EE267
Heather Shen, Jason Liu

Rendering with Positional Tracking

Many current VR models have not yet implemented positional tracking and instead can only track the movement of the head. However, position-tracking VR is increasingly important for realistic experiences as well as any augmented reality experience[2]. Thus, we plan to extend our solution from HW6 to implement the rendering component associated with our position estimation as well as integrate the rotation component from the LM algorithm.

Specifically, we would support translation (using the translation values from the positional tracking) from the origin of the reference frame. This involves knowing the relative orientation between the VRduino and the lighthouse because the coordinate systems for the two would be different. We will also combine the two sources of orientation -- IMU and LM output -- to constrain orientation and reduce noise.

To showcase the project at the end, we will have live demo, where a user can put the headset on and then roam around the lab area with their position being tracked within the virtual world. One limitation of this technique is that the VRduino must be facing the lighthouse for the photodiodes to work properly. Therefore, users cannot walk with their backs facing the lighthouse and thus this limits our natural motion.

Previous Work

Currently, positional tracking is divided into two different problems. In inside-out positional tracking, the camera/sensor is located on the HMD, and no external devices are needed for tracking. In outside-in positional tracking, external cameras and sensors are required, which constrains the tracking experience to a specific area. The former problem is currently being researched more, as it as more impactful applications to the field of VR/AR. Last year, in 2016, at a Google press event, Johnny Lee of Project Tango stated that inside-out positional tracking had been solved for the smartphone; the current limitations were that it was too computationally expensive and the phones would get too hot.[3]

Our current lectures demonstrate a camera problem where, based on the number of unknowns, having four photodiodes on a plane makes it possible to fully determine a solution for the camera matrix. Raharijaona et. al, proposed a different model that involves three photodiodes being placed on different sides of a small cube. Using this setup, and the knowledge that sensors were all 90º apart, they were able to determine the azimuth and elevation of flickering IR LEDs. While their setup involved tracking remote markers, one could imagine this setup being adapted so that the cube localizes its own position.

In 2008, Thomas Petersen of Aalborg University performed a study in which he compared various 2D-3D pose estimation methods.[5] The LM algorithm we use is an extension of the Gauss-Newton method, which can also be used for pose estimation, but is less robust. PosIt is a method published by DeMenthon et. al in 1995 that finds a pose of an object from an image. The method is split in 2 where the first part approximates the perspective projection and finds the rotation matrix and translation vector from a linear set of equations. The second part iteratively uses a scale factor for each point to enhance the found orthographic projection and then repeats on the new points until some convergence.

Another method, called CamPoseCalib (CPC) is used to estimate the relative rotation and translation of an object from an initial pose to a new pose. It used the LM algorithm in its optimization, and only requires 3 points of correspondence to work.

Timeline

Our timeline right now is to first communicate the LM output via the node server and be able to retrieve the data in some render HTML page. Then, we will work on applying the correct matrix functions to properly transform an object's position and rotation in response to changes in the LM output.

References:

1. Pintaric, Thomas, and Hannes Kaufmann. "Affordable infrared-optical pose-tracking for virtual and augmented reality." *Proceedings of Trends and Issues in Tracking for Virtual Environments Workshop, IEEE VR*. 2007.
2. Bowman, Doug A., and Ryan P. McMahan. "Virtual reality: how much immersion is enough?." *Computer* 40.7 (2007).
3. https://uploadvr.com/inside-out-google-solve-tracking/
4. Raharijaona, Thibaut, et al. "HyperCube: A small lensless position sensing device for the tracking of flickering infrared LEDs." *Sensors* 15.7 (2015): 16484-16502.
5. Petersen, Thomas. "A Comparison of 2D-3D Pose Estimation Methods."