

Breaking Binocular Redundancy Through Virtual Reality

Zongzhen Yang¹, Bryan E. Rubio-Perez², Mikhail A. Kats²

¹Department of Computer Sciences, University of Wisconsin–Madison, Madison WI 53706.,

²Department of Electrical and Computer Engineering, University of Wisconsin–Madison, Madison WI 53706

Abstract

The human visual system perceives the world by combining information received from convergence of two eyes and infer three-dimensional cues such as depth. Here, we demonstrate a virtual reality system that breaks the inherent binocular redundancy by projecting two different contents, both dynamically and statically, to each eye. We implemented the system using Unity3D game engine due to its virtual reality capability. We created two cameras in the virtual scene, representing the two eyes. Then we added identical but subtly contrasting objects that are visible to only one eye, so they appear overlapping when viewed binocularly and different when viewed monocularly, therefore creating binocular disparities. This technique can be used to simulate binocular rivalry and have implications on future virtual reality application designs.

1. Introduction

In the typical vision system, humans perceive the world by aligning both eyes to project two-dimensional images to two retinas and the brain infers depth through those images^{[1][2]}. In fact, binocular vision provides ecological benefits such as additional performance over monocular vision^[3]. Binocular rivalry occurs when the two perceived images differ, which could cause nausea, perceptual suppression and other physiological effects^{[4][5][6]}. The first study on binocular rivalry was conducted by Sir Charles Wheatstone, who developed a mirror stereoscope to elicit binocular disparities and the device is still being used today^[7]. There is a resurgence of growing interest on the subject of binocular rivalry over the last few decades and the devices used for simulation remain variations of the original mirror stereoscope^{[8][9]}. Some of the limitations of such devices include subjects are unable to look around and the displays are often confined to static images. Thanks to virtual reality (VR), some binocular disparity experiments were performed with head-mounted displays (HMD) instead of mirrors^{[10][11]}. With the recent development in VR display technology and VR-capable game engines such Unity and Unreal, researchers can look into combining the two to create immersive and customized binocular disparities. The following sections will entail the techniques and considerations to simulate binocular rivalry in virtual reality through Unity.

2. Background

This section provides a brief overview of Unity and its VR components. Unity is a cross-platform game engine for creating interactive media from movies to games^[11]. Its project can be 2D or 3D, and for virtual reality purposes, we developed our simulation in 3D. In Unity, a virtual world is called a *Scene*, and all the objects in it are referred to as *GameObjects*. Each *GameObject* has a *Transform* component, which

stores its spatial information such as position, rotation and scale. Developers can edit and manipulate objects by attaching C# scripts and editing properties directly through the *Inspector* panel. A key property for our application is *Layer*. Layer allows developers to indicate functionalities of certain GameObjects to interact with one another. To view a Scene and all the components in it, we need a *Camera* to project the world onto a display, in this case, our head-mounted displays. All properties are shown in Fig. 1.

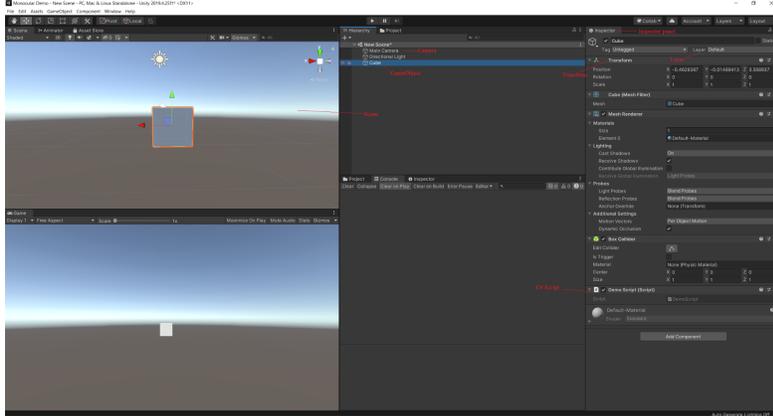


Figure 1: Unity overview: The Unity Interface

Unity has some existing device-specified VR plug-in providers for Oculus and Windows Mixed Reality; however, for this application to be compatible with most available HMD, we implemented the SteamVR Unity Plugin, which requires the SteamVR software. SteamVR has a feature to display the left and right eye views separately, which is very useful for experiment purposes, as shown in Fig 2.

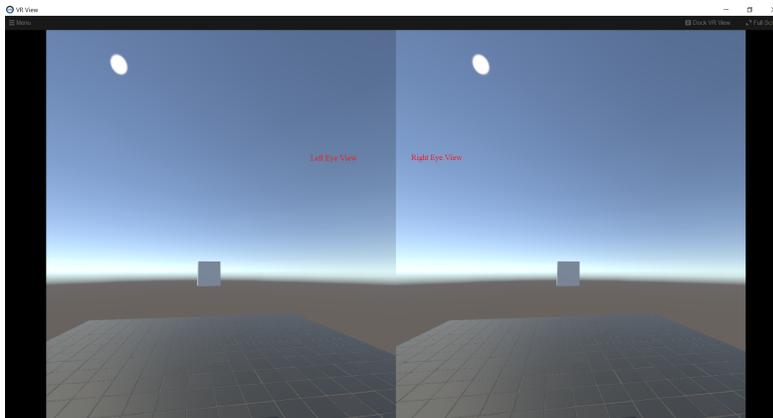


Figure 2: SteamVR's Two Eye Displays

3. Methods

In this section, we will discuss how we simulate static and dynamic binocular disparity using Unity3D and SteamVR.

Overview

Our approach to simulate binocular rivalry is to place two cameras that render to each specific eye. Thanks to Unity's layer mechanics, we set objects to only be visible to certain layers. By placing two identical objects with layers targeted toward each eye, the user will experience binocular disparities when viewing through the headset, as shown in Fig 3.

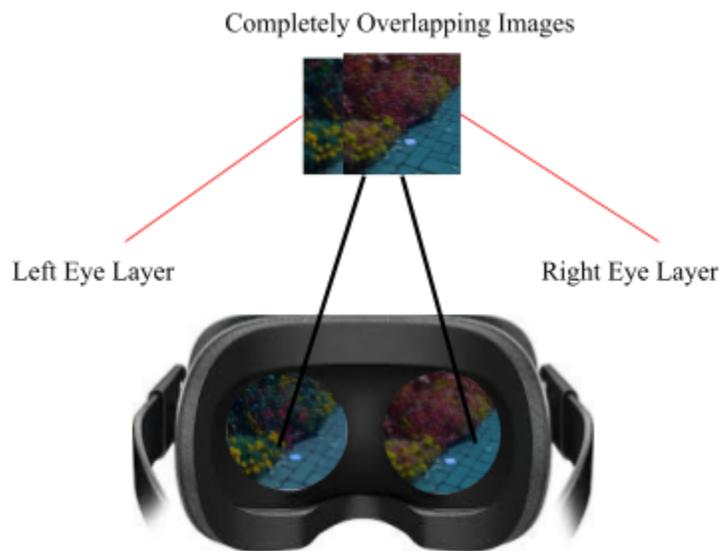


Figure 3: Visualization of our simulation approach

Static Images Binocular Disparity

We first achieved binocular rivalry in VR with static images by replicating outputs of mirror stereoscopes. To simulate the two eyes, we created two cameras called *LeftEyeCamera* and *RightEyeCamera* respectively and placed them in the same position and rotation. For each camera to render only to its respective eye, we changed the *Target Eye* to Left or Right. We then added two new layers called *LeftEye* and *RightEye*. Layers allow developers to set the visibility of the objects to a certain eye. To do so, we changed the culling mask of the cameras to see everything except the opposite eye layer (shown in Fig 4a). Next, we placed two identical but subtly different images in the exact same transform so that they overlap when viewed binocularly. Since Unity renders images through *Canvas* components, to have images be visible only to specific eyes, we set the *Event Camera* to the camera of the desired eye camera. Last but not least, we assigned the images to their according layers (shown in Fig 4b).

(a)

(b)

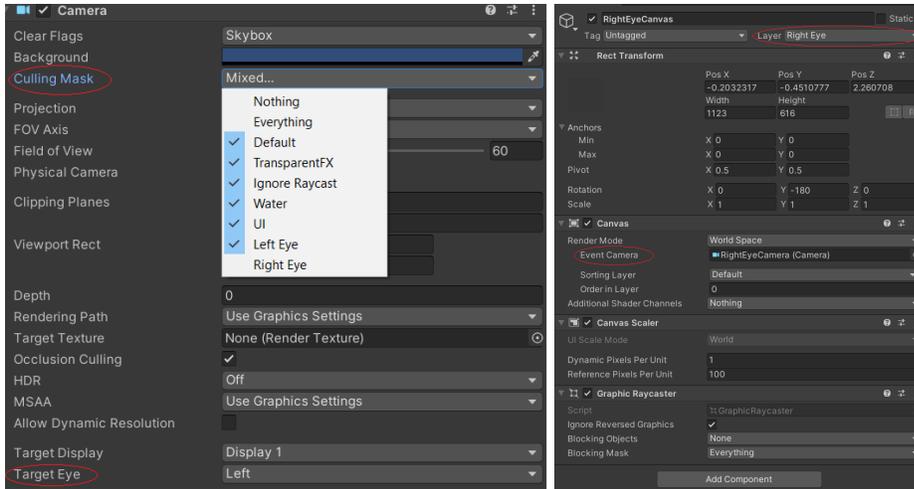


Figure 4: GameObjects Setup for Cameras, a) Camera Setup, b) Canvas setup for images.

If all settings are configured correctly, the two different images should be rendered to each eye, causing binocular disparity, as shown in Fig 5. Demo application and source code are provided in Supplement Material.

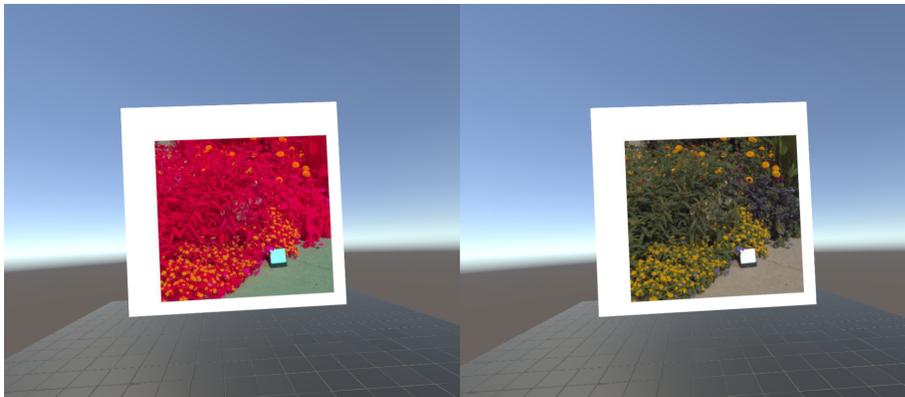


Figure 5: Left eye view and Right eye view of static image binocular disparity simulation

Dynamic Objects Binocular Disparity

Once we reproduced the output of mirror stereoscope through virtual reality, we implemented a variation of binocular rivalry with moving objects using the same technique. Similar to the previous simulation, we added *LeftEyeCamera* and *RightEyeCamera* and set them to render only to the respective eyes. Instead of images, we created two exactly identical cube GameObjects. The two cubes shared the same location, rotation, and scale so that they completely overlap when viewed with both eyes. Next, we set layers of the cubes to *LeftEye* or *RightEye* so they are only partially visible. To add movement to the cube, we attached a C# script to each cube and update its axis position using Sine function. A simple demonstration is shown in Fig 6 and complete implementation can be found in the source code under Supplementary Material section.

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class SimpleMovement : MonoBehaviour
{
    // Update is called once per frame
    void Update()
    {
        // Change Sin function to the desire axis
        transform.position = new Vector3(Mathf.Sin(Time.time), 0, 0);
    }
}

```

Figure 6: Simple Movement Script to control cube movement

If implemented properly, the two cubes should move along the desired axis and converge at one point. In our demo, the left cube is moving along the Y axis while the right cube is moving across the X axis, creating binocular rivalry, as shown in Fig 7. Complete implementation and source code can be found in the Supplementary Material section.

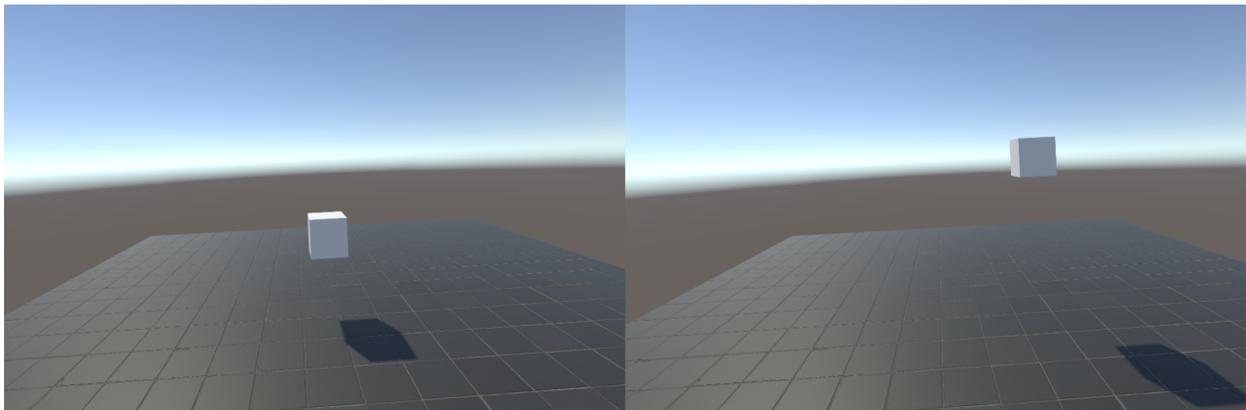


Figure 7: Left eye view and Right eye view of dynamic objects binocular disparity simulation

4. Conclusions

By breaking binocular redundancy in virtual reality, our method aims to provide new ways to customize and simulate binocular rivalry without the need for physical equipment such as a mirror stereoscope. In the present design, we created two cameras that render to each eye and layered virtual objects to only be visible monocularly. When the user views through a head-mounted display with both eyes, binocular rivalry occurs. Given the capability of game engines and virtual reality, it should be possible to create more complex and personalized binocular disparities using our approach. This technique yields implication to binocular disparity experiments, virtual reality comfort design, and potentially human vision enhancement^{[12][13]}.

Supplementary Materials

Note: SteamVR software and head-mounted display are required for the first two applications.

1. Demo Application for Static Image Binocular Disparity
<https://github.com/jackyangzzh/Monocular-Vision-VR-demo/releases/tag/v1.0>

2. Demo Application for Dynamic Object Binocular Disparity
(<https://github.com/jackyangzzh/Monocular-Vision-VR-demo/releases/tag/v1.0>)
3. Source Code for Demo Applications
(<https://github.com/jackyangzzh/Monocular-Vision-VR-demo>)

References

1. Wandell, B., & Thomas, S. (1997). Foundations of vision. *Psychocritiques*, 42(7).
2. De Valois, R. L., & De Valois, K. K. (1980). Spatial vision. *Annual review of psychology*, 31(1), 309-341.
3. Jones, R. K., & Lee, D. N. (1981). Why two eyes are better than one: the two views of binocular vision. *Journal of Experimental Psychology: Human Perception and Performance*, 7(1), 30.
4. Masson, G. S., Busetini, C., & Miles, F. A. (1997). Vergence eye movements in response to binocular disparity without depth perception. *Nature*, 389(6648), 283-286.
5. Blake, R., & Logothetis, N. K. (2002). Visual competition. *Nature Reviews Neuroscience*, 3(1), 13-21.
6. Frecska, E., White, K. D., & Luna, L. E. (2004). Effects of ayahuasca on binocular rivalry with dichoptic stimulus alternation. *Psychopharmacology*, 173(1), 79-87.
7. Wheatstone, C. (1838). XVIII. Contributions to the physiology of vision.—Part the first. On some remarkable, and hitherto unobserved, phenomena of binocular vision. *Philosophical transactions of the Royal Society of London*, (128), 371-394.
8. Blake, R., & Wilson, H. (2011). Binocular vision. *Vision research*, 51(7), 754-770.
9. Carmel, D., Arcaro, M., Kastner, S., & Hasson, U. (2010). How to create and use binocular rivalry. *Journal of visualized experiments: JoVE*, (45).
10. Mon-Williams, M., Warm, J. P., & Rushton, S. (1993). Binocular vision in a virtual world: visual deficits following the wearing of a head-mounted display. *Ophthalmic and Physiological Optics*, 13(4), 387-391.
11. Grudin, J. (Ed.). (2002). Rivalry and interference with a head-mounted display. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 9(3), 238-251.
12. Ellis, S. R., Breant, F., Manges, B., Jacoby, R., & Adelstein, B. D. (1997, March). Factors influencing operator interaction with virtual objects viewed via head-mounted see-through displays: viewing conditions and rendering latency. In *Proceedings of IEEE 1997 Annual International Symposium on Virtual Reality* (pp. 138-145). IEEE.
13. Gundlach, B. S., Frising, M., Shahsafi, A., Vershbow, G., Wan, C., Salman, J., ... & Kats, M. A. (2018). Design considerations for the enhancement of human color vision by breaking binocular redundancy. *Scientific reports*, 8(1), 1-9.