Improving the Perception of Relative Depth of Image-Based Objects in a Virtual Environment

Jooyoung Whang

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Science and Applications

Nicholas F. Polys, Chair
Douglas A. Bowman
Wallace S. Lages

June 26th, 2020
Blacksburg, Virginia

Keywords: Virtual Reality, Displacement map, Image-Based Rendering, Depth Cue

Improving the Perception of Relative Depth of Image-Based Objects in a Virtual Environment

Jooyoung Whang

(ABSTRACT)

In appreciation of High-Performance Computing, modern scientific simulations are scaling into millions and even billions of grid points. As we enter the exa-scale, new strategies are required for visualization and analysis. While Image-Based Rendering (IBR) has emerged as a viable solution to the asymmetry between data size and its storage and required rendering power, it is limited in its 2D image portrayal of 3D spatial objects. This work describes a novel technique to capture, represent, and render depth information in the context of 3D IBR. We tested the value of displacement by displacement map, shading by normal, and image angle interval with our technique. We ran an online user study of 60 participants to evaluate the value of adding depth information back to Image-Based Rendering and found significant benefits.

Improving the Perception of Relative Depth of Image-Based Objects in a Virtual Environment

Jooyoung Whang

(GENERAL AUDIENCE ABSTRACT)

In scientific research, data visualization is important for better understanding data. Modern experiments and simulations are expanding rapidly in scale, and there will come a day when rendering the entire 3D geometry becomes impossible resource-wise. Cinema was proposed as an image-Based solution to this problem, where the model was represented by an interpolated series of images. However, using flat images cannot fully express the 3D characteristics of a data. Therefore, in this work, we try to improve the depth portrayal of the images by protruding the pixels and applying shading. We show the results of a user study conducted with 60 participants on the effect of pixel protrusion, shading, and varying the number of images representing the object. Results show that this method would be useful for 3D scientific visualizations. The resulting object almost accurately resembles the 3D object.

Contents

List of Figures

List of Tables

List of Abbreviations

IBR Image-Based Rendering

VE Virtual environment

Image-Based Rendering is the technique for rendering 3-dimensional objects using two or more 2-dimensional images.

Virtual environment is an artificial environment that creates a sense of presence and immersion.

Chapter 1

Introduction

1.1 Motivation and Problem Statement

Due to the recent advancements in the field of High-Performance Computing, modern scientific simulations are scaling exponentially to inevitably reaching exa-scale ($\geq 10^{15}$ FLOPS). In scientific simulations, the visualization of the result plays a critical role in understanding and explaining the data. However, the storage bandwidth of these massive scale data is getting in the way of scientific advancement. It is expected that the hardware will not be able to scale larger simulations in the future. *In-situ* analysis is being proposed as the solution to overcoming this obstacle. This method proposes to process the analysis while the data is still in simulation memory. The key focus of in-situ analysis lies in preserving the important analytic information of the simulation while reducing the size of the data that needs to be stored.

As a part of these in-situ solutions, the Data Science at Scale team at Los Alamos National Laboratory proposed an Image-Based Rendering (IBR) method using a novel image database structure, called the Cinema database [2] [6]. In contrast to traditional model-based rendering that calculates the real geometry of a 3D model, image-based rendering utilizes a set of 2D images to render the perceived 3D model or the scene. Cinema presents a series of interpolated camera view snapshots around a simulation visualization (Example: [1]). Cinema can save the space and rendering power to view 3D simulation results. We denote

these objects rendered in a virtual environment as Cinema IBR objects. While this method can effectively solve the problem of rendering 3D visualizations with extreme data size in real time, no further research went into improving the technique's visual quality. While a 2D image can present 3D results, it cannot offer information about depth effectively due to lack of depth cues such as stereoscopy. Ideally, we would want an interactive visualization with real depth.

The focus of Cinema was to reduce the cost of saving and rendering large-scale simulations. The goal of this study is to improve Cinema's ability to show depth.

1.2 Proposed Solution

Bowman provides basis on why virtual reality is excellent for spatial information visualization and analysis ([28], [26]). By this regard, we specifically wanted to replicate and expand the method proposed by Cinema science in a virtual reality environment.

In scientific research, accurately representing the visualized data that is easy for researchers to discover patterns is very important. For inspecting 3D visualizations, it is not only important to the observers to be able to accurately perceive the silhouette and the surface colors of an object, but it is also equally, if not more, important to them to accurately judge relative depth. In this study, we improved the Cinema IBR objects by adding three depth cues that would improve relative depth judgment for the viewer. We applied varying relative size of front planes, shading, and motion parallax. Since simulation visualization results are often very complex and hard to conduct a user study with, we used simple 3D geometric shapes called geons. Details on the definition of geons is described in 2.3.2.

Relative size is a strong indication of depth. We apply relative size by adding protrusion to

the vertices of a subdivided plane with a Cinema image texture on the surface. Shading is a good depth cue, especially for comparing curved surfaces. We apply shading by calculating the normals from the protruded pixels. Motion parallax is another frequently used depth cue. We can apply motion parallax by showing interpolated snapshots from appropriate camera angles when a user's view point moves. We vary the degree of motion parallax by differing the number of camera angles in the Cinema database. Table 1.1 summarizes each depth cue and how they were applied to a Cinema IBR object. We call the techniques used to apply the depth cues as depth portrayal techniques.

Depth Cue	Application
Use of front surface relative size difference	Vertex protrusion using displacement map
Shading as an indication of front surface shape	WebGL shader & Normals calculation
Varying degree of motion parallax	Varying number of images in Cinema database

Table 1.1: Mapping of depth cues to depth portrayal techniques

The application of displacement maps to a 2D image is a popular way to add depth to an otherwise flat object. It is also a deeply studied topic with lots of optimization research done. In this study, We look at the effect of displacement maps in accurately representing depth to a user in a virtual environment.

In addition to adding real depth to the image, We have applied shading to the protruded geometry to add more definition to the rendered objects. Many 3D visualizations for scientific research purposefully eliminate shading to remove bias, especially if the colors on the surface of the geometry is of importance. However, shading is one of the key depth cues that a person uses to identify the shape of an object. If shape is the key feature of a visualized data, then shading is valuable.

In Cinema science's proposed method, the presented image of the data model changes according to the viewer's camera orientation. Snapshots of the object are taken across multiple angle steps along the delta (horizontal) and phi (vertical) rotations. This method is explained

with more detail in 2.1.2. A better representation of the data is possible when the images are taken from many angles, with a smaller angle step size. In the study, we also observe the effect of the number of images in perceiving relative depth of a Cinema IBR object.

We provide details and examples of how Cinema IBR objects appear with different combinations of depth portrayal techniques applied in 3.1.1.

1.3 Contribution

We report the results of a user study with 60 subjects testing their relative depth judgments using our depth-enhanced IBR object pairs to perform 2-AFC tasks. We conducted these 2-AFC tasks with Cinema IBR objects rendered using different combinations of three depth techniques: application of displacement map, application of shading, and varying the number of images (varying camera angle step sizes). The results provide insight on what parameters are useful for improving Cinema's visual qualities and relative depth perception for the viewer.

Due to the COVID-19 pandemic and an unfortunately timed update to the Oculus browser and the end of the WebVR API support, the initial plan to conduct the user study in Virginia Tech's Visionarium lab using VR headsets has pivoted to an online experiment in each participant's personal desktop environments. Accordingly, we had to exclude from the study a depth cue parameter that was planned to be tested: the application of stereoscopic rendering. If stereoscopic rendering had minimal effect on people's relative depth perception, one could decide to exclude it to further save rendering computation. We still report the implementations done to support VR (Chapter 3), so that anyone who would want to conduct the same study in a VR setting can use this work.

For this study, we expected the application of a displacement map to Cinema IBR objects to have the most effect on improving a viewer's depth perception, followed by shading, and finally the number of images. We also expected that users who have fully viewed the entire set of snapshots from all angles of a Cinema IBR object to have better relative depth perception of the object. Finally, we expected the more relative depth difference there exists between two objects, the less time it takes for a user to notice. The set of research hypotheses are described in 4.0.3.

To summarize, the main contributions of this study are the followings:

1. We improve the depth portrayal of Cinema's image-based method to render 3D scientific models with additional combinations of depth cues.

2. We provide the results of a user study testing the relative depth perception of users' Cinema IBR objects rendered in a virtual environment by applying three depth cues.

3. We present a better version of Cinema

Chapter 2

Background

In this chapter we provide the background and prior works that have been done related to this topic. Specifically, we will introduce Image-Based Rendering, Cinema, and depth perception. Related works suggest that stereoscopic view is very useful for depth perception. This motivated me to conduct this study in a VR environment before I, unfortunately, had to move to desktop.

2.1 Image-Based Rendering

The term Image-Based Rendering (IBR) first appeared in a paper by Chen [10]. Since rendering the full 3-dimensional geometry of an object was computationally expensive as well as required good hardware, Chen introduced a image-based method of rendering. Interestingly, IBR appeared for the purpose of using it in VR. In a image-based rendering setting, the computation of mapping the object's geometry into camera space and then computing the pixel values is no longer needed. In this paper, Chen surrounded the camera with images to emulate a 3D scene.

Shum and Kang categorize image-based rendering into three groups: rendering without geometry, with implicit geometry, and with explicit geometry [31]. In rendering without geometry or implicit geometry, IBR techniques try to construct a continuous plenoptic function of a scene or an object. A plenoptic function maps multiple properties of the camera

(i.e. camera position, angle) to the intensity of light rays passing through the camera center [5]. In other words, a plenoptic function of a camera defines all the rendered views possible over all camera orientation, time, and lighting conditions. Using a discrete set of images each representing a partial solution of the plenoptic function, many IBR techniques tried to estimate the rendered view at a new camera location and angle [11], [36], [29].

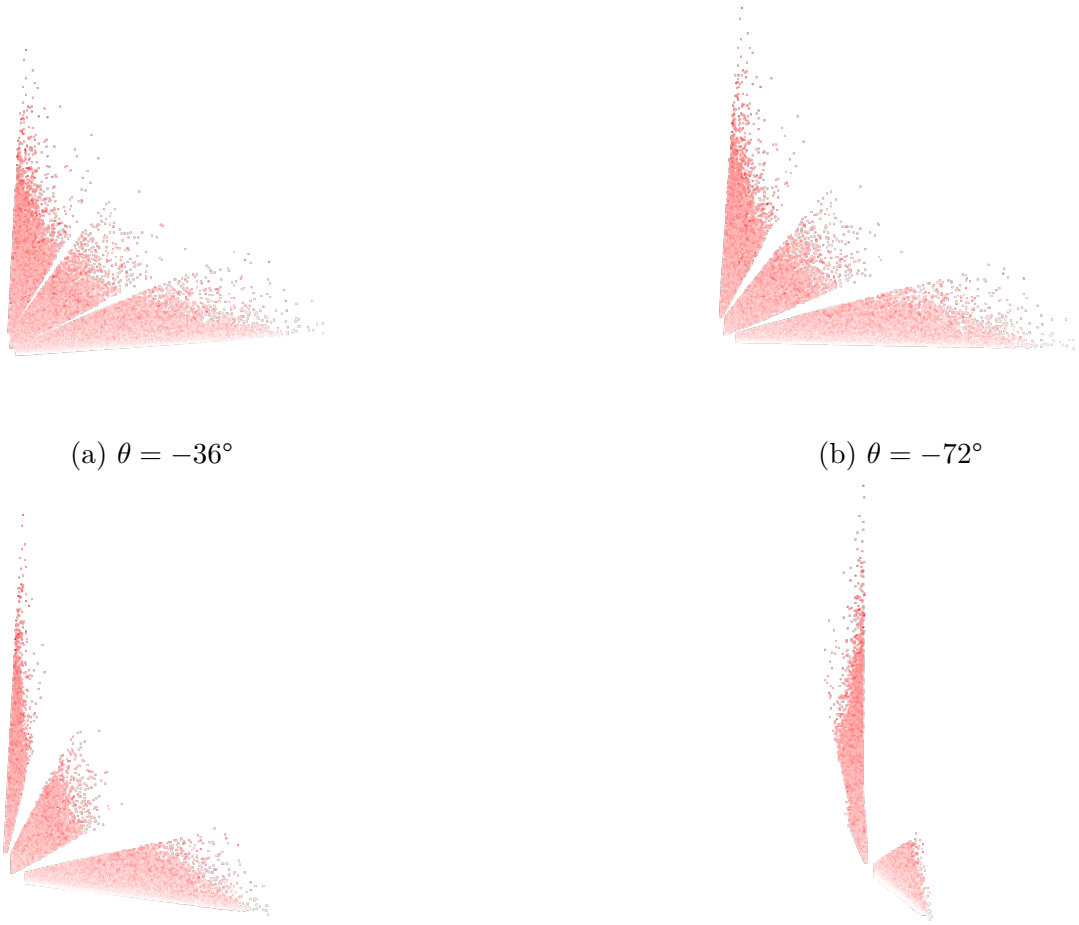
Rendering with explicit geometry requires depth information along with the image that serves a role similar to a texture on a 3D model. Techniques such as 3D warping or layered depth image (LDI) apply to this category [21], [30]. In this work, our technique is based on the idea of 3D warping. We try to physically extrude the surface of an image to a correct location in 3D virtual space.

Debevec shows a method of using image-based graphics and light probes to overlay virtual objects on real-world photograph [14]. This method requires radiance-information (lighting information) to correctly construct the objects. In this study, we calculate lighting by physically extruding an image and calculating the normals from the extruded geometry.

2.1.1 Cinema

Cinema is an in-situ image-based approach to handling extreme scale scientific simulation visualizations in real time [6]. Rather than trying to go through the heavy rendering process of the massive simulation at runtime, Cinema simply takes image snapshots of the visualization at multiple camera angles during the simulation (pre-rendered images). Using this method, the rendering process is simplified down to rendering a flat image. Cinema defines a database called the Cinema database, which is a collection of these pre-rendered images, stored according to the camera angle it was taken from. Figure 2.1 shows some images in a Cinema database of a Pion particle distribution from the HERMES experiment taken at

certain camera angles [3]. Cinema does not always require the user to export with multiple camera angles around a simulation result. Based on the need of the user, Cinema can save a single snapshot per simulation. Figure 2.2 shows a set of HERMES particle portraits exported with a single snapshot per particle type.

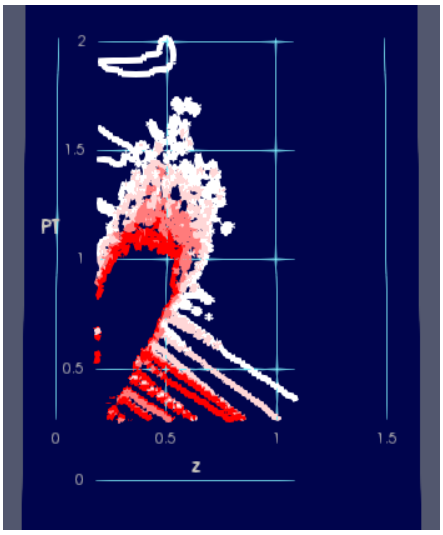


(a) $\theta = -36°$ (b) $\theta = -72°$

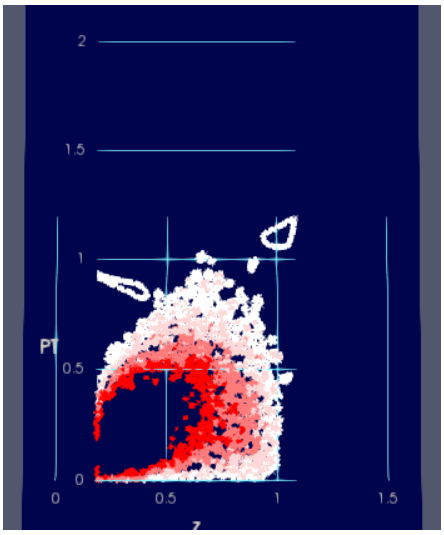
(c) $\theta = -108°$ (d) $\theta = -144°$

Figure 2.1: HERMES data visualization with Cinema exported using multiple camera angles

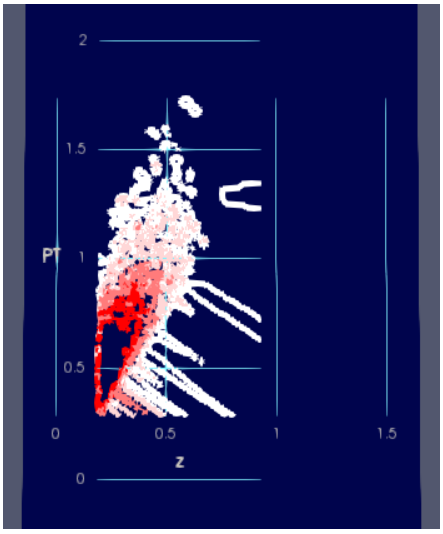
This way of rendering visualizations using images is extremely useful. For example, in Virginia Tech's Visionarium Lab, an application was developed called the Graphically-Linked



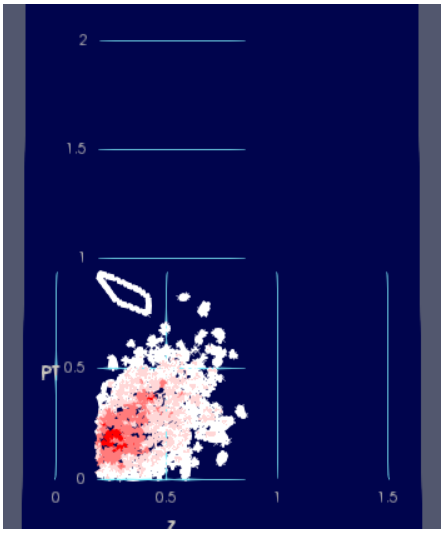
(a) Pi+ density contour plot



(b) Pi- density contour plot



(c) K+ density contour plot



(d) K- density contour plot

Figure 2.2: HERMES particle portraits with Cinema

Ensemble Explorer (GLEE), which allows a user to view an ensemble of Cinema image objects and perform semantic interaction to discover interesting relations between data objects and attributes [13]. A user can rotate the Cinema objects using a slider. Because the application uses images instead of the fully rendered 3D object, it is lightweight and very fast. The application can focus on attribute processing and statistics without allotting too much resource to rendering. While it is a great exploratory tool, it can be enhanced if the shown

images depicted the 3D geometry better.

2.1.2 Potential improvements to Cinema

The Cinema database is a good solution for modeling exa-scale experimental results in compact 2D images. However, some problems exist that can be solved to improve the approach.

Since Cinema stores a set of images taken at discrete camera angles, rendering continuity can be a problem, especially if the user is interacting with Cinema objects inside a 3D scene. One could try to take a snapshot at all possible camera angles (Assuming taking an image at one Euclidean degree angle, 360 * 180 images), but this increases the export time of the image and makes the database unnecessarily large. Szeliski introduces an interpolation technique that computes the interpolated views between images [33]. Two images each with a displacement map version is used to generate the interpolated views. A problem with this approach lies in the accuracy of the interpolated views. An interpolated view may deliver an incorrect observation to a viewer, which is dangerous in a scientific research. In this work, therefore, we keep the discrete steps between camera angles, but try to increase the definition of the object for each camera angle steps.

Another problem is the perception of depth. A flat image fails to show convincing 3D geometry to a viewer. This problem is minimal when a feature's pattern is 2-dimensional, but if a feature is inherently spatial, it can be hard to discover from a flat image. In this work, we try to accurately protrude the vertices of a subdivided plane with an image texture so that spatial depth information of the image is also captured and presented to the viewer.

2.1.3 Paraview

We used a software called Paraview to export the Cinema image database and generate the displacement map textures used in the user study.

Paraview is an open-source data analysis and visualization tool. This software lets the user define a pipeline of data filters to process the data and acquire a visualization. It supports exporting Cinema databases. In fact, this tool was what Ahrens et al. used in their proposal of Cinema. Figure 2.3 shows the interface of Paraview. The left-side tree diagram is the pipeline of filters that were used to render the visualization at the center. The bottom is the Python console where a user is allowed to run simple lines of Python code or run a script.

At the time of starting the image generation, Paraview had a bug with the Cinema export functionality. Since we also needed a way to export displacement maps that Paraview did not support, We implemented a Python script that exported a Cinema database from a point cloud data. More about the implementation is explained in 3.1.3.

2.2 VR in scientific research

Due to limitations in inspecting and interacting with 3D geometry on a flat monitor surface projection, new models of interaction are being proposed. Olshannikova et al. found wearable VR and AR to be an effective medium for supporting improved interactions with big data visualizations while stating that the technology is still lacking to support it [24]. Kreylos et al. regard effective interaction and real time high-definition visualization as the key conditions to enabling scientific research in VR [19]. A lightweight big-data rendering method like Cinema deployed in a VR or AR environment would be extremely useful in regard to performance. Again, the visual representation needs to be improved for this purpose because a flat image

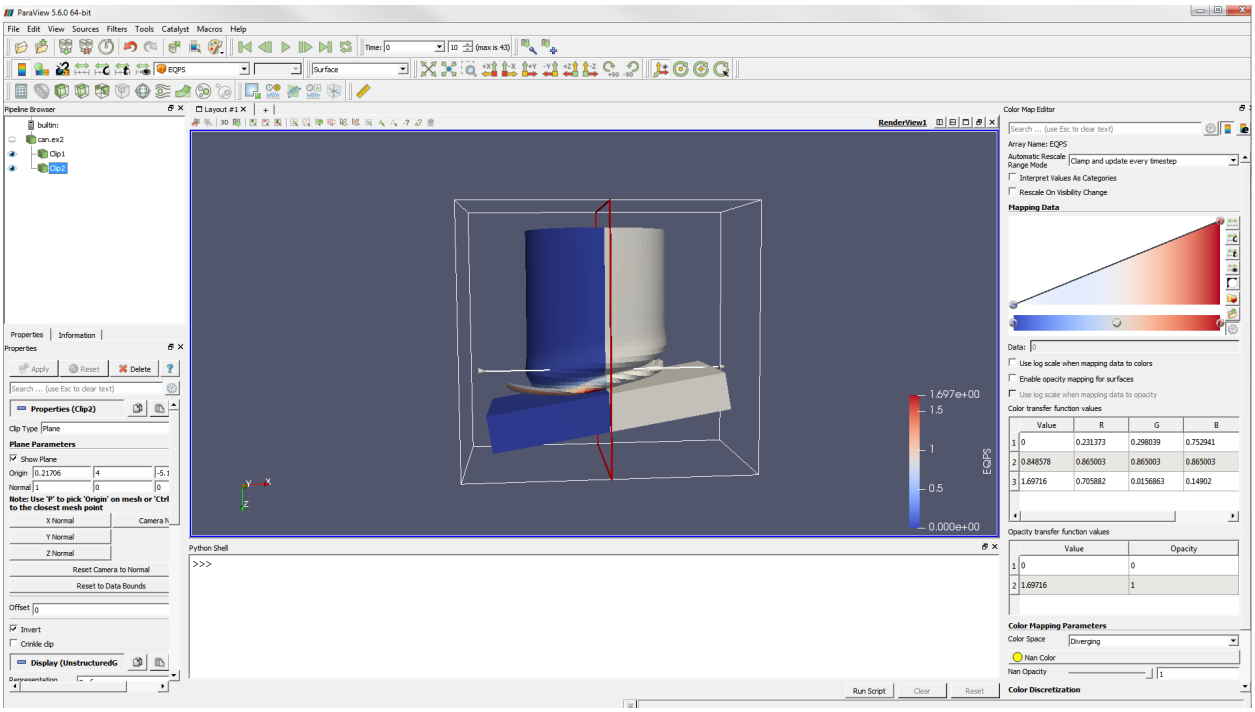


Figure 2.3: Screenshot of the Paraview interface

of the Cinema database is not enough to fully represent the 3D geometry.

Another popular option for VR to wearable devices is fish tank VR by Ware [35]. The idea of fish tank virtual reality was keeping the desktop environment but allowing the viewer to use head tracking as well as view the screen with stereoscopic rendering. Ware thought of this as a way to seamlessly integrate VR into everyday offices, but at the same time, it limits user interaction to the desktop metaphor. Studies suggest free hand gestures give more options and freedom of interaction for the viewer [25], [22]. To address this limitations, studies have been done to improve interactions with fish tank VR systems [27], [34].

Okura et al. tried to render an explorable MR scene using image-based rendering. They constructed multiple spheric images from structured viewpoints placed around the scene [23]. This work allows a good freedom of movement in the constructed world, but does not have object manipulation and interaction in focus (i.e. the focus was on rendering the scene, not

a specific object).

2.3 Depth Cues

Depth cues are hints that the human brain uses to recognize the depths, size, and locations of objects in the world. There are many depth cues available. Specific to an image or a movie, application of a displacement map is a method to add real depth information to a flat surface. In this work, we regard the use of this technique as a way to present relative size of Cinema IBR objects.

Cutting and Vishton describe that humans perceive distance from 9 sources of depth and this varies according to levels of distance [12]. We wanted to keep the depth application techniques to displacement map, shading, and number of camera angles (resembling motion parallax). In this study, we tried to eliminate other depth cues such as occlusion, viewer height, and aerial perspective. Occlusion was not used as a depth cue by placing objects next to each other, horizontally. If an object is below the viewer's height in visual field, the relative height of the objects can be used to infer the distance of the objects. To remove this depth cue, we place the camera at the same height of the objects for this study. Aerial perspective (difference in color due to the moisture in the atmosphere) was removed by simply not using it in the rendering. Cutting and Vishton did not consider shading as a depth cue but rather a shape cue. In contrast, Hubona et al. studied the effect of computer rendered shadows on human perception of a 3D object's position and size [18]. In their work, rather than simply toggling between shaded and non-shaded objects, Hubona et al. tried varying the number of light sources. Their work stated that one light in the stereoscopic rendering environment is an effective depth cue, but additional lights after this hinders depth perception. They also found stereoscopy is the stronger depth cue than shadows. In a future extended work with

VR, experimenting with stereoscopy as another depth cue condition would be interesting.

Cutting and Vishton mentions some of the depth cues dissipate with increased distance. One of them is motion parallax. Since this study was done in a desktop environment which is considered personal space (within arms reach), we can claim motion parallax can be fully used for sensing depth.

2.3.1 Displacement map

A displacement map imagery holds depth information coded with gray-scale color values. Displacement maps are often used in the context of computer vision. Depth cameras are able to retrieve depth information, allowing computers to detect edge, surfaces, and objects. A good application of displacement maps is Microsoft's Hololens device [4]. Hololens reconstructs the world around the user wearing it using depth sensors. This reconstruction can be used to naturally integrate virtual objects on top of the real world.

Other than inferring depth information from an image, the image itself can be enhanced using displacement maps. For example, Chen and Huang's work generates a displacement map from a static image and uses the displacement map to synthesize stereo images [9]. As another method, the pixels on an image can be protruded based on the depth information, and interpolating between protruded images can be used to resemble 3D objects [33]. This application of displacement maps is what we used to improve the visual quality and boost depth information of a Cinema database.

2.3.2 Geons and shape recognition

Biederman introduces the Geon theory, explaining that simple 2D and 3D geometric features are the essential building blocks for humans to recognize the shape of any object [7]. Geons are viewpoint-invariant, volumetric primitives. There are many geons such as bricks or cylindars. We specifically take note to boxes (or bricks) and spheres, as they are good basic geons that has either a flat or round surface. Casey and Exton took special notice to geon's property of being viewpoint-invariant and built an effective 3D UML diagram tool [8]. We put into focus a geon's property of being the building blocks of more complex structures. The use of geons simplified the effort of defining what geometry to use for depth perception in the area of scientific research. As geons exist in any complex shape, depth perception performance on geons can apply to complex shapes.

2.4 Summary

IBR techniques in the previous works have focused either on rendering the full scene using spheric images or using a series of orbital camera perspectives. While displacement maps are frequently used in the context computer vision and gaming graphics, IBR - and Cinema specifically- have note taken advantage of this capability to enhance the viewer's depth perception. In this study, we evaluate the combinations of three depth portrayal techniques (displacement map, shading, motion parallax) to improve the visual quality of Cinema images. As 3D objects can be decomposed into simple shapes called geons, we take two of the geons, a box and a sphere, as representative shapes to measure depth perception from different rendering combinations.

Chapter 3

Implementation

In this study, we conducted a user study involving a set of 64 2-AFC tasks. The 2-AFC tasks were done in a within-subjects fashion. The study application ran was ran on any internet browser at each of the subjects' desktops. For each tasks, the subjects were shown a pair of two Cinema IBR objects. One of the shown Cinema IBR objects contained a geon that had a longer protrusion towards the user. The subjects were asked to pick which one of the objects appeared to protrude more towards them. The presented Cinema IBR objects were rendered with various experimental conditions with different combinations of depth portrayal techniques applied. We explain the experimental conditions in depth in 3.1.1. We describe the 2-AFC tasks more in 4.0.1.

In this chapter, we introduce and describe the generation of geon images, and implementations of the depth portrayal techniques. We also describe how the components are put together to finish the application.

3.1 Approach

For this study, we assume an extreme-scale situation where rendering the full 3D object is impossible in real time. The focus of this study is improving Cinema's image-based method by applying various combinations of depth portrayal techniques to the database. The depth conditions we varied were:

- Application of displacement map
- Application of shading
- Varying the number of images in the Cinema database (Changing camera angle step size)

We explain all the implementations for each of these experimental settings for the application.

3.1.1 Experimental settings

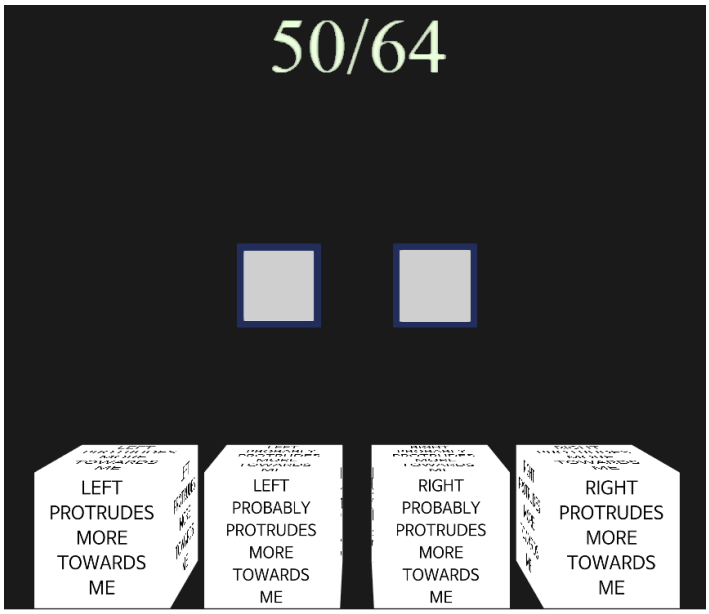
In this subsection, we describe each of the 8 experimental settings with screenshots. The binary number at the front of each subsubsection's title is the binary-encoded representation of each experimental settings.

(000) Displacement map *disabled*, shading *disabled*, number of images *few*

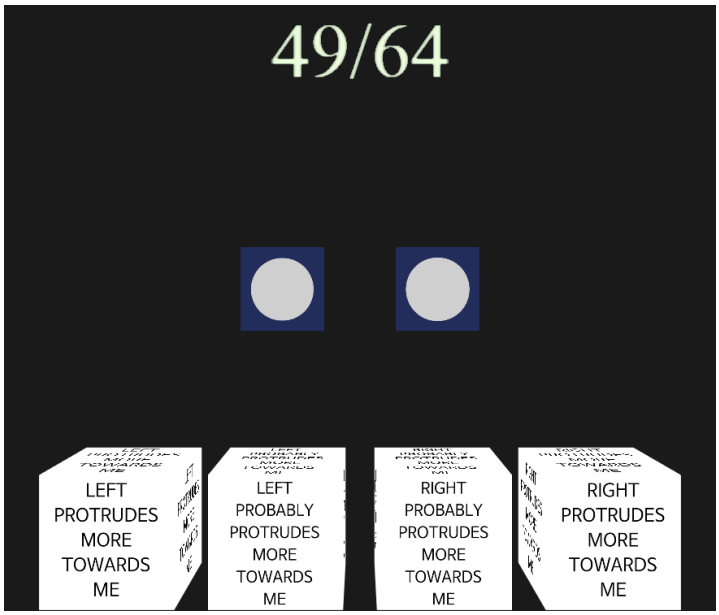
This setting serves as the baseline, where none of the depth portrayal techniques are applied (Figure 3.1). The Cinema IBR object rendered with this setting is simply a plane object with the texture as the Cinema image database. This object can be placed in a virtual environment as an ordinary 3D element. Based on the view camera's position and the Cinema IBR object's position, we set the texture of the Cinema IBR object with an image matching the viewing angle.

(001) Displacement map *enabled*, shading *disabled*, number of images *few*

This setting has the depth portrayal via vertex protrusion applied to the Cinema IBR object on top of the baseline (000) (Figure 3.2). We used a displacement map texture exported



(a) box (Right side has more protrusion)

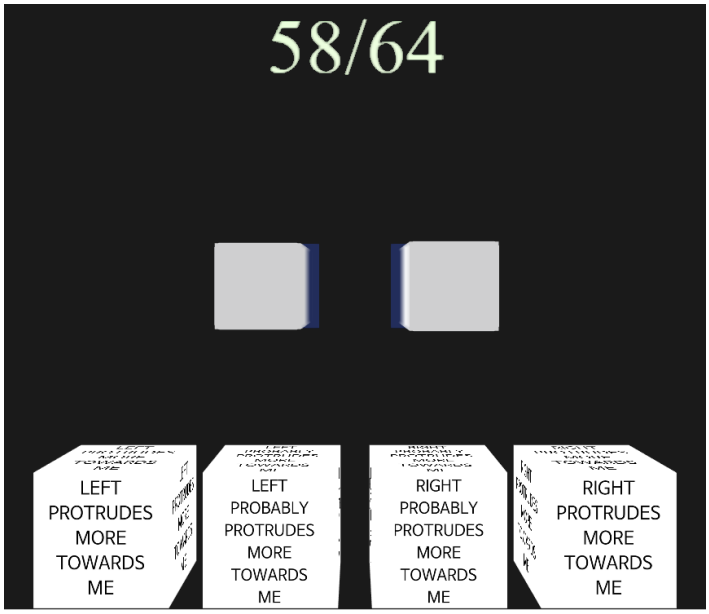


(b) spheroid (Right side has more protrusion)

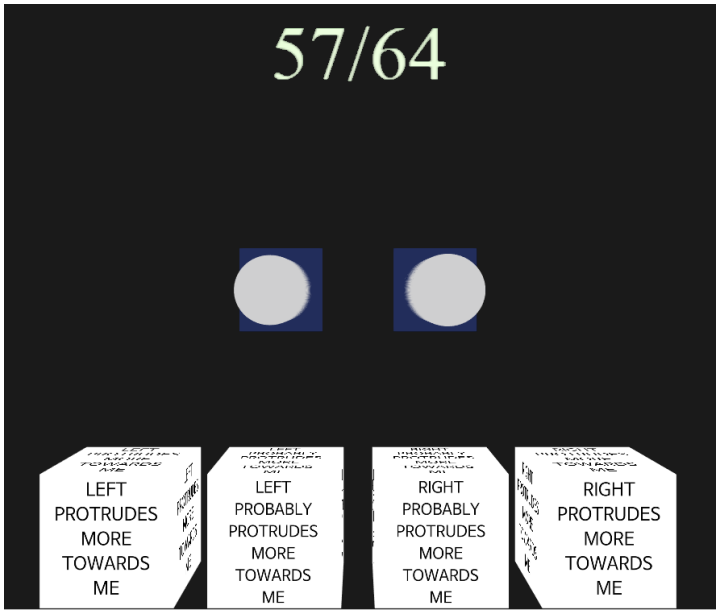
Figure 3.1: setting 000

along with the diffuse map of the geon to appropriately protrude the vertices of a subdivided plane representing the Cinema IBR object. We sampled the displacement map texture at appropriate UV coordinates to figure out how far to protrude each vertex parallel to the

local z-axis.



(a) box (Right side has more protrusion)



(b) spheroid (Right side has more protrusion)

Figure 3.2: setting 001

(010) Displacement map *disabled*, shading *enabled*, number of images *few*

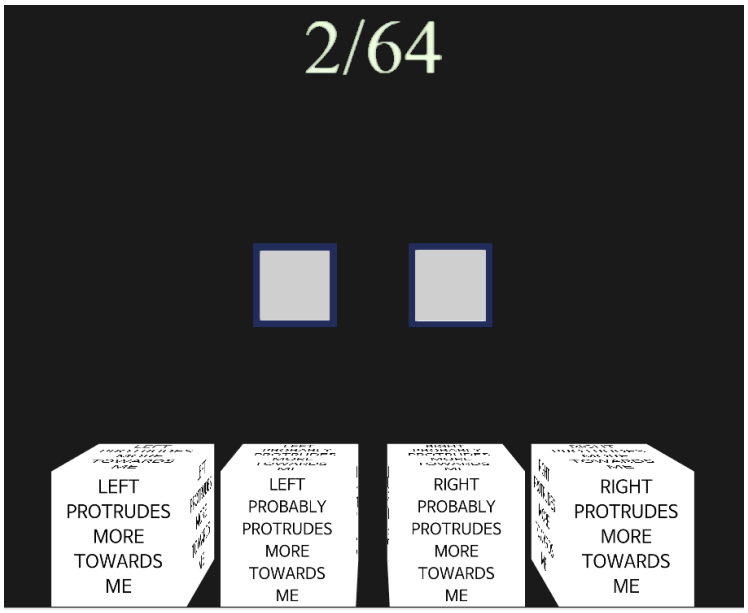
This setting has the WebGL shading applied to the Cinema IBR object on top of the baseline (Figure 3.3). We recalculate the normal vectors of the plane's vertices in the vertex shader using the protrusion level of each vertex. We placed a point light in the scene above the pair of Cinema IBR objects to use the light direction vectors in the vertex shader. This setting does not look different from the baseline because the displacement map is not used and therefore no vertex is protruded.

(100) Displacement map *disabled*, shading *disabled*, number of images *many*

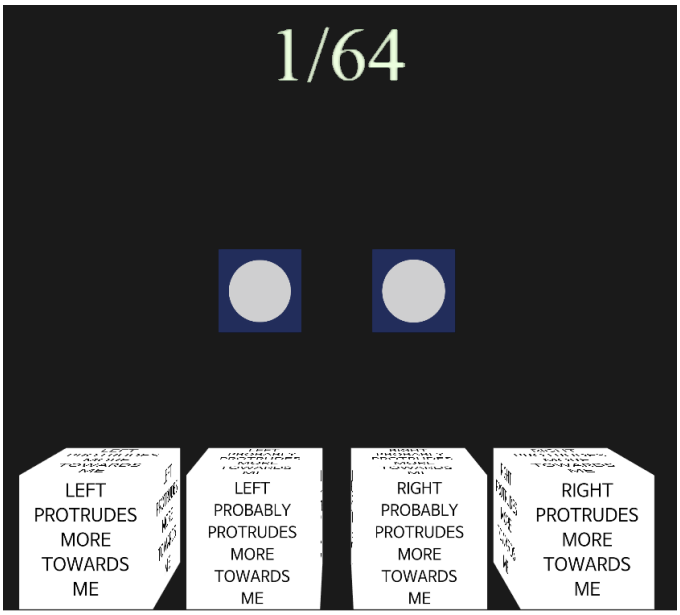
This setting increases the number of images used to present the Cinema IBR object with the rest of the settings the same as the baseline (Figure 3.4). As explained with the baseline, the Cinema IBR object shows to the viewer an image texture that was taken closest to the viewer's current viewing angle. These images are taken from the Cinema image database. For this setting, we use a Cinema image database generated using a smaller camera angle step size. The baseline's camera angle step size is 15°, while this setting's angle step size is 5°. For this study, we disabled movement along the vertical axis. Therefore, the exported images were taken horizontally around the object (The camera's vertical position was 0).

(011) Displacement map *enabled*, shading *enabled*, number of images *few*

This setting applies both vertex protrusion using a displacement map and shading. As the Cinema IBR objects using this setting has real geometry, shading starts adding definition to the object's shape (Figure 3.5). We found that shading added additional indication of the front surface's curvedness, especially for a spheroid.



(a) box (Right side has more protrusion)

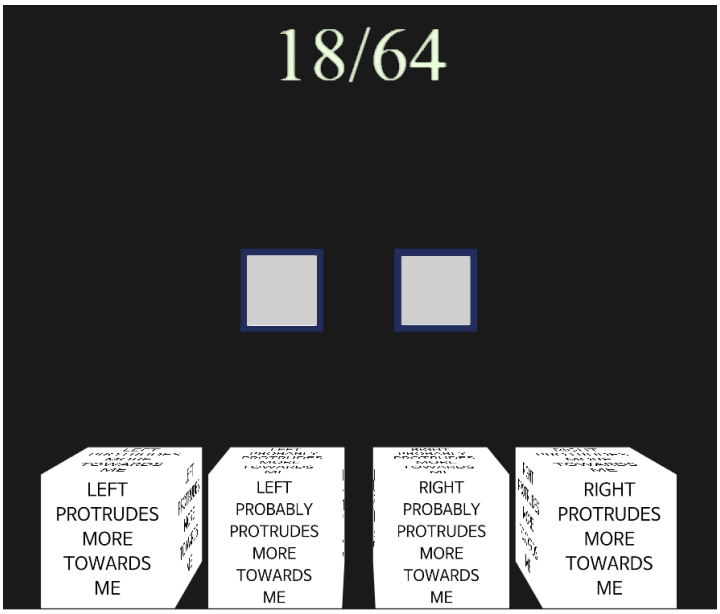


(b) spheroid (Right side has more protrusion)

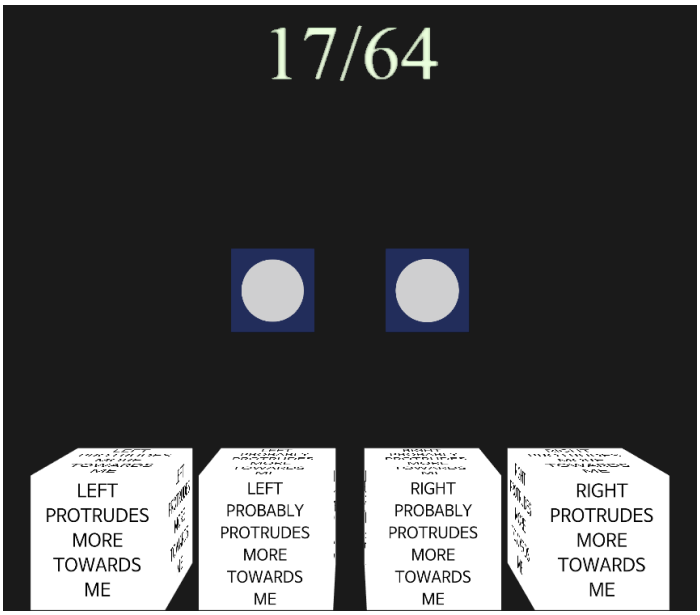
Figure 3.3: setting 010

(101) Displacement map *enabled*, shading *disabled*, number of images *many*

This setting increases the number of images in the Cinema image database used to represent the Cinema IBR object with the plane vertices protruded using the displacement map (Figure



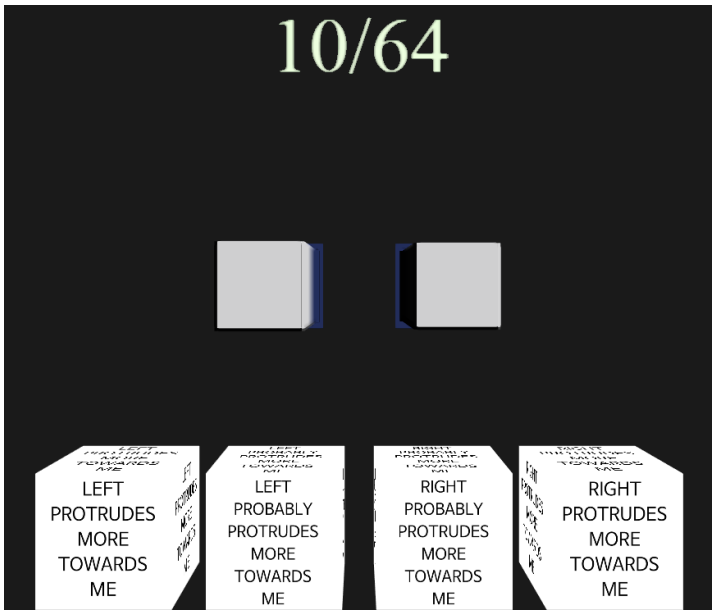
(a) box (Left side has more protrusion)



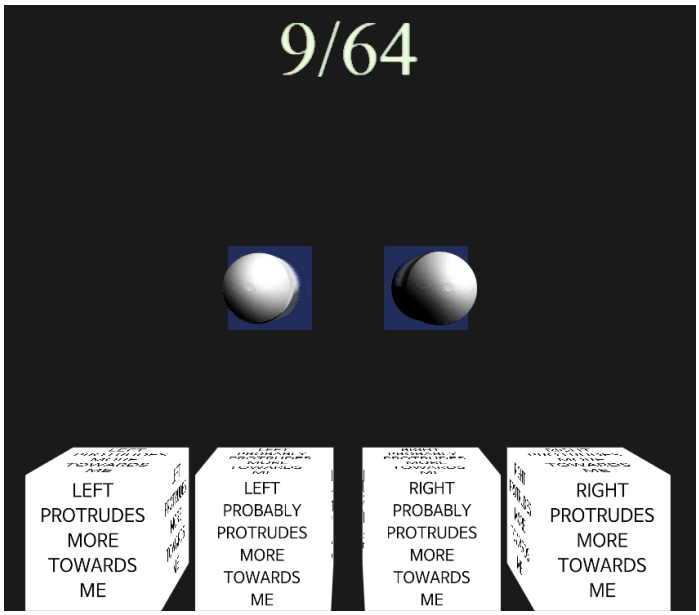
(b) spheroid (Right side has more protrusion)

Figure 3.4: setting 100

3.6). Since we were showing the diffuse texture with the closest view angle to the viewer without morphing in-between views, the Cinema IBR objects looked unnatural if the side of the objects were visible. Increasing the number of images in the Cinema image database



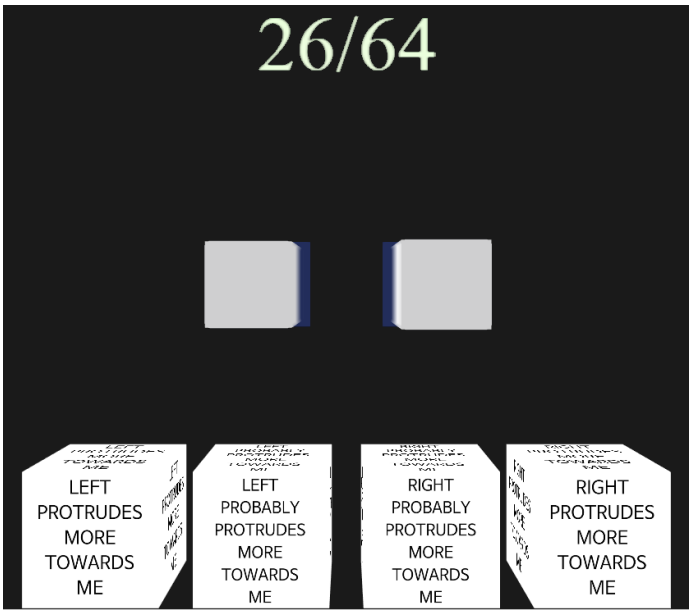
(a) box (Left side has more protrusion)



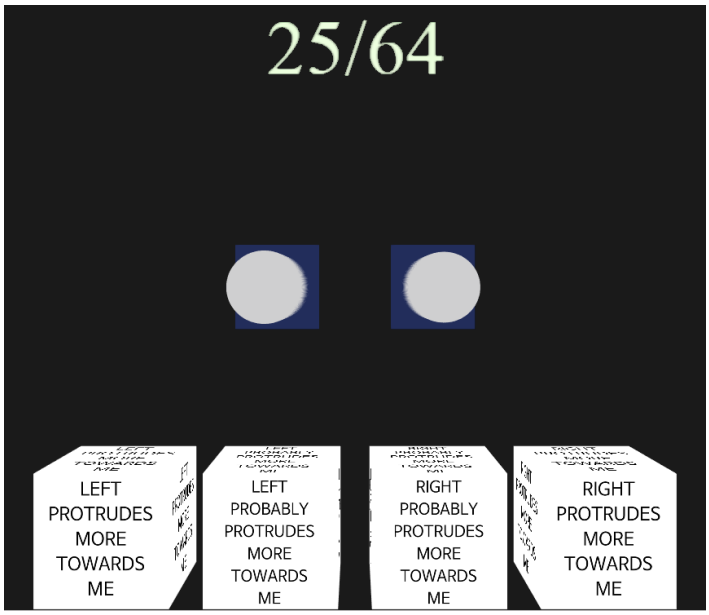
(b) spheroid (Right side has more protrusion)

Figure 3.5: setting 011

significantly decreased the occurrence of this problem, and the Cinema IBR objects appeared to better represent the 3D object.



(a) box (Right side has more protrusion)



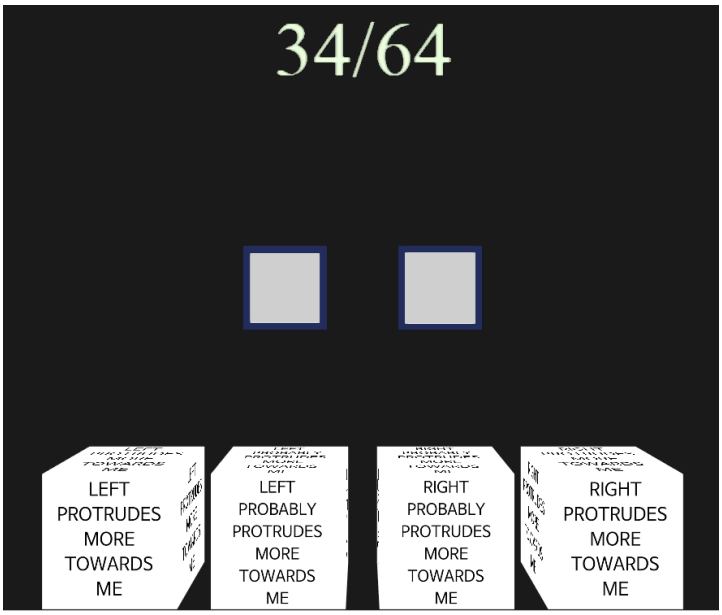
(b) spheroid (Left side has more protrusion)

Figure 3.6: setting 101

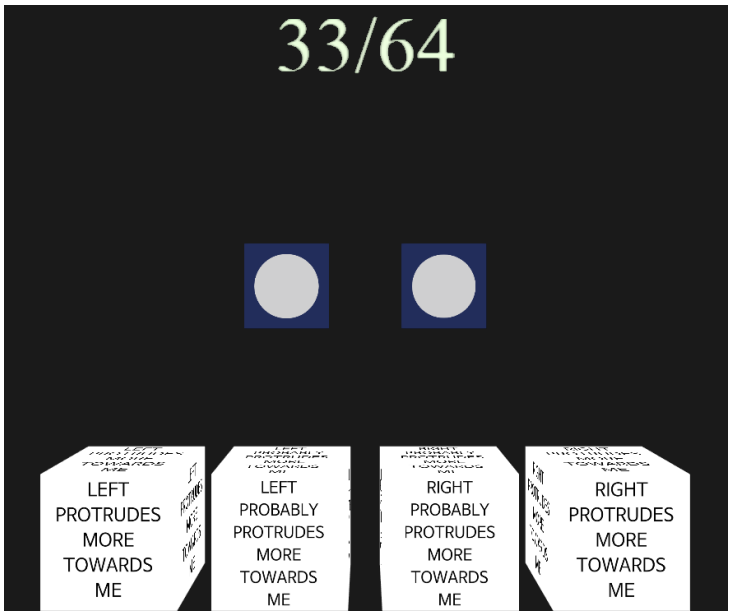
(110) Displacement map *disabled*, shading *enabled*, number of images *many*

This setting increases the number of images in the Cinema image database used to represent the Cinema IBR object with shading on (Figure 3.7). However, as vertex protrusion is not

applied to this setting, this setting closely resembles setting 100.



(a) box (Right side has more protrusion)



(b) spheroid (Left side has more protrusion)

Figure 3.7: setting 110

(111) Displacement map *enabled*, shading *enabled*, number of images *many*

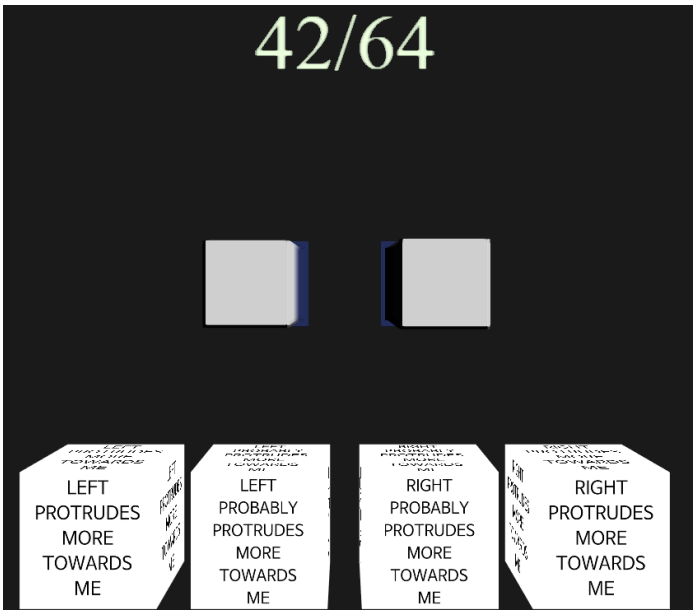
This setting applies all three depth portrayal techniques on the Cinema IBR object (Figure 3.8). The plane's vertices are protruded using the displacement map texture and the object is shaded with recalculated normals. The Cinema IBR object is represented using a Cinema image database with images taken every 5° horizontal camera angles.

3.1.2 Virtual environment and X3DOM

We constructed a web-based virtual environment for the application. The library we used for rendering the virtual objects is X3DOM. X3DOM allows developers to add 3D content to a WebGL canvas using the X3D standard. It also allows selection and manipulation of the X3D content using regular DOM operations using JavaScript. X3DOM also implements WebVR, an API to support VR content on the web. The functionality is very basic, however. To fully interact with the scene, some modifications to the X3DOM library had to be made. Therefore, we added in more features such as controller interaction to the source code of the library and used the recompiled version. While we have not used the added VR functionalities due to moving the user study to a desktop environment, it is still available for anyone to use. We provide a detailed explanation of what and how features have been added in 3.1.4.

Virtual Environment Set-up

We constructed a virtual environment where users could go through 2-AFC tasks. The virtual environment that we constructed consists of: the two IBR objects, four decision option buttons, and a black fade screen. We set a dark-gray (RGB 0.2 0.2 0.2) background to keep focus on the task objects. The black fade screen was set to fade in after a task was completed and fade back out after the new task was configured. The IBR objects were *1*



(a) box (Right side has more protrusion)

(b) spheroid (Right side has more protrusion)

Figure 3.8: setting 111

meter apart from each other in the x direction, and its center was placed *5* meters away from the user's eye position in the -z direction. To keep the users focus on the task, the 2-AFC option buttons were placed near the ground (1.5 meters below the IBR objects), where the

user needed to look down to locate.

We have performed a pilot study and interviewed a physicist that we had worked with previously to decide how to position and scale the Cinema IBR objects. We wanted the users unable to directly compare the relative scale (implying protrusion difference), so we placed them apart the same length as the objects (1m). Also, we did not want the users to view the objects from the side, which will give away which one is longer. To let the users fully use the information obtained from the front plane, we allowed the users move as far as they could view the objects from a 45° angle. The users were restricted to only move in the x-direction to prevent getting close to the objects. In the desktop environment, the users were not allowed to rotate the camera. We set the object's height equal to the user's view height.

3.1.3 Image Generation

The user study asks users to compare two IBR objects in a virtual environment. In this subsection, we explain how the images used to render the IBR objects were created. The user study required 10 IBR objects of different depths, and thus we needed 10 Cinema databases. Therefore, the same procedures were repeated with different parameters.

Geon Selection

As this study's main interest was on a person's relative depth judgment, we were not able to use real scientific visualizations which often have very complex shapes. We abstracted the shape of scientific visualizations' features into geons, which are the smallest components forming a 3D shape.

We selected two geons, a box and a spheroid, to be used in the presentation of the IBR

objects. As the users are restricted to viewing the objects from at most 45° on each side, the back geometry of the object was not of concern. During our review of common 3D visualized data, we noticed that many of the interesting features consist of either flat or curved surfaces. Spikes can be considered an extreme case of curved surfaces. Therefore, we selected each a box and a spheroid as they represent a flat and curved surface. While we claim that interesting features is a combination of flat and curved surfaces and therefore conducting the study using boxes and spheroids is enough, a future study using a real scientific visualization asking the participants to accurately identify interesting features is further needed to additionally confirm this approach's effectiveness.

Having the geometry alone made it difficult for a person to compare differences between objects. To provide a common reference point, we also included a back plane behind the geons. The back plane was positioned so that the back side of the geons touched the back plane.

Distance Percentage Difference Consideration

We performed a pilot study both in the real environment and a virtual environment. The purpose of the pilot study was to estimate the threshold for relative depth judgment.

In the real environment, we placed two non-textured cardboard boxes with a wall serving as the common reference plane. We tried to find the threshold for relative depth judgment by varying relative depth protrusion of the cardboard boxes. To change the relative depth protrusion, we moved one of the boxes forward or backward. We could perceive up to approximately 2% difference in forward protrusion. Taking this into account, we tried the same set-up in a VR environment. We constructed a virtual scene where two boxes with different protrusion levels were placed next to each other. The positions of the objects were

set up to replicate the pilot study done in the real environment. We used an Oculus Go to gauge the relative depth judgment threshold between the two boxes.

We noticed it was significantly harder to compare the difference with virtual boxes. During the pilot study, one person's threshold for perceiving depth difference was about 5% while the other person's threshold was 10%. We assume the different relative depth thresholds were caused by the additional depth cues that were available in the real world, such as detailed lighting and shading.

Using the resulting relative depth thresholds of the two investigators from the pilot study, we decided that the relative depth judgment threshold was within the interval between 5% to 10%. Therefore, we presented pairs of geons where the objects' depth percentage differences were 5%, 7.5%, 10%, and 12.5%.

Data Preparation

The depth calculation method we used specifically required the data type to be a point cloud data covering the surface of a geometry. For each cube and sphere, we generated a point cloud and stored it as a Comma-Separated Values (CSV) file where each row stored a point. To do this, we used Python version 2.7.15 without any additional packages or libraries. The points were sampled from the surface of the geometry by using parametric equations of planes and ellipses in 3D space. The CSV file contained 5 columns: ID, color, x, y, and z.

Points from five sides of a cube were sampled for generating cubes of various dimensions. We did not sample from the back face of a cube since it would never be visible to the camera. All cubes' x and y lengths were set to 1 meter and only the z length was varied across the different cubes. To preserve the density of the points throughout the data, a variable called *dpm* (dots per meter) was defined and used with a value of 500. To center the object at the

back plane, the points were translated half the z length forward. (Figure 3.9) The points for the back reference plane was also generated in this step, with a x and y length of 1.5 meters.

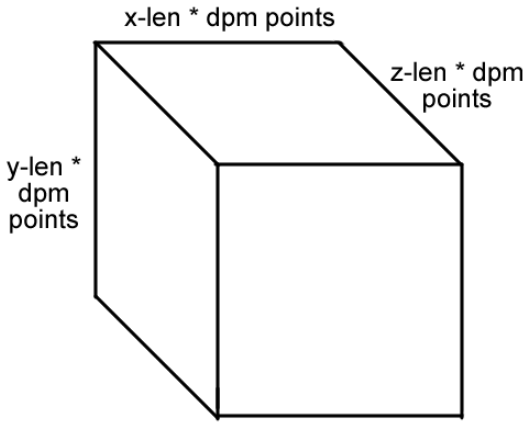


Figure 3.9: Number of points per side of the box

A similar method was used for generating point clouds of spheroids. The same variable dpm was used with a value of 500. The x and y length was set to 1 meters with only the z length varying across different spheroids. A cross-section of a spheroid at some y coordinate from a top-down view is an ellipse. A function called numPtAtY was defined to estimate the number of points required around an ellipse at a specific y coordinate. The function estimates the circumference of the ellipse by using Ramanujan's estimation given by

$$Circumference = \frac{(a+b)(1+3\lambda^2)\pi}{10+\sqrt{4-3\lambda^2}} \tag{3.1}$$

where

$$\lambda = \frac{a-b}{a+b} \tag{3.2}$$

and a, b are the major and minor axes of the ellipse. Then, this estimation is multiplied by dpm to obtain the number of points required for the cross-section at a specified y-coordinate.

Taking the cross-section of the ellipsoid cut by the y-z plane, the number of points needed along the y direction would be half the number of points needed for the cross-section ellipse. Since this ellipse has exactly the same shape as the cross-section ellipse taken at y coordinate 0, we can use the following formula to determine the number of cross-sections needed. (Figure 3.10)

$$numCrossSection = \frac{dpm * numPtAtY(0)}{2} \tag{3.3}$$

The sampled points were translated half the z length forward to center the object at the

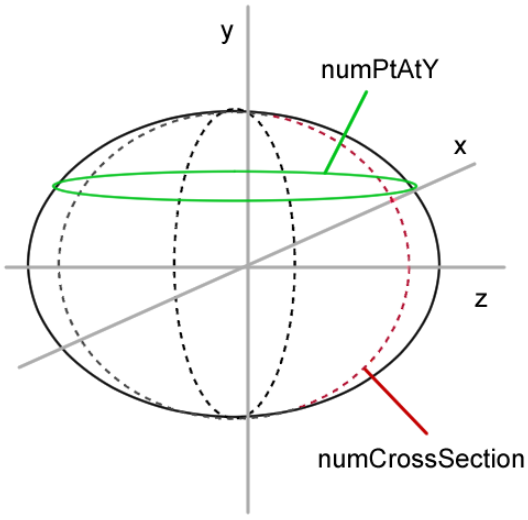


Figure 3.10: Number of points along cross-sections of the ellipsoid

back plane. Same as with the cube, points for the back reference plane was generated with a x and y length of 1.5 meters. The resulting point clouds had smooth surfaces without any noticeable empty spots.

To make the front planes of the geons look more noticeable, we assigned different colors for different parts of the point clouds. More specifically, an integer value of either 0, 1, and 2 was assigned to non-front planes, front plane, and back reference plane, respectively. This assigned value was used as a key to color the points differently in Paraview. The front plane

of the spheroid was defined as the front hemispheroid. Figure 3.11 illustrates this color scheme. The plane colors were opaque without transparent faces.

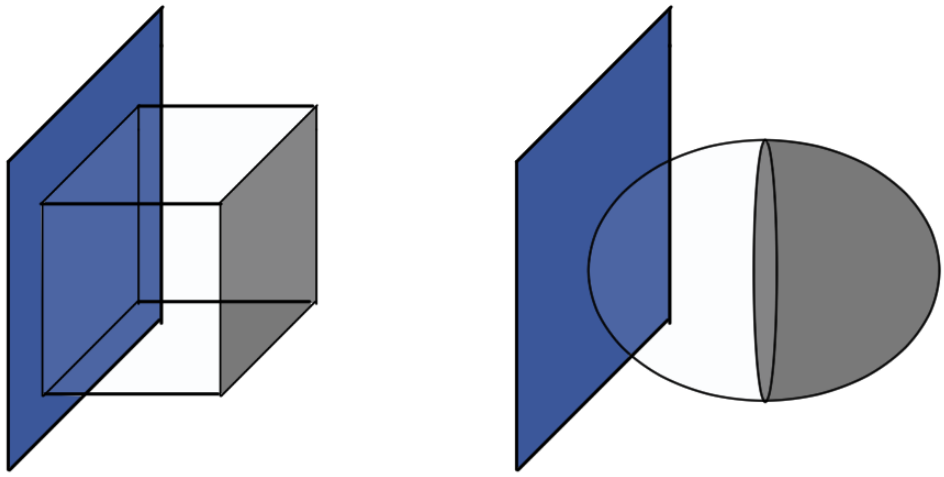


Figure 3.11: Differently colored planes. Blue = back reference plane, white = non-front planes, grey = front plane

Cinema Database Exportation

To generate the images needed for the IBR objects, we loaded the synthesized data from 3.1.3 into Paraview. We used Paraview version 5.6 to export the images. The following is the pipeline.

1. Load CSV file
2. Convert from table to points
3. Save a screenshot of a color image and a depth image
4. Take screenshots for all angles.

To automate the process, we wrote a Python script and ran it using the Python shell in Paraview.

To make sure that the entire geometry was captured within the images for all camera angles, the minimum camera distance to fit the object was calculated for all camera angles. Then, the maximum distance was used to capture all the images and preserve uniform scale throughout the multiple angles. Additionally, as this distance was different between different objects of varying z length, the camera distance used for the longest object was used to capture all other objects' images.

Paraview can color the created visualizations by a chosen variable, using either one of the default color mapping schemes or a custom one. We chose the column *col* in the synthesized data to color by and defined a custom color mapping scheme that colored white for value 0, light-grey for value 1, and dark-blue for value 2.

To generate the depth images, we measured the distance for each point in the point cloud to the camera. Based on the distance, the point was colored in grey scale where black means furthest away and white means closest to camera. To ensure the value (brightness) range stayed consistent across all images, the minimum and maximum distance across all points and across all camera angles were pre-calculated and used to determine a universal value range.

The exported images were named according to whether it is a color or depth image, and what camera angle it was taken from. Both color and depth images for the same geon was saved in the same directory. All images had the resolution of 512px x 512px.

3.1.4 Modification of the X3DOM Library

For this study, we used X3DOM version 1.18.1 and customized it for this study's specific needs. This subsection explains what and how some of the features of X3DOM were modified or added. The detailed implementation can be accessed from the supporting documents.

Navigation Via Oculus Go Controller

The original navigational method was similar to the FLY mode on a desktop, where the camera moved based on the user's head orientation. To fly around in the environment, a user needed to press on the Oculus Go controller's thumb pad. Based on which corner of the thumb pad the user's finger is on, the camera moved in one of 4 directions: forward, backward, left, and right relative to the head orientation. However, this made it difficult for the user to look around the environment while moving in one direction. This was a critical component of this study. Therefore, a modification had to be made.

Processing user input via a game pad controller is implemented in a script called *VRController.js* in the util directory of X3DOM's source. In the update function which gets called every frame, we have altered the code so that the controller's orientation is considered instead of the head. Also, we made the user only move in the direction of the controller regardless of which part of the thumb pad the user was pressing on to reduce the learning curve for navigation.

Restricting Movement

In addition to changing the navigational method, we needed to restrict movement and maximum distance the users could travel from the objective so that they would not accidentally lose the target objects.

We modified *VRController.js* so that only the x component of the controller's orientation is passed to the movement update function. This restricted the user to only be able to move in the horizontal direction relative to the IBR objects.

We set the maximum distance away from the IBR objects in the horizontal direction to be 5 meters. We set this distance because the widest supported angle of the IBR objects was 45°.

To detect the user's head position, we used a given event called by X3DOM called *onViewpointChange*. When entering VR, X3DOM creates two new view matrices and projection matrices to use for the left and right eye. These matrices' translation are set to the origin point, independent of the active view point position. This creates a gap between the acquired user position from *onViewpointChange*'s event parameter and the local matrix translation. To go around this problem, we set a grouping translation node containing all the IBR objects and moved this instead of restricting the user's movement. Whenever the user got too far from the IBR objects, the IBR objects were positioned to the maximum threshold distance away from the user.

As we had to port the study to a desktop environment, we also added in support for keyboard interaction. The distance limit was kept the same. We checked every frame if the user was pressing on the left-arrow or right-arrow key, and accordingly translated the viewpoint.

VR Button Implementation

X3DOM supports object selection with the mouse cursor in the desktop environment but there is none for WebVR. Therefore, we created a new script called *VRButton.js*. We modified *VRController.js* to dispatch an event called *TriggerClicked* whenever the trigger button of the controller is pressed. *VRButton.js* iterates over all objects whose HTML class is tagged

VRButton to calculate the distances of these objects to a ray protruding from the front of the controller. If the closest object was less than a certain threshold away, we called OnClick on that object.

3.1.5 Depth Map

WebGL Shader

There is a X3D node called *ComposedShader* which we could use to add GLSL codes for rendering a plane object. We implemented a vertex shader and a fragment shader for the depth-enhanced IBR objects. The Cinema IBR object starts with a simple textured plane that is a 1m x 1m plane with 512 x 512 subdivisions (vertices on each side). The vertex shader displaces each vertex appropriately, and the fragment shader colors and shades the new geometry.

The vertex shader takes a pixel from the depth image as input. We use the UV map of each vertices to acquire the pixel color in the displacement map. Based on the value (brightness) of the pixel, the shader protrudes the vertex in the local z direction by a coefficient float which we can change programmatically at runtime. We set this coefficient to an appropriate value when we wanted to enable the displacement map, and set it to 0 when we wanted to disable the displacement map.

Another purpose of the vertex shader is to calculate the correct normals for the geometry. Because the original object is a simple plane with a diffuse texture, the original normals for the Cinema IBR object's vertices point in the local positive z-direction. This becomes a problem when we try to shade the IBR objects after it has been protruded to resemble 3D geometry. Therefore, for each vertex, we sampled four nearby points in the displacement map. We sampled the four points located in -x, +x, -y, and +y directions relative to the

current vertex. Ideally for complete accuracy, we would want to sample points that are a pixel away from the current point. However, we found that sampling too close points resulted in jagged moire patterns. Therefore, we sampled points about 3 - 4 pixels away in the displacement map. Using the acquired pixel values, we constructed four nearby point coordinates and used the four positions plus the current vertex position to calculate the normal vector for the current vertex.

Using two of the four points and the current vertex position, we can form a triangle, such as the one shown in figure 3.12. Using the triangle, we can calculate the normal vector. For example, taking the cross product of v_{y+} and v_{x+}, we can get a normal vector of triangle $p_{y+}vp_{x+}$. Using a similar method, we ended up with four normal vectors. We used the average of the four normal vectors to get the final normal vector for a vertex (Figure 3.13).

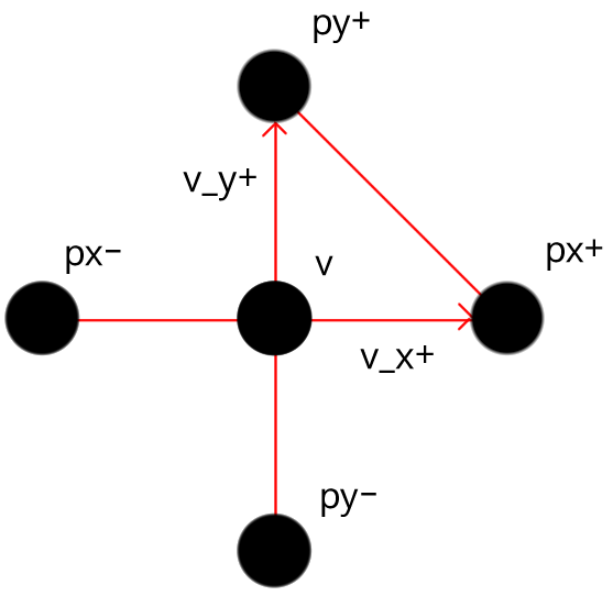


Figure 3.12: Two of the four near points forming a triangle

The fragment shader simply takes in the diffuse map (the colored image texture) and maps the color values on the protruded verices. We made the shader take in the alpha channel of the texture as well, so that empty spaces were rendered transparent. The fragment shader

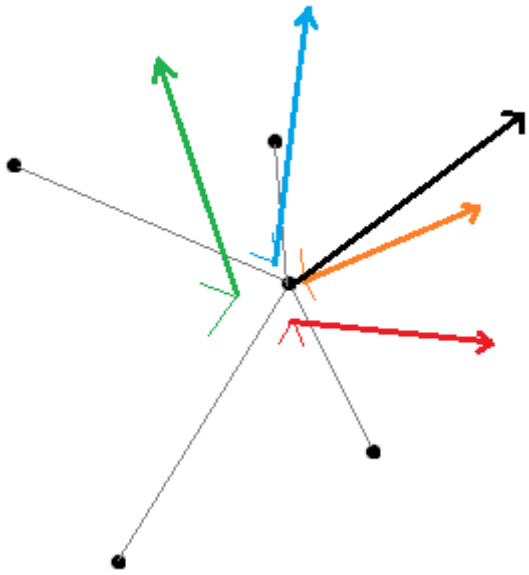


Figure 3.13: The black vector (normal of the point) as a result of averaging the four colored vectors

also uses the normals calculated from the vertex shader to calculate the shading for the object. The shading formula is simply given by the following (3.4).

$$diff = -light\_direction \cdot normal \tag{3.4}$$

We multiplied this value to the color vector to produce the final colors.

3.1.6 Stereoscopic and Binocular Rendering

X3DOM Canvas

Stereoscopy was one of the depth cues to vary before moving to a desktop environment. Therefore, we implemented an option to toggle on and off stereoscopy. When entering VR, X3DOM takes the projection matrices and view matrices for both eyes to render the scene

on two canvases. These are then combined into a single rendered image and sent to the headset for stereoscopic rendering. To allow us to toggle stereoscopic rendering, we made a global boolean variable that set to *false* feeds the left projection matrix and view matrix for both eyes.

3.1.7 Varying the number of images

Changing Angle Steps

When the user moves to a new location in the environment, a Javascript compares the user's angle with regard to the Cinema IBR object. It then locates the image taken from the closest angle, and feeds it to the shaders introduced in 3.1.5. The Cinema IBR object is also rotated in the opposite direction. We provided two different angle step sizes, 15° and 5°, which would differ the degree that motion parallax can be used to determine depth. We made the system look for the images by taking into account what angle step is currently being used.

Relation of the Number of Images with Motion parallax

Motion parallax is a depth cue where the difference of two objects' change in orientation according to user movement is used to sense depth. In regard to Cinema, less image, or large camera angle step, means the changes will be abrupt and big. More image allows smoother transition, so the changes will be subtle and natural. We assume there is a relation between number of images and motion parallax, and that is why we included the number of images as a variable.

Chapter 4

Evaluation

In this chapter, we introduce the user study design and the evaluation methods that we used. The objective of the user study was to observe and compare the effects of added depth cues to traditional IBR objects in respect of relative depth judgments. We also tried to compare subjects' relative depth perception between geons with flat and round surfaces.

4.0.1 User study design

As mentioned in the introduction, the original user study was intended to be conducted in a VR environment. WebVR was a VR API developed for Firefox and Chromium-based browsers to support VR on the web. The application for this study used X3DOM's implementation of WebVR to run on the Oculus Go or the Samsung Gear VR head-mounted displays. Both of these devices allow Internet browsing using a default Chromium-based browser called the Oculus Browser. As of Oculus Browser v.9.0 WebVR support was officially dropped and the application was no longer supported until X3DOM ported WebVR to WebXR. Therefore, the following user study design was done where subjects interacted in a desktop/laptop web browser.

The user study consisted of multiple 2-Alternative Forced-Choice (2-AFC) tasks where the user was asked to pick from two IBR objects which one he or she thought was protruding more towards him or her (Figure 4.1).

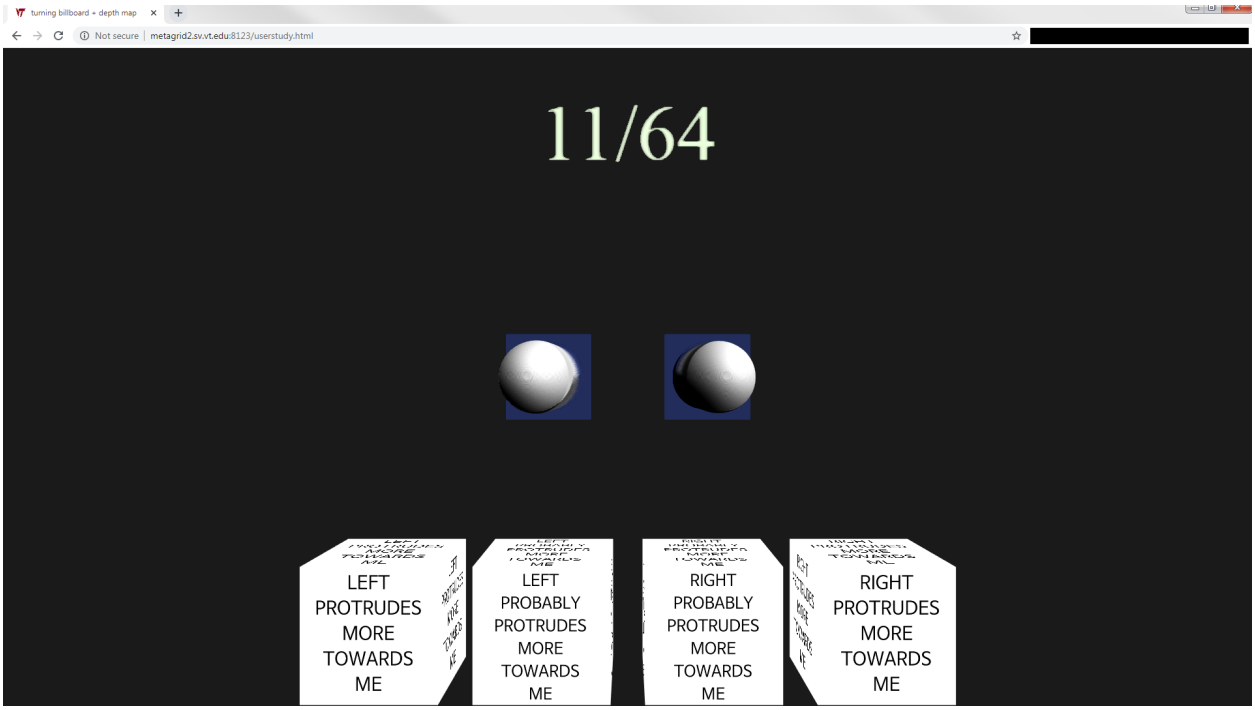


Figure 4.1: A screenshot of the application

Each 2-AFC task contained a different geon image pair (either a box pair or a spheroid pair with different relative protrusions. Table 4.2) rendered using a different combination of depth portrayal techniques. Table 4.1 shows the possible combinations. These combinations were coded as a binary number, where the each digit respectively represented the state of displacement map, shading, and number of images (As explained in 3.1.1). The study contained 64 tasks in total (8 experimental settings x 8 different geon pairs). The different rendering combinations and the different geon pairs were presented in a way that prevented ordering effect between users. This is explained further in 4.0.2.

For each task, a user was allowed to move sideways using the keyboard left-arrow and right-arrow keys to view the IBR objects from various positions. The users were not given the ability to rotate the camera. To prevent the users from losing the objects from the screen, we set up a maximum distance that the user can get away from the IBR objects. If the user tried

		displacement map			
		On		Off	
shading	On	num Imgs		num Imgs	
		Many	Few	Many	Few
	Off	num Imgs		num Imgs	
		Many	Few	Many	Few

Table 4.1: Experimental design independent variables

	Protrusion percent difference			
Cube pair	5%	7.5%	10%	12.5%
Spheroid pair	5%	7.5%	10%	12.5%

Table 4.2: Geon pairs and their relative protrusion differences

to go further, the IBR objects and the buttons below followed the user. In world-space unit, the users were allowed a maximum of 10 meters horizontally away from the IBR objects.

The users were asked to click on the box buttons below using their left mouse button. The options available were:

1. LEFT PROTRUDES MORE TOWARDS ME
2. LEFT PROBABLY PROTRUDES MORE TOWARDS ME
3. RIGHT PROBABLY PROTRUDES MORE TOWARDS ME
4. RIGHT PROTRUDES MORE TOWARDS ME

We gave the 2nd and 3rd options to help the users make a choice even if they were not sure about their decision. These options were added because the study was conducted in a non-monitored environment where the investigators could not help. We were concerned that the subjects may get lost if they were unable to make a judgment. Placing ”maybe” options assists the subjects in making a decision while keeping the task to 2-AFC. During

the evaluation, however, the 1st, 2nd choices and the 3rd, 4th choices were treated equally as strictly choosing left and right. This was because we wanted to keep the study's focus on the correctness of the subjects.

For every task, one of the Cinema IBR object had a protrusion of 1 meter (based), and the other was longer (1.05, 1.075, 1.1, 1.125 meters). The order of the base object and the longer object was randomized for each task to prevent gaming.

For each 2-AFC task, we measured the time to make a decision, the correct answer, the user's choice, and the maximum distance moved in both left and right directions.

Prior to starting the user study, we required all subjects to fill out a pre-study questionnaire mainly asking about demographic information such as age and gender. After the study, we required all subjects to fill out a post-study questionnaire asking about their experience using the application and feedback.

Study procedure

The following is the exact study procedure that each subject followed.

1. Accept the study's Mturk HIT
2. Click on the Google Form survey link inside the HIT.
3. Read the provided electronic consent form and confirm it.
4. Complete the pre-study questionnaire.
5. Follow a link to the web application.
6. Shown two Cinema IBR objects, use the four option buttons below to decide which of the objects were protruding more towards the subject. (Repeat 64 times)

7. Enter the pass code provided by the application at the end of the study in the Google Form survey to proceed to the post-study questionnaire.

8. Complete the post-study questionnaire.

9. Enter the pass code provided at the end of the Google Form survey back in the Mturk HIT.

We made sure that the subjects could not progress in the Google Form survey before signing the electronic consent form. Also, we approved the Mturk work of each subject only if the submitted pass code was correct.

4.0.2 Handling Ordering Effect Between Subjects

In this study, we had a lot of options for arranging the order of experimental set-ups. There are 3 depth cues that can be presented in any order ($(2^3)! = 8!$ set-ups) and 8 combinations of geon pairs that can be arranged in any order (8! set-ups). This is $(8!)^2$ set-ups, which is a incredibly large space to explore. Therefore, we had to reduce the sample space to a manageable size, specifically 72.

When using a smaller sample space, it is important to counter the ordering effect in a fair way, where ideally no two subjects experience a similar set-up. Using a simple sampling without replacements have the possibility where close neighbor samples are selected. Therefore, we used a sampling method that would try to sample 72 experimental set-ups that are all far enough to each other in the sample space.

Switching between experimental set-ups

It is up to the investigator to choose in what order the depth portrayal technique should be turned on or off. We used a binary switch scheme to control the experimental settings. For example, the code 5 in the binary representation is 101. This means the first and third depth portrayal technique should be on. We incremented this code number for the next task until looping back.

Even if we start each subjects' study with a different code, the order of toggling the depth portrayal techniques will stay the same. Therefore, we decided to vary what depth portrayal technique each binary switch digit toggled. The following is an example of two ways of permuting the three depth portrayal techniques.

- 0 = number of images, 1 = shading, 2 = displacement map

- 0 = shading, 1 = number of images, 2 = displacement map

In this example, the same binary code 5 will define a different setting. The toggling order of the experimental settings will also be different for the subjects if we user various orders. There are 3! = 6 ways to arrange the depth portrayal technique positions. We made every 12 subjects start the study with different permutations. For the 12 subjects that are served the same permutation, we randomly picked a different start position. Since there are only 8 possible configurations (2^3), some subjects inevitably participated in the study with the same settings permutation with the same start position. We resolved this issue by serving the geon pairs in a different order.

Switching between geon pairs

There are 4 geon pairs for each cube and sphere. There are 8! different ways of ordering the geon pairs. From this vast number of options, we wanted to pick 72 serving orders that are not too similar to each other.

The first geon pair can be any one of the 8 available pairs. For each geon pair, we randomly picked 9 geon presentation orders that started with the pair. This ensures that at least the first presented geon pairs are evenly distributed between subjects.

4.0.3 Hypotheses

For this study, our hypotheses are,

1. Displacement map is the most significant factor for a user's relative depth judgment between two Cinema IBR objects.

2. Shading will not be as effective as a depth map but better than the number of images at improving a user's relative depth judgment.

3. Applying shading allows better relative depth judgment for spheroid pairs than box pairs.

4. A user who have moved while making the judgment has better accuracy than those who did not.

5. The number of images will not have any effect on a user's relative depth judgment.

6. It will take a shorter time to make a relative depth judgment the more relative depth difference exists between two Cinema IBR objects.

During a series of pilot studies, displacement maps played a significant role in the pilot's ability to sense depth percent difference. Shading was also important in providing additional shape information, especially for the round surface of a sphere. The pilots had to view multiple angles of the geon pairs to confidently make a decision. While it was noticeable when using a fewer number of camera angles versus more angles, the pilots did not seem to show a difference in performance. The geon pairs with the most protrusion percent difference (12.5%) was quite obvious to the pilots, hence we expected less time for the users to make a correct decision.

Chapter 5

Results

5.1 Subjects demographics and recruitment

74 subjects have been recruited for this study. Due to network issues, 14 of the collected data were corrupted or missing, and were thus removed from the evaluation (60 subjects in total). Regarding age, 10% were between 18 to 25, 53.3% were between 26 to 35, 21.6% were between 36 to 45, and 15% were above 45 (Figure 5.1). 60% of the subjects were male, 38.3% female, and 1.6% preferred not to say (Figure 5.2.

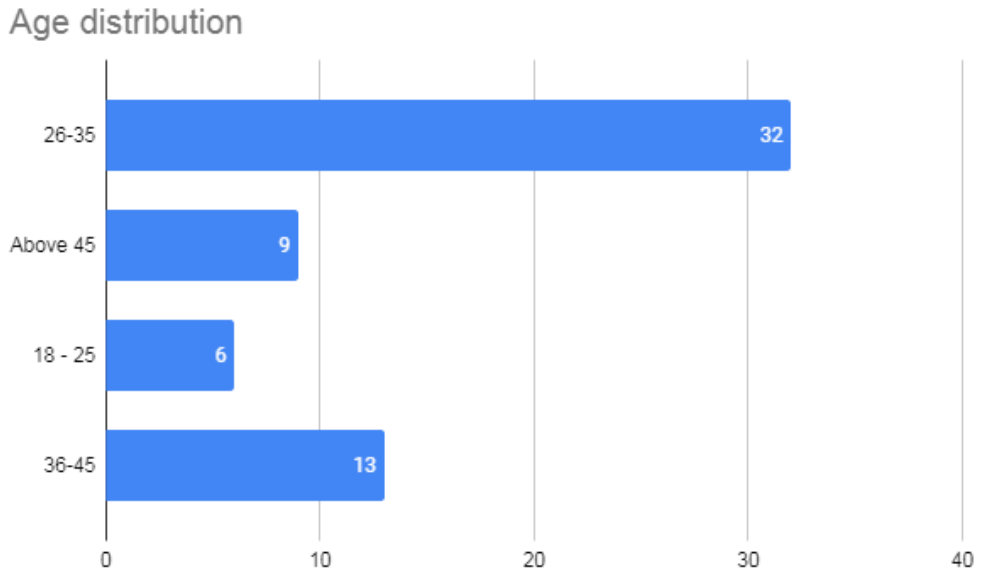


Figure 5.1: Age demographics

The subjects were gathered using an online crowd sourcing platform called Amazon Me-

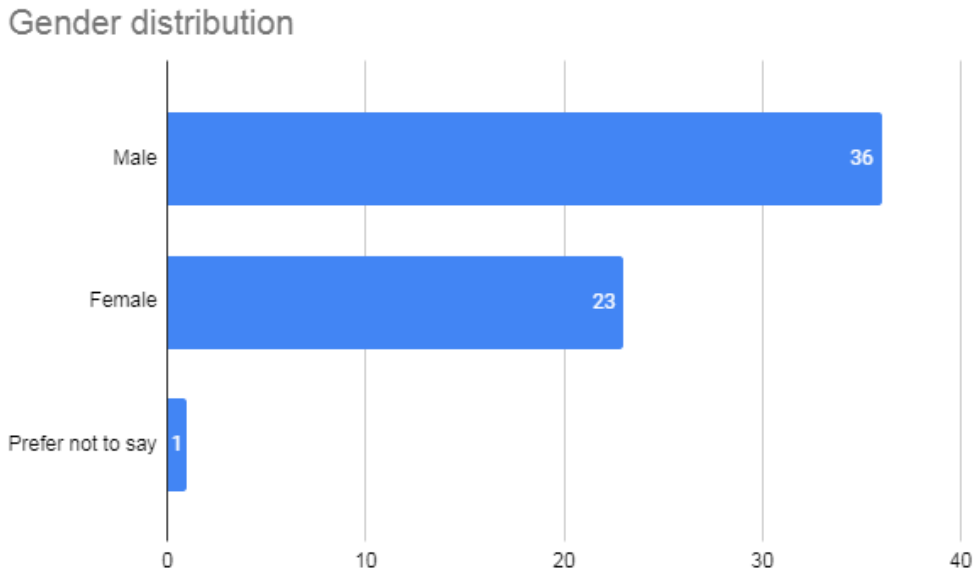


Figure 5.2: Gender demographics

chanical Turk (Mturk). Using Mturk's worker qualifications service, we recruited subjects that were 18 years old or older, had a HIT (Mturk work) approval rate of over 90%, and finished at least 50 HITs previously. The recruitment was done in three sessions. 24 data were collected during the first session, 10 data were collected during the second session, and 40 data were collected in the last session. The second session was initially planned to collect 41 data points, but we noticed that some of the responses from the subjects were lost during upload, so we had to shut down the second session. Data collection took approximately 12 hours in total.

5.2 Results

For each subject's task, we collected the time it took to make a decision (in milliseconds), user selection, correct answer, and the maximum movement lengths (world-space meter unit) in the left and right directions. For the user selections, we further processed the data to create

a column with value 1 for correct answer and 0 for incorrect answer. Each graph in figure 5.3 shows the accuracy comparison for each geon pair between the 8 different experimental settings. In this graph, the x-labels are the binary representations of the experimental settings. The 1st digit represents displacement map, the 2nd digit represents shading, and the last digit represent the number of images. Similarly, figure 5.5 shows the average time to make a correct decision for each geon pair between the 8 different experimental settings. The total average time was 6387.98 ($\sigma = 14258.32$).

The resulting set of graphs did not show an obvious trend. Therefore, we ran an ANOVA analysis to see which of the depth cues were the most significant. The average accuracy per geon pairs are shown in Table 5.1. The total average accuracy was 0.57 ($\sigma = 0.06$).

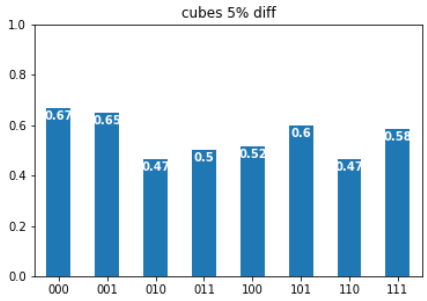
	5%	7.5%	10%	12.5%
Cube	μ 0.56 σ 0.07	μ 0.54 σ 0.08	μ 0.59 σ 0.08	μ 0.59 σ 0.04
Sphere	μ 0.58 σ 0.05	μ 0.56 σ 0.06	μ 0.60 σ 0.07	μ 0.60 σ 0.03

Table 5.1: Accuracy of user response by geons

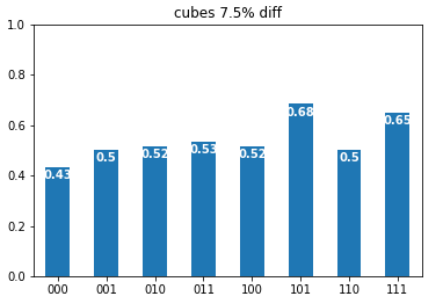
5.2.1 ANOVA (Accuracy)

We performed ANOVA between the depth portrayal techniques with accuracy as the dependent variable. We averaged the 8 responses between all geon pair set-ups to get a subject's average accuracy for each experimental settings. This was required to get a continuous dependent variable for ANOVA analysis. We ran a within-subjects ANOVA using IBM's SPSS statistics tool, with $\alpha = 0.05$.

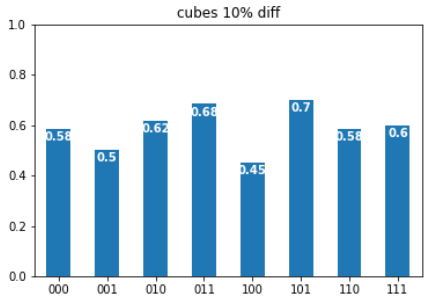
According to the results seen in Figure 5.6, the independent variable displacement map has a F value of 5.442 and a P value (denoted as sig.) of 0.023. Therefore, displacement map



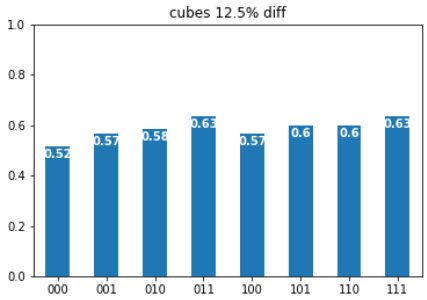
(a)



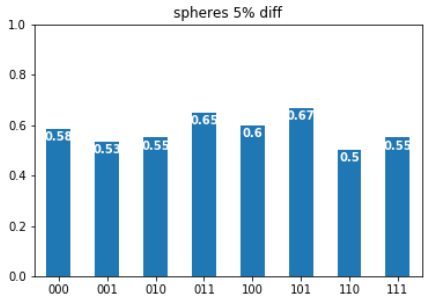
(b)



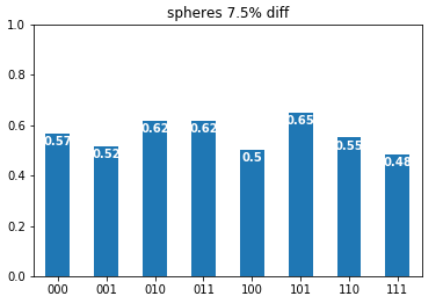
(c)



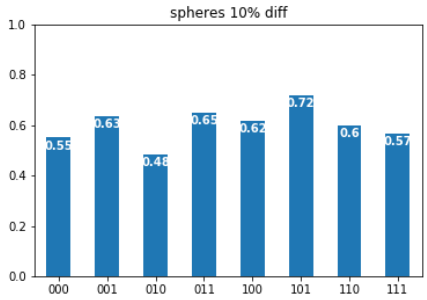
(d)



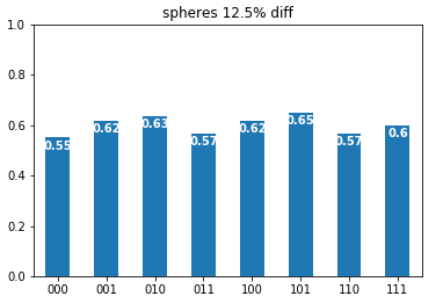
(e)



(f)



(g)



(h)

Figure 5.3: Accuracy distributions between different geon pairs of different protrusion percent difference. 1st, 2nd, and 3rd binary digits of x-axis labels show state of conditions displacement map, shading, and number of images, respectively.

Descriptive Statistics

	Mean	Std. Deviation	N
@000	.55625	.213953	60
@001	.56458	.243008	60
@010	.55833	.181896	60
@011	.60417	.194150	60
@100	.54792	.196758	60
@101	.65833	.195102	60
@110	.54583	.182623	60
@111	.58333	.213012	60

(a) accuracy

Descriptive Statistics

	Mean	Std. Deviation	N
@000	5712.533	9391.8226	480
@001	6089.496	14346.2394	480
@010	7813.521	24849.2291	480
@011	5416.348	10997.8875	480
@100	5493.390	14370.7304	480
@101	6206.704	12392.5828	480
@110	6105.392	12618.5173	480
@111	5967.752	12337.9429	480

(b) time (ms)

Figure 5.4: Descriptive statistics for each experimental setting. 1st, 2nd, and 3rd binary digits show state of conditions displacement map, shading, and number of images, respectively. Accuracy is calculated by averaging a user's answers between 8 geon pairs for each condition.

is found to be significant. The P values of shading and number of images were respectively 0.588 and 0.347, indicating that these two depth portrayal techniques are not significant.

Estimated marginal means (EMM) of the accuracy was 0.552 when displacement map was off, and 0.603 when displacement map was on. The users had approximately 5% higher accuracy when displacement map was on. The other two independent variables didn't seem to appear statistically significant according to ANOVA.

5.2.2 Accuracy and displacement map

Based on the results of ANOVA analysis, we plotted the average accuracy by the displacement map condition. As expected, average accuracy was higher when displacement map was on for all geon pairs (Figure 5.8). The best average accuracy was achieved when there was a 10% relative depth difference and when displacement map was on for both cubes and sphere geon pairs, each with an average accuracy of 0.62 ($\sigma = 0.29$) and 0.64 ($\sigma = 0.28$), respectively.

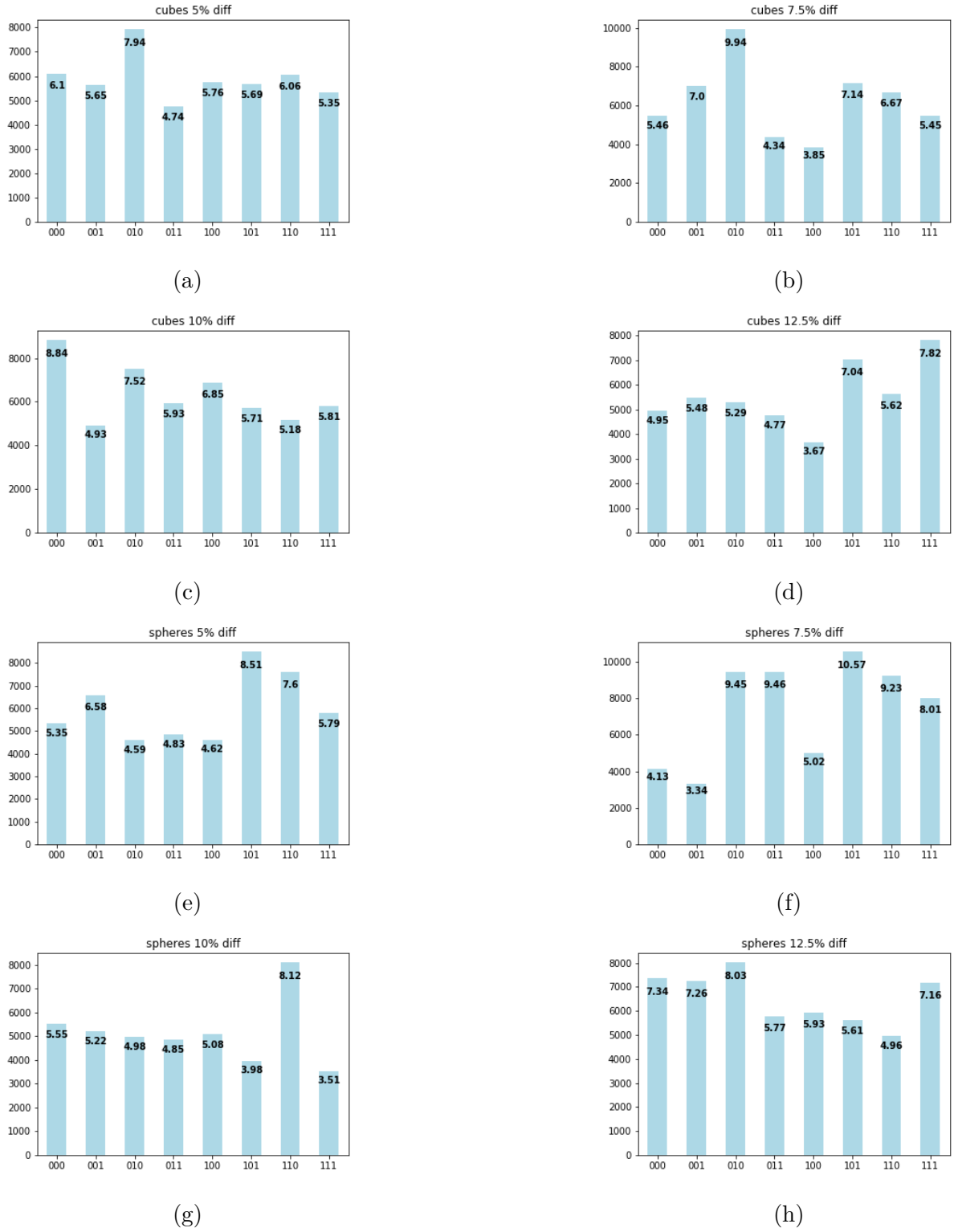


Figure 5.5: Average time distributions (in seconds) between different geon pairs of different protrusion percent difference (only correct responses). 1st, 2nd, and 3rd binary digits of x-axis labels show state of conditions displacement map, shading, and number of images, respectively.

Multivariate Tests[a]

Effect		Value	F	Hypothesis df	Error df	Sig.	Partial Eta Squared	Noncent. Parameter	Observed Power[c]
numImgs	Pillai's Trace	.015	.900[b]	1.000	59.000	.347	.015	.900	.154
	Wilks' Lambda	.985	.900[b]	1.000	59.000	.347	.015	.900	.154
	Hotelling's Trace	.015	.900[b]	1.000	59.000	.347	.015	.900	.154
	Roy's Largest Root	.015	.900[b]	1.000	59.000	.347	.015	.900	.154
shading	Pillai's Trace	.005	.297[b]	1.000	59.000	.588	.005	.297	.084
	Wilks' Lambda	.995	.297[b]	1.000	59.000	.588	.005	.297	.084
	Hotelling's Trace	.005	.297[b]	1.000	59.000	.588	.005	.297	.084
	Roy's Largest Root	.005	.297[b]	1.000	59.000	.588	.005	.297	.084
depthmap	Pillai's Trace	.084	5.442[b]	1.000	59.000	.023	.084	5.442	.631
	Wilks' Lambda	.916	5.442[b]	1.000	59.000	.023	.084	5.442	.631
	Hotelling's Trace	.092	5.442[b]	1.000	59.000	.023	.084	5.442	.631
	Roy's Largest Root	.092	5.442[b]	1.000	59.000	.023	.084	5.442	.631
numImgs * shading	Pillai's Trace	.046	2.855[b]	1.000	59.000	.096	.046	2.855	.383
	Wilks' Lambda	.954	2.855[b]	1.000	59.000	.096	.046	2.855	.383
	Hotelling's Trace	.048	2.855[b]	1.000	59.000	.096	.046	2.855	.383
	Roy's Largest Root	.048	2.855[b]	1.000	59.000	.096	.046	2.855	.383
numImgs * depthmap	Pillai's Trace	.040	2.479[b]	1.000	59.000	.121	.040	2.479	.341
	Wilks' Lambda	.960	2.479[b]	1.000	59.000	.121	.040	2.479	.341
	Hotelling's Trace	.042	2.479[b]	1.000	59.000	.121	.040	2.479	.341
	Roy's Largest Root	.042	2.479[b]	1.000	59.000	.121	.040	2.479	.341
shading * depthmap	Pillai's Trace	.005	.285[b]	1.000	59.000	.595	.005	.285	.082
	Wilks' Lambda	.995	.285[b]	1.000	59.000	.595	.005	.285	.082
	Hotelling's Trace	.005	.285[b]	1.000	59.000	.595	.005	.285	.082
	Roy's Largest Root	.005	.285[b]	1.000	59.000	.595	.005	.285	.082
numImgs * shading * depthmap	Pillai's Trace	.043	2.681[b]	1.000	59.000	.107	.043	2.681	.364
	Wilks' Lambda	.957	2.681[b]	1.000	59.000	.107	.043	2.681	.364
	Hotelling's Trace	.045	2.681[b]	1.000	59.000	.107	.043	2.681	.364
	Roy's Largest Root	.045	2.681[b]	1.000	59.000	.107	.043	2.681	.364

a. Design: Intercept
Within Subjects Design: numImgs + shading + depthmap + numImgs * shading + numImgs * depthmap + shading * depthmap + numImgs * shading * depthmap

b. Exact statistic

c. Computed using alpha = .05

Figure 5.6: Within-subjects ANOVA results (Accuracy)

To see which of the geons benefited more from applying the displacement map, we plotted the accuracy differences of the geon pairs (Figure 5.9). On average, cube pairs had 0.0625 ($\sigma = 0.023$) accuracy increase from applying a displacement map, while spheroid pairs had 0.0375 ($\sigma = 0.027$)) accuracy increase. Maximum of 0.1 accuracy increase for cube pairs from the application of the displacement map was achieved when there was a 7.5% relative depth difference. For spheroid pairs, a maximum of 0.08 accuracy improvement was seen

Estimates

Measure: acc

depthmap	Mean	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
1	.552	.015	.521	.583
2	.603	.019	.564	.641

Figure 5.7: Estimated marginal means of displacement map

when there was a 10% relative depth difference.

5.2.3 ANOVA (Time)

We ran ANOVA with time as the dependent variable with the same independent variables as we did for accuracy (Figure 5.10). The values were not averaged for this since time in milliseconds was already a continuous variable. Thus, there were 480 (60 * 8) samples for each experimental conditions.

This time, none of the individual depth portrayal techniques were shown to be statistically significant as all of the P values were above 0.05. The null hypothesis could not be rejected for any of the independent variables. However, the P value for the interaction between shading and displacement map was was 0.27, showing that this interaction is significant. Figure 5.11 shows the EMM. When shading was not applied, subjects tended to spend less time if displacement map was also not applied. When shading was applied, subjects spent less time if displacement map was also applied. With slightly less confidence, the interaction between number of images and displacement map was also found to be significant, with a P value of 0.055. In contrast to the interaction between shading and displacement map, subjects spent less time to make a decision when only one of number of images or displacement map was applied (Figure 5.12).

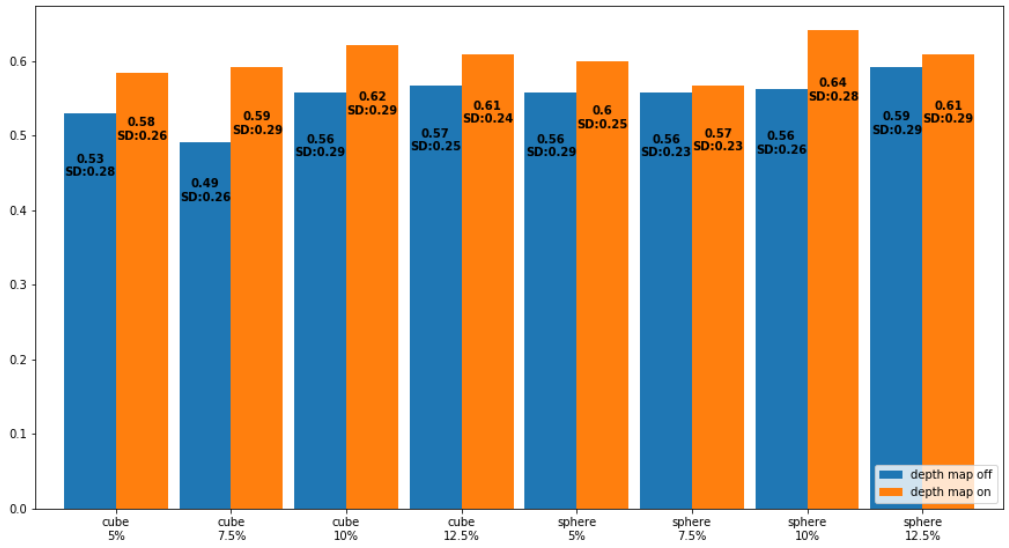
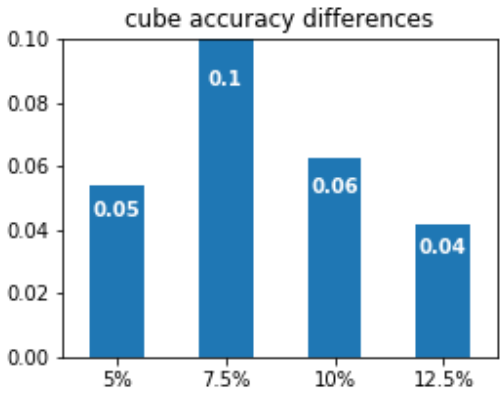
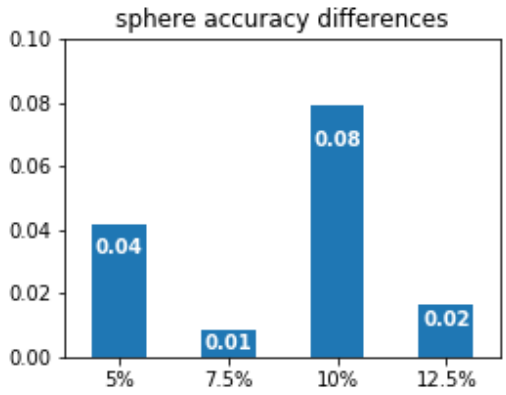


Figure 5.8: Accuracy comparison by displacement map condition



(a) Accuracy improvements of cube pairs for each protrusion percent difference



(b) Accuracy improvements of spheroid pairs for each protrusion percent difference

Figure 5.9: Accuracy improvements for the two geon pairs

Multivariate Tests[a]

Effect		Value	F	Hypothesis df	Error df	Sig.	Partial Eta Squared	Noncent. Parameter	Observed Power[c]
numImg	Pillai's Trace	.001	.466[b]	1.000	479.000	.495	.001	.466	.105
	Wilks' Lambda	.999	.466[b]	1.000	479.000	.495	.001	.466	.105
	Hotelling's Trace	.001	.466[b]	1.000	479.000	.495	.001	.466	.105
	Roy's Largest Root	.001	.466[b]	1.000	479.000	.495	.001	.466	.105
shading	Pillai's Trace	.003	1.201[b]	1.000	479.000	.274	.003	1.201	.194
	Wilks' Lambda	.997	1.201[b]	1.000	479.000	.274	.003	1.201	.194
	Hotelling's Trace	.003	1.201[b]	1.000	479.000	.274	.003	1.201	.194
	Roy's Largest Root	.003	1.201[b]	1.000	479.000	.274	.003	1.201	.194
depthmap	Pillai's Trace	.002	1.040[b]	1.000	479.000	.308	.002	1.040	.174
	Wilks' Lambda	.998	1.040[b]	1.000	479.000	.308	.002	1.040	.174
	Hotelling's Trace	.002	1.040[b]	1.000	479.000	.308	.002	1.040	.174
	Roy's Largest Root	.002	1.040[b]	1.000	479.000	.308	.002	1.040	.174
numImg * shading	Pillai's Trace	.001	.411[b]	1.000	479.000	.522	.001	.411	.098
	Wilks' Lambda	.999	.411[b]	1.000	479.000	.522	.001	.411	.098
	Hotelling's Trace	.001	.411[b]	1.000	479.000	.522	.001	.411	.098
	Roy's Largest Root	.001	.411[b]	1.000	479.000	.522	.001	.411	.098
numImg * depthmap	Pillai's Trace	.008	3.705[b]	1.000	479.000	.055	.008	3.705	.485
	Wilks' Lambda	.992	3.705[b]	1.000	479.000	.055	.008	3.705	.485
	Hotelling's Trace	.008	3.705[b]	1.000	479.000	.055	.008	3.705	.485
	Roy's Largest Root	.008	3.705[b]	1.000	479.000	.055	.008	3.705	.485
shading * depthmap	Pillai's Trace	.010	4.893[b]	1.000	479.000	.027	.010	4.893	.598
	Wilks' Lambda	.990	4.893[b]	1.000	479.000	.027	.010	4.893	.598
	Hotelling's Trace	.010	4.893[b]	1.000	479.000	.027	.010	4.893	.598
	Roy's Largest Root	.010	4.893[b]	1.000	479.000	.027	.010	4.893	.598
numImg * shading * depthmap	Pillai's Trace	.003	1.575[b]	1.000	479.000	.210	.003	1.575	.240
	Wilks' Lambda	.997	1.575[b]	1.000	479.000	.210	.003	1.575	.240
	Hotelling's Trace	.003	1.575[b]	1.000	479.000	.210	.003	1.575	.240
	Roy's Largest Root	.003	1.575[b]	1.000	479.000	.210	.003	1.575	.240

a. Design: Intercept
Within Subjects Design: numImg + shading + depthmap + numImg * shading + numImg * depthmap + shading * depthmap + numImg * shading * depthmap

b. Exact statistic

c. Computed using alpha = .05

Figure 5.10: ANOVA results (time)

5.2.4 Accuracy with motion parallax

We re-computed the accuracy for each geon pair after filtering the data where the users moved during the task (i.e. used motion parallax) (Figure 5.15). As there were not many subjects that used motion parallax during the tasks, the filtered data size was much smaller than the full data (308 responses, 8.02% of the total data). We color coded the bars calculated

Estimates

Measure: time

shading	depthmap	Mean	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
1	1	5602.961	437.116	4744.060	6461.863
	2	6148.100	504.856	5156.093	7140.107
2	1	6959.456	706.833	5570.581	8348.332
	2	5692.050	432.855	4841.520	6542.580

Figure 5.11: Estimated marginal means of interaction between shading and displacement map

Estimates

Measure: time

numImg	depthmap	Mean	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
1	1	6763.027	653.190	5479.555	8046.499
	2	5752.922	516.467	4738.102	6767.742
2	1	5799.391	479.588	4857.035	6741.746
	2	6087.228	446.134	5210.606	6963.851

Figure 5.12: Estimated marginal means of interaction between number of images and displacement map

using at least 5 responses in blue, and orange for others. The annotated integers below each bar's accuracy are the number of responses to calculate the accuracy. The average accuracy between the responses with at least 5 responses was 0.67 ($\sigma = 0.23$). We re-calculated the average accuracy using the accuracy values in 5.3 for the same conditions in 5.15 with at least 5 responses, that gave a value of 0.56 ($\sigma = 0.06$) (Figure 5.13). We ran an independent samples T-test, and found that the mean difference was significant, with a P value of 0.009. Therefore, motion parallax was found to be a significant factor for accuracy (Figure 5.14).

Group Statistics

	move	N	Mean	Std. Deviation	Std. Error Mean
acc	0	35	.5619047619	.0589354677	.0099619123
	1	35	.6704761905	.2266744049	.0383149676

Figure 5.13: Average accuracy with (1) and without (2) motion parallax

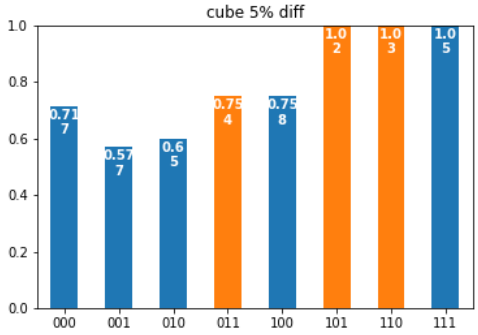
Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
acc	Equal variances assumed	41.682	.000	-2.742	68	.008	-.108571429	.0395888423	-.187569733	-.029573124
	Equal variances not assumed			-2.742	38.576	.009	-.108571429	.0395888423	-.188675577	-.028467281

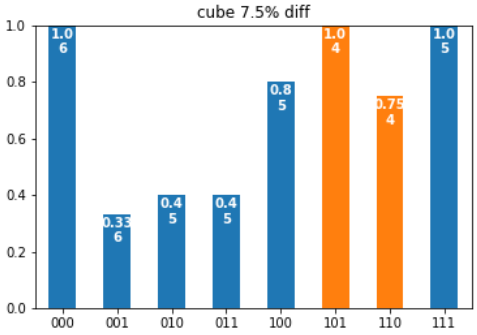
Figure 5.14: T test results for average accuracy

5.2.5 Accuracy trends for each experimental set-ups

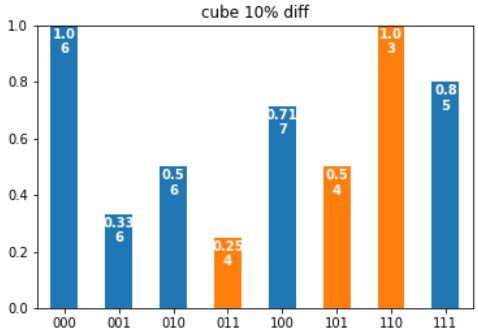
Figure 5.16 shows the accuracy trends for each experimental set-ups along increasing relative protrusion of the geon pairs. Each point is the ratio of correct responses and the number of subjects. We found 4 settings (101, 011, 110, 000) to have had the highest accuracy when there was a 10% relative depth difference between the geons. We see a similar trend for settings 111 and 100 where the number of images were many.



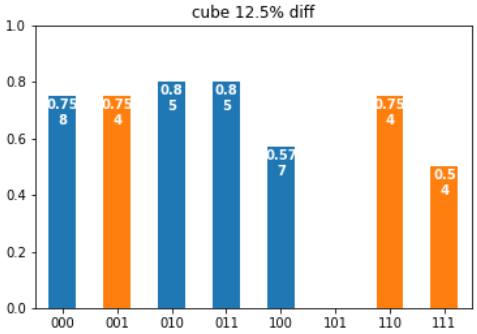
(a)



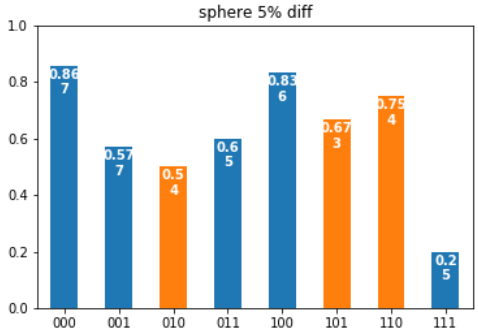
(b)



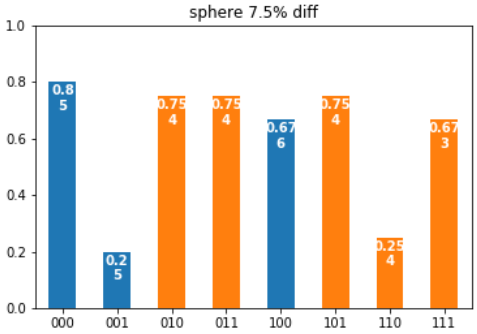
(c)



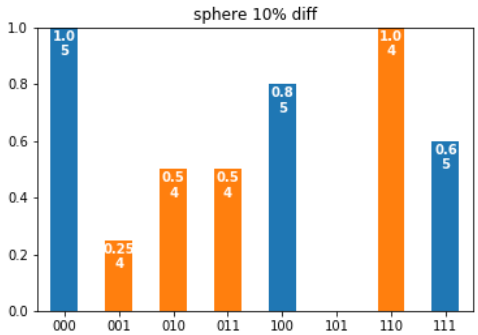
(d)



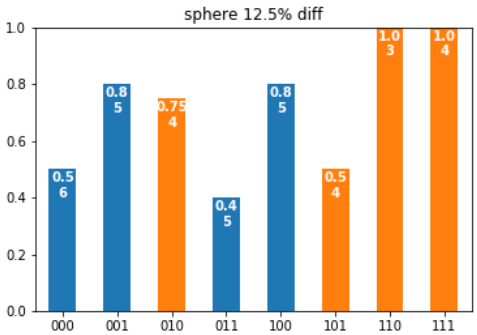
(e)



(f)



(g)



(h)

Figure 5.15: Accuracy comparison between geons when motion parallax was used

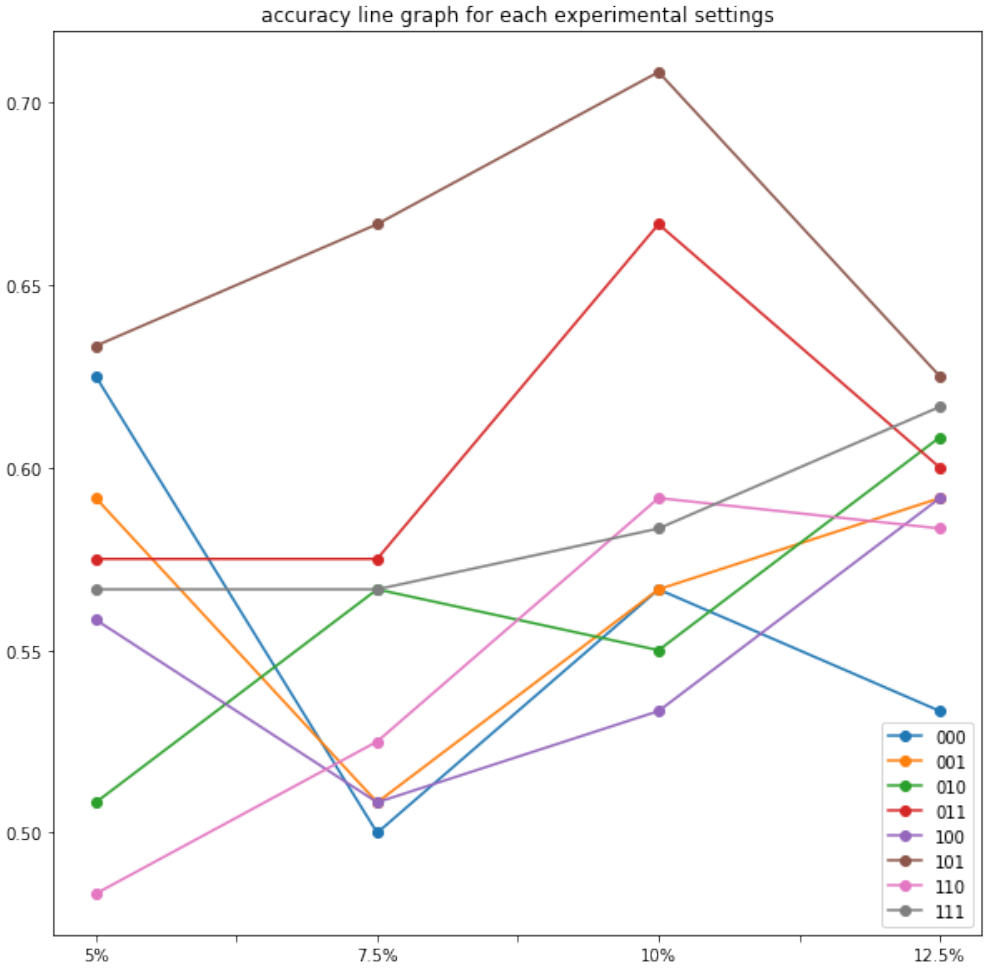


Figure 5.16: Accuracy under each experimental conditions for each protrusion levels

Chapter 6

Discussion

6.1 Accuracy difference between geon pairs of different relative depth

The result doesn't show significant improvement in accuracy as the protrusion percent difference between geon pairs increase. For all geon pairs, the users showed a rough average of 0.5 when displacement map wasn't applied and 0.6 when it was. This suggests that the relative depth judgment signal threshold of the subjects does not exist in the range of 5% to 12.5% relative depth difference. It exists in a higher value. This study was originally designed to be conducted in VR, where stereoscopy is available. We think conducting the study again in a VR setting would significantly change the results.

6.2 Effect of the depth cues

As seen from the ANOVA analysis, displacement map showed the most significance for affecting accuracy. This confirms hypothesis 1. This was an expected result. The application of a displacement map explicitly protrudes the vertices in world space and effectively portrays depth. We believe the front plane of the Cinema IBR objects appeared larger the more protrusion they had, and the relative size difference of front planes gave an indication of

relative depth difference.

An interesting result was that shading did not seem to be a significant factor. In Cutting and Vishton's work [12], shadows and shading was not considered a depth cue but rather an element defining shape. Hubona et al. found shadows to be useful cues if there existed only one in the scene [18]. The application for this study included shading on the object's surface, but did not have shadows and backgrounds as in Hubona's work. From the results, we agree with Cutting and Vishton that shading is not a useful depth cue. Hypotheses 2 and 3 were disconfirmed.

From looking at the accuracy improvements after applying displacement map (Figure 5.9), we could see more average improvements in the cube pairs' case. This suggests that flat surfaces benefit more from physical protrusions for a user's relative depth perception.

6.3 Participant movement and significance of number of images

For the increased number of images (i.e. more camera angle steps) to be useful, a user needs to move around in the virtual environment. However, the results showed that only 17 subjects (28.3%) used the movement functionality at least once, and the majority of these 17 subjects tried it only once at the start of the study. Only 8.02% of the tasks were done using motion parallax. In the instructions of the study, we had encouraged subjects to use their keyboard arrow keys to move. We assume due to the nature of crowd work tasks that are focused on finishing the job as fast as possible, the encouragement was quickly forgotten.

In the results, we have also reported accuracy for the filtered data where users did use movement during the the tasks (Subsection 5.2.4). We found that the accuracy for this

filtered data was higher than the average obtained from the entire data and that the result was significant (Figure 5.14). Thus, we can confirm hypothesis 4.

ANOVA analysis results show that the number of images did not have any influence in a user's ability to perceive relative depth difference. However, the number of images is only relevant if a subject looks and compares the different set of camera angle images. As many subjects made their decisions based on one image for each task, it is hard to completely confirm nor disconfirm hypothesis 5.

6.4 Participant decision time

We found that decision time was affected by the interaction between shading and displacement map (Figure 5.11). Responses were made faster when either both shading and displacement map was not applied, or when both were applied. This was an expected result since shading was only noticeable when the image was protruded using the displacement map. We believe that when both settings were applied, the relative depth difference was more apparent. This allowed faster decision. For when both settings were off, we assume the users could not easily notice the difference, and thus randomly chose an object without spending much time for consideration.

We also found the interaction between number of images and displacement map to be affecting decision time (Figure 5.12). The certainty with this result was below 95% by a small margin. This result showed that users can make an easier decision if only one of the two depth portrayal techniques (number of images, displacement map) were applied. There is a possibility that the protrusion of the plane vertices hindered the signal strength of motion parallax. However, as stated in 6.3, the sample size of participants who moved during the study is small. Thus, we cannot conclude for certain that relative size and motion parallax

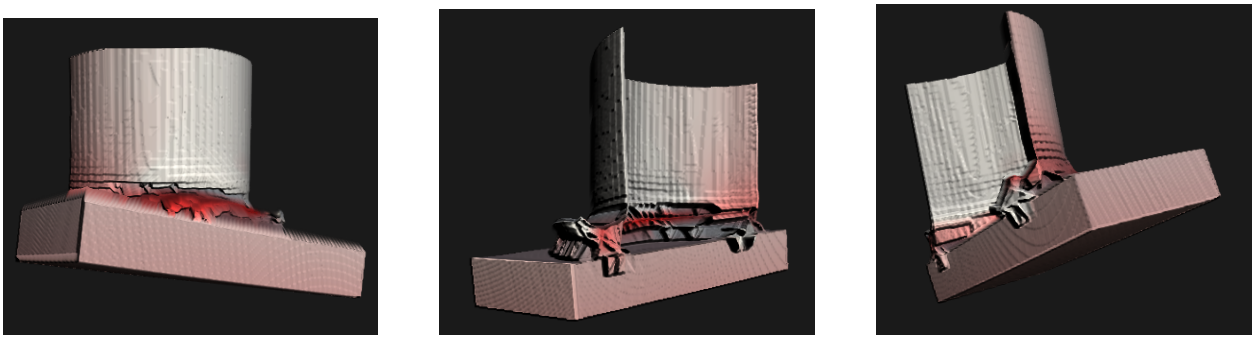
counter each other in affecting relative depth judgment.

We had to disconfirm hypothesis 6 from looking at figure 5.5. It did not appear that time was influenced by the level of relative protrusion difference between geons. As previously stated, we believe that the relative depth judgment signal threshold does not lie in the experimental range that we set up. We believe the lack of perceived signal influenced the decision time.

6.5 Real world application

During this study, we was able to come up with a method that renders an image as if it's a 3D object in a convincing way. For the sake of this study, we rendered two simple geons (box and spheroid). However, this method is capable of rendering more complex geometry.

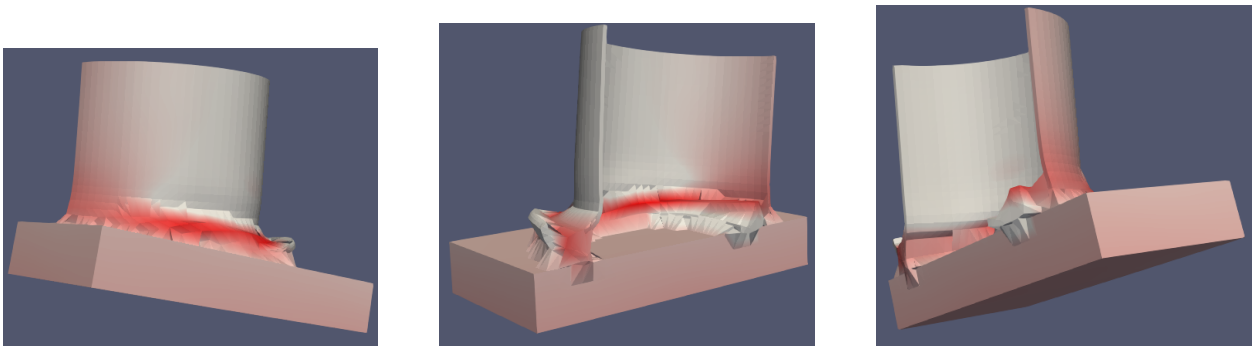
The following figure 6.1 is a visualization of the crushed can timeseries example, which is part of the Paraview Tutorial. Figure 6.2 are snapshots of the real 3D model.



(a) (b) (c)

Figure 6.1: Crushed can example view from multiple angles (Cinema IBR object)

This visualization is colored using one of the attributes in the data (y-Velocity). Just as the original Cinema method can save multiple images each colored with data attributes, our method can show multiple 3D geometry colored by any attribute available in the data. It has a strong presentation of 3-dimensionality, which is desired for such spatial data.



(a) (b) (c)

Figure 6.2: Crushed can example view from multiple angles (Real geometry)

Table 6.1 shows a comparison of file size between the original Cinema image database without the displacement map textures, our method's Cinema image database, the X3D file of the real geometry, and the binary file of the real geometry. Since our method requires one additional image per snapshot compared to the original Cinema image database, the required file size approximately doubles. Compared to the X3D file, the Cinema database saves approximately 400KBs. The X3D binary file is smaller than our approach for this example. However, given that the Cinema database's file size is only dependent on the number of camera angles and the image resolution regardless of the geometric complexity, our method greatly reduces the cost of preserving depth information.

	Cinema image database	Cinema IBR object	X3D file	X3Db file
filesize	677 KB	1.3 MB	1.76 MB	78.7 KB

Table 6.1: File size comparison between Cinema and X3D

Since table 6.1 was generated using a very small data with only 10,089 data points, we ran a similar analysis with the HERMES data with 150,000 data points (Table 6.2). We start to see significant save of memory. The Cinema image database file sizes have increased by approximately twice the size compared to Table 6.1, but this is due to file compression of the PNG image format. For the HERMES data, even the X3D binary file is approximately

14 times larger than the Cinema database including the displacement map. Figure 6.3 shows the file size trend, plotted with the two tables, in logarithmic scale. The dashed lines are extrapolations for larger data.

	Cinema image database	Cinema IBR object	X3D file	X3Db file
filesize	1.38 MB	2.64 MB	1.44 GB	36.9 MB

Table 6.2: File size comparison between Cinema and X3D with larger data

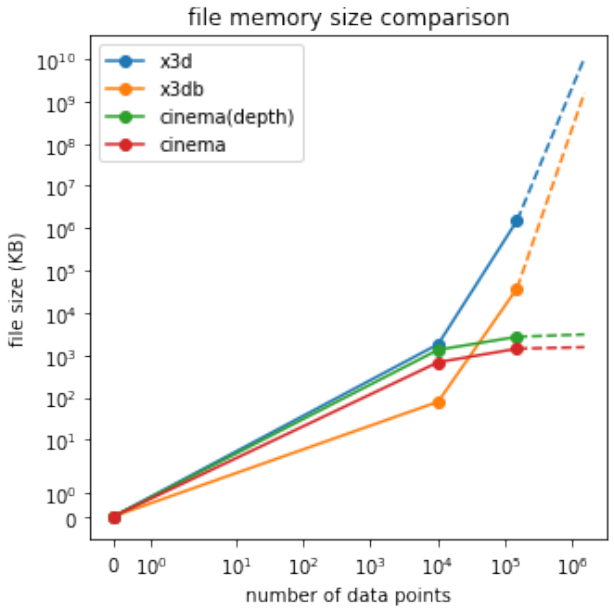


Figure 6.3: File size comparison trend

Some components need to be improved to better resemble actual 3D geometry. Currently, the only data type that our method supports is point data. We directly sample the displacement map values from each point. A better way to extract depth information from a 3D model will allow more freedom in regards of data type. For generating this crushed can example however, a slight improvement was already made, using one of Paraview's filters (Delaunay) that can tesselate a 3D model mesh from a point cloud.

Improvements must also be made in regard to shading and surface resolution. Currently,

our method's shader simply compares the cosine value between the light direction and the vertex normals. This method also leaves high-frequency artifacts on the surface due to the current normal calculation method. A more sophisticated normal calculation method and better shading scheme such as Phong shading or Lambert shading could be used to smoothen the rendered surface. Alternatively, coding the normals using another normal texture map could be a good solution.

A valuable functionality of this implementation is that the shading option can be turned on or off by changing a programmatically accessible boolean value. This allows a researcher to view both the exact geometry using shading and the correct color without shading. This can be utilized to allow a researcher to accurately perceive both shape and color of a visualization.

Additionally, as the IBR objects are independent of the scene and render as a separate object, they can be placed in any virtual environment and fit in naturally with other real 3D geometry. In the X3D scene graph and X3DOM runtime implementation of my technique, it is possible to have numerous depth-enhanced IBR objects in an interactive Web3D scene.

Chapter 7

Conclusion and future work

In this study, I developed a lightweight rendering technique to add depth cues back into IBR presentations such as Cinema. Using one additional image (displacement map) to the diffuse map, I have demonstrated three strategies to portray depth information: displacement, shading, and camera continuity. With this new technique in hand, I evaluated user performance with different combinations of depth cues using an online study of 60 participants through Amazon Mechanical Turk.

The physical displacement of the image pixels (i.e. the vertex whose UV mapping is the pixel) using a pre-generated displacement map was shown to have the most effect on a user's ability to compare relative depth difference. Thus, hypothesis 1 is supported by the data. Shading the geon surfaces was found to have no significant effect on the users' relative depth judgments, thus not supporting hypothesis 2 and 3. Finally, Varying the number of images (reducing camera angle step size) seemed to not have a significant effect on users' time or accuracy performance.

One interpretation of this result could be that users did not move their virtual camera much during the study. We found that by filtering the results to only subjects who moved during the study, does show an improvement in performance accuracy. Time needed to perceive relative depth was not affected by any of the depth cues, thus hypothesis 6 was not supported.

While we originally designed our study to be run on an Oculus Go and Samsung GearVR

HMDs, we were not able to implement this as on online experiment. In future work, we can expand this study to evaluate the role and interactions of stereoscopy as an additional depth cue. Along with the nature of the fast-paced online crowd source work environment, we believe the controls for moving left or right seemed unnatural for the users, and this prevented the majority of the subjects from moving. In a VR environment, movement can be natural, and we believe in those contexts, subjects would gain an advantage in perceptual judgment from the additional depth cue.

The new application of a displacement map looks promising for inspecting scientific models generated from an exa-scale data with a relatively low cost. As of now, the generation of the displacement map for our method is limited to point clouds. But with further improvements, we believe this way of rendering will be valuable to the scientific research community.

One exciting future application will be the integration of depth-enhanced IBR objects to Immersive Analytic Workspaces [16] [20] [32], such as GLEE [13], where Cinema thumbnails are used to drive human-in-the-loop Machine Learning known as Semantic Interaction [17] [15]. Semantic Interaction in 2D workspaces has shown benefits, especially in the visualization of ensembles. As Virtual Reality and Mixed Reality become more widespread and interoperable, this novel approach to IBR provides a way to represent and portray depth information at interactive frame rates.

Bibliography

[1] Cinema Example. https://dsscale.org/cv/cvlib/html/SpecA-Single.html?cinemadb/astervolume.cdb/image/info.json, .

[2] Cinema Science. https://cinemascience.github.io/, .

[3] The HERMES Experiment. http://www-hermes.desy.de/, 2015.

[4] Microsoft Hololens. https://www.microsoft.com/en-us/hololens, 2016.

[5] Edward H Adelson, James R Bergen, et al. *The plenoptic function and the elements of early vision*, volume 2. Vision and Modeling Group, Media Laboratory, Massachusetts Institute of Technology, 1991.

[6] James Ahrens, Sébastien Jourdain, Patrick O'Leary, John Patchett, David H. Rogers, and Mark Petersen. An image-based approach to extreme scale in situ visualization and analysis. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '14, pages 424–434, Piscataway, NJ, USA, 2014. IEEE Press. ISBN 978-1-4799-5500-8. doi: 10.1109/SC.2014.40. URL https://doi.org/10.1109/SC.2014.40.

[7] Irving Biederman, Eric Cooper, John Hummel, and Jozsef Fiser. Geon theory as an account of shape recognition in mind, brain and machine. 01 1993. doi: 10.5244/C.7.18.

[8] Ken Casey and Chris Exton. A java 3d implementation of a geon based visualisation tool for uml. In *Proceedings of the 2nd International Conference on Principles and Practice of Programming in Java*, PPPJ '03, page 63–65, USA, 2003. Computer Science Press, Inc. ISBN 0954414519.

[9] Ju-Chin Chen and Meng-yuan Huang. 2d-to-3d conversion system using depth map enhancement. *KSII Transactions on Internet & Information Systems*, 10(3), 2016.

[10] Shenchang Eric Chen. Quicktime vr: An image-based approach to virtual environment navigation. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 29–38. ACM, 1995.

[11] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 279–288, 1993.

[12] James E Cutting and Peter M Vishton. Perceiving layout and knowing distances: The integration, relative potency, and contextual use of different information about depth. In *Perception of space and motion*, pages 69–117. Elsevier, 1995.

[13] M Dahshan, NF Polys, RS Jayne, and RM Pollyea. Making sense of scientific simulation ensembles with semantic interaction. In *Computer Graphics Forum*. Wiley Online Library, 2019.

[14] Paul Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, page 189–198, New York, NY, USA, 1998. Association for Computing Machinery. ISBN 0897919998. doi: 10.1145/280814.280864. URL https://doi.org/10.1145/280814.280864.

[15] Michelle Dowling, John Wenskovitch, Peter Hauck, Adam Binford, Nicholas Polys, and Chris North. A bidirectional pipeline for semantic interaction. In *Proc. Workshop on Machine Learning from User Interaction for Visualization and Analytics (at IEEE VIS 2018)*, volume 11, 2018.

[16] Tim Dwyer, Nathalie Henry Riche, Karsten Klein, Wolfgang Stuerzlinger, and Bruce Thomas. Immersive analytics (dagstuhl seminar 16231). In *Dagstuhl Reports*, volume 6. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

[17] Alex Endert, Patrick Fiaux, and Chris North. Semantic interaction for sensemaking: inferring analytical reasoning for model steering. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2879–2888, 2012.

[18] Geoffrey S Hubona, Philip N Wheeler, Gregory W Shirah, and Matthew Brandt. The relative contributions of stereo, lighting, and background scenes in promoting 3d depth visualization. *ACM Transactions on Computer-Human Interaction*, 6(3):214–242, 1999.

[19] Oliver Kreylos, Gerald Bawden, Tony Bernardin, Magali I. Billen, Eric S. Cowgill, Ryan D. Gold, Bernd Hamann, Margarete Jadamec, Louise H. Kellogg, Oliver G. Staadt, and Dawn Y. Sumner. Enabling scientific workflows in virtual reality. In *Proceedings of the 2006 ACM International Conference on Virtual Reality Continuum and Its Applications*, VRCIA '06, page 155–162, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933247. doi: 10.1145/1128923.1128948. URL https://doi-org.ezproxy.lib.vt.edu/10.1145/1128923.1128948.

[20] Kim Marriott, Jian Chen, Marcel Hlawatsch, Takayuki Itoh, Miguel A Nacenta, Guido Reina, and Wolfgang Stuerzlinger. Immersive analytics: Time to reconsider the value of 3d for information visualisation. In *Immersive Analytics*, pages 25–55. Springer, 2018.

[21] Leonard McMillan. *An image-based approach to three-dimensional computer graphics*. PhD thesis, Citeseer, 1997.

[22] Pranav Mistry, Pattie Maes, and Liyan Chang. Wuw - wear ur world: A wearable gestural interface. In *CHI '09 Extended Abstracts on Human Factors in Computing*

Systems, CHI EA '09, page 4111–4116, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605582474. doi: 10.1145/1520340.1520626. URL https://doi.org/10.1145/1520340.1520626.

[23] Fumio Okura, Masayuki Kanbara, and Naokazu Yokoya. Mixed-reality world exploration using image-based rendering. *J. Comput. Cult. Herit.*, 8(2), March 2015. ISSN 1556-4673. doi: 10.1145/2700428. URL https://doi-org.ezproxy.lib.vt.edu/10.1145/2700428.

[24] Ekaterina Olshannikova, Aleksandr Ometov, Yevgeni Koucheryavy, and Thomas Olsson. Visualizing big data with augmented and virtual reality: challenges and research agenda. *Journal of Big Data*, 2(1):22, 2015.

[25] Thammathip Piumsomboon, Adrian Clark, Mark Billinghurst, and Andy Cockburn. User-defined gestures for augmented reality. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, page 955–960, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450319522. doi: 10.1145/2468356.2468527. URL https://doi.org/10.1145/2468356.2468527.

[26] Nicholas F. Polys, Seonho Kim, and Doug A. Bowman. Effects of information layout, screen size, and field of view on user performance in information-rich virtual environments. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, VRST '05, page 46–55, New York, NY, USA, 2005. Association for Computing Machinery. ISBN 1595930981. doi: 10.1145/1101616.1101626. URL https://doi-org.ezproxy.lib.vt.edu/10.1145/1101616.1101626.

[27] Wen Qi. Geometry based haptic interaction with scientific data. In *Proceedings of the 2006 ACM International Conference on Virtual Reality Continuum and Its Applications*, VRCIA '06, page 401–404, New York, NY, USA, 2006. Association for Computing

Machinery. ISBN 1595933247. doi: 10.1145/1128923.1128999. URL https://doi-org.ezproxy.lib.vt.edu/10.1145/1128923.1128999.

[28] Philip Schuchardt and Doug A. Bowman. The benefits of immersion for spatial understanding of complex underground cave systems. In *Proceedings of the 2007 ACM Symposium on Virtual Reality Software and Technology*, VRST '07, page 121–124, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595938633. doi: 10.1145/1315184.1315205. URL https://doi-org.ezproxy.lib.vt.edu/10.1145/1315184.1315205.

[29] Steven M. Seitz and Charles R. Dyer. View morphing. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, page 21–30, New York, NY, USA, 1996. Association for Computing Machinery. ISBN 0897917464. doi: 10.1145/237170.237196. URL https://doi.org/10.1145/237170.237196.

[30] Jonathan Shade, Steven Gortler, Li-wei He, and Richard Szeliski. Layered depth images. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 231–242, 1998.

[31] Harry Shum and Sing Bing Kang. Review of image-based rendering techniques. In *Visual Communications and Image Processing 2000*, volume 4067, pages 2–13. International Society for Optics and Photonics, 2000.

[32] Richard Skarbez, Nicholas F Polys, J Todd Ogle, Chris North, and Doug A Bowman. Immersive analytics: Theory and research agenda. *Frontiers in Robotics and AI*, 6:82, 2019.

[33] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[34] Colin Ware and Kathy Lowther. Selection using a one-eyed cursor in a fish tank vr environment. *ACM Trans. Comput.-Hum. Interact.*, 4(4):309–322, December 1997. ISSN 1073-0516. doi: 10.1145/267135.267136. URL https://doi-org.ezproxy.lib.vt.edu/10.1145/267135.267136.

[35] Colin Ware, Kevin Arthur, and Kellogg S. Booth. Fish tank virtual reality. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, CHI '93, page 37–42, New York, NY, USA, 1993. Association for Computing Machinery. ISBN 0897915755. doi: 10.1145/169059.169066. URL https://doi-org.ezproxy.lib.vt.edu/10.1145/169059.169066.

[36] Kamer Ali Yüksel, Alp Yucebilgin, Selim Balcisoy, and Aytul Ercil. Real-time feature-based image morphing for memory-efficient impostor rendering and animation on gpu. *The Visual Computer*, 29, 02 2012. doi: 10.1007/s00371-012-0718-8.

Appendices

Appendix A

User study application demo video

The following URL shows a demo video of the investigator performing the 2-AFC tasks on a Google Chrome web browser.

http://metagrid2.sv.vt.edu/~joo918/thesis_recording.mp4

Appendix B

Source code and image database

The following URL holds the complete code that the investigator used to build the web application for the user study.

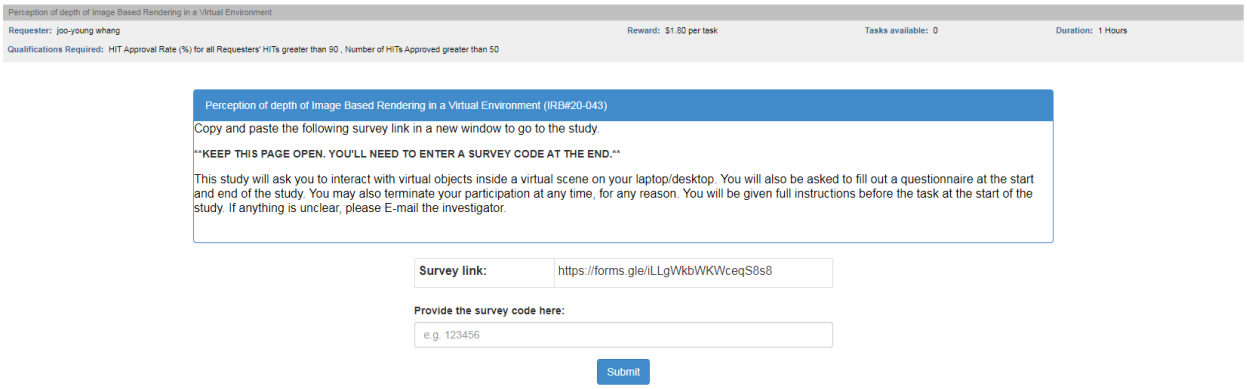
http://metagrid2.sv.vt.edu/~joo918/thesis_final/thesis_depthmap/

The *server* directory contains the NodeJS web application, and the directory *ibrobjs* inside it contains the WebGL shader code for the Cinema IBR objects and the Cinema image databases of the geons.

Appendix C

User study materials

C.1 Mturk HIT page



C.2 Google Form survey

This survey included the consent form, pre-study questionnaire, instructions to the user tasks, and the post-study questionnaire.

Depth Perception User Study

CONSENT FORM

Please click on the Google Doc URL and read the consent form carefully.
https://docs.google.com/document/d/1vkSwxVwY2eJyacQbfQdq3vgsHP5A_EzIdQrfaEKuR60/edit?usp=sharing
* Required

1. I have read the above consent form and I agree to participate in this study. *

   Mark only one oval.

   - ◯ Yes *Skip to question 2*
   - ◯ No

Pre-study Questionnaire

The following questions ask about your Mturk ID, demographic information, and experience using a VR headset.

2. What is your Mturk worker ID? *

   _____

3. What is your age? *

   Mark only one oval.

   - ◯ 18 - 25
   - ◯ 26-35
   - ◯ 36-45
   - ◯ Above 45

4. What is your gender? *

Mark only one oval.

- ◯ Male
- ◯ Female
- ◯ Prefer not to say

5. Do you wear glasses or contact lenses? *

Mark only one oval.

- ◯ Yes
- ◯ No

6. Do you have any known eye conditions? *

Mark only one oval.

- ◯ Near-sightedness
- ◯ Far-sightedness
- ◯ Color blindness
- ◯ Stereoblindness
- ◯ Other: _____

Please press next to continue.

Study

Please read through the ENTIRE instruction carefully before starting the study.

INSTRUCTION:

Click on the following link:
https://tinyurl.com/y7w7octh

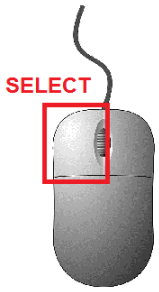
Wait until the "Please Wait..." message disappears.

After a short delay, you will be shown two boxes or two spheres.
You will try to decide which of the two boxes/spheres are protruding more towards you.
In the scene, you will see four boxes with the following options: "Left", "Maybe Left", "Maybe Right", "Right"
Click on the appropriate box to make a decision.
You can use the arrow keys on your keyboard to move left or right. (As shown below)
If the arrow keys do not work, click on a blank space in the scene once with your left mouse button.

Upon making a decision, a new set of boxes/spheres will appear. Repeat the process 64 times.
** If you experience nausea or motion sickness, please take a break immediately.**
The study is expected to take less than 20 minutes.

At the end of the study, you will be shown a passcode.
Please enter the passcode below and click next.
You will not be able to proceed if you enter the wrong passcode!
We will be keeping track of your activity in the application.
If you decide to stop the study, please close the window tab.

How to Move and Select



7. Enter the passcode: *

Post-study Questionnaire

The following questions ask about your experience about the study.

8. What was the most difficult part about performing the tasks? *

9. What was the most difficult part about using the mouse and keyboard interface? *

10. Please share any other feedback you have about the study. *

Thank you for your participation! The survey code is: 8190

Please click submit to finish the survey.
We will validate your answer and accept your Mturk HIT response.
MAKE SURE TO ENTER THE SURVEY CODE IN THE ORIGINAL MTURK PAGE.
THE SURVEY CODE IS 8190
MAKE SURE TO SUBMIT THE SURVEY!