EGVE Symposium (2008)
B. Mohler and R. van Liere (Editors)

Tangible Interaction for 3D Widget Manipulation in Virtual Environments

K.J. Kruszyński[1] and R. van Liere[1,2]

[1]Centrum voor Wiskunde en Informatica, Amsterdam, The Netherlands
[2]TU Eindhoven, The Netherlands

Abstract

In this paper we explore the usage of tangible controllers for the manipulation of 3D widgets in scientific visualization applications. Tangible controllers can be more efficient than unrestricted 6-DOF devices, since many 3D widgets impose some restrictions on how they can be manipulated. In particular for tasks that are in essence two-dimensional, such as drawing a contour on a surface, tangible controllers have advantages over 6-DOF devices. We have conducted a user study in which subjects draw a contour on a three-dimensional curved surface using a 3D contour drawing widget. We compared four different input methods for controlling the contour drawing widget and the viewpoint of the surface: using one 2D mouse for drawing and viewpoint selection, using a 6-DOF pen for drawing and a 6-DOF cube device for viewpoint selection, using a 6-DOF pen for drawing on a tangible 6-DOF cube which implements a Magic Lens style visualization technique, and using a 2D mouse for drawing and a 6-DOF cube for viewpoint selection. We show that while the mouse outperforms 6-DOF input methods, the tangible controller is superior to unrestricted 6-DOF input.

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Methodology and Techniques – Interaction Techniques H.5.2 [Information Interfaces and Presentation]: User Interfaces

1. Introduction

Interaction with visualizations of three-dimensional data often involves 3D user interface widgets. A widget is a combination of geometry with certain behavior which is used to control the visualized data, or to query information about that data. The widget is placed in the 3D scene and can then be manipulated. Examples of 3D widgets are widgets for probing data values inside a data set, placement of seed points for creating streamlines in a vector field, placement and orientation of slicing planes, drawing and manipulation of selection contours, manipulation of bounding boxes for extracting part of a data set, or scaling, orienting and placing a geometric structure.

While widgets are interaction techniques for manipulating data, the widgets themselves are manipulated using an input controller. These controllers vary in the number of degrees of freedom (DOF) they have. For example a desktop mouse has two degrees of freedom, while a sensor which reports position and orientation in space has six degrees of freedom, thus offering a very natural and direct way of interaction with three-dimensional objects. However, 3D widgets often place restrictions on the way in which they can be manipulated, and often not all degrees of freedom are necessary to use such a widget, while sometimes the additional freedom of movement makes widget manipulation more difficult.

Tangible 3D input controllers restrict the number of degrees of freedom, and thus can provide a method of interaction which corresponds better to the way in which a widget allows itself to be manipulated. This can especially be the case for manipulations that are in essence restricted to two dimensions, such as placing points on a plane or drawing on a surface. In addition the use of tangible controllers can give the user a better perception of where the manipulation is taking place than manipulation using an unrestricted 6-DOF controller.

We have conducted a user study to determine the advantage of using a tangible controller to restrict the number of degrees of freedom of an input device when manipulating certain 3D widgets in a 3D environment. We compared the use of four different input methods both for manipulating a widget for drawing a contour on a 3D surface, and for controlling the camera viewpoint. The first method uses only the mouse to control both the widget and the camera. The second method uses two 6-DOF devices for a total of 12 degrees of freedom, six for camera manipulation and six for unrestricted use of the widget. The third method uses a 6-DOF cube device as a tangible interaction surface to restrict a 6-DOF pen device to just two degrees of freedom, for a total of eight degrees. A Magic Lens style projection on the tangible surface allows precise manipulation of the widget with two degrees of freedom, while the camera can still be freely manipulated. The last method retains the 6-DOF camera control, while the mouse is used for manipulating the widget. We show that in a desktop environment a mouse is the best input method for this widget, while for a virtual environment lacking a mouse the tangible controller has a significant advantage over unrestricted 6-DOF input.

2. Related Work

One method of interacting with 3D worlds is by using 3D versions of 2D user interface elements, positioned in the 3D world. Lindeman *et al.* [LST01] discovered that using 3D representations of 2D widgets has mixed effects on user performance. However, even simulated constraints on the input devices already can improve the performance. Kok and van Liere [KvL04] have investigated the use of standard 2D GUI widgets which were placed in a 3D scene and controlled with 6-DOF devices in a mirror-based desktop VR environment. They noted performance improvements with co-location as well as when using tangible controllers. Lindeman *et al.* [LSH99] also used tangible 6-DOF controllers to place a virtual 2D window in a VR scene in order to use symbolic 2D interaction techniques in such an environment.

Conner *et al.* [CSH*92] proposed to use true three-dimensional widgets instead of 3D versions of 2D widgets or direct manipulation of 3D objects using 3D input devices. In the absence of tactile feedback Mine *et al.* [MFPBS97] propose a framework for 3D interaction exploiting a person's sense of the location and orientation of his limbs and body.

Bier *et al.* [BSP*93] introduced Magic Lens filters, a 2D visualization technique which interactively shows a different representation of data in a movable region of the screen using the lens metaphor, combining this with click-through interaction. Viega *et al.* [VCWP96] extended the Magic Lens to 3D environments by using flat and volumetric lenses to select a part of the scene to be rendered using a different representation. Brown and Hua [BH06] projected alternate views of a scene onto the virtual representations of input devices in an AR environment.

We combine tangible 3D input controllers, in the form of drawing with a pen on a cube, with a Magic Lens style projection of a part of an iso-surface onto one of the sides of the cube in order to draw contours on that surface. This creates a virtual 2D display with 2D interaction in the 3D world, not unlike an ordinary 2D window, but one that offers 2D controls for 3D widgets.

3. Evaluation

3.1. Task

We have conducted a simple user experiment on a desktop VR setup. The subject is presented with a surface representation of a part of a coral, to which a sphere has been attached. This object is shown in figure 1. The task is to 'remove' the sphere by drawing a contour on the object, along a pre-computed 'optimal' line which is shown on the surface of the object. The subject can draw a contour by marking points on the surface of the object, using various input methods to determine the location of these points. The contour widget automatically pins the points to the surface of the object, and interpolates a contour between them. Once a point is placed it cannot be modified or removed, but it is possible to select the first point, which is therefore shown in a different color. Each trial ends when the first point is selected again, thus closing the contour. The widget also includes functionality for modifying or removing already placed points, but this has been disabled for our study. As a small 'reward' for the subject, after closing a contour the sphere is removed from the object along the contour, and the hole in the surface is tessellated.

The task is inspired by the need of coral researchers to interactively remove unwanted parts of a CT scan of marine corals before automated analysis or simulation. A future version of the analysis system will likely include this functionality.

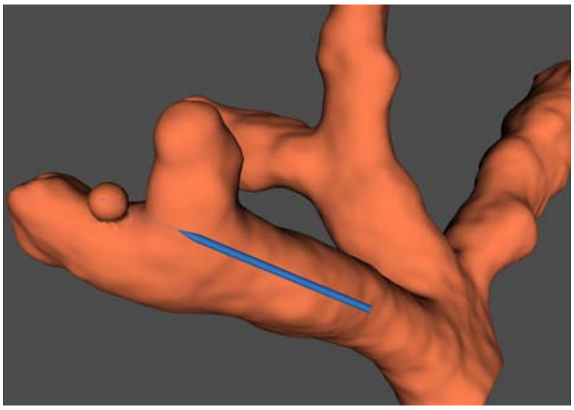


Figure 1: *The test object with a sphere attached to it. The blue object is the virtual representation of the 6-DOF pen input device*

3.2. Setup

The software used for the experiment has been implemented using the Visualization Toolkit (VTK) [SML96, Kit03], a popular framework which provides various scientific visualization techniques as well as numerous 3D widgets, including the contour widget used for the experiment. We have extended this toolkit with support for the use of 6-DOF input controllers both to manipulate the viewpoint and to control the 3D widgets included in the toolkit, as VTK does not provide support for such devices but only supports 2D mouse and keyboard input. The 6-DOF input support was added using the Virtual Reality Peripheral Network (VRPN) [RMTHS*01] library.

The hardware setup for 3D input consists of a Polhemus Fastrak electromagnetic 6-DOF tracker with two wired sensors, and three foot pedals located on the floor. One of the Fastrak sensors is attached to a pen, the other is attached to a 7.5*cm* cube made out of thick cardboard. The center of the working space for these devices is located halfway between the display and the user. The display is a standard 20 inch LCD panel. The left and right foot pedals are swapped if a test subject is left-handed. A photograph of the setup is shown in figure 2.

Figure 2: *Photograph of the test setup while using the tangible cube.*

3.3. Input Methods

The input methods not only differ in how contour points are placed but also in the way in which the viewpoint can be manipulated. Each trial starts from the same distant viewpoint showing the whole object, with the sphere not necessarily being visible. The test subject then has to orient and zoom the view before the removal task can be started. It is also necessary to adjust the viewpoint during the task, because the ‘optimal’ contour is always partially occluded by the object.

Four different input methods are compared:

3.3.1. Method 1

The location of a point on the rendered surface is selected using 2D point and click input with the mouse. The camera is also controlled with the mouse. The orientation of the viewpoint can be rotated around a focal point by dragging the mouse cursor over the background of the 3D scene while pressing the left mouse button. Pressing the middle button and dragging moves the whole scene parallel to the viewing plane. Pressing the ‘f’ key on the keyboard picks the location of the mouse cursor as the new location for the focal point and causes the camera to move closer to this point. Rotating the mouse wheel moves the camera closer to and farther from the focal point. No 6-DOF input is used in this method. A diagram of this method can be seen in figure 3(a).

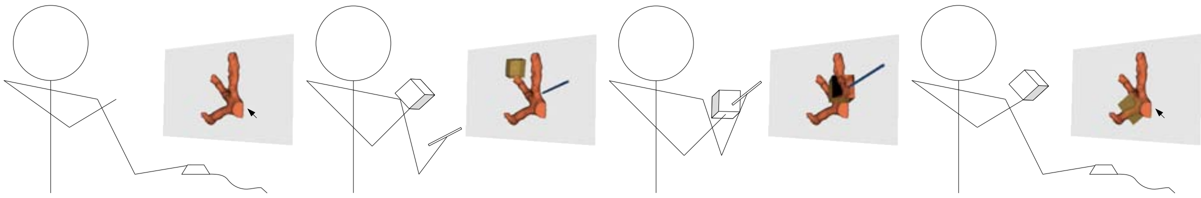
3.3.2. Method 2

The location of a point is selected by using the 3D pen. A line protrudes from the tip of the pen, and pressing the right pedal places a contour point at the intersection between this line and the surface. The line is terminated with a thicker dot in order to make it easier to determine whether the line intersects with the surface or not: if the dot is not visible then the two intersect. The camera is controlled with the 3D cube held in the other hand, employing the scene in hand metaphor. Pressing the left pedal causes the whole scene to move along with the cube and to rotate around the center of the cube. Pressing the middle pedal and moving the pen away from the screen causes the camera to zoom in towards the center of the cube. A diagram of this method can be seen in figure 3(b).

3.3.3. Method 3

In this method a Magic Lens style projection of the scene is rendered onto one of the sides of the virtual representation of the 3D input cube. The location of a contour point is selected by touching the cube with the pen on the appropriate side of the cube at the location where the point is to be placed and pressing the right pedal. A white cross-hair cursor on the projection facilitates selecting the correct location. A diagram of this method can be seen in figure 3(c).

The camera is controlled in exactly the same manner



(a) Method 1: 2-DOF mouse (b) Method 2: 12-DOF input (c) Method 3: 8-DOF tangible (d) Method 4: 8-DOF with mouse

Figure 3: *Diagrams of the four methods compared in this paper.*

as in method 2 with the cube controlling the orientation and position when the left pedal is pressed, and the pen controlling the zoom towards the cube when the middle pedal is pressed. Note that while the left pedal is pressed the projection on the cube will remain unchanged when the cube moves, which allows for greater precision when selecting a contour point location on the projection. This fact is explicitly explained to each test subject.

3.3.4. Method 4

In this method the location of contour points is selected with the mouse like in method 1, but the camera is controlled with the 3D cube like in methods 2 and 3. The zoom function is controlled with the mouse wheel, with the difference that the camera zooms in on the center of the cube instead of the focal point. A diagram of this method can be seen in figure 3(d).

3.4. Measurements

We measure the total trial time, the time needed to draw the contour and the accuracy with respect to the reference contour. The trial time is the time elapsed from the moment the test subject is presented with the view of the scene until the moment the subject closes the contour. The contour drawing time is the time between the moment when the first point is placed and the moment when the contour is closed by selecting the first point again. The accuracy is defined as the root mean square distance between the contour points selected by the test subject and the reference contour shown on the surface of the test object during the trial.

The different methods are tested in the order in which they are described. Each user is first instructed how the input method works and is given time to practice with the input method until he or she gains sufficient skill in placing contour points and modifying the viewpoint. The subject is then asked to use the method to carry out the removal task as quickly and as accurately as possible, repeating it a number of times with the sphere at different locations. The locations and the order of the locations of these spheres are previously generated randomly but are always the same for all methods and all subjects.

After finishing the test with the last method, the subject is asked to order the methods from the easiest to use to the hardest to use, and is asked which of the two methods not involving a mouse they would prefer to use in an environment where there is no mouse available.

3.5. Tangible Magic Lens

The Magic Lens shows a parallel projection of the scene visible in the main view using a camera position and orientation which is controlled by the 6-DOF cube, and which is constantly updated as the cube moves. This projection is rendered to an off-screen window, which is then texture-mapped to the side of the virtual representation of the 6-DOF cube in the main view. The cube thus works like the LCD viewfinder on a digital camera. The scale of the parallel projection is chosen to exactly match the size of the virtual representation of the cube, and the direction of projection is opposite to the normal of the side of the cube, so the lens will show what is directly behind the cube. This is shown with the white dashed lines in figure 4. The front clipping plane of the camera coincides with the side of the cube, thus the lens never shows any part of the scene which is located in front of the cube. This makes it possible to use the lens to inspect some difficult to reach occluded areas of the scene without changing the viewpoint of the main view.

The lens also acts like a magnifying glass. Because of the parallel projection the view in the lens will have the same size regardless of the distance between the cube and the object, while the perspective of the main view will enlarge the cube, and thus the projection, when the input device is moved away from the screen. This effect can also be seen in figure 4. The lens can additionally be used as a mirror, since the virtual representation of the cube is largely transparent and the back of the projection is visible through the cube. This makes it possible to perform interaction on the rear side of an object without adjusting the main view.

The main purpose of the lens is not inspection but interaction. All 3D widgets are placed both in the main scene and in the Magic Lens view, and any manipulation of a widget in the Magic Lens projection is also instantly visible in the main view. Widgets are manipulated using the cube as a tangible controller; a second input device is used to select a location on this cube. This location on the actual

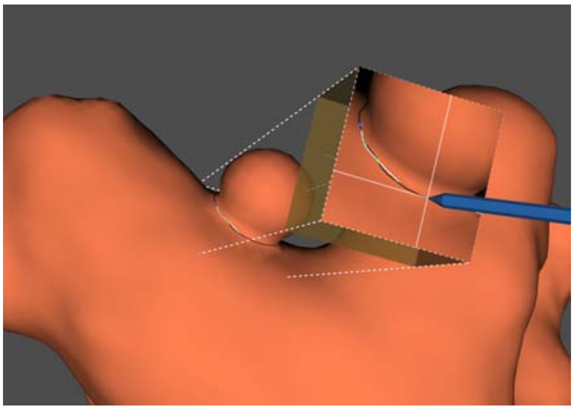


Figure 4: *Screen capture of the tangible Magic Lens, the white dashed lines show which part of the scene the lens is currently showing.*

device matches the corresponding location on the virtual representation of the cube. This location of the second input controller is then translated to a location on the texture used for the Magic Lens, and interaction takes place by emulating mouse input on the off-screen window. When the two devices touch a translucent white cross-hair cursor is shown in the lens, extending to the edges of the cube and centered on the input location, serving to facilitate selection of the desired location. Since the interaction sends mouse commands to the off-screen window in principle any kind of interaction which is possible in an on-screen window with a mouse can also be performed in the same way on the Magic Lens projection using the tangible controller.

We have chosen to use a second 6-DOF pen device to select a location on the cube by using this pen as a 2-DOF device on the surface of the cube, but the same could be achieved using any (two-handed) 8-DOF controller, for example if a 6-DOF input sensor would be attached to a digital graphics tablet.

4. Results

We conducted the experiment with 11 users with various levels of experience with 3D input devices. The users completed 7 trials for every method. The practice time before each method ranged from 1 to 10 minutes depending on the user and the particular method.

We assumed and also observed that the contours for the different spheres cannot be drawn in the same time and with the same accuracy even by the same users and when using the same method, and that some users are more meticulous than others. To compare all the different trials all results for a particular user and sphere are normalized with respect to the results when using method 1, which therefore always have a value of 1. Also an analysis of variance (ANOVA) was performed on the data.

The average contour drawing time for each of the methods, relative to method 1, is shown in figure 5(a). The difference in time between method 2 and method 3 is not significant, while all other differences are significant ($p < 0.01$).

The average trial time for the four methods, relative to method 1, is shown in figure 5(b). The difference in time between method 2 and method 3 is not significant, while all other differences are significant ($p < 0.01$).

The contour time as a percentage of the total trial time is shown in figure 5(c). The difference in mean trial time percentage between method 1 and the other methods is significant ($p < 0.01$), while all differences between methods 2, 3 and 4 are not significant.

The accuracy of each method relative to method 1 is shown in figure 5(d). The difference between method 1 and method 4 is not significant, while all other differences between the methods are significant ($p < 0.01$).

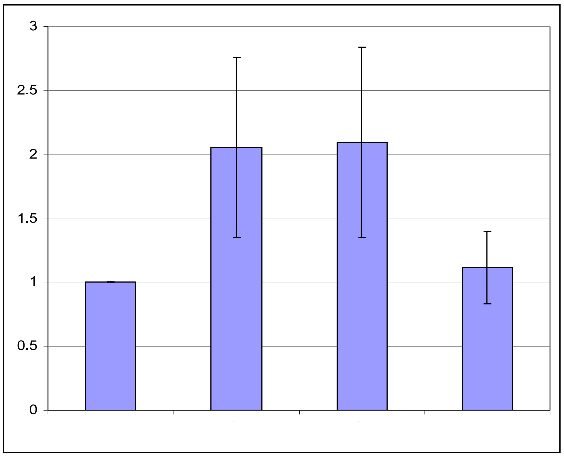
All users preferred the methods using the mouse with and without the cube over the methods using only 3D devices. All but two users preferred using the mouse with the cube over using only the mouse, and all but one user preferred using the tangible Magic Lens cube over using the freehand technique.

5. Discussion

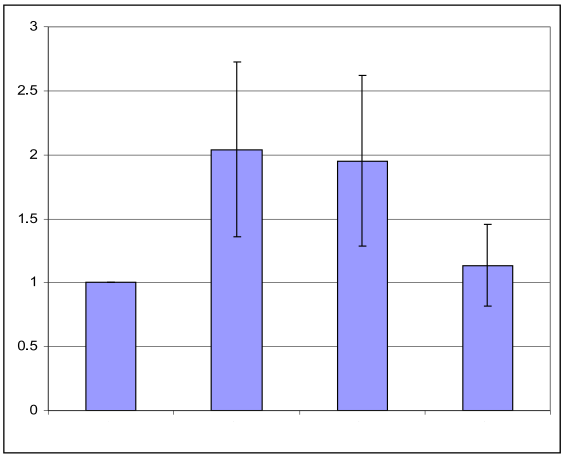
The accuracy and precision of the 3D tracking system is an important issue. Especially the tangible input controller used in method 3 relies on a correct calibration between the pen and the cube, as the on-screen representation of these devices needs to exactly match their relative positions and orientations, otherwise it is not possible to draw on the cube. The standard 2D mouse has a higher accuracy than the 3D input device. On the other hand 3D devices are more efficient for manipulating the world or 3D objects. Many novice users are also accustomed to using a mouse, while they find the 3D devices difficult to use.

The most preferred method was using the 2D mouse with the 3D world in hand, while using only the mouse was preferred over both 3D-only methods. Of these methods, the Magic Lens method was considered easier to use than the freehand drawing method.

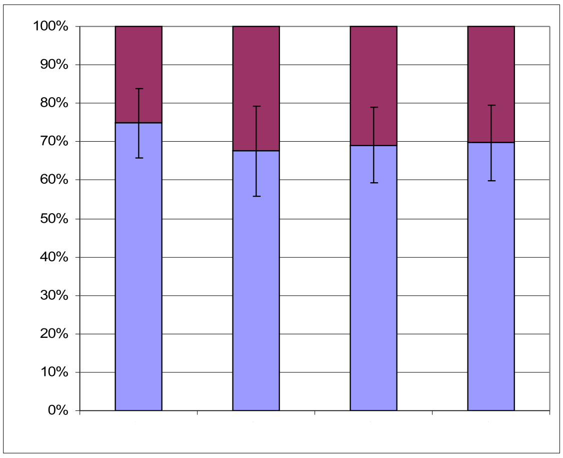
The results show that the methods using the mouse are more accurate and certainly faster than the 3D input methods. Combining the mouse with 3D camera control is actually slightly slower than using the mouse alone, even though most users preferred this method. The users perceived the 6-DOF camera control of this method as an advantage over using only the mouse for this purpose, although the measurements do not support their view.



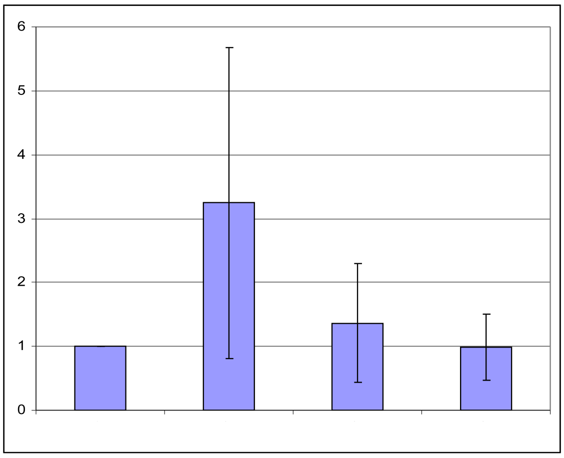
(a) Relative contour drawing times with standard deviation for the four methods.



(b) Relative trial times with standard deviation for the four methods.



(c) The contour drawing time (bottom bar) as a percentage of the total trial time.



(d) Relative accuracy of each method with standard deviation.

Figure 5: *Graphs of the measurement results of all four methods relative to method 1, with standard deviation.*

The tangible controller of method 3 is as fast as the freehand input of method 2, but it is significantly more accurate. This may be the result of a better sense of where on the surface of the object manipulation is taking place, or because the movement of the pen on the cube is initiated from the wrist as opposed to the whole arm with the unrestricted controller, which causes the movement to be more precise. In addition the pen rests on the cube, while method 3 requires constant muscle tension in the arm to keep the pen in a steady position.

The fact that in method 1 the fraction of the trial time spent during the drawing phase is higher, and thus that the initial viewpoint selection took less time, seems to imply that viewpoint selection using a 6-DOF device is somehow slower than viewpoint selection with the mouse. The probable cause for this difference is that in method 1 zooming the camera always occurs relative to a point which was previously selected with the 'f'-key or relative to the center of the object if no point was selected, while for the other methods the cube always has to be actively moved into the desired center location for the zooming to have the desired effect, even if the desired location was the same one as during the previous zoom operation.

6. Conclusion

This study explored the performance differences and user preferences between a number of 2D and 3D input methods for selecting points on a surface using a 3D contour widget. The order of user preference of the methods corresponds to the order of their accuracy: the more preferred methods are more accurate. However, they are not necessarily faster. The pure 3D methods are slower and considered more difficult to use than the mouse.

Using the tangible controller is much more accurate and easier than freehand drawing in 3D, and it provides a replacement for 2D input devices in 3D environments which offer only 3D input, such as the CAVE. However, it is not as fast or precise as a 2D mouse, and it requires a more accurate and better calibrated 6-DOF input system than 3D freehand drawing. In addition the Magic Lens can reduce rendering performance when used in complex scenes, as the scene has to be rendered twice.

6.1. Future Work

In future studies we would like to consider different 3D widgets and tasks. It is likely that for some 3D widgets a different tangible control method is preferable over the Magic Lens, while for others freehand control is actually the optimal method. We would also like to experiment with using more possibilities of the Magic Lens, for example using it to display additional information about the viewed object, or to show a different visualization of the viewed object.

It would also be interesting to investigate the performance of the tangible controller in a different virtual environment. For example, the Magic Lens needs some modifications to make it a suitable and efficient tool for use in an immersive virtual world where the user navigates through a world in which objects are not within arm's reach and where the scene in hand metaphor is unsuitable for camera control.

Acknowledgments

This work was carried out in the context of the Virtual Laboratory for e-Science project (`www.vl-e.nl`). This project is supported by a BSIK grant from the Dutch Ministry of Education, Culture and Science (OC&W) and is part of the ICT innovation program of the Ministry of Economic Affairs (EZ).

7. References

References

[BH06] BROWN L. D., HUA H.: Magic lenses for augmented virtual environments. *IEEE Computer Graphics and Applications 26*, 4 (2006), 64–73.

[BSP*93] BIER E. A., STONE M. C., PIER K., BUXTON W., DEROSE T. D.: Toolglass and magic lenses: the see-through interface. In *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1993), ACM, pp. 73–80.

[CSH*92] CONNER B. D., SNIBBE S. S., HERNDON K. P., ROBBINS D. C., ZELEZNIK R. C., VAN DAM A.: Three-dimensional widgets. In *SI3D '92: Proceedings of the 1992 symposium on Interactive 3D graphics* (New York, NY, USA, 1992), ACM, pp. 183–188.

[Kit03] KITWARE, INC.: *The Visualization Toolkit User's Guide*, January 2003.

[KvL04] KOK A. J. F., VAN LIERE R.: Co-location and tactile feedback for 2d widget manipulation. In *VR '04: Proceedings of the IEEE Virtual Reality 2004 (VR'04)* (Washington, DC, USA, 2004), IEEE Computer Society, p. 233.

[LSH99] LINDEMAN R. W., SIBERT J. L., HAHN J. K.: Hand-held windows: Towards effective 2d interaction in immersive virtual environments. *IEEE VR '99* (1999), 205–212.

[LST01] LINDEMAN R. W., SIBERT J. L., TEMPLEMAN J. N.: The effect of 3d widget representation and simulated surface constraints on interaction in virtual environments. In *VR '01: Proceedings of the Virtual Reality 2001 Conference (VR'01)* (Washington, DC, USA, 2001), IEEE Computer Society, pp. 141–148.

[MFPBS97] MINE M. R., FREDERICK P. BROOKS J., SEQUIN C. H.: Moving objects in space: exploiting proprioception in virtual-environment interaction. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1997), ACM Press/Addison-Wesley Publishing Co., pp. 19–26.

[RMTHS*01] RUSSELL M. TAYLOR I., HUDSON T. C., SEEGER A., WEBER H., JULIANO J., HELSER A. T.: Vrpn: a device-independent, network-transparent vr peripheral system. In *VRST '01: Proceedings of the ACM symposium on Virtual reality software and technology* (New York, NY, USA, 2001), ACM, pp. 55–61.

[SML96] SCHROEDER W. J., MARTIN K. M., LORENSEN W. E.: The design and implementation of an object-oriented toolkit for 3d graphics and visualization. *Visualization '96* (1996), 93–100.

[VCWP96] VIEGA J., CONWAY M. J., WILLIAMS G., PAUSCH R.: 3d magic lenses. In *UIST '96: Proceedings of the 9th annual ACM symposium on User interface software and technology* (New York, NY, USA, 1996), ACM, pp. 51–58.