Hands-Free Multi-Scale Navigation in Virtual Environments

Joseph J. LaViola Jr.   Daniel Acevedo Feliz   Daniel F. Keefe   Robert C. Zeleznik

Brown University
Department of Computer Science, Box 1910
Providence, RI 02912
{jjl, daf, dfk, bcz}@cs.brown.edu

Abstract

This paper presents a set of interaction techniques for hands-free multi-scale navigation through virtual environments. We believe that hands-free navigation, unlike the majority of navigation techniques based on hand motions, has the greatest potential for maximizing the interactivity of virtual environments since navigation modes are offloaded from modal hand gestures to more direct motions of the feet and torso. Not only are the users' hands freed to perform tasks such as modeling, notetaking and object manipulation, but we also believe that foot and torso movements may inherently be more natural for some navigation tasks. The particular interactions that we developed include a leaning technique for moving small and medium distances, a foot-gesture controlled Step WIM that acts as a floor map for moving larger distances, and a viewing technique that enables a user to view a full 360 degrees in only a three-walled semi-immersive environment by subtly amplifying the mapping between their torso rotation and the virtual world. We formatively designed and evaluated our techniques in existing projects related to archaeological reconstructions, free-form modeling, and interior design. In each case, our informal observations have indicated that motions such as walking and leaning are both appropriate for navigation and are effective in cognitively simplifying complex virtual environment interactions since functionality is more evenly distributed across the body.

CR Categories and Subject Descriptors: I.3.6 [Computer Graphics]: Methodology and Techniques - Interaction Techniques; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism - Virtual Reality
Additional Key Words: Navigation techniques, Auto Rotation, Gestural Interaction, Virtual Reality

1 Introduction

Each year computational power inexorably increases and virtual environments become richer as increasingly realistic displays of virtual scenes become viable. Despite this continuing progress in visual realism, the ability of a user to engage in rich interactions in a virtual environment fails to follow such a steady advance. In fact, it can be argued that the quality of user interaction moves in opposition to the complexity of virtual environments because the number

Figure 1 A user examining the Step WIM.

of possible tasks increases whereas the fundamental capacities of the human remains constant. To address the issue of increasing task complexity, we must consider techniques that make better use of the finite set of human capabilities. The common approaches to increasing task range and complexity are to either squeeze more data out of an existing human channel, perhaps by distinguishing more or different gestures, or to offload task complexity from one overloaded channel to a different channel. In the context of virtual environment interaction, the former approach has been extensively applied to hand gestures as exemplified by the interaction techniques of SmartScene[20]. The latter approach has received increased attention and has achieved notably positive results, for example, when task details that are tedious to express with gestures alone are naturally given through speech commands[3].

The goal of this paper is to present techniques for offloading a range of virtual environment navigation techniques onto previously under-exploited human capabilities for walking, leaning, bending and turning. Certainly most virtual environments allow users to do all the above actions with effects similar or identical to those if performed in the physical world. Our techniques, however, extend and amplify the effects of those actions additionally to support multi-scale navigation through virtual environments and full 360 surround viewing of semi-immersive environments consisting of only three vertical walls.

1.1 Organization

The remainder of this paper is organized in the following manner. First, we discuss previous work related to navigation in virtual environments. Second, we describe the Step WIM (see Figure 1), a tool for quickly traveling to any part of a virtual world, and describe the interaction techniques used to invoke, scale and dismiss it. Third, we describe our leaning technique for navigating short to medium range distances. Fourth, we present auto rotation, a technique for

automatically viewing a full 360 degrees given the constraints of a three-walled display. Finally, we discuss future work and our conclusions.

2 Previous Work

Previous techniques for navigation within virtual environments have covered a broad categorization of approaches ranging from directly manipulating the environment with hand gestures, to indirectly navigating using hand-held widgets, to simulating physical metaphors, to identifying body gestures, and even to recognizing speech commands. Although no systematic study has attempted to evaluate this gamut of controls, Bowman presented preliminary work[1] that evaluated a smaller class of immersive travel techniques and discussed relevant considerations for the design of new techniques. Our review of techniques focuses on general-purpose unconstrained, floor-based navigation controls, although we note the relevance of some application-specific, constrained navigation controls, such as Galyean's guided navigation technique[10].

Perhaps the most prevalent style of navigation control for virtual environments is to directly manipulate the environment with hand gestures. For example, SmartScene provides clever controls for the user to navigate through a virtual world by treating the world as one large object that can be gesturally grabbed, moved and scaled with both hands to achieve the effect of user navigation[20]. Others demonstrated techniques for orbiting about and moving relative to objects specified by hand gestures[15][18]. All of these techniques can be quite effective yet, in complex applications with large sets of interaction tasks, hand-worn or hand-held devices are often overloaded making them more difficult to learn and use.

Widget-based controls are also popular for navigating within virtual environments[7][9][23]. The most relevant widget to our work is Stoakley's implementation of a navigation technique based on "flying" into a hand-held world-in-miniature (WIM)[21]. This technique allows a user first to indicate a desired new viewing location using a hand-held miniature representation of the virtual environment, and second to be seamlessly "flown" to that location by an animated transformation of the hand-held WIM. The primary disadvantage of this and most other widget-based techniques used for navigation is that they again require the involvement of the users hands. Our Step WIM technique provides two important benefits over Stoakley's hand-held WIM. First, it offloads all WIM interactions onto the user's feet and body thereby freeing the hands to perform other interaction tasks. Second, the Step WIM, in addition to working in head mounted displays, also is effective in Cave-based projective environments because the full resolution of the floor is available and the user's hands do not obstruct the WIM display.

A less-generally applied alternative to hand-controlled techniques, is to control navigation with body gestures often coupled to mechanical devices. Darken explored the use of an omni-directional treadmill[4] that with only limited success enabled a user to navigate by walking. Brooks[2] and Iwata[11] have also developed treadmill-based navigation techniques. Others have provided unicycles, bicycles or automobile shells that allow at least some aspects of navigation, such as speed or bearing to be controlled with the feet or a body posture. These techniques tend to be quite restrictive and are generally appropriate for only specific types of simulation environments such as military battlefield training.

A more general although less frequently explored approach is to map body gestures directly to virtual environment navigation. Fuhrmann, et al[8] developed a head-directed navigation technique in which the orientation of the users head determined the direction and speed of navigation. Their technique has the advantage of requiring no additional hardware besides a head tracker, but has

Figure 2 The Step WIM widget which allows users to quickly navigate anywhere in the virtual world. The small sphere by the user's foot indicates his position in the miniature.

the disadvantage that casual head motions when viewing a scene can be misinterpreted as navigation commands. In addition, a severe drawback of this and other head-based techniques, as Bowman discusses[1], is that it is impossible to perform the common and desirable real-world operation of moving in one direction while looking in another. An alternative technique that is often based on head-tracking[22] is to control navigation by walking in place[19]. The speed of movement is coupled to the user's pace, but again the direction of motion is restricted to the user's head orientation. The Placehold VR system[12] allowed a user to walk in place but determined the direction of motion from the orientation of the user's torso thus allowing the decoupling of the user's head orientation from their direction of movement. We feel that this technique is close in spirit with our hands-free navigation philosophy and perhaps should be integrated with our suite of controls to afford another technique for navigating short distances. Another direct body-based navigation technique is found in the Osmose system which uses sensors to measure the tilt of the user's spine and the expansion of their chests[5]. These measurements allow users to move horizontally with body tilt and up and down with chest expansion and contraction.

The final category of techniques for motion control is based on speech recognition. Speech allows a user to modelessly indicate parameters of navigation and can often be used in conjunction with gestures to provide rich, natural immersive navigation controls[3]. We believe that speech controls should play a role in virtual environment navigation, but we also feel that it is also critical to support effective, speech-free navigation techniques for the common situations where speech recognition is unavailable, inappropriate or simply not desired.

3 The Step WIM

The Step WIM is a miniature version of the world that is placed on the ground, under the user's feet in the virtual environment. The idea is derived from Stoakley's hand-held World In Miniature which was used for selecting and manipulating virtual objects[21] as well as navigation and locomotion[16]. However, instead of treating the WIM as a hand-held object, we wanted to achieve an effect similar to walking through a miniature environment landscape, such as Madurodam in The Hague. Consequently, when a user invokes the Step WIM, a miniature version of the virtual environment is placed beneath their feet such that the actual position of the user in the virtual environment coincides with the approxi-

mate location of the user's feet in the miniature (see Figure 2). The Step WIM then functions as an augmented road map. The user can either walk around the Step WIM to gain a better understanding of the virtual environment, or he or she can use the Step WIM to navigate to a specific place by simply walking to a desired location in the WIM and invoking a scaling command, causing the Step WIM to animate scaling up around the user's feet[1], thereby seamlessly transporting the user to the specified virtual environment location. As Bowman[1] and Pausch[16] discuss, animation of the Step WIM is essential to the user's sense of location. In situations where the Step WIM is either too large or too small, the user can, upon command, increase or decrease the size of the wim.

3.1 Invoking, Scaling and Dismissing the Step WIM

In addition to the effect of walking through a virtual environment in miniature, a second critical aspect of the Step WIM is that it can be entirely controlled with a single foot gesture. We determined that a single gesture is sufficient for controlling all three operations of invoking, scaling, and dismissing the Step WIM by performing an informal Wizard of Oz experiment. In this experiment, we asked six people to control the Step WIM by tapping their foot and saying what they wanted the Step WIM to do. When the Step WIM wasn't displayed, it was clear that tapping was meant to invoke the Step WIM. When the Step WIM was displayed, we observed that users looked down at the Step WIM when they wanted tapping to transport them to a new location, but looked away from the Step WIM when they wanted tapping to dismiss the Step WIM.

Based on this very informal experience, we prototyped a number of gestures for controlling the Step WIM. From this set of prototypes, we culled out two different styles of foot gestures that we feel are both natural and robustly recognizable. We outline each of the two gestures both because each gesture requires different sensing technology and because we have not yet conducted a formal evaluation to clarify the advantages of one gesture over the other. In either case, we disambiguate the user's intention to either dismiss the Step WIM or transport him or herself to a new location by estimating the gaze direction when the foot gesture is performed. If the user's head direction is 25 degrees below horizontal (i.e., he or she is looking down at the Step WIM), then we determine the user is trying to move to a new location, otherwise we determine that the user wants to dismiss the Step WIM.

3.1.1 The Foot-Based Interface

In our Cave, a four-sided (three walls and a floor) semi-immersive projection-based virtual environment, users have to wear slippers over their shoes to protect the display floor from scuff marks and dirt from the outside world. Therefore, we developed a pair of Interaction Slippers, shown in Figure 3, that are instrumented to make it easy to identify toe and heel tapping. We were also inspired by the scene in The Wizard of Oz where Dorothy taps her heels to return to Kansas.

To invoke the display of the Step WIM with these Slippers, the user taps his or her toes together, establishing a conductive cloth contact which is easily sensed and treated as a "button" press. Once displayed, the user can move to a new location by simply walking to a desired place in the Step WIM and clicking the toes together again, while looking at the Step WIM. To dismiss the Step WIM,

Figure 3 The Interaction Slippers allow users to tap either their toes or heels to trigger Step WIM functionality.

the user makes the same clicking gesture while looking away from the floor.

Two important design considerations when creating the Interaction Slippers were that they be both comfortable and untethered. We addressed these considerations by embedding a Logitech Trackman Live!™ wireless trackball device that uses digital radio technology[13] into a pair of commercially available slippers. We chose wireless radio technology over other approaches, such as infrared, because it provides a range of up to 30 feet, and does not require unoccluded line-of-sight to a sensor. We inserted the Trackman into a hand-made pouch on the right slipper and rewired two[2] of the Trackman's three buttons by connecting each one to a pair of conductive cloth[14] patches on the instep of the right slipper. On the instep of the left slipper, we placed two more conductive cloth patches. Touching a cloth patch on the left slipper to a cloth patch pair on the right slipper completes the button press circuit. This design enables us to distinguish two gestures corresponding to heel and toe contacts respectively.

3.1.2 Body Controlled Gestural Interface

The alternative interface that we present for controlling the Step WIM works exactly the same as the toe-based controls except the gesture is an upward bounce instead of a toe tap. An upward bounce gesture is detected whenever the user rises on the balls of his or her feet and then quickly drops back down again. Although this gesture can, in theory, be detected solely through head tracking, we employ a simpler gesture recognition algorithm that uses a waist tracker. Waist tracking is accomplished by having the user wear a conventional belt that has a magnetic tracker mounted on the belt buckle. The advantage of a waist tracking gesture recognizer is that it will not inadvertently classify a bouncing head motion (e.g., looking down and up) as a bounce gesture in the same way a head-based gesture recognizer might. There is a clear disadvantage to wearing an augmented belt to track waist position since the user must wear another tethered device, but waist tracking is required for other parts of our interface (See Section 4), so we simply take advantage of the availability of this more robust data.

Our algorithm for recognizing a bounce gesture is initially calibrated by storing the user's waist height h from the tracker attached to the user's belt. We record this value in the display device's coordinate system, since that frame of reference will remain constant as the user moves through the virtual environment. We then monitor each tracker data record, checking whether the user's waist is above the initial waist calibration height by more than a distance of Δh. We found a Δh of 1.5 inches to work well. We then accumulate the amount of time, t_{up}, in which the waist height is above the given

---

[1]The world actually scales up around the projection of the user's head onto the floor, as indicated by a small green icon. By tracking the head instead of the feet, we avoid obscuration issues with a projected display and afford the user fine-grained control of his or her location in the Step WIM via head movement.

[2]Our current implementation of interaction slippers utilizes only two of three Trackman buttons. In future work we plan to use of the third button as well as the trackball.

$h + \Delta h$. If t_{up} is between a threshold, t_{min} and t_{max} we consider this a bouncing gesture.

3.2 Step WIM Scaling

For many environments, a single-sized Step WIM is sufficient to encompass the entire virtual environment. However, some environments are so large that a single-sized Step WIM is inadequate for both providing access to the entire virtual environment and for providing enough detail to accurately control navigation. For example, if the Step WIM for a large environment is scaled to provide reasonable detail, then it will not fit within the physical walking area of the user (in the case of our Cave, an 8 foot square). Alternatively, if the Step WIM is scaled to fit within the user's physical walking area, there may not be enough detail for the user to control navigation precisely.

We present two additional Step WIM controls that address this problem. The first control allows the user to interactively change the scale of the Step WIM. The second control, presented in Section 4, allows the user to navigate the Step WIM, thus providing the user with access to distant regions of the Step WIM that were not previously within his or her physical walking area.

Although the control for Step WIM scaling requires just a single additional gesture, we again provide two different gestures corresponding to whether or not the Interaction Slippers are used.

3.2.1 Foot-Based Scaling

When wearing Interaction Slippers to control the Step WIM, the user activates and deactivates Step WIM scaling mode by clicking the heels together (as distinguished from clicking the toes together). This scaling mode overrides the previous Step WIM controls until it is deactivated by a second heel click. When the user first enters Step WIM scale mode by making heel cloth contacts, the user's head position is projected onto the Step WIM and stored. This projected point is used as the center of scale for changing the size of the Step WIM. As the user walks, instead of moving about within the Step WIM, the Step WIM is translated, so that the center of scale always lies at the projection of the user's head onto the floor (See Figure 4). In addition, if the user walks forward with respect to the Step WIM, as if to get a closer look, the Step WIM gets larger. If the user walks backward within the Step WIM, as if to see a larger picture, the Step WIM scales smaller. To return to the standard mode of controlling the Step WIM with toe taps, the user must again click his or her heels together. This second heel click freezes the scale of the Step WIM until the next time the user enters Step WIM scale mode.

Figure 4 A user prepares to scale the Step WIM upward using the Interaction Slippers (left). As the user moves forward the Step WIM gets larger and his position in the miniature is maintained (right).

3.2.2 Body Controlled Gestural Scaling

Alternatively, when controlling the Step WIM with bounce gestures, the user can change the scale of the Step WIM directly without having to enter a special scaling mode. The control for scaling the Step WIM smaller about the projection of the user's waist is for the user to simply rise up on the balls of his or her feet for longer than the bounce time threshold, t_{max}. Once this time threshold has been exceeded, the Step WIM will start to shrink at a rate of 2 percent per second. To make the Step WIM larger, the user assumes a slight crouching posture by bending his or her knees enough to lower the waist position by Δh. Once again, if this posture is held for longer than t_{max}, the Step WIM will begin to grow at a rate of 2 percent per second.

As an example of when changing the size of the Step WIM is useful, consider a large virtual environment the size of the Earth. The user can scale the Step WIM down so that the entire map of the Earth fits within the physical space of the available walking area. Then the user can walk roughly to a position of interest, perhaps a country, and then partially scale the Step WIM up about that position in order to more easily identify a desired location, perhaps a city, before finally transporting him or herself to that location by scaling the Step WIM up completely.

4 Navigation By Leaning

The Step WIM controls provide a hands-free navigation technique for moving medium to large distances through virtual environments. To also support hands-free navigation of small to medium distances, we refined the navigation-by-leaning technique proposed by [6] to reduce the likelihood of accidentally recognizing a relaxed posture as a leaning action. Our leaning technique allows the user to navigate through the virtual environment by simply leaning at the waist in the desired direction of movement. An added benefit of leaning controls over head-based navigation techniques is that the user can look in a direction that is different than the one in which he or she is moving. For example, in an art gallery environment the user can lean in order to move along a wall of pictures while always concentrating on the picture immediately in front of him or her (See Figure 5).

In addition to navigating relatively small distances by leaning, the user can also lean to translate the Step WIM to gain access to Step WIM locations that would not otherwise fit within the user's physical walking area. The decision to move the Step WIM widget by leaning instead of navigating through the virtual environment depends on whether the Step WIM is active and whether the user's gaze direction is 25 degrees below the horizontal (i.e., at the Step WIM).

Figure 5 A user leans to the right (left) to view a painting (right).

To detect the direction and magnitude of leaning, both the user's waist and head are tracked. We compute the direction of leaning, the *leaning* vector $\vec{L_R}$, by projecting the vector from the waist to the head onto the horizontal floor plane. To map $\vec{L_R}$ to navigation controls, we simply map the navigation heading to the direction of $\vec{L_R}$ and we map the navigation speed to $\|\vec{L_R}\|$, such that the more the user leans in a given direction, the faster he or she will go.
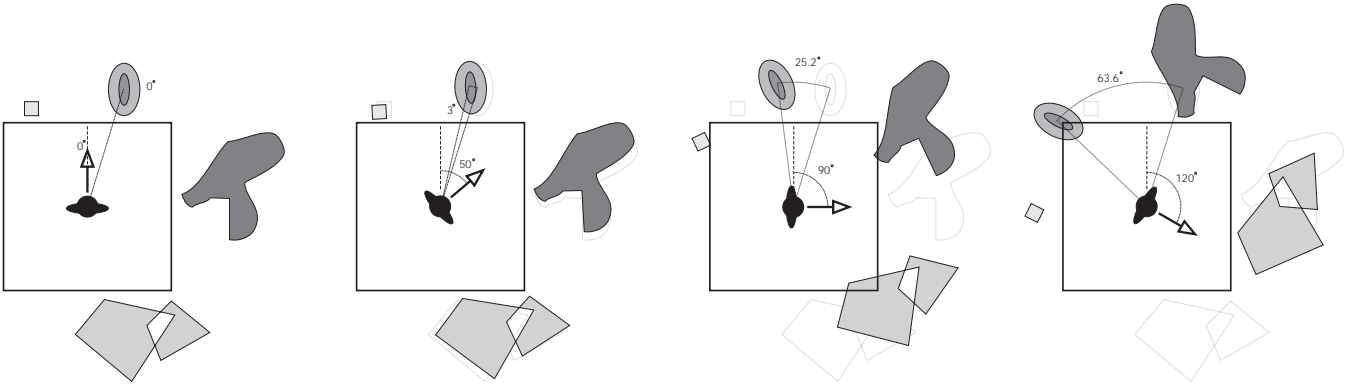
Figure 6 An illustration of the auto rotation technique. As the user rotates to the right, the world auto rotates in the opposite direction based on the scaled 2D Gaussian function (see equation 5).

The actual mapping function we use between $\|\vec{L_R}\|$ and navigation speed is given by a function that is dependent on where the user is located with respect to his or her walking area. This position dependence derives from observations of how people work in a virtual environment with a relatively small walking area. Typically, we find that people choose to walk in the direction they want to go until they cannot walk any further, at which point they switch to a navigation technique. Therefore, our mapping function is most sensitive to leaning in a given direction when the user cannot physically move any farther in that direction. This varied sensitivity to leaning makes navigation control more robust since accidental leaning caused by normal human head and body postures tend to be de-emphasized. For example, when users want to move forward, they will have to lean farther forward, if they are standing in the center of their physical walking area, than if they are already close to the front of it. Thus we fine-tuned our leaning function so that inadvertent variation in the user's leaning is essentially discarded when the user is at the center of the working area, while the user needs to lean only subtly in the direction he or she wants to go when already close to a boundary (i.e., a wall of the Cave).

Our position dependent function is a linear function which provides the minimum amount L_T the user has to lean to produce a translation.

$$L_T = a \cdot D_{min} + b \tag{1}$$

where D_{min} is the minimum distance between the user and a physical boundary in the direction he or she is leaning.

We calculate a and b given the minimum leaning value L_{min} when the distance to a boundary is close to 0, and the maximum L_{max}, when the distance is equal to the diagonal of our Cave, approximately $11.3'$. Table 1 shows the values we have used for these thresholds, L_{min} and L_{max}, both when leaning forward and backwards[3] and the corresponding *slope* and *y-intercept* values, a and b, used in each case.

The user's unscaled velocity v is then calculated by

$$v = \|\vec{L_R}\| - L_T. \tag{2}$$

4.1 Exponential Mapping For Navigation Velocity

During informal user testing of the leaning control, we noticed that when users wanted to move somewhere in the virtual world their

---

[3]We decided to modify the thresholds when the user is leaning backwards, since it is a more difficult gesture to perform.

	Leaning Forward	Leaning Backwards
L_{min}	$4.5''$	$1.0''$
L_{max}	$7.0''$	$7.0''$
a (Slope)	0.018	0.044
b (Y-Intercept in feet)	0.375	0.083

Table 1 The minimum and maximum leaning thresholds used to calculate the slope and y-intercept to our linear mapping function.

gaze was generally focused on the place they wanted to go even as this location was moving towards them. Since objects in our virtual environments are generally lower than the user's head height, we improved our mapping function by recognizing that, as distant objects come closer, the user's head naturally tilts down to maintain focus. Thus, we correlate the rate of movement to the user's head orientation with respect to the vertical axis such that the movement rate is exponentially decreased as the user's head rotates increasingly downward even though the amount of lean is constant. This exponential mapping has proven useful, especially for navigating the Step WIM, since the object of focus appears to smoothly decelerate as it draws near and the user's head tilts further down to see it. In general, the user's head orientation can be thought of as a vernier control that modifies the rate of movement indicated by the user's amount of lean.

We have found that a scaled exponential function

$$F = \alpha e^{-\beta |\vec{head} \cdot \vec{V_{up}}|} \tag{3}$$

where α is the maximum speed factor, β defines the steepness of the exponential curve, $\vec{head}$ is the user's head orientation vector, and $\vec{V_{up}}$ is the vertical vector coming out of the display floor, provides smooth translations and works well in the different environments we have tested.

The final leaning velocity is calculated by

$$v_{final} = F \cdot v \tag{4}$$

which is then applied to the leaning vector L_R. The coefficients for the exponential function, α and β, change depending upon the scale at which the navigation is taking place. When users are leaning to navigate the virtual world, values of 3.0 and 6.0 for α and β provide a good fall-off for movement as they focus on a point closer to their position in our virtual environment. For translating the Step WIM, these values were different since the user is mostly looking down towards the floor. When the size of the Step WIM exceeded the

physical space of our Cave, values for α equal to 2.5 and β equal to 5.0 worked well.

5 Auto Rotation

In fully immersive virtual environments, there is generally no need to provide any explicit control for rotating the virtual environment relative to the user, since the user can turn to face any direction in the virtual environment. However, a common semi-immersive display configuration is a three-walled Cave which affords only a 270 degree view of the world when the user stands at the Cave center. In such semi-immersive environments, the user generally needs an explicit user interface control for rotating the virtual world so that the full 360 degrees of the virtual environment can be viewed.

To provide a complete suite of hands-free navigation controls for semi-immersive virtual environments with only three walls, we developed an amplified rotation technique, a non-isomorphic 3D rotational technique that implicitly allows a user to view a full 360 degrees even though the physical display environment does not surround the user (see Figure 6). Poupyrev most recently studied non-isomporhic 3D rotational techniques and found them to be effective for hand-held object manipulation[17]. However, applying non-isomorphic rotation to navigation is more complicated, since rapid, unexpected rotation of a virtual environment can easily cause cybersickness. Therefore, we prototyped a number of different techniques that all attempt to make the automatically generated amplified rotation as subtle and gentle as possible.

Our first prototype amplified a user's head rotation by linearly counter-rotating the world such that a 120 degree rotation by the user effectively amounts to a 180 degree rotation of the user in the virtual world. Although this technique allows the user to see a full 360 degrees in a three walled Cave, most of our trial users felt at least some degree of cybersickness after only a few minutes. As our second prototype, we keyed the linear counter-rotation of the world to the orientation of the user's torso instead of his or her head, thus eliminating most of the unnatural world rotation when the user was just glancing about. Despite reduction in cybersickness, this technique felt unnatural to our trial users who generally thought that there was continually too much rotational motion, especially as they walked around the environment.

The final technique that balances most user concerns is to apply a non-linear mapping function in which amplified rotation effectively kicks in only after the user has rotated beyond a threshold that is dependent on the user's orientation vector and position in the Cave.

In order to calculate the correct viewing angle, we first define the user's waist orientation as the angle θ between the waist direction vector and the horizontal vector to the front Cave wall projected onto the floor plane[4]. Next we define d as the distance the user is from the back of the Cave. Using these two variables, we calculate the rotation factor ϕ using a scaled 2D Gaussian function

$$\phi = f(\theta, d) = \frac{1}{\sqrt{2\pi}\sigma_1} \cdot e^{-\frac{(|\theta| - \pi(1 - d/L))^2}{2\sigma_2^2}} \quad (5)$$

where σ_1 is a Gaussian height parameter, σ_2 is a Gaussian steepness parameter, L is a normalization constant which is used to lessen the effect of d, and the function's μ value is set to π. Using ϕ, we find the new viewing angle by

$$\theta_{new} = \theta(1 - \phi). \quad (6)$$

---

[4]We do not take into account whether the user is looking up or down.
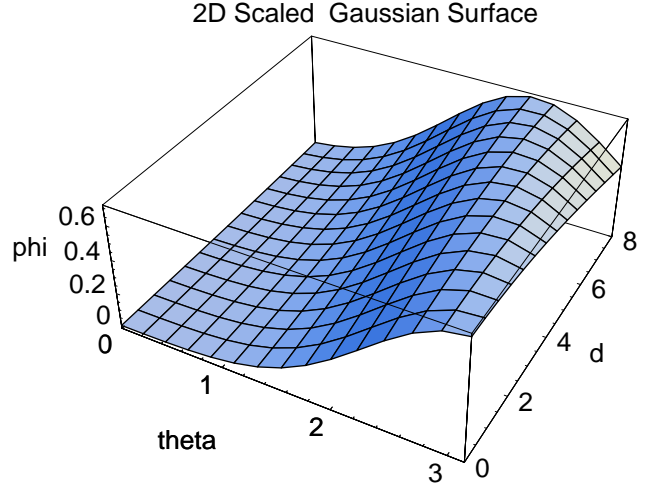

2D Scaled Gaussian Surface

Figure 7 A visual representation of the scaled 2D Gaussian surface we use to find the rotation factor, which determines the degree of rotation amplification. For this graph, σ_1 equals 0.57, σ_2 equals 0.85, and L equals 30.

In order to get a better understanding of what the 2D Gaussian surface does, consider Figure 7. In the figure, we see that if the user is close to the front wall of the Cave, d equal to 0, the user has more visual display in which to rotate about and, as a result, the Gaussian bump is shifted closer to π on the θ-axis, reducing the amount of rotation amplification as the user's rotation angle gets larger. Conversely, if the user is closer to the back of the Cave (d equal to 8), he or she only has 180 degrees of rotation available before looking out of the display. Therefore, we need greater rotation amplification for the user to see a full 360 degrees. So the Gaussian bump is shifted closer to 0 on the θ-axis.

In combination with the leaning metaphor described in the previous section, users can also travel in directions that were originally directly behind them when they faced the front wall of our three-sided Cave by first turning to face either the right or left wall. We have observed that users need time to adjust to this distorted spatial mapping, but can at least navigate in any direction after only a few minutes. However, we have not yet attempted to quantify the effect of this auto rotation technique on a user's sense of spatial relations and this remains an important area of new research.

6 Future Work

We hope to explore a number of specific avenues in future work including improvements to our techniques for tracking the user and extensions to our current suite of interaction techniques.

Our current implementations require that users minimally wear a head and belt tracker; although we believe that it may be possible to robustly perform all operations, except the toe-tapping gestures, with only one accurate head-tracking device. A further improvement would be to completely untether the user by developing appropriate vision or wireless tracking techniques. Furthermore, we believe that our leaning gestures could be made even more subtle by incorporating multiple pressure sensors onto the soles of our Interaction Slippers.

We believe that our current set of controls are adequate for navigating through a broad range of virtual environments, although we believe that additional hands-free controls would be helpful for navigating around specific objects and for navigating through spaces

that do not have a floor-plane constraint. In addition we would like to extend our navigation controls to include the walking in place technique and we would like to explore multi-modal navigation techniques based on speech recognition.

7 Conclusion

We have presented a cohesive suite of hands-free controls for multi-scale navigation through a broad class of floor-constrained virtual environments. Since all our controls are hands-free, virtual environment designers have greater flexibility when mapping additional functionality since the user's hands are completely offloaded.

Specifically, our controls allow a user to move small and medium distances, users can simply lean in the direction they want to move independent of their head orientation. Unlike previous leaning techniques[5][6], our leaning control is modified for robustness to consider both the users' location relative to their physical walking area and their head orientation. To move coarse distances, the user can gesturally invoke an adaptation of a conventional WIM, a *Step WIM*, that is displayed on the floor of the user's physical walking area. The Step WIM is controlled either by toe-tapping with wireless *Interaction Slippers* or by tip-toe bouncing gestures. The Step WIM is further extended to support interactive scaling using heel-tapping or crouching gestures, and to support translational movement by leaning.

References

[1] Bowman, D. A., Koller, D., and Hodges, L. F. Travel in Immersive Virtual Environments: an Evaluation of Viewpoint Motion Control Techniques. In *Proceedings of IEEE VRAIS'97*, 45-52, 1997.

[2] Brooks, F. P. Walkthrough: A Dynamic Graphics System For Simulating Virtual Building. In *Proceedings of the 1986 Workshop in Interactive 3D Graphics*, 9-21, 1986.

[3] Cohen, P. R., Johnston, M., McGee, D., Oviatt, S., Pittman, J., Smith, I., Chen, L., and Clow, J. (1997). QuickSet: Multimodal interaction for distributed applications, In *Proceedings of the Fifth International Multimedia Conference (Multimedia '97)*, ACM Press, 31-40, 1997.

[4] Darken, R., Cockayne, W. and Carmein, D. The Omni-Directional Treadmill: A Locomotion Device for Virtual Worlds. Interaction. In *Proceedings of UIST'97*, ACM Press, 213-222, 1997.

[5] Davies, C. and Harrison, J. Osmose: Towards Broadening the Aesthetics of Virtual Reality. *Computer Graphics* 30(4): 25-28, 1996.

[6] Fairchild, K., Hai, L., Loo, J., Hern, N. and Serra, L. The Heaven and Earth Virtual Reality: Design Applications for Novice Users. In *Proceedings of IEEE Symposium on Research Frontiers in Virtual Reality*, 47-53, 1993.

[7] Fisher, S. The AMES Virtual Environment Workstation (VIEW). In *SIGGRAPH 89 Course Notes #29*, 1989.

[8] Fuhrmann, A., Schmalstieg, D. and Gervautz M. Strolling through Cyberspace with Your Hands in Your Pockets: Head Directed Navigation in Virtual Environments, In *Virtual Environments '98 (Proceedings of the 4th EUROGRAPHICS Workshop on Virtual Environments)*, Springer-Verlag, 216-227, 1998.

[9] Furness, T. Configuring Virtual Space for the Super Cockpit. *Human Interface Technology (HIT) Laboratory Technical Report*, University of Washington, HITL-M-89-1, 1989.

[10] Galyean, T. Guided Navigation of Virtual Environments. In *Proceedings of Symposium on Interactive 3D Graphics*, ACM Press, 103-104, 1995.

[11] Iwata, H. Walking about Virtual Environments on an Infinite Floor, In *Proceedings of Virtual Reality'99*, 286-293, 1999.

[12] Laurel, B., Stickland, R., and Tow, R. Placeholder: Landscale and Narrative in Virtual Environments, ACM Computer Graphics Quarterly, Volume 28, Number 2, May 1994.

[13] Logitech, http://www.logitech.com, 2000.

[14] Mann, S. Smart Clothing: The Wearable Computer and WearCam. *Personal Technologies*, Volume 1, Issue 1, March, 1997.

[15] Pierce, J., Forsberg, A., Conway, M., Hong, S., Zeleznik, R. and Mine, M., Image Plane Interaction Techniques in 3D Immersive Environments. In *Proceedings of Symposium on Interactive 3D Graphics*, ACM Press, 39-43, 1997.

[16] Pausch, R., Burnette, T., Brockway, D. and Weiblen, M. Navigation and Locomotion in Virtual Worlds Via Flight into Hand-Held Miniatures. In *Proceedings of SIGGRAPH 95*, ACM Press, 399-400, 1995.

[17] Poupyrev, I., Weghorst, S., and Fels, S. Non-Isomorphic 3D Rotational Techniques. In *Proceedings of the Conference on Human Factors in Computing Systems, CHI 2000*, 540-547, 2000.

[18] Mine, M. Moving Objects In Space: Exploiting Proprioception In Virtual Environment Interaction. In *Proceedings of SIGGRAPH 97*, ACM Press, 19-26 , 1997.

[19] Slater, M, Steed, A., and Usoh, M. The Virtual Treadmill: A Naturalistic Metaphor for Navigation in Immersive Environments. *First Eurographics Workshop on Virtual Reality*, 71-86, 1993.

[20] SmartScene™ is a product of Multigen, Inc. More information on SmartScene™ is available from Multigen's website at http://www.multigen.com/products/smarts.com, 2000.

[21] Stoakley, R., Conway, M. J., and Pausch, R. Virtual Reality on a WIM: Interactive Worlds in Miniature, In *Proceedings of Human Factors and Computing Systems, CHI'95*, 265-272, 1995.

[22] Usoh, M., Arthur, K., Whitton M., Bastos R., Steed, A., Slater, M., and Brooks, F. Walking > Walking-in-Place > Flying, in Virtual Environments", In the *Proceedings of SIGGRAPH 99*, ACM Press, 359-364, 1999.

[23] Ware, C. and Osborne, S. Exploration and Virtual Camera Control in Virtual Three Dimensional Environments. *Computer Graphics* 24(2): 175-183, 1990.