3D Modeling and Design Supported via Interscopic Interaction Strategies

Frank Steinicke, Timo Ropinski, Gerd Bruder and Klaus Hinrichs

Visualization and Computer Graphics (VisCG) Research Group,
Department of Computer Science, Westfälische Wilhelms-Universität Münster,
Einsteinstr. 62, 48149 Münster, Germany
{fsteini, ropinski, g_brud01, khh}@math.uni-muenster.de

Abstract. 3D modeling applications are widely used in many application domains ranging from CAD to industrial or graphics design. Desktop environments have proven to be a powerful user interface for such tasks. However, the raising complexity of 3D dataset exceeds the possibilities provided by traditional devices or two-dimensional display. Thus, more natural and intuitive interfaces are required. But in order to get the users' acceptance technology-driven solutions that require inconvenient instrumentation, e.g., stereo glasses or tracked gloves, should be avoided. Autostereoscopic display environments in combination with 3D desktop devices enable users to experience virtual environments more immersive without annoying devices. In this paper we introduce interaction strategies with special consideration of the requirements of 3D modelers. We propose an interscopic display environment with implicated user interface strategies that allow displaying and interacting with both mono-, e.g., 2D elements, and stereoscopic content, which is beneficial for the 3D environment, which has to be manipulated. These concepts are discussed with special consideration of the requirements of 3D modeler and designers.

Keywords: HCI, autostereoscopic displays, 3D user interfaces, interscopic interaction techniques, 3D modeling and design

1 Introduction

3D modeling applications are widely used in many application domains ranging from CAD to industrial or graphics design. Desktop systems are used as de facto standard user interface (UI) for such 3D design environments. Since in these domains most data is three-dimensional, many concepts have been proposed in order to improve 3D manipulation when just using ordinary two-dimensional desktop devices [8].

Current 3D user interfaces are technology-driven solutions providing more immersive exploration of and interaction with complex datasets, in particular by using stereoscopic projection and tracked six degrees of freedom (DoFs) input devices.

Although, such virtual reality (VR) systems have great potential to open up new vistas for 3D modeling, these concepts are often not applied in current working environments. The main reason for this fact is that the user usually has to be equipped with inconvenient glasses in order to perceive content stereoscopically [8]. Furthermore, devices that enable control over multiple DoFs simultaneously still involve problems, which are often avoided by the usage of their 2D counterparts; as a matter of fact 2D interactions are performed best with 2D devices [3, 6, 9, 16].

However, while in real life humans are able to move and turn objects in a single motion, this natural interaction is absent in two-dimensional interfaces, where the user is forced to decompose 3D tasks into several 2D tasks. In addition, shortage of spatial input in typical 3D applications leads to the constant need to switch modes. This procedure results in ineffectiveness, in particular when switching between manipulation and exploration techniques is required again and again.

Autostereoscopic (AS) displays can be used to view 3D data stereoscopically without wearing any devices [8]. When using such a setup the user is able to perceive a stereoscopic image in a fixed area called *sweet spot*. When the AS display features a head tracker or when multiple sweet spots are supported, the user can even move in front of the display. Most 3D modeling applications combine 2D UI items, such as menus, texts and images, with 3D content, i.e., 3D virtual environments or objects which have to manipulated. While 3D content usually benefits from stereoscopic display, 2D GUI items often do not require immersive visualization. However, the required separation of stereo half images introduced by AS displays influences viewing of monoscopic content in such a way that the most essential elements of the GUI are distorted. Although some displays to switch of stereoscopic mode, simultaneously display of mono- as well as stereoscopic is not possible. Therefore, an additional regular display was required until now; alternatively expensive hardware technologies can be used [17]. However, only few applications support rendering of a stereoscopic window on a different display. Nevertheless, problems arise from decoupling interaction and visualization. Hence interactions between mono- and stereoscopic elements, which we refer to as *interscopic interactions*, have not been investigated with special consideration of the interrelations between both worlds as they occur in 3D modeling and design.

In this paper we present interscopic interaction strategies with special consideration of the requirements of 3D modelers. Appliance of the proposed interscopic UI environment enables users to perceive and interact with arbitrary regions either mono- or stereoscopically. The remainder of this paper is structured as follows. Section 2 resumes related work. In Section 3 we present the interscopic display environment, implicated interaction strategies are discussed in Section 4. In Section 5 we present the results of a user study with respect to 3D modeling. Section 6 concludes the paper.

2 Previous Work

In 2000, the Heinrich-Hertz-Institute built an AS display system consisting of a gaze tracker, a head tracker and a hand tracker [11]. The head tracker gives the user a look-around capability, while the gaze tracking activates different applications on the desktop. Hand tracking systems enable users to navigate and manipulate objects in 3D space via simple gestures [11, 21]. However, these approaches rather address tracking technologies than advanced 3D user interfaces. Although, current stereo-in-a-window [11, 23] systems show stereoscopic content either in one window time-sequentially or using filtering techniques, these technologies are restricted to only one rectangular window and glasses are still required. Hardware-based approaches have been proposed to display mono- and stereoscopic content simultaneously on one AS display [17]. However, interaction concepts have not yet been developed for these displays and these systems only exist as prototype solution.

In recent years, many frameworks have been proposed which extend 2D GUIs for operating systems (OSs) to so called *3D desktops* [13, 14, 20]. These approaches provide a virtual 3D space in which 2D GUI elements are replaced by three-dimensional representations in order to provide more space. Although these environments provide a fancy visualization, it has not been investigated in how far they improve the interaction process, since they force the user to perform 3D interactions where 2D interactions are intended. Due to the shortcomings of VR interfaces, hybrid approaches have been proposed which combine 2D and 3D interaction using different display or interaction technologies [4]. However, an instrumentation of the user is still required. Desktop-based VR devices, such as 3D trackball or space mouse allow to prompt also 3D input, while requiring an appropriate level of instrumentation increasing the user's acceptance – if not needed these devices can be laid down and they can be combined with traditional desktop devices, i.e., mouse and keyboard. Indeed, when interacting bi-manually several factors have to be considered. With respect to the tasks, the hands may be moved symmetrically or asymmetrically, some tasks can be performed better with the dominant, others with the non-dominant hand. Also the used input devices have a major impact on how bi-manual interactions are performed. These approaches are applied in everyday tasks and their corresponding concepts are used in most UIs.

However, the combination of traditional devices and 3D user interfaces in AS display environments that run ordinary 3D applications has not been considered until now. The aim of this paper is not to debate the validity of desktop-based interaction concepts – there is no need to through away 40 years of 2D UI research – neither the benefits of technology-driven UI approaches. The objective is to explore in how far these concepts can mutually adapt to each other in order to provide efficient interfaces that will be accepted by users as setups for their typical modeling tasks.

3 Interscopic User Interface for AS Display Environments

In this section we propose the UI, which we believe, has the potential to be accepted for 3D modeling and design since although natural and immersive interactions are supported, instrumentation of the user is avoided.

On current AS displays users can see 3D data without wearing any instruments, for example by using lenticular rasters [8]. The raster operates as a beam splitter and ensures that the pixels displayed in each odd column are seen by the user's left eye,

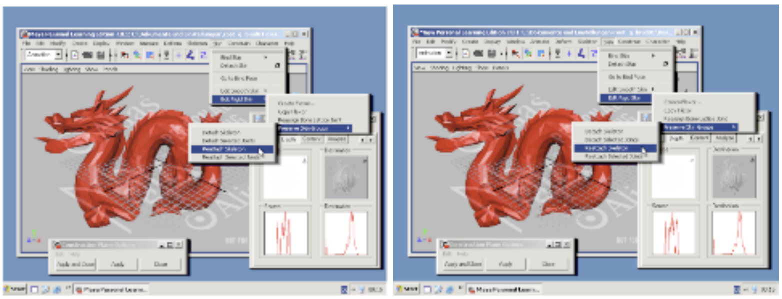

Fig. 1: Interscopic user interface showing a Maya scene with mono- as well as stereoscopic content for the (left) left and (right) right eye.

while the pixels displayed in each even column are perceived with the right eye. The separation of the stereo half images influences viewing of monoscopic content in such a way that the most essential elements of the GUI are distorted. To dissolve this effect we have implemented an *interscopic software framework*, which provides full control about the graphical user interface (GUI) of the operating system (OS). Thus, any region or object can be displayed either mono- or stereoscopically. Furthermore, we are able to catch the entire content of any 3D graphics application based on OpenGL. Our framework allows modifying the corresponding function calls such that visualization can be changed arbitrarily, e.g., stereoscopic views can be generated for scenes specified in applications that in fact do not support stereoscopic display.

To allow simultaneous viewing of mono- as well as stereoscopic content we need to modify the monoscopic data in order to make it perceivable on AS displays and we need to generate a stereo pair out of the 3D content. When viewing unadapted 2D content on vertical-interlaced AS displays viewing two separated images leads to an awkward viewing experience. To make this content perceivable we have to ensure that the left and the right eye perceive almost the same information, resulting in a flat two-dimensional image embedded in the image plane. To achieve this effect with vertical-interlaced AS displays an image of the 2D content rendered into a virtual display has to be scaled in order to ensure that on both the odd and the even columns almost the same information is displayed. In order to have a virtual desktop at our disposal we had to develop an appropriate display driver. Since only a few 3D applications natively support stereoscopic viewing on certain AS displays, in most cases we have to adapt also the 3D content in order to generate stereocopic images. Therefore, we trace and cache all 3D function calls of an OpenGL-based 3D application and execute them twice, once for each eye as described above.

To separate the 2D and the 3D content, we have to know which window areas are used for stereoscopic display. This can be either determined manually or automatically. Manual selection is beneficial, for example, when having a 3D design application providing multiple 3D views showing an object from different views. In this case one may only want to have the perspective 3D preview stereoscopic but not the orthographic side views. When merging 2D and 3D content an obvious problem may arise, when 2D and 3D content areas overlap each other. In this case the separation cannot be performed on the previous 3D window selection process only. To properly render the mouse cursor, context menus and pop-up windows, which may

appear on top of the 3D canvas we apply a masking technique. This is for example important, when dealing with 3D modeling applications, whereas context menus provide convenient access to important features. When merging 2D and 3D content the mask ensures that only those areas of the 3D window are written to the display, which are not occluded by 2D objects. Figure 1 shows a resulting screenshot of the interscopic user interface with mono- as well as stereoscopic content for the left and for the right eye.
The interaction performed in our setup is primarily based on mouse and keyboard (see Figure 2). However, we extend these devices with 3D user interfaces, e.g., 3D trackball or space mouse. Such 3D input devices provide intuitive concepts to input 3D information. However, these devices often lack 2D control, whereas their 2D counterparts are beneficial. Hence, we support also ordinary desktop devices that can be used in conjunction with 3D interaction with 3D devices.

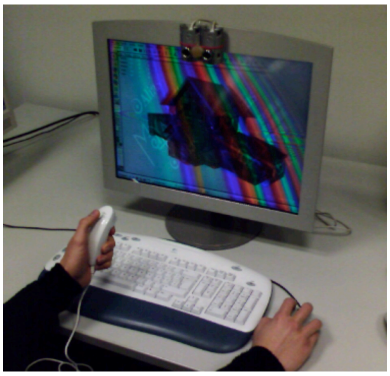

Fig 2. The interscopic display environment includes an AS display, traditional mouse and keyboard. Moreover, the modeler can use 3D devices in order to modify three-dimensional content of a 3D scene. Due to the lenticular raster the user perceives a stereoscopic image.

4. Interscopic Interaction Strategies

When modeling a 3D scene, designers are provided with several tools that enable comfortable and rapid interactions. Since UIs for 3D modeling applications are adapted to the needs of designers, we maintain these concepts and focus on improving the interaction with respect to the opportunities provided by the described interscopic UI for AS display environments.

4.1 3D Surface Cursor

Using well-established 2D input paradigms for interacting with 3D stereoscopic content involves problems in terms of inconsistent depth perception. For example,

when using stereoscopic display in ordinary desktop systems, the mouse cursor appears as two-dimensional object over the 3D objects, which disturbs the stereoscopic impression. Hence, we provide a 3D surface cursor that remains aligned on the surface of the 3D objects when the user moves the mouse cursor. Thus, the position of the mouse cursor gives an additional cue about the depth of 3D objects.
An alternative is to display the cursor always at the image plane. In contrast to ordinary desktop environments the mouse cursor gets invisible when it is obscured by another object extending out of the screen. Thus the stereoscopic impression is not disturbed by the mouse cursor, indeed the cursor is hidden during that time.

4.2 Monoscopic Lens

Many 2D as well as 3D applications provide interaction concepts which are best applicable in two dimensions using 2D interaction paradigms. One example, are 3D widgets [] that reduce the degrees of freedom manipulated simultaneously. These concepts are frequently used in 3D modeling applications. We propose a monoscopic lens, through which 2D interactions can be performed without loosing the entire stereoscopic effect. Within an arbitrary lens shape surrounding the mouse cursor the content appears monoscopically. Thus the user can use familiar tools to perform 2D as well as 3D interactions. Figure 3 shows a monoscopic lens used to control a 3D translation widget. The content within the spherically shaped lens centered around the mouse cursor is rendered monoscopically, whereas the residual part of the 3D model is rendered stereoscopically.

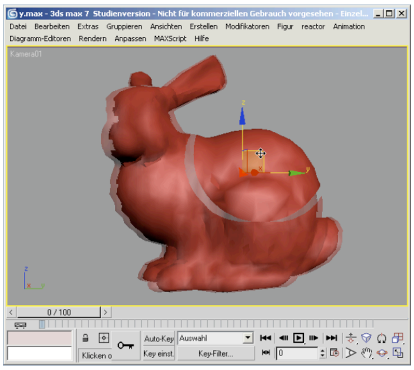
y.max - 3ds max 7 Studienversion - Nicht für kommerziellen Gebrauch vorgesehen - Einzel...
Datei Bearbeiten Extras Gruppieren Ansichten Erstellen Modifikatoren Figur reactor Animation
Diagramm-Editoren Rendern Anpassen MAXScript Hilfe
Kamera01
z
y
x
z
x
y
0 / 100
Auto-Key
Auswahl
Key einst.
Key-Filter...
Klicken o
0

Fig 3: Monoscopic lens applied to a 3D translation widget used in 3D Studio Max modeling application. Content within the lens is rendered monosopically, whereas the rest is displayed using stereoscopic display.

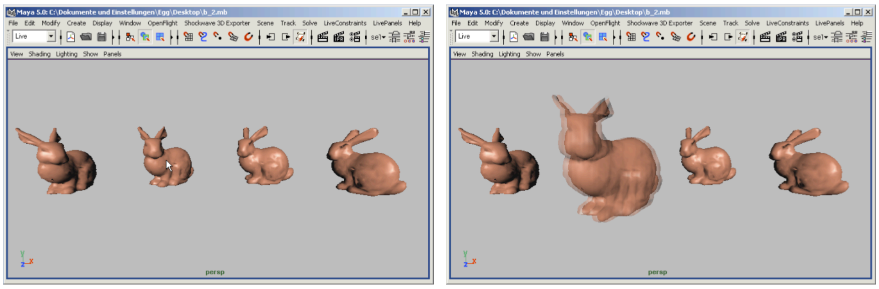
Maya 5.0: C:\Dokumente und Einstellungen\Egg\Desktop\b_2.mb
File Edit Modify Create Display Window OpenFlight Shockwave 3D Exporter Scene Track Solve LiveConstraints LivePanels Help
Live
View Shading Lighting Show Panels
persp
Maya 5.0: C:\Dokumente und Einstellungen\Egg\Desktop\b_2.mb
File Edit Modify Create Display Window OpenFlight Shockwave 3D Exporter Scene Track Solve LiveConstraints LivePanels Help
Live
View Shading Lighting Show Panels
persp

Fig. 4: 3D Maya scene with (a) four objects rendered monosopically and (b) after selecting one object this object is rendered stereoscopically.

4.3 Virtual Stereo Showcase

Our framework enables us to hijack display lists, e.g., used in 3D modeling applications based on OpenGL. This opportunity can be exploited to extract 3D objects out of arbitrary OpenGL applications and to paste them into a 3D modeler in order to manipulated them. Moreover, the user can select individual objects within a 3D modeling scene to display them stereoscopically as illustrated in Figure 4. Furthermore, we provide the concepts of a *stereo showcase* in which 3D content based on display lists can be imported by using the drag-and-drop metaphor. Hence the user can pick a 3D object from an arbitrary application and drag it into a showcase that is represented as a bounding box. Within this showcase the user can apply well-known interaction techniques to explore the 3D objects. These techniques are the same no matter from where the 3D objects have been imported.

5 Evaluation

In informal tests, most users have evaluated the usage of stereoscopic display for 3D modeling applications as very beneficial. In particular, two modeling professionals revealed stereoscopic visualization for 3D content in their 3D modeling environments, i.e., Maya and Cinema4D, as very helpful. However, in order to evaluate the described interscopic user interface we have performed a preliminary usability study. We have used the experimental environment described in Section 3.

5.1 Experimental Task

We restricted the evaluation tasks to simple interactions in which four users had to delete several doors and windows from a virtual building using Maya (see Figure 2). The building consisted of 290 triangles, where windows and doors included 20 triangles uniformly separated. We have performed three series. In the first series the user could use all provided input paradigms, i.e., mouse, keyboard, and space mouse, in combination with stereoscopic display. In this series we have also performed subseries, where the space mouse has been constrained to 3, 2 and 1 supported DoF. In the second series, only the mouse and keyboard could be used. In the last series, only monoscopic visualization in combination with the traditional devices was supported. We have measured the required time for the entire task and we have measured how long each device has been used.

5.2 Results

Table 1 shows the average portion of the usage of 3D and traditional devices during the task. The results indicate that the less DoF are available the less the 3D input is used. This is due to the fact *constraint-based* interaction supports the user when arranging virtual objects for what 3D input has been mainly used. When using constrained rotation, for example, the objects could only be rotated along one axis or two axes in space. Thus simpler and therefore fewer manipulations of virtual objects are required in order to define their orientations.

Table 1. Usage of 3D input device in comparison to traditional input devices using (left) 3 DoF, (middle) 2 DoF and (right) 1 DoF during the entire interaction process.

DoF	3D devices	Traditional devices
3	28%	72%
2	17%	83%
1	16%	84%

As pointed out in Table 2 using 3D input in modeling applications based on our interscopic user interface enhances the performance, in particular when the 3D manipulation is constrained appropriately. However, the usage of monoscopic display strikes over the usage of stereoscopic display if the same devices are supported. This is not unexpected since the tasks have not involved any advanced design or exploration requirements; they were restricted to simple manipulations where stereoscopic display was not really essential.

Table 2. Required time for a simple interaction task with stereoscopic display and 3D user interface supporting 3, 2 and 1 DoF, as well as stereoscopic resp. monoscopic display using only mouse and keyboard.

Input paradigms	Required time
Stereoscopic display, 3 DoF space mouse	58,2 sec
Stereoscopic display, 2 DoF space mouse	57,6 sec
Stereoscopic display, 1 DoF space mouse	45,2 sec
Stereoscopic display, mouse, keyboard	72,0 sec
Monoscopic display, mouse, keybpoard	57,8 sec

6 Conclusion

In this paper we have addressed an interscopic user interface for autostereoscopic display environments and we have presented implicated interaction concepts. We have demonstrated the benefits with respect to 3D modeling and design applications. The proposed techniques have proven the capability to increase the interaction performance in such setups. Moreover, these strategies have the potential to be accepted by users as new user interface paradigm for specific modeling tasks as well as for standard desktop-based, in particular, when simultaneous interaction with monoscopic and stereoscopic content is intended.

A preliminary evaluation of these concepts has proven the advantages of the described approaches. Users and participants have remarked that working in such an environment open up new vistas and has the potential to enhance the quality and efficiency of 3D modeling tasks.

In the future we will integrate further functionality and visual enhancements using more stereoscopic and optionally physics-based motion effects. Moreover, we plan to examine further interaction techniques, in particular, for domain-specific interaction tasks.

References

1. A. Agarawala and R. Balakrishnan. Keepin' It Real: Pushing the Desktop Metaphor with Physics, Piles and the Pen. In Proceedings of the SIGCHI conference on Human Factors in computing systems, pages 1283–1292, 2006.

2. Z. Y. Alpaslan and A. A. Sawchuk. Three-Dimensional Interaction with Autostereoscopic Displays. In A. J. Woods, J. O. Merritt, S. A. Benton, and M. R. Bolas, editors, Proceedings of SPIE, Stereoscopic Displays and Virtual Reality Systems, volume 5291, pages 227–236, 2004.
3. R. Balakrishnan. A Grant of 3D. Keynote speach Symposium on 3D User Interfaces, 2006.
4. H. Benko, E. W. Ishak, and S. Feiner. Cross-Dimensional Gestural Interaction Techniques for Hybrid Immersive Environments. In Proceedings of the Virtual Reality, pages 209–216. IEEE, 2005.
5. P. Bourke. Autostereoscopic Lenticular Images (http://local.wasp.uwa.edu.au/~pbourke/), 2006
6. D. Bowman, E. Kruijff, J. LaViola, and I. Poupyrev. 3D User Interfaces: Theory and Practice. Addison-Wesley, 2004.
7. D. B. Conner, S. C. Snibbe, K. P. Herndon, D. C. Robbins, R. C. Zeleznik, and A. van Dam. Three-Dimensional Widgets. In Symposium on Interactive 3D Graphics, 1992.
8. N. A. Dodgson. Autostereoscopic 3D Displays. In Computer, volume 38, pages 31–36, 2005.
9. A. J. Hanson and E. Wernert. Constrained 3D Navigation with 2D Controllers. In Proceedings of Visualization '97, pages 175–182. IEEE Computer Society Press, 1997.
10. J. Liu, S. Pastoor, K. Seifert, and J. Hurtienne. Three Dimensional PC toward novel Forms of Human-Computer Interaction. In Three-Dimensional Video and Display Devices and Systems SPIE, 2000.
11. J. D. Mulder and R. van Liere. Enhancing Fish Tank VR. In Proceedings of Virtual Reality, pages 91–98. IEEE, 2000.
12. Philips 42-3D6C01. (http://www.inition.co.uk).
13. Project Looking Glass. (http://www.sun.com/software).
14. G. Robertson, M. van Dantzich, D. Robbins, M. Czerwinski, K. Hinckley, K. Risden, D. Thiel, and V. Gorokhovsky. The Task Gallery: A 3D Window Manager. In Proceedings of the SIGCHI conference on Human factors in computing systems, pages 494–501, 2000.
15. T. Ropinski, F. Steinicke, G. Bruder, and K. Hinrichs. Simultaneously Viewing Monoscopic and Stereoscopic Content on Vertical-Interlaced Autostereoscopic Displays. In Poster-Proceedings of the 33rd International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH06), 2006.
16. T. Salzman, S. Stachniak, and W. Stürzlinger. Unconstrained vs. Constrained 3D Scene Manipulation. In 8th IFIP International Conference on Engineering for Human Computer Interaction, volume 2254, 2001.
17. SeeReal Technologies. Next Generation Display Technology (http://www.seereal.com/en/products/nextgen/nextgen.pdf).
18. B. Shneiderman. Designing the User Interface: Strategies for Effective Human-Computer Interaction. Addison-Wesley, 1997.
19. G. Smith, T. Salzman, and W. Stürzlinger. 3D Scene Manipulation with Constraints. In B. Fisher, K. Dawson-Howe, and C. O'Sullivan, editors, Virtual and Augmented Architecture, pages 35–46, 2001.
20. SphereXP. (http://www.spheresite.com).
21. D. Stotts, J. C. Smith, and K. Gyllstrom FaceSpace: Endo- and Exo-Spatial Hypermedia in the Transparent Video Facetop Proceedings of the fifteenth ACM conference on Hypertext and hypermedia, pages 48 - 57, 2004.
22. C. van Berkel. Touchless Display Interaction. In SID 02 DIGEST, 2002.