Implementing Virtual Reality Metrics for Eye Contact Therapy

Caroline Braviak
Stanford University
Electrical Engineering
`cbraviak@stanford.edu`

Jonathan Lin
Stanford University
Electrical Engineering
`jolituba@stanford.edu`

Abstract

Eye contact is a crucial aspect of public speaking which greatly influences the effectiveness of a speaker. Virtual environments have been shown to be useful for simulating physical environments to train skills such as eye contact. In this project, we focus on metrics that can be integrated into a virtual environment. We include two different feedback mechanisms in our setup, a heatmap and a foveation game, to inform users of their eye contact while speaking. To combat noisy eye tracking data, we implement two filtering methods, a naive average and a weighted average, to smooth out the response. We present results for 10 users who rated the effectiveness of each filter, foveation level, and feedback mechanism. We find that there is a tradeoff between accuracy and precision, and that users' preferences for one or the other is split. Furthermore, we find that aggressive blurring can be useful for forcing a user's gaze into a specific location. Lastly, we find that users prefer immediate analytics to improve their gaze over practicing tools.

1. Introduction

Exposure therapy has been used by mental health practitioners as a treatment method to help patients confront their fears [18]. It consists of recreating a live environment or situation in which the fear presents itself so the patient can practice techniques to mitigate their anxiety in a controlled environment. However, situations can be difficult to recreate and rather embarrassing for the participant. Consequently, virtual reality (VR) provides a means to realize exposure therapy in an easy and efficient manner.

VR is a very beneficial technique for immersive therapies, a subcategory of exposure therapies. Research shows that VR has been successful in treating many phobias, including fear of heights, driving, flying, spiders, and small places [18]. The virtual environments induce symptoms of these fears, such as anxiety and stress, which were shown to be enhanced with the addition of audio and haptic cues [18]. Public speaking is a similarly stressful activity, so it is reasonable to suggest that immersive therapy can be used to aid in the treatment of the fear of public speaking.

2. Motivation

Public speaking is a critical skill in any professional context. However, according to the National Society Anxiety Center, the fear of public speaking affects about 73% of the population [13]. Public speaking places the presenter and their ideas in a vulnerable position in front of their peers. The resulting desire for acceptance stimulates an intense anxiety surrounding the potential for judgment and embarrassment [13].

This fear manifests itself in many ways, with one of the most common symptoms being poor eye contact. While avoiding eye contact with one's audience detracts from the speakers authority and credibility, establishing good eye contact can foster a connection with the audience [4]. It draws them into the speech or conversation and allows the presenter to control the flow and emphasis of their points.

Consequently, there is need for a practical method to allow presenters to practice speaking in an attempt to overcome such a paralyzing fear. VR provides a unique opportunity to simulate these anxiety-provoking situations in a virtual environment. It affords individuals an immersive experience in which they can learn techniques to mitigate the anxiety associated with public speaking and become effective presenters.

Previous studies have established that users experience symptoms of anxiety in a virtually rendered environment. For example, Slater et al. investigated the ability of a virtual environment to produce a similar response as a live audience [17]. Participants were split into two groups: one using a head-mounted display, and one using a monitor. All participants were exposed to three different types of audiences and then asked to rate their performance and symptoms, and fill out a general questionnaire. The study concluded that the participants had the same response to the virtual environments as to the corresponding real environment. These study results suggest that VR can be a useful tool in mitigating the anxiety and resulting lack of eye contact associated with public speaking [17].

Given that a virtual environment can produce the same physical and emotional responses as a real environment, our goal is to create components that can be used in a virtual public speaking environment. Specifically, we focus on creating metrics that can be used to inform a user of their public speaking habits. In what follows, we present an overview of previous virtual public speaking environments, the system developed for this project, and results for user interactions with our system.

3. Related Work

Several VR solutions have been proposed, each of which explores various metrics for analyzing a user's eye contact. The current practices range from almost no immersion to fully immersive experiences using head mounted displays and auditory cues.

3.1. Non-Immersive Solutions

Ummo is an application that allows the user to detail specific words that they wish to avoid. Ummo records the speech, listening for these words, and provides the user with feedback about where in the speech they faltered [12]. However, Ummo neglects the effect an audience can have on a speaker's ability to remain calm and confident. It also does not address eye contact, but rather focuses on the oral delivery of the speech.

3.2. Semi-Immersive Solutions

A second class of solutions, known as semi-immersive, combines the training aspects of the non-immersive solutions with a virtual environment. The Virtual Orator platform allows the user to practice their speech via a head mounted display [19]. The display renders various types of environments or can be configured to allow the audience to ask questions or record the speech to be reviewed later [19]. It can provide the user feedback on the timing of a presentation as well as their eye contact spread [19]. Speech Center VR is another semi-immersive platform that provides the user with two public speaking courses and various practice environments [8]. The user can upload their presentation and practice in various environments using the techniques learned from the educational sessions [8]. While these two platforms take advantage of the immersive enhancements of a head mounted display, the focus is directed more heavily toward practice rather than the various feedback mechanisms.

3.3. Fully-Immersive Solutions

The final class of solutions consists of the fully immersive approaches which attempt to simulate the live environment, including distractions and other noise, and provides a user with feedback metrics.

VirtualSpeech is an application that simulates various speaking environments, including conferences and peer networking, and provides the user with feedback on their eye contact [4]. The feedback comes in the form of a heatmap which represents the places or people in the room where the user focused. The app also provides training courses in which the user can learn how to distribute their eye contact amongst audiences of varying sizes. To increase user interactivity, the app scores the user out of 10 and tracks their progress over time [4].

Lee et al. describes two subtypes of fully-immersive VR setups for public speaking practice: one that is relatively static and based on computer generated models, and another based on movie clips [11]. However, the authors argue that both of these techniques have many disadvantages. The static nature of the models do not render a realistic environment, and the movie clips do not allow for the manipulation of an audiences emotions or reactions [11]. Lee et al. suggest a technique in which one can capture various snap shots of an audience and assemble them in real time to create a wide array of speaking environments [11]. The setup also allows for the user to interact with a therapist during simulation, allowing for real-time feedback and coaching [11].

Kang et al. further suggest that the ability to manipulate a virtual environment may prove effective in mitigating the fear of public speaking [9]. Kang et al. creates a model that can change the mood, attitude, and personality of an audience during a speech [9]. Like [11], Kang et al. suggest that this ability would enhance the virtual experience. Anderson et al. investigates these approaches and finds them effective in reducing the fear of public speaking in their participants [3]. The participants have 8 sessions, half of which consist of learning anxiety management techniques while the other half consist of fully immersive experiences using a head mounted display. During these sessions the participants spoke to a simulated audience. The therapist could manipulate the reactions of the audience as well as coach the participant throughout the speech [3]. In all of these approaches, having a coach detracts from the realistic nature of the problem. However, the ability of the therapist to manipulate the reactions of the audience seems to simulate a more realistic environment. It allows the participant to learn techniques for coping with various distractions and stressors, while still maintaining eye contact, that could arise during a presentation.

Vanni et al. notes that the common theme amongst the most effective solutions is the manipulation of the audience's emotions and reactions in order to realistically simulate the experience of presenting to an audience [20]. However there are few indicators that could be used to assess the outcomes of these studies. Some used basic biometrics, like heart rate, but most relied on self-reported results about outcomes of each study for the individuals [20].

4. Methods

Unlike previous work which sought to develop an integrated simulation system, our project focuses specifically on measuring and providing the user with feedback on eye contact while speaking. However, this task is complicated by the fact that a standard metric for evaluating eye contact is lacking. Here, we first outline our simulated environment setup, and then describe two eye contact metrics that we explore in this project.

4.1. Development Tools

To create our virtual environment, we develop in JavaScript, HTML and CSS. We also employ MATLAB for various compute-heavy components of our design. Additionally, we use The Eye Tribe Tracker to follow a user's gaze while interacting with the product. The tracker is positioned below the screen and directed at the user. The user must remain within the trackbox, which is the spatial volume monitored by the tracking software [14]. Because users have different eye characteristics, such as interpupillary distance, the hardware must be calibrated per person.

4.1.1 Calibration

The Eye Tribe Tracker's built-in calibration consists of the user following a moving circular target rendered on a blank screen. The target moves to various points on the screen and stalls for about two seconds at each location. We use 9 locations, the minimum number, for calibrating the tracker. Calibration takes approximately 20-30 seconds and the quality of the calibration is measured on a scale out of 5, as seen in Figure 1. For our experiments, the goal is to calibrate the tracker until the user receives a score of at least three out of five, which provides a visual angle accuracy of about 1° [14]. After a successful calibration, the user's gaze can be streamed as 2D (x,y) coordinates. However, The Eye Tribe Tracker has a mode which redirects the mouse to the user's gaze. Rather than stream the data, we leverage the cursor position utilities in JavaScript to obtain the eye tracking data via the cursor coordinates.

4.1.2 Interface

For the purposes of this experiment, we focus mainly on the various feedback and training mechanisms regarding the use of eye tracking rather than the environment itself. We conduct our simulation in a 2-dimensional environment and focus our efforts on the analytics and interpretation of the eye tracking data. The chosen scene is a 2-dimensional image of a theater audience. While the 2-dimensional environment may be less realistic than a fully-immersive, 3-dimensional environment, it is sufficient for evaluating the effects of providing users with eye contact feedback.

Figure 1. Calibration of The Eye Tribe Tracker [14]

Figure 2. Heatmap rendered over the original image. Red patches indicate areas of frequent gaze direction, green indicates infrequently viewed points, and areas with no heatmap overlay represent areas that were not viewed.

4.2. Feedback Mechanisms

Our implementation provides users with two feedback modes: the heatmap and foveated rendering.

4.2.1 Heatmap

The heatmap allows the user to visualize their gaze coverage and density in various parts of the room [4]. This allows users to visualize that they are covering the entire room for equal amounts of time.

In the heatmap mode, the user begins the simulation by pressing the space bar. They practice giving the speech and looking around the room, making eye contact with those in the audience. The screen also displays a train of blue dots that represents user's gaze direction and location. The underlying architecture for this trail is taken from Codepen, an open source code site, and is originally written by Ryan Boone [6]. Once the simulation has ended, a heatmap appears over the original image as seen in Figure 2.

To create the heatmap, we leveraged an existing JavaScript library `heatmap.js` [22]. The data passed to the library is an array of points consisting of (x,y) coordinates corresponding to the participant's gaze. The heatmap reveals the temporal information regarding the participant's performance. The size and color of a particular part of the heatmap is correlated to the amount of time spent in this area. It does not necessarily reflect continuous time, but rather the summation of all the time spent in that region throughout the duration of a trial. A larger and darker spot on the screen corresponds to a longer period of time in that area of the image.

In order to leverage the heatmap library, we create a $K \times 2$ matrix of gaze locations A, where $A_{i:}$ represents the (x, y) point that the user's gaze is located at during time sample i. Since our eye tracker uses tracking data to directly move the mouse, we leverage JavaScript's mousemove event listeners to sample for our heatmap. The tracker constantly streams data at a rate of 60 Hz [14], which is registered by the mousemove listener at the same rate. While the mousemove event listener only samples if the mouse actually moves, the noise of the tracking data means that there is no concern regarding skipping samples. This is because the data is highly unlikely to be constant over two consecutive samples.

4.2.2 Foveated Rendering

We also propose a novel metric based on circular occlusions implemented via foveated rendering. The fovea is the point on the retina at which the image is clearest [21]. Our approach leaves only a few unblurred regions in the scene, which forces the user's gaze towards these areas.

To provide an efficient implementation, we pre-render several versions of the audience scene in MATLAB. This scene consists of a mostly blurred image of the audience, with 6-7 unblurred circles of various radii placed uniformly throughout the scene as shown in Figure 3.

The motivation for pre-rendering is twofold. First, pre-rendering prevents program slowdowns due to computing foveated blurs in JavaScript. Second, having non-overlapping, unblurred circles necessitates hard-coding the locations rather than choosing locations randomly, so real-time image blurring is unnecessary. The foveation is implemented first by using a 2-dimensional convolution to channel-wise blur the entire image according to the following formula:

$$I_{\text{out}}^{RGB} = I_{\text{in}}^{RGB} ** \frac{1}{K^2} \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix} \tag{1}$$

where K is the width and length of the square ones matrix. Qualitatively, this corresponds to taking the average of the K^2 pixels on all 4 sides of the pixel for each channel. Moreover, we used various kernel sizes to allow the user the ability to choose the degree to which they would like to blur the audience. These various blurring strengths were motivated by creating different levels of "handicaps" for the user. This motivation followed the idea that blurrier regions outside the desired gaze location would make it easier to focus on the unblurred regions.

Figure 3. Foveated rendering with unblurred focus cues. Top: blur kernel size of 34×34. Bottom: blur kernel size of 70×70.

Next, the unblurred portion of the image within a circular radius r is replaced at its appropriate location. However, because convolution increases the output dimension in every direction by K, the replacement location is offset from the origin of the image by an additional factor of $K/2$.

To help participants train their eye contact, we develop a game. When a visual cue appears in one of the unblurred circles, the user must direct their eye contact within the radius of the focal point and remain in a particular circle for 2 seconds before the next cue renders. This ensures the participant is making brief, but meaningful eye contact. The user triggers the timer by gazing into the smaller circle, and must maintain their gaze inside the larger circle until they have passed. The larger circle is necessary because of the noise involved in eye tracking.

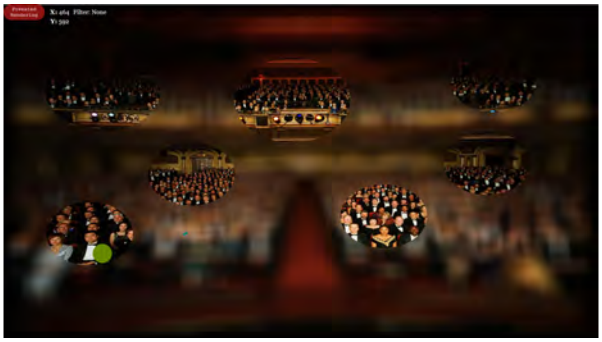

Figure 4. Screenshot from the foveated rendering game. The green circle indicates the unblurred target area to direct gaze.

4.3. Filtering

Finally, the user can practice the above two modes using three different settings for the data: no filter, a naive filter or a weighted average filter. As mentioned above, our calibration score goal is three out of five, although this is not always attainable, depending on the person and room conditions. According to The Eye Tribe Tracker documentation, this is a relatively poor calibration result [14]. Thus we compare three variations of the data to see if they improve the resulting tracking data.

For both the naive and weighted average filters, we employ a history length of 30 previous samples. This number of samples is chosen based on experiments with various history parameter values. We determine that 30 samples provides the lowest latency with the highest accuracy regarding the user's true gaze.

4.3.1 Naive Filter

The first filter we implement is a simple moving average. The goal is to smooth the data and provide a seamless interface for displaying the eye tracking data. The resulting (x,y) coordinates at each time step is the average of the current recorded position and previous 29 samples. Thus, updating the (x,y) position with a full history is done by removing the oldest element from the running history and then adding the current (x,y) position. This is demonstrated in the following equation with $h = 30$:

$$\text{next}_{x,y} = \text{prev}_{x,y} - \frac{\text{oldest}_{x,y}}{h} + \frac{\text{newest}_{x,y}}{h} \tag{2}$$

For the first 30 samples (where a full history is not yet measured), the (x,y) coordinates are updated using the following equations:

$$\text{curr sum}_{x,y} = \text{prev}_{x,y} * h_{\text{curr}} \tag{3}$$

$$\text{next}_{x,y} = \frac{\text{curr sum}_{x,y} + \text{newest}_{x,y}}{h_{\text{curr}} + 1} \tag{4}$$

4.3.2 Weighted Average Filter

The second filter we implement is a weighted average filter. We reason that more recent inputs would more accurately approximate the current (x,y) coordinate and thus should be weighted more heavily. The algorithm we implement uses a fixed weight α that is exponentiated to reduce the corresponding points weight as it became older relative to the current data point. Thus once a full history is achieved, there is a constant denominator. The coordinates with a full history were updated via the following equations with $h = 30$ and $\alpha = 0.9$:

$$\text{sum}_{x,y} = \text{prev}_{x,y} * d - \text{oldest}_{x,y} * \alpha^{(h-1)} \tag{5}$$

$$\text{next}_{x,y} = \text{sum}_{x,y} * \frac{\alpha}{d} + \frac{\text{newest}_{x,y}}{d} \tag{6}$$

Where d represents the constant denominator, which is calculated via the following formula:

$$d = \sum_{n=0}^{h-1} \alpha^n \tag{7}$$

Until a full history is obtained, the (x,y) coordinates is updated differently. The denominator of the weighted average is calculated for each sample. For example, if there is a history of length three, and we are seeking the fourth point, the denominator is $1 + \alpha + \alpha^2$, which represents the denominator at the previous time step. The coordinate is updated via the following equations:

$$\text{sum}_{x,y} = \text{prev}_{x,y} * \alpha * \text{prev weight} + \text{newest}_{x,y} \tag{8}$$

$$\text{next weight} = \text{prev weight} * \alpha + 1 \tag{9}$$

$$\text{next}_{x,y} = \frac{\text{sum}_{x,y}}{\text{next weight}} \tag{10}$$

5. Results

In order to assess the outcomes of our final product, we conduct a set of user trials, where 10 different users calibrate and experiment with the different filters (no filter, naive, weighted). The eye tracker presents a score out of 5 for calibration success, and we ask them to rate each filter out of 5 in terms of the response to their eye movements and true gaze trajectory. The users do not know which filter is which, only A, B, or C. The results are displayed in Table 1.

We also ask participants to try the foveated rendering game for two different kernel sizes: a kernel size of 34×34 and a kernel size of 70×70. They rate the two kernel sizes out of 5 in terms of how easiness of focusing on the unblurred regions. Their scores are given in Table 2.

Finally, we ask the participants to name which method seems more useful for training their eye contact and gaze

User	Calib.	No Filter	Naive Avg	Weight. Avg
1	2	5	3	4
2	2	3	3	2
3	4	3	4	4
4	5	5	3	4.5
5	3	3	3	4
6	5	5	4.5	4.8
7	5	3.5	4	3.75
8	4	3	5	4.8
9	1	3	3	3
10	5	3	4	4

Table 1. Scores for the various filters, on a scale of 1-5. Scores represent how well the user thought the response of the filter matched their eye movements.

User	Kernel Size = 34	Kernel Size = 70
1	4	5
2	5	4
3	4	5
4	4	5
5	4	5
6	5	5
7	5	5
8	5	5
9	3	4
10	3	5

Table 2. User ratings for the foveated rendering game with varying kernel sizes. Scores are on a scale of 1-5, with 5 being the best. Scores represent how easy it is to focus on the unblurred areas of the scene.

levels. Most participants responded that the heatmap was the most useful. One participant felt that the game was more useful, and one feels that they seemed equally useful. The participants who preferred the heatmap said that the heatmap's immediate feedback and tangible analysis was more useful for improving gaze, since it revealed precisely the areas the user neglected and the areas the user tended to focus their gaze. The user who prefered the foveated game felt that the practice given by the game is more useful because it informs users exactly where they should be looking.

Our filter trial results show that on average, participants found the weighted average filter to more accurately display the true trajectory of their gaze when compared to the naive average filter. However, for the filterless mode, participants fell into two categories. The first group found no filter to be more responsive. They note that the faster response time, compared to the other two filters, and the accuracy of the trajectory outweighed the downsides of noisy tracking. By contrast, the second group cited the noise as a distraction and preferred the smoothed trajectory, although it had a higher latency. These two groups demonstrate a trade-off in user preference for accuracy versus precision of eye tracking, since the first group prefers accurate but noisy tracking, while the second group prefers noiseless tracking, even if latency causes it to be slightly temporally inaccurate.

The foveated rendering trials show that a larger blur kernel was consistently rated higher, meaning the participants perceive it as easier to focus on the unblurred regions when the background image is more blurred. Although some participants note that both kernels were very similar in perception, the data above suggests that as the blur kernel increases, the larger blur tends to force the participant's gaze to particular regions.

6. Discussion

Overall, we see several strong trends across all trials. First, there is the stark distinction between providing the user with feedback on eye contact versus practicing public speaking. The participants appear to desire immediate analytical feedback on their performance. They feel that the feedback is easily interpretable and significantly more informative about aspects of their eye contact that could be improved, both spatially and temporally. This suggests that the heatmap is an effective metric for evaluating a participant's eye contact. Second, the latency vs. accuracy tradeoff divides user's preference for the weighted average filter or no filter at all. Some said that the noise of the data was distracting while executing the tasks, which detracted from the benefits of the analytical feedback. Therefore, it is necessary to find a filtering option that neither compromises on latency nor accuracy to effectively and efficiently train a participant's eye contact. Finally, our data for foveated rendering suggests that while all blur kernels force the participants gaze to the unblurred regions of the screen, a larger kernel size appears to do so more aggressively.

We also observe several limitation to our approach. The lack of a head-mounted display means our environment is less realistic and less immersive. The stationary monitor and eye camera also cause the user's yaw movement to be restricted, which is an unrealistic constraint for public speaking [16]. The audience scene is static with respect to movement and sound, making for an unrealistic environment. Other limitations are primarily due to the hardware that was used. The Eye Tribe tracker's sampling rate of 60 Hz [14] is sufficient for tracking macro-movements such as in the heatmap, but causes issues for smaller micromovements in the foveated rendering game. Small jitters have little effect on heat map due to its temporal nature, but can cause targets to incorrectly be triggered in the foveated rendering game. Finally, an ideal implementation of the foveated rendering game would employ real-time rendering of focus cues, this would be too computationally intensive for our JavaScript implementation. Consequently, it was necessary to pre-render the images in MATLAB, which detracted from the reality of the situation.

7. Conclusion

In this project, we have developed a tool to help users evaluate eye contact while speaking and provide them with feedback. We found that on average, most users prefer the immediate feedback provided by the heatmap as opposed to the training received from the foveal rendering game. Moreover, we find that there is a trade-off between latency and accuracy with regard to the filter setting, suggesting that some user's prefer a faster response time, while others prefer a more accurate trajectory of their true gaze. Finally we show that a larger blue kernel in the foveal rendering game is more effective at forcing a user's gaze.

Future work should focus on integrating these feedback mechanisms into a head-mounted display setup. The head-mounted display would be able to simulate movement, noise and other interactions associated with public speaking, while providing the benefits of heatmap feedback or foveal rendering training that we have observed here. A future iteration should also support real-time foveated rendering to more accurately mimic the true blur of changing one's gaze.

8. Acknowledgements

Thanks to the EE 267 teaching team: Prof. Gordon Wetzstein, Robert Konrad, Hayato Ikoma, and Marcus Pan for their help and support throughout this project. Also thanks to the WIM writing tutors Mary McDevitt and Tim Anderson for their help in proofreading this paper. We would also like to acknowledge the 10 members of Caroline's residence who served as test subjects for our experiments and Kevin Yang for his consultation on JavaScript details. Finally, we would like to acknowledge [10][6][22][1] for code that we used in various forms as part of our implementation, [5][7][2] for providing our sound clips, and [15] for providing our background image.

9. Demonstration Videos

Foveated: https://youtu.be/vhymDR9mnwo
Heatmap: https://youtu.be/OGExTjUT9DU

References

[1] Game sound. `https://www.w3schools.com/graphics/game_sound.asp`.

[2] 14FPanskaBubik_Lukas. Ping pong hit.wav. `https://freesound.org/people/14FPanskaBubik_Lukas/sounds/418556/`.

[3] P. Anderson, E. Zimand, L. Hodges, and B. Rothbaum. Cognitive behavioral therapy for public speaking anxiety using virtual reality for exposure. *Depression and Anxiety*, 22:156–158, 2005.

[4] D. Barnard. Practice in virtual reality to improve eye contact. `https://virtualspeech.com/blog/practice-virtual-reality-improve-eye-contact`, 2017. accessed April 18, 2018.

[5] B_Lamerichs. Regular game sound effects 1. `https://freesound.org/people/B_Lamerichs/sounds/220000/`.

[6] R. Boone. `https://codepen.io/falldowngoboone/pen/PwzPYv`, 2014.

[7] edsward. Ping.wav. `https://freesound.org/people/edsward/sounds/341871/`.

[8] P. Graham. Conquer those public speaking fears with speech center vr. `https://www.vrfocus.com/2016/08/conquer-those-public-speaking-fears-with-speech-cent` 2016. accessed May 11, 2018.

[9] N. Kang, W.-P. Brinkman, M. B. van Riemsdijk, and M. A. Neerincx. An expressive virtual audience with flexible behavioral styles. *IEEE Transactions on Affective Computing*, 4(4):326–340, 2013.

[10] R. Konrad, H. Ikoma, M. Pan, and G. Wetzstein. EE 267 Assignment 3, 2018.

[11] J. Lee, J. Ku, D. Jang, D. Kim, Y. Choi, I. Kim, and S. Kim. Virtual reality system for treatment of the fear of public speaking using image-based rendering moving pictures. *Cyberpsychology and Behavior*, 5(3):191–195, 2002.

[12] N. McAlone. These 5 harvard business school students built an app to help you sound more confident when you speak. `http://www.businessinsider.com/ummo-app-tracks-your-speech-2016-5`, 2016. accessed May 10, 2018.

[13] J. Montopoli. Public speaking and fear or brain freezes. `http://nationalsocialanxietycenter.com/2017/02/20/public-speaking-and-fear-of-brain-freezes`, 2017. accessed April 18, 2018.

[14] A. B. Pedersen, H. Skovsgaard, J. S. Agustin, and M. Kelager. Basics. `http://theeyetribe.com/dev.theeyetribe.com/dev.theeyetribe.com/general/index.html`, 2014.

[15] M. Powell. Theater.png. `https://www.telegraph.co.uk/culture/theatre/11283846/theatre-audiences-too-white-janet-suzman.html`.

[16] V. Sitzmann, A. Serrano, A. Pavel, M. Agrawala, D. Gutierrez, B. Masia, and G. Wetzstein. How do people explore virtual environments? *IEEE Transactions on Visualization and Computer Graphics*, 2017.

[17] M. Slater, D.-P. Pertaub, and A. Steed. Public speaking in virtual reality: facing an audience of avatars. *IEEE Computer Graphics and Applications*, 19(2):6–9, 1999.

[18] D. Strickland, L. Hodges, M. North, and S. Weghorst. Overcoming phobias by virtual reality exposure. *Comm. ACM*, 40(8):34–39, 1997.

[19] V. H. Technologies. What is virtual orator. `https://publicspeaking.tech/`, 2015. accessed May 10, 2018.

[20] F. Vanni, C. Conversano, A. D. Debbio, P. Landi, M. Carlini, C. Fanciullacci, M. Berhamasco, A. D. Fiorno, and L. Dell'osso. A survey on virtual environment applications to fear of public speaking. *European Review for Medical and Pharmacological Sciences*, 7:1561–1568, 2013.

[21] G. Wetzstein. EE267 Lecture 5, 2018.

[22] P. Wied. heatmap.js. `https://www.patrick-wied.at/static/heatmapjs/`.