Eurographics Symposium on Virtual Environments (2006)
Roger Hubbold and Ming Lin (Editors)

Friction surfaces: scaled ray-casting manipulation for interacting with 2D GUIs

C. Andujar and F. Argelaguet

Universitat Politècnica de Catalunya, Barcelona, Spain

Abstract

The accommodation of conventional 2D GUIs with Virtual Environments (VEs) can greatly enhance the possibilities of many VE applications. In this paper we present a variation of the well-known ray-casting technique for fast and accurate selection of 2D widgets over a virtual window immersed into a 3D world. The main idea is to provide a new interaction mode where hand rotations are scaled down so that the ray is constrained to intersect the active virtual window. This is accomplished by changing the control-display ratio between the orientation of the user's hand and the ray used for selection. Our technique uses a curved representation of the ray providing visual feedback of the orientation of both the input device and the selection ray. The users' feeling is that they control a flexible ray that gets curved as it moves over a virtual friction surface defined by the 2D window. We have implemented this technique and evaluated its effectiveness in terms of accuracy and performance. Our experiments on a four-sided CAVE indicate that the proposed technique can increase the speed and accuracy of component selection in 2D GUIs immersed into 3D worlds.

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Methodology and Techniques - Interaction Techniques

1. Introduction

In the recent years a considerable amount of research has been devoted to develop techniques for making 2D applications available from within VEs. As a consequence, a number of tools for launching and/or sharing existing 2D applications into VE and Augmented Reality (AR) applications have been proposed. *Hardware oriented* approaches provide access to external applications through additional display devices such as PDAs and see-through-displays [WDC99]. *Software oriented* approaches [BBH03, DNH03, Elm03, AFA06] access 2D display contents generated by external applications and display them as texture-mapped rectangles. These techniques let the users interact with 2D applications through VR input devices such as gloves and 6-DOF sensors (see Figure 1). VNC (Virtual Network Computing) [RSFWH98] is a remote display system which allows viewing a computing desktop running elsewhere on a network. VNC provides a distribution mechanism for desktops by transmitting frame buffer contents to the remote client and receiving keyboard and pointing device events. VNC is the foundation of most systems providing immersion of 2D applications into 3D space [BBH03, DNH03, Elm03].

At a low level, user-interaction tasks with 2D GUIs immersed into 3D worlds can generally be characterized as selection or manipulation tasks [BH99]. However, selection and manipulation of 2D GUIs has some specific characteristics which must be considered when designing appropriate interaction techniques:

- Interaction with 2D GUIs is often dominated by selection, and components designed for direct manipulation provide much simpler and constrained motion. Placement and rotation in 2D GUIs is mostly 1 DOF (e.g. moving a slider or rotating a dial).
- Application-control GUIs (such as those enabling the adjustment of display parameters) are often manipulated frequently but in short intervals, thus making inappropriate the use of techniques that require some kind of user set-up (e.g. the Voodoo Dolls technique [PSP99]) or have a large impact on the displayed image (e.g. the Scaled-World Grab technique [MFBS97]).

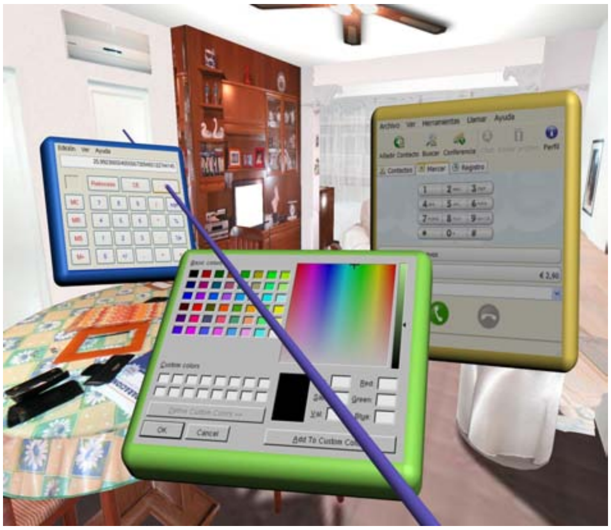


Figure 1: *External 2D applications (color selector, calculator and P2P telephony) immersed into a 3D world through a modified VNC client.*

An additional problem involved in accessing external applications from within VEs is that the only possible adaptation of the GUI to the VE is to adjust the size and location of the virtual window containing the representation of the different controls (*widgets*). For example, a window might include small, nearby buttons which can be difficult to select using 3D interaction (see e.g. the color selector in Figure 1). Therefore, interaction techniques for such applications could differ from those used for GUIs specifically tailored to 3D environments.

A fundamental technique for manipulating 3D objects is *pointing*. Pointing techniques enable the user to select and manipulate objects by simply pointing at them. A number of studies have demonstrated that pointing techniques often result on better selection effectiveness than *virtual hand* techniques [BKLP04]. Different variations of this technique differ basically on three aspects: the computation of the *pointing direction* (i.e. the mapping of the input device position and orientation onto the direction of the ray), the shape of the selection volume, and its visual representation (feedback). Ray-casting is the simplest and most popular pointing technique. In classic ray-casting implementations, the pointing direction is given directly (isomorphically) by a virtual ray controlled by a 6-DOF sensor and visual feedback is provided by drawing a line extending out from the user's hand. Unfortunately, ray-casting techniques do not perform well when selecting small or distant objects [PWBI98]. Small rotations of the wrist sweep out large arcs at the end of the selection ray. Therefore hand trembling and tracking errors are amplified with increasing distance, thus requiring a high level of angular accuracy. Accurate selection is also compromised by the hand instability caused by the absence of constraints on the hand movements (lack of physical support for manipulation) [LSH99]. As a result, users attempting to select small buttons with this technique have to make a considerable effort to stabilize their wrist.

In this paper we present a new interaction technique for fast and accurate selection of 2D widgets over a virtual window. The main idea is to provide more accuracy to the ray-casting technique by changing the control-display (C-D) ratio [BGBL04] when the user is interacting with the active window. The C-D ratio is a coefficient that maps the physical movement of the pointing device to the resulting on-screen movement in a system where there is an anisomorphism between the pointing device and the display (e.g. a 2D mouse). The C-D ratio often defines the distance that must cover the device in the physical world (dx) to move the cursor on the screen by a given distance (dX). The adaptation of the C-D ratio dx/dX has been successfully used in many interaction techniques (see [FK05] for a review). However, these techniques have been designed for manipulating 3D objects and thus do not address the specific problems involved in 2D GUI manipulation.

Our technique adapts the C-D ratio in order to scale down hand rotations and enable accurate selection and manipulation of distant or small GUI objects. When the user is pointing at a virtual window, we increase the CD-ratio between the user's hand and the ray used for selection, so that the ray rotates more slowly than the user's hand, thus reducing the effect of hand instability. Unlike other techniques designed for accurate 3D object manipulation which define the C-D ratio inversely proportional to the hand speed [FK05], we compute the C-D ratio by considering the size and position of the virtual window relative to the user's hand at the moment the scaled mode is activated. Our technique uses a curved representation of the ray providing integrated visual feedback of both the orientation of the input device and the selection ray. We call this technique *friction surfaces* because the users' feeling in scaled mode is that they control a flexible ray that gets curved as it moves over a virtual friction surface defined by the 2D window (see Figure 2).

We conducted an informal usability evaluation to measure the performance and effectiveness of the technique. Our experiments on a four-sided CAVE indicate that the proposed technique can be used to increase the accuracy of component selection in 2D GUIs immersed into 3D worlds without sacrificing speed.

The main contributions of the paper are:

- A new technique for interacting with 2D applications immersed into VEs. The technique has its roots in ray-casting selection and C-D based techniques [FK05] but adopts a completely different approach for activation/deactivation of the scaled mode and for computing the C-D ratio.

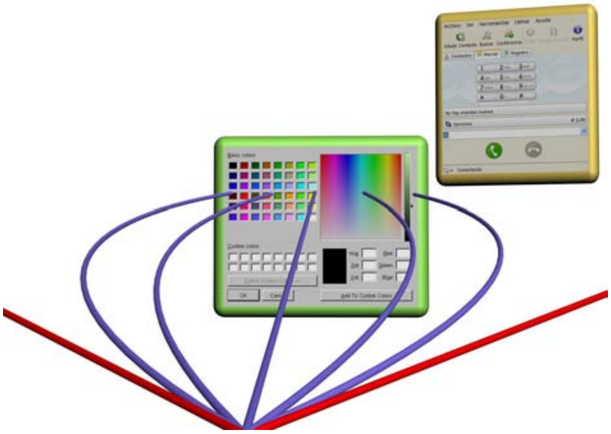


Figure 2: *Visual feedback supporting the proposed technique: several rays corresponding to different orientations of the input device are shown. The device orientation can be identified by the tangent direction at the ray's origin.*

- An evaluation of the performance and usability of the technique on a four-sided CAVE that indicates that it is particularly suitable for interaction with external applications immersed into VEs.

The rest of the paper is organized as follows. Section 2 reviews related work on interaction techniques for accurate selection and/or manipulation. Section 3 describes the proposed interaction technique and discusses the main differences with related approaches. We present effectiveness and usability results in Section 4, and provide concluding remarks in Section 5.

2. Previous Work

A number of 3D interaction techniques have been proposed for manipulating objects in immersive VEs, including *exocentric techniques* [SCP95], *egocentric techniques* such as virtual hand and virtual pointer [PBWI96] and *hybrid techniques* [BH97, MFBS97, PSP99]. The ray-casting technique is a powerful virtual pointer technique for 3D manipulation. A number of studies have demonstrated that ray casting often results on better selection effectiveness than *virtual hand* techniques [BKLP04]. However, classic ray-casting does not perform well when selecting small or distant objects [PWBI98]. Indeed, accurate interaction with small or distant objects is one of the main challenges in 3D manipulation. A number of techniques have been proposed for achieving more accuracy on such tasks.

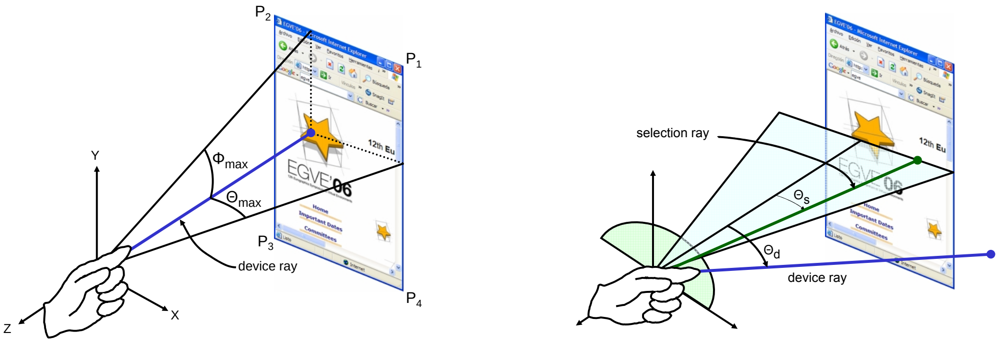
One technique is to explicitly scale or zoom in the workspace in order to provide accurate manipulation. The WIM (World-In-Miniature) [SCP95] is an exocentric technique that provides the user with a miniature handheld model of the VE. Scaled-World grab [MFBS97] provides a manipulation mode where the entire VE around the user's viewpoint is scaled down so that the selected object can be manipulated using the virtual hand technique. HOMER [BH97] uses ray-casting the select an object and then the user's virtual hand instantly moves to the object and attaches to it. The Voodoo Dolls [PSP99] is a two-handed manipulation technique that enables users to scale their workspace by selecting a voodoo doll of appropriate size. All these techniques put the emphasis on 3D object manipulation rather than selection.

The use of physical props to constrain the interaction can help to reduce hand instability [LSH99]. Pen-and-tablet interfaces [IU97] register a virtual window with a physical prop held in the non-dominant hand. Users interact with these handheld windows using either a virtual hand or a virtual pointer held in the dominant hand. Hand-held windows also take advantage of the proprioceptive sense as they are close to the non-dominant hand. Some systems use handheld windows whose physical prop is a lightweight, transparent surface that the user carries around, increasing precision. The Transparent Props [SES99] technique consists of a tracked hand-held pen and a pad. The pad can serve as a palette for tools and controls as well as a window-like see-through interface. Although combining transparent props with other two-handed techniques can be difficult and storing the physical surface when not in use can be an issue, these techniques provide a powerful and flexible interface for manipulating 2D GUIs, provided that the two 6-DOF sensors are available.

A different technique uses static or dynamic adjustment of the C-D ratio. This concept has been applied to many different VE-related tasks including navigation [TRC01], perception [DLB*05], selection [BGBL04] and manipulation [FK05, PWF00]. Blanch *et at*. [BGBL04] improve selection through *semantic pointing*. Semantic pointing is based on defining two independent sizes for each potential target presented to the user. One size is used in *visual space* and it is adapted to the amount of information conveyed by the object. The other size is used in *motor space* and its adapted to the object's importance for the manipulation. This decoupling between visual and motor size is achieved by changing the control-to-display ratio according to cursor distance to nearby targets.

Some techniques amplify rotations so that most manipulations can be accomplished with a single motion, thus minimizing the need for releasing the virtual object and grabbing it again [PBWI96, PWF00], at the expense of some accuracy loss. Usability properties of different rotation mappings are discussed in [PWOI99, PWF00] and different C-D gains between real and virtual hands are evaluated in [BH97].

The PRISM (Precise and Rapid Interaction through Scaled Manipulation) [FK05] uses C-D adjustment to provide fast and accurate manipulation of 3D objects. PRISM uses the hand speed of the user to gradually switch between



(a) (b)

Figure 3: *Elements involved in the computation of the CD-ratio: spherical coordinates used at activation time to fix the CD-ratio (a), and spherical coordinates used for computing the selection ray direction during scaled mode (b).*

different modes by altering the C-D ratio. The mapping is controlled basically by two speed constants defining three intervals. In the first interval, the speed is below a certain minimum velocity and the movement is considered jitter, so the C-D ratio is set to ∞. The next interval corresponds to scaled motion, where C-D ratio is inversely proportional to the hand speed; above a second speed constant (about 20 degrees/second), the C-D ratio is set to 1. PRISM translation operates on each principal axis independently, thus providing an anisotropic mapping. Although similar to PRISM in concept, our technique adopts completely different strategies for activation/deactivation, computing the C-D ratio and providing integrated visual feedback alleviating the anisomorphism of the movement.

3. Our approach

3.1. Overview

The *friction surfaces* technique has been conceived to facilitate the interaction with external 2D applications being accessed from immersive VEs. The main goal is to provide accurate selection and manipulation of 2D GUIs that have not been particularly designed for VEs. To this end, we use a modified ray-casting technique which adapts the C-D ratio according to the size and position of the virtual window.

We start by introducing some notation that will be used in the rest of the paper. The *Device Coordinate System* (DCS) is an orthonormal frame centered at the position of the 6-DOF sensor attached to the user's hand. We assume the DCS is oriented as depicted in Figure 3, with the negative Z axis defining the user's hand pointing direction. This pointing direction will be referred to as the *device ray*. The device ray plays the role of the control component in the C-D ratio. The *Zero orientation* is an orthonormal reference basis used for scaling down the rotation of the user's hand while in scaled mode. The zero orientation is defined from DCS' orientation when the scaled mode is activated (discussed below) and then remains unchanged until the mode is deactivated. We call *selection ray* the ray used for defining the pointing direction. As the name suggests, the intersection of the selection ray with the virtual window's plane is used as the cursor for selection and manipulation purposes. In our approach, both the selection ray and the device ray originate at the current device position, as we only scale down rotations, not translations. The selection ray's direction is the result of applying an isotropic linear mapping to the rotation measured from the zero orientation. The selection ray is the display component in the C-D ratio. Finally, the *feedback ray* is the curved line segment that will displayed for providing visual feedback about the linear mapping (the device and selection rays are not rendered). Note that in the classic ray-casting manipulation, the selection ray coincides with the device ray, thus providing isomorphic manipulation. In our case, isomorphism is preserved only when no virtual window is active.

3.2. Activation

We use two distinct modes: one which scales hand rotations when accuracy is needed (scaled mode) and one which provides direct, isomorphic interaction (normal mode). We now describe different strategies for activating the scaled mode and discuss the associated computations. We have considered both manual and automatic activation. Manual activation/deactivation requires the user to issue a trigger event (e.g. a button press). Automatic activation simply takes place whenever the selection ray enters a virtual window. The mode is set back to normal mode when the selection ray leaves the window; that happens when the hand rotation with respect to the zero orientation reaches a certain maximum value (e.g. 45 degrees). The deactivation causes the selection ray to coincide again with the device ray. We have found that this lost of continuity is not disturbing to the users because it simply causes a curvature change in the bent ray used for feedback (see Section 3.4).

When the scaled mode is activated, the zero orientation is set to the orientation of the current DCS and we compute the range of directions the selection ray can travel before leaving the active window. Let P_i be the i-th vertex of the virtual window (see Figure 3). Let θ_i be the azimuthal angle (longitude) of P_i in the XZ-plane, measured from the negative Z-axis, with $0 \leq \theta_i < 2\pi$. Likewise, let ϕ_i be the zenith angle (latitude) from the XZ-plane, with $-\frac{\pi}{2} \leq \phi_i \leq \frac{\pi}{2}$. These spherical coordinates can be computed using Equations 1 and 2, where (x, y, z) are the P_i coordinates relative to the DCS and where the inverse tangent must be suitably defined to take the correct quadrant of into account:

$$\theta_i = \tan^{-1}\left(\frac{-x}{-z}\right) \tag{1}$$

$$\phi_i = \frac{\pi}{2} - \cos^{-1}\left(\frac{y}{\sqrt{x^2+y^2+z^2}}\right) \tag{2}$$

We can compute the maximum rotation angles of the selection ray in each direction as $\theta_{max} = \max_i\{|\theta_i|\}$ and $\phi_{max} = \max_i\{|\phi_i|\}$. Since we want to use an isotropic scale on both directions, we just use the maximum of these two angles. Therefore, the C-D ratio r is simply

$$r = \frac{\psi}{\max(\theta_{max}, \phi_{max})},$$

where ψ is a constant that defines the range of directions of the input device that approximately map onto a selection ray within the virtual window. In the user studies described in next section we used $\psi = \pi/4$, thus providing the user with a 90 degrees arc for interacting with the active virtual window. If $r < 1$, the window is sufficiently close to the user and thus the scaled mode is not activated.

3.3. Computation of the rotation angles

In scaled mode, the selection ray is computed using the C-D ratio defined at activation. Let θ_d and ϕ_d be the spherical coordinates of an arbitrary point of the device ray (distinct from its origin) with respect to the zero orientation basis. The selection ray is computed from a target point T whose spherical coordinates are computed by scaling down the device rotation, $\theta_s = \theta_d/r$, $\phi_s = \phi_d/r$ (see Figure 3). The coordinates of T with respect to the current *DCS* require a simple conversion back to cartesian coordinates,

$$x = -r\sin(\theta)\cos(\phi)$$

$$y = r\sin(\phi)$$

$$z = -r\cos(\theta)\cos(\phi)$$

where $r > 0$ is an arbitrary value.

3.4. Feedback

We have considered two distinct options for providing visual feedback. The first option consisted in drawing both the device ray and the selection ray, using different visual attributes (such as color and thickness). This option appears to be quite distracting so we have opted for a single bent ray providing feedback of both the device ray and the selection ray. Curved line segments have been proposed for different purposes related with ray-casting selection. IntenSelect [dHKP05] uses dynamic rating of the objects falling inside a conic selection volume to bent the selection ray so that it snaps with the highest ranking object. Flexible pointers are used in [OF03] to point more easily to fully or partially obscured objects using two-handed interaction. We draw the curved line segment using a Bézier spline (see Figure 2). The curve originates at the user's hand and ends at the intersection P of the selection ray with the virtual window's plane. These two points define the first and last control points of the Bézier curve. The second control point is computed on the device ray so that the tangent direction at the origin is that of the device ray. Finally, the third control point is the point on the device ray closest to P. When the selection ray leaves the virtual window, the scaled mode is deactivated and the displayed ray instantly goes straight. The users' feeling is that a flexible ray gets curved as it is moved over a virtual friction surface defined by the window.

3.5. Comparison with previous techniques

As stated below, friction surfaces is similar to the PRISM [FK05] in concept. Both techniques adapt the C-D ratio to provide accurate manipulation of distant objects. We now synthesize the main differences between both approaches:

- In PRISM the CD-ratio depends on the speed of user's movements; in our case, the CD-ratio depends on the size and position of the window with respect to the user, and it remains constant while in scaled mode. Therefore our adjustment of the control-display ratio is static, unlike the dynamic adjustment used in PRISM.
- In PRISM, the CD-ratio is defined between the user's hand and the virtual object being manipulated; in our system the controlled object is the ray used for selection purposes. Moreover, PRISM acts over both position and orientation; we only scale down rotations.
- In PRISM the activation of the scaled mode depends on several speed constants; in our case activation takes place when the ray enters a virtual window.

- Our technique provides integrated feedback of both the device and the selection ray through a curved line segment (PRISM does not provide feedback about the hand orientation).

Besides these aspects, an important difference is that our technique does not force users to slow down their movements to gain precision. Our approach uses a larger range of movements for controlling the ray in a more reduced region.

4. Evaluation

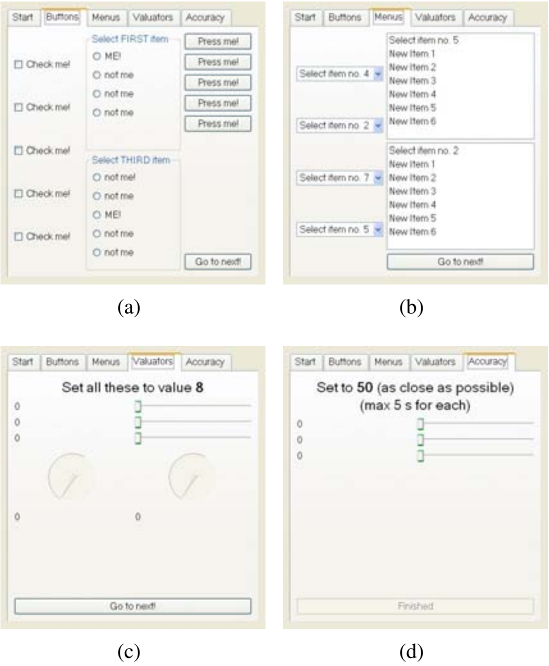
We conducted an informal usability evaluation to measure the effectiveness of the technique compared with classic (isomorphic) ray-tracing. Fourteen users participated in the study. Before each experiment users were provided with a short training session which required them to complete practice trials using both interaction techniques. After this exploratory phase, subjects were ready to begin a more focused task-based evaluation.

4.1. Evaluation test

The evaluation test has been designed to evaluate the task performance in terms of time-to-complete a given task and maximum accuracy achieved in a fixed period of time. All the experiments were conducted on a four-sided CAVE with a 6-DOF wanda. The virtual window used in the experiments was initially placed at 1.5 m from the CAVE center, covering about 20 degrees of the user's field-of-view. The scaled mode was activated automatically each time the device ray entered a virtual window. The zero orientation was forced to have the Z axis coincident with the segment joining the device position and the window's center.

The test dialogs are shown in Figure 4. The first two dialogs are designed to measure task performance on selecting small objects. The first dialog contains different kind of buttons whereas the second dialog includes basically combo boxes and selection lists. The third dialog is designed also to measure speed but putting the emphasis on manipulation rather than on selection. Finally, the fourth dialog is designed to measure the accuracy during object manipulation. In all cases the labels attached to each widget indicate the requested task, so users can be more focused at purely interaction tasks.

For the first three dialogs, users were requested to complete the involved tasks as quickly as possible, using isomorphing ray-casting and scaled ray-casting in a random order. For the fourth dialog, users were asked to manipulate several sliders to get a certain value as accurately as possible, but giving only five seconds of time for each slider, starting from the first click on it. After that time, the slider was disabled and the user was forced to proceed with the next slider.



Set all these to value 8

Set to 50 (as close as possible)
(max 5 s for each)

Figure 4: *The test dialogs used in the evaluation test.*

4.2. Experimental results

The results of these tests are shown in Figures 5 and 6. Figure 5 shows the time spend to complete the tasks involved in the first three dialogs (see Figure 4 a-c). Note that on average the tasks involved in the three dialogs can be completed faster using our technique. The largest difference between the two techniques can be observed by comparing the times for the third dialog. Times for friction surfaces are much better because setting the sliders to a given value required accurate pointing.

Figure 6 shows the results of the accuracy test. The plot shows the average deviation (in slider units) from the target value when the user had only five seconds to adjust it (see Figure 4-d). Each slider had increasing ranges and thus increasing levels of difficulty. Again, our technique provided better results, which were particularly noticeable with increasing levels of accuracy.

4.3. Survey

After using the interface, subjects completed a small survey that asked them to compare both techniques. They rated the ease of selecting the buttons and manipulating the sliders. Questions used a 7-point Likert scale, where 1 mean *near impossible* and 7 mean *as easy as a desktop computer*. We also asked subjects about what they found most difficult and what was the easiest for them. The results were similar from

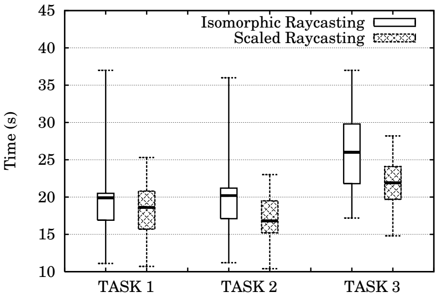


Figure 5: *Time to complete the tasks involved in the first three dialogs.*

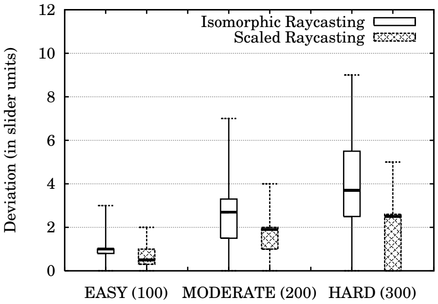
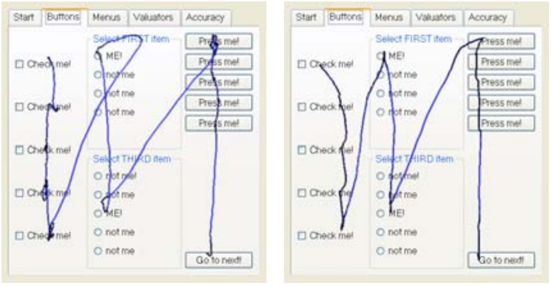


Figure 6: *Deviation from the target value on the accuracy test. The integer value between parentheses is the slider's range.*

all users. All of them agreed on having less problems on selecting buttons and manipulating sliders with our technique and this is mirrored in their average usability rating of 5.6 for friction surfaces compared with a 4.2 for classic ray casting. The most difficult task was to achieve a certain value on the sliders, because the free movement of the wand on their hand makes it difficult to maintain the value when the finger is moved to press or release the button, nicknamed the Heisenberg effect. This problem was noticeably alleviated with our technique. Most users complained about the effort required for selecting small buttons with normal ray-casting because of the considerable effort to stabilize the wrist. Figure 7 shows the path followed by the cursor on a typical exercise. Note that the lack of accuracy of isomorphic ray-casting forced the user to perform many attempts before the right button was selected. On the other hand, a few users pointed out that the scaled mode was a bit unnatural compared with isomorphic raycasting. Nevertheless, their performance was better with the anisomorphic technique.



(a) (b)

Figure 7: *Paths described by the 3D cursor over the virtual window in a typical exercise with isomorphic ray-casting (left) and scaled raycasting (right). Note that check boxes can be toggled by clicking on their label.*

5. Concluding remarks and future work

The accommodation of conventional 2D GUIs with Virtual Environments (VEs) can greatly enhance the possibilities of many VE applications. In this paper we have presented a variation of the well-known ray-casting technique for accurate selection of 2D widgets over a virtual window immersed into a 3D world. An initial evaluation indicate on a four-sided CAVE indicates that the proposed technique can be used to increase the accuracy of component selection in 2D GUIs immersed into 3D worlds without sacrificing speed. An important issue is how this technique compares with PRISM [FK05] and related techniques [Mul05] in terms of task performance. We plan to conduct this evaluation as part of the future work. We also plan to integrate friction surfaces with dynamic rating techniques [dHKP05] so that the displayed ray is further bent to snap with the highest ranking object. This would further improve object selection, with little or no effect on object manipulation.

6. Acknowledgements

The authors would like to thank all the participants who took part in this experiment for their kindness and patience. This work has been funded by the Spanish Ministry of Science and Technology under grant TIN2004-08065-C02-01.

References

[AFA06] ANDUJAR C., FAIREN M., ARGELAGUET F.: A cost-effective approach for developing application-control guis for virtual environments. In *Proc. of 1st IEEE Symposium on 3D User Interfaces, 3DUI'06* (2006).

[BBH03] BUES M., BLACH R., HASELBERGER F.: Sensing surfaces: bringing the desktop into virtual environments. In *Proceedings of the 9th Eurographics Workshop on Virtual environments EGVE'03* (2003), pp. 303–304.

[BGBL04] BLANCH R., GUIARD Y., BEAUDOUIN-LAFON M.: Semantic pointing: improving target acquisition with control-display ratio adaptation. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2004), pp. 519–526.

[BH97] BOWMAN D., HODGES L.: An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In *I3D '97: Proceedings of the 1997 ACM Symposium on Interactive 3D graphics* (1997), pp. 35–38.

[BH99] BOWMAN D. A., HODGES L. F.: Formalizing the design, evaluation, and application of interaction techniques for immersive virtual environments. *Journal of Visual Languages and Computing 10*, 1 (1999), 37–53.

[BKLP04] BOWMAN D. A., KRUIJFF E., LAVIOLA J. J., POUPYREV I.: *3D User Interfaces : Theory and Practice*. Addison Wesley, 2004. ISBN: 0-2017-5867-9.

[dHKP05] DE HAAN G., KOUTEK M., POST F.: IntenSelect: Using Dynamic Object Rating for Assisting 3D Object Selection. In *Proceedings of the 9th IPT and 11th Eurographics VE Workshop (EGVE) '05* (2005), Kjems E., Blach R., (Eds.), pp. 201–209.

[DLB*05] DOMINJON L., LECUYER A., BURKHARDT J., RICHARD P., RICHIR S.: Influence of control/display ratio on the perception of mass of manipulated objects in virtual environments. In *Proceedings of IEEE International Conference on Virtual Reality (IEEE VR)* (2005).

[DNH03] DIVERDI S., NURMI D., HOLLERER T.: A framework for generic inter-application interaction for 3D AR environments. In *Augmented Reality Toolkit Workshop, 2003* (2003), IEEE Computer Society, pp. 86 – 93.

[Elm03] ELMQVIST N.: *3Dwm: A Platform for Research and Development of Three-Dimensional User Interfaces*. Tech. Rep. CS:2003-04, Chalmers Department of Computing Science, 2003.

[FK05] FREES S., KESSLER D.: Precise and rapid interaction through scaled manipulation in immersive virtual environments. In *Proceedings of IEEE Virtual Reality 2005* (2005), pp. 99–106.

[IU97] ISHII H., ULLMER B.: Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (1997), pp. 234–241.

[LSH99] LINDEMAN R., SIBERT J., HAHN J.: Towards usable VR: an empirical study of user interfaces for immersive virtual environments. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (1999), pp. 64–71.

[MFBS97] MINE M., FREDERICK BROOKS J., SEQUIN C.: Moving objects in space: exploiting proprioception in virtual-environment interaction. In *Proceedings of SIGGRAPH'97* (1997), pp. 19–26.

[Mul05] MULDER J.: Menu selection in desktop virtual reality. In *Proceedings of the 2005 Central European Multimedia and Virtual Reality Conference* (2005).

[OF03] OLWAL A., FEINER S.: The flexible pointer—an interaction technique for selection in augmented and virtual reality. In *Conference Supplement of ACM Symposium on User Interface Software and Technology (UIST '03)* (2003), pp. 81–82.

[PBWI96] POUPYREV I., BILLINGHURST M., WEGHORST S., ICHIKAWA T.: The go-go interaction technique: Non-linear mapping for direct manipulation in VR. In *ACM Symposium on User Interface Software and Technology* (1996), pp. 79–80.

[PSP99] PIERCE J. S., STEARNS B. C., PAUSCH R.: Voodoo dolls: seamless interaction at multiple scales in virtual environments. In *Proceedings of the 1999 Symposium on Interactive 3D graphics* (1999), pp. 141–145.

[PWBI98] POUPYREV I., WEGHORST S., BILLINGHURST M., ICHIKAWA T.: Egocentric object manipulation in virtual environments: Evaluation of interaction techniques. *Computers Graphics Forum 17*, 3 (1998), 41–52.

[PWF00] POUPYREV I., WEGHORST S., FELS S.: Non-isomorphic 3D rotational techniques. In *Proceedings of the 2000 ACM conference on Human factors in computing systems (CHI 2000)* (2000), pp. 540–547.

[PWOI99] POUPYREV I., WEGHORST S., OTSUKA T., ICHIKAWA T.: Amplifying spatial rotations in 3d interfaces. In *ACM Conference on Human Factors in Computing Systems, CHI'99* (1999), pp. 256–257.

[RSFWH98] RICHARDSON T., STAFFORD-FRASER Q., WOOD K. R., HOPPER A.: Virtual network computing. *IEEE Internet Computing 2*, 1 (1998), 33–38.

[SCP95] STOAKLEY R., CONWAY M., PAUSCH R.: Virtual reality on a WIM: Interactive worlds in miniature. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (1995), pp. 265–272.

[SES99] SCHMALSTIEG D., ENCARNACAO M., SZALAVRI Z.: Using transparent props for interaction with the virtual table. In *Proceedings of the 1999 symposium on Interactive 3D graphics* (1999), pp. 147–153.

[TRC01] TAN D., ROBERTSON G., CZERWINSKI M.: Exploring 3d navigation: combining speed-coupled flying with orbiting. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2001), pp. 418–425.

[WDC99] WATSEN K., DARKEN R., CAPPS M.: A handheld computer as an interaction device to a virtual environment. In *Proceedings of the International Projection Technologies Workshop* (1999), pp. 303–304.