



Mobile Telesurgery Platform in Mixed Reality

Documentation

Team member: Guanhao(Dean) Fu

gfu3@jhu.edu

Mentors:

Dr. Peter Kazanzides

Ehsan Azimi

05/08/2020

This document is meant to provide directions on how to use the delivered code for this project.

1. Dependencies

Refer to <https://lp-research.com/support/> to download any desired software for your particular system, however, the delivered code is based on a pre-compiled Unity code provided by LP-RESEARCH Inc., which can be found here:

<https://bitbucket.org/lpresearch/openzenunity/src/master/>

IMPORTANT: Please download the Unity code from the project wiki page, otherwise if you like to use the unmodified version provided by LP-RESEARCH Inc., please download the following driver and copy and paste it in your Unity directory: openzenunity/Assets/Plugins

<https://www.dropbox.com/s/rswcdxmgrvfkldz/OpenZen.dll?dl=0>

Without using this driver, you will have trouble connecting the IMU in Unity.

2. Unity Scenes and Scripts

- **HumanJointAngles - HumanJointAngles.cs**

This is the first scene you want to run, which you can check if the required “Ch36_nonPBR” [2] model is imported correctly in your system. When you run the scene, you can see 7 sliders to the left of your screen, which controls the 7 individual DOF’s of the model’s arm. The joint limits are set to be rather realistic to what a normal human’s arm can do (range of motion).

- **ConnectByNameScene - OpenZenMoveObject.cs**

This scene and attached script OpenZenMoveObject.cs is modified by me on the basis of LP-RESEARCH Inc.. If you wish to see the original unmodified version, please refer to the link listed in the dependencies section.

The modified script has the following structure:

- Under the `OpenZenMoveObject : MonoBehaviour`, you need to specify 2 things: one is the connection type of your IMU (line 22), the second is the MAC address of your IMU (line 24-25).
- In `void Start()`, the script tries to connect to the IMUs specified above, and will throw messages according to the result of connection. From line 88-89 and line 130-131, you can choose to perform other offset reset described by the user manual [1], which are “alignment reset”, “heading reset”, and “object reset”. Here, I chose to perform a “heading reset”.
- In `void Update()`, line 198, 230, and 233 are the 3 lines that control the model’s joint angle, you can choose to change this if your kinematic algorithms are better than what is currently on there. Note that the shoulder angles and the wrist angles are fairly accurate, only the elbow (line 230) is worth improving.
- There are multiple `Debug.DrawRay` present in the code to visualize the local frame of each joint of interest; this is to help me develop the forward kinematics. You can comment them out if you find them annoying.

After you run the scene:

- You can use the key “V” and “C” to change between the views.
- At any time when the scene is running, you can perform “heading rest” of each IMU of the system, by clicking the button on the top right corner of the screen.

- **Virtual_workspace - Calibration.cs**

This scene is a further modified scene and script from `ConnectByNameScene` and `OpenZenMoveObject.cs`. You need to align the middle of your palm to your calibration posts, and for each time you align it, do not move your arm, and hit “R” on your keyboard to record the transformation matrices for the 3 frames of your arm; they are in the order of: shoulder, elbow and wrist. After you obtained all the metrics, copy and paste them into MATLAB code,

calibration.m included in the root folder of the code directory: (/openzenunity/calibraton.m). After you obtained the calibrated link length of your arm, go to line 244-248, and add Matrix4x4.Translate and input your link length.

(<https://docs.unity3d.com/ScriptReference/Matrix4x4.Translate.html>)

Here, in my code, I used the true link length of the Mannequin in Unity because my calibration result has huge errors.

After you inputted the correct link length from calibration, align your palm with the bottom right corner of the mousepad, and click “Calibrate Mousepad”. This will move the mousepad to the same position of where you have it in reality.

Then, you can see a sphere aligned with the bottom right corner of the mousepad; this sphere represents the middle of the palm of yours, which is the cartesian space control of the system.

3. Hardware

Any hardware related documentations can be found on LP-RESEARCH Inc.’s website here:

<https://lp-research.com/9-axis-bluetooth-imu-lpmsb2-series/>

Reference

1. IMU User Manual:

<https://lp-research.com/wp-content/uploads/2020/04/20200330LpmsUserManual.pdf>

2. Mannequin model: <https://www.mixamo.com>