CSE 493V Final Project Report

Experience spatial audio with any headphones.

YURII HALYCHANSKYI and SAYUJ RAJ SHAHI, University of Washington

Fig. 1. This screenshot depicts a scene where a cube serves as the sound source. The cube can be repositioned in various directions, including the z-axis, which alters the perception of the sound. Additionally, the user has the ability to adjust the yaw rotation of the camera using their head position, further influencing the perception of the sound.

Spatial audio, which recreates immersive soundscapes by simulating sound sources' positional cues, has become increasingly popular in today's age. However, achieving spatial audio reproduction for diverse headphone models remains a challenge. This paper presents a novel approach that utilizes a machine learning model, FaceMesh, to track the user's face and accurately position sound sources based on the user's perspective. Furthermore, the relative position of a virtual cube in a game environment is leveraged to generate spatial audio effects.

The proposed system capitalizes on the advancements in computer vision and machine learning techniques to track the user's face in real-time. By obtaining the facial movement data, the system can determine the user's head position respective to the cube. Using this information, spatial audio effects are created to reflect the perceived positions of virtual sound sources relative to the user's perspective.

Moreover, the system incorporates occlusion effects to simulate the obstruction of sound when it passes through solid objects in the virtual environment. By detecting the presence of hard mediums between the sound source and the user's virtual perspective, the system dynamically adjusts the audio to provide a realistic perception of occlusion, enhancing the immersive experience.

Overall, this report offers a solution towards providing widespread access to spatial audio for a broad range of customers who possess a camera and a headphone.

Authors' address: Yurii Halychanskyi, yhalyc@cs.washington.edu; Sayuj Raj Shahi, sayujs@cs.washington.edu, University of Washington.

1 INTRODUCTION

Our objective was to make spatial audio accessible to everyone with a camera and headphones, considering that not everyone can afford high-priced headphones marketed with spatial audio capabilities. We embarked on this challenge armed with new machine learning models capable of tracking human faces. Initially, we believed that existing state-of-the-art machine learning models for tracking head position in three dimensions, along with readily available machine learning libraries in C#, would suffice for our project in Unity.

However, we encountered setbacks when some of the machine learning libraries we intended to use in C# failed to run on our machines. Overcoming these technical hurdles, we obtained the FaceMesh model from Google, which provided us with head orientation information. Nevertheless, we faced an inherent drift issue in the z-direction (pitch) of head position tracking. Even when the user faced the camera directly, the head position drifted upwards. To simplify our approach, we decided to focus solely on tracking yaw, as it had a significant impact on spatial audio. By allowing users to turn their heads, the perceived direction of sound sources changed, providing a sense of directionality and sound localization.

In order to enhance our control over spatial audio, we have made the decision to create a spatial audio library from the ground up in

Unity. Specifically, we have successfully integrated the following functionalities: the Doppler effect, stereo panning/spread, volume roll-off, and audio occlusions. Detailed explanations of each effect will be provided in the upcoming sections.

Despite this limitation, our model performed exceptionally well, delivering directional cues to the user based on sound. This achievement aligned with our primary goal outlined in the proposal. While the model only supported yaw head rotation, we strongly believe that the spatial audio enhancements it offers will greatly benefit games that do not require full 3D head rotation. By accurately locating the origin of sound effects, such as footsteps and gunshots, the model adds a heightened sense of realism to gameplay that relies on sound awareness. Remarkably, we managed to create this project within a few weeks, but we recognize that it opens up numerous avenues for exploration in making spatial audio universally accessible.

Moving forward, we envision integrating this model with inertial measurements to provide more robust and accurate measurements. This would further expand the possibilities of our spatial audio solution and enable a more immersive audio experience for a wider audience.

1.1 Contributions

- Made spatial audio work, regardless of the type of headphones, to users with access to a camera.
- Developed a pipeline that enables the utilization of a diverse array of machine learning algorithms within the Unity environment.
- Implemented a feature that enables users to control the camera by utilizing their head position in Unity.

2 RELATED WORK

As the popularity of VR systems continues to soar, the need for spatial audio has also surged, leading to extensive research in this field. In Chapter 11 of the book titled "Virtual Reality," Steven M. LaVelle [3] offers an extensive exploration of the physics of sound in three-dimensional space. The theoretical foundations of the spatial audio methods implemented in this project draw significant inspiration from this work.

In the publication "3-D Ambisonics Experience for Virtual Reality," [2] the authors present a valuable review on creating a more immersive spatial audio experience through advanced techniques like ambisonics and head-related transfer function (HRTF). The paper provides insights into leveraging these technologies to achieve a heightened sense of depth and realism in virtual environments.

3 METHOD

Our implementation focused solely on the software aspect. Initially, we attempted to utilize Machine Learning libraries available in C#, but after numerous unsuccessful attempts, we decided to explore alternative options. We discovered that we could leverage a machine learning library in Python while still being able to write scripts in C# for Unity.

To achieve this, we devised a server-side approach where we could write the machine learning code and predictions in Python. We then established communication between the server and our C# scripts by making calls to the server to access the model's predictions. Based on these predictions, we implemented logic to adjust the orientation of the main Camera in Unity, ensuring that it mirrored the user's head position.

In addition, we incorporated spatial audio scripts into the cube, which served as the sound source. This allowed us to achieve the desired spatial audio effect, enhancing the overall immersive experience for the user. By harnessing the characteristics of the audio source object, we successfully generated a diverse range of effects, such as audio occlusions, the Doppler effect, stereo panning/spread, and volume rolloff.

4 IMPLEMENTATION DETAILS

On the hardware side, we relied on the use of headphones and utilized the laptop's built-in camera. To enable head tracking functionality, we incorporated powerful tools such as OpenCV and MediaPipe, leveraging the FaceMesh model developed by Google. By tracking the position of the lateral canthal region of the eye, we were able to estimate the user's head position accurately.

To seamlessly integrate these components, we employed the Flask web framework. It played a crucial role in handling routing and establishing the connection between our function responsible for executing the model and a designated URL. This URL was continuously called within our Unity script to retrieve the model's results effectively.

To ensure smooth communication, we took advantage of Unity's network API. This enabled us to establish consistent contact with the server and retrieve the outcomes generated by the model. Leveraging this data, we computed the variance between the user's current and previous facial positions. By applying specific thresholds, we were able to determine the yaw of the user's head. We then applied this information to adjust the rotation of Unity's main camera.

As a result, the user's perspective within the virtual environment dynamically responded to their head movements, resulting in a truly immersive experience.

Regarding spatial audio, we have implemented the following features:

1. **Doppler Effect**: The Doppler Effect occurs when a sound source and a listener are moving relative to each other. This relative motion causes the frequency of the sound perceived by the listener to change. This is why a car speeding past you sounds different as it approaches you versus when it is moving away. The observed frequency is given by:

$$f' = f\frac{c + v_r}{c + v_s}$$

where:
- f' is the observed frequency,
- f is the emitted frequency,
- c is the speed of sound,
- v_r is the speed of the receiver relative to the medium,
- v_s is the speed of the source relative to the medium.

2. **Stereo Panning/Spread**: This effect is used to control the perceived location of a sound source in the stereo field. By adjusting

the relative amplitude of the sound in the left and right channels, the sound can appear to come from any point between the two speakers. The amplitude for the left and right channels is calculated as:

$$A_L = P \cdot A$$
$$A_R = (1 - P) \cdot A$$

where:
- A_L and A_R are the amplitudes for the left and right channels, respectively,
- A is the initial amplitude,
- P is the pan position (0.0 = hard left, 1.0 = hard right).

3. **Volume Rolloff**: This is a simulation of the natural phenomenon that sound gets quieter as you get further away from the source. This is implemented by reducing the amplitude of the sound based on the distance between the source and the listener. The amplitude at a distance d can be calculated with:

$$A_d = \frac{A}{1 + k \cdot d}$$

where:
- A_d is the amplitude at distance d,
- A is the original amplitude,
- k is a rolloff factor,
- d is the distance from the source.

4. **Audio Occlusions**: An audio occlusion occurs when an object in the environment blocks or reduces sound from a source. This effect is used to simulate the real-world behavior of sound being muffled or blocked by objects in the environment. The amplitude after occlusion can be modeled as:

$$A_o = A \cdot e^{-\alpha d}$$

where:
- A_o is the amplitude after occlusion,
- A is the original amplitude,
- α is an absorption coefficient,
- d is the thickness of the occluder.

5 EVALUATION OF RESULTS

Fig. 2. The game scene when the user looks to the left.

Fig. 3. The game scene when the user looks to the right.

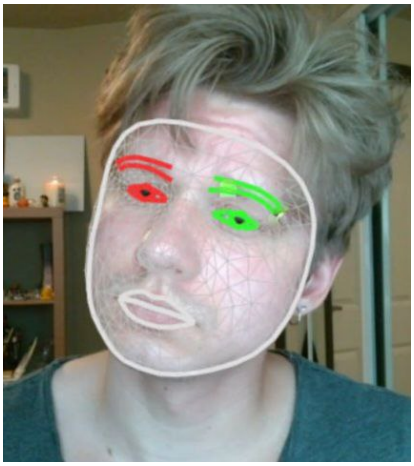

Fig. 4. The visualization of face tracking.

Utilizing the innovative amalgamation of camera-based machine learning and spatial audio scripts, we've crafted a model that opens up the world of spatial audio to anyone with access to a simple camera and a pair of headphones. This model not only democratizes spatial audio experience but also introduces the concept of directional cues, adding a new layer of immersion to your audio-visual journey. It holds potential to help users experience a sense of realism in games, particularly in the games that lean heavily on sound awareness, enriching the gaming experience by creating an environment where sound sources can be perceived directionally and with a sense of depth. Additionally, it incorporates occlusion effects, offering an enriched, realistic soundscape that modifies audio based on objects in the environment.

The model has certain limitations that need to be addressed for further improvement. Currently, it can only track the yaw, or horizontal rotation, of the head. To provide a more accurate spatial audio experience, the model requires enhancements to track the complete 3D orientation of the head using quaternions.

Another notable drawback is that the model relies on a stable internet connection to operate. This dependence on connectivity restricts users from enjoying the spatial audio experience in offline or low-connectivity situations, limiting its accessibility.

Furthermore, the model imposes limitations on the user's freedom of movement within the camera's field of view. Users must remain within a specific range or angle to maintain accurate tracking, which can be restrictive and hinder natural movement during audio-visual experiences.

Lastly when we tested the model in differnt laptops [with different] cameras, we found the model to perform well with some laptops over the other.

6 DISCUSSION OF BENEFITS AND LIMITATIONS

We have repeatedly iternated that this model allows anyone with a headphone and a camera to experience spatial audio. So, in this section, let's focs on the limitations.

The user must provide a stable internet connection for the model to run. Recall, that our script makes repeated calls to the server to access the results of the FaceMesh model, so the model won't be able to serve the users in times of network inavailability or failures. Also recall that the model simply tracks yaw due to the inherent drift, so it limits the users from playing games that requires 3d head rotation.

The user's face must be visible at all times to the camera to allow the FaceMesh model to track the face. So, the model also restricts user movement.

Lastly, since the model's performance appeared to be influenced by the specific characteristics and capabilities of the cameras used in different laptops, the model exhibited limitations in terms of accuracy and reliability.

7 FUTURE WORK

Incorporating advanced spatial audio techniques such as ambisonics, binaural audio, and head-related transfer function (HRTF) would serve as a valuable expansion for this project. By doing so, users would be able to discern the source of sound with greater precision, resulting in an enhanced spatial audio experience overall. In terms of head position tracking, extending the algorithm to encompass all three dimensions of yaw, pitch, and roll for the camera would provide users with increased freedom and an improved immersive experience.

8 CONCLUSION

This project presents a development in the field of Augmented Reality (AR) and Virtual Reality (VR), offering a novel system that leverages FaceMesh, a machine learning model, to monitor user facial movements and accordingly position spatial audio effects in a gaming environment. By integrating real-time computer vision and machine learning techniques, the system provides a more realistic auditory experience to users. This includes simulating occlusion effects, which replicate the nuanced audio changes experienced when sound travels through solid objects. This advancement in AR/VR could significantly enhance the immersion and realism of virtual environments, and, with its accessibility to users with a camera and headphones, democratize high-quality VR experiences. If more researchers invest in this area, we might see a shift in VR from visual-heavy to a more complete sensory experience.

ACKNOWLEDGMENTS

We would like to extend our thanks to the instructor and the TAs for their continuous help and support throughout this project.

REFERENCES

[1] Robert Konrad et al., *Explorations in Spatial Audio and Perception for Virtual Reality*, (2016).
[2] Cedric Yue and Teun de Planque, *3-D Ambisonics Experience for Virtual Reality*, (2017).
[3] Steven M. LaValle, *Virtual Reality*, Cambridge University Press, (2023).