in contrast to standard mouse behavior. However, it is influenced by the warp direction.

## PROPERTIES OF POINTER WARPING

Standard mouse behavior for multi-monitor or more complex MDE navigation usually "stitches" adjacent display device spaces at their closest edges (*e.g.*, [5]). Pointer warping shows some inherent differences to this standard mouse behavior:

- Instead of moving the mouse pointer continuously across a display edge, the user is required to explicitly invoke the pointer warping operation, for instance by pressing a mouse button or keyboard shortcut.
- Pointer warping usually minimizes the required pointer travel distance – and thereby the index of difficulty (ID) described in Fitts' Law [6] – at the expense of an increased visual-device space mismatch.
- The pointer warping operation can be initiated from any display location, leading to a dynamically changing visual-device space mismatch.
- Depending on the outcome position on the target display, targets may lie in between start and outcome position, necessitating the user to correct the pointer movement direction after performing the warp.

Despite potentially minimized ID, previous research has shown that pointer warping does not outperform standard

*NordiCHI 2010*, October 16–20, 2010, Reykjavik, Iceland.

mouse behavior on conventional dual-monitor displays [3]. Benefits of pointer warping on homogeneous multi-monitor settings become evident only if a larger distance in device space (from 4000 pixels [3]) needs to be bridged. But, how is pointer warping affected, compared to standard mouse behavior, if a large physical gap with short device space distance has to be traversed?

## EVALUATION OF POINTER WARPING

To better understand the differences between standard mouse behavior and pointer warping, we designed an experiment to answer the following research questions:

***Q1:*** *Does an increased visual distance affect pointer warping differently than standard mouse behavior?*
Nacenta *et al.* [8] showed that with a large physical distance between monitors, minimizing the ID by warping the mouse pointer across the gap outweighs advantages of minimizing the visual-device space mismatch by using a mouse ether [2] (a technique that eliminates warping effects by compensating for visual discontinuities in the mouse device space). However, they also discovered a performance loss for warping the mouse across the gap, which they explain with extended movement planning periods due to the visual-device space mismatch and target overshooting. As pointer warping does not require a continuous movement – and therefore less amount of movement planning – it may be expected that overshooting will be less distinct compared to standard mouse behavior. We aim to evaluate the impact of visual-device space mismatch on pointer warping by changing the physical distance between adjacent monitors and comparing the effects with standard mouse behavior.

***Q2:*** *Are targets between start and outcome position harder to reach?*
Pointer warping might relocate the mouse pointer "farther" than the anticipated target location. With respect to the direction of the pointer warp, users therefore may have to readjust the pointer movement direction after performing the warp. This aspect has never been investigated before and we expect that it may be – in part – responsible for a lower performance of pointer warping on standard dual-monitor settings. To evaluate this effect, we compare task times of targets located before, on, and after the outcome position relative to the start position, and evaluate overshooting characteristics.

The experiment was conducted on a homogeneous dual-monitor setup consisting of two identical 22" wide-screen monitors (1680x1050 pixels). The study followed a 2x2x5 within-subjects factorial design with the following factors: As **navigation techniques** we employed standard mouse behavior (*mouse*), where the inner monitor edges are directly attached in the device space, and pointer warping (*warp*), where a transition to the adjacent display could be triggered by pressing the space bar. After the pointer warp, the outcome position was set to the center of the target display. Center placement is not necessarily the most adequate placement strategy for many tasks [3, 4]. However, we chose this placement for our experiment as it keeps the outcome position consistent and is therefore easier to compare across the experimental conditions.

As second factor, we varied the **distance** between the two monitors. In the *near* condition, the two monitors were placed directly next to each other, separated only by a 3.5 cm monitor bezel. In the *far* condition, the display-less space between the monitors (including bezels) was approximately the width of one monitor. In both distance conditions, the user was sitting in front of the left monitor. Mind that changing the monitor distance only altered the physical setup. The device spaces of mouse and warp were unaffected.

As third factor, we analyzed the movement **path**, defined by start- and target location. The users were asked to press alternating 50x50 pixels start and target buttons. Start buttons were always located on the left (source) display, target buttons on the right (target) display, so the experiment was limited to one movement direction. Start and target buttons were distributed to five locations on source and target display, respectively: left top (LT), left bottom (LB), center (C), right top (RT), and right bottom (RB), resulting in 25 cross-display movements. Each target location was separately analyzed as movement path (combined from five start locations). Note that the path to center (C) could be accomplished with warp without moving the mouse, as the target C was located at the outcome position of the mouse cursor after the warp. The left targets (LT and LB) were located in between the source display and the outcome position after warping. LT, LB, RT, and RB have the same ID on the target display for warp (with center placement).

Besides the task time between start and target button selections, we evaluated activity measures, such as the *time spent on the source display*, which indicates extended orientation or planning periods, as well as *distance traveled* for source and target display, respectively. For an optimal target-selection task with pointer warping, the movement distance and time spent on the source display is 0. Furthermore, we analyzed the amount of overshooting by defining a task axis [7] on the target display – from the first position the mouse pointer appears on the target display to the center of the target. We discriminate two overshooting measurements: classic *target overshooting*, and *entry overshooting*, *i.e.*, the amount of pointer movement away from the target after warping the pointer (Figure ).

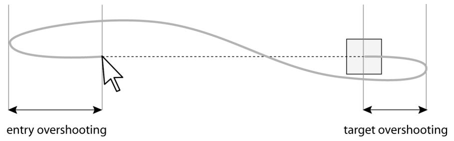

**Figure 1. Task axis connecting pointer outcome position and target position on the target display with entry and target overshooting.**

Fifteen right-handed participants (13 male, 2 female, aged 25 to 37) attended the experiment. Each participant had to accomplish four runs with 25 cross-display path sequences. 25 path sequences on the left monitor were added to prevent users from immediately switching the monitor after clicking the start button. The order of navigation technique and monitor distance, as well as path sequences, was balanced.

## Results

Apart from performance measures (*i.e.*, completion time between pressing start and target button), we additionally logged all mouse movement and keyboard events. Data was logged at a maximum frequency of 125 data points per second. Accuracy measures – like entry overshooting, target overshooting, distance traveled, and time spent on display – were extracted from these logs. All measures were evaluated using a 2 (navigation technique) x 2 (monitor distance) x 5 (path) repeated measures ANOVA with $\alpha = .05$ for main effects and interactions and Bonferroni adjustments for post-hoc comparisons. Results are summarized in Table 1.

| | *df* | *F* | | | | | |
|---|---|---|---|---|---|---|---|
| | | CT | EO | TO | DT | DS | TS |
| N | $(1, 74)$ | 23.033** | – | 44.6** | 101.1** | 208.7** | 37.5** |
| D | $(1, 74)$ | 118.745** | 2.1 | 81.1** | 118.9** | 7.1** | 38.2** |
| P | $(4, 269)$ | 42.664** | 56.3** | 12.6** | 82.5** | 2.1 | 7.4** |
| N*D | $(1, 74)$ | 11.151* | – | 78.2** | 65.2** | 17.8** | 4.8* |
| N*P | $(4, 269)$ | 23.953** | – | 52.7** | 113.8** | 1.5 | 14.5** |
| D*P | $(4, 269)$ | 2.562* | 0.6 | 26.2** | 31.7** | 0.5 | 2.8* |
| N*D*P | $(4, 269)$ | 2.585* | – | 28.9** | 33.4** | 0.9 | 0.7 |

**Table 1. ANOVA for (**$p < .001$, *$p < .05$) completion time (CT), entry overshooting (EO), target overshooting (TO), distance traveled on target display (DT), distance traveled on source display (DS), and time spent on source display (TS). Main effects for navigation technique (N), distance (D), path (P) and interactions are shown.**

### *Effect of Monitor Distance*

Post-hoc comparisons of completion time (Figure 2) show that both, mouse and warp, were significantly faster with monitors near than with monitors far ($\Delta t_m = 289.6$ and $\Delta t_w = 167.5$). However, mouse seems to be affected stronger by the changing physical gap: while mouse was faster than warp with monitors near ($t_m = 1435.1$ and $t_w = 1621.3$), there is no statistically significant difference between mouse and warp with monitors far ($t_m = 1724.6$ and $t_w = 1788.8$).

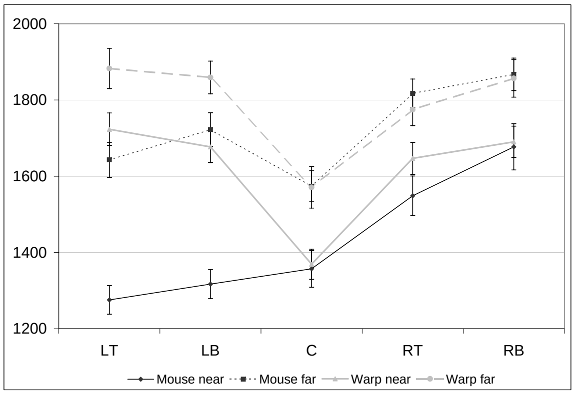

**Figure 2. Average task completion times (ms) and standard error of mouse and warp in near and far.**

Participants did not like having the monitors spaced apart, as they had to turn their head to see the target display. But with increasing distance between the monitors, they started to appreciate pointer warping: a two-factorial ANOVA of seven-point Likert scale ratings for mouse and warp on near and far, respectively, revealed an interaction between navigation technique and distance ($F_{1,14} = 17.148, p = .001$). Mouse was evaluated higher for monitors near, but there was no difference in the ratings for monitors far. Users mentioned that they felt like they "had to move the mouse farther" with monitors far and that "the mouse was too slow", whereas with pointer warping they "always knew where the mouse was located" after the warp. One user stated that "as the monitors were no longer spatially connected, the mouse pointer path was not intuitive".

### *Effect of Target Location*

As expected from Fitts' Law, left targets (LT, LB) could be selected fastest with mouse ($t_{LT} = 1459.2$ and $t_{LB} = 1519.5$) and center was selected fastest with warp ($t_C = 1470.0$). With warp, RT was selected significantly faster than LT ($t_{RT} = 1710.7$ and $t_{LT} = 1803.0$) – despite equal ID. This difference partially confirms that targets located between start and outcome position (*i.e.*, LT and LB) are harder to reach with warp.

### *Overshooting*

Target overshooting was significantly higher for mouse (57.2 px) than for warp (22.8 px). For mouse, target overshooting was higher for the left targets (108.3 px). In contrast, for warp, target overshooting was highest for the right targets (47.5 px). As also observed by Nacenta *et al.* [8], target overshooting for mouse was higher with monitors far (94.6 px) than with monitors near (19.8 px). For warp, there is no target overshooting difference between near and far (22.1 px and 23.5 px, respectively). We also measured entry overshooting for warp: For the left targets, there was significantly more entry overshooting (157.8 px) than for the right targets (2.1 px). However, there was no main effect of entry overshooting for distance (89.2 px for near and 101.8 px for far, respectively). All users in our experiment were aware of entry overshooting in warp and most could recall that the initial movement direction was towards the right. All users stated that this movement was performed unconsciously and that it was somehow annoying.

### *Activity Measures*

For mouse, there was more distance traveled on both, source and target display, than for warp. This is not surprising, as the ID for warp was much lower in our experiment than for mouse. However, the time spent on the source display was significantly higher for warp (524.8 ms) than for mouse (448.3 ms) – although there was actually no movement required on the source display for warp. For both navigation techniques, there was more time spent on the source display in the far condition than with monitors near ($\Delta t_m = 109.3$ and $\Delta t_w = 61.1$).

## DISCUSSION

We will discuss the implications of our experiment based on our research questions.

***Q1:*** Increasing the physical gap between the monitors affected both, standard mouse behavior and pointer warping. However, the increase in task completion time was higher for the mouse (about 20%) than for pointer warping (about 10%). For both techniques, we could observe an extended initial non-movement period with monitors far compared to monitors near. This is an indication for the additional effort to turn the head to the distant monitor and find the target there. For the mouse, we additionally discovered increased target overshooting for the targets located near the left display boundary – an effect also observed by Nacenta *et al.* [8]. This extended overshooting can explain the decreasing performance for mouse in contrast to warp. A longer time spent on the source display despite a lower amount of pointer travel indicates that pointer warping requires an extended planning period as compared to standard mouse behavior – irrespective whether the monitors are far or near.

Benko and Feiner [3] demonstrated a benefit of pointer warping for bridging long distances in the device space. Complementing their work, our experiment indicates an advantage of pointer warping for bridging gaps in the visual space with unchanged device space: If users do not perceive the visual space as continuous due to large physical gaps, standard mouse behavior leads to increased targeting problems so pointer warping achieves comparable performance and user acceptance. Designers of MDEs therefore should consider pointer warping not only for bridging large gaps in the device space, but also for overcoming large physical gaps between displays.

***Q2:*** Although pointer warping is not a continuous operation, the direction of the warp influences the subsequent pointer movements of the users: mouse pointer movement is first initiated in the direction of the warp and is then corrected towards the actual target location. This is reflected in the higher amount of target overshooting for the right targets and entry overshooting for the left targets with a warp direction from the left to the right monitor. Targets lying in between start and outcome location showed a slightly weaker targeting performance than those lying on the right (*i.e.*, the extension of the warp direction) or directly at the outcome position. The amount of this overshooting behavior is not related to the distance of the warp.

Due to the noticeable performance decrease and subjective annoyance by the users, situations where the user has to re-adjust the movement direction after the warp should be avoided. Designers of MDEs should consider dynamic placement strategies taking into account the start location of the warp and areas with high probability of user interaction on the target display. By setting the warp outcome position between the intersecting display boundary of the warp and the closest interaction area, important interaction regions can be reached by continuing the warp movement direction, instead of causing a contrary direction adjustment. Alternatively, additional information from head-tracking (*e.g.*, [1, 3]) or eye-tracking can help to select the optimal outcome position. However, tracking equipment can be rather obtrusive and is not always available.

## CONCLUSION

Pointer warping has been widely accepted as an alternative to conventional stitching to enable cross-display navigation in MDEs. However, pressing the trigger to initiate the warp and uncontrolled steering after the warp add an additional overhead, which makes pointer warping a slower choice if targets can be easily reached by a directed mouse movement. On the other hand, we could show that pointer warping is not strongly influenced by large physical gaps between displays. In fact, if the display space is not perceived continuous any more, pointer warping can achieve comparable performance results as standard mouse behavior, which suffers from targeting difficulties (such as overshooting), caused by the visual-device space mismatch. To improve pointer warping performance, we recommend dynamic placement strategies which take the warp direction into account so users do not need to correct their movement direction after the warp.

## ACKNOWLEDGMENTS

We would like to thank all users of our experiment for their participation. This work was funded by FFG BRIDGE 822716.

## REFERENCES

1. M. Ashdown, K. Oka, and Y. Sato. Combining head tracking and mouse input for a gui on multiple monitors. In *Proc. CHI 2005*, pages 1188–1191. ACM, 2005.
2. P. Baudisch, E. Cutrell, K. Hinckley, and R. Gruen. Mouse ether: accelerating the acquisition of targets across multi-monitor displays. In *Ext. Abstracts CHI 2004*, pages 1379–1382. ACM, 2004.
3. H. Benko and S. Feiner. Multi-monitor mouse. In *Ext. Abstracts CHI 2005*, pages 1208–1211. ACM, 2005.
4. H. Benko and S. Feiner. Pointer warping in heterogeneous multi-monitor environments. In *Proc. GI 2007*, pages 111–117. ACM, 2007.
5. B. Johanson, G. Hutchins, T. Winograd, and M. Stone. Pointright: Experience with flexible input redirection in interactive workspaces. In *Proc. UIST 2002*, pages 227–234, 2002.
6. I. S. MacKenzie. Fitts' law as a research and design tool in human-computer interaction. *Hum.-Comput. Interact.*, 7(1):91–139, 1992.
7. I. S. MacKenzie, T. Kauppinen, and M. Silfverberg. Accuracy measures for evaluating computer pointing devices. In *Proc. CHI 2001*, pages 9–16. ACM, 2001.
8. M. A. Nacenta, R. L. Mandryk, and C. Gutwin. Targeting across displayless space. In *Proc. CHI 2008*, pages 777–786. ACM, 2008.
9. M. Waldner and D. Schmalstieg. Experiences with mouse control in multi-display environments. In *Proc. PPD 2010*, pages 6–10, 2010.
10. J. R. Wallace, R. L. Mandryk, and K. M. Inkpen. Comparing content and input redirection in mdes. In *Proc. CSCW 2008*, pages 157–166. ACM, 2008.