



**HAL**  
open science

## Design and Evaluation of Mouse Cursors in a Stereoscopic Desktop Environment

Leila Schemali, Elmar Eisemann

► **To cite this version:**

Leila Schemali, Elmar Eisemann. Design and Evaluation of Mouse Cursors in a Stereoscopic Desktop Environment. 3D User Interfaces (3DUI), Mar 2014, Minneapolis, United States. pp.67-70, 10.1109/3DUI.2014.6798844 . hal-01026538

**HAL Id: hal-01026538**

**<https://hal.science/hal-01026538v1>**

Submitted on 21 Jul 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Design and Evaluation of Mouse Cursors in a Stereoscopic Desktop Environment

Leïla Schemali\*

Telecom ParisTech CNRS / LTCI / XtremViz

Elmar Eisemann†

Delft University of Technology

## ABSTRACT

In a standard desktop environment a mouse has proven to be an efficient interaction device. However, when working with stereoscopic content, the shape and behavior of a standard 2D mouse cursor is not suitable, as depth conflicts between the cursor and the content can lead to confusion.

In this paper, we show that specific 3D cursors can increase accuracy and user comfort. We first discuss several situations in which traditional cursors fail and explain their limitations and shortcomings. Based on these observations, we propose new mouse cursors that are designed for the use in a stereoscopic environment. In particular, we investigate the integration of the cursor in the scene in terms of occlusion, depth matching and shape and propose a new cursor that adapts to the 3D scene. We conducted an experiment to investigate accuracy, speed and comfort of the various interaction solutions.

**Index Terms:** I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques

## 1 INTRODUCTION

For work and play, pointing in a 3D environment is crucial; e.g., to select, modify, or place objects. The typical interaction involves a mouse to let the cursor hover over targets. While for 2D projections, artists often prefer such interaction [12, 4, 8], the situation changes when manipulating 3D content in stereovision.

Pointing has previously been explored [3, 14] and a mouse cursor seems to remain a valid tool in a 3D environment in stereovision. Nonetheless, adapting the 2D interaction to 3D is ill-defined in the case of stereovision. Both eyes receive differing information; some points in the view of one eye might not appear in the other. Hence, there is no direct relation between a 2D screen location (on a pixel level) and a corresponding location in the 3D environment. The situation also holds for 2D to 3D conversion, a common scheme in recent movie productions, in which disparity values are defined, but a camera movement is impossible. Also a cursor at screen distance can be hidden by closer objects, or it seems to float in front of objects behind the screen. Further, our human visual system allows us to focus mostly on one depth, while others are seen doubled, or diplopic, making interaction ambiguous.

Consequently, one would like to integrate the cursor in the scene and keep it on the surface of the object. Nonetheless, its projection remains ambiguous and for an arrow, the tail can become hidden, even if its tip stays visible, making the actual location difficult to judge. We examined several mouse cursors in a user study and tested the influence of various parameters. Based on the user study, we can conclude that 3D positioning of a cursor is crucial for accuracy and comfort and, in a desktop environment with active shutter glasses, a one-eyed cursor should be avoided.

\*e-mail: leila.schemali@telecom-paristech.fr

†e-mail: e.eisemann@tudelft.nl

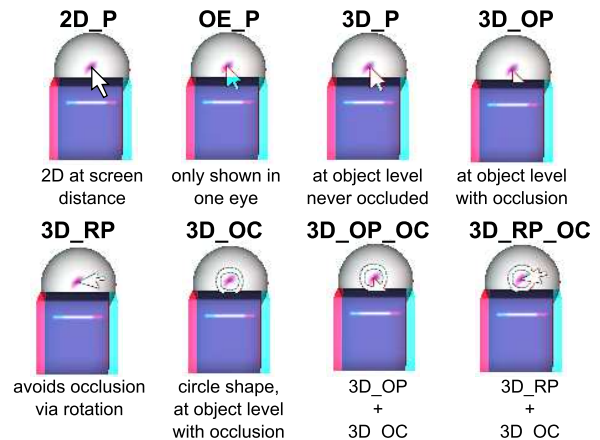


Figure 1: Cursors tested in the user study (in red-cyan anaglyph).

## 2 MOUSE CURSORS

The abbreviated names of the tested cursors reflect their functioning (Fig. 1). For the 3D mouse cursors, that define their depth based on the underlying object, we suppose to have access to the disparity value for each pixel of the stereo image.

### 2.1 Existing mouse cursors

**2D Pointer (2D\_P):** the baseline; an arrow pointing to the upper left (placed on a rectangle of  $13 \times 19$  pixels) that is always visible and shown at screen depth.

**One-Eye Pointer (OE\_P):** the 2D\_P can be shown to the dominant eye only [16]. This cursor has proven quite successful for pointing in 3D [14, 13], but we believe the single-view display leads to discomfort. Also, the cursor itself has no depth, which is disturbing.

**3D Pointer (3D\_P):** appears like the 2D\_P, but is 3D and its disparity is adjusted to project it to the underlying surface [11], while occlusion is neglected, whose influence we investigate. Calibration can be used to define the projection of the cursor [15], but, ultimately, the central position gave the best results, i.e., the cursor is shifted by half the disparity in both views. In contrast to the screen movement, movement in depth does not have to be continuous. The outcome was relatively consistent, and a discontinuous movement preferred (see Appendix).

### 2.2 Novel mouse cursors

Additionally, we investigated new mouse cursor variations, more strongly connected to the 3D scene.

**3D Occluded Pointer (3D\_OP):** similar to the 3D\_P, but can be occluded. Although we expect occlusion to be helpful in some situations — indicating the context the cursor is placed in — it proves problematic in others.

**3D Rotating Pointer (3D\_RP):** novel and rotates smoothly around its tip to move the arrow tail away from occluders (A speed of roughly  $320^\circ$  per second is preferred with high tolerance - see Appendix). The cursor integrates occlusion information in an indi-

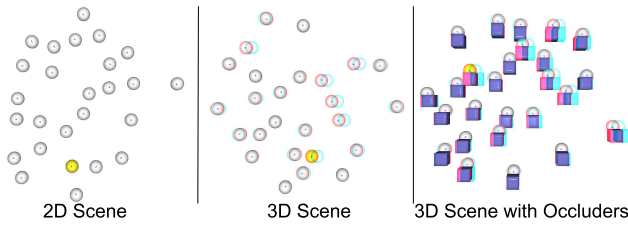


Figure 2: Scene configurations used in the user study (displayed in red-cyan anaglyph).

rect manner; as it maintains its shape and avoids occlusions (and, hereby, visible conflicts). For these reasons, we expected this cursor to be comfortable to use, while providing high accuracy.

**3D Occluded Circle (3D\_OC):** an extension to the 2D circle cursor [9]. While mouse cursor shape and orientation influences interaction in 2D or 3D [7, 9, 6], the arrow-shaped cursor is quite efficient [14, 2, 3]. However, for stereo the aforementioned problems arise, so, we also integrated circular ones. The 3D\_OC is represented as a ring, but otherwise similar to the 3D\_OP. It has no orientation and is placed at the location indicated by the ring's center, which also serves for targeting. Using occlusion, we expected the local elevation to become clearer and the symmetry to help, as it is not uncommon that such pointers can outperform others [9].

**3D Occluded Pointer with Occluded Circle (3D\_OP\_OC):** a combination of 3D\_OP and 3D\_OC.

**3D Rotating Pointer with Occluded Circle (3D\_RP\_OC):** a combination of 3D\_RP and the 3D\_OC. The cursor is rotated when occluded, but the ring can be partially occluded. Our goal was to see if a static element in the cursor would help the user; the circle follows the input perfectly, whereas the arrow rotates.

### 3 USER STUDY: METHODOLOGY

We used a point-and-click experiment in a stereoscopic desktop environment to measure clicking accuracy and speed.

#### 3.1 Participants

Twelve participants took part (6 male and 6 female, aged 24 to 33 - mean 26 years, normal or corrected-to-normal vision, with stereo perception [10]). They were compensated, but naive with respect to our research. It took in average 15 minutes per participant, including instructions, experiment and subjective feedback.

#### 3.2 Apparatus

We used a Samsung 2233RZ (120Hz, 1680x1050 pixels) screen and NVIDIA NVision 3D shutter glasses on a Windows 7 machine (Intel i7 @ 2.67GHz, 6 GB RAM, NVIDIA Quadro FX380 card) with a classic 3-button mouse device. The participants wore the stereo glasses during the whole experiment and seated at a distance of 60 cm from the screen.

#### 3.3 Procedure

We used an OpenGL application and asked participants to click on a centered target area of 1.5 arcmin size on immobile spherical targets. Non-active targets were blue cubes (Fig.2), the target sphere was rendered in yellow, the target area in red. All rendered with a simple Lambertian model and a directional light source along the view direction. Simple objects were used to minimize the bias due to scene or complexity, inspired by the recommendations of the ISO 9241-9 [1] for 2D pointing devices. All target spheres kept constant screen size (51 arcmin in our setup) independent of depth.

Spheres were placed randomly and declared targets in a random fashion. However, to make a time-performance evaluation possible, the 2D screen distance between two clicks was ensured to be

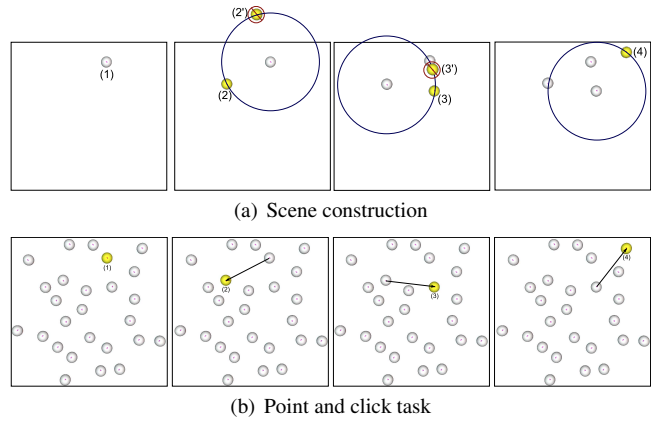


Figure 3: **a) Scene construction:** The target spheres are placed in order at random locations but subject to a constant distance, no overlap, and within the image boundary. **b) Point and click task:** During the point and click task, the current target is highlighted in yellow (1). After a click, the next target sphere is highlighted (2) until all target spheres have been clicked upon.

constant via the scene definition (explained below). Under these conditions we can assume that the task difficulty, as stated by the Fitt's law [5], was constant over all the scene configurations and cursors. The scene generation and the pointing task are illustrated in Fig. 3.

#### Scene Construction

To achieve a constant distance between two consecutive targets, we rely on a procedural randomized scene construction per configuration and participant. The scene is composed of twenty-six target spheres and other non-target objects for one of them.

**2D Scene:** 26 target spheres were positioned at screen distance. The first location is drawn randomly within the image boundaries. The next target sphere is placed randomly within the image, but with its center at a fixed 2D euclidian distance from the previous sphere center (we used 330 arcmin, which corresponds to 1/3 of the screen height). Collisions between spheres are avoided by re-drawing the random location. For this 2D scene, 2D and 3D cursors should perform similarly.

**3D Scene:** The target spheres are placed at +6 cm, +2 cm, 0 cm, -2 cm and -6 cm from the screen plane. The spheres at -2 cm and -6 cm are perceived in front of the screen, while the ones at +6 cm and +2 cm are perceived behind. The list of depths for the twenty-six spheres was pre-established such that all the combinations of two consecutive depths are covered. Here, the difference between 2D and 3D cursors should become apparent.

**3D scene with occlusion:** Additionally, we occlude the target sphere with a blue cube to test the impact of occlusion, or depth conflicts. The blue cube depth is defined such that it is perceived in front of the target it occludes, but we ensured that the target area is not occluded.

#### Tasks

The current target sphere was indicated by highlighting it in yellow (Fig. 3). The actual target (center of the sphere) was marked with a red dot. The participants clicked on all the spheres in the presented scenes with all the cursors. The order of the scenes was randomized for each participant. The order of the tested mouse cursors was randomized for each scene configuration and participant.

The participants were instructed at the beginning of the test to click as precisely as possible on the red dot of the highlighted

	d.f.	Accuracy			Movement time		
		$\epsilon$	F	p	$\epsilon$	F	p
<b>2D Scene</b>							
Cursor	7, 11	0.3	1.15	.33	0.5	2.82	*
<b>3D Scenes</b>							
Scene	1, 11	1	0.04	n.s.	1	46.5	***
Depth	4, 11	0.92	23.3	***	0.83	28	***
Cursor	7, 11	0.56	13.3	***	0.59	1.6	.19
S x D	4, 44	0.96	1.53	.21	1	0.98	n.s.
S x C	7, 77	0.45	0.52	n.s.	0.56	2.79	*
D x C	28, 308	0.24	17.3	***	0.76	4.23	***
S x D x C	28, 308	0.46	1.05	.4	0.73	0.62	n.s.

Table 1: Statistical results of the repeated measures ANOVA analysis with Huynh-Feldt sphericity correction  $\epsilon$ . The level of significance is marked with \*\*\* if  $p < 0.001$ , \*\* if  $p < 0.01$ , \* if  $p < 0.05$ .

sphere, while working swiftly, yet not with the goal of optimizing speed. At the end of the test, we asked for feedback.

### Design

We evaluated the cursors' accuracy by measuring the 2D Euclidian distance as seen in the middle-eye view between the projection of the cursors target on the surface and the target itself. We recorded the time between two consecutive clicks.

The dependent variables were the distance from the target (in arcmin) and the movement time (in ms). For the 2D scene, the independent variable was the cursor type and we used a one-way within-subjects Anova analysis. For the 3D scenes, the independent variables were the scene, the target depth and the cursor type. We analyzed the data using a 2x5x8 within-subjects Anova. We completed the Anova analysis with post-hoc Tukey-Kramer tests.

Overall, we collected 7488 clicks. We rejected 18 clicks ( $\approx 0.2\%$ ) that were identified as outliers — their distance to the target being greater than 3 times the standard deviation above the mean.

### 3.4 Results

The results of the analysis are reported in table 1.

There were no significant differences regarding the cursor type in the 2D scene when considering distance from the target. The use of a 3D cursor in a 2D environment does not lead to a difference in pointing accuracy. However, there was a significant difference for the movement time between the cursors. The Tukey-Kramer test showed that the OE.P was significantly slower than the 2D.P and the 3D.P. We think that this difference can be explained by the discomfort reported by the participants with the one-eyed pointer.

**Accuracy:** There was no main effect of the scene, but of the depth, regarding accuracy - the distance to the target. The post-hoc test revealed that the clicks were less precise for spheres perceived at  $-6$  cm than for the other depths. The cursor type also had a significant main effect on pointing accuracy. In particular, the 2D cursor is significantly less precise than all other cursors. Also, the one-eyed cursor is significantly less accurate than all 3D cursors. However, there was a significant interaction between the target depth and the cursor (Fig. 4). The 2D.P was significantly less accurate at  $-6$  cm than all the other combinations of cursors and depths. At  $-6$  cm, the one-eyed pointer was significantly less accurate than the 3D.P and the 3 circled cursors, while at  $+6$  cm it was significantly worse than the 3D.OP.OC.

There was no significant interaction between the scene and the target depth, or between the scene and the cursor. The three-way interaction between the scene, the cursor type and the target depth was not significant either.

**Movement time:** The scene had a significant main effect on the movement time in the 3D scenes. In overall, the time required be-

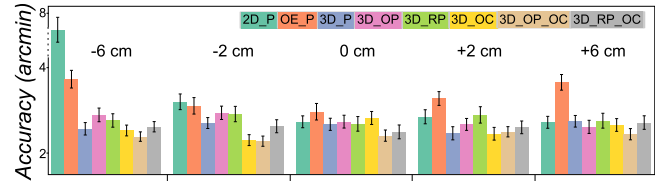


Figure 4: Distance from target (in arcmin) for the point-and-click study by target depth. Vertical error bars show the standard error.

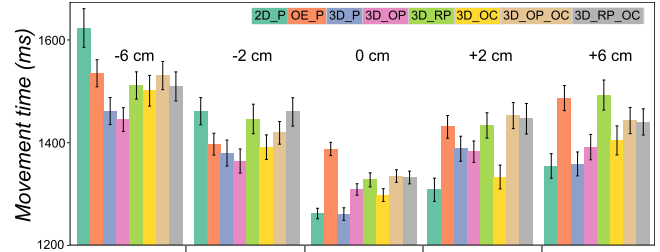


Figure 5: Movement time (in ms) for the point-and-click study by target depth. Vertical error bars show the standard error.

tween two clicks for the 3D scene with occluders was greater than for the 3D scene, due to the increased difficulty related to the occluders. The target depth had also a significant main effect on the movement time. The spheres that were at  $-6$  cm required significantly more time than the spheres at the other depths. The cursor type had no main effect on the movement time.

There was a significant two-way interaction effect between the scene and the cursor in term of movement time. The rotating pointers, the 3D.RP and the 3D.RP.OC, required significantly more time in the 3D scene with occluders. The rotating pointer 3D.RP in the 3D scene with occluders was also significantly slower than the 2D.P and the 3D.OP in the same scene. The target depth and the cursor type had a significant interaction effect (Fig. 5). The worst movement time was for the 2D.P at  $-6$  cm.

**Subjective feedback:** Once all the tasks were completed, we asked the participants to describe their study experience. Nearly all, 10 out of 12, mentioned experiencing diplopia with the 2D cursor. Eight participants out of 12 also found the one-eyed cursor disturbing because of the asymmetric viewing condition.

### 3.5 Discussion

The results show that 2D-based cursors are not suitable for 3D stereoscopic environments and are judged also subjectively by the participants as inadequate with a low comfort. This is consistent with previous studies showing the inability to use a 2D cursor in a stereoscopic scene [15]. The one-eyed cursor is also reported by the participants as uncomfortable and its accuracy is significantly lower than for the 3D cursors. Yet, the one-eyed cursor was reported by Teather *et al.* [13] to give good results in a stereo head-tracked environment. They also reported that the participants in their study could not tell the difference between a one-eyed cursor and one displayed to both eyes. We registered a different feedback from our participants, who were disturbed by the asymmetric display. We cannot explain this drastically different finding, but the setup, the equipment type or even the rendering of the cursor might play a role. Yet, it seems for desktop conditions alternative cursors are a better choice. Nevertheless, there is room for further investigations.

In our experiment, the movement time was penalized mainly by the difficulty of the scene or the target depth. The depth of the pointed object adds a difficulty to the pointing task. In our study, the worst movement time was recorded for targets presented at  $-6$  cm

in front of the screen. Identical results were found by [13], where the movement time was greater for targets in front of the screen. In [14], the *sliding cursor* – that corresponds to our 3D\_P – exhibited an increased movement time compared to other cursors. They explained this difference by the fact that the participants tended to slide all the way from the bottom of the box, along the cylinder that supported the sphere. In our scenes, the spheres were not supported by a cylinder and the participants could aim directly at the next target. Depending on the type of scenario, in which the 3D cursors are used, this aspect could be considered.

While there is no significant difference between the 3D cursors, the circled cursors, and in particular the 3D\_OP\_OC, tend to show better performance than the others for more difficult tasks. The non-oriented circle component of these cursors makes it possible to test for occlusion without the risk of hiding the cursor completely. The combination of the two depth cues, the occlusion and perceived depth of the cursor, enables users to better situate the mouse cursor in space. The rotation component of the 3D\_RP and 3D\_RP\_OC does not result in more accuracy, but increase the necessary time for the task. In addition, several participants reported to be disturbed by the rotation of the cursor that was not controlled by the movement of the physical mouse device.

The use of a 3D cursor can necessitate more rendering time as it should be positioned and occluded correctly in the 3D scene. It needs to have access to the disparity map of the scene in order to adapt its disparity to the pointed object's one. In case of a rotating pointer, the disparity map has to be inspected where the cursor is placed to detect the occlusions and rotate the pointer away from them. For mouse cursors that are occluded by the surrounding object, a test has to be added at the fragment shader level to discard the pixels of the cursor that are occluded. However, in most 3D applications, the disparity map is available and the occlusion test occurs only on the pixels of the mouse cursor. The computation overload for a 3D mouse cursor with occlusion is limited while the working comfort, accuracy and speed are increased.

We additionally found a few rules for 3D cursors that lead to best performance; the highest accuracy is reached when assuming that the clicked point lies in the central position, the *middle-eye* view. The movement in depth induced from the underlying geometry for 3D cursors can and should be made discontinuous. Finally, pointing is most challenging for objects in front of the screen. Hence, if one is to improve cursor performance, a specialized solution for these cases seems most fruitful.

## 4 CONCLUSION

For applications involving stereoscopic interactions in a desktop environment, a 3D mouse cursor is crucial. We presented several cursor types and investigated their use in different tests. We found that the 2D pointer, as well as one-eyed solutions should usually not be employed in stereo-viewing conditions in a desktop environment; comfort and precision both suffer. 3D cursors are more useful when involving the underlying geometry. Unfortunately, using a standard arrow with occlusion can be harmful as there is a potential that the cursor is hidden. By rotating the cursor, an arrow can remain useful, but the movement time increases significantly. A better result with higher accuracy and comfort is obtained when using circle pointers, optionally combined with an arrow.

## 5 ACKNOWLEDGEMENTS

This work was partially supported by the Intel VCI at Saarland University and Useful Progress.

## REFERENCES

[1] ISO 9241-9 Ergonomic requirements for office work with visual display terminals (VDTs) - Part 9: Requirements for non-keyboard input devices. International Organization for Standardization, 2000.

[2] F. Argelaguet and C. Andújar. Visual feedback techniques for virtual pointing on stereoscopic displays. In *VRST*, pages 163–170. ACM, 2009.

[3] F. Bérard, J. Ip, M. Benovoy, D. El-Shimy, J. R. Blum, and J. R. Cooperstock. Did "minority report" get it wrong? Superiority of the mouse over 3D input devices in a 3D placement task. In *Human-Computer Interaction INTERACT 2009*, pages 400–414, 2009.

[4] E. Eisemann, S. Paris, and F. Durand. A visibility algorithm for converting 3D meshes into editable 2D vector graphics. *ACM Trans. Graph.*, 28(3), 2009.

[5] P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47:381–391, 1954.

[6] D. A. G. Jáuregui, F. Argelaguet, and A. Lécuyer. Design and evaluation of 3D cursors and motion parallax for the exploration of desktop virtual environments. In *3DUI*, pages 69–76. IEEE, 2012.

[7] A. Kadri, A. Lécuyer, and J.-M. Burkhardt. The visual appearance of user's avatar can influence the manipulation of both real devices and virtual objects. In *3DUI*, page 11. IEEE, 2007.

[8] J. McCann and N. S. Pollard. Local layering. *ACM Trans. Graph.*, 28(3), 2009.

[9] B. A. Po, B. D. Fisher, and K. S. Booth. Comparing cursor orientations for mouse, pointer, and pen interaction. In *CHI*, pages 291–300. ACM, 2005.

[10] W. Richards. Anomalous stereoscopic depth perception. *J. Opt. Soc. Am.*, pages 410–414, 1971.

[11] F. Steinicke, T. Ropinski, G. Bruder, and K. Hinrichs. Interscopic user interface concepts for fish tank virtual reality systems. In *VR*, pages 27–34. IEEE Computer Society, 2007.

[12] M. Stroila, E. Eisemann, and J. Hart. Clip art rendering of smooth isosurfaces. *IEEE Trans. Vis. Comput. Graph.*, 14(1):135–145, 2008.

[13] R. J. Teather and W. Stuerzlinger. Pointing at 3D target projections with one-eyed and stereo cursors. In *CHI*, pages 159–168. ACM, 2013.

[14] R. J. Teather and W. Stuerzlinger. Pointing at 3D targets in a stereo head-tracked virtual environment. In *3DUI*, pages 87–94. IEEE, 2011.

[15] D. Valkov, F. Steinicke, G. Bruder, and K. Hinrichs. 2D touching of 3D stereoscopic objects. In *CHI*, pages 1353–1362. ACM, 2011.

[16] C. Ware and K. Lowther. Selection using a one-eyed cursor in a fish tank VR environment. *ACM Trans. Comput.-Hum. Interact.*, 4(4):309–322, 1997.

## A MOUSE PARAMETERS

Two preliminary studies regarding cursors parameters were examined with 7 participants on the same system and setup.

**Depth movement** A non-continuous depth motion is advantageous for all 3D cursors. The study used a simple city model seen straight from above and had the participant click on highlighted buildings. While moving the mouse, the cursor passed over several buildings. Here, the 3D cursors adapt their height to the corresponding roof tops. The 3D\_OP cursor in this task was used with a direct and a slow adaptation to the underlying depth. Six participants preferred a direct and discontinuous depth adjustment and one perceived no difference. People do not seem to follow the mouse movement with their eyes, they travel large screen distances intuitively by focusing on their target. The cursor is only important near the target. A slow depth movement can lead to a cursor off the surface when approaching the target, leading to a waiting time that was judged as uncomfortable even when in the orders of milliseconds.

**Rotation time** A rotation time of roughly 320° per second is preferred. For the study participants had to click on a sphere using the 3D\_RP cursor. The sphere was half occluded so the cursor needed to rotate by 135° to avoid occlusion. The task was repeated 20 times for different speeds in  $2\pi \times p$  per second, with  $p \in [0, 1]$ . A slow rotation was considered disturbing, if it induces a waiting time. A very fast rotation was judged disturbing at first and takes some time getting used to. For  $p \geq 0.9$ , little disturbance was reported and we used  $p = 0.9$ .