

HybridSpace: Integrating 3D Freehand Input and Stereo Viewing into Traditional Desktop Applications

Natalia Bogdan¹, Tovi Grossman², George Fitzmaurice²

Autodesk Research

ABSTRACT

Technologies for 3D input and output are rapidly advancing, and are becoming more common in home and workplace environments. However, viewing a stereoscopic display can cause eye strain and fully relying on 3D spatial interactions can be fatiguing and may be less efficient for common 2D tasks. In this paper we explore the design possibilities of transitioning between 2D and 3D modalities to best support the user's current task. In a formal study, we demonstrate that a Hybrid interaction technique, that transitions between 2D display and input, to 3D, mid-task, outperforms 2D only and 3D only techniques. Guided by this result, we present HybridSpace, a proof-of-concept modeling environment that combines the benefits of 2D and 3D input and output modalities within a single coherent interface.

Keywords: Stereoscopic, 2D, 3D, Fish Tank VR, Gestures.

Index Terms: H.5.2 [Information Interfaces and Presentation]: User Interfaces - Graphical user interfaces.

1 INTRODUCTION

Modeling software is used around the world by artists, architects, engineers and hobbyists alike to create and work with three dimensional objects. The majority of such systems rely on 2D user interfaces. While 2D interfaces are familiar and comfortable, there are many tasks that would benefit from 3D control and stereo rendering [4][26].

With the recent resurgence in stereoscopic displays, the affordability and availability of 3D displays for desktop environments has grown. Similarly, there has been a recent surge in markerless hand tracking technology, making desktop spatial interactions a possibility [31][17]. As such, many 3D systems in the research literature have been developed to include free-hand spatial input [15][19][31].

However, even with all the advancements in display technology, viewing stereoscopic content for a long period of time will still cause a degree of eye strain and eye fatigue [20]. Furthermore, fully relying on 3D spatial interactions can quickly become fatiguing for users [16], and may be less efficient for tasks such as 2D UI access.

What is needed is a good way to fuse a traditional 2D user interface with a stereoscopic system that supports spatial input. Although some recent work has attempted to bring "natural" free-hand input into a desktop environment [14][15][19], only few Hybrid 2D-3D interface environments have ever been explored. In particular, we are unaware of work that explicitly explores the possible design space of ways in which users can transition between 2D and 3D for both input and output within a single seamless interface.

¹talia.bogdan@gmail.com

²{firstname}.{lastname}@autodesk.com

In this paper we contribute an exploration of transitioning between stereoscopic and non-stereoscopic content, as well as transitioning between traditional 2D input and 3D spatial input. For the stereoscopic display, we consider a desktop usage scenario, where the user sits in front of a 3D monitor, and uses shutter glasses, but no other special headgear or head tracking.

Our work is grounded by two empirical studies. The first examines the viewing experience of animating between 2D and 3D stereoscopic display modes. The second demonstrates the empirical benefits of fusing together 2D and 3D input and output. In this second study we show that a Hybrid 2D-3D technique significantly outperforms both 2D only and 3D only techniques in a selection and placement task.

Guided by the two studies, we describe the implementation of HybridSpace, a proof-of-concept modeling environment that combines the benefits of 2D and 3D input and output modalities with a single coherent interface (Figure 1). Within the HybridSpace environment, we explore various techniques for triggering the transition between 2D and 3D stereo display modes. We also explore interactions that seamlessly combine both standard 2D and 3D spatial input.

2 RELATED WORK

2.1 Bringing 3D to the Desktop

One of the early attempts at bringing 3D viewing to a desktop environment was the Fish Tank VR approach [35], where stereo and motion parallax depth cues, are included to add to the sense of depth in the environment.

Studies of such configurations have shown that the stereo display can improve depth perception and scene recognition tasks [26][33]. However, such displays cause a discrepancy in the viewer's convergence and accommodation depth cues. This discrepancy can cause asthenopia, with possible symptoms of eye strain, dizziness, or nausea [25].

There have also been a number of non-stereo 3D desktop environments implemented. Traditional 3D modeling software uses a



Figure 1: The HybridSpace system allows users to interact with standard 2D input and freehand spatial input. The system can also transition between 2D and 3D display modes.

ground-plane to aid the perspective of 3D. BumpTop [1] and Data Mountain [24] both offer new paradigms for organizing desktop content using a combination of 3D perspectives and physics-based interactions. Such approaches have the benefit of utilizing standard keyboards and mice which are familiar to users.

There have been also some interesting applications that combine immersive 3D environments with 2D displays. ErgoDesk combines a rear-projected table-sized display with 3D tracking, to support both 2D stylus input and 3D spatial interactions [9]. The system automatically transitions to a stereo rendering when users hold a 3D-tracked prop. Toucheo [14] uses a series of stereoscopic displays and mirrors to support both monoscopic and stereo viewing. The user interacts with a multi-touch surface that is behind a semi-transparent mirror. The HoloDesk [15] uses a similar display paradigm but a Microsoft Kinect is used to track the user's hands and allows them to directly manipulate a 3D scene that exists below the mirror display. While these systems use both 2D and 3D modalities, their configurations vary greatly from a traditional desktop environment and do not give access to standard devices such as a mouse and keyboard.

To that end, SpaceTop [19] also uses a see-through display but allows for traditional 2D touchscreen and keyboard input as well as 3D spatial input. However, a stereoscopic display is not used, so the transition between 2D and 3D display was not explored.

2.2 3D Desktop Input

Traditionally, 3D applications provide 3D widgets [6] to support content manipulations with 2D input devices. However, using a widget to accomplish 3D manipulations can be tedious [2], so alternative 3D input devices have been considered. A full review of such devices is beyond the scope of our work, we refer the reader to existing thorough surveys on the topic [16][37].

One notable example is the Rockin'Mouse [2] which took the form of a standard mouse but added tilt capabilities to allow the device to gain 2 more degrees of freedom. The Rockin'Mouse performed well in a 3D docking task, but the device does not support truly spatial interactions. The GlobeFish and GlobeMouse [9] allowed users to interact with their objects with full 6 DOF, by modifying a trackball to be suspended by an elastic.

Spatial freehand input [16] for 3D manipulations have also been explored, such as multi-finger gesture techniques with volumetric displays [12], and the freehand interactions of HoloDesk [15] and SpaceTop [19]. Such systems, and others, have supported the "natural" [36] interaction of pinching objects to directly grab and move them. A novel handle bar metaphor was recently used for virtual object manipulation with mid-air interaction [27].

Recent advances in tracking technology have made such freehand interaction within desktop environments feasible. Examples include the Leap Motion device [17] and 6D Hands [31]. Unfortunately, extended use of 3D input can be fatiguing for users [16]. An interesting approach is Balloon Selection [3], which supports 3D selection using a two-handed multi-finger technique, where the hands can rest on a multi-touch surface, minimizing fatigue. This modality of combining multi-touch surfaces with 3D freehand interactions in generalized into the Continuous Interaction Space [22], and has also been used in 3D modeling applications [8]. It has been shown that such indirect interactions can improve efficiency and precision for 3D interactions [17].

While 3D input can be useful for full 3D manipulations [16], studies have shown that a standard mouse is still fastest for selecting objects [22][34], even in 3D environments. For example, for stereo displays, dominant-eye only cursors have been shown to outperform true 3D techniques in Fitts' Law trials [22].

3 STUDY 1: ANIMATING BETWEEN 2D AND 3D VIEWS

In order to implement a system that can use both non-stereoscopic and stereoscopic 3D content there needs to be a method to transition between these display modes.

In our early tests, we found that an immediate cut from a non-stereo scene to a stereo scene was disruptive and uncomfortable. This is because the user requires time to adjust to a new convergence angle. Based on existing studies in Psychology and dynamic stereo displays, we would expect the time to adjust to be in the range of one to several seconds [27][32]. Thus, we propose to use a short animated transition between the two view modes, so that the eyes can continuously adjust during the transition. Animations have been previously used between perspective and orthographic views [11], but have not been explored in detail when transitioning from 2D to stereo [14]. This study serves to determine if an animation eases viewing comfort and to determine appropriate transition times.

3.1 Participants

A total of 6 participants (all male) took part in the study, aged 22-36. Participants were recruited from an internal mailing list but were not affiliated with our research group. All participants had normal stereo vision, confirmed by an online stereopsis test.

3.2 Apparatus and Implementation

For the stereo display, a 40" (1920×1080 pixel) Samsung UN40ES6500 3D TV was used. The display operated in a top-down mode for 3D thus allowing an effective resolution of 1920×540 for 3D mode using active shutter glasses, with a frame-rate of 60Hz. Users sat in a chair positioned to create a constant viewing position approximately 60 cm from the display. Users were asked to keep the viewpoint constant during the study, but no physical constraints were used.

The transition between 2D and 3D stereo was implemented by animating the location of the two OpenGL camera positions for each eye's asymmetric view frustum. A linear interpolation of positions was used for the animation.

3.3 Design

The experiment consisted of rendering a single object, and having the user assess the animation of that object between a 2D and 3D Stereo display mode.

The study had two independent variables, *Transition Direction* (2D to 3D, 3D to 2D), and *Object Position* (*in*, *at*, *out*). Transition direction refers to whether the animation is from 2D to 3D stereo, or vice versa. The *Object Position* refers to the location of the single 3D object that the participant viewed during the animation. The three object positions were: behind the screen (*in*), at the screen (*at*) and in front of the screen (*out*). The object was displayed at a position of approximately 25cm away from the screen for *in* and *out* conditions. A fully crossed design resulted in a total of 6 conditions. A Latin Square design was used to counterbalance the ordering of the 6 conditions.

3.4 Procedure

For each of the six conditions, a descending staircase design [7][21] was used to determine the fastest transition that participants judged to be comfortable. A single white wireframe sphere, with a radius of 8cm was rendered in the middle of the screen. The object was rendered for 4 seconds before the transition occurred.

After the transition, the user was prompted to answer Y or N if the transition was comfortable or not. Participants were instructed to answer N if they experienced discomfort or double vision.

The procedure started with an immediate transition (0s) and increased the transition time in discrete steps. When the user signified that the transition was comfortable or that they did not experience any double vision, the step size was reduced and the transition time decreased until the participant signified they did not like the transition time. The process continued until 6 reversals were detected. The last 3 transition times for each trial were used as the average value for that trial. Participants performed this once for each of the six conditions.

3.5 Study Results

The average transition times for each of the 6 conditions can be seen in Figure 2. A repeated measures ANOVA showed that *Object Position* was marginally significant ($F_{2,10}=3.877, p=0.057$), *Transition Direction* was not significant ($F_{1,5}=3.123, p=0.137$), and there was no significant interaction between the two variables ($F_{2,10}=2.221, p=0.159$). Although there was a lack of statistical significance, likely due to the low number of participants, the goal of this study was to isolate good transition times and not validate the differences statistically.

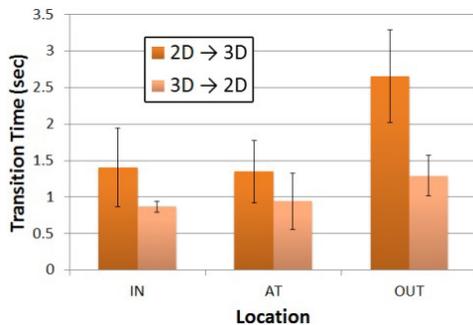


Figure 2: Average comfortable transition times for each condition. Error bars show the standard error.

From Figure 2, it can be seen that the *2D to 3D* transitions are consistently shorter than the *3D to 2D* transitions. The average transition time for a non-stereo to stereo scene was 1.801 second while the average transition time for a stereo to non-stereo scene was 1.034 second. Based on these results, the *HybridSpace* system uses a time of 1 second when transitioning from a stereo scene to a non-stereo scene and a time of 1.8 seconds when transitioning from a non-stereo scene to a stereo scene.

4 STUDY 2: WITHIN-TASK MODALITY TRANSITIONS

HybridSpace is designed to make use of the benefits of both 2D and 3D input and output. We are unaware of any studies investigating a user’s ability to transition between a fully 2D and fully 3D environment mid-task. In this study, we look at a representative task that may benefit from such a transition: object selection and placement, where a user is required to select an object, and then position it in 3D space. This is a common interaction required in 3D modelling and animation applications.

We compare our proposed technique of transitioning between 2D and 3D modalities mid-task to two baseline conditions: a fully 2D input and output environment, and a fully 3D input and output environment. Both of these configurations are used in the literature for 3D tasks, making them suitable baselines.

4.1 Participants

We recruited 12 participants to complete the study, 4 female and 8 male, 22 to 36 years old. All participants had normal stereo vision, confirmed by an online stereopsis test. Three participants were left handed but all used the mouse with their right hand.

4.2 Apparatus

The same stereo display and viewing position as the first experiment was used. An optical mouse was used for 2D input, with the default Windows 7 mouse acceleration. For 3D input, a 6-camera OptiTrack motion tracking system was used, which tracked the 3D position of two markers placed on the user’s thumb and index finger (Figure 3a). A single pinching gesture (Figure 3b) was detected by computing the Euclidean distance between the two tracked points.

4.3 Task

The task began with a start target, a white 4cm wireframe sphere, rendered at the center of the display volume (Figure 4a). This target was not part of the timed trial – it was only used to control the cursor’s initial position. Once the start object was selected, a goal target, a red wireframe sphere, was displayed (Figure 4b). The user then had to select and translate the goal target to a docking location. The docking location was rendered as a white wireframe sphere, always larger than the goal target to allow for some error tolerance (Figure 4c). The docking location was not displayed until the goal target was selected, so that users wouldn’t confuse the target and docking locations. When the goal target entered the docking location, the goal target turned green. A trial was completed once the goal target was successfully released within the bounds of the docking location.

The size of the goal target varied, as did the required docking accuracy. The size was either *Large* (2.0cm) or *Small* (1.2cm). The accuracy determined the amount of error allowed during the docking portion of the task, which was visually represented by the radius difference between the goal and docking targets. The two accuracies tested were *High* (2.0cm) and *Low* (4.0cm).

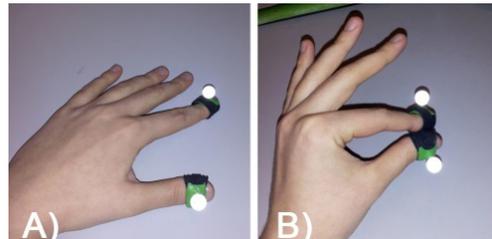


Figure 3: a) Finger tracking markers. b) The pinch gesture.

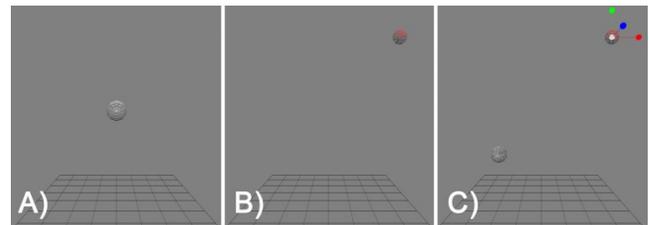


Figure 4: a) The start target began a trial. b) The user then selected a goal target. c) Translating the goal to a docking position.

Similar to previous docking experiments [2], there were 8 possible goal target and docking target locations. The goal target could be at any of the 8 corners of a 40cm cube, centered at the display center. The docking location was always at the opposite corner to the goal target.

4.4 Techniques

4.4.1 2D Only Technique

In the *2D Only* technique the scene was displayed without stereoscopic 3D, and behaved similarly to traditional desktop 3D model-

ing applications. The user performed object selection with a standard cursor and 2D mouse. Once the goal target was selected, a traditional translation widget was displayed (Figure 5a). The widget allowed the user to translate the object along any of the three major axes by dragging the appropriate handle. The user could also drag a central widget to translate the object with two simultaneous degrees-of-freedom, in the plane of the camera. A 3D ground plane was provided, but additional depth cues such as shadows were not provided, to keep this condition similar to traditional desktop 3D modeling applications.

The user also had the option to orbit the scene by dragging with the right mouse button depressed. The user could not rotate the scene until the target object was selected.

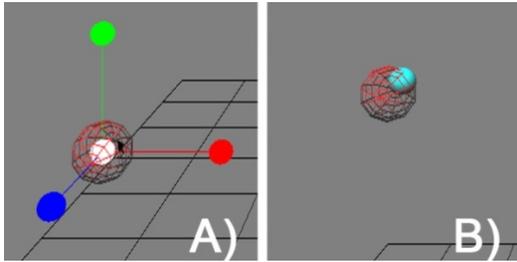


Figure 5: a) The translation widget used in the 2D Only technique. b) The 3D cursor used in the 3D Only condition.

To maintain consistent resolution and brightness across conditions, the user was required to wear the stereo glasses and the scene was rendered with two cameras but their separation was zero.

4.4.2 3D Only Technique

In the 3D ONLY mode the scene was displayed in stereoscopic 3D. For input, the user placed the OptiTrack markers on their dominant hand.

For selection, a small blue sphere was used as the cursor (Figure 5b). Its position was controlled by the 3D location of the midpoint between the index finger and thumb. An indirect absolute mapping was used, which mapped the area in front of the user to the environment space, with a control gain ratio of approximately 1.125 for the X and Y axes and 0.75 for the Z axis. This mapping was chosen as it provided comfortable access to all potential target locations in the study environment. The cursor color changed when a pinch was detected and when it entered the goal target.

To perform a selection, the user pinched the fingers when the cursor was inside of the goal target location. While still pinching the user could translate the object to the docking position. For visual consistency, the translation widget was shown during the translation, although it was not interactive. The translation was completed when the pinch was released. Scene rotations were not required to accomplish the task.

4.4.3 Hybrid Technique

For the Hybrid technique both the mouse and the OptiTrack system were used. The participant used the mouse with the right hand and wore the tracking markers on the left hand. We kept this hand assignment constant since the majority of users, and all of our participants, used the mouse with their right hand.

At the start of each trial, the scene was rendered in the 2D display mode. Users selected the start target and goal target with the mouse. Once the goal target was selected the system transitioned to the 3D display mode with an animation lasting 1.8sec, as guided by our first experiment.

As soon as the goal target was selected, the user could translate the object with a spatial pinch and drag action. Similar to the 3D

Only technique, the translation widget was shown for visual feedback only. With this technique, the freehand translation mapping was relative – the user could pinch anywhere to start the translation, since the target had already been selected by the 2D cursor. The translation could begin before the animated transition to the stereo display mode was completed. Scene rotations were also disabled for this technique.

4.5 Design

The main independent variables for the study were *Technique* (2D Only, 3D Only, Hybrid), *Size* (Small, Large), and *Accuracy* (High, Low). Users completed all conditions for one technique before moving to the next. For each technique, the user completed four blocks of 16 trials. Within each block, trials for the four combinations of *Size* and *Accuracy* were completed in random order. Within each of these *Size-Accuracy* combinations, there were four repetitions, with each occurring at a different goal and docking location. Between Blocks 1 and 2, and Blocks 3 and 4, all 8 locations were used for each *Size-Accuracy* combination. For each technique, the user completed a practice block of 16 trials.

The main dependent variable was the *trial completion time*, defined as the time between selecting the start target and successfully releasing the goal target within the boundaries of the docking location. This time was further divided into the *selection time*, defined as the time between selecting the start target and selecting the goal target, and the *manipulation time*, defined as the time taken to complete the trial once the goal target had been selected.

4.6 Results

4.6.1 Trial Completion Time

The average trial completion times can be seen in Figure 6. *Technique* had a significant effect ($F_{2,22}=69.991, p<0.05$), with overall completion times of 5.06 for Hybrid, 6.69sec for 3D Only, and 13.49sec for 2D Only. Post-hoc pairwise comparison using Bonferroni correction showed that all differences were significant ($p<0.05$). This result is encouraging, with the Hybrid techniques being 24.4% and 62.5% faster than 3D Only and 2D Only respectively. We found a significant effect of block number ($F_{3,33}=10.464, p<0.05$) but no significant interactions between block and *Technique*.

The *Size* ($F_{1,11}=5.816, p<0.05$) and *Accuracy* ($F_{1,11}=46.141, p<0.05$) both had a significant effect. There were significant interactions between *Technique* and *Size* ($F_{2,22}=8.264, p<0.05$) and *Technique* and *Accuracy* ($F_{2,22}=7.623, p<0.05$). Changes in *Size* had the largest impact on 3D Only (Figure 6a), while changes in *Accuracy* had the largest impact on 2D Only (Figure 6b).

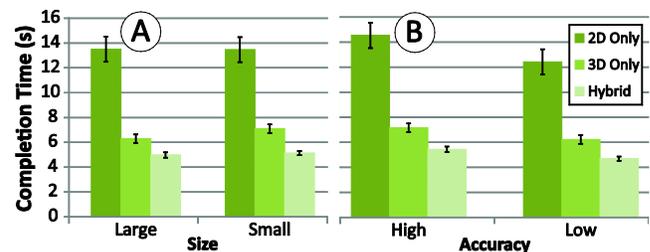


Figure 6: a) Average task completion time by size (b) Average task completion time by accuracy.

4.6.2 Subtask Times

The average *selection times* can be seen in Figure 7a. We found a significant effect of *Technique* ($F_{2,22}=170.278, p<0.05$), with times of 1.14sec for Hybrid, 3.22sec for 3D Only, and 1.13sec for 2D Only. As expected, the selection times for the 3D Only tech-

nique were higher ($p < 0.05$), due to the added dimensional constraint [13][22]. The *Hybrid* and *2D Only* selection times are similar, since the techniques are equivalent during the selection stage. There was a significant effect of *Size* on selection time ($F_{1,11} = 204.856$, $p < 0.05$) but not *Accuracy*, since it did not impact the selection subtask.

The average *manipulation times* can be seen in Figure 7b. We found a significant effect of *Technique* e ($F_{2,22} = 92.459$, $p < 0.05$), with times of 3.90sec for *Hybrid*, 3.23sec for *3D Only*, and 12.27sec for *2D Only*. In this case, *Hybrid* was similar to *3D Only*, but *2D Only* took significantly longer ($p < 0.05$), due to the difficulties of performing the 3D docking task with 2D widgets and without stereoscopic depth cues. There was a significant effect of *Accuracy* on the manipulation time ($F_{1,11} = 49.554$, $p < 0.05$), but not *Size*, since it did not impact this phase of the task.

This data shows the limitations of relying solely on either 2D or 3D techniques, while our *Hybrid* technique is robust across both phases of the task, providing the best overall performance. It is interesting to see that *Hybrid* was slower than *3D Only*, despite both using freehand translations. This indicates that there is an overhead (0.67sec) in transitioning between modalities mid-task.

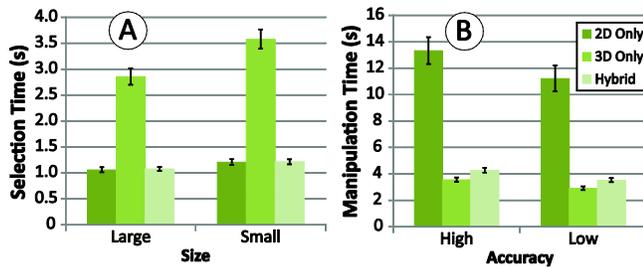


Figure 7: a) Average selection time by Size. b) Average manipulation time by Accuracy.

5 HYBRIDSPACE

As we have discussed and demonstrated, there are certain 3D tasks that are better accomplished using 3D technologies while for others it is better to use traditional 2D modes of input and output. Because 3D applications typically reside solely on either 2D or 3D platforms, but not both, users do not have access to the most efficient methods for all of the potential 3D related tasks they may perform. In this section, we present *HybridSpace*, a 3D modeling application that runs on a *Hybrid* 2D-3D platform. The system only provides simple modeling functions, but allows us to explore the possibilities of merging 2D and 3D interactions, including both input and output mechanisms.

5.1 Design Goals

5.1.1 Dual-Mode Input and Output

We want to support viewing and interaction in both 2D and 3D to take advantage of the benefits of each while avoiding the costs as much as possible, such as the eye strain related to the extended use of 3D viewing and arm fatigue related to the extended use of spatial input.

5.1.2 Implicit and Explicit Transitions

Users are unlikely to switch back and forth between modalities if there is a large transaction cost to doing so. As such we wish to explore low-cost methods for transitioning between 2D and 3D, both manually and automatically.

5.1.3 Avoid the Parallax Problem

With stereo displays, discrepancies in depth cues can occur when 2D UI elements, such as cursors, menu bars, or tool palettes, oc-

clude 3D geometry that are perceived in front of the display plane. A similar effect can also occur if a user's hand occludes geometry that is rendered in front of the display [29]. This *parallax problem* [29] generates incorrect visual information and can cause visual discomfort.

5.2 System Implementation

The system configuration of *HybridSpace* is illustrated in Figure 8. The hardware implementation was equivalent to the *Hybrid* technique in the second study. The system used the same 40" 3D TV, and the OptiTrack motion tracking system for tracking the user's left hand. A USB optical mouse was used for 2D input.

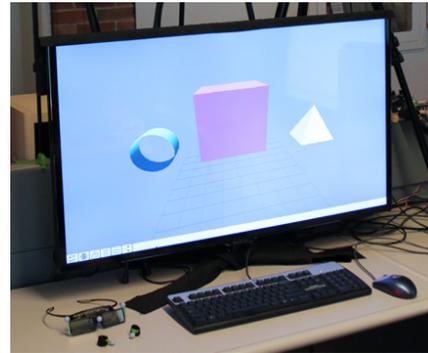


Figure 8: The *HybridSpace* hardware configuration.

We also tested implementations using the Leap Motion and 6D hands markerless hand tracking systems. However, both of these technologies did not quite provide the desirable level of position tracking or pinch detection. Our use of marker-based tracking should thus be seen as enabling technology only, and we would anticipate the use of markerless solutions as these technologies improve.

HybridSpace is written in C++ and makes use of OpenGL and FreeGLUT for graphics and window management. The NatNet SDK is used to interact with the Natural Point Tracking Tools software that streams live marker data.

5.3 HybridSpace User Interface

The system user interface consists of a main 3D workspace a 2D menu bar at the bottom of the display, and a cursor for object and UI interactions (Figure 9).

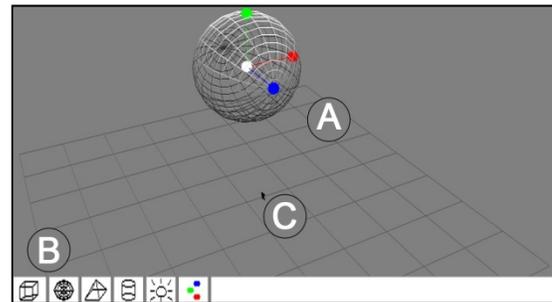


Figure 9: The *HybridSpace* User Interface. a) The 3D Workspace. b) The 2D Menu Bar. c) The 2D Cursor.

The majority of the interactions occur within the 3D workspace (Figure 9a). A 3D ground plane is used to give the user a sense of global position and to add depth cues. The 3D workspace can be rendered in either 2D (*2D Mode*) or in 3D stereo (*3D Mode*), depending on the current state of the system. The scene can be globally rotated by clicking and dragging with the right mouse button.

The 2D menu bar allows the user to add objects to the scene, and also access the color and lighting features (Figure 9b). The menu bar is only displayed in *2D Mode* – if the system transitions to *3D Mode* then the menu fades out to prevent the parallax problem. A standard mouse cursor is used to select 3D objects in the scene, and interact with the 2D menus and its associated widgets (Figure 9c). When the system transitions to *3D Mode* the mouse cursor fades out to prevent the parallax problem.

5.4 Object Manipulations

5.4.1 Object Creation

Objects can be added to the scene by clicking on the associated icon with the 2D Menu Bar (Figure 9b). The available objects are a sphere, cube, pyramid, and cylinder. When an object is added to the scene it is set as the default selected object and added to the center of the scene.

5.4.2 Object Translation

In *2D Mode*, a traditional 3D translation widget is displayed for the selected object (Figure 5a). The widget behaves as described in the *2D Only* technique of our second study.

In *3D Mode*, the translation widget is also displayed for the selected object, but only for visual reference. To translate the object, the user pinches their fingers in the free space in front of the monitor. The user can then move their hand in free space to control the 3D position of the selected object using a relative mapping (Figure 1). Releasing the pinch will end the translation. If necessary, the user can clutch by releasing the pinch, repositioning the hand, and pinching again to continue.

By supporting both mouse-based and freehand object translations, users can leverage both configurations when it is most beneficial. Coarse 3D placement can be efficiently performed with freehand input, and mouse input can be used for detailed axis-constrained movements.

5.5 Object Selection

Based on the results of our second study, our system only supports selection in *2D Mode*. In this mode, object selection is accomplished with the 2D cursor controlled by the mouse. With this implementation, users take advantage of the hybrid selection-placement interaction, as demonstrated in our second study.

5.6 Lighting Placement

An additional tool on the 2D menu is used for positioning the lighting source for the 3D scene. When the light is selected from the 2D menu bar the system displays a translation widget representing the light source.

In *2D Mode*, the light widget controls are similar in 2D to the movement controls of regular objects. Movement is accomplished by using the handles on the translation widget. In *3D Mode* the light widget works in the same way as positioning an object in 3D. The user pinches and moves their hand to translate the light widget the same as translating a scene object. The scene lighting updates in real time as the position of the light source is moved.

5.7 Color Selection: A Dual Mode Widget

Users can access a color widget by clicking the color icon in the 2D menu bar. This displays a Hue Saturation Value (HSV) color widget that is used to assign a color to the currently selected object. The color widget is an example of a *dual mode widget* which automatically adapts to the user's current input paradigm.

In *2D Mode*, a circle ranges in hue and value, and a slider controls saturation (Figure 10a). The selected color is displayed in a preview box shown beside the widget.

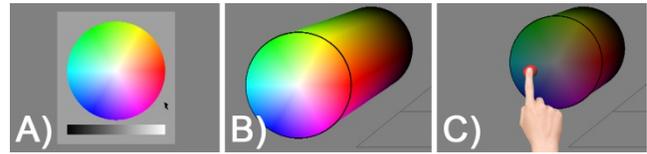


Figure 10: a) The 2D color palette. b) The full 3D color palette. c) An illustration of the cutaway behavior for the 3D color palette during direct interaction.

In *3D Mode*, the color widget transforms into a cylinder extending out of the display surface. The top slice of the cylinder is identical to the 2D color widget, providing access to the hue and value parameters. The height of the cylinder is mapped to the saturation value, progressing to black towards the bottom (Figure 10b). In this mode, the widget supports direct spatial interaction to allow the user to select all three of the HSV parameters in parallel. The user can interact with this widget by moving their fingers towards the cylinder. As the user moves towards the bottom, the cylinder is cut away, to prevent the parallax problem caused by the user's hand (Figure 10c). Pinching confirms the selected color.

5.8 Transitioning between 2D and 3D Mode

The modeling features described above were intentionally kept simple as the primary purpose of developing *HybridSpace* is to explore the techniques and triggers for transitioning between a 2D and 3D Stereo workspace. We explored manual and automatic triggers. In all cases the transition between *2D Mode* and *3D Mode* is the same animation as described in our second study for the *Hybrid* technique. The animation from 2D to 3D took 1.8 seconds while the transition from 3D back to 2D took 1 second. A transition can also reverse mid-way through if necessary.

5.8.1 Manual Triggers

Manual triggers are explicitly executed by the user. They can be useful if a user wants to spend an extended amount of time in a single viewing mode. They may also be useful if the user wants to override any of the automatic triggers that will be discussed later.

Keyboard Shortcut: The most straightforward method for transitioning between viewing modes is for the user to explicitly toggle the display using a keyboard shortcut ('z' in our implementation). This shortcut has a much lower transaction cost than modifying the 3D display settings of a 3D TV or monitor.

Flip-Up Stereo Glasses: The other explicit transition we explored is to allow users to flip their stereo glasses up and down (Figure 11). This is beneficial to a user who is going to be temporarily working in a 2D display mode, since they may not want to view the display through the glasses. The Flip-Up Stereo Glasses were built by mounting the lenses of the stereo glasses onto a custom built frame with a snap hinge. We placed three reflective markers on the stereo glasses so their orientation could be tracked by the OptiTrack system. If the orientation crossed a preset threshold, the display mode would transition appropriately. The Flip-Up Stereo Glasses were not used during the two evaluation studies mentioned earlier.

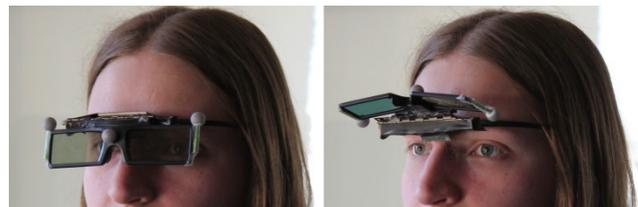


Figure 11: The Flip-Up Stereo Glasses.

5.8.2 Automatic Input Device Triggers

Manual triggers can become cumbersome if a user wants to continuously transition modes. In this section we describe automatic transitions triggered by the input device.

Freehand movement: If the system is in *2D Mode* and freehand motion is detected, the system transitions to *3D Mode*. We trigger the transition if the hand position is detected to be more than 15cm above the surface of the desk, above the keyboard area. If the hand returns below the height threshold while not pinched, the system returns to *2D Mode* after a timeout period of 3.5 seconds.

Mouse movement: Once the system is in *3D Mode*, we hide the menu and cursor to avoid the false occlusion problem. To support efficient access to menu items and object selection, the system animates back to *2D Mode* as soon as the cursor moves. If the mouse stops moving for a period of 3.5 seconds, and the hand remains above the height threshold, the system transitions back to *3D Mode*.

Spring Loaded Mouse Clicks: If the user's hand remains in mid-air during a mouse movement, the system will immediately animate back into *3D Mode* if an object or the 2D menu is clicked. This allows users to rapidly select an object in 2D with the mouse, and then position it in 3D with freehand movements.

5.8.3 Automatic Operation Triggers

In addition to matching the viewing mode to the current input device, we also include two techniques for transitioning to *3D Mode* based on the current operation.

Rotation Trigger: When the user initiates a scene rotation, *HybridSpace* transitions to *3D Mode* (Figure 12). A rotation is often used to get a better view or understanding of the scene, so a stereo view could be beneficial [35]. The system displays the scene in *3D Mode* for the duration of the rotation, and then animates back to *2D Mode*.

Translation Trigger: The system also animates to *3D Mode* during *2D Mode* object translations. This transition only occurs if the user selects a handle that has a trajectory within 30° of the view plane normal vector (Figure 13). The rationale for this transition is that the object is moving in or out of the screen, so additional depth information could be beneficial [30]. Positioning an object in XY space would not benefit from a stereo viewing mode, so we do not transition when this type of translation is occurring.



Figure 12: A scene rotation triggers a transition to 3D Stereo. The system returns to 2D when the operation is completed.



Figure 13: A translation along the z axis ($\pm 30^\circ$) triggers a transition to 3D Stereo. The system returns to 2D when after the operation.

5.9 Initial Usage Observations

We have not performed a formal user study of *HybridSpace*, but have had several internal lab members test out the system to gain preliminary insights and observations. In general users have found the system compelling and welcome the idea of not having to fully rely on freehand gestures and stereo viewing. The mixed modality interactions seemed easy for users to learn.

Of particular relevance is that we found that the transitions between display modes were not disruptive. Users performed a variety of tasks during which the display would transition between 2D and 3D stereo. Regardless of frequency or speed the users felt comfortable with how the transitions were occurring, and in fact, often didn't even realize the transition had occurred. This indicates that the animation times we used, based on the results of our first study, successfully provided comfortable transitions.

6 DISCUSSION

With *HybridSpace* we have explored the transition between 2D and 3D modalities. Our first study examined the timing of such a transition, and our second study demonstrated that mixing 2D and 3D modalities within a task can have significant benefits.

In the second study we looked at one example task, however, this only serves as an example of many possibilities for which 2D and 3D modalities can be efficiently combined.

Furthermore, we compared our hybrid technique to a fully 2D input and output environment and fully 3D input and output environment. This was sufficient to validate the benefit of a mixed-modality environment. However, it would be interesting in the future to isolate the individual input and output factors, to further understand the benefits of the Hybrid technique.

We also only compared the *Hybrid* technique to a *3D Only* technique that used 3D cursor selection. Other selection techniques, such as the ray cursor [5], could be more efficient in a 3D only modality. However, ray casting causes problems due to jitter when acquiring distant objects [5], and diverges from the "natural" freehand input paradigm seen in recent systems [15][19][31].

In addition, we did not use head tracking, for either the *3D ONLY* or *Hybrid* techniques, to simplify our study implementation. User's viewing position remained steady in both studies, so this was unlikely to be a factor. However, it is known that head tracking does have the potential to improve performance in 3D environments [35], so this can be viewed as a limitation of our current work. It would be interesting to explore how head tracking impacts the hybrid technique in the future.

With the variety of triggers used to activate the *3D Mode* in *HybridSpace*, we were initially concerned that frequent transitions could be disruptive to a user's workflow. However, we found that after calibrating the transition times based on the first study, the transitions were quite appealing and hardly noticeable. For example, during our second study, users often didn't realize that the system was transitioning during the *Hybrid* technique trials until the experimenter pointed it out. Similarly, during our initial usage observations, the automatic triggers often went unnoticed, and users would find themselves viewing in stereo, without realizing when the transition occurred.

7 FUTURE WORK

Based on the results from our first study, we implemented fixed animation times between 2D and 3D stereo based on the direction of the transition. It would be interesting in the future to support dynamic animation times based on the scene geometry, similar to dynamic stereo displays [32].

Making use of a markerless tracking system would also be desirable. Our implementations with the Leap Motion and 6D Hands

Toolkits indicate that these technologies still require further improvements before they can be reliably used for the types of interactions supported in our system.

Our system only supported basic modeling operations since it was not the focus of our contributions. However, it would be interesting to consider how additional techniques, such as object scaling and object rotation, could benefit from our dual-modality interaction paradigm.

One interesting issue that we did not directly address was the possibility of falsely triggering a transition to 3D due to unintentional freehand movements. Our time threshold seemed to work well, but this could possibly be combined with specific gestures above the keyboard that would be required to activate the 3D mode of *HybridSpace*.

Finally, it would be interesting in the future to perform more rigorous user studies to gain feedback on the *HybridSpace* modeling environment. Our system contains a number of functionalities that were not formally studied, and it would be useful to perform thorough quantitative evaluations [17][27] on these features, such as the manual and automatic triggers, colour selection, and lighting placement.

8 CONCLUSIONS

Our work explores the novel idea of rapidly switching between a 2D and 3D environment within a single workflow or task. The current work demonstrates that there are potential benefits from transitioning between the two modalities. *HybridSpace* contributes several novel techniques and design opportunities, including a series of novel methods for triggering environment transitions. Overall, we believe that the first steps in implementing *HybridSpace* have proved successful and that it can serve as a useful reference for future designs of user interaction and display techniques as 3D technologies advance.

REFERENCES

- [1] Agarawala, A. and Balakrishnan, R. (2006). Keepin' it real: pushing the desktop metaphor with physics, piles and the pen. *ACM CHI*, 1283 – 1292.
- [2] Balakrishnan, R., Baudel, T., Kurtenbach, G., and Fitzmaurice, G. (1997). The Rockin'Mouse: integral 3D manipulation on a plane. *ACM CHI*, 311 – 318.
- [3] Benko, H., and Feiner, S. (2007). Balloon selection: A multi-finger technique for accurate low-fatigue 3d selection. *IEEE 3DUI*, 79-86.
- [4] Boritz, J., (1999) The Effectiveness of Three-Dimensional Interaction. PhD thesis, UBC, Canada.
- [5] Bowman, D. A., and Hodges, L. F., (2005) Exploring individual differences in raybased selection; strategies and traits, *VR*, 163-170.
- [6] Conner, B. D., Snibbe, S. S., Herndon, K. P., Robbins, D. C., Zeleznik, R. C., and van Dam, A. (1993) Three-dimensional widgets. *ACM SIG3D*, 183-188.
- [7] Cornsweet, T. N. The Staircase-Method in Psychophysics. *Am. J. Psych.*, 75(3), 485 – 491.
- [8] De Araújo, B. R., Casiez, G., Jorge, J. A., & Hachet, M. (2013). Mockup Builder: 3D modeling on and above the surface. *Computers & Graphics*.
- [9] Forsberg, A. S., LaViola Jr, J. J., & Zeleznik, R. C. (1998). ErgoDesk: a framework for two-and three-dimensional interaction at the ActiveDesk. *Immersive Projection Technology Workshop*.
- [10] Froehlich, B., Hochstrate, J., Skuk, V., and Huckauf, A. (2006). The GlobeFish and the GlobeMouse: two new six degree of freedom input devices for graphics applications. *ACM CHI*, 191 – 199.
- [11] Grossman, T., Balakrishnan, R., Kurtenbach, G., Fitzmaurice, G., Khan, A., and Buxton, B. (2002) Creating principal 3D curves with digital tape drawing. *ACM CHI*, 121-128.
- [12] Grossman, T., Wigdor, D., and Balakrishnan, R. (2004) Multi-finger gestural interaction with 3d volumetric displays. *ACM UIST*, 61-70.
- [13] Grossman, T., and Balakrishnan, R. (2004). Pointing at trivariate targets in 3D environments. *ACM CHI*, 447-454.
- [14] Hachet, M., Bossavit, B., Cohé, A., Rivière, J.B. (2011). Toucheo: Multitouch and Stereo Combined in a Seamless Workspace. *ACM UIST*, 587 – 592.
- [15] Hilliges, O., Kim, D., Izadi, S., Weiss, M., Wilson, A. (2012) HoloDesk: direct 3d interactions with a situated see-through display, *ACM CHI*, 2421 – 2430.
- [16] Hinckley, K., Pausch, R., Goble, J., and Kassell, N. (1994). A survey of design issues in spatial input. *ACM UIST*, 213 – 222.
- [17] Knoedel, S., & Hachet, M. (2011, March). Multi-touch RST in 2D and 3D spaces: Studying the impact of directness on user performance. *IEEE 3DUI*, 75-78.
- [18] Leap Motion. 2013. <https://www.leapmotion.com>.
- [19] Lee, J., Olwal, A., Ishii, H., and Boulanger, C. (2013). SpaceTop: Integrating 2D and Spatial 3D Interaction in See-through Desktop Environment. *ACM CHI*, 189-192.
- [20] Leroy, L., Fuchs, P., and Moreau, G. (2012). Real-time adaptive blur for reducing eye strain in stereoscopic displays. *ACM Trans. Appl. Percept.* 9(2), 18 pages.
- [21] Levitt, H. (1971), Transformed up-down methods in psychoacoustics. *J. Acoust. Soc. Am.*, 49(2), 467 – 477.
- [22] Marquardt, N., Jota, R., Greenberg, S., & Jorge, J. A. (2011). The continuous interaction space: interaction techniques unifying touch and gesture on and above a digital surface. *Interact* (461-476).
- [23] R. Teather, W. Stuerzlinger, (2011). Pointing at 3D Targets in a Stereo Head-Tracked Virtual Environment, *IEEE 3DUI*, 87 – 94.
- [24] Robertson, G., Czerwinski, M., Larson, K., Robbins, D., Thiel, D., and Dantzich, M. (1998). Data mountain: using spatial memory for document management. *ACM UIST*, 153 – 162.
- [25] Shibata, T., Kawai, T., Otsuki, M., Miyake, N. Yoshihara, Y., Iwasaki, T. (2005). Stereoscopic 3D display with dynamic optical correction for recovering from asthenopia. *Stereoscopic Displays and Virtual Reality Systems XII*, 9 pages.
- [26] Sollenberger, R. and Milgram, P. (1993). Effects of stereoscopic and rotational displays in a three-dimensional path-tracing task. *Human Factors*, 35(3). 483-499.
- [27] Song, P, Wooi Boon Goh, William Hutama, Chi-Wing Fu, and Xiaopei Liu. (2012). A handle bar metaphor for virtual object manipulation with mid-air interaction. *ACM CHI*, 1297-1306.
- [28] Suryakumar, R. and Bobier, W. R. Gain and Movement Time of Convergence-Accommodation in Preschool Children. *Optom. Vis. Sci.*, 81(11), 835 – 843.
- [29] Valkov, D., Giesler, A., Hinrichs, K. H. (2012) Evaluation of Depth Perception for Touch Interaction with Stereoscopic Rendered Objects, *ACM ITS*, 21 – 30
- [30] Wang, G., McGuffin, M. J., Bérard, F., and Cooperstock, J. R. (2011) Pop-up depth views for improving 3D target acquisition, *GI*, 41-48.
- [31] Wang, R. Y., Paris, S., Popovic, J., (2011) 6D hands: markerless hand-tracking for computer aided design. *ACM UIST*, 549 – 558
- [32] Ware, C. (1995). Dynamic stereo displays. *ACM CHI*, 310-316
- [33] Ware, C. and Franck, G. (1996). Evaluating stereo and motion cues for visualizing information nets in three dimensions. *ACM Trans. on Graphics*, 15(2). p. 121-140
- [34] Ware, C. and Lowther, K. (1997) Selection using a one-eyed cursor in a fish tank VR environment, *ACM TOCHI*, 309-322.
- [35] Ware, C., Arthur, K., and Booth, K. (1993). Fish tank virtual reality. *ACM CHI*, 37 – 42.
- [36] Wilson, A. (2006). Robust computer vision-based detection of pinching for one and two-handed gesture input. *ACM UIST*, 255-258.
- [37] Zhai, S. Human performance in six degree of freedom input control. *University of Toronto*, 1995.