Depth and layout encoded simulated prosthetic vision for obstacle avoidance and object recognition

Alex Rasla
UC Santa Barbara
alexrasla@ucsb.edu

Abstract

Retinal prostheses have the potential to treat degenerative eye diseases that often result in low vision or complete blindness. These devices rely on an external camera to capture the visual scene, which is then translated frame-by-frame into an electrical stimulation pattern that is sent to the implant in the eye. To highlight more meaningful information in the scene, recent studies have tested the effectiveness of deep-learning based computer vision techniques, such as room layout estimation and relative depth extraction. However, nobody has attempted to combine the two, either by presenting them together (*DepthAndLayout*) or by giving the user the ability to flexibly switch between them (*DepthOrLayout*). Here, we used simulated prosthetic vision in a virtual reality environment to test the effectiveness of highlighting the room layout and segmenting nearby objects to avoid obstacles and recognize objects. We found that most users preferred *DepthOrLayout* over the other options, even though they did not necessarily perform better with it. This study is an important first step towards a retinal prosthesis that uses computer vision to improve a user's scene understanding.

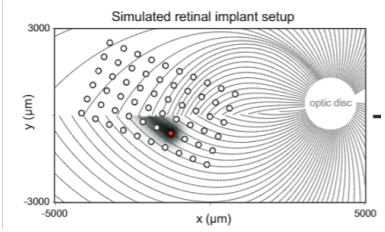
1 Introduction

Retinal prostheses are a very innovative and exciting technology that has the potential to treat degenerative diseases such as retinitis pigmentosa or age-related macular degeneration. These diseases continuously become worse over time and often result in low-vision or complete blindness. Many of these devices work by capturing information about a user's surroundings using an external camera attached to a pair of glasses or headset [10], [3], [5]. This information is then mapped onto the device's electrode array by translating the captured information into an electrical stimulation pattern. Each electrode's amplitude on the electrode array is proportional to the grayscale value of the captured image, creating a holistic reconstructed image. The intention of many of these devices is to most accurately replicate and restore natural vision. While this is an inevitable first step in exploring what is possible with these implants, studies have shown that the simulated prosthetic vision (SPV) generated from this total scene reconstruction method is hard to interpret. Previous works have explored ways of using deep-learning computer vision based approaches such as object segmentation, depth perception, saliency, and structural and semantic segmentation to reconstruct the scenes in unique ways [6], [7], [9], [8]. However, nobody has attempted to reconstruct and provide information about scenes by combining multiple deep-learning computer vision based techniques. By providing users with the flexibility in how they visualize and interpret their surroundings, we believe that we can provide an effective and alternative approach to simply trying to restore natural vision using retinal implants.

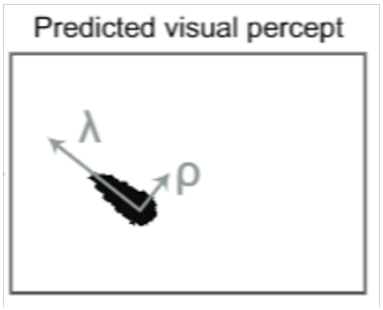
2 Background

2.1 Simulated Prosthetic Models

When retinal prosthetic devices such as Argus I and Argus II started to be tested on patients, researchers quickly noticed that the phosphenes generated by stimulating individual electrodes on the retinal implant were inconsistent with what they expected. Previously, researchers believed that the electrodes on the implant would follow a linear scoreboard model. In this model, the patient would be expected to see a phosphene point at the stimulated electrode proportional to the stimulated amplitude. However, user studies on these patients showed that the phosphenes generated by a given stimulated electrode were elongated and different for each electrode on the implant, but also reproducible [1]. Because of this inconsistency when compar-



(a)



(b)

Figure 1: This figure depicts the axon map model. 1a shows a retinal implant on the axon fibers of a retina. When one electode is activated, the black streak is produced because of the axon fibers. 1b shows the values of λ and ρ in relation to the streak

ing actual patients to the linear scoreboard model, more complex models have been proposed with the goal of being able to accurately simulate these electrode stimulations [4]. Each of the different ways to model these phosphene perceptions produces a unique SPV.

2.2 Axon Map Model

One such model used to simulate phosphenes is the axon map model [1]. This model takes into account the ganglion axon fibers on the retina, and suggests that the phosphene distortion produced in actual patients is a result of the density of these fibers. The model suggests that an electrode can be placed on many axon fibers, and thus when stimulated, many ganglion cells across these axon fibers get stimulated as well. As a result, the phosphene from a stimulated electrode travel a certain distance along the axon (λ) and a certain distance perpendicular to the axon fiber (ρ). The authors argue that this ultimately causes the undesired effect of patients seeing phosphenes as streaks or blobs rather than points.

3 Methods

In order to test the effectiveness of using multiple deep learning based approaches for visualizing scenes, we designed an experiment in virtual reality using the HTC Vive where we tasked participants with navigating a set of rooms that were filled with obstacles to avoid and objects to select using a set of SPV modes. Each room had two parts – the first part being avoiding the obstacles within the room, and the second part being choosing a medium sized cube off of a table.

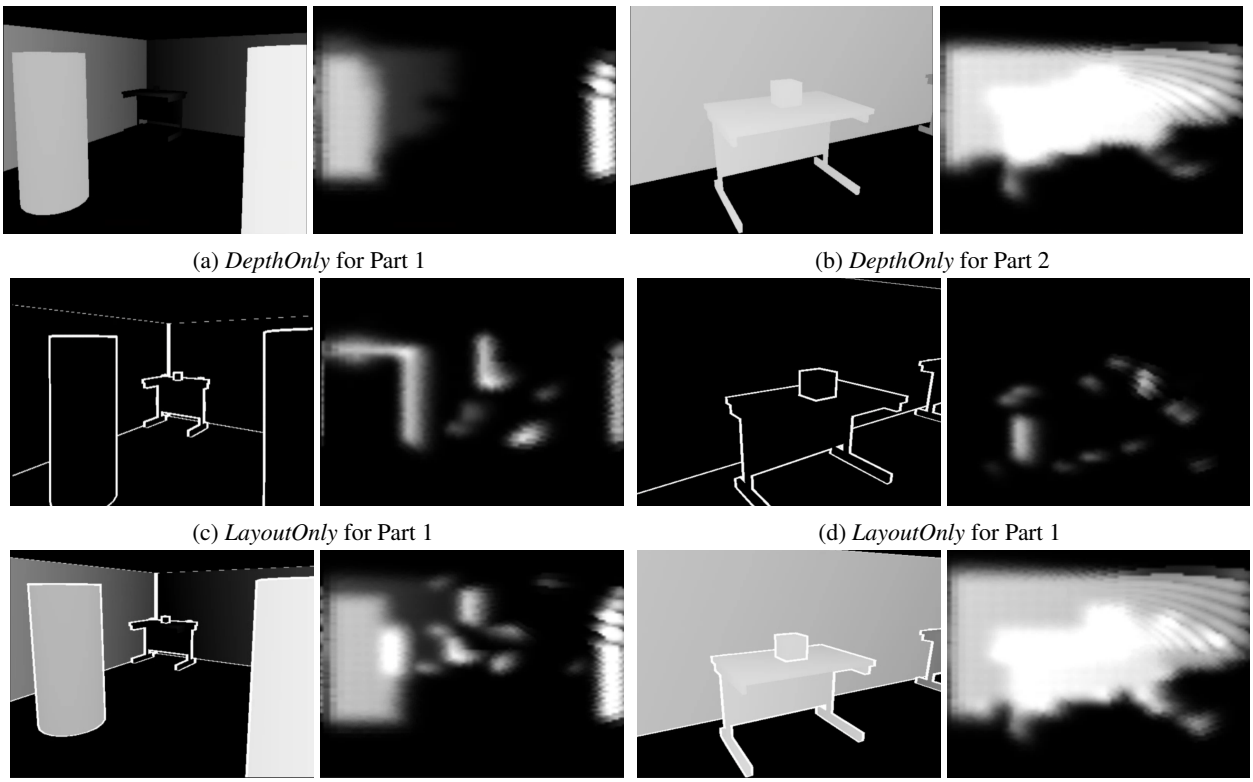
3.1 Modes

Each SPV mode gave the user a unique way of seeing the objects within the room using the axon map model [1]. To create the set of modes tested in this experiment, we used the pulse2percept [2] library to create a simulated retinal implant that used the axon map model [1] with a 20 x 15 electrode array where $\rho = 300$ and $\lambda = 550$.

A visual representation the different SPV modes with the both normal vision and the SPV vision can be seen in 2. In the *DepthAndLayout* mode, the user was able to see both the depth and structural layout of the walls and the objects within the room together. In the *DepthOnly* mode, the user was able to see only the depth of the walls and every object within the room. Similarly, in the *LayoutOnly* mode, the user was able to see only the structural layout of the walls and every object within the room. Finally, in the *DepthOrLayout* mode, the user was given the flexibility of being able to switch between the *DepthOnly* mode and *LayoutOnly* mode.

3.2 Rooms

The experiment consisted of 6 rooms, shown in Figure 3, that each user tested with the 4 different SPV modes mentioned in the previous section. In the first part, there were between 3 and 7 different obstacles to avoid (depending on the room) that were each placed in a unique position in a particular room. The user knew they collided with an object if they heard a “thud” sound. In the second part, there was a collection of three objects (shown in Figure 4) – a sphere, a cylinder, and a medium cube; or a small cube, a medium cube, and a large cube – to select on either one or three different tables. If there were three tables, the objects were evenly spread out across these tables — one object per table. In each room, the users were tasked with selecting the one and only medium cube. This second part was always on the opposite side of the room from where the participants started, making them have to avoid the collision obstacles. The participants knew they had crossed an imaginary plane into the

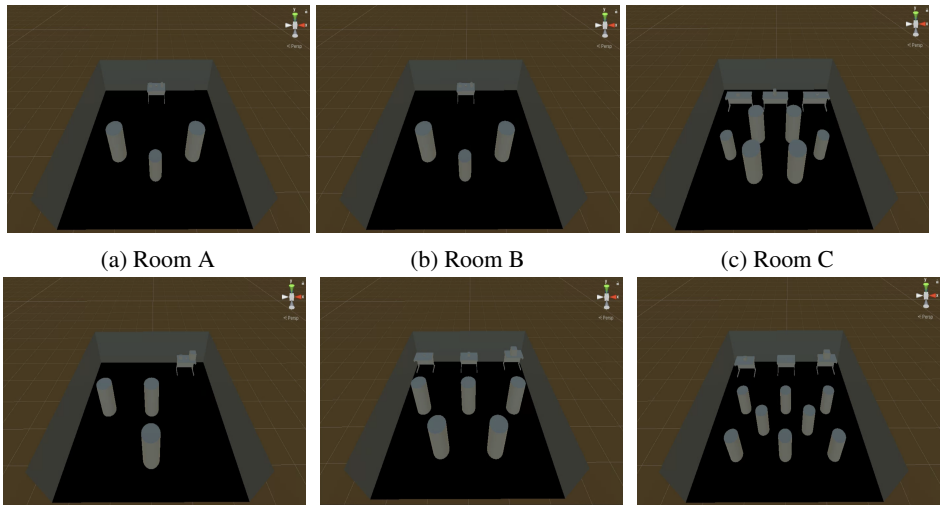


(a) *DepthOnly* for Part 1 (b) *DepthOnly* for Part 2

(c) *LayoutOnly* for Part 1 (d) *LayoutOnly* for Part 1

(e) *DepthAndLayout* for Part 1 (f) *DepthAndLayout* for Part 2

Figure 2: The different SPV modes



(a) Room A (b) Room B (c) Room C

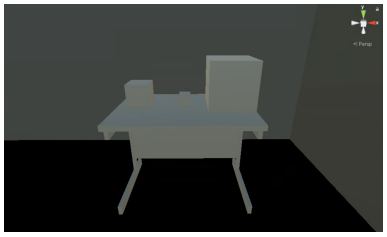
(d) Room D (e) Room E (f) Room F

Figure 3: The perspective views of the room layouts

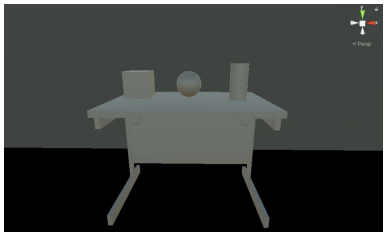
second part when they heard a chime through the Vive headset.

3.3 Procedure

Before the start of each new SPV mode, the user was allowed to explore a "tutorial" room that contained one collision obstacle to avoid, and a medium sized cube on a table. They were able to freely navigate the room for as long as they wanted to get familiar with the SPV mode and how the different features — walls, collision objects, selection object — of the room looked. When they felt ready to start the experiment, they selected the medium sized cube off the table, and were given 30 seconds to relocate back to the starting position. At this point the experiment started on a random



(a)



(b)

Figure 4: In this figure, we can see the different objects to select. 4a shows the collection of objects with different cube sizes, and 4b shows the collection of different objects. In both collections, the medium-sized cube is always the correct object to choose. In cases where the rooms have 3 tables, one object is on one of the three tables.

room. Similarly to the tutorial room, during the actual experiment, once an object was selected off a table with the Vive controller in the second part – or the participants had reached the time limit of 4 minutes per room – the user was similarly relocated back to the starting position, and started in the next room.

3.4 Data

While the user was navigating each room, their position was recorded every 0.5 seconds. Further, if they collided with a collision obstacle when tasked with avoiding objects, the time of collision and specific obstacle were recorded. There would then be a timeout period of 3 seconds before other collisions would be recorded so multiple collisions of the same object would only be counted once. Next, the time the user crossed the plane into the second part was also recorded. Once the user crossed this plane, no further collisions were recorded. Once the user selected an object off a table in the second part, the type of object and whether or not it was the correct object was recorded. If time ran out before they were able to select an object, we recorded "None" as the object type and "Not Correct". Finally, in order to prevent users learning the order of the rooms through the different SPV modes, the order of the rooms were randomized for a given SPV mode. Further, in order to standardize the data across each mode, the order of the SPV modes were also randomized.

3.5 Data Analysis

To determine whether performance was significantly different across modes and rooms, we ran a regression analysis for the performance on each task and for the times taken for each task. We used the Ordinary Least Squares model (OLS) to isolate each variable and determine if modes, rooms, VR experiences, gender, or age had an effect on the results of the variable in question. An example of a model that analyzes the effect of rooms, modes, and VR experience on the number of collisions can be seen below.

$$Collisions \sim C(Mode) + C(Room) + C(gender) + C(VRExperience)$$

3.6 Participants

We recruited 24 participants from the SONA program at UCSB. The participants ranged from ages $18-20$ in age, 7 identifying as males and 17 identifying as females. Of these participants, 4 had used virtual reality (VR) 0 times, 16 had used VR between $1-5$ times before, 2 had used VR between $10-20$ times, and 2 had used VR $20+$ times. Potential participants were excluded if they did not have normal or corrected-to-normal vision, or if they were prone to cybersickness. The study was approved by the UCSB Institutional Review Board.

4 Results

The main results from this study can be summarized through two different perspectives – the first is the significant differences within rooms, and the second is the significant differences across all rooms.

4.1 Results Within Rooms

For the results within each individual room, there are a few notable differences when comparing the modes each other.

When analyzing the average number of collisions in RoomC, RoomD, and RoomF, there are significant differences between a few modes. In Figure 5, we can see that on average in RoomC, users collided with objects ~ 0.6 times in the *LayoutOnly* mode and only ~ 0.18 and ~ 0.0 times using the *DepthOrLayout* and *DepthOnly* modes. We can see a similar relationship between the two

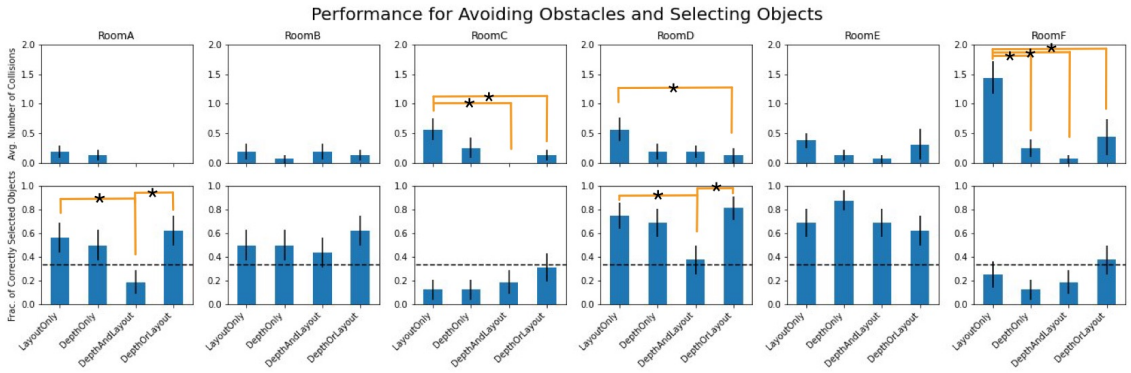


Figure 5: The number of collisions for each room and each mode, and the fraction of correct objects selected for each room and each mode across all subjects. The vertical lines coming from the top of the bar graphs represent the error of the mean values for each mode. Further, the yellow lines and the asterisks between different modes in a graph represent statistically significant ($p < 0.05$) differences between the bar plot values. In the fraction of correct objects graphs, the dashed line at 33% represents the chance levels of selecting the correct object since there were 3 objects per room.

modes in RoomD, where there is a statistical significance between *LayoutOnly* at ~ 0.6 and *DepthOrLayout* at ~ 0.18. Further, in RoomF, there was a significant difference between *LayoutOnly* and every other mode. In this room, participants collided with objects ~ 1.4 times using the *LayoutOnly* mode, and ~ 0.25, ~ 0.1, and ~ 0.5 times using the *DepthOnly*, *DepthAndLayout*, *DepthOrLayout* modes respectively. In general, we can see this same relationship — where *DepthOrLayout* performs better than *LayoutOnly* – across across all rooms for the average number of collisions, but the differences between the collision values for the modes are not statistically significant.

When analyzing the difference in the number of correctly selected objects for a given mode, the only statistically significant difference in values is in RoomA and RoomD. As can be seen in Figure 5, in both these rooms, participants were significantly better at using the *DepthOrLayout* mode and the *LayoutOnly* mode to select the correct object compared to the *DepthAndLayout* mode. In RoomA, participants were able to select the correct object $\sim 62\%$ of the time using *DepthOrLayout* and $\sim 52\%$ using *LayoutOnly*, but only $\sim 18\%$ of the time using *DepthAndLayout*. Similarly, in RoomD, participants were able to select the correct object $\sim 81\%$ of the time using *DepthOrLayout* and $\sim 75\%$ using *LayoutOnly*, but only $\sim 37\%$ of the time using *DepthAndLayout*. For this task, chance lies at $\sim 33\%$ because there are 3 objects present in every room. Similar to the comparisons of modes in the average number of collisions, this trend between *DepthOrLayout* out performing *DepthAndLayout* and *LayoutOnly* is generally present across all rooms (aside from RoomE), but the difference in values between modes is not statistically significant.

For the amount of time taken to avoid obstacles and select the correct object, there were also significant differences for some rooms. In RoomE, users took longer to avoid obstacles using the *DepthOrLayout* mode compared to the *DepthAndLayout* and *DepthOnly* modes. On average in RoomE, it took participants ~ 27 seconds for *DepthOrLayout*, ~ 17 seconds for *DepthAndLayout*, ~ 16 seconds for *DepthOnly*. In RoomF, there were significant differences between *LayoutOnly* and *DepthOnly*, *LayoutOnly* and *DepthAndLayout*, *DepthOnly* and *DepthOrLayout*. Participants took ~ 35 seconds using *LayoutOnly*, ~ 18 seconds using *DepthOnly*, ~ 21 seconds using *DepthAndLayout*, ~ 29 seconds using *DepthOrLayout*.

For the time taken to select the correct object, there is only one statistically significant value in RoomC between *LayoutOnly* and *DepthOnly*. The values for these modes are ~ 30 seconds and ~ 10 seconds for *LayoutOnly* and *DepthOnly* respectively.

4.2 Results Across All Rooms

Alongside significance within rooms, there are also notable results and consistencies across all rooms.

When users were tasked with avoiding obstacles in the first part of the experiment, participants collided with significantly more objects when using the *LayoutOnly* mode compared to the all other

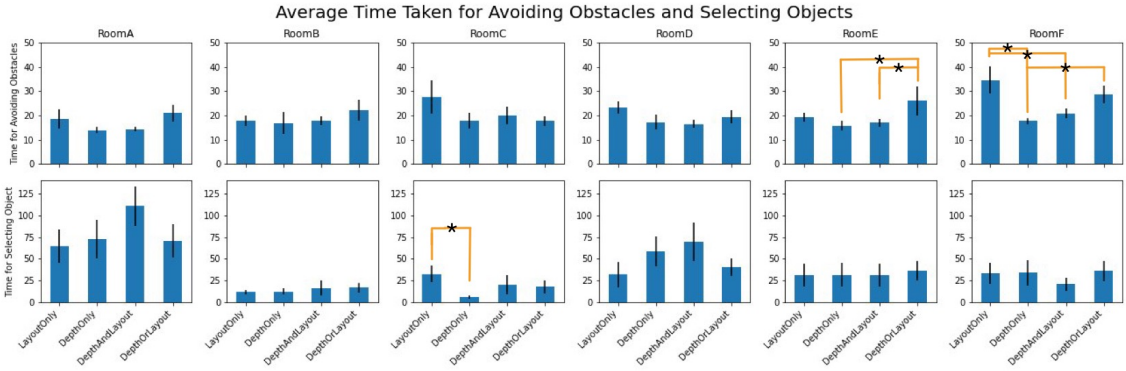


Figure 6: The times taken for avoiding obstacles and selecting the correct objects for each room and each given mode. Again, in this figure, the vertical lines at the top of the bar graphs represent the error of the mean values for each mode, and the yellow lines and the asterisks between different modes in a graph represent statistically significant ($p < 0.05$) differences between the bar plot values.

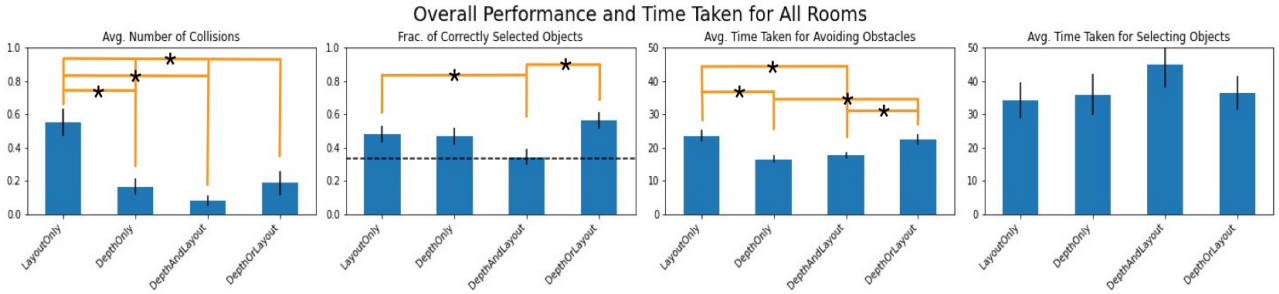


Figure 7: The overall performance of each mode for each task, averaged across all rooms for each participant. Similar to the figures above, the vertical lines at the top of the bar graphs represent the error of the mean values for each mode, and the yellow lines and the asterisks between different modes in a graph represent statistically significant ($p < 0.05$) differences between the bar plot values.

modes mode. As can be seen in Figure 7, on average, users collided ~ 0.53 times in the *LayoutOnly* mode and only ~ 0.18, ~ 0.08, and ~ 0.2 times for the *DepthOnly*, *DepthAndLayout*, and *DepthOrLayout* modes respectively. In Figure 10, we can see the different paths taken for each participant in each room.

Further, when compared to the *DepthAndLayout* mode, the *DepthOrLayout* and *LayoutOnly* modes performed better across the set of all the rooms when users were tasked with recognizing and selecting the correct object on the tables. This again can be seen in Figure 7 where there is a significant difference between the two modes. On average, users selected the correct object $\sim 57\%$ of the time using *DepthOrLayout* and $\sim 48\%$ of the time using *LayoutOnly*, but only $\sim 33\%$ of the time using *DepthAndLayout*. Notably, this is the percentage for selecting the correct object at chance.

When analyzing the results for the average time taken to avoid obstacles in the first part of the experiment and cross the plane into the second part of the experiment, there is at least one significant values between one given mode and another. When looking at the results for *LayoutOnly*, we can see that there is a significant difference in the time table when comparing the mode to *DepthOnly* and *DepthAndLayout*. Further, *DepthOnly* also has a significant difference when compared to the *DepthOrLayout* mode. Finally, there is also a significant difference between the *DepthOrLayout* and *DepthAndLayout* modes. The values for each of these modes was $\sim$ 25 seconds, $\sim$ 16 seconds, $\sim$ 18 seconds, and $\sim$ 24 seconds to cross the plane into the second part of the experiment for *LayoutOnly*, *DepthOnly DepthAndLayout*, and *DepthOrLayout* respectively. There were no notable differences in time taken for selecting the correct object.

Beyond the difference in modes, the difference in users demographics and their experiences with virtual reality was another notable factor that influenced performance. Specifically, users that had no prior VR experiences performed worse in selecting the correct object and took longer to avoid obsta-

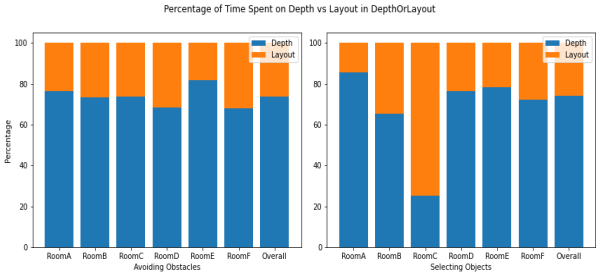


Figure 8: The percentage plots of the average time spent using Depth versus using Layout for the DepthOrLayout mode. The left bar plot is the percentage of times spent on each mode when avoiding obstacles and the right bar plot is the percentage of time when selecting objects.

cles than users who had used VR one or more times. Further, users' ages and genders also had an impact on their overall performance. Specifically, 18 year olds had fewer number of collisions across all rooms than 19 and 20 year olds, and males outperformed females on all tasks – time and accuracy for avoiding objects and time and accuracy in selecting the correct objects.

Finally, from Figure 8, we see that when having the choice in the *DepthOrLayout* mode to use either the *DepthOnly* mode or the *LayoutOnly* mode, participants overwhelmingly preferred to see depth for each room. The percentages of time spent for each room lied at around $\sim 80\%$ for depth and at around $\sim 20\%$ for layout. The closest percentage of time spent between the two modes was in RoomC, but even in this room, depth was used $\sim 60\%$ of the time and layout was used $\sim 40\%$ of the time.

4.3 Qualitative Results

Alongside the quantitative results mentioned above, we also asked the participants which modes they preferred through each of the individual parts of the experiments, and the mode they preferred to complete the experiment collectively. The answers to these questions can be seen in Figure 8.

In this figure, we can see that the most preferred modes for avoiding obstacles were the *DepthOrLayout* and the *DepthOnly* mode, with 9 participants preferring each of these modes. For the task of recognizing and selecting objects, 7 participants — the majority of participants — preferred the *LayoutOnly* mode. Finally, when asked what mode they found easiest to navigate both parts collectively, and the majority (9 participants) preferred *DepthOrLayout*.

5 Discussion

From the results above, we can make various conclusions about the different modes that were tested and the effectiveness of using a combination of different modes. When analyzing the average number of collisions in Figure 7, it is clear that the *LayoutOnly* mode led to significantly more collisions than modes that had any sort of depth-encoded information. This is expected because collider objects are clearer and easier to see when they are segmented as opposed to just outlined. Next, when analyzing the fraction of correctly selected objects, it is clear that the over-stimulation of electrodes in the *DepthAndLayout* led to participants having a harder time selecting the correct objects. This is also expected because the combination of the depth and layout encoded vision causes more axon streaks and blur than any of the other modes. Thus, the participants could not make an accurate judgment of which object was the correct one because the the object, table, and back wall became blended into an uninterpretable mess and the shapes of the objects could not be determined well.

From Figure 8, we can see that when given the choice between depth and layout, participants chose to see the depth encoded information most of the time. This is because generally, depth provided more information about the scene than layout did. Further, depth appears to be more similar to natural vision that layout since we can see whole objects in the real world, not just the outlines of the objects.

Finally, from the qualitative analysis in Figure 9, we can see that participants preferred the *DepthOrLayout* overall compared to the other modes. Though the quantitative results may not reflect that there was a significant difference in performance across all rooms when comparing the *DepthOrLayout* mode to the others, participants still preferred the flexibility of being able to switch between the depth and layout modes.

5.1 Limitations

Through the analysis of the results of this study, we noticed a variety of limitations. First, we noticed that most of the differences between modes were not statistically significant. This could be for a variety of reasons — including the fact that perhaps there was truly no difference between some modes — but some modes did seem like they had differences between them, just not enough to be statistically different. When this was the case, it is

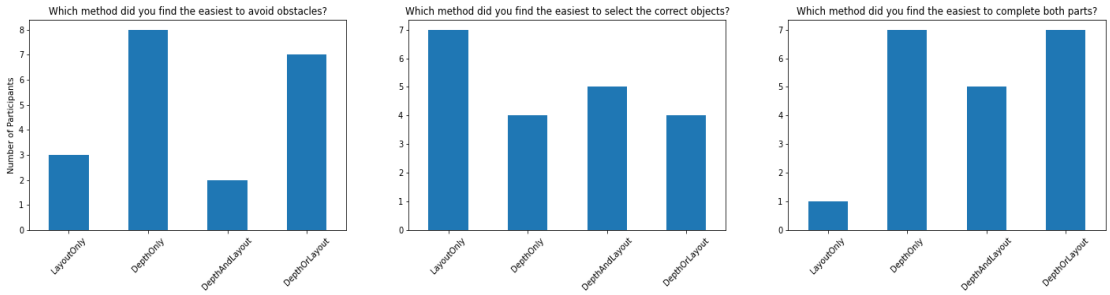


Figure 9: A histogram of each subject's preferred mode for each part of the experiment and overall

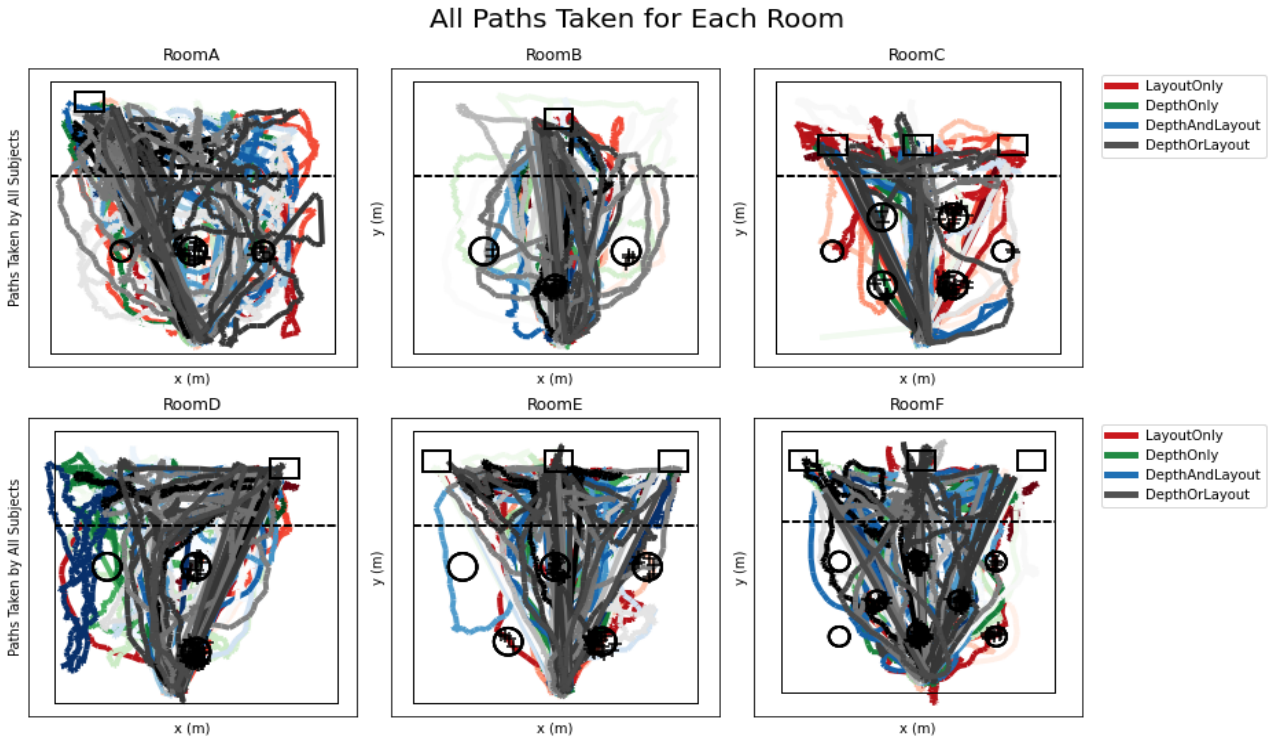


Figure 10: Each path taken for each room and each mode

likely because there were not enough participants tested and not enough data collected.

Another limitation we noticed was that there were too many variables to account for. At first glance, the rooms all have 3 objects to choose from that are each placed on the same table, or on 3 different tables. However, the these tables are placed in different locations in the second part of the room, the objects were either different sized cubes or a cube, sphere, and a cylinder, and the number of collider objects varied for each room. Beyond the layouts of the rooms, the differences in VR experience, gender, and age, also significantly effected performance. Ultimately, this caused our results to be noisy overall, and led to some inconclusive and undesired results. In future iterations of the work, it would be best to keep more of the variables mentioned above constant, and focus on the differences in modes.

Other reasons that likely contributed to the noisy results was the motivation of the users. Typically, as the experiment progressed into the later stages, participants often became more fatigued and simply wanted to finish the study (especially users who took a long time). This would often lead to participants becoming reckless: walking through obstacles, trying to finish rooms really fast fast, randomly selecting objects off the table once they found one. This too definitely created some undesired noise and effect on the data.

References

[1] Michael Beyeler et al. “A model of ganglion axon pathways accounts for percepts elicited by retinal implants”. In: *Scientific Reports* 9 (June 2019), p. 9199. DOI: 10.1038/s41598-019-45416-4.

[2] Michael Beyeler et al. “pulse2percept: A Python-based simulation framework for bionic vision”. In: *bioRxiv* (2017). DOI: 10.1101/148015. eprint: https://www.biorxiv.org/content/early/2017/07/10/148015.full.pdf. URL: https://www.biorxiv.org/content/early/2017/07/10/148015.

[3] Edward Bloch, Yvonne Luo, and Lyndon Cruz. “Advances in retinal prosthesis systems”. In: *Therapeutic Advances in Ophthalmology* 11 (Jan. 2019), p. 251584141881750. DOI: 10.1177/2515841418817501.

[4] Spencer C. Chen et al. “Simulating prosthetic vision: I. Visual models of phosphenes”. In: *Vision Research* 49.12 (2009), pp. 1493–1506. ISSN: 0042-6989. DOI: https://doi.org/10.1016/j.visres.2009.02.003. URL: https://www.sciencedirect.com/science/article/pii/S0042698909000467.

[5] Naïg Chenais, Marta Airaghi Leccardi, and Diego Ghezzi. “Photovoltaic retinal prosthesis restores high-resolution responses to single-pixel stimulation in blind retinas”. In: *Communications Materials* 2 (Mar. 2021). DOI: 10.1038/s43246-021-00133-2.

[6] Nicole Han et al. “Deep Learning–Based Scene Simplification for Bionic Vision”. In: *Augmented Humans Conference 2021*. AHs’21. Rovaniemi, Finland: Association for Computing Machinery, 2021, pp. 45–54. ISBN: 9781450384285. DOI: 10.1145/3458709.3458982. URL: https://doi.org/10.1145/3458709.3458982.

[7] Melani Sanchez-Garcia, Ruben Martinez-Cantin, and José Jesús Guerrero. “Indoor Scenes Understanding for Visual Prosthesis with Fully Convolutional Networks.” In: *Visigrapp (5: Visapp)*. 2019, pp. 218–225.

[8] Jing Wang et al. “The application of computer vision to visual prosthesis”. In: *Artificial Organs* 45.10 (2021), pp. 1141–1154. DOI: https://doi.org/10.1111/aor.14022. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/aor.14022. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/aor.14022.

[9] Ying Zhao et al. “Image Processing Strategies Based on Deep Neural Network for Simulated Prosthetic Vision”. In: *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*. Vol. 01. 2018, pp. 200–203. DOI: 10.1109/ISCID.2018.00052.

[10] David D. Zhou, Jessy D. Dorn, and Robert J. Greenberg. “The Argus® II retinal prosthesis system: An overview”. In: *2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*. 2013, pp. 1–6. DOI: 10.1109/ICMEW.2013.6618428.