

Project Requirements:

Hand-Controlled Environment Explorer

Background:

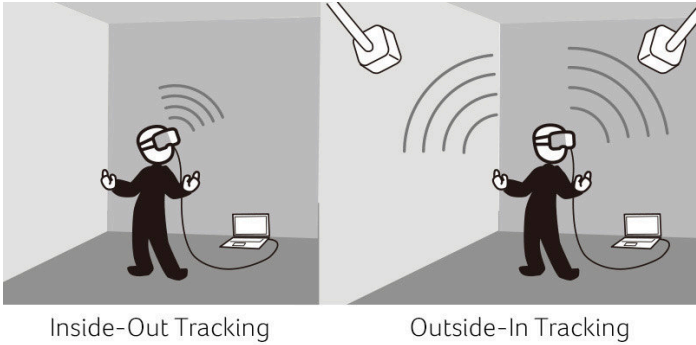
Hand tracking in a computer context, surprisingly, refers to tracking the position of a hand. This can be done in a variety of ways, such as outside-in (lighthouse) based tracking, using sensors in the environment to calculate positional data (used in the HTC Vive VR Headset¹). Alternatively, inside-out tracking uses sensors positioned relative to the hands themselves, such as the Oculus Quest's camera based tracking. These "sensors" could be a complicated system of photosensitive resistors, or simply a camera (or cameras) that tracks using computer vision.

Computer vision refers to the "field of computer science that focuses on enabling computers to identify and understand objects and people in images and videos"². Computer Vision may involve techniques such as making tracking markers out of regions of high contrast within the object, or predicting the future path of an object based on its current motion. Another approach to hand tracking with computer vision is a machine learning based approach, in which a model is trained to recognize hands (specific fingers, etc.) and place markers on them. This seems to be what Google's MediaPipe library is doing.

¹ <https://hackaday.com/2016/12/21/alan-yates-why-valves-lighthouse-cant-work/>

²

<https://azure.microsoft.com/en-ca/resources/cloud-computing-dictionary/what-is-computer-vision/#:~:text=Computer%20vision%20is%20a%20field,tasks%20that%20replicate%20human%20capabilities.>



What is being built?

This project aims to implement a computer-vision based hand-tracker that will be used to navigate a *virtual environment*. The iterative-design approach to this project will be developed as a succession of several modules, starting simple and adding more complexity as the project develops.

The hand tracking component of this project will rely on various libraries (MediaPipe⁴ for the bulk of the hand tracking, and OpenCV⁵ mainly for image manipulation) that can track the X, Y, (and possibly Z!) position of the user's hands (and possible fingers depending on feasibility of implementation) and send that movement data off to the "environment".

I use the term 'environment' to refer to whatever visual contraption I end up implementing. Initially, this will start off as just moving around a picture, but ideally will evolve into moving around an animated (possibly real-time rendered?) image with some interactive elements. While we're *wishing for a unicorn*, a further development of this implementation could involve navigating a 3D environment, which would be contingent on z-axis tracking (a very difficult thing to do), and, of course, the rendering of an actual 3D environment. EDIT: Another basic implementation of this environment (which was ultimately what the project ended up with) is to make a text-based maze (made out of special characters) that the user could navigate.

Input:

- Footage from a WebCam or other camera device for tracking
- Possibly a pre-recorded video file for testing purposes

Output:

- The visual representation of the "environment", controlled by hand movements
- WebCam feed in separate window

3

https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.researchgate.net%2Ffigure%2FInside-out-versus-Outside-in-tracking-for-HMDs-Acer-2019_fig26_349734845&psig=AOvVaw0ORDXfTpXHo_QpPp4bWGII&ust=1686096220462000&source=images&cd=vfe&ved=0CA4QjRxqFwoTCJCK1YSsrf8CFQAAAAAdAAAAABAH

⁴ <https://developers.google.com/mediapipe>

⁵ <https://docs.opencv.org/4.x/>

Modules:

As mentioned previously, this project will be broken down into 'modules' to iterate through the development. These effectively work as a schedule. The requirements version of the doc will focus on the *what* of each iteration, the design doc will focus on the *how*.

1. Basic Hand Tracking - Output Movement

This part is focused on actually getting setup with hand tracking, and outputting the movement of the hand in a usable way, accompanied by some sort of output to console to confirm it works.

This will look like:

- Getting feed from webcam (possibly displaying it)
 - In testing, video feed will likely be presented so that debugging can be accurately performed
- Moving your hand(s)
- Output to console (something like "Left 20 Pixels")

2. Constructing Environment

Next step! Making something visual that can be navigated. Initially this will be a simple, low-resolution image, which can branch out as time goes on. Right now I'm thinking like a simple square maze, but later this could be something like a map. It could evolve further (see branches), but this will depend on available time.

- *Note: consider looking into GUI libraries later (or writing something) for experimenting with this environment*⁶

This will look like:

- A visual image
- Being able to move around the image

3. Branch

Multiple options to pursue here. Subject to change. Branches include:

Moving Map

- Could look into making "environment" something more fluid, like a map that animates when you hover over it

⁶ Highgui - https://docs.opencv.org/4.x/d7/dfc/group___highgui.html

AR-Like Interactions

- Figuring out z-axis tracking and a 3D "Environment"
 - Would look like: being able to move your hand in a way that it can interact with 3D space

Application

Hand tracking finds itself useful in many different fields - in particular, I am interested in its applications for AR (Augmented Reality) and MR (Mixed Reality). These concepts involve fabricating a virtual environment that is superimposed on to a physical environment, with the former having interaction facilitated through a touchscreen, whereas the latter involves physical interaction, not dissimilar to holograms. The latter kind of interaction naturally requires the same style of tracking found in this project, as one's hand would need to be tracked in 3D space in order to interact with various virtual elements. It is actually a future project of mine to create an engaging interactive AR/MR game to be played on an AR headset, and this project here can act as a stepping stone towards that.