

# Content-rich and Expansive Virtual Environments Using Passive Props As World Anchors

Steven G. Wheeler  
steven.wheeler@pg.canterbury.ac.nz  
University of Canterbury  
Christchurch, New Zealand

Simon Hoermann  
simon.hoermann@canterbury.ac.nz  
University of Canterbury  
Christchurch, New Zealand

Robert W. Lindeman  
rob.lindeman@canterbury.ac.nz  
University of Canterbury  
Christchurch, New Zealand

George Ghinea  
george.ghinea@brunel.ac.uk  
Brunel University London  
London, United Kingdom

Alexandra Covaci  
a.covaci@kent.ac.uk  
University of Kent  
Canterbury, United Kingdom

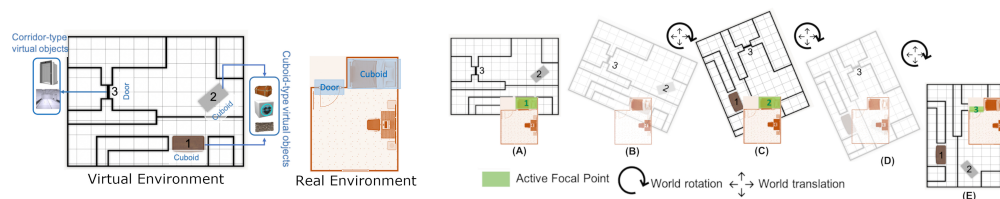


Figure 1: A demonstration of the virtual world translating/rotating to align different virtual and real objects.

## ABSTRACT

In this paper, we present a system that allows developers to add passive haptic feedback into their virtual reality applications by making use of existing physical objects in the user's real environment. Our approach has minimal dependence on procedural generation and does not limit the virtual space to the dimensions of the physical play-area.

### ACM Reference Format:

Steven G. Wheeler, Simon Hoermann, Robert W. Lindeman, George Ghinea, and Alexandra Covaci. 2021. Content-rich and Expansive Virtual Environments Using Passive Props As World Anchors. In *27th ACM Symposium on Virtual Reality Software and Technology (VRST '21)*, December 8–10, 2021, Osaka, Japan. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3489849.3489947>

## 1 INTRODUCTION

Since its inception, virtual reality (VR) has been primarily focused on the visual and aural senses. To enhance the sense of presence in a virtual environment (VE), researchers have since investigated the addition of touch via passive haptics [1, 2]. However, the creation of physical props to be used in VEs for providing passive haptics is often an arduous and time-consuming task. This motivated recent research to focus on procedural generation of VEs using the user's surroundings as a template [4, 5]. While procedural generation removes the need to make bespoke props, it adds limitations to the

development of rich, in-depth, story-driven experiences, lending itself more to the creation of labyrinth-like, random experiences. Furthermore, using the user's surroundings as a template for VEs, limits the virtual world to the size of the available physical space.

We propose a solution for this limitation – *Prop-Oriented World Rotation (POWR)* – which aims to assist developers in incorporating passive haptics into their VR applications with minimal reliance on procedural generation. While this technique only permits interaction with a single real/virtual object pairing at any given time, the virtual world is not bound to the size and dimensions of the physical play space as with related work [4, 5].

## 2 SYSTEM OVERVIEW

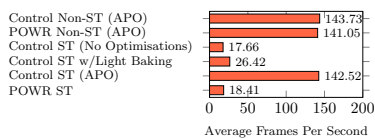
*POWR* consists of a series of scripts that developers can add to their *Unity* VR applications. For this, developers must define a set of virtual objects (by applying and configuring the relevant *POWR* scripts) that they wish to be physically reinforced by objects from the user's real environment. Prior to starting the main VR application, the user enters a separate VR application – the *POWR* calibration room – in which they define the real world objects they wish to be used to physically reinforce virtual objects in the VE.

While loading the main application, all virtual objects that have been previously designated by the developer to be used by *POWR* will be scaled to match the real world counterpart's size and dimensions. Once loaded, a single virtual object (chosen by the developer) is aligned with its real world counterpart by rotating and translating the entire VE (Figure 1). The user may then teleport to other designated points around the VE which, once again, enables *POWR* to rotate and translate the VE to align a new virtual/real object pairing.

Currently, each designated teleportation point only has a single real/virtual object pairing assigned to it thus the user is only able

to interact with a single pairing at a time. Furthermore, due to *POWR* requiring the real/virtual object pairings' positions to be synchronized, movement around the VE is limited to physically moving around the immediate play-area or teleporting between the designated points.

To test the effects on the user's feeling of safety and level of presence with and without *POWR*, we developed a demonstration game. Our game is divided into several areas where we placed different virtual objects the player can use to hide behind while fighting enemies. The player progresses through the game once they shoot all the enemies that are spawned in any given area. When the enemies are defeated, a teleportation point appears and, once players interact with it, the entire level is moved to align a new virtual object with its associated real object.



**Figure 2: Chart of average frames per second of all tests. ST - Stress Test. APO - All Possible Optimisations (as per Unity documentation<sup>1</sup>).**

### 3 EFFECT ON PERFORMANCE

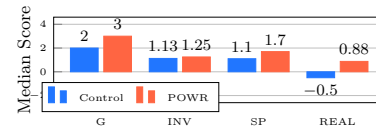
Due to *POWR* continuously moving the VE, many objects in the Unity scene cannot be marked as *static* – a requirement of the Unity engine to apply a number of common performance optimisations such as Light Baking or Occlusion Culling. As such, to test the significance of lacking such optimizations, we conducted a "stress test" in which we placed a very computationally demanding object in the scene and measured the average frame rate with and without *POWR*. The performance of both demos in the non-stress test scene had negligible performance differences (Figure 2). However, during the stress test, the inclusion of Light Baking in the control demo increased average frame rate significantly and with the inclusion of *Occlusion Culling* performance was on par with the non-stress test scene thereby emphasizing the importance of such optimizations.

### 4 USER STUDY

We conducted a user study where six participants experienced our demo with and without *POWR*/passive haptics. In the *POWR* condition, users were able to select two real world objects in their surroundings for in-game interaction. In the *control* condition, users experienced the game without haptic feedback but used the same designated teleportation points as the *POWR* demo (as opposed to being able to freely teleport anywhere in the scene). The mean age of the participants was 45, and four identified as male and two as female. The testing followed a within-subjects design with each participant testing both conditions, (*POWR* and *control*), in a randomised order. Due to the COVID-19 pandemic, the study was conducted remotely in three separate rooms with three participants in the first, a single participant in the second and two

<sup>1</sup><https://learn.unity.com/tutorial/optimizing-your-vr-ar-experiences#5e60ff9cedbc2a002071e0f5>

participants in the third; each participant only experienced a single room. After completing each experimental condition, participants filled in a questionnaire consisting of the validated and widely used Igroup Presence Questionnaire (IPQ) [3] and a customized safety questionnaire.



**Figure 3: Results of the IPQ Questionnaire – Generic presence (G), Involvement (INV), Spatial Presence (SP), Realism (REAL)**

Participants reported an enhanced feeling of presence in the *POWR* condition which is supported by the IPQ results (Figure 3) where the *POWR* condition had a higher median score than the control for the generic feeling of “being there”, Spatial Presence and Realism and similar for the attention paid to the VE (Involvement).

Regarding the feeling of safety experienced by participants in both conditions, all but one user answered that they felt “very safe” in the *POWR* condition. For the same question, only two participants awarded a maximum score for their feelings of safety in the control demo. This sentiment was reflected in the user feedback in which participants expressed a “*feeling of trepidation*”, with some noting they felt “*afraid to move*” despite the protection of the default *Chaperone* system.

### 5 CONCLUSION

*POWR* is a solid *proof of concept* which can be applied to VR applications, thereby allowing users to experience haptic feedback. The user evaluation was encouraging and the demo was successfully played in three distinct rooms of different sizes and layouts, thereby demonstrating its robustness and versatility. Although *POWR*'s performance in the stress test was less than ideal, with additional time, future iterations could work around limitations of the Unity engine and incorporate custom optimisation scripts, such as providing *Occlusion Culling*, which would significantly improve *POWR*'s performance.

### REFERENCES

- [1] Ken Hinckley, Randy Pausch, John C. Goble, and Neal F. Kassell. 1994. Passive Real-World Interface Props for Neurosurgical Visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Boston, Massachusetts, USA) (CHI '94). Association for Computing Machinery, New York, NY, USA, 452–458. <https://doi.org/10.1145/191666.191821>
- [2] H. G. Hoffman. 1998. Physically touching virtual objects using tactile augmentation enhances the realism of virtual environments. In *Proceedings. IEEE 1998 Virtual Reality Annual International Symposium (Cat. No.98CB36180)*. 59–63.
- [3] Holger Regenbrecht and Thomas Schubert. 2002. Real and illusory interactions enhance presence in virtual environments. *Presence: Teleoperators & Virtual Environments* 11, 4 (2002), 425–434.
- [4] Adalberto L. Simeone, Eduardo Velloso, and Hans Gellersen. 2015. Substitutional Reality: Using the Physical Environment to Design Virtual Reality Experiences. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) (CHI '15). Association for Computing Machinery, New York, NY, USA, 3307–3316. <https://doi.org/10.1145/2702123.2702389>
- [5] Misha Sra, Sergio Garrido-Jurado, and Pattie Maes. 2017. Oasis: Procedurally Generated Social Virtual Spaces from 3D Scanned Real Spaces. *IEEE transactions on visualization and computer graphics* 24, 12 (2017), 3174–3187.