1-1-2020

Assessing Distance Perception In Virtual And Augmented Realities With Electroencephalography

Ethan Luckett

ASSESSING DISTANCE PERCEPTION IN VIRTUAL AND AUGMENTED REALITIES WITH ELECTROENCEPHALOGRAPHY

A Thesis
presented in partial fulfillment of requirements
for the degree of Master of Science
in the Department of Computer and Information Science
The University of Mississippi

by

Ethan Luckett

May 2020

ABSTRACT

A common finding in spatial perception research is that subjects tend to underestimate distances in virtual reality compared to the real world. The degree and methods of measurement of underestimation vary between studies, but the trend of underestimation is consistent. This study uses electroencephalography as a neuroimaging technique to examine patterns of brain activity when fixating objects in near space and far space in the real world, in virtual reality, and in augmented reality. For the augmented reality condition, a custom optical see-through augmented reality head-mounted display (HMD) was built and calibrated. A calibration method was developed to correct the geometric distortion introduced by the HMD's optical combiners. This method also calibrates a motion tracker mounted on the HMD to allow for tracking of head movements.

ACKNOWLEDGEMENTS

I would first like to thank Dr. Adam Jones for his help along the way and for being an awesome advisor in general. Without both his undergraduate and graduate classes, this project would not have been possible. Second, I thank the other members of my committee, Dr. Yixin Chen and Dr. Dawn Wilkins, for teaching two of my favorite graduate classes. I would like to thank Dr. Dwight Waddell for letting me use his EEG lab and equipment and for his advice on EEG data collection and analysis.

Finally, I thank all of my friends and family for their support and patience.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

A common finding in spatial perception research is that subjects tend to underestimate distances in virtual reality compared to the real world. The degree of underestimation varies widely among studies from mild to severe. This is usually measured by evaluating a subject's accuracy in walking without vision to a target seen for a short time (i.e., blind walking) or by moving an indicator in depth to match the depth of a target (i.e., perceptual matching). The present study takes a different approach by aiming to measure brain activity in response to a target stimulus placed at specific depths in the real world, in virtual reality, and in augmented reality. Some previous studies have examined spatial perception-related brain activity in the real world, while others have done the same in virtual reality. This study is the first to examine both simultaneously with identical methodology and thus provides a side-by-side comparison. Brain activity is measured via electroencephalography (EEG), which has the benefit of needing much smaller equipment than some other neuroimaging methods, such as positron emission tomography (PET) or functional magnetic resonance imaging (fMRI). The latter two require room-sized equipment which can limit the kinds of virtual reality (VR) and augmented reality (AR) head-mounted displays (HMDs) that can be used. For this study, a custom AR HMD had to be built and calibrated. The following sections will discuss previous work in spatial perception research, the construction and calibration of the AR HMD, and a user study conducted using the AR HMD.

CHAPTER 2

RELATED WORK

2.1 Egocentric distance perception in virtual and augmented realities

A review by Renner et al. (2013) of 78 articles on egocentric distance perception in virtual environments found a mean perceived egocentric distance of only 74% of the actual distance in virtual environments. Jones et al. (2008) found that subjects estimated distances to be 94% of the actual distance in a real environment and 91% in a virtual environment. Another study by Willemsen et al. (2004) found 91% in a real environment and 47% in a virtual environment. Numerical results vary widely across experimental tasks and setups, but the trend of underestimation is clear. Interrante et al. (2006) showed that using a virtual environment that accurately models the real environment improves estimations to nearly 100% of the true distance. Interrante also found that scaling the virtual environment in size altered distance judgments. Kunz et al. (2009) tested both high and low visual quality virtual environments and found estimates of 83% and 78%, respectively. In Kunz's study, subjects practiced in a real-world hallway, but data was collected for the two virtual environments only. The environments were the same size and shape, but one had higher quality textures and contained objects of familiar size (e.g., a desk, wall clock, etc). Phillips and Interrante (2011) found that adding black lines to highlight major edges in a realistic environment caused subjects to underestimate distances by 17%. They suggested that the unrealism of the outlines was enough to degrade subjects' sense of presence in the environment. These studies suggest the importance of ensuring the virtual environment closely matches the real environment.

Swan et al. (2007) examined depth perception through an optical see-through AR HMD. They found a shift from underestimating to overestimating distances at a distance of 23 meters. When objects were mounted on the ceiling of the virtual environment, observers did not show a bias in estimations, but they underestimated distances when objects were placed on the floor. In both cases, distance estimations improved with each repetition.

2.2 Two visual pathways

There is a growing body of evidence to support a double dissociation between the areas of the brain that process near space (within arms' reach) and far space (beyond arms' reach) [Weiss et al. (2000); Goodale et al. (1992); Milner and Goodale (2008)]. Goodale et al. (1992) define this as a dissociation between *vision for action* and *vision for perception*. The two pathways are commonly referred to as the dorsal and ventral pathways. The dorsal, or upper, pathway includes areas in the upper occipital cortex and parietal lobe. The ventral, or lower, pathway includes areas in the lower occipital cortex and temporal lobe. The dorsal visual pathway processes near space and is responsible for processing the sizes of objects and their egocentric locations. This pathway is responsible for our ability to reach to and grasp objects near us. On the other hand, the ventral pathway processes far space and is responsible for determining the positions of farther objects relative to each other. The ventral pathway is responsible for what we consciously perceive as vision [Milner and Goodale (2008)].

Berti and Frassinetti (2000) tested the effects of tool use on distance perception in a single subject with lesions which negatively affected perception of near stimuli in the left visual field. At distances beyond near space, the subject performed normally. They found that using a tool that extends the subject's reach decreased performance when the tool brought the object within reach. Thus, the boundary between near and far space was extended by using the tool. Witt et al. (2005) showed that this occurs when the subject intends to use the tool but not when it is only held in the hand. These two studies show

that a subject's ability and intent to physically interact with a stimulus influences which pathway is used to visually process the stimulus.

A study by Beck et al. (2010) had subjects perform two bisection tasks in virtual environments in both near space and far space and measured brain activity using functional magnetic resonance imaging (fMRI). The horizontal bisection task asked the subject to gauge whether a vertical test tube was centered horizontally along a wooden shelf, while the vertical bisection task asked them to judge whether a horizontal toilet paper roll was centered vertically in a holder. The authors found stronger activations in dorsal areas in near space and stronger activations in ventral areas in far space, which contradicts the results found for real environments. The authors note that the HMD used had a limited resolution and low field of view. The environment itself also presented few depth cues.

CHAPTER 3

OPTICAL SEE-THROUGH AUGMENTED REALITY HMD

This study required an optical see-through AR HMD, but an HMD robust enough for use in studies was not available. This chapter describes the construction and calibration of a custom-built display.

3.1 Construction

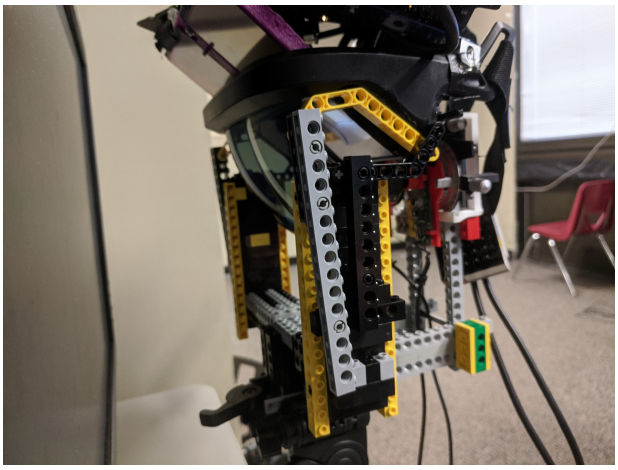

Figure 3.1. Front view of the AR HMD

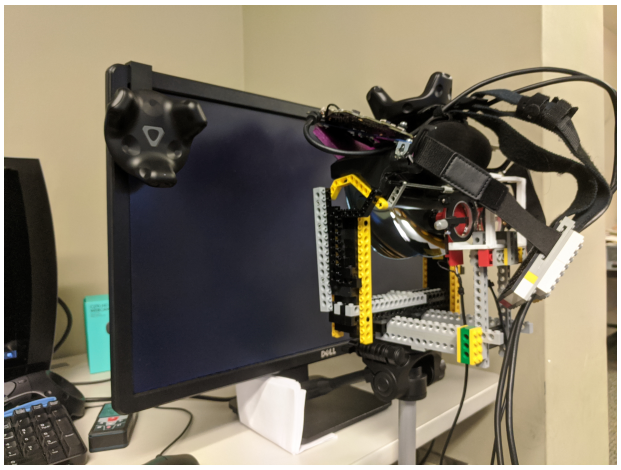

Figure 3.2. Apparatus used to calibrate the AR HMD

Figure 3.1 shows a front view of the AR HMD resting on the calibration apparatus. The base of the HMD is a plastic frame fitted with parabolic optical combiners and an elastic head strap. There is a slot on the top of the display that was designed to fit a smartphone. The smartphone's screen is reflected by the optical combiners and focused into the wearer's eyes. The optical combiners are partially transparent mirrors, which allows virtual objects to be superimposed on the view of the real world. In order to be used in the

study, the pixels sent to the HMD needed to be controlled precisely. To accomplish this, the display panels from two OSVR HDK1s [OSVR (2016)] were attached to a 3D printed mount, which was then placed in the smartphone slot. Each panel is a 5.5" OLED display with a resolution of 1920x1080 oriented vertically, which limits the maximum per-eye resolution to approximately 1080x800. The per-eye horizontal field of view is approximately 48°. An HTC Vive tracker [HTC (2016)] was rigidly attached to the top of the HMD to track the wearer's head movements.

3.2 Calibration

The HMD needs to be calibrated before it can be used. This process is broken into two phases: optical calibration, which ensures the image seen by the wearer is rectilinear, and tracker calibration, which ensures head positions and movements are tracked accurately. These steps guarantee that the light from the virtual world matches the light from the real world on which it is superimposed.

3.2.1 Apparatus

The apparatus used to calibrate the HMD is shown in Figure 3.2. The HMD rests on an adjustable mount attached to a camera tripod. Two Logitech C270 web cameras with 180°fisheye lenses are secured to an ophthalmic trial frame with 3D printed mounts. A trial frame is a special eyeglass frame with lens mounts and fine adjustment knobs used by ophthalmologists to fit a person for glasses. The 3D printed camera mounts are designed to fit in the trial frame's lens mounts. The trial frame is attached to the calibration apparatus and positioned to approximate human eye positions relative to the optical combiners. The optics of the HMD are fixed, but using a trial frame allows the distance between the cameras to be adjusted to match an individual's interpupillary distance (IPD). For this study, the IPD was fixed to 63mm, which is the population average [Jones et al. (2016)]. The calibration rig is placed in front of a monitor and aligned so that the front of the rig is parallel to the monitor's surface. Both are levelled with respect to gravity with the digital level function of

a laser rangefinder.

3.2.2 Optical distortion

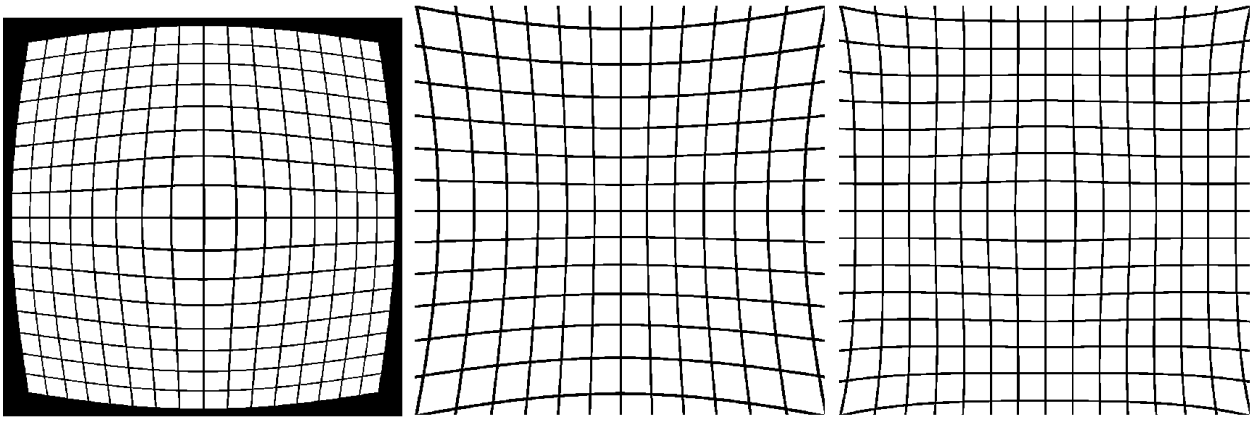

Figure 3.3. Types of radial distortion. Images show barrel, pincushion, and mustache distortion, respectively.

As with any optics, the optical combiners introduce distortion to the image displayed to each eye. Many HMD lenses exhibit radial distortion, which is where each pixel's position is shifted along the ray from the optical center to the pixel position as a function of the distance from the optical center. This comes in the form of barrel distortion, pincushion distortion, and mustache distortion, which can be seen in Figure 3.3. Barrel distortion shifts pixels more toward the center as the distance from the optical center increases, pincushion shifts pixels away from the center, and mustache distortion is a combination of the two where the distortion direction changes at a certain distance. These forms of distortion are often a result of the shape of the lens or the materials used to make it. Another form of distortion unrelated to the composition and shape of the lens is tangential distortion, where straight lines are no longer parallel as a result of the plane of the lens not being parallel to the camera sensor. Both types of distortion can be corrected simultaneously by the Brown-Conrady model Brown (1966); Conrady (1919).

The distortion introduced by the AR HMD's optical combiners is different than a typical lens as can be seen in Figure 3.4. There is a vertical tangential component; vertical

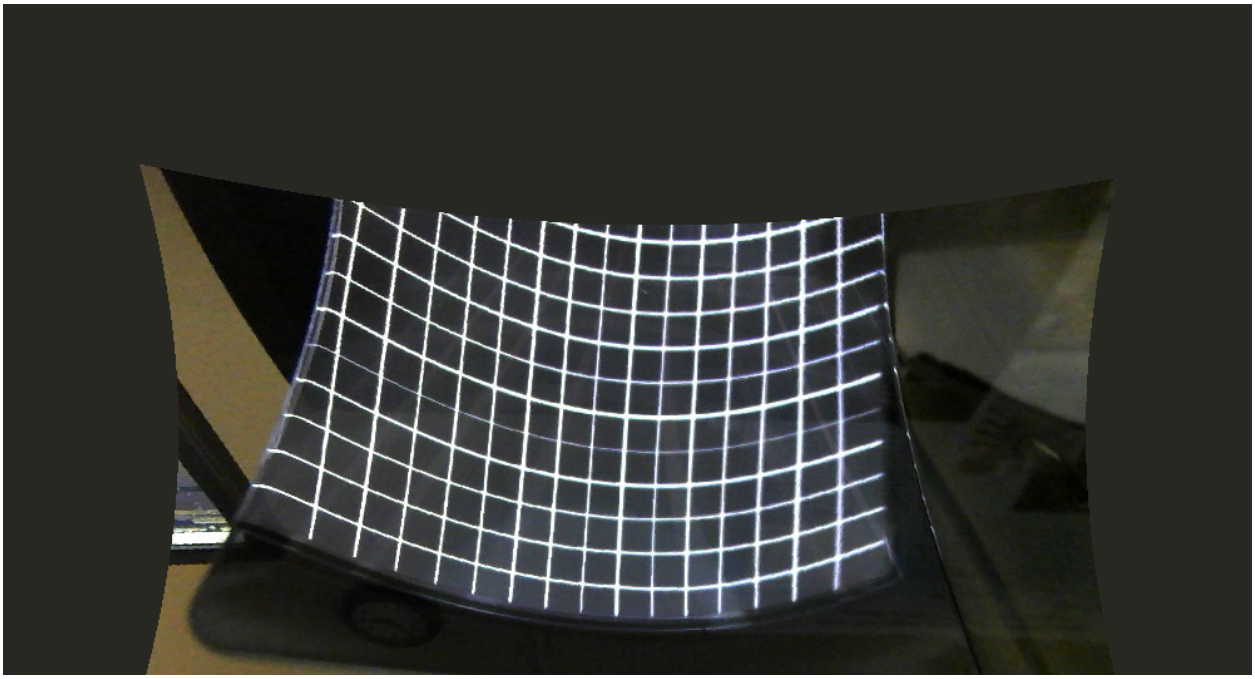

Figure 3.4. Distortion from the AR HMD's optical combiners. This image was taken with one of the web cameras. The strong pincushion distortion of the edges is due to the correction applied to remove the fisheye lens's barrel distortion.

lines are skewed more inwards closer to the top of the field of view. This could be due to the top-to-bottom curvature of the optical combiners towards the wearer's head. There is also a horizontal component which is neither radial nor tangential. Horizontal lines become convex parabolas, which could be a result of the side-to-side curvature of the lenses. Since the optical combiners are mirrors, they also reflect the image horizontally. A method was adapted from Jones et al. (2015) which is capable of correcting all of these in a single procedure. To perform the correction, it is necessary to establish a mapping from each pixel in the distorted image to the corresponding pixel in the correct image. *Distorted image* means the image the wearer sees in the absence of any software correction, and *correct image* means the image as it appears on the screen that we wish to be displayed to the wearer. The optical calibration process is broken into four steps. First, a mapping from camera pixels to monitor pixels is created. Second, camera pixels are mapped to virtual pixels on the AR HMD. Third, these two maps are used to create a map from virtual pixels to monitor pixels. Lastly, the final

map is converted into a counterdistortion texture, which is then used by a graphics shader as a lookup table to map individual pixels to their correct, undistorted locations.

3.2.3 Mapping camera space to monitor space

A mapping from each camera pixel (in *camera space*) to the corresponding pixel on the monitor (*monitor space*) is created. Using pyglet, a python package that allows easy access to OpenGL (a graphics rendering library) calls, a vertical white line is drawn that scans horizontally across the monitor one pixel at a time. At each position, an image is captured from the camera via OpenCV. OpenCV is a computer vision library that includes functions for capturing images from a camera. The image is converted to grayscale, and a list of y, x tuples is created, where y is the vertical pixel position and x is the horizontal pixel position of the most luminous pixel at that y position. Low luminosity values ($<$50 on a 0-255 scale) are filtered out to ignore y positions where the line is out of frame, and images with too few remaining points are discarded. A parabola is fit to the set of remaining scanline points. The result is a mapping from x positions in monitor space to parabolas in camera space. The coefficients of these parabolas are then each fit to a cubic polynomial. The second fitting step creates a set of distortion functions that smoothly describes the horizontal distortion at every point. This process is repeated with a vertical scan of horizontal lines, which gives a description of the vertical distortion.

In practice, capturing an image from the camera is a relatively slow process. The camera buffers images, so it is necessary to capture multiple images in order to get an updated view and remove rolling shutter effects. To reduce the calibration time to a reasonable level, the line is set to move 5 pixels at a time instead of 1. This still leaves plenty of points to obtain a good fit in the regression steps.

We can now create the mapping from camera space to monitor space. The map is an array of dimensions $h \times w \times 2$, where h and w are the camera height and width in pixels, respectively. For each y, x coordinates in the map, the first element of the tuple is set to

the x position of the vertical line which appears at y, x in the camera image, and the second element is similarly set to the y position of the horizontal line. Note that the array is indexed in row-order, but the values are column-order. If more than one line appears at the same camera coordinates, lines with greater y or x values overwrite previous values as the map is incrementally created. The full procedure is repeated for the left and right cameras, giving two monitor maps, m_l and m_r.

3.2.4 Counterdistortion

The procedure described above is also used to generate a map from the left and right cameras to the corresponding screen on the AR HMD, giving the left and right virtual maps v_l and v_r. By combining the monitor and virtual maps, virtual pixels can be mapped to monitor pixels via their corresponding camera pixels.

The virtual-monitor maps are arrays of dimensions $w \times w \times 2$, where w is the width in pixels of one of the AR HMD screens. The virtual map is used as a lookup table to set the value at each coordinate in this new array, which is shown in Algorithm 1.

Algorithm 1: virtual-monitor map construction

input : Virtual and monitor maps v and m of size $r \times c$
output: Virtual to monitor map vm

for $y \leftarrow 0$ **to** $r - 1$ **do**
    for $x \leftarrow 0$ **to** $c - 1$ **do**
        $vc \leftarrow v[y][x]$
        $vm[vc.y][vc.x] \leftarrow m[y][x]$

In order to convert this map into a counterdistortion texture, the point where each camera's optical axis intersects the monitor must be known. This is simply the point in the center of the monitor map. Ideally, the optical axes would be normal to the surface of the monitor, the pixel y value of this point for the left and right eyes would be the same, and the x values would differ by the IPD divided by the width of a monitor pixel. Due to misalignments caused by imperfect mounts, this was not the case. Fortunately, the offsets

were small and can be corrected without adding any visible distortion. First, the horizontal pixel position of the midline of the AR HMD is measured by rendering a vertical line on the monitor and moving it until the line, the midline of the HMD, and a reference point on the back of the calibration apparatus appear collinear. The correction is then given by Algorithm 2.

Algorithm 2: Optical center correction

input : Monitor maps m_l and m_r of size $r \times c$
Size of a pixel s_p
Midline x position r
Interpupilary distance d

output: Adjusted optical centers adj_l and adj_r

$c_l \leftarrow m_l[r/2][c/2]$
$c_r \leftarrow m_r[r/2][c/2]$
$o_l \leftarrow (d/2)/s_p + |r - c_l.x|$
$o_r \leftarrow (d/2)/s_p + |r - c_r.x|$
$h \leftarrow (c_l.y + c_r.y)/2$
$adj_l \leftarrow c_l + (o_l, h)$
$adj_r \leftarrow c_r + (o_r, h)$

Next, the *extent* of the distortion texture must be determined. This affects the angular size of the rendered images in the wearer's field of view and is measured in pixels. For each of the HMD mirrors, a vertical line is displayed on the monitor and moved so that it lines up with the left edge of the mirror as seen through the camera. The pixel x position is measured, and this is repeated for the right edge. The difference in pixel positions defines the *extent*. The distance between the camera and monitor is measured with a ruler, which allows the field of view (FOV) of each mirror to be calculated trigonometrically. The counterdistortion texture is then computed in Algorithm 3.

This defines a window of size (extent, extent) centered at the adjusted optical center and normalizes the values within it. Values outside the interval [0, 1] are set to 0. A layer of zeros is added to the array to increase the dimensions to $w \times w \times 3$.

Algorithm 3: Texture map construction

input : Adjusted optical center *center*
Virtual-monitor map *vm*
output: Texture map T

$s = (extent, extent)$
$corner = center - s/2$
$T = (vm - corner)/s$

3.2.5 Applying the texture

In Unity3D, the horizontal FOV of each virtual camera is set to the mirror FOV measured earlier, and the aspect ratio is set to 1. The view of the virtual camera is rendered to a RenderTexture. RenderTextures allow the view of a virtual camera to be rasterized to a texture. A custom shader was written that uses the counterdistortion texture to map positions in the RenderTexture to positions in a new, corrected texture. Unity3D's `Graphics.Blit` function is used to feed the RenderTexture into the shader and render the resulting texture directly to the screen.

3.2.6 Tracker calibration

Once the optical distortion has been corrected, it is necessary to align the pose of the virtual cameras to match the pose of the physical cameras with respect to the HMD-mounted Vive tracker. This ensures virtual objects are visually aligned with their real-world counterparts and remain aligned during head movements. The virtual cameras are attached to an empty GameObject (Unity3D's term for a container object) and offset laterally by half of the IPD. The camera container is then attached to the GameObject representing the HMD's Vive tracker. This simplifies the tracker calibration to finding the relative pose C of the camera container object with respect to the HMD tracker object H.

The same apparatus used for optical calibration is used for tracker calibration. A second Vive tracker is attached to a 3D printed mount that snugly fits against the top-left corner of the monitor (see Figure 3.2). The Vive tracker developer guidelines specify

precisely where the tracking origin is on the physical tracker [HTC Corporation (2017)]. This gives a reference point M on the monitor whose position is known in both real and virtual coordinates. Unity3D is used to capture 2 seconds of pose data from each tracker, which is averaged to give two poses.

To find the translational offset T, the x, y pixel offset from the top left monitor pixel to the midpoint between the adjusted left and right optical centers is computed. This is then multiplied by the pixel size to give the x and y components of T. The z component of T is the previously measured distance from the camera to the monitor surface. This gives the position of C relative to M. The offset between M and H is easily calculated from the tracker positions recorded in Unity3D, and the final offset between C and H is the sum of these offsets.

The computation of the rotational offset between C and H is straightforward. First, a relative rotation must be determined that, when applied to the rotation of M, gives the desired absolute orientation of C. The desired orientation has the z-axis of the coordinate system pointing toward the monitor and normal to its surface with the y-axis pointing up. Since the rigid 3D printed mount ensures the coordinate axes of M are all either parallel or orthogonal to the axes of the monitor, this can be determined by eye as a combination of rotations by 90°or 180°. The offset of C relative to H is given by $h^{-1} * m * o$, where $*$ represents quaternion multiplication and h, m, and o are quaternion representations of H, M, and the offset required to reach C from M, respectively.

3.2.7 Calibration quality

The view through one of the AR HMD's combiners is shown in Figure 3.5. This was taken with a smartphone, and the camera's autofocus is focused on the lens itself. In reality, the virtual objects appear much sharper than this. At near distances, the overall quality of the calibration is good. There is an offset of approximately 1-2° though, and this becomes more apparent as the distance between the object and the observer increases. This was

Figure 3.5. View of a virtual Vive controller superimposed on a real Vive controller through the AR HMD

mitigated by manually adjusting the position of the camera container in Unity3D, but there is some remaining translational offset that cannot be corrected this way. This offset varies if, for example, the wearer turns their entire body while holding the controller in front of them. However, the direction of offset remains constant relative to the real world. This appears to be an artifact of the Vive tracking rather than an issue with the AR HMD's calibration.

CHAPTER 4

USER STUDY

A study was conducted to examine brain activations in subjects fixating on an object in near space and far space in different environments. Previous studies have examined brain activity related to spatial perception in the real world (e.g. Weiss et al. (2000)) and in virtual reality (e.g. Beck et al. (2010)), but the present study is the first to examine both simultaneously in order to make a direct comparison. Initially, the study design planned for eight subjects, but the number of subjects had to be limited for two reasons. The original EEG cap had multiple nonfunctional electrodes, and there were concerns about the quality of the data gathered with it. This delayed the start of data collection until a new cap was ordered, which arrived in early March 2020. Shortly after data collection began, concerns grew about the subjects' safety during the novel coronavirus outbreak, which forced the study to stop after three subjects. Thus, this should be treated as a pilot study, although that was not the original intention.

All three subjects were male and were recruited from the Department of Computer and Information Science of the University of Mississippi. Each subject filled out an informed consent form approved by the Institutional Review Board of the University of Mississippi as well as a demographics questionnaire and a simulator sickness questionnaire (SSQ). Once subjects finished the study, they filled out a second SSQ and were compensated for their time. The SSQs served to determine whether the subjects had any symptoms of motion sickness as a result of experiencing a virtual environment.

Table 4.1. Condition order for each subject

Subject	Conditions					
1	RealNear	RealFar	VRNear	VRFar	ARNear	ARFar
2	RealFar	RealNear	VRFar	VRNear	ARFar	ARNear
3	RealFar	RealNear	VRNear	VRFar	ARNear	ARFar

4.1 Experimental Setup

The real environment was a cluttered 4.3m x 5.8m space featuring multiple desks, chairs, computers, and other furniture. The subject was seated in a comfortable chair placed facing parallel to the long edge of the room and approximately 2.5m from the wall. The center of the subject's field of view from the chair was kept visually sparse. A virtual environment (VE) with dimensions matching those of the real environment was created in Unity3D. Some smaller objects, namely computers and monitors, that were visible in the far periphery in the real world were not modeled in the virtual environment. Shadows were reproduced in the VE by placing point light sources at the approximate locations of the corresponding real-world light sources. A computer monitor was placed on a rolling table, and both were modeled and textured. A white crosshair was displayed at the center of the monitor in all environments. In the AR environment, the crosshair was rendered through the AR HMD and superimposed on the center of the real-world monitor. In the real world, a laptop running a simple OpenGL script rendered the crosshair, and in the virtual environment, the crosshair was modeled using 1mm thick cubes placed 0.1mm in front of the surface of the virtual monitor screen. In all environments and conditions, the crosshair subtended 5° in the visual field. A Vive tracker was attached to the rolling table to align the pose of the virtual table and monitor to the real ones. A Vive controller placed on the corner of a desk was used as a reference point to automatically align the virtual room with the real room.

4.2 Experimental Design and Procedure

The study uses a 3×2 (*environment* $\times$ *distance*) design. The environment was either the real, augmented, or virtual environment. The distance was either 75cm (near) or 2m

(far). The near distance of 75cm was chosen as it was the closest the table and monitor could be placed to the chair. At this distance, subjects were required to lift their legs to rest them on a horizontal bar on the bottom of the table. According to the Man-Systems Integration Standards [NASA], this distance would be beyond arms' reach for approximately 5% of American males. For each subject, the distance between the subject's back and the tip of their outstretched hand was measured similarly to the standard measurement. This distance varied between 87 and 89 cm, which puts the near distance well within arms' reach. Each subject's eye height was measured, and the height of the table was adjusted so that the center of the monitor was approximately at eye level. Conditions were counterbalanced according to Table 4.1. The table was initially placed at the midpoint between the two distance conditions. Subjects closed their eyes, and the table was moved to the first distance condition. After a countdown, they were instructed to open their eyes and focus on the center of the crosshair while thinking about pointing to it. EEG data was collected for 60 seconds after which subjects closed their eyes again. The rendering of the crosshair was disabled and the table was moved to the other distance. The crosshair was re-enabled, and the procedure was repeated. The movement of the table was likely audible, but this was consistent across all conditions. Each distance condition occurred once per environment. At the end of each environment condition, the table was reset to the midpoint position, and subjects were allowed to open their eyes while the Vive HMD or AR HMD was placed on their head. Each HMD was adjusted until it was comfortable and the image was in focus.

4.3 Noise and the HMDs

During the AR condition for subject 1, all channels on the EEG cap became disconnected. The AR HMD is quite heavy and awkward to wear, so it is possible that the ground electrode lost connection with the scalp while getting the HMD in place. An attempt to resolve this was unsuccessful, so the procedures continued and the data for the AR condition for subject 1 was excluded from analysis. To account for the possibility of this disrupting

data collection for subsequent subjects, the AR condition was tested last for the remaining subjects. During the AR condition for subject 3, the AR HMD began to slide down the subject's forehead, and the subject tensed his forehead to prevent this. This resulted in extremely noisy EEG with a magnitude many times larger than the other conditions, so the AR condition for subject 3 was excluded as well.

4.4 Data acquisition

Data were recorded with a Compumedics Neuroscan Quik-Cap with 40 electrodes. A2 placed on the right mastoid process was used as the reference electrode. The four ocular channels were not used and so were removed, which leaves 35 usable channels. The cap was connected through a NuAmps amplifier to a computer running Windows XP. Scan4.3 was used to acquire the data. Each of the six possible conditions listed in Table 4.1 was saved to a separate file for a total of 18 datasets.

4.5 Data analysis

Continuous EEG recordings were analyzed with EEGLAB [Delorme and Makeig (2004)]. Data were re-referenced to the A2 electrode after acquisition. This equates to subtracting the signal at A2 from all other electrodes, and its purpose is to remove noise. After re-referencing, a 1-40Hz band pass filter was applied. Each dataset was inspected by eye for clearly anomalous sections (regions where the amplitude on most channels was much higher than the surrounding regions) and bad channels, which were removed. Independent component analysis (ICA) was then performed on all datasets simultaneously through the STUDY functionality of EEGLAB. This decomposes the continuous multi-channel EEG data into weighted sums of the channel data, whose positions in 3D can then be estimated. The ICLabel EEGLAB plugin was used to identify and mark ICA components that corresponded to artifacts such as eye-blinks and heartbeats. These components were excluded from further analysis. In order to obtain a 3D position for each component, a 3D position for each

electrode is needed. The Neuroscan EEG cap uses the standard 10-20 layout, in which distances between electrodes are either 10% or 20% of the total front-back or left-right distance over the skull. Since the size and shape of each subject's skull varies, the exact positions of electrodes can vary. Ideally, an MRI scan of each subject's head would facilitate determining electrode positions, but since none were available, the BESA model included in EEGLAB was used. The BESA model assumes a spherical skull shape and positions each electrode on a sphere according to the 10-20 layout. The DIPFIT plugin of EEGLAB [Oostenvelt et al. (2003)] was used to fit dipole locations for each component for each dataset. Since the dipole positions will vary between subjects, it is necessary to create clusters of nearby components in order to make observations about the subject population as a whole. This was accomplished with EEGLAB's component clustering functionality, which uses principal component analysis to reduce the dimensionality of the continuous data and KMeans to cluster the result. The dipole density was computed for each cluster.

4.6 Results and Discussion

For the real and VR conditions, data from all three subjects were analyzed. Figures 4.1, 4.2, 4.3, and 4.4 show the overall activity for each of the four combinations of environment and distance. Figure 4.5 shows an image comparison between each of the six pairs of conditions. Comparisons for individual subjects are available in the appendix. In the comparisons, brighter colors indicate a higher similarity between the two images. There is a large amount of between-subjects variability in the regions of activity in all conditions. This could be partially due to varying head shapes and sizes among subjects. Another possible source of variance is the number of electrodes available. After removing bad channels, 32 channels were available, but this could be too few to get an accurate position for the large number of sources of activity. In general, EEG is fairly noisy and has a lower spatial resolution than other neuroimaging methods. Additionally, at least twice as many subjects as were run here would be needed to perform a proper analysis. Still, some general patterns can be seen.

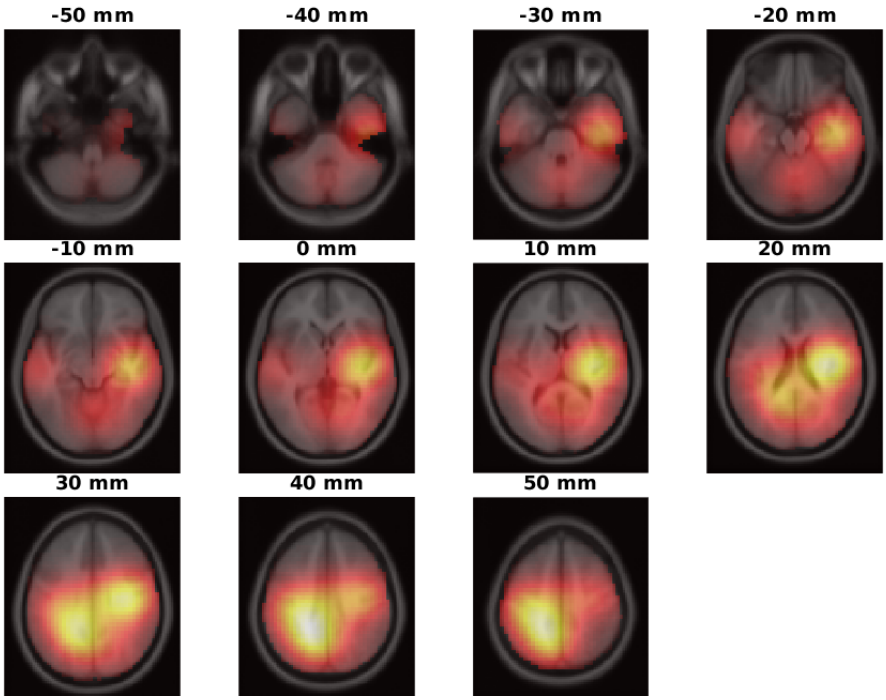

Figure 4.1. Overall activity in the real near condition

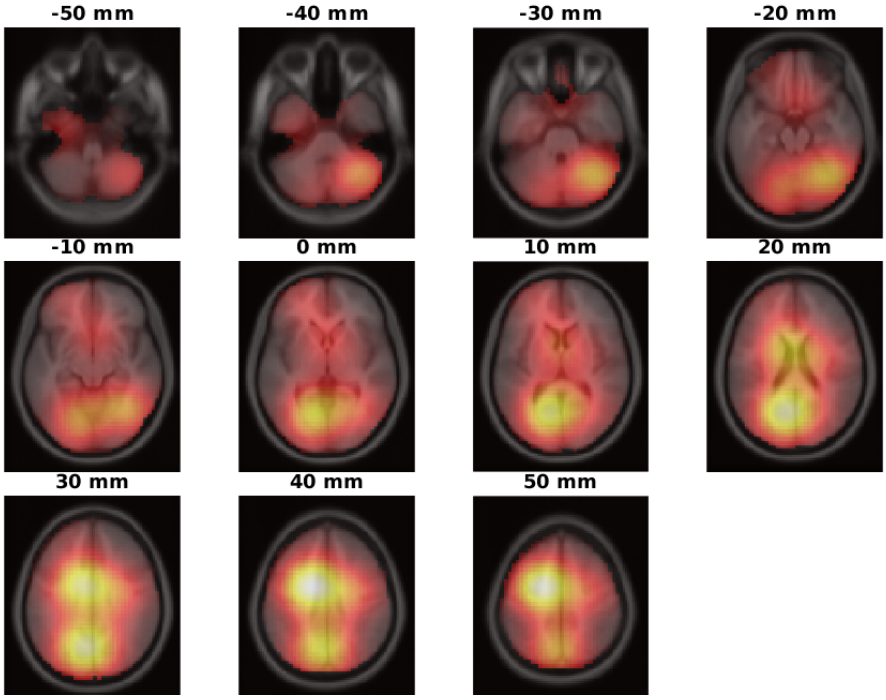

Figure 4.2. Overall activity in the real far condition

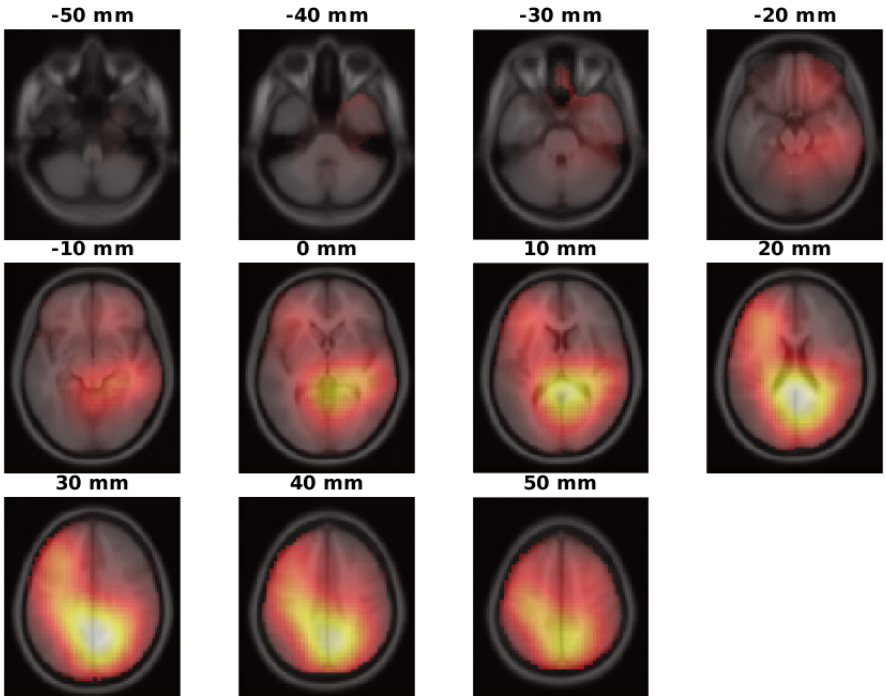

Figure 4.3. Overall activity in the VR near condition

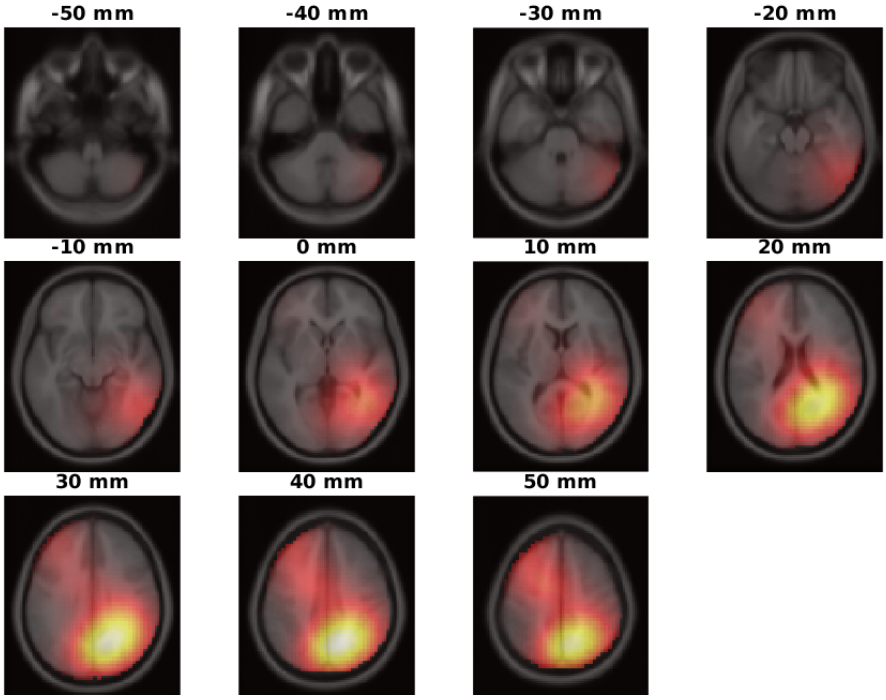

Figure 4.4. Overall activity in the VR far condition

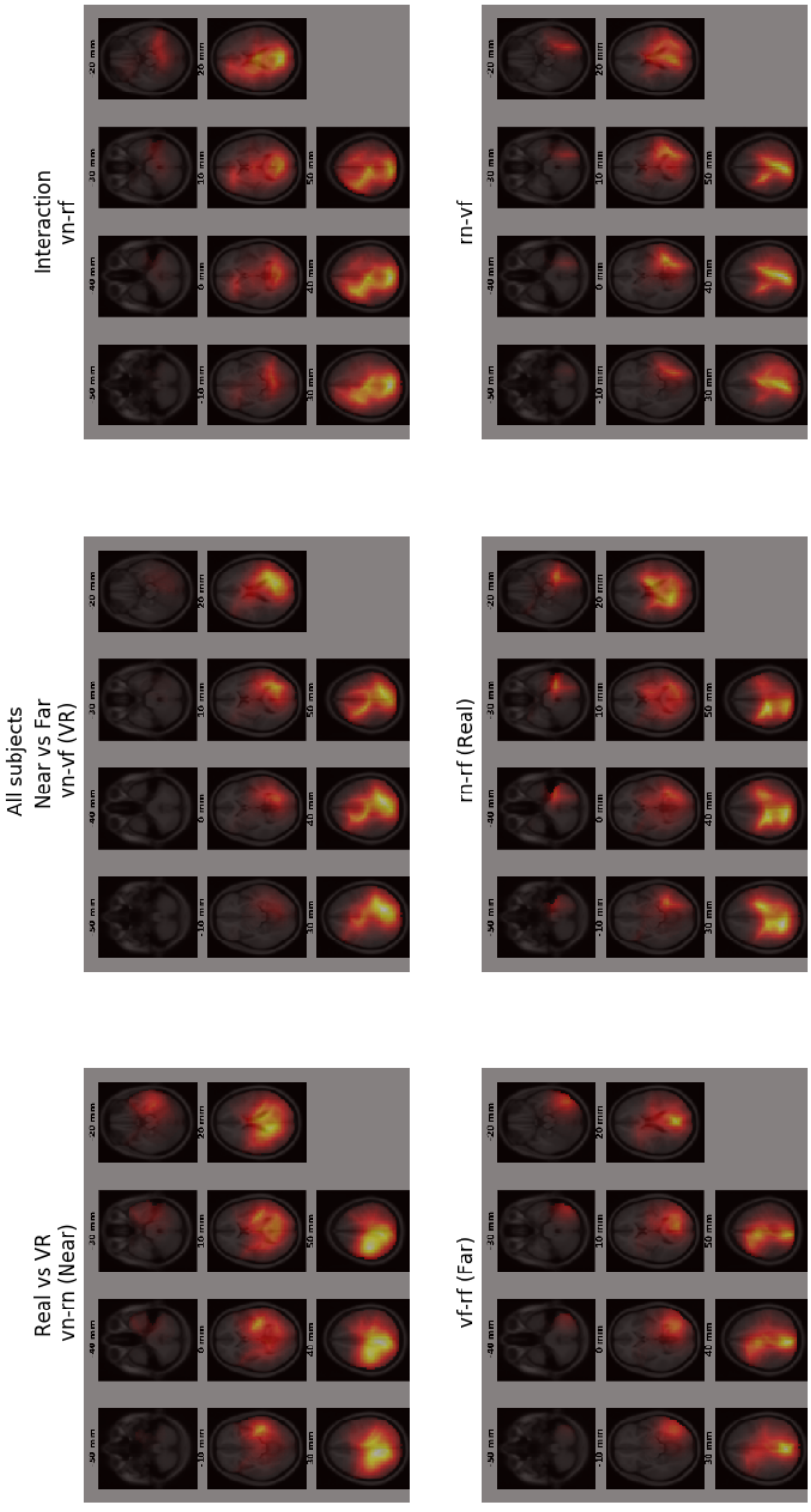

Figure 4.5. Comparison between conditions for overall activity

All conditions share some similar regions of activity, which is to be expected since the task varies only slightly between conditions. Overall, in the near conditions in both the real world and in VR, the activity in the parietal lobe appears to be greater than in the far conditions. This agrees with previous research in the real world. The comparisons between conditions for subjects 2 and 3 are similar, which suggests the regions of activity are similar. Another interesting comparison is between the VR near and real far conditions. In general, these conditions had a higher similarity than any other pair of conditions for all three subjects, which suggests they were processed by similar regions of the brain. This could be the result of the straps from the Vive HMD pressing parts of the EEG cap against the head, which may affect the electrodes' impedance. Another possible explanation is that this is the result of the Vive's optics being focused at infinity. The Vive's optics, like most modern VR HMDs, are designed to collimate the light emitted from the display panel. This ensures the full field of view through the lens remains in focus as the wearer's eyes move. At near distances, the eyes must converge to focus on the object. At the same time, the accommodation reflex changes the shape of the eye to focus the light coming from the object. When an object is nearby but the light coming from that object is focused as if it were far away, convergence and accommodation are in conflict. This conflict is common among nearly all VR HMDs and is known to cause eye strain. It is possible that this conflict could be causing the brain to process the VR near condition similarly to the far conditions. However, considering the low number of subjects and high variability between subjects, it is impossible to say this with certainty.

A separate analysis was performed for subject 2 that included the AR conditions. Results are shown in Figure 4.6. There are similarities in the regions of activity across all three environments for both the near and far conditions. Overall, the real world, VR, and AR seem to show some similarities in the way they are processed by the brain. In VR, the processing of objects in near space seems to share some characteristics of the processing of objects in far space.

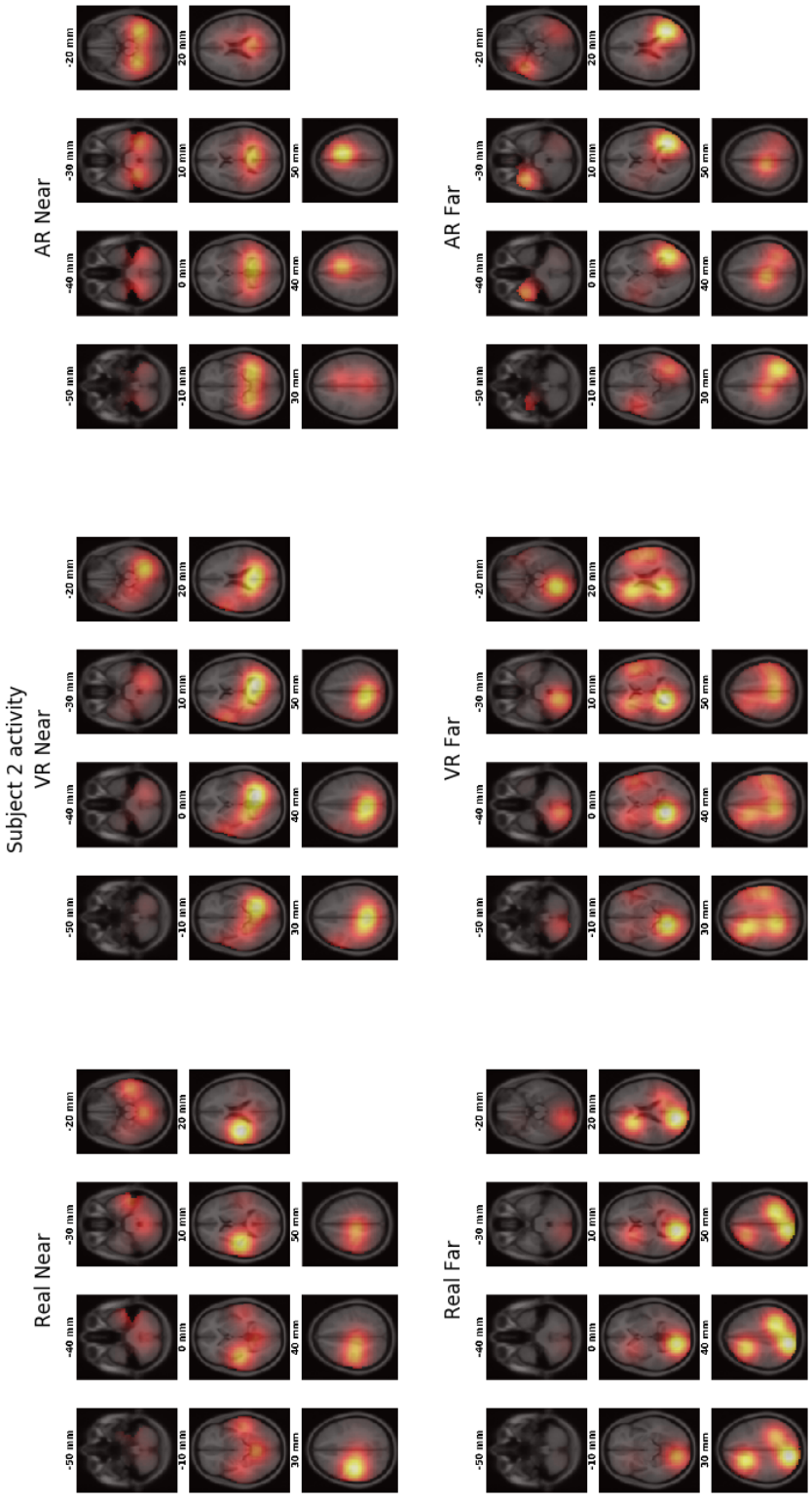

Figure 4.6. Activity across all conditions for subject 2

4.7 Future Work

The AR HMD needs to be improved. It is quite heavy, and the current head mount makes it difficult to put on another person. The straps are difficult to adjust, and it is hard to determine when the HMD is positioned properly. A more rigid head mount and better weight balance would solve these problems. Additionally, there is still a small offset after the calibration procedure which had to be corrected manually. Part of this offset seems to be an issue with Vive tracking. The other part could be a problem with the tracker calibration method, which could be related to the Vive tracking issue.

In a future study, more subjects could be run, which would likely give more conclusive results. The study procedures also have room for improvement. Currently, placing the HMDs on the subject is risky; without great care, the cap can be disrupted enough to negatively affect data collection. One way to solve this would be to rigidly mount the HMD to a pivoting arm that could be adjusted to position the HMD on the subject's face without the use of tight straps.

BIBLIOGRAPHY

BIBLIOGRAPHY

Beck, L., M. Wolter, N. F. Mungard, R. Vohn, M. Staedtgen, T. Kuhlen, and W. Sturm (2010), Evaluation of spatial processing in virtual reality using functional magnetic resonance imaging (fmri), *Cyberpsychology, Behavior, and Social Networking*, 13(2), 211–215.

Berti, A., and F. Frassinetti (2000), When far becomes near: Remapping of space by tool use, *Journal of cognitive neuroscience*, 12(3), 415–420.

Brown, D. C. (1966), Decentering distortion of lenses, *Photogrammetric Engineering and Remote Sensing*.

Conrady, A. (1919), Lens-systems, decentered, *Monthly notices of the royal astronomical society*, 79, 384–390.

Delorme, A., and S. Makeig (2004), Eeglab: an open source toolbox for analysis of single-trial eeg dynamics including independent component analysis, *Journal of neuroscience methods*, 134(1), 9–21.

Goodale, M. A., A. D. Milner, et al. (1992), Separate visual pathways for perception and action.

HTC (2016), Htc vive, https://www.vive.com/us/.

HTC Corporation (2017), HTC Vive Tracker Developer Guidelines.

Interrante, V., B. Ries, and L. Anderson (2006), Distance perception in immersive virtual environments, revisited, in *IEEE virtual reality conference (VR 2006)*, pp. 3–10, IEEE.

Jones, J. A., J. E. Swan, G. Singh, E. Kolstad, and S. R. Ellis (2008), The effects of virtual reality, augmented reality, and motion parallax on egocentric depth perception, in *Proceedings of the 5th symposium on Applied perception in graphics and visualization*, pp. 9–14.

Jones, J. A., L. C. Dukes, D. M. Krum, M. T. Bolas, and L. F. Hodges (2015), Correction of geometric distortions and the impact of eye position in virtual reality displays, in *2015 International Conference on Collaboration Technologies and Systems (CTS)*, pp. 77–83, IEEE.

Jones, J. A., D. Edewaard, R. A. Tyrrell, and L. F. Hodges (2016), A schematic eye for virtual environments, in *2016 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 221–230, IEEE.

Kunz, B. R., L. Wouters, D. Smith, W. B. Thompson, and S. H. Creem-Regehr (2009), Revisiting the effect of quality of graphics on distance judgments in virtual environments: A comparison of verbal reports and blind walking, *Attention, Perception, & Psychophysics*, 71(6), 1284–1293.

Milner, A. D., and M. A. Goodale (2008), Two visual systems re-viewed, *Neuropsychologia*, 46(3), 774–785.

NASA (1995), Man-System Integration Standards, Volume 1, Section 3, *Standard*, NASA.

Oostenvelt, R., A. Delorme, and S. Makeig (2003), Dipfit: equivalent dipole source localization of independent components.

OSVR (2016), OSVR website, http://www.osvr.org/hdk1.html.

Phillips, L., and V. Interrante (2011), A little unreality in a realistic replica environment degrades distance estimation accuracy, in *2011 IEEE Virtual Reality Conference*, pp. 235–236, IEEE.

Renner, R. S., B. M. Velichkovsky, and J. R. Helmert (2013), The perception of egocentric distances in virtual environments-a review, *ACM Computing Surveys (CSUR)*, 46(2), 1–40.

Swan, J. E., A. Jones, E. Kolstad, M. A. Livingston, and H. S. Smallman (2007), Egocentric depth judgments in optical, see-through augmented reality, *IEEE transactions on visualization and computer graphics*, 13(3), 429–442.

Weiss, P. H., J. C. Marshall, G. Wunderlich, L. Tellmann, P. W. Halligan, H.-J. Freund, K. Zilles, and G. R. Fink (2000), Neural consequences of acting in near versus far space: a physiological basis for clinical dissociations, *Brain*, 123(12), 2531–2541.

Willemsen, P., M. B. Colton, S. H. Creem-Regehr, and W. B. Thompson (2004), The effects of head-mounted display mechanics on distance judgments in virtual environments, in *Proceedings of the 1st Symposium on Applied perception in graphics and visualization*, pp. 35–38.

Witt, J. K., D. R. Proffitt, and W. Epstein (2005), Tool use affects perceived distance, but only when you intend to use it., *Journal of experimental psychology: Human perception and performance*, 31(5), 880.

APPENDICES

APPENDIX A

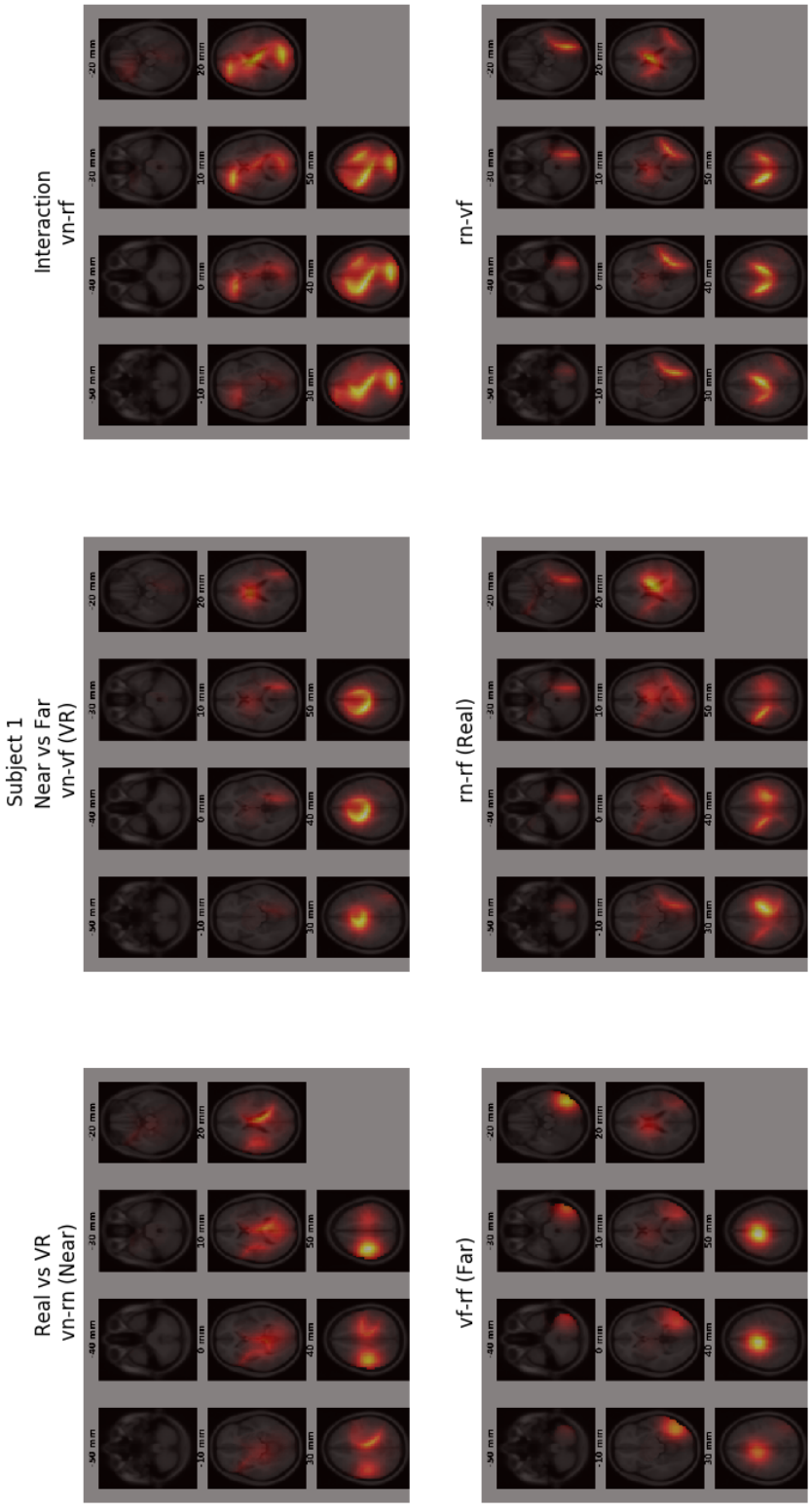

Figure A.1. Activity comparison between conditions for subject 1

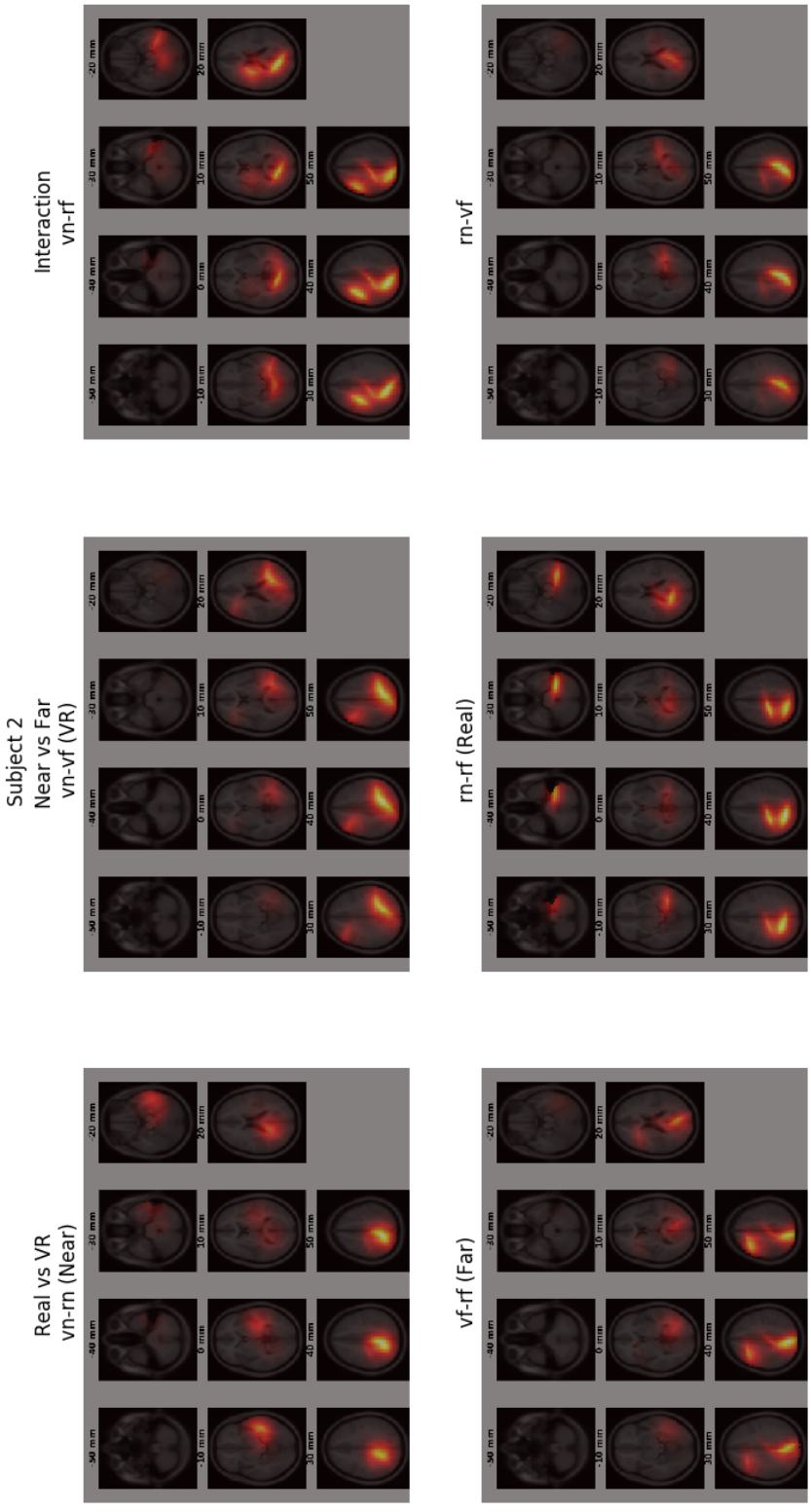

Figure A.2. Activity comparison between conditions for subject 2

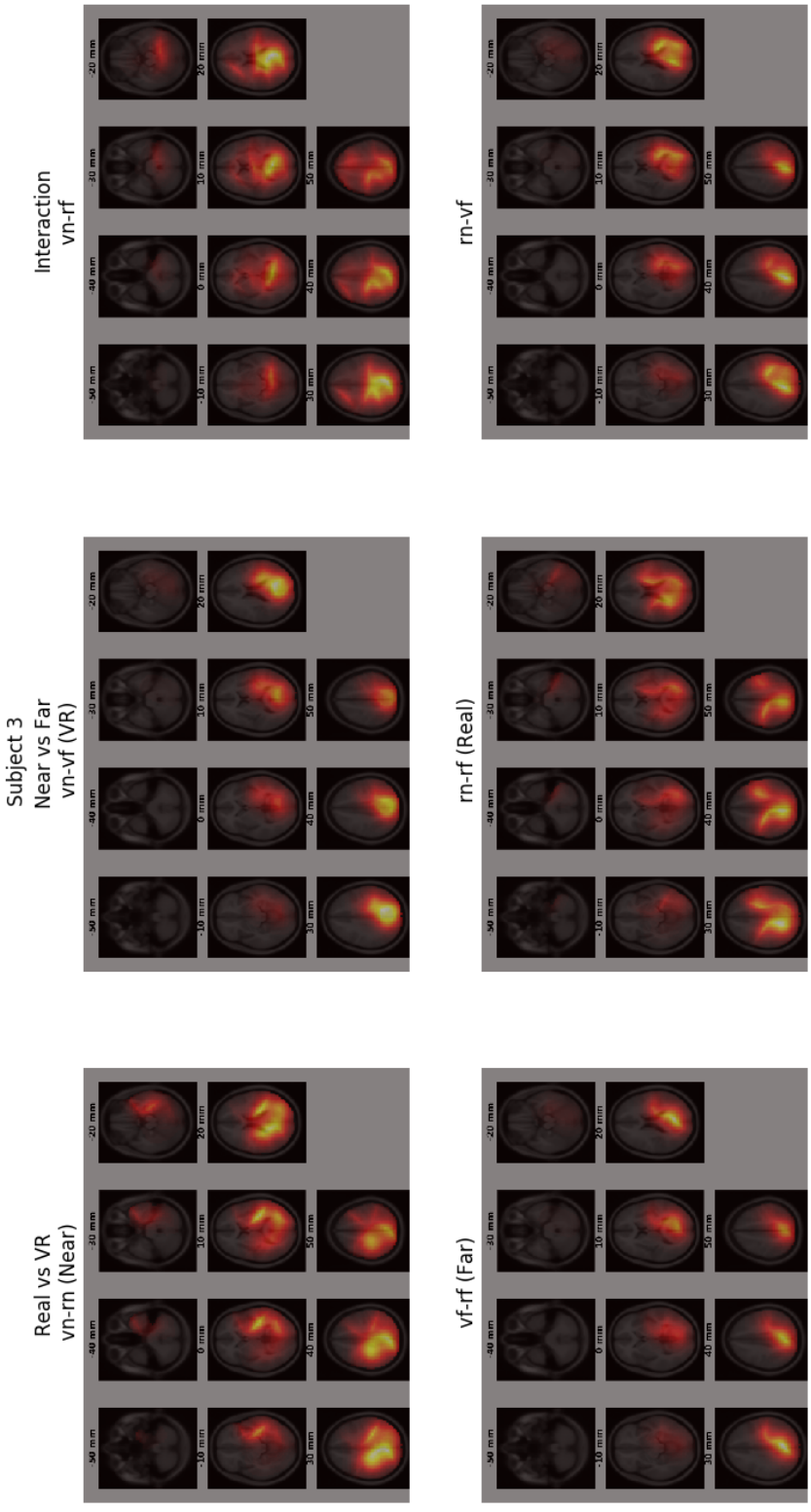

Figure A.3. Activity comparison between conditions for subject 3

VITA

Ethan Luckett was born in 1995. He graduated from high school in 2014 and received a B.S. in Computer Science from the University of Mississippi in 2018. In April 2020, he defended his thesis and will graduate with a M.S. in Computer Science from the University of Mississippi in May 2020.