EE267 Virtual Reality：Project Report

MedSpace : a virtual reality environment for a meditative experience

Abhijeet Phatak
aphatak@stanford.edu

Abstract

A virtual reality based visual meditation space offers the user a way to meditate even in today's busy lives. Realistic audio-visual elements designed to soothe the senses and minimalistic interactions can deliver a positive experience. In this project, I worked on comparing different hand pose estimation/gesture detection methods using RGB webcam images, in terms of accuracy and feasibility for this application. In the future, a social dimension to this can be added, where people can meditate/chant collectively together through their VR headsets. The future goal is create more environments that the user can choose from and explore procedural worlds.

1. Introduction

Meditation is a practice where an individual uses a technique – such as mindfulness, or focusing their mind on a particular object, thought or activity – to train attention and awareness, and achieve a mentally clear and emotionally calm and stable state. It is practiced by people across the globe in different forms and is supposed to have amazing effects on our psyche.[1]

Virtual reality is a conducive platform for this application because the user has less visual information about the environment as they are accommodated to the screen. This would possibly let the user focus better. There has been some research on how the human brain reacts differently to a VR scene as opposed to a natural scene which also motivated me to work on this project.[2]

Although we want to keep interactions to a minimum, we still need to make the user be able to switch between scenes, change volume, adjust brightness (if use is prolonged). Conventionally, VR applications either leverage a depth camera (Intel realSense, Kinect) or use a hand controller. Depth cameras have become relatively inexpensive but still cost more than conventional RGB cameras. Also, webcams are already available in modern laptops and desktop monitors. At least in a tethered setting, one can make use of this for interaction design. Even in an untethered setting, one can attach low resolution ego-centric camera to the VR headset that can then

2. Related Work

There has been a lot of progress in using VR for therapeutic applications like overcoming phobias, correcting eye disorders, treating depression *etc*.[3-5] I was inspired by an award winning art called "where do thoughts go"[6] where a user can store and listen to his recorded thoughts. They are stored as objects in the virtual space. Here the goal is similar in the sense that the visual stimulus shouldn't be too dynamic. However, looking at a static scene for prolonged durations might lead to eye strain and vergence-accommodation conflict (VAC) leading to discomfort.

3. Methods

In the following sections, different methods for hand pose estimation/ gesture recognition are explained briefly along with relevant citations. These methods have been proposed to be used in VR/AR systems using only a simple RGB camera and do not require additional hardware

3.1 Energy + Hand Model based methods

Here, a SNAKES like algorithm is used to track the model of a hand using two energy functions - one that tracks the hand and the other estimating the pose. The total energy is minimized using gradient descent. I was able to find a fast open source implementation of this idea[9,10]. This implementation needs to be calibrated for different lighting conditions. It is very simple and requires the least amount of compute resources. **File : /OpenHPE/OpenHPE_test** (might have to recompile)

3.2 Convexity Defects of fingers

Here, we try to leverage the 2D projection of the 3D geometry of the hand to identify the convexity defects. Based on the count of these convexity defects, one can map the hand pose to 5 possible outputs. This is also a computationally efficient algorithm implemented using python and openCV bindings. **File : /convexity/track2.py**

3.3 Convolutional pose machines

This is a modern method that can be applied for either ego-centric or non-ego centric cameras looking at the hands given sufficient training data is available. It implements a pose prior for the pose estimation part and relies on convolutional neural networks for visual feature extraction[11,12]. This method is accurate and resistant to occlusions. Although this is very accurate, it requires at least 6GB of GPU memory which will not give Unity access to the GPU for rendering. Hence this method was not chosen for the final implementation. **File : /hand3d/run.py**

The expected outcomes from these different models are shown below from presented results in literature:

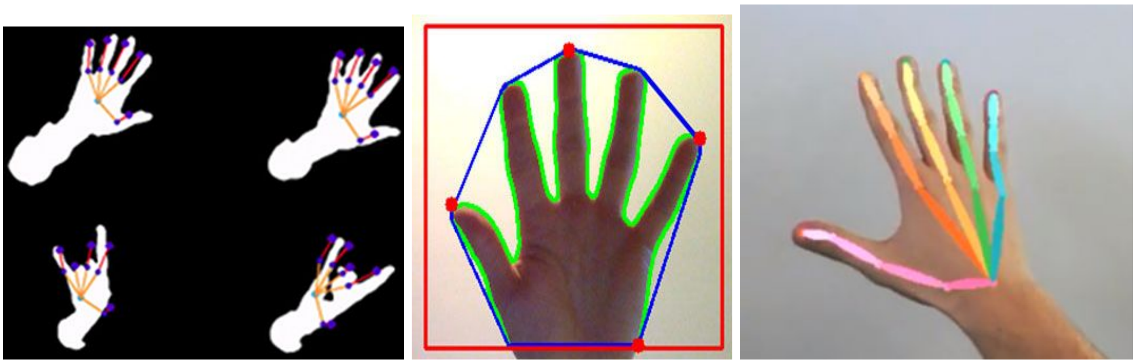


Fig : Expected Hand tracking outcomes
Energy minimization, Convexity Defects, Convolutional Pose Machine (egocentric)

Software:

The application was created using Unity3D. Assets were created using primitives supplied by Unity Engine and standard assets. A few other sources of assets included HDRI maps (https://hdrihaven.com/) for creating skyboxes which are computationally efficient way to model realistic (almost real-world) environment reflections and lighting in scenes. The provided Unity Cardboard SDK was used to model the stereo image formation and multi-stage rendering pipeline.

Hardware:

Basic hardware provided in EE267 lab (VR kit + VRduino + Teensy) for pose-tracking and I/O of audio-visual content. All algorithms for hand tracking can be run on typical laptops/desktop systems. However the convolutional pose machine requires a GPU for real-time frame rates which is also included in modern high-end systems.

Interfacing:

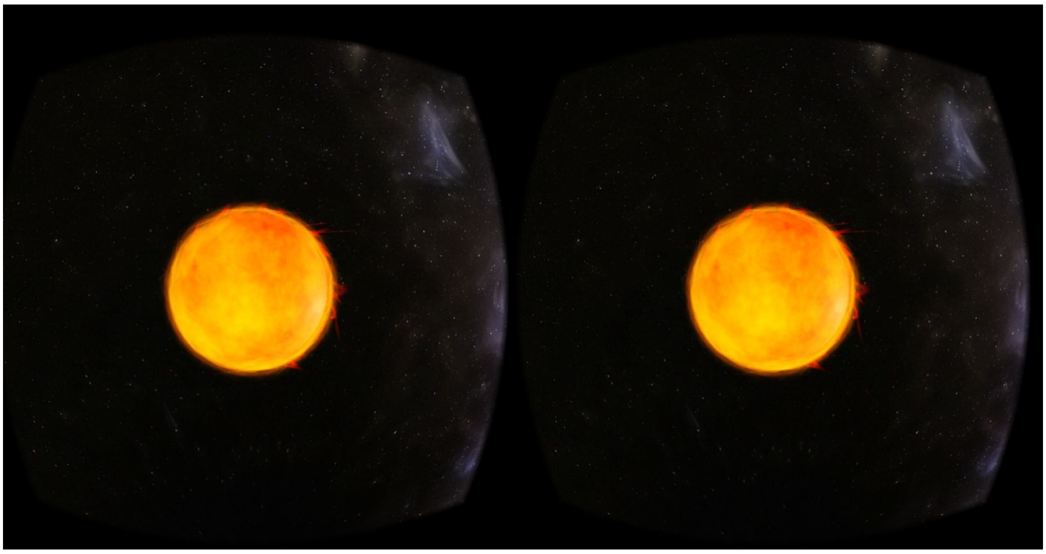
The data from the IMU after filtering is output in the form of quaternions. This is conveyed through a node server which was provided in homework assignments. In Ubuntu, the serial port communication requires a lot of OS environment variable changes and hacks, and so the decision was to use the streamed data through the node server. The hand gesture applications are python scripts that can be run with minimum requirements like openCV and tensorflow. For the final implementation of the project, I chose to use the 2nd method owing to its simplicity yet being effective.

4. Results and Discussion

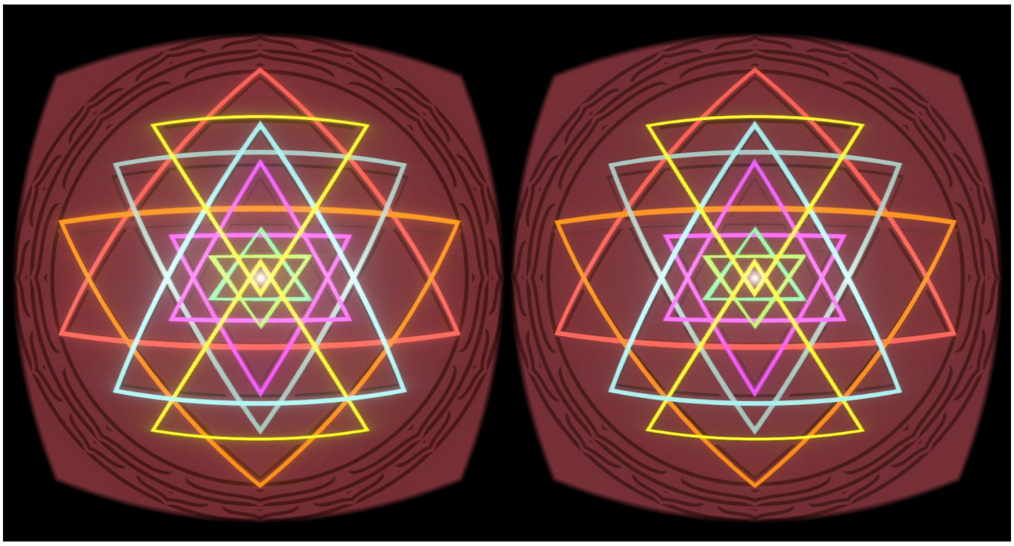
Scenes

For this iteration of the application, I created three basic scenes for the user to choose from. All models and effects were created by me.

Space : Focus on a bright glowing sun rendered using simple particle systems in outer space in a heliocentric orbit.



Sri Yantra **:** A beautiful geometric representation of the various *chakras* referenced in Yoga as energy centers in the spine ultimately leading to the pituitary gland (third eye) often used as a tool to focus in ancient Indian yoga.



Forest : A beautiful forest land with flowing stream of water as most practitioners of meditation actually seek help from nature.

Hand tracking

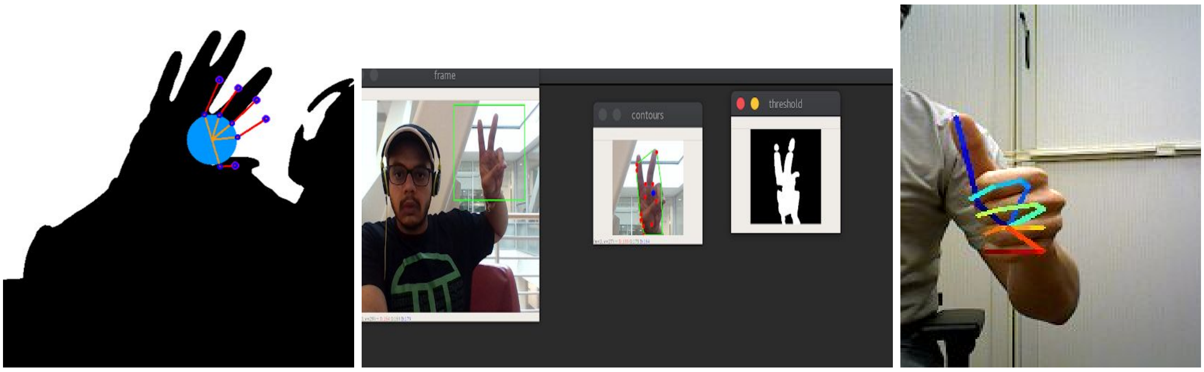


Fig : Actual Hand tracking outcomes
Energy minimization, Convexity Defects, Convolutional Pose Machine (non-egocentric)

Results and analysis of the hand gesture recognition and the feasibilities are summarized in the table below.

	Energy model based	Convexity Defect	Convolutional Pose
Function	Track - yes (2D) Only gestures	Track - yes (2D) Only gestures	Track - yes (3D) Gestures + position
Egocentric/Non ego-centric	Yes with changes	Yes with changes	Yes. Need new training data
Stability	Needs smoothening	Need a little bit of processing	Stable because pose prior
Speed	>24fps (realtime)	>24fps (realtime)	~10fps (GPU ~ 6GB)
GPU	Not needed	Not needed	Preferred

Accuracy	Not very accurate	Accurate	Highly accurate
Calibration	Required	Required	Not required

5. Learning Outcomes, Future work and Acknowledgements

I learnt how to use Unity to create VR content. Through the course material and design, I also learnt about the various considerations in making VR applications. Also learnt a great deal about hand pose estimation which is an important topic in HCI.

The future goal is to add support for other hand tracking mechanisms like leap motion or interfacing with other sensors like Intel RealSense 3D. Also to add some more scenes and improve gesture recognition. The ultimate goal is to improve the rendering and publish the application on Steam.

I extend my heartfelt gratitude to the EE267 staff (Prof. Gordon Wetzstein, project mentor Hayato Ikoma and the TA team) for the wonderful course material and providing the necessary support for easy integration with Unity.

The entire application along with necessary scripts can be downloaded from Google Drive as a zip: https://drive.google.com/file/d/1JuOdbWFIWzKAe3n2EzFdtiqa1j8bhkLN/view?usp=sharing

References

[1] https://en.wikipedia.org/wiki/Meditation
[2] Aghajan, Zahra M., et al. "Impaired spatial selectivity and intact phase precession in two-dimensional virtual reality." *Nature neuroscience* 18.1 (2015): 121.
[3] Parsons, Thomas D., and Albert A. Rizzo. "Affective outcomes of virtual reality exposure therapy for anxiety and specific phobias: A meta-analysis." *Journal of behavior therapy and experimental psychiatry* 39.3 (2008): 250-261.
[4] https://en.wikipedia.org/wiki/Virtual_reality_therapy
[5] Backus, B. T., Dornbos, B. D., Tran, T. A., Blaha, J. B., & Gupta, M. Z. (2018). Use of virtual reality to assess and treat weakness in human stereoscopic vision. *Electronic Imaging, 2018(4), 109-1.*
[6] https://store.steampowered.com/app/846080/Where_Thoughts_Go/
[7] Mueller, Franziska, et al. "GANerated hands for real-time 3d hand tracking from monocular rgb." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.
[8] Wei, Shih-En, et al. "Convolutional pose machines." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016.
[9]http://www.morethantechnical.com/2010/12/28/hand-gesture-recognition-via-model-fitting-in-energy-minimization-wopencv/
[10] https://github.com/royshil/OpenHPE
[11] Zimmermann, Christian, and Thomas Brox. "Learning to estimate 3d hand pose from single rgb images." *Proceedings of the IEEE International Conference on Computer Vision*. 2017.
[12] Wei, Shih-En, et al. "Convolutional pose machines." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016.