

Withindows: A Framework for Transitional Desktop and Immersive User Interfaces

Alex Hill[†]

University of Illinois at Chicago

Andrew Johnson[‡]

University of Illinois at Chicago

ABSTRACT

The uniqueness of 3D interaction is often used to justify levels of user fatigue that are significantly higher than those of desktop systems. Object manipulation and symbolic manipulation techniques based strictly on first person perspective are also generally less efficient than their desktop counterparts. Instead of considering the two environments as distinct, we have focused on the idea that desktop applications will likely need to transition smoothly into full immersion through intermediate states. The Withindows framework uses image-plane selection and through-the-lens techniques in an attempt to smooth the movement of both traditional and immersive applications across transitional states such as desktop stereo and multi-display setups. We propose using a virtual cursor in the dominant eye and a reinforcing cursor in the non-dominant eye to avoid ambiguity problems that have discouraged the use of image-plane selection in stereo. We show how image-plane selection resolves non-linear control-display relationships inherent in some approaches to desktop stereo. When combined with through-the-lens techniques, image-plane selection allows immersive viewpoint management and 2½D object manipulation techniques analogous to those on the desktop. This approach resolves global search and scaling problems inherent in prior through-the-lens implementations. We describe extensions for 6 DOF input devices that do not supersede the default interaction method. We developed a single-authored virtual world builder as a proof of concept application of our framework. Our evaluations found alternate perspectives useful but our implementation of viewing windows proved fatiguing to some users.

KEYWORDS: Through the Lens, Image Plane, Virtual Reality, Augmented Reality

INDEX TERMS: H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems - Artificial, augmented, and virtual realities; I.3.6 [Computer Graphics]: Methodology and Techniques - Interaction Techniques

1 INTRODUCTION

Immersive virtual environments (IVEs) in the domain of simulation are now generating significant return on investment in areas such as flight simulation, industrial and surgical training, design reviews, and psychological treatment. The modest goals of immersion and reproduction of physical tasks mean that fatigue and efficiency concerns remain only those of the underlying task. However, once the domain changes to that of augmenting the

individual, the productivity levels of 3D user interfaces stand in sharp contrast to those of typical desktop environments. Three dimensional user interfaces routinely exhibit poor manipulation of symbolic data, high levels of fatigue and workflow patterns that restrict most development work to the desktop. If they are to find broad acceptance, 3D user interfaces will have to begin manipulating symbolic information on a par with desktop systems [1]. There has been a persistent belief that there are inherent efficiencies in physically intuitive methods. This materialism, or corporealism, has often resulted in techniques that are fundamentally less efficient than their desktop equivalents. A prime example is the proliferation of first person techniques for manipulating objects at a distance. Such techniques remain popular despite the obvious inefficiency of positioning objects along the depth axis. The results of corporealism are high technical and cost barriers to the application of 3D technology to the augmentation of the individual.

The trends of device miniaturization, ubiquitous display, wireless communications and internet geo-referencing are likely to converge in an environment where computation moves beyond the boundaries of physical devices and into the space around us. We believe that it is more likely that desktop applications will make a gradual transition into this space than it is that there will be a paradigm shift into using 3D interfaces. We have chosen to focus our efforts on the idea that contemporary applications will make this transition through intermediate states such as desktop stereo, large-format display and projector-based augmented reality. Our goal has been to develop a framework that not only facilitates the smooth transition of desktop computation into immersion but also allows immersive applications to make a similar transition onto the desktop.

There are a number of benefits to supporting a transitional framework between desktop and immersion. Most importantly, it creates the potential for the normally difficult task of programming applications for both environments to be accomplished in an integrated development environment (IDE). There is also a benefit to creating a route for legacy applications to make the move into transitional states without significant redevelopment. Some of the most important benefits of a transitional framework involve using IVE applications on the desktop. Because immersive technology has traditionally been expensive and difficult to maintain, users are frequently limited in their access to it. Providing access to applications developed for immersion on the desktop allows users to learn program functionality prior to immersive use. Prior exposure to interface and menu elements is likely to let users focus their efforts on the immersive features of the application and may even provide some resilience to the frequent resolution limitations associated with the technology.

An important part of a transitional framework is the choice of a canonical input method and associated input device. We use image-plane selection as the primary interaction method because it is a more general case of the point-and-click desktop interface. This property allows the primary elements of immersion to be

[†]e-mail: ahill@evl.uic.edu

[‡]e-mail: ajohnson@uic.edu

added and removed with greater flexibility than other methods like ray-casting. Image plane selection relies either on a simple tracked mouse or computer vision techniques applied to the un-instrumented hand. We propose using a virtual cursor in the dominant eye and a reinforcing cursor in the non-dominant eye to avoid ambiguity problems that have discouraged image-plane techniques in stereo. We show how image-plane selection can solve non-linear control-display relationships inherent in some implementations of desktop stereo. We also show how image-plane concepts work naturally with other best in practice techniques for large-format displays, multiple-display setups and projector-based augmented reality.

To bridge the gap between the primary tasks of immersive and traditional applications, we combine image-plane selection with through-the-lens techniques (TTL). When combined with image-plane selection, TTL techniques create interactions that are analogous to the manipulation and view management techniques used by desktop 3D modeling and CAD applications. Our image-plane TTL techniques resolve several problems with prior implementations; the most important of these being scaling problems during object manipulations. We also describe techniques to leverage 6 DOF input devices without superseding the default interaction methods. The third and, in some ways, most important aspect of the framework is a strategy to significantly reduce fatigue by moving immersive interactions with traditional interfaces and TTL windows into a position below the hand of the user.

2 RELATED WORK

Object selection and manipulation in 3D environments has had a strong focus on first person perspective. Methods such as Go-Go and HOMER both apply a scaling factor between hand input and virtual hand distance [3,19]. The inability to modulate this scaling factor prevents accurate positioning of objects at a distance and is known as the scaling problem [20]. Pierce resolved the problem by using a combination of image-plane selection and a reference object for his Voodoo Dolls technique. Although Voodoo Dolls solves the scaling problem, it manipulates objects outside of their context and, thus, only allows positioning relative to a single object [18]. Stoakley exploited the power of exocentric viewpoints to create his Worlds in Miniature (WIM) techniques for object manipulation and virtual travel. As with the Voodoo Dolls technique, users find the superimposition of WIM content over the surrounding scene confusing [26]. Influenced by WIM and 3D magic lens, Stoev and Schmalstieg introduced through-the-lens techniques using alternate viewpoint windows into the scene [32,28]. This approach overcomes problems with producing WIM models and disambiguates alternate viewpoints from surrounding content.

Pierce used image-plane, or occlusion-based, selection to choose the first object intersected by a ray cast from the eye through a point on the hand [17]. Significant user fatigue associated with selecting objects out in front of the user with this method has contributed to making ray-casting the predominant choice for object selection at a distance. However, Pierce and Poupriev found image-plane selection combined with waist level manipulations produced less fatigue than methods such as HOMER that use ray-casting exclusively [18]. Ray-casting selection has been described as effectively a two-dimensional task when objects are sufficiently far from the user [20]. However, Wingrave has shown that image-plane selection is faster and no less preferred than ray-casting when targets are more evenly distributed in depth [31]. A number of studies have also shown that laser-pointer based techniques, effectively ray-casting without the visual reinforcement allowed in an IVE, are less accurate at a distance than either a traditional mouse or a virtual mouse, an

image-plane technique where hand motion and the resulting onscreen cursor motion are only relative [29]. Lee argued that users were more accurate using a virtual mouse than with ray-casting or image-plane because they could steady their arm against their body [10].

The predominant methods for symbolic manipulation in 3D have been virtual hand and ray-casting techniques. Virtual hand techniques without haptics tend to be inefficient because they impose depth as an unconstrained degree of freedom [12]. Bowman effectively combined both pseudo-haptic input and alternate viewpoint manipulations for his Designing Animal Habitats application [2]. For less spatially constrained interface elements such as cascading menus, ray-casting has been popular but has some problems. Ray-casting is confusing at close distances unless it is coupled with high quality stereo separation, and placing interface elements a significant distance from the user creates potential conflicts with the surrounding content [4]. The limitations of 3D symbolic manipulation have motivated hybrid solutions that use both a standard mouse on XWindows interfaces and a 6 DOF tracker for immersive tasks [7]. This approach has not proven popular because of the frequent input device switching it forces upon the user. Efforts to migrate desktop applications into 3D include that of Feiner to register XWindows bitmaps with trackers in an augmented environment [8]. More recently, these efforts have involved the development of 3D APIs or the use of VNC clients to render applications to 3D polygons [9,5]. Such efforts neither seek a canonical interaction methodology nor a cross-platform development framework between desktop and immersive environments.

The construction of 3D content has been a popular domain for immersive applications. Butterworth created a fully immersive modeler with many functions analogous to desktop modelers [6]. Leigh and Johnson incorporated alternating egocentric and exocentric perspectives and collaborative desktop interfaces in their CALVIN and NICE applications respectively [11,23]. Holm created a collaborative desktop and immersive virtual world building application [22]. A common attribute of these efforts is that any collaborative or workflow supporting applications on the desktop must be developed separately and employ a significantly different interface. Stenius created a collaborative application for 3D modeling using the DIVE platform [25]. Applications that use the DIVE system can potentially be used tele-collaboratively in immersive settings, but the necessary user interface development was never undertaken.

3 THE WITHINDOWS FRAMEWORK

The Withindows framework seeks to create a unified methodology for developing traditional desktop and fully immersive interfaces that allows applications to easily move along a continuum between both environments. These transitional setups include desktop stereo, large-format displays, multiple-display environments and projector-based augmented reality. The framework is composed of three main features; image-plane selection as a canonical input method, through-the-lens techniques to optimize interaction with 3D space and the movement of most immersive interactions to a location underhand to avoid fatigue.

Image-plane selection works by occluding content with the hand or a virtual cursor attached to it. Image-plane selection can be considered as a generalization of the more special desktop point-and-click interface. The desktop setup uses a fixed monoscopic viewpoint in front of desktop content and restricts cursor movement to a plane in front of that content (figure 1). Image-plane selection merely represents relieving the constraints on both cursor position and user viewpoint. This conception of the desktop allows piecemeal addition of the full head tracking, high DOF hand tracking and stereo display attributes that make up

most transitional configurations. A number of best-in-practice techniques for such transitional setups can be improved by considering them within the context of image-plane selection.

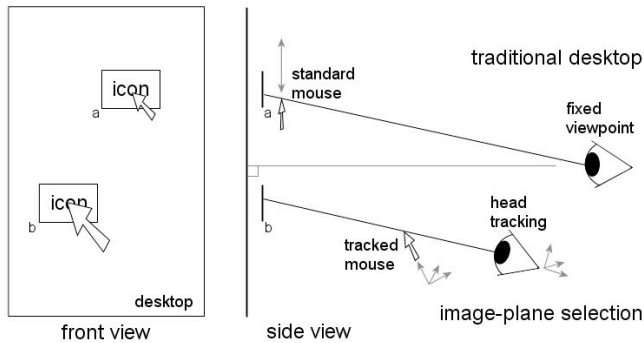


Figure 1: a) Constraining the cursor to a plane and fixing the viewpoint emulates the desktop interface. b) Relieving constraints leads to image-plane selection and other transitional configurations.

Through-the-lens techniques use an alternate viewing window to accomplish the primary immersive tasks of object selection, object manipulation and global search. By using image-plane selection on TTL viewing windows we can create a direct analog to techniques commonly used to manage viewpoints and manipulate 3D content on the desktop (figure 2). These desktop object manipulation techniques, known as 2½D techniques, avoid scaling problems because window zoom factors adjust the control-display relationship between mouse and object. Using image-plane selection on TTL windows solves several problems with prior implementations that relied primarily on virtual hand techniques for viewpoint management and object manipulations. The encapsulation of primary immersive tasks within TTL windows creates the opportunity to easily transition immersive applications onto the desktop. When all application functionality is made available within TTL windows, they can be presented unaltered on the desktop or in other transitional environments.

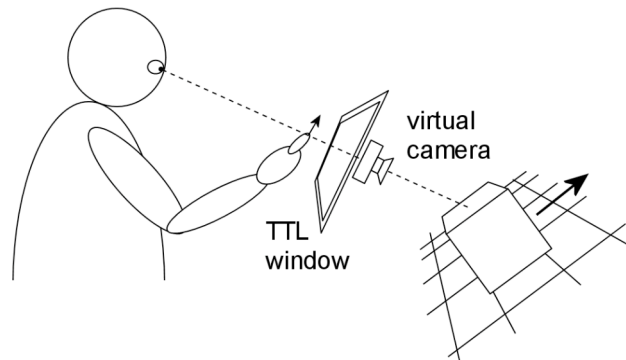


Figure 2: Image-plane selection on alternate viewpoint windows facilitates 2½D techniques analogous to those on the desktop that restrict object movement to a plane under the cursor.

This framework presents an opportunity to reduce the fatigue commonly associated with the long-term interactions in 3D environments. Most approaches to immersive tasks seek to leverage physically intuitive input techniques at the expense of increased fatigue. By avoiding reaching, bending and arm movement away from the body, users can work for longer durations in 3D environments. Using image-plane selection on TTL windows placed in a comfortable position below the hand emulates the familiar mouse interactions used on the desktop. Such a formulation does not preclude the use of more direct interactions with the surrounding environment for selection,

manipulation and search but, when approached in a consistent manner, does offer tangible options for reducing fatigue. Most of the 2½D manipulation and 6 DOF viewpoint management techniques described in the sections that follow have also been previously used directly within the surrounding environment.

3.1 Image-plane Selection in Stereo

Stereo display of content on the desktop has traditionally used a cursor presented at the depth of the desktop. This technique can disrupt the stereo image and creates problems when it becomes necessary to select a location on the stereo content. A solution to this problem involves rendering the cursor at the same depth as the content underneath it [24]. The small range of depth on the desktop is unlikely to cause user discomfort due to abrupt cursor depth changes. However, immersive environments routinely place selectable content anywhere between the user and the horizon. Maintaining the virtual cursor at a fixed depth such as the hand creates an ambiguous selection condition because each eye may see the cursor occluding a different object (figure 3). For these reasons, implementations of image-plane selection have routinely used a mono display setup.

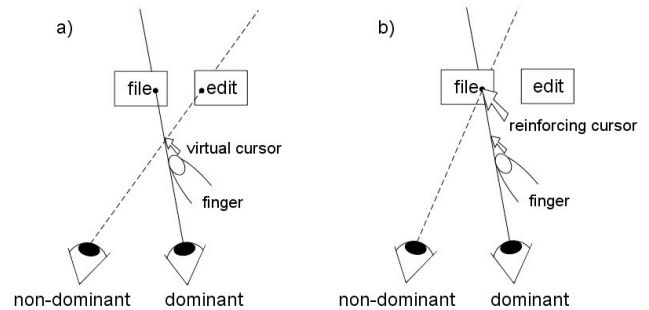


Figure 3: a) Restricting the virtual cursor to the dominant eye avoids selection ambiguity. b) A non-dominant eye cursor at content depth reinforces depth without altering the virtual cursor.

One solution to the ambiguity problem is to use a virtual cursor exclusively in the dominant eye [30]. When content has been selected, it is natural to focus at content depth and the lack of a cursor in both eyes may cause some discomfort with long-term use. When the virtual cursor is over selectable content, a reinforcing cursor can be introduced to the non-dominant eye to make the cursor appear at selection depth with no alteration of the virtual cursor. While a virtual cursor should have visual priority over all content in the scene, the reinforcing cursor can be rendered at content depth and potentially occluded by content in the non-dominant eye. Like color and shape changes, a reinforcing cursor can provide feedback that content underneath the cursor is selectable.

In a strict image-plane selection case, moving either the head position or hand position will change the content below a virtual cursor located at the hand. In a desktop situation, it is more appropriate to prevent viewpoint changes from affecting cursor selections. By moving the actual plane that the cursor moves within to the depth of content, cursor movement remains unaffected by head motion (figure 4). In order to maintain a consistent control-display relationship, an effective cursor plane can be used to adjust the gain between cursor and mouse. The gain applied to the actual cursor plane movement is the ratio of the depth of the actual cursor plane d_a to the depth of the effective cursor plane d_e both with respect to the user.

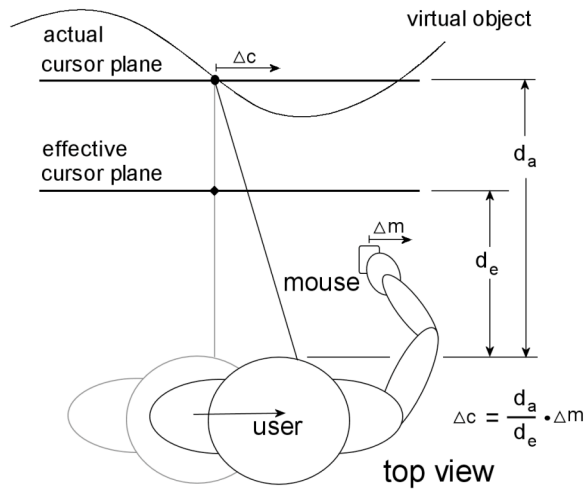


Figure 4: Modulating mouse to cursor transfer function relative to content depth simulates an effective cursor depth on the desktop.

When viewed from a fixed viewpoint, an image plane approach will have the same behavior as desktop stereo implementations that merely place the cursor at stereo content depth. However, when head tracking is added, a naive implementation will produce a non-linear or even discontinuous control-display relationship from oblique angles as the cursor moves across stereoscopic content (figure 5). Using an image-plane approach results in a linear cursor motion across content with changing depth and prevents the disappearance of the cursor behind content.

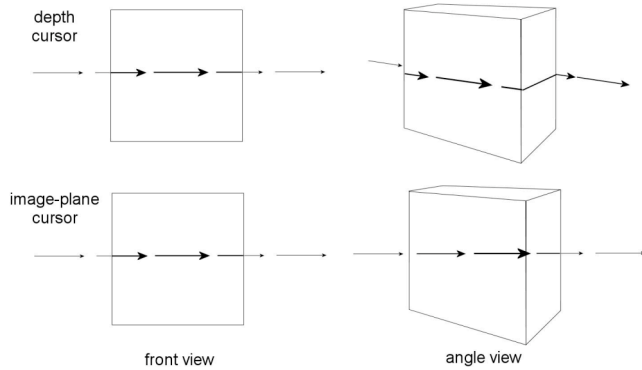


Figure 5: An image-plane implementation avoids conditions where a simple depth cursor moves non-linearly across stereo content.

3.2 Transitional Configurations

Techniques for using image-plane selection on stereo content can be applied to other transitional configurations such as large-format displays, multi-display setups and projector-based augmented reality. Not only does an image-plane approach to large-format displays facilitate selection of stereo content but it also reflects current best practices. Virtual mouse techniques, a variation of image-plane selection, have proven more accurate at a distance than ray-based techniques. A virtual mouse results from fixing the user viewpoint and using a clutching mechanism to translate the virtual cursor transfer function similar to picking up the desktop mouse.

Cursor control in multiple-display environments usually requires a display stitching technique to allow the cursor to move over a single virtual workspace. A recent technique, called perspective cursor, uses screen configurations and a tracked user position to model smooth transitions across overlapping displays [13]. When conceived of as an implementation of image-plane

selection, perspective cursor merely involves moving the actual cursor plane onto the physical surface of each display device. In this more generalized case of image-plane selection, the cursor plane remains in a plane perpendicular to the ray cast between user viewpoint and the virtual cursor. The advantage of using an image-plane perspective cursor implementation is that strict image-plane selection can be easily implemented by registering the actual and effective cursor planes with the depth of the a tracked hand. When standing to interact with a multi-display setup, registering the virtual cursor with the hand allows the user to initiate cursor actions without first acquiring its position.

Projector-based augmented reality also presents a unique opportunity for the implementation of image-plane techniques. The most frequently used interaction techniques in these environments are the standard mouse, laser-pointers and touch interaction implemented with computer vision techniques. Laser-pointer and touch based interactions can be implemented using a single projector aligned camera [14]. Strict image-plane selection can also be implemented with similar hardware by merely moving the camera to a position at the dominant eye of the user. Standard fiducial marker techniques can then be used to determine the desired position of the virtual cursor within the camera image-plane [16]. Usually this technique requires either fiducial markers in the scene or a 6 DOF tracking system to determine camera pose. A virtual mouse implementation can still be used when only the 3D position of the camera and a clear view of the finger within the camera are available (figure 6). By placing the cursor on a plane perpendicular to the ray cast between user viewpoint and the current cursor position, the cursor of a virtual mouse can be adjusted relative to hand input, projected onto the database model of the environment and then displayed via projector.

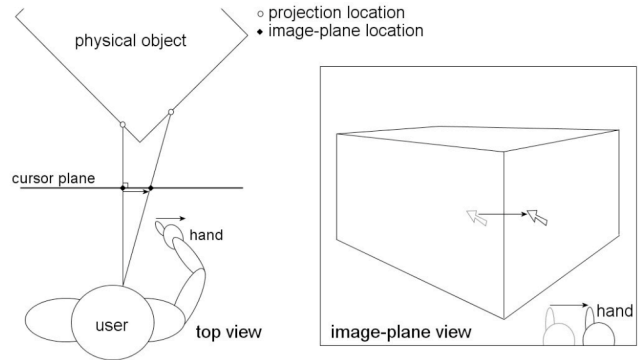


Figure 6: A virtual mouse implementation only requires camera position to control a virtual cursor in the camera image-plane.

3.3 Advantages over Ray-casting

The flexibility of image-plane selection to add and remove the elements of tracking and stereo is not shared by ray-casting. A minimum setup for a ray-casting implementation on the desktop is full 6 DOF tracking and accurate stereo display. Although ray-casting can be used in fully immersive environments without stereo, it still requires full 6 DOF tracking. In addition to these practical concerns, image-plane selection also has several theoretical advantages over ray-casting. Although both image-plane and ray-based selection techniques devolve into touch at a surface, image-plane selection has the advantage of making the transition more smoothly (figure 7). Ray-casting is more likely to cause confusion because of the rapidly changing nature of the ray intersection point. Even if the ray is emitted directly from the finger, as it approaches the physical surface, the intersection point and resulting virtual cursor will be moving unless the ray remains pointed directly at the desired touch location.

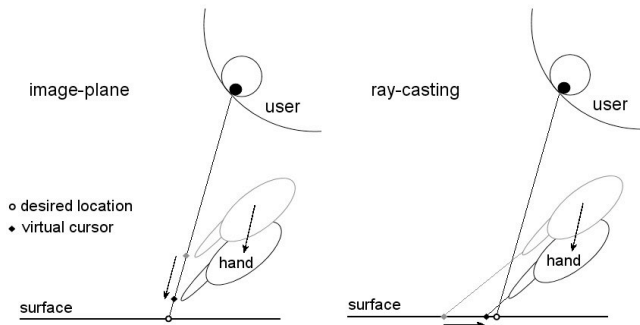


Figure 7: An image-plane selection virtual cursor remains steady as the hand is moved towards the desired location on a surface.

Both transitional and fully immersive environments introduce the likelihood that 2D interface elements will appear on a surface oriented away from the user. Even when initially oriented towards the user, a traditional cascading menu becomes more oblique as lower submenus move away from the center. The naturally isomorphic nature of occlusion-based techniques ensures that the relationship between hand motion and cursor position remains constant on oblique surfaces. In contrast, ray-casting on oblique surfaces creates a non-linear relationship between hand orientation and the resulting intersection point on the surface. Even when surfaces are viewpoint orientated, image-plane selection is also less sensitive to changes in distance. The control-display ratio of two identical interfaces presented at a different depth may appear the same to a ray-casting user but will not have the same control-display relationship (figure 8). Not only does this effect increase as interface elements move closer to the user, but the obliqueness of the angle between ray and surface also increases at reduced depth. Such effects are reduced if the hand is held closer to the line of sight, but this negates the fatigue advantage of using ray-casting and approaches image-plane selection in the limit.

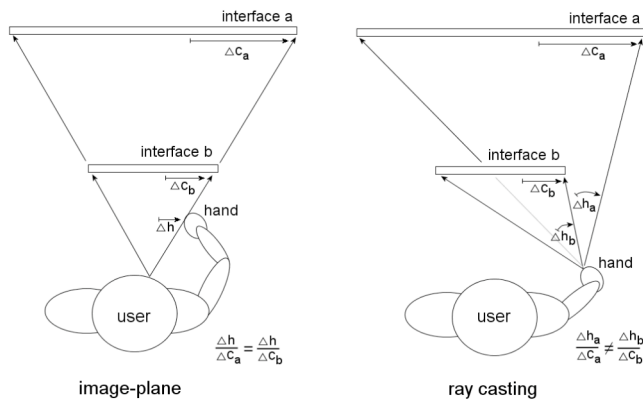


Figure 8: The control-display ratio of ray-casting is likely to change with depth while that of image-plane always remains the same.

3.4 Through-the-lens Techniques

The through-the-lens methods developed by Stoev use alternate perspectives to overcome occlusions and bring object manipulations within reach for virtual hand manipulations. Two virtual hand techniques, an eyeball-in-hand and a scene-in-hand technique, are used to manage window viewpoints. A third technique, TTL-Wim, allows the user to zoom into a rectangular region from a top view with a click-and-drag technique. The reliance on virtual hand techniques creates a number of usability problems. The first problem is that routine viewpoint and object manipulations require switching between tools that use a virtual

hand metaphor and a click-and-drag metaphor. This inconsistency leads to poor usability.

Because the virtual hand techniques require a clutching mechanism to translate the viewpoint, objects in the distance must regularly be zoomed into using the top-down TTL-Wim tool. This forces the user to reacquire targets from a top perspective and results in a poor ability to execute global search tasks. The third problem with the techniques is an inconsistent approach to solving the scaling problem. The TTL-Wim tool scales the scene within the window to bring objects on the ground plane within reach of the virtual hand. This somewhat arbitrary scaling results in an inability to reach object locations beyond a fixed distance when viewpoint orientation is adjusted with the other virtual hand tools (figure 9). A fourth problem with a virtual hand approach is that it prevents the use of parallel projection views within the window. Finally, virtual hand limits the usage of drag-and-drop between viewing windows and the surroundings. Only relatively small objects can be brought into the surrounding scene and larger objects cannot be moved into a window without first traveling to within their reach.

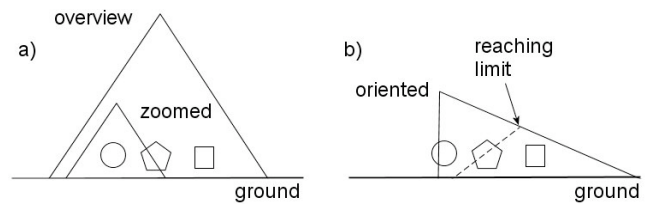


Figure 9: a) Using TTL-Wim to zoom the window viewpoint scales the scene with respect to the horizon. b) Once oriented away, some object locations in view will no longer be within arms reach.

3.5 Image-plane Selection on TTL Windows

Using image-plane selection on TTL windows resolves the problems with the original through-the-lens tools. The fundamental problem with the prior tools stems from the lack of an object-centric focus for both viewpoint management and object manipulation. As is common on the desktop, image-plane selection allows objects to be used as the focal point for orbital viewing, zoom and pan operations via click-and-drag operations over the viewing window. These desktop derived techniques solve global search problems with the TTL tools by allowing them to proceed without shifting focus away from the desired target. Familiar tools such as view-all-selected, focus-on-selected and click-and-drag selection rectangles allow distant objects to be selected, zoomed into and orbited while remaining in view. Image-plane selection does not preclude rendering window viewpoints in parallel projection. In immersive settings, display real estate can be preserved and awareness maintained by making smooth transitions between top, front and side viewpoints. The transition from perspective into parallel projection can be accomplished with a minimum of visual discontinuity by fixing the forward clipping plane at that of the current virtual camera position and moving the camera position backwards towards infinity (figure 10).

Alternate viewing windows address the subject of virtual travel by allowing destinations to be previewed before teleportation. The original TTL implementation advocated bringing the window over the user viewpoint to avoid disorientation as is done with WIM techniques. While this technique is intuitive and useful for novice users, it can prove tedious for expert users. Two alternatives that operate with less fatigue are a teleportation function on the viewing window and a teleportation to surfaces function. Some of the disorientation associated with teleportation can be reduced by teleporting to a location that merely registers the surrounding

scene with the TTL window scene. This has the effect of filling in the content around the window without changing its contents. Because not all viewpoints share the TTL window orientation, this technique is most likely to be effective when the window position and destination are out in front of the user. One problem with preview teleportation schemes is the lack of specifics about the actual position and height the teleportation will place the user at. Depth insensitive techniques such as image-plane selection facilitate selecting a distant surface within the TTL window. This allows the user to specify an exact teleportation location within the scene and ensures the user will arrive at the surface height.

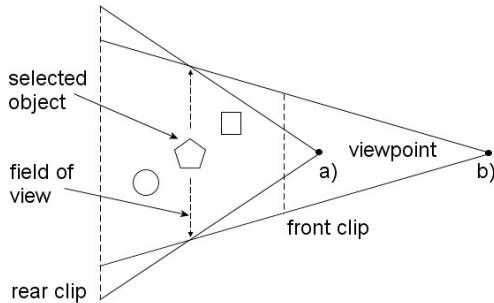


Figure 10: a) Object-centered approach establishes a reference field of view at the object. b) Fixing each clipping plane and moving the virtual camera towards infinity transitions into parallel projection.

Image-plane selection on TTL viewing windows is analogous to object manipulation techniques known as 2½D methods on the desktop. This class of techniques obviates scaling problems by using window zoom to adjust the scale of isomorphic manipulations that keep objects under the cursor at all times. These techniques also resolve the limitations of drag-and-drop actions with TTL windows. Distant objects can be moved into or out of TTL windows and moved along the ground plane or other chosen axes with a single click-and-drag action. Once an object has been relocated outside of a TTL window, executing the view-all-selected function easily re-centers the selected object within the TTL window for further manipulation.

3.6 Incorporating 6 DOF Input Devices

Although using 2½D techniques on TTL windows smoothes transitions between configurations and avoids the fatigue of using virtual hand techniques, it does not preclude the use of higher DOF devices. One way to utilize 6 DOF tracking is to adjust multiple components of zoom, pan and rotation concurrently. In his original paper on image-plane techniques, Pierce used an image-plane navigation technique that mapped hand rotation and distance from the eye to orbital position and distance to a selected object. These same principles easily translate to the manipulation of viewpoints within TTL windows and need not supersede the image-plane techniques described above. In this context a non-isomorphic mapping between hand rotation and viewpoint can actually be of benefit by overcoming the limitations of the human wrist [21].

Pierce also suggested an image-plane grab technique similar to the HOMER technique. Selected objects can be rotated in place and moved spatially around the user in an isomorphic manner. As with other depth scaling techniques, distance can be modulated by establishing a scaling between hand depth and initial object depth relative to the viewer. Image-plane grab works naturally with TTL window zooming to adjust the scaling relationship during manipulations within those windows. Instead of establishing a scaling between hand depth and object depth relative to the user, the object depth relative to the virtual camera position is used

(figure 11). An object arms length from the virtual camera that is grabbed with the hand extended will have a one-to-one depth scaling ratio. Zooming the virtual camera back to twice the distance doubles the ratio between hand depth and object depth. The result is that the range of object depth scales up as the TTL window is zoomed to reveal a larger area. This method also gives the user control over drag-and-drop scaling relationships into and out of TTL windows by using the object depth at the time of transition as the initial object depth.

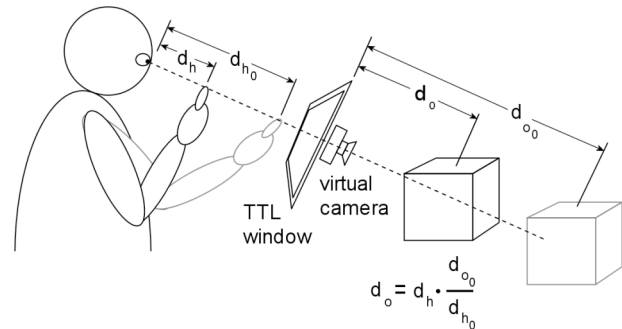


Figure 11: Manipulating the depth of TTL window objects relative to virtual camera position allows the window zoom to adjust scaling.

Viewing windows can be an aid to viewpoint management by capturing viewpoints of the surrounding scene. Registering the TTL scene with the surrounding scene makes the window frame appear transparent. The window can then be moved over content, the secondary scene locked to it, and then returned to a comfortable working position. Allowing windows to be oriented away from the user also has some benefits. Two-dimensional content within windows can be compressed along horizontal or vertical axes in order to conserve display real estate. Changing the orientation of windows registered to the surrounding scene raises questions of whether TTL windows should operate like a picture window and change their viewpoint in response to user movement. The original taxonomy proposed for TTL windows includes a class of windows that are invariant to user position [27]. One advantage of ignoring user position is that viewpoints can be captured in TTL windows with less fatigue by merely orienting them towards content like an LCD viewfinder in a “shoot from the hip” fashion.

4 A PROOF OF CONCEPT APPLICATION

An obvious application domain for the Withindows framework is in situations that require the search and manipulation of 3D space. We have chosen to build our proof of concept application in the domain of virtual world construction because it is of obvious utility to simultaneously develop and test virtual worlds in their target environment. Also of consideration in choosing a domain was an existing user community affiliated with the University of Illinois at Chicago (UIC) using the Ygdrasil authoring system. Ygdrasil, developed at the Electronic Visualization Laboratory, is an interpreted scene-graph language similar to VRML that creates distributed tele-immersive VR worlds by default [15]. The Ygdrasil software runs under SGI, Linux and Windows platforms and has primarily been used with rear-projected stereo immersion.

Because there was no existing desktop IDE, we developed interface widgets and our resulting application from first class elements of the Ygdrasil language. An advantage of this approach was that it provided an opportunity to improve the underlying language and create a rich set of interface templates for the creation of future applications within the development environment. Following the Withindows framework, we created a stencil buffered viewing window into the user scene graph and

added the typical zoom, pan, rotate, front, top, side, view-all-selected and focus-on-selected icons within the window (figure 12). We also incorporated a node hierarchy viewer that replaces the scene viewpoint and is browsed using the same click-and-drag viewpoint management techniques. We encapsulate the remaining application functionality in context sensitive menus that can be accessed by right-clicking either within the viewing window or in the surrounding environment (figure 13). Global functions for scene loading and saving are accessed by clicking on the background before right-clicking. For user convenience and to emulate a typical desktop application, we duplicated some global functionality in drop-down menus within the viewing window.

The stereo rendered application uses a dominant eye cursor at the dominant hand but does not implement a reinforcing cursor. When used in an immersive setting, a lock icon on the window frame controls the relationship between secondary scene and viewing window. When held down, this same button initiates 6 DOF viewpoint manipulations. The image-plane grab technique also appears as an option during immersive use in addition to the typical 2½D translate, scale, and orient manipulation functions.

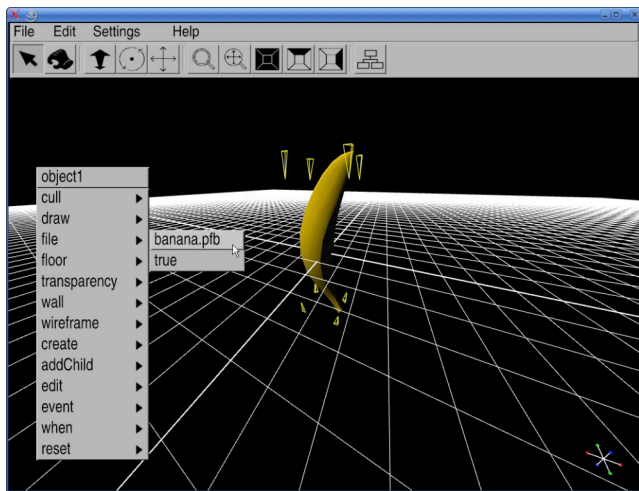


Figure 12: Ygdrasil development environment showing icons for viewpoint management, context sensitive and drop-down menus.

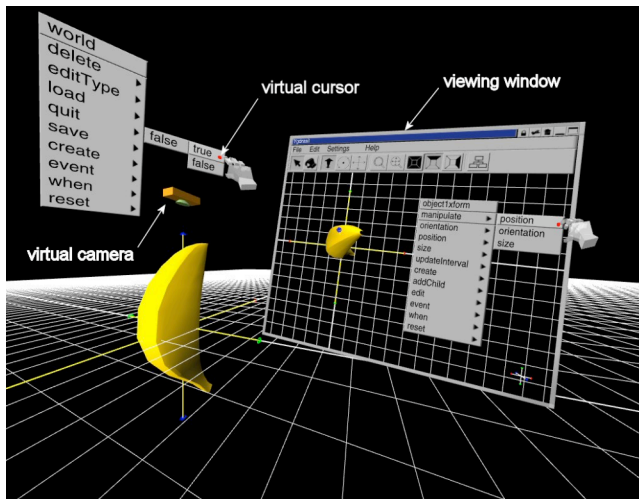


Figure 13: Immersive IDE use showing global and context sensitive menus in the environment and the viewing window respectively.

4.1 Classroom Evaluation

The new system was used for laboratory and student projects during a full semester of a long-standing class on the Ygdrasil

language. The class consisted of five students, one of which had taken the class before. The full semester of classroom use allowed us to debug the application extensively. Designing laboratory exercises around the new graphical interface helped address usability issues related to actual workflow scenarios. Equipment and time limitations restricted student use of the application to a desktop scenario.

The prior development workflow consisted of editing text files and executing them in a desktop simulator. Students have had significant problems with syntax errors in prior semesters. As expected, a graphical interface practically eliminated syntax errors. The prior scene construction workflow was replaced by one allowing students to interactively position content and assign behaviors. The interface included a function to minimize the viewing window and reveal the usual simulation environment for testing scene functionality. Although the new tool improved student productivity and was well received, there was also evidence that students did not learn the underlying language to the extent they had in previous semesters. Time spent previously on learning the details of Ygdrasil appears to have been replaced with greater time devoted to scene appearance.

4.2 Expert Evaluation

Three projects involving expert users made use of the new development environment at different stages of development. Five users with significant Ygdrasil experience participated in the evaluation. A project to develop a public installation related to Swedish folklore used the interface at all stages of development on a single wall stereo system. An ongoing project related to meditation utilized the interface during the testing and debugging phase in a three wall active stereo environment. And, the testbed environment for a user study on 3D user interaction was developed using the Ygdrasil interface both on the desktop and in stereo using a single rear projection wall.

Overall, users felt that image-plane selection worked well on cascading menus, sliders and other symbolic manipulation tasks. None of the users complained of problems using a virtual cursor in the dominant eye. One result of this evaluation was the addition of settings to adjust virtual cursor size and the transparency of the virtual hand. The most appreciated aspect of the system was the ability to simultaneously manipulate objects in the viewing window from an appropriate viewpoint while appreciating the results of the action in the surrounding environment. This feature proved especially useful for the user study project because the design called for subjects to make selections in a cluttered 3D environment from a fixed sitting position. The main complaint about the system centered on the picture window implementation of the TTL window. Having arranged an appropriate viewpoint within the window, users would often look away at the surrounding content only to find upon returning that they had to then reacquire their initial viewpoint in order to continue working. Users felt the system effectively restricted their ability to move their head, and that this subsequently led to increased fatigue.

A somewhat unexpected result of the user evaluations was the extent to which the node hierarchy view was used as a debugging tool. Users found it convenient to inspect node attributes and trigger events via the hierarchy while evaluating the scene. It became clear that the initial design produced hierarchies that were very broad and subsequently difficult to navigate. Two strategies were used to address this concern. A node was added to preemptively prune parts of the scene graph from being displayed within the hierarchy, and a node depth filter was added to the viewer. The depth filter fostered a strategy of adding proxy nodes

near the top of the scene graph with the dedicated purpose of adjusting nodes deeper in the hierarchy.

5 CONCLUSIONS AND FUTURE WORK

The Withindows framework takes a common sense approach to the primary tasks of virtual reality by eschewing first person perspective and physically intuitive interactions in favor of interactions under hand. Adopting a generalized version of the desktop interaction scheme and proven desktop viewpoint management techniques creates a canonical scheme within which both traditional applications and immersive applications can be developed once and then used across a continuum of transitional configurations. This work contributes new techniques for using image-plane selection in stereo environments and for accomplishing object manipulations and viewpoint management in through-the-lens viewing windows.

In the near future we plan to change our current picture window TTL implementation to an LCD viewfinder strategy based on rendering to textures. A benefit of this approach is that it will simplify the process of implementing transitions into parallel perspectives. We will also be adding a reinforcing cursor in order to continue our user evaluations of virtual cursors in stereo environments. We are also in the process of conducting a user study comparing image-plane selection and ray-casting on 2D surfaces below the hand and within the surrounding environment.

REFERENCES

- [1] Bolter, J., Hodges, L. F., Meyer, T. and Nichols, A., Integrating Perceptual and Symbolic Information in VR, *IEEE Computer Graphics and Applications*, v.15 n.4, p.8-11, July, 1995.
- [2] Bowman, D. A., Wineman, J., Hodges, L. F. and Allison, D., Designing Animal Habitats Within an Immersive VE, *IEEE Computer Graphics and Applications*, v.18 n.5, p. 9-13, Sept, 1998.
- [3] Bowman, D., Hodges, L., An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments. *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, p. 35-38, 1997.
- [4] Bowman, D. and Wingrave, C., Design and Evaluation of Menu Systems for Immersive Virtual Environments. *Proceedings of IEEE Virtual Reality*, Yokohama, Japan, p.149-156, Mar 13-17, 2001.
- [5] Bues, M., Blach, R., and Haselberger, F., Sensing surfaces: bringing the desktop into virtual environments. *Proceedings of the Workshop on Virtual Environments*, p. 303-304, Zurich, Switzerland, May 22-23, 2003.
- [6] Butterworth, J., Davidson, A., Hench, S., and Olano, M. T., 3DM: A Three-Dimensional Modeler Using a Head-Mounted Display, *Proc. of the Symp. on Interactive 3D Graphics*, ACM Press, New York, pp. 134-138, 1992.
- [7] Coninx, K., Van Reeth, F. and Flerackers, E., A hybrid 2D/3D user interface for Immersive Object Modeling, *Proceedings of Computer Graphics Intl.*, Computer Society Press, Belgium, p.47-55, 1997.
- [8] Feiner, S., MacIntyre, B., Haupt, M. and Solomon, E., Windows on the world: 2D windows for 3D augmented reality, *Proc. of the 6th Annual ACM Symp. on User Interface Software and Technology*, p. 145-155, Atlanta, Georgia, December, 1993.
- [9] Larimer, D. and Bowman, D., VEWL: A Framework for Building a Windowing Interface in a Virtual Environment, *Proc. of Int. Conf. on Human-Computer Interaction Interact '2003*, IOS Press, Zürich, Switzerland, p. 809-812, September, 2003.
- [10] Lee, S., Seo, J., Kim, G. J. and Park C., Evaluation of pointing techniques for ray casting selection in virtual environments, *Third International Conference on Virtual Reality and Its Application in Industry*, Zhigeng Pan, Jiaoying Shi, Editors, p. 38-44, April, 2003.
- [11] Leigh, J., Johnson, A. E., Vasilakis, C. A., DeFanti, T. A., Multi perspective collaborative design in persistent networked virtual environments, *Proceedings of the 1996 Virtual Reality Annual International Symposium (VRAIS 96)*, p.253, March-April, 1996.
- [12] Lindeman, R. W., Sibert, J. L. and Hahn, J. K., Hand-Held Windows: Towards Effective 2D Interaction in Immersive Virtual Environments, *Proc. of IEEE Virtual Reality*, p. 205-212, 1999.
- [13] Nacenta, M. A., Sallam, S., Champoux, B., Subramanian, S., Gutwin, Carl, Perspective cursor: perspective-based interaction for multi-display environments, *Proceedings of the SIGCHI Conference on Human factors in Computing Systems*, p. 289-298, 2006.
- [14] Olsen, D. R., Nielsen, T., Laser pointer interaction, *Proceedings of the SIGCHI Conference on Human factors in Computing Systems*, p.17-22, Seattle, Washington, March, 2001.
- [15] Pape, D., Anstey, J., Dolinsky, M., Dambik, E. J., Ygdrasil – a framework for composing shared virtual worlds, *Future Generation Computing Systems*, v.19 n.6, Elsevier Press, August, 2003.
- [16] Piekarski, W., Avery, B., Thomas, B. H., Malbezin, P., Integrated Head and Hand Tracking for Indoor and Outdoor Augmented Reality. *Proc. of IEEE Virtual Reality*, Chicago, IL, March 2004.
- [17] Pierce, J. S., Forsberg, A. S., Conway, M. J., Hong, S., Zeleznik, R. C. and Mine, M. R., Image plane interaction techniques in 3D immersive environments. *Proc. of the 1997 Symp. on Interactive 3D Graphics*, p.39-43, Providence, Rhode Island, April 27-30, 1997.
- [18] Pierce, J. S., Pausch, R., Comparing voodoo dolls and HOMER: exploring the importance of feedback in virtual environments, *Proceedings of the SIGCHI Conf. on Human factors in Computing Systems*, p. 105-112, Minneapolis, Minnesota, April, 2002.
- [19] Poupyrev, I., Billinghurst, M., Weghorst, S., and Ichikawa, T., The Go-Go Interaction Technique: Nonlinear Mapping for Direct Manipulation in VR, *Proc. ACM Symposium on User Interface Software and Technology*, ACM Press, NY, p. 79-80, 1996.
- [20] Poupyrev, I., Weghorst, S., Billinghurst, M. and Ichikawa, T., Egocentric object manipulation in virtual environments: empirical evaluation of interaction techniques, *Computer Graphics Forum, EUROGRAPHICS'98 issue*, 17(3): pp. 41-52, 1998.
- [21] Poupyrev, I., Weghorst, S., Fels, S., Non-isomorphic 3D rotational techniques, *Proceedings of the SIGCHI Conf. on Human Factors in Computing Systems*, p.540-547, The Netherlands, April, 2000.
- [22] Holm, R., Stauder, E., Wagner, R., Priglinger, M., Volkert, J., A Combined Immersive and Desktop Authoring Tool for Virtual Environments, *Proc. of IEEE Virtual Reality*, p.93, March, 2002.
- [23] Roussos, M., Johnson, A. E., Leigh, J., Vasilakis, C. A., Barnes, C. R. and Moher, T. G., NICE: combining constructionism, narrative and collaboration in a virtual learning environment, *ACM SIGGRAPH Computer Graphics*, v.31 n.3, p.62-63, August, 1997.
- [24] Steinicke, F., Ropinski, T., Bruder, G. and Hinrichs, K., Interscopic user interface concepts for fish tank virtual reality systems, *Proceedings of IEEE Virtual Reality*, p. 27-34, 2007.
- [25] Stenius, M., Collaborative object modeling in virtual environments, *Master Thesis*, KTH, School of Computer Science and Engineering, Royal Institute of Technology, Stockholm, Sweden, 1996.
- [26] Stokley, R., Conway, M. J. and Pausch, R., Virtual reality on a WIM: interactive worlds in miniature, *Proc. of the SIGCHI Conf. on Human Factors in Comp. Systems*, p. 265-272, May 07-11, 1995.
- [27] Stoev, S. L. and Schmalstieg, D., Application and taxonomy of through-the-lens techniques, *Proc. of the ACM Symp. on Virtual reality software and technology*, Hong Kong, China, Nov, 2002.
- [28] Stoev, S., Schmalstieg, D. and Wolfgang Straßer, W., Two-Handed Through-the-Lens-Techniques for Navigation in Virtual Environments, *Proceedings of the Eurographics Workshop on Virtual Environments*, Stuttgart, Germany, May 16-18, 2001.
- [29] Vogel, D. and Balakrishnan, R., Distant freehand pointing and clicking on very large, high resolution displays, *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*, Seattle, Washington, October, 2005.
- [30] Ware, C. and Lowther, K., Selection using a one-eyed cursor in a fish tank VR environment, *ACM Transactions on Computer-Human Interaction (TOCHI)*, v.4 n.4, Pages 309-322, December, 1997.
- [31] Wingrave, C., Tintner, R., Walker, B., Bowman, D., and Hodges, L., Exploring Individual Differences in Raybased Selection: Strategies and Traits. *Proceedings of IEEE Virtual Reality*, pp. 163-170, 2005.
- [32] Viega, J., Conway, M. J., Williams, G., and Pausch, R., 3D magic lenses. *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology*, p.51-58, Seattle, Washington, November 6-8, 1996.