Homework 6: Solutions
CSE 490V: Virtual Reality Systems

Student Name
student@uw.edu

1 Theoretical Part

1.1 Image Formation Model for Lighthouse Pose Tracking (*10pts*)

Let's say we have the ground truth 6-DOF pose, i.e., orientation $\boldsymbol{\theta}$ (in degrees) and position $\vec{t}$ (in mm), of a "VRduino" device given as

$$\boldsymbol{\theta} = \begin{pmatrix} \theta_x \\ \theta_y \\ \theta_z \end{pmatrix} = \begin{pmatrix} 45^\circ \\ 0^\circ \\ 45^\circ \end{pmatrix}, \quad \vec{t} = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} = \begin{pmatrix} 10 \\ 10 \\ -50 \end{pmatrix}.$$

Assume that the Lighthouse base station is located at the origin in world coordinates. We also know that the four photodiodes, as mounted on the VRduino, are located at the following positions (specified in mm):

$$p_0 = \begin{pmatrix} -42 \\ 25 \\ 0 \end{pmatrix} \quad p_1 = \begin{pmatrix} 42 \\ 25 \\ 0 \end{pmatrix} \quad p_2 = \begin{pmatrix} 42 \\ -25 \\ 0 \end{pmatrix} \quad p_3 = \begin{pmatrix} -42 \\ -25 \\ 0 \end{pmatrix}.$$

As the Lighthouse base station sweeps its horizontal and vertical laser lines through the room, each of the four photodiodes will be triggered at a particular time stamp, measured in clock ticks of the microcontroller. Given the parameters above, what are these time stamps for all four photodiodes? List both horizontal and vertical time stamps for each photodiode, i.e., 8 values in total.

For your solutions, round the clock ticks to the nearest integer. Assume that the microcontroller runs at 48 MHz. Similar to the lecture and course notes, we define the sequence of rotations from local to world coordinates as yaw-pitch-roll, i.e., $R = R_z(\theta_z)\, R_x(\theta_x)\, R_y(\theta_y)$.Note: You may use your preferred tool for this problem (e.g., Matlab, Python, etc.). There is no need to submit code, but do show your derivations in your write-up.

Answer:

Write your answer to this question here.

1.2 Robustness of the Inverse Method (*15pts*)

Usually, we do not know the ground truth pose of a tracked object, such as the VRduino. To compute it from the measured time stamps, we can use the homography method as discussed in class and in the course notes. In this case, we need to construct a linear system of equations $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{h}$ and solve it for $\boldsymbol{h}$. This requires inverting the matrix $\boldsymbol{A}$. As discussed in class, the robustness of a solution to such a linear inverse problem (with respect to sensor noise or slight errors in the measurements) is defined by the condition number of the matrix. As before, can may your preferred tool for the following computations (e.g., Matlab, Python, etc.). There is no need to submit code, but show your derivations in your write-up.

(i) Show the matrix $\boldsymbol{A}$ constructed from the eight measurements you calculated in Section 1.1. (*5pts*)

(ii) Compute the singular values $\sigma_{1,\ldots,8}(\boldsymbol{A})$ and the condition number $\kappa(\boldsymbol{A}) = \frac{\sigma_{max}(\boldsymbol{A})}{\sigma_{min}(\boldsymbol{A})}$ for this matrix. Briefly discuss what this number means for the robustness of your solution to this inverse problem (with respect to noise and measurement errors). (*10pts*)

Answer:

(i) Write your answer to this question here.

(ii) Write your answer to this question here.

1.3 Arranging Photodiodes in 3D *(20pts)*

On the VRduino, all photodiodes are arranged in the plane of the device. While this makes the pose calculations a bit easier, it may not be the best approach when robustness and precision of the tracking matter. Indeed, the arrangement of the photodiodes on the controllers and the HMDs that Lighthouse is designed to track often have much more complex 3D configurations. Let's look at such a 3D photodiode arrangement in more detail. Once again, you can use your favorite tool for the following computations (e.g., Matlab, Python, etc.). There is no need to submit code, but show your derivations in your write-up.

(i) What is the minimum number of photodiodes that must be arranged in a non-planar 3D configuration to result in a "square" or "tall" matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, for $m \geq n$, in the linear system of equations? How did you come up with that number? (*10pts*)

(ii) Assume that we have the same four co-planar photodiodes listed in Section 1.1 and two additional photodiodes that extrude from the device plane and are located at:

$$p_4 = \begin{pmatrix} 0 \\ 25 \\ 10 \end{pmatrix} \quad p_5 = \begin{pmatrix} 0 \\ -25 \\ -10 \end{pmatrix}.$$

On this modified VRduino, assume we have measured the following horizontal and vertical time stamps for all 6 photodiodes and have converted them into normalized coordinates:

photodiodes	x^n	y^n
p_0	-0.2926	-0.0822
p_1	0.3296	0.9955
p_2	0.6459	0.4924
p_3	0.1919	-0.2940
p_4	0.0948	0.4814
p_5	0.3403	0.1154

Using the least squares solution $\boldsymbol{h} = (\boldsymbol{A}^T \boldsymbol{A})^{-1} \boldsymbol{A}^T \boldsymbol{b}$ to the linear system, retrieve the translation vector $\vec{t}$ and report the yaw, pitch, and roll angles in degrees. Round translations and angles (in degrees) to the nearest integer. (*10pts*)

Hint: First derive the linear system, invert it to obtain the rotation matrix R and translation vector $\vec{t}$ based on $\boldsymbol{h}$. Then refer to Appendix B of the course notes for details on how to calculate yaw, pitch, and roll from R.

Answer:

(i) Write your answer to this question here.

(ii) Write your answer to this question here.

Programming Part PDF Deliverables

2.1.2 Assessing Your Marker Tag

What are the limitations you observe with your marker-based tracking? How might these limitations be addressed, either with improved software or hardware components? Do you think marker-based tracking could be adopted for commercial systems? If so, why don't you think it's commonly used in products? If not, what limitations make marker-based tracking a poor choice and how do other systems address these limitations?

Answer:

Write your answer to this question here.

2.2.2 Tuning the Positional Tracker Filter

Report the positional filter value that you prefer (i.e., the value that suppresses positional tracking errors, without introducing objectionable latency).

Answer:

Write your answer to this question here.

2.2.3 Assessing your Positional Tracking Results

Comment on the quality of marker-based orientation tracking. What could we change about the camera and/or markers to address the limitations you observe?

comment on the quality of your positional tracking. How does it compare to commercial systems you're tried? What are the primary issues you observe, even after tuning the positional and orientation tracking filters? How might you improve the performance of 6-DOF tracking with both software and hardware upgrades to the CSE 480V kit, assuming it is still based on marker tags for positional tracking?

Answer:

Write your answer to this question here.

2.3.1 Calibrating Your Camera using OpenCV

Include your calibration parameters and details about your webcam (e.g., 2019 13-inch MacBook Pro).

Answer:

Write your answer to this question here.

2.3.2 Assessing your Camera Calibration Results

Explain why the 9 values of the recovered `camera_matrix` in `camera.yml` are reasonable. Hint: Estimate the field of view for your webcam and relate this to the values in the camera matrix. What else can you conclude about the construction of your webcam from the calibration results?

Answer:

Write your answer to this question here.
