VirtualWars: Towards a More Immersive VR Experience

Fahim Dalvi,[*] Tariq Patanam[†]
Stanford EE 267, Virtual Reality, Course Report, Instructors: Gordon Wetzstein and Robert Konrad

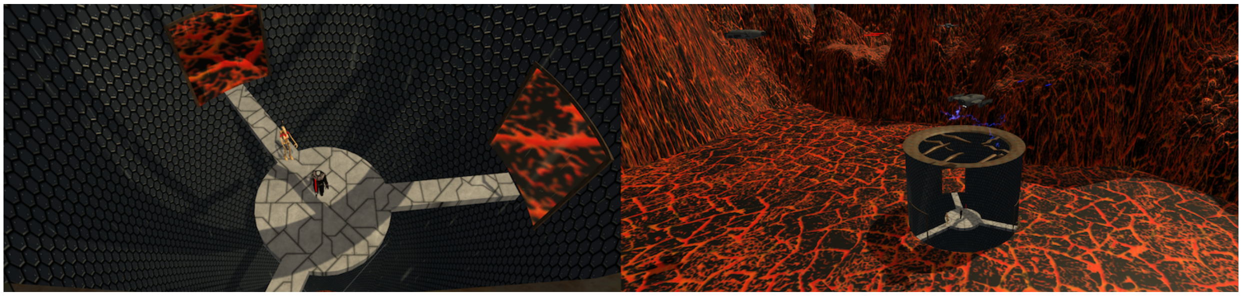

Figure 1: *Scene Overview*

Abstract

Ensuring that virtual reality experiences are immersive is key to ensuring the success of VR and even VR. However, despite impressive commercial advancements from the Oculus Rift to the HTC Vive, a number of inherent limitations remain when comparing virtual experiences to real experiences: field of view, limb (mainly hand) tracking, position tracking in the world, haptic feedback, and more. In this study we seek to test a number of creative workarounds to create a fully immersive experience with current technological limitations. We found that overall, immersive experiences could be created, but because of the limitations of the technology, limitations had to be imposed on the virtual world such as how the content had to be presented (interactively and not passively), how objects were destroyed, and more.

Keywords: virtual reality, content generation, immersion, positional tracking

1 Introduction

Recent advances in the virtual reality experience from the Oculus Rift to the HTC Vive are exciting and groundbreaking, but still retain many of the traditional limitations: an unnatural field of view, latencies in the graphics pipeline that can conflict with headtracking, minimal haptic feedback, if any at all, and more. The human perceptual system is extremely good at picking up on such limitations, detracting from a truly immersive experience. As stated by Michael Abrash in the 2015 Oculus developer conference, driving the perceptual system and physical interaction are two of the three keys to making VR truly immersive.

Previous work has defined an immersive virtual reality (IVR) experience by the number of sensorimotor contingencies (SC) that it supports [Slater 2009]. For instance, moving one's head or eyes should change the visual experience we perceive and is an SC that most VR experiences support. Our work focuses on reviewing and testing a variety of techniques to support as many SC's as possible with current technological limitations, drawing from areas including detailed content, realistic visual and auditory cues, position and hand tracking, and more.

*e-mail:fdalvi@stanford.edu
†tpatanam@stanford.edu

2 Related Work

Most fundamental in an IVR experience is probably the environment and content itself. In fact Slater et. al posit that the quality of an IVR experience is a function of place illusion and plausiblity illusion. Place illusion is the sense of presence, of 'being there' in the virtual world and not where one actually is in the physical world. This is not only a factor of how interactive and detailed the scene is, but also how much the participant seeks to interact with the environment. For instance, if a person insists on touching things in a very visually appealing virtual environment without haptics, place illusion (PI) would be very low for this person. Then, as we will consider, it is important to direct the attention of the user appropriately for high place illusion. Plausibility illusion (Psi) is that the idea that parts of the environment not directly in your control actually refer to you. A person smiling at you in the virtual world for instance would provide high plausibility illusion. This too we consider [Slater 2009].

Our work is both a survey of a host of previous techniques used to increase place illusion and plausibility illusion, as well as an exploration of new, related ideas. Besides visual sensory cues, we focused on three areas. Auditory cues are key to human perception. Multiple studies have shown that auditory cues increase the sense of presence in virtual environments, especially when they are paired with corresponding visual cues [Hendrix and Barfield 1996][Riecke et al. 2009]. Second, human locomotion plays a large role in interaction with the real world, and therefore should play a large one in the previous world. Slater et. al previously showed that subjects walking in place had increased sense of presence within a virtual environment, but Usoh et. al demonstrated that actual walking in the physical world corresponding with walking in the virtual world created an even greater sense of presence for users [Slater et al. 1995][Usoh et al. 1999]. Finally, haptic systems have long since been produced, but are either very expensive or limited. For instance, the most accessible commercially available technology to provide force feedback, the Novint Falcon, has only 3 degrees of freedom. Another more promising technologies is the haptic glove which can provide up to 20 degrees of freedom and can provide up to 6N of force feedback. However such technologies are still intricate and far from commercially available[Blake and Gurocak 2009]. In order to explore what can be done in terms of haptics with current technology then, we decided to focus on improving the haptic experience without employing such novel methods but rather through the scene itself. More will be said later about these techniques.

3 Approach

As aforementioned, we classify our approach into four primary categories:

3.1 Content

Content and how it is presented plays a crucial role in making a VR scene immersive. Beautiful photorealistic content can be presented passively, but because of the limitations of VR itself, such as field of view and limitations of the graphics pipeline to render photorealistic scenes in real-time, is easily perceived as unrealistic. This idea relates back to place illusion and the sense of being in the virtual world and not in the physical one. In a passive scene, where content is simply presented and not forcing any particular kind of interaction, the user could insist on, for instance to look at how his feet move on the ground rather than at the people chatting in front of him. This would bode badly for a virtual reality experience which does not track ones feet, even if it does present a stunning photorealistic scene. Therefore, in order to increase place illusion, a primary technique we employed is interactive content that forced the user to focus on the intended experience. The lightsaber can be moved and played around with. After a brief relaxing opening in the beginning, the droids after being killed, respawn to come to attack the user continuously so the user is not given a lot of time to passively observe. Furthermore, we wrote a script such that the once droids reach the platform, if the user moves around the platform, the droids follow the user. Not only does this increase place illusion but focusing the user's attention, but it also increases plausibility illusion by instilling the fear that the droids are coming specifically for them.

In order to make the scene cohesive, we decided to model a large portion of the scene ourselves. We decided to not use a lot of different models from various sources to ensure that each individual entity blends well with the rest of the scene. We used Blender and Unity to do most of the modelling, and Gimp to handle textures.

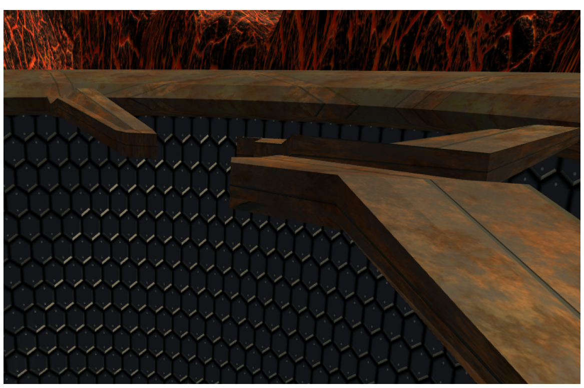

Figure 2: *Example of normal maps on the walls*

In order to increase the fidelity of the scene, we also modelled, keeping several rendering techniques in mind. We used normal maps and bump maps to give extra details to the object without increasing the vertex count significantly. An example is shown in Figure 2. We also used several shaders to add real-time effects to our scene. For instance, we used a disintegration shader to slowly disintgerate away the droids when they are hit with the lightsaber. A separate shader was to generate random electric currents to simulate the source of lightning.

Next, we focused on improving the environment. We used the particle system in Unity to add rain to the scene (Figure 3). Custom scripts were used to augment the rain with bright flashes of light in the scene to simulate lightning.

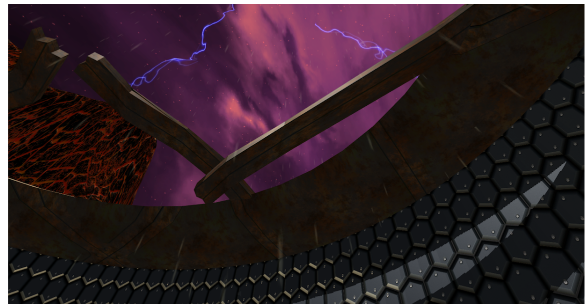

Figure 3: *Rain and lightning effects*

Finally, we also ensured that our virtual world conformed to some of the constraints in the real world, so as to minimize the disconnect between the virtual and the real world. For instance, our virtual world has a platform that is suspended in air, and as such the player will fall off if they move beyond the edge of our platform. The edges occurs at the limits of our positional tracking system (described in 3.4), and hence we are ensuring that the user does not leave the area under which we operate very naturally.

3.2 Visual and Auditory cues

The next major category we focused on was auditory cues and aligning visual cues to match with the sounds being played. We incorporated stereo sound in our scene on various objects to give user spatial cues. If we do not take the 3D world into consideration, sounds do not change when the user moves, which is completely unnatural to the human perceptual system. The effect is amplified with moving objects, since far away objects would then sound the same as if they were next to the user's ears. Hence, after incorporating stereo sounds, real world phenomena such as the Doppler effect were simulated correctly, improving the level of immersion.

We also added several other background noises to make sure the scene does not feel unnaturally quiet. Passing ships and flying bullets both carry noise, so the user hears them move around them. We also add sound effects such as rain and thunder, and made sure that these sounds corresponded to the lightning flashes.

We also used Unity's rendering engine to cast shadows, with a combination of hard and soft shadows to increase the fidelity of the scene without taking a significant hit on the rendering loop efficiency.

3.3 Animations

The next primary category we worked on was animations. Unnaturally moving things catch the attention of the human perceptual system. Hence, it was crucial that the moving objects in our scene follow natural patterns. Animating the ships was done using scripts, since they travel in fairly simple patterns. We also had to add some randomization, so that the patterns did not seem too repetitive.

Animating the droids was much more complex, and we used Blender and Unity in combination to animated them. We first rigged our droid model with bones in Blender, and used the inverse kinematics engine built into Blender to realistically adjust droid poses (Figure 4).

Figure 4: *Droid model rigged in Blender. Right figure shows the bones with X-Ray vision.*

Once we had the individual pose animations done (walking, hitting etc.), we used Unity's animation engine to move the droids around in our scene. We used the animation curves that Unity allows us to create to ensure smooth and realistic motion. We also wrote scripts that change the animations depending on how close the droid is to the player. For instance, the hitting animation is not played if the droid is too far away, since that would not be the natural thing to do. Another animation on the droids is the disintegrating and explosion animation, which is done with the help of shaders and some scripting (Figure 5). Finally, we wrote scripts to properly respawn the droids at the correct times, so that the user is neither underwhelmed nor overwhelmed.

3.4 Sensors

The final category we wanted to focus on was sensors. We wanted to use realtime real world data such as the user's movements to be incorporated in our virtual world, so that the experience is even more seamless and immersive. We use two Inertial Measurement Units (IMU's) and the Kinect.

The first IMU is used to perform head tracking. The stereo rendering is handled by the CardboardSDK provided by Google, and it takes care of ensuring that the head and neck model is applied so that head tilts feel natural. The second IMU is used to control the lightsaber in our virtual scene. Because of the limitations of positional tracking using a gyroscope and accelerometer alone, we only make use of the orientation from the IMU. We use the hilt of the lightsaber as the center of rotation. Before using the orientation values, we clamp the angles within certain limits to ensure that the lightsaber movement is realistic in the scene.

Finally, we use the Kinect to perform high level position tracking of the user's body. We use the segmentation and depth analysis built into the Kinect to track the users motion in the two axes (The vertical axis is ignored, since jumping is not an essential component of our scene). The kinect sensors work well from around 2 meters to 6 meters while tracking the entire body, and using our scripts in unity, we mapped these physical bounds to correspond to the size of our platform.

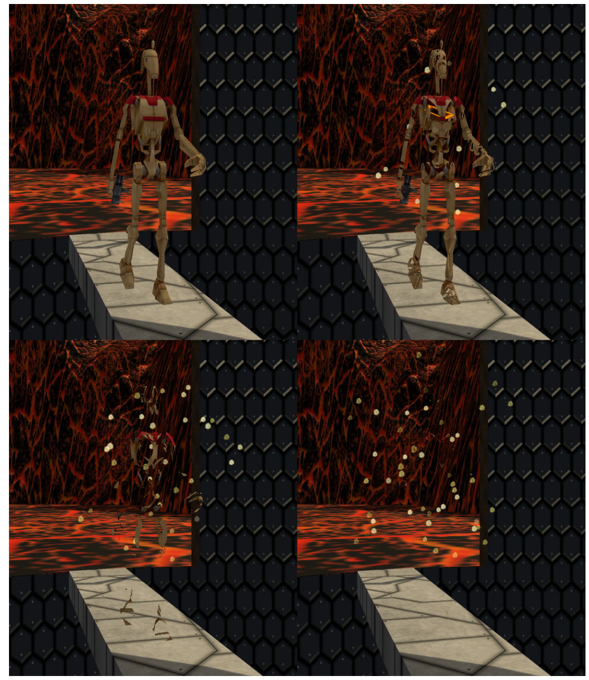

Figure 5: *Droid Disintegration. Read left to right, top to bottom.*

4 Results and Analysis

The immersive qualiy of VR is hard to measure scientifically. Questionnaires are commonly used but are often unstable because prior information about ratings system can change how presence or the sense of being in the virtual world is affected [Freeman et al. 1999]. It has also been shown that users using questionnaires to rate virtual experience and real experience on the amount of presence, rated both statistically the same [Usoh et al. 2000]. Therefore we evaluated our methods primarily on two other bases.

Second, some qualitative user testing was employed, following the advice of prior research where, for instance, presence was measured by how similarly users respond in virtual experiences to how they would respond in real experiences [Sanchez-Vives and Slater 2005]. The following are user reactions to the different techniques we employed to increase PI, Psi, and immersion overall.

An analysis of the techniques individually show that some, perhaps even peripheral effects were very effective while other techniques, even more central effects were ineffective or things users did not comment on.

Taken overall, however, many users were heavily immersed in the scene, at least for the first 30 seconds or so. This shows that while most users did not comment on small factors, especially peripheral ones like rain, individually, each technique contributed to the scene as whole. Such an understanding makes sense because humans do not consciously perceive every minor detail of the real world, but each detail corresponds to our perception as a whole.

The employment of audio was especially immersive. Most users did not hear or understand outside conversation, not because the audio was too loud but because of the numerous auditory cues from thunder to the rain to the droid ships, within the environment. The most commonly cited reason for lack of immersion was the uncharacteristic swinging of the lightsaber. Because only only IMU was used for hand tracking, only 3 degrees of freedom, that is rotations along the three axes could be provided. Therefore users resorted

User Reactions to Immersion Techniques		
Technique	**Positive Reaction**	**Negative Reaction**
Rain	None	Some questioned why it was raining
Lightning followed by thunder in sync	Impressed and caused users to look overhead	None
Flickering and turning off light	Some users jumped at the instant the lights turned off	Most users were indifferent to the change
Droid ships fighting overhead	Impressed by the fighting performance between the fighters and encouraged users to look up	None
Droids following user	Increased the fright of users especially after lights turned off	None
Droids attacking from all sides to force turning and not standing	perhaps the most successful technique at keeping the users engaged. A few users even stepped back in the real world when they were surprised to find droids behind them	Some users did not turn around without prompting
Disintegrating droids instead of slicing solid droid to negate the need for haptic feedback	Users did not question the lack of haptic feedback	Some users commented on the unnatural particles that emanated from the droid dissolving
Spatial audio	Caused users to turn around as they followed noises moving across the screen	Not adding spatial audio to the droid meant many users did not turn around to face them till prompted
Position tracking	Users were engaged longer and more involved in interacting with the scene when positional tracking was enabled	Users complained of not being able to rotate the body on its own plane

Table 1: *Table showing how users responded to a variety of the immersion techniques we employed*

to either holding their elbow at the side and swinging around their forearm, or simply moving their wrist. Both interactions are not natural for lightsaber swinging. A few users also complained of the lag between their hand and the movement of the light saber in the virtual world. Interestingly enough they did not complain of any lag in the headtracking, though almost identical hardware and code was used for both head tracking and hand tracking.

5 Discussion

5.1 Challenges

We faced several challenges at various stages in the project. One of the first aspects we worked on was to make the lightsaber interaction part of the scene wireless. We used the CardboardSDK and wrote an iOS app that was able to communicate the orientation of the phone wirelessly to the machine where the scene was being rendered. We were using Unity's built in Networking API, and unfortunately, the network latency was two high. Even though we were using the low level API's in Unity to avoid any unnecessary overhead, the overall latency was still too high. We tried using unreliable transport channels to reduce the overhead even more, but that did not improve the quality of the simulation. We eventually shifted to a tethered solution to reduce the latency enough so that the simulation is smooth.

Another challenge that we faced was the lack of haptics. Since we could not prioritize techniques such as vibration feedback, we needed to ensure that the discrepancy generated by physical contact in the virtual world was minimized. Firstly, we use a lightsaber instead of a sword, since one would expect lasers to just cut through objects without much feedback. We also reduced the density of the droid models in the scene, so they react more naturally to hits from the light saber.

We also spend a significant amount of time to make the droid animations look natural. The inverse kinematics engine was very helpful in making the poses look natural, but we still had to adjust specific joints that humans normally move when performing the motions we were trying to incorporate.

Finally, we also faced challenges with the position tracking limitations of Kinect. As talked about in Section 3.4, we overcame this challenge by integrating the limits directly into our virtual scene so that the discrepancy between the real and virtual world is minimized.

5.2 Future Work

There are several further improvements that we can make to make the experience even more immersive. Firstly, few of the challenges we faced can be solved with more advanced techniques. For example, we could use lower latency networks such as Bluetooth or write our own network manager that is not dependent on Unity's networking implementation. We could also incorporate piezo actuators placed on the user's hand to simulate some force feedback. Finally, we can also employ more accurate position tracking that has a wider range to make the scene bigger and even more interactive.

Concerning the implementations that we already have, one possible improvement is in the orientation tracking we do for the lightsaber. We could use more IMU's (for example, one on the user's elbow and one on the wrist for more accurate hand orientation detection. We could also employ more powerful filters for all of our IMU code like the Extended Kalman filter to get more precise calculations, instead of the simple complementary filter approach we currently use.

5.3 Conclusion

We demonstrated that an immersive experience can be created with currently existing technology, largely by adjusting how the user interacts with the scene. Haptic feedback problems were minimized and position tracking was incorportated successfully into the scene without removing from the immersive experience. This bodes well for the success of VR in the short term, but also shows that with current technology, limitations, such as constraints on the platform size in the virtual world, must be imposed.

Acknowledgements

We would like to thank Gordon Wetzstein and Robert Konrad for their support during the entire course. We would also like Vasanth Mohan for his advice and direction for Unity. Finally we are also grateful to our friends who helped us test our scenes throughout the project and gave us immensely useful feedback.

References

BLAKE, J., AND GUROCAK, H. B. 2009. Haptic glove with mr brakes for virtual reality. *Mechatronics, IEEE/ASME Transactions on 14*, 5, 606–615.

FREEMAN, J., AVONS, S. E., PEARSON, D. E., AND IJSSELSTEIJN, W. A. 1999. Effects of sensory information and prior experience on direct subjective ratings of presence. *Presence 8*, 1, 1–13.

HENDRIX, C., AND BARFIELD, W. 1996. The sense of presence within auditory virtual environments. *Presence: Teleoperators & Virtual Environments 5*, 3, 290–301.

RIECKE, B. E., VÄLJAMÄE, A., AND SCHULTE-PELKUM, J. 2009. Moving sounds enhance the visually-induced self-motion illusion (circular vection) in virtual reality. *ACM Transactions on Applied Perception (TAP) 6*, 2, 7.

SANCHEZ-VIVES, M. V., AND SLATER, M. 2005. From presence to consciousness through virtual reality. *Nature Reviews Neuroscience 6*, 4, 332–339.

SLATER, M., USOH, M., AND STEED, A. 1995. Taking steps: the influence of a walking technique on presence in virtual reality. *ACM Transactions on Computer-Human Interaction (TOCHI) 2*, 3, 201–219.

SLATER, M. 2009. Place illusion and plausibility can lead to realistic behaviour in immersive virtual environments. *Philosophical Transactions of the Royal Society of London B: Biological Sciences 364*, 1535, 3549–3557.

USOH, M., ARTHUR, K., WHITTON, M. C., BASTOS, R., STEED, A., SLATER, M., AND BROOKS JR, F. P. 1999. Walking¿ walking-in-place¿ flying, in virtual environments. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., 359–364.

USOH, M., CATENA, E., ARMAN, S., AND SLATER, M. 2000. Using presence questionnaires in reality. *Presence: Teleoperators and Virtual Environments 9*, 5, 497–503.