Pop-up Depth Views for Improving 3D Target Acquisition

Guangyu Wang*
Centre for Intelligent Machines
McGill University

Michael J. McGuffin†
Dept. Software and IT Engineering
École de technologie supérieure

François Bérard‡
Laboratoire d'Informatique de Grenoble
University of Grenoble

Jeremy R. Cooperstock§
Centre for Intelligent Machines
McGill University

Abstract

We present the design and experimental evaluation of *pop-up depth views*, a novel interaction technique for aiding in the placement or positioning of a 3D cursor or object. Previous work found that in a 3D placement task, a 2D mouse used with multiple orthographic views outperformed a 3D input device used with a perspective view with stereo. This was the case, even though the mouse required two clicks to complete the task instead of only the single click required with the 3D input device. We improve performance with 3D input devices with pop-up depth views, small inset views in a perspective display of the scene. These provide top- and side-views of the immediate 3D neighborhood of the cursor, thereby allowing the user to see more easily along the depth dimension, improving the user's effective depth acuity. In turn, positioning with the 3D input device is also improved. Furthermore, because the depth views are displayed near the 3D cursor, only tiny eye movements are required for the user to perceive the 3D cursor's depth with respect to nearby objects. *Pop-up* depth views are a kind of depth view, only displayed when the user's cursor slows down. In this manner, they do not occlude the 3D scene when the user is moving quickly. Our experimental evaluation shows that the combination of a 3D input device used with a perspective view, stereo projection, and pop-up depth views, outperforms a 2D mouse in a 3D target acquisition task, in terms of both movement time and throughput, but at the cost of a slightly higher error rate.

Keywords: 3D target acquisition, 3D input device, popup view, pointing facilitation, Fitts' law, empirical evaluation, interaction design

Index Terms: H.5.2 [Information Interfaces and Presentation]: User Interfaces—Graphical user interfaces (GUI); H.5.2 [Information Interfaces and Presentation]: User Interfaces—Input devices and strategies, prototyping;

1 Introduction

After more than two decades of research on interacting with 3D virtual spaces using 3-or-more degree-of-freedom (DoF) input [31], the status quo input device in this area remains the largely unevolved 2D mouse. A variety of other input devices are employed under certain conditions, e.g., a wand or data glove in immersive VR environments or for mobile interaction with large displays. Nevertheless, the mouse remains the dominant input device for practitioners involved in 3D tasks for extended periods, e.g., 3D modelers and animators, CAD or game designers, architects, and medical data analysts. Whether this is due to some inherent superiority of the mouse for 3D tasks, or simply to user familiarity and the inconvenience of switching input devices [2], remains the source of some speculation.

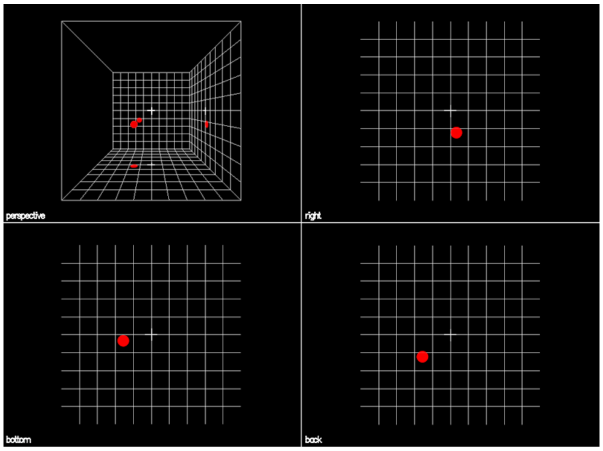
In the physical world, humans routinely demonstrate their capacity to position objects in space in a coordinated manner, controlling three (or more) DoF simultaneously. One might expect, then, that being forced to use a 2D mouse to interact with a virtual 3D world would handicap the user. In practice, however, the 2 DoF of the mouse are sufficient for many tasks: ray-casting can be used to select discrete objects [32] and to position objects on a ground plane or on the face of another object. Ray-casting selection can also be improved by exploiting the empty space between discrete targets, e.g., with a bubble cursor [13] or semantic pointing [8].

However, there are cases in 3D virtual worlds where the user must perform truly unconstrained 3D placement tasks. One example is if the user is positioning a 3D object within a 3D scene to "float" in midair. Another is selecting a 3D location within volumetric medical data, perhaps to specify a point along the boundary of a tumor or tissue. In such cases, a mouse cannot enable simultaneous manipulation of all 3 DoF, and techniques such as snapping, the bubble cursor, or semantic pointing may be of no help, because only the user may be able to judge when they have pointed with sufficient precision at the target they have in mind. In such unconstrained 3D placement, a 2D mouse requires that the task be decoupled into at least two separate 2D sub-tasks. A reasonable assumption would be that input devices affording three or more DoF would support a more natural transfer of our manual manipulation skills from the physical world to the virtual world.

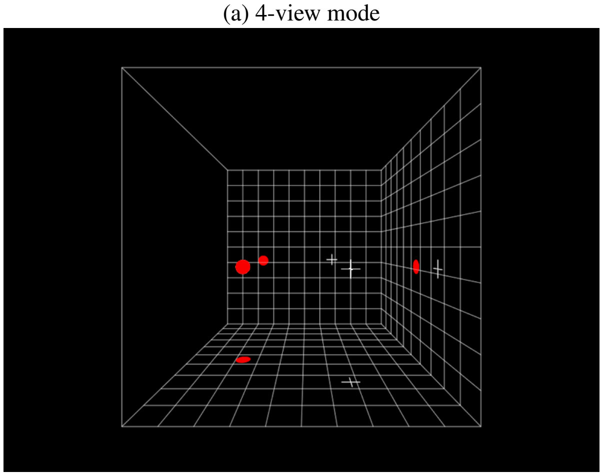
Somewhat surprisingly, a recent study found evidence of the opposite: in a 3D placement task, the mouse, combined with orthographic views (similar to Figure 1(a)), was more efficient and less error prone than 3D devices [3], regardless of whether orthographic or perspective (similar to Figure 1(b)) views were used with the 3D devices. This was despite the fact that the 2D input device required 2 clicks to complete the task versus the 1 click necessary with the 3D input devices, and despite the use of stereo projection with the perspective views with 3D input devices. We suspect this previous result may have been due to the mouse benefiting from a high precision and stability provided by the surface on which it was manipulated, and that three orthographic projections used with the mouse provided a better visualization of the depth dimension than does a stereoscopic perspective display. In other words, the top- and side-views included in the orthographic views provide depth maps, making it easier to position the cursor along the depth dimension.

We were unconvinced, however, that a 2D input device is necessarily the best mechanism for 3D placement. We hypothesized that performance of a 3D input device could be improved if its precision and stability approached that of the mouse, and it were used in conjunction with a visualisation technique that provides good depth perception, while still supporting the ability to perform 3D placement in a coordinated manner. In the current work, we used a Phantom® as our 3 DoF input device because its mechanical

*e-mail: gywang@cim.mcgill.ca
†e-mail: michael.mcguffin@etsmtl.ca
‡e-mail: Francois.Berard@imag.fr
§e-mail: jer@cim.mcgill.ca

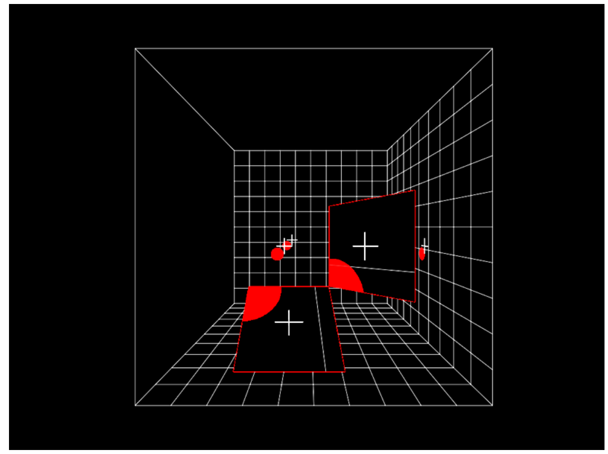


(a) 4-view mode

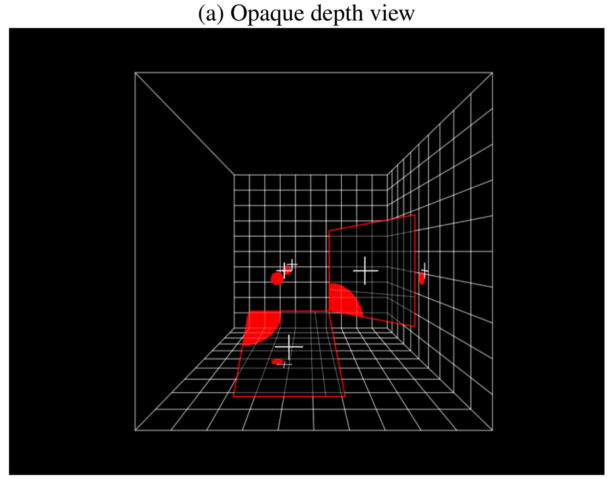


(b) Perspective mode

Figure 1: Graphical displays used in our study. (a) Four-view mode, with 1 perspective view, and 3 orthographic views (right, bottom, and back). (b) Fullscreen perspective mode. In all views, the white 3D crosshair is the controlled cursor and the opaque red sphere is the target.



(a) Opaque depth view



(b) Semi-transparent depth view

Figure 2: Two types of depth views, displaying the neighborhood of the cursor in Figure 1, providing magnified top- and side-views in the vicinity of the cursor.

arm provides some amount of self-stabilization. For the visualization technique, we designed *pop-up depth views* (Figure 2), a novel technique that involves displaying magnified projections of the *xz*- and *yz*-planes in the vicinity of the cursor position when the user's motion slows down below a certain threshold. This compensates for the difficulties of assessing depth in a perspective display, thereby allowing 3D input devices to be used more effectively in 3D environments. In this manner, the user may perform a coordinated initial ballistic motion toward the target, and then use the pop-up depth views for a final, precise, corrective closed-loop motion along all three dimensions. Our experimental evaluation showed that the combination of 3D input with the Phantom, perspective and stereo viewing, and pop-up depth views, allows users to perform 3D placement tasks significantly faster than with a 2D mouse, and with a slightly higher throughput, though at the cost of a slightly higher error rate.

The contributions of this paper are the introduction of pop-up depth views, as well as an experimental evaluation of them in a 3D placement task. The experiment was designed to establish the relative contributions of both input device and viewing condition to 3D placement task performance. The experiment fully crossed the two input devices (mouse and Phantom) with three viewing conditions: orthographic, normal stereo perspective, and stereo perspective with pop-up depth views. The results indicate that there is hope for 3D interaction without the mouse, and indeed, that 3D input devices are capable of supporting greater task efficiency even for targets with a high index of difficulty (ID).

2 Related Work

The literature contains many studies of devices with higher DoF than the mouse for interacting in 3D virtual spaces ([2, 12, 21, 27, 30, 24]). However, most of these studies failed to include a regular mouse for comparison, which we believe was due to a common assumption that 3 DoF devices must be more efficient than the mouse for 3D tasks.

An exception in this regard was Balakrishnan et al. [2], who demonstrated the superiority of their 3D device, the *Rockin' Mouse*, over a regular 2D mouse for 3D placement. However, their experiment may have been biased against the mouse for two reasons. First, their study involved a single perspective view for both devices, which, as noted below, is suboptimal for the mouse. Second, Balakrishnan et al. performed selection of the translational operator with the mouse by clicking on a face of the object's bounding box. These faces could sometimes be seen at a glancing angle with respect to the camera, making them small targets, resulting in a high ID for the sub-task of specifying movement direction. In contrast, for the *Rockin' Mouse* condition, clicking anywhere selected the entire object. Finally, we note that the task in their study did not require the user to position the object with high precision (i.e., target regions were relatively large). In contrast, our current study indicates that the mouse has a greater advantage over 3D input when it comes to placement tasks requiring high precision in the final position.

In previous work [3], we showed that a regular 2D mouse, used in conjunction with front, side, and top 2D orthographic projections, is more efficient than each of three different 3D input devices in a 3D placement task. This result was a central motivation for the work reported here. The 3D devices tested were a SpaceNavigator™(elastic, rate control), a free-space device (isotonic, position control) and the DepthSlider (a 2D mouse augmented with a physical slider for depth control, in position control). The 2D mouse was always used with the conventional four-views display (similar to Figure 1(a)). In a first experiment, the mouse outperformed the 3D devices in completion time and error rate when all devices were used with the four-views display. Our interpretation was that participants were compelled by the orthographic projections to decompose the 3D placement into (at least) two 2D placement sub-tasks, even with 3D input devices that could execute the task in an integrated manner. In a second experiment, the 3D devices were used with a more natural 3D stereo perspective rendering (similar to Figure 1(b)). This resulted in task completion times with the free-space 3D device that were on par with the mouse, but with an error rate at least three times higher. As we discuss later, this high error rate can be associated with the increased difficulty of perceiving depth in a frontal perspective projection, compared with a top- or side-view othographic projection. This motivated the design of our pop-up depth views visualization strategy.

The superiority of the mouse over 3D devices that we demonstrated [3] contradicts the findings of Jacob et al. [16]. The work of the latter on the integrality and separability of tasks and input devices found that the task structure should be matched to the device structure. A 3D placement is an integral task *par excellence*: in the real world, we do not move an object by decomposing the placement in three translations aligned with some axis. Instead, we conceive the action as a single gesture. According to Jacob et al., performance should improve with a device allowing control of the three dimensions in an integral manner. The mouse, however, separates the 3D placement into two 2D placements. The fact that its performance nevertheless remains superior to an integral 3D input device led us to believe that other factors may have a stronger effect. This motivated us to identify these factors, such as the stability of the device and the quality of depth perception offered by feedback, and attempt to improve these for the integral device.

Depth perception in the context of augmented or virtual reality has been well studied in the literature [14, 17]. The consensus is that depth perception is improved when additional cues beyond stereopsis, such as motion parallax, or shadows, are incorporated in the rendering. However, these studies are restricted to *immersive* rendering, e.g., a first-person perspective projection, and none have included orthographic top- or side-views. In another study, Hubona et al. [15] investigated the effects of natural shadows, cast by a single light source above the scene. Although such shadows offer additional depth cues, we suspect these are less valuable than artificial shadows cast orthographically on each of the sides of the workspace, to provide relative 2D position information, as shown in Figure 1(a). Note that shadows shown on a ground plane and/or side plane, whether computed to simulate reality or cast orthographically, can be thought of as showing information similar to that shown in our pop-up depth views. However, our depth views have the advantage that they are displayed *near* the cursor, and so require only tiny eye movements to be seen.

Ramos et al. [22] employed a strategy of pop-up “lenses”, which are comparable to our pop-up depth views. However, their focus was on the input (motor problem) of pointing to small 2D targets by reducing the control-display gain, whereas our study is concerned with the performance improvements possible from an enhancement of the output (display) for 3D placement. Perhaps more similar to our approach, various authors have proposed the use of a small pop-up window, which may provide additional information in the user interface. For example, the multitouch virtual mouse introduced by Vlaming et al. [28] acts not only as a 2D mouse but also as a magnifying lens. Similarly, virtual pads [1] utilized pop-up menus and pop-up windows to decouple the control and visual spaces, so that users can customize the working volume whose location and size are completely independent from the visual representation of the application.

3 Experimental Design

3.1 Why 3D Placement?

One of the most frequent tasks in 2D graphical interaction is the selection of objects through spatial acquisition. However, when the same task is performed in a 3D scene, it is frequently simplified to a 2D task, selecting the *projection* of the object on the screen, otherwise known as picking or ray-casting. There is little incentive, then, to use a high DoF device when the mouse is available and has been highly optimized for this task, especially when combined with smart selection techniques such as the bubble cursor [13], or semantic pointing [8]. Another frequent task is the placement of objects at arbitrary locations. This may involve changing the position of some object in a 3D modeling application, or positioning a marker within volumetric medical data. In either case, the system is unable to provide the assistance it can with picking tasks, because it has no clue as to where the user wants to place the object. The user is thus required to specify all three coordinates in some manner: in other words, this is a true 3D task. Indeed, there is presently no way to perform this task directly with a regular mouse. Instead, the standard solution is to decompose the task into two 2D placements, e.g., one in the x-y plane, and a second in the x-z plane. This suggests that 3 DoF devices should prove superior in performance by supporting task completion in a single action.

3D placement can be reduced to a 2 DoF task by the clever use of constraints: objects are, most of the time, placed next to each other. They are not left floating in the air. Hence, placement can be performed by sliding the moving object on top of the other (static) objects in the scene [26, 20]: a 2 DoF task. While this kind of clever interaction technique illustrates the possibilities of performing 3D tasks with a 2D mouse, we want to compare the efficiency of the mouse with higher DoF devices in a true 3D task. For this reason, we chose the unconstrained placement task in our experiment. Other candidate tasks involving three or more DoF include object rotation, docking (a 6 DoF object placement including both a translation and a rotation), and navigation, which is equivalent to docking the camera that is viewing the scene. Our rationale for selecting placement as the benchmark task is as follows: if a number of 3D input operations are required by the application, we believe that users will choose the best suited device for the most fundamental and frequent of these, i.e., placement. Furthermore, previous studies [19] have shown that coordinating 6 DoF is too difficult to perform in a single action, and thus, 6 DoF tasks such as docking are actually performed by a sequence of 3 DoF translations (placements) and rotations even on 6 DoF devices.

3.2 Display Modes

Figure 1 shows two of the display conditions in our experiment: a four-view condition (perspective, right, bottom, and rear orthographic projections[1]) and a single perspective view. The scene consists of a 3D crosshair cursor and a spherical target, placed inside a 3D cube. Both the 3D crosshair cursor and target cast shadows orthographically on the sides of the cube. The scene is rendered with standard OpenGL (version 1.4) anti-aliasing.

3.3 The Depth View

Several design dimensions and tradeoffs can be considered when deciding how and when to display depth views:

Size How large should the depth views be? Larger views occlude more of the surrounding perspective view, but smaller ones are more difficult to perceive.

Tilt angle To reinforce the user's mental model of the depth views as planes that are perpendicular to the camera plane, we display the depth views as trapezoids, whose converging sides recede toward the more distant edge of the view. How inclined should the planes of the depth views be? As the tilt angle becomes more glancing, the visual cue that this is a perpendicular plane becomes stronger, but at the cost of making it harder to see content within the trapezoid.

Magnification factor We were inspired by previous magnification techniques [4, 11] that replace the original contents under the cursor with a magnified view of the corresponding region. During placement tasks, we expect users to perform an initial coarse movement toward a target using their natural depth acuity, and only later use the depth views when fine positioning is necessary to "home in" on the target. If this is the case, depth views that show magnified views of their planes could be even more helpful in the fine positioning phase of the task. What magnification factor is best?

Transparency Should the depth views be rendered semi-transparently, or opaque? What alpha value should be used?

Persistence Should the depth views always be displayed, or only displayed ("popped up") automatically when the system thinks the user would want them, or popped up manually?

Criterion for popping up If popped up automatically, under what conditions should the the depth views appear?

For some of the above dimensions, we simply chose what seemed reasonable. We set the size of the depth views to 1/3 the size of the cube containing the 3D scene, the tilt angle to 45 degrees, and the magnification factor to 15×.

We then compared two designs: one where the depth views were always displayed, and rendered semi-transparently with an alpha of 0.5 (Figure 2(b)), and another where the depth views were opaque (alpha=1.0, Figure 2(a)) but only popped up when the user slowed down to a speed below 35 units/s for at least 0.4 seconds. ("Units" are defined below.) The first design has the potential disadvantage of occluding the 3D scene when the user is not interested in seeing the content of the depth views, and the second design has the potential disadvantage of forcing the user to wait for the depth views to appear when they *are* desired.

We performed a pilot experiment to compare both designs, and found that performance with the pop-up opaque depth views was superior. In may be, however, that with different kinds of 3D content (such as 3D volumetric medical images), a non-opaque design would be better. Another option that could be considered in future work is to adopt the Shift technique [29] when underlying content is occluded. Elmqvist and Tsigas [9] defined a taxonomy of occlusion management of 3D interaction techniques, intended to increase the benefit of pop-up views in augmenting the visual information provided to users.

Our implementation utilizes OpenGL's stencil buffer.[2] We first draw the 3D scene, then specify a trapezoid-shaped stencil within a mini-frustrum, as seen in Figure 3, in which the magnified depth view is displayed. The use of a mini-frustrum for this purpose avoids deformation of the trapezoid as the cursor position changes in the main frustrum. The area outside of the stencil is masked, and thus, not affected by the depth view rendering.

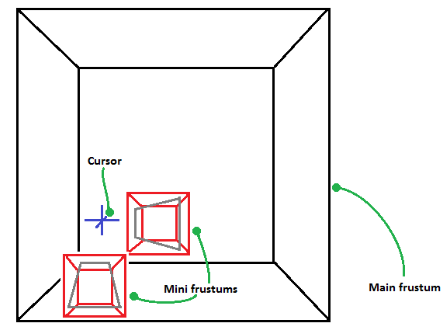


Figure 3: The implementation of depth views.

3.4 Task and Conditions

A basic goal of our study is to compare the mouse with a 3D input device for the task of 3D placement. For the 3D input device, we chose the Phantom, a device which senses the 3D position and 3D orientation of a stylus attached to an articulated arm. This device was selected because of its benefit of inertial stability, which helps reduce the impact of hand tremor that is experienced in free-space movement, making it a reasonable candidate for surpassing the performance of the mouse.

Each trial in our experiment required the user to move the 3D crosshair cursor in Figures 1 and 2 so as to intersect the red spherical target, and confirm the placement by pressing a button. Our study crossed two input devices with three display modes, for a total of six conditions, each denoted by a 3-letter acronym:

Condition	Device	Display	Pop-up Depth-View
MON	Mouse	Orthographic (4-views)	No pop-up
MPN	Mouse	Perspective with stereo	No pop-up
MPP	Mouse	Perspective with stereo	Pop-up
PON	Phantom	Orthographic (4-views)	No pop-up
PPN	Phantom	Perspective with stereo	No pop-up
PPP	Phantom	Perspective with stereo	Pop-up

As indicated in the table above, perspective mode was always displayed stereoscopically.

In all three mouse conditions (MON, MPN, MPP), in addition to the 3D crosshair cursor seen in Figures 1 and 2, the normal system mouse cursor was also displayed. In the MON condition (mouse with 4-views and no pop-up depth views), the user moved the mouse cursor into any of the four views to translate the 3D crosshair cursor within the plane of that view. Note that the user did not have to move the mouse cursor on top of the 3D crosshair cursor to move it: simply moving the mouse cursor into one of the four views, then clicking (with the left mouse button) and dragging

[1]The choice of these views, in contrast with the conventional choice of top, left, and front, was motivated by a desire for consistency with the format of the pop-up depth view, described below. However, this distinction is more conceptual than practical, since the user perceives the same information regardless.

[2]http://www.opengl.org/resources/faq/technical/clipping.htm

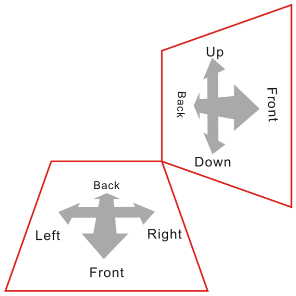


Figure 4: In the experimental conditions that combine mouse input with perspective viewing (the MPN and MPP conditions), trapezoids were displayed beside the 3D crosshair cursor. Dragging within these trapezoids allowed the user to move the 3D crosshair cursor within the plane of the trapezoid.

would cause the 3D crosshair cursor to be dragged by the same relative displacement. Thus, switching views to translate within a different plane had a very low cost, since the user only had to move the mouse cursor into a different quadrant of the screen to start dragging within the plane of that view, and each quadrant is a relatively large target.

Note that, in general, multiple orthographic views have the disadvantage of requiring mental effort to integrate their content into an internal 3D representation. However, for the simple 3D placement task in our study, such a mental integration of multiple views is in fact not necessary. The user simply positions the 3D crosshair cursor in one orthographic view to be over the target, and then repeats this operation within another orthographic view, and is finished. There is no need to return to the first view, or to position the crosshair within more than two views, so long as the positioning within each of the two planes is done correctly. This makes it easy to break down the placement task into two steps, as is necessary anyway with a 2D input device.

In the other mouse conditions (MPN and MPP), the scene was displayed within a perspective view (Figure 1(b)), and trapezoids were displayed beside the 3D crosshair cursor (Figure 4). The user moved the 3D crosshair cursor by positioning the mouse cursor within one of the trapezoids and dragging (with the left mouse button) to move within the plane of that trapezoid (either the xz- or yz-plane). In the MPN condition, these trapezoids, displayed with full transparency, were provided only as affordances for dragging the 3D crosshair cursor within different planes. However, in the MPP condition, the area within the trapezoids was used for display of depth views whenever the mouse cursor slowed down.

Note that the mouse interactions used in our experiment avoid one of the potential biases with the Rockin' Mouse study mentioned in Section 2: the view quadrants in MON, and the trapezoids in MPN and MPP, are relatively large targets that never appear oriented at a glancing angle, hence there is a low cost in switching between translation planes. The other potential biases mentioned earlier are also avoided in our study, since we fully cross display conditions with input devices, and also tested small targets, as we describe later.

In all mouse conditions (MON, MPN, MPP), the user confirmed their placement of the 3D crosshair cursor and completed the trial by clicking with the right mouse button.

In all three Phantom conditions (PON, PPN, PPP), the normal system mouse cursor was not displayed. Only the 3D crosshair cursor was visible, which was manipulated in absolute mode, "attached" to the position of the Phantom's stylus. In the PPP condition, the trapezoids were also displayed.

Regardless of the display mode used, no switching of planes was necessary with the Phantom, and the user was free to perform fully 3-dimensional coordinated movements. To complete each trial, we could have required users to press a button on the Phantom stylus, however this would have disturbed the position of the stylus at the moment the button is pressed. Therefore, we instead had users use their non-dominant hand to click the right mouse button. to complete the trial, without moving their dominant hand from the Phantom.

At first glance, it might seem strange that our experiment required users to position a cursor on a discrete target, when real-world discrete targets could probably be selected faster with 2D ray casting and/or pointing facilitation techniques such as snapping, or a bubble cursor [13] or semantic pointing [8]. However, as already explained in the introduction, the anticipated application of our pop-up depth views is in unconstrained 3D placement tasks, such as positioning a 3D crosshair within a 3D volumetric medical image. In such situations, only the user knows what their target is and how precisely they must acquire it. Our experiment, therefore, simply simulates the situation where the user has some imagined 3D target that they wish to acquire, and this is displayed in the form of a 3D sphere.

3.5 Apparatus

The experiment was conducted on an HP xw9300 workstation. Stereoscopic display was achieved by polarized projection using two Mitsubishi XD490U DLP projectors, vertically aligned. Monoscopic display was achieved using a single projector, with the polarizing filter left in place. To ensure that participants experienced the same luminance in all conditions, we removed their polarized glasses during monoscopic viewing. The projected graphics on the non-depolarized silver screen was approximately 170$\times$130 cm at a resolution of 1024$\times$768 pixels. Subjects were seated approximately 250 cm in front of the screen, and manipulated the input devices at a comfortable position.

A SensAble Phantom Omni and a Logitech Gaming Mouse G500 were used as the input devices. Both the mouse and Phantom were sampled at high update rates and read by the computer at a minimum of 200 Hz. The graphic rendering achieved a refresh rate of at least 1250 fps (in stereo mode). The bottleneck was thus the projector refresh rate of 60 Hz. At this value, there was no effect of input device latency, which was well studied by Teather et al. [25]. For both input devices, we assumed that any difference in latencies at the application level was relatively minor.

3.5.1 Input Device Mapping

The Phantom input device has a physically constrained working volume, so a fixed amount of physical motion must correspond to a movement through the full range of any of the axes of our experimental working volume, e.g., a translation of 6.35 cm moves the cursor from the left to the right wall of the virtual 3D space. For the mouse, however, there are no such hard constraints. We therefore considered what mouse mapping parameters to use for the experiment. Our first pilot investigated task performance with the mouse, comparing performance at its default control-display ratio at a resolution of 800 dpi and at the lower resolution of 183 dpi, equalized in software to that of the Phantom. In the latter case, the mouse requires a physical displacement identical to that of the Phantom to achieve the equivalent cursor movement. The results indicated faster performance with the default mouse mapping, most evident with larger targets, although with a slightly higher error rate. Since our focus was on performance time, we opted to use the normal mouse at 800 dpi, in conjunction with its default mapping.

We also decided to leave the default acceleration for the mouse turned on for the experiment. Although this arguably biases results in favor of the mouse, this is how mice are almost always used in the real world. Furthermore, the literature indicates that mouse performance benefits only modestly from acceleration [5]. Future work could study enhancing 3D input with analogous acceleration techniques.

3.6 Subject Pool

Twelve participants (11 male and one female) took part in the study, ranging between 23 and 39 years in age. All participants were right-handed and had an average, normal stereo acuity of 41.25 (SD 13.16) seconds of arc at 40.64 cm (16 inches), evaluated with the Randot® SO-002 test.[3]

3.7 Design and Procedure

In order for a trial to succeed, the center of the 3D crosshair cursor had to be inside the target when the user clicked on the right mouse button. Otherwise, when the button was pressed, a beep was sounded to indicate the error and the subject was forced to attempt again, from the current position, up to a maximum of three attempts. This discouraged participants from simply clicking anywhere to bypass a difficult target, as the forced re-attempts would slow them down considerably. Targets were positioned pseudorandomly throughout the volume, one at a time, requiring movement across all three axes between successfully acquired targets.

Each trial required movement along all three axes, but not equal distances: for each trial, the required movement involved four times as much motion along one of the x-, y-, or z-axes as along the other two axes. The straight-line distance to the target was always 320 units.[4] The diameter of the spherical target was either 6, 12, or 18 units, corresponding to indices of difficulty ID of 5.76, 4.79, and 4.23 bits, respectively[5].

Each of the 12 participants performed six blocks of trials (one for each of the MON, MPN, MPP, PON, PPN, PPP conditions). Each block consisted of 144 trials: three target sizes (6, 12, 18) crossed with 6 movement directions ($\pm x$, $\pm y$, $\pm z$) repeated 8 times each. In total, there were
12 participants $\times$
6 conditions (MON, MPN, MPP, PON, PPN, PPP) $\times$
3 target widths (6, 12, 18) $\times$
6 directions ($\pm x$, $\pm y$, $\pm z$) $\times$
8 repetitions = 10368 trials

Within each block, the target sizes were pseudorandomly ordered, with the same random seeds used for each participant to reproduce the same pseudorandom sequence. Furthermore, the ordering of blocks was varied so that half the participants used the mouse first, and the other half used the Phantom first. In addition, the ordering of the three blocks for each device was counterbalanced using a factorial design. The ordering of the six blocks for each participant is given by the following table:

Participant	Ordering					
1	MON	MPN	MPP	PON	PPN	PPP
2	MON	MPP	MPN	PON	PPP	PPN
3	MPN	MON	MPP	PPN	PON	PPP
4	MPN	MPP	MON	PPN	PPP	PON
5	MPP	MPN	MON	PPP	PPN	PON
6	MPP	MON	MPN	PPP	PON	PPN
7	PON	PPN	PPP	MON	MPN	MPP
8	PON	PPP	PPN	MON	MPP	MPN
9	PPN	PON	PPP	MPN	MON	MPP
10	PPN	PPP	PON	MPN	MPP	MON
11	PPP	PPN	PON	MPP	MPN	MON
12	PPP	PON	PPN	MPP	MON	MPN

Participants took short breaks after each sequence of 24 trials to rest, and took a longer break at the half-way point of the experiment before testing the second device. For most participants, this meant testing on two separate days, and for the remaining subjects, testing was split between a morning and afternoon session.

4 Results and Analysis

We removed error trials, i.e., those in which users had not successfully acquired the target on the first click, and then computed the mean and standard deviation of movement time for each of the 18 (condition, target size) combinations. Any trials whose movement time was more than three standard deviations away from the mean were removed as outliers.[6]

No significant learning effects were seen across the data, likely because of the simplicity of the task and the ease of use of the two input devices.

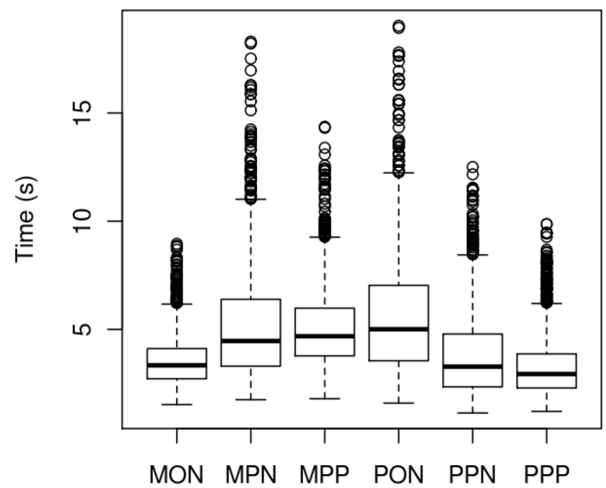


Figure 5: Movement times by condition. Boxes extend from the 1st to the 3rd quartile, and whiskers are located at the minimum or maximum value, *or* at 1.5 times the interquartile range, whichever is closer to the median.

Figure 5 shows the movement times for each condition. The following table breaks down the data in further detail, listing average movement times in seconds (with error rates in parentheses). Error rates above 6% are in bold.

target size	MON	MPN	MPP	PON	PPN	PPP
6	4.41 (2.3%)	6.59 (**17%**)	6.13 (2.1%)	7.63 (**20%**)	4.96 (**20%**)	4.25 (2.4%)
12	3.31 (0.9%)	4.94 (2.1%)	4.74 (0.9%)	5.39 (5.6%)	3.76 (4.0%)	3.12 (1.4%)
18	2.95 (0.7%)	4.25 (1.4%)	4.39 (1.6%)	4.16 (2.1%)	2.96 (3.6%)	2.51 (1.7%)

A 2×3 Analysis of Variance (ANOVA) found that device ($F_{1,11} = 154.12$) and display technique ($F_{2,11} = 53.73$) both had a significant effect on time ($p < 0.0001$). We also performed a multi-factor ANOVA treating the six conditions as a single six-level factor, to make it easier to compare the six (device,display technique) combinations. This analysis found that participant, condition, and target size each had a significant effect on movement time (all $p < 0.0001$), and furthermore there was a significant condition $\times$ target size interaction ($F_{10,11} = 30.14, p < 0.0001$). A Tukey HSD test found that all three target sizes had significantly different movement times, and also found that all six conditions

[3] `http://www.stereooptical.com/html/stereo-test.html`

[4] Thus, the required movement was approximately 302 units along one axis, and 75 units along each of the other two axes, since $302^2 + 75^2 + 75^2 \approx 320^2$.

[5] Based on the Shannon formulation [7, 18] of $ID = \log_2(320/W+1)$, where W is the target width.

[6] These accounted for between 0.7% and 1.9% of our data.

had significantly different movement times, except for the (MPN, MPP) pair.

The results show that the three worst conditions in terms of movement time are MPN, MPP, and PON. It is not surprising to us that MPN and PON did poorly, as our previous experience indicated that the mouse is best matched to orthographic output, whereas a 3D input device such as the Phantom is best suited to perspective output. Notice in particular the high error rates for MPN and PON at the smallest target size of 6. Interestingly, although MPP was also slow, it had a much lower error rate than MPN at the smallest target size, demonstrating the pop-up depth view helped users correctly acquire the target before clicking the first time. We suspect that the high error rates in PON (and PPN) were partially due to the difficulty of holding the Phantom still while checking two different views or shadows. Checking depth views, in contrast, requires much smaller eye movements.

The three best conditions, in increasing order of movement time, were PPP, MON, and PPN, which were significantly different from each other and significantly faster than the other conditions. PPP also had lower error rates than PPN, again demonstrating the pop-up depth view helped users correctly acquire the target before clicking the first time. PPP significantly outperformed MON in terms of movement time, however this came at the cost of a slightly increased error rate.

Although the task in our experiment was not exactly a traditional Fitts' Law pointing task [23], it can be thought of as pointing in 3D. Thus, a throughput analysis seems appropriate as a way to compare different techniques that involve a tradeoff between time and accuracy. We note, as well, that the MON condition involved at least two targeting actions in each trial whereas the PPP condition involved only one. However, we argue that these deviations from a pure Fitts' Law task do not detract from the value of our analysis, since we are, in any case, concerned with how long it takes the user to express ID bits with each technique. To compare MON and PPP in a way that takes into account both their average movement times and error rates, we computed the effective width of each target based on its error rate, and then computed the index of performance for each technique. We used the definition in §3.4 by Crossman [7] to find the effective width W_e for a given error rate ER:

$$\begin{aligned} W_e &= W \times 2.066.../\text{z\_score}(1-ER) \\ &= W\sqrt{\pi e/2}\,/(\sqrt{2}\,\text{erf}^{-1}(1-ER)) \\ &= W\sqrt{\pi e}/(2\,\text{erf}^{-1}(1-ER)) \end{aligned}$$

Next, we used the effective widths to compute effective indices of difficulty ID_e, and then computed the index of performance of each technique as $IP = \overline{ID_e}/\overline{MT}$, where $\overline{ID_e}$ and $\overline{MT}$ are the average effective index of difficulty and average movement time, respectively. The results: for **MON: 1.46 bits/s**; and for **PPP: 1.55 bits/s**. Thus, PPP appears to enable slightly more efficient expression of information.

After the participants finished all six conditions, we let them subjectively rate each condition, on a 5-1 Likert scale, 5 being the best. They rated the conditions according to three aspects: personal preference, efficiency and comfort. Results are illustrated in Figure 6. The MON and PPP condition where preferred to the other ones, but there was not significant differences between the two.

5 Conclusions and Future Work

Motivated by the suspicion that a higher-degree-of-freedom input device should offer some advantage over the 2D mouse for a 3D acquisition task, we developed the pop-up depth view as a display augmentation that compensates for the limits of depth cues in stereoscopic perspective displays. We then conducted an experiment to evaluate the performance benefits of this approach. The results demonstrated that the Phantom with a stereo perspective view

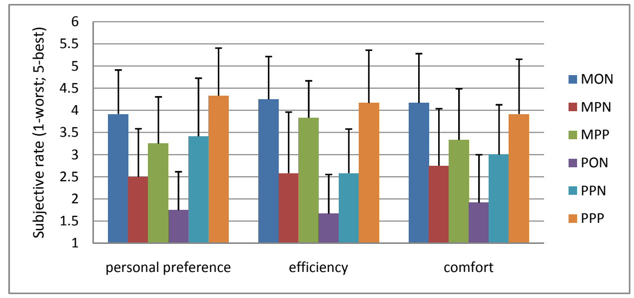


Figure 6: Subjective ratings of six conditions by the participants. 1 is the worst, 5 is best.

and pop-up depth views (PPP) was significantly faster than the other tested conditions. In particular, this proved to be significantly faster than the condition of mouse with an orthographic view (MON), and had a higher effective index of performance than MON, despite the facts that (1) users have much more experience with 2D mice than with 3D input devices, (2) the Phantom did not offer any form of cursor acceleration, whereas the mouse in the experiment had its default cursor acceleration enabled, and (3) unlike the case of more real-world tasks, in the MON condition of our experiment, the 3D acquisition could be decoupled trivially into two separate pointing tasks, with no cognitive effort required to integrate the multiple views or otherwise spend time trying to understand which element in one orthographic view corresponds to a desired element in another orthographic view. Thus, in a real-world system, we could expect an interface based on PPP (3D input, perspective with stereo, and popup depth views), perhaps combined with a pointing acceleration technique, to exhibit an even greater performance advantage over an interface based on MON.

In future work, it may be interesting to perform another experiment involving a variant of the MON condition (mouse with orthographic views) crossed with multiple alternative methods for switching between planes. The switching methods might include hotkeys, a foot pedal, a 3D widget [6], or clicking on different faces of an object [2]. Also of interest would be to test 3D input conditions without stereo, and/or with head tracking, and/or acceleration techniques for the 3D input that are analogous to standard mouse acceleration. It could also be worthwhile to conduct a longitudinal study to allow users to acquire greater expertise with the 3D input device, to investigate the effects of long-term learning on their ability to exploit the pop-up depth views.

Finally, pop-up depth views might also be applied to discrete selection tasks where targets are discrete objects but densely fill the 3D space, precluding the use of standard 2D ray casting. In such situations, depth views might also be combined with a bubble cursor [13] or other related techniques.

6 Acknowledgments

This research was supported by the Natural Sciences and Engineering Research Council (NSERC). The authors would like to thank Roberto Lopez-Hernandez for his implementation of the pop-up view technique and Jessica Ip for her input to discussions.

References

[1] C. Anclujar and F. Argelaguet. Virtual pads: Decoupling motor space and visual space for flexible manipulation of 2d windows within ves. In *3D User Interfaces, 2007. 3DUI '07. IEEE Symposium on*, pages 99–106, Mar. 2007.

[2] R. Balakrishnan, T. Baudel, G. Kurtenbach, and G. Fitzmaurice. The Rockin'Mouse: Integral 3D manipulation on a plane. In *ACM Confer-*

ence on Human Factors in Computing Systems (CHI), pages 311–318. ACM, 1997.

[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 *IFIP Conference on Human-Computer Interaction, INTERACT*, volume 5727 of *Lecture Notes in Computer Science*, chapter 45, pages 400–414. Springer, 2009.

[4] E. A. Bier, M. C. Stone, K. Pier, W. Buxton, and T. D. DeRose. Toolglass and magic lenses: the see-through interface. In *ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 73–80. ACM, 1993.

[5] G. Casiez, D. Vogel, R. Balakrishnan, and A. Cockburn. The impact of control-display gain on user performance in pointing tasks. *Human-Computer Interaction*, 23(3):215–250, 2008.

[6] D. B. Conner, S. S. Snibbe, K. P. Herndon, D. C. Robbins, R. C. Zeleznik, and A. V. Dam. Three-dimensional widgets. In *Symposium on Interactive 3D Graphics (SIGGRAPH)*, volume 25, pages 183–188. ACM, 1992.

[7] E. R. F. W. Crossman. The speed and accuracy of simple hand movements. Technical report, Department of Engineering Production, University of Birmingham, 1957.

[8] N. Elmqvist and J.-D. Fekete. Semantic pointing for object picking in complex 3D environments. In *Proceedings of graphics interface (GI)*, pages 243–250, Toronto, Ont., Canada, Canada, 2008. Canadian Information Processing Society.

[9] N. Elmqvist and P. Tsigas. A taxonomy of 3d occlusion management techniques. In *Virtual Reality Conference, 2007. VR '07. IEEE*, pages 51 –58, Mar. 2007.

[10] T. T. Elvins, D. R. Nadeau, and D. Kirsh. Worldlets-3d thumbnails for wayfinding in virtual environments. In *Proceedings of the 10th annual ACM symposium on User interface software and technology*, UIST '97, pages 21–30, New York, NY, USA, 1997. ACM.

[11] L. Findlater, A. Jansen, K. Shinohara, M. Dixon, P. Kamb, J. Rakita, and J. O. Wobbrock. Enhanced area cursors: reducing fine pointing demands for people with motor impairments. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, UIST '10, pages 153–162, New York, NY, USA, 2010. ACM.

[12] B. Froehlich, J. Hochstrate, V. Skuk, and A. Huckauf. The GlobeFish and the GlobeMouse: two new six degree of freedom input devices for graphics applications. In *ACM Conference on Human Factors in Computing Systems (CHI)*, pages 191–199. ACM, 2006.

[13] T. Grossman and R. Balakrishnan. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In *ACM Conference on Human Factors in Computing Systems (CHI)*, pages 281–290. ACM, 2005.

[14] T. Grossman and R. Balakrishnan. An evaluation of depth perception on volumetric displays. In *ACM Conference on Advanced Visual Interfaces (AVI)*, pages 193–200. ACM, 2006.

[15] G. S. Hubona, G. W. Shirah, and D. K. Jennings. The effects of cast shadows and stereopsis on performing computer-generated spatial tasks. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 34(4):483–493, 2004.

[16] R. J. K. Jacob, L. E. Sibert, D. C. McFarlane, and J. M. Preston Mullen. Integrality and separability of input devices. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 1(1):3–26, 1994.

[17] J. A. Jones, J. E. Swan, II, G. Singh, E. Kolstad, and S. R. Ellis. The effects of virtual reality, augmented reality, and motion parallax on egocentric depth perception. In *Symposium on Applied Perception in Graphics and Visualization (APGV)*, pages 9–14. ACM, 2008.

[18] I. S. MacKenzie. Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction*, 7:91–139, 1992.

[19] M. R. Masliah and P. Milgram. Measuring the allocation of control in a 6 degree-of-freedom docking experiment. In *ACM Conference on Human Factors in Computing Systems (CHI)*, pages 25–32, New York, NY, USA, 2000. ACM.

[20] J.-Y. Oh and W. Stuerzlinger. Moving objects with 2D input devices in CAD systems and desktop virtual environments. In *Proceedings of Graphics Interface (GI)*, pages 195–202. Canadian Human-Computer Communications Society, 2005.

[21] M. Ortega and L. Nigay. AirMouse: Finger gesture for 2D and 3D interaction. In *IFIP Conference on Human-Computer Interaction, INTERACT*, volume 5727 of *Lecture Notes in Computer Science*, chapter Chapter 28, pages 214–227. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[22] G. Ramos, A. Cockburn, R. Balakrishnan, and M. Beaudouin-Lafon. Pointing lenses: facilitating stylus input through visual-and motor-space magnification. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '07, pages 757–766, New York, NY, USA, 2007. ACM.

[23] R. W. Soukoreff and I. S. MacKenzie. Towards a standard for pointing device evaluation, perspectives on 27 years of fitts' law research in hci. *Int. J. Hum.-Comput. Stud.*, 61:751–789, December 2004.

[24] R. Stoakley, M. J. Conway, and R. Pausch. Virtual reality on a wim: interactive worlds in miniature. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '95, pages 265–272, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[25] R. J. Teather, A. Pavlovych, W. Stuerzlinger, and I. S. MacKenzie. Effects of tracking technology, latency, and spatial jitter on object movement. In *Proceedings of the 2009 IEEE Symposium on 3D User Interfaces*, 3DUI '09, pages 43–50, Washington, DC, USA, 2009. IEEE Computer Society.

[26] R. J. Teather and W. Stuerzlinger. Assessing the effects of orientation and device on (constrained) 3D movement techniques. In *IEEE Symposium on 3D User Interface*, pages 43–50, 2008.

[27] D. Venolia. Facile 3D direct manipulation. In *ACM Conference on Human Factors in Computing Systems (CHI)*, pages 31–36. ACM, 1993.

[28] L. Vlaming, C. Collins, M. Hancock, M. Nacenta, T. Isenberg, and S. Carpendale. Integrating 2d mouse emulation with 3d manipulation for visualizations on a multi-touch table. In *ACM International Conference on Interactive Tabletops and Surfaces*, ITS '10, pages 221–230, New York, NY, USA, 2010. ACM.

[29] D. Vogel and P. Baudisch. Shift: a technique for operating pen-based interfaces using touch. In *ACM Conference on Human Factors in Computing Systems (CHI)*, pages 657–666. ACM, 2007.

[30] C. Ware. Using hand position for virtual object placement. *The Visual Computer*, 6(5):245–253, 1990.

[31] C. Ware and D. R. Jessome. Using the bat: a six dimensional mouse for object placement. In *Proceedings of Graphics Interface (GI)*, pages 119–124. Canadian Information Processing Society, 1988.

[32] C. Ware and K. Lowther. Selection using a one-eyed cursor in a fish tank VR environment. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 4(4):309–322, December 1997.