IMU Calibration

SteamVR™ Tracking

Introduction

The IMU or Inertial Measurement Unit is a hardware component on a trackable object PCB that is responsible for measuring the motion and rotation of the object. For proper tracking, it's critical that the IMU measurements be as accurate as possible. Manufacturing tolerances, the assembly process, and temperature variations cause variations in IMU measurement accuracy. A calibration routine performed on each unit after final assembly reduces the impact of these inaccuracies on tracking performance.

Overview

The calibration process involves taking positive and negative IMU measurements from three orthogonal orientations. You can think of these orientations as aligned with the +X, -X, +Y, -Y, +Z, and -Z axes, though they do not need to align with the actual axes of the IMU inside the device. With these six measurements, the calibration tool will generate data that can be inserted into your object's JSON file to improve performance.

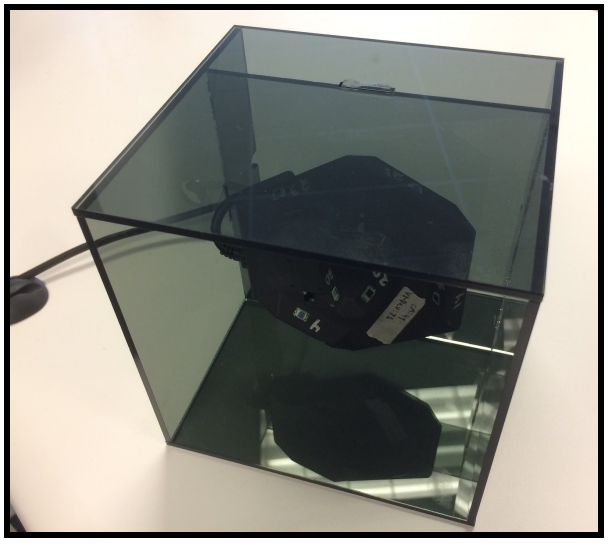
Setup

To perform calibration, you will need the following:

1. **imu_calibrator.exe** - This command-line utility is located in the following folder.

   ```
   C:\Program Files (x86)\Steam\steamapps\common\SteamVR Tracking HDK\tools\bin\win32
   ```

2. **Trackable object** - To verify the performance of the object after calibration, the object should contain a valid JSON file. IMU calibration information is added to the JSON file for each object. If the object already contains a JSON file, download the file using lighthouse_console.
3. **Sturdy surface** - For best results, the trackable object should rest on a solid, stationary, level surface free from movement or vibration.
4. **Trackable Object Mount** - To guarantee that measurement orientations are as orthogonal as possible, it is recommended that the device be mounted securely in a rectilinear object during calibration. An example of such a mount is shown below.



Running Calibration

To perform calibration, follow these steps:

1. Connect your trackable object to the host PC over USB.
2. Run imu_calibrator.exe from the command line. It will attempt to connect to your trackable object. If successful, it will say:

   ```
   Press enter to sample, 'q' to quit.
   ```

3. Place your object in its first orientation, make sure it isn't moving, and press enter. You will see:

   ```
   Recording 5000 samples
   ```

   and a few seconds later, an output similar to the following:

   ```
   Accel Average = (-0.179, -0.4414, 9.769)
   Accel Variance = (0.003485)
   Gyro Average = (-0.03114, -0.001897, 0.01296)
   Gyro Variance = (0.0001806)
   ACCEPTED. Sample 1 / 6.
   ```

4. Again, it will prompt you to:

   ```
   Press enter to sample.
   ```

5. Move the device to a new orientation orthogonal to the first and press enter.
6. Repeat steps 3-5 for the remaining four sides of the cube being careful not to repeat the same side twice. If you repeat a side, you will see the following message and be returned to the previous prompt.

   ```
   REJECTED. Duplicate orientation
   ```

   When finished, you will see:

   ```
   DONE.
   ```

Calibration Results

When complete, the calibration tool prints the output to the command line.

```
Calibrating to gravity sphere, radius 9.8066
0.04417 accelerometer fit error (6 sample vectors x 8 subsamples per vector)
```

After a successful calibration the accelerometer fit error should be less than 0.1. If the fit error is too high, verify that the object is mounted securely in its frame and that the table surface is stable and level. If the fit error is repeatedly above 0.1, it's possible that the IMU has been damaged.

The final output of the IMU calibration tool is a snippet of JSON code for the object's JSON file.

```
"acc_scale" : [ 0.998, 0.9983, 0.9915 ],
"acc_bias" : [ 0.05089, -0.03676, -0.2253 ],
"gyro_scale" : [ 1.0, 1.0, 1.0 ],
"gyro_bias" : [ 0.06253, 0.01054, -0.02128 ],
```

If calibration was successful, copy the code into the "imu" section of the JSON file and upload it to the object using lighthouse_console. For more information about how to edit the JSON file, see the document **The JSON File**.

Complete Output Example

The following is a complete example of the expected output from imu_calibrator.exe:

```
C:\>imu_calibrator
Version:  imu_cali.exe (buildbot_vortex-windows_steamvr_rel_win32@vortex-windows)
03586200
FindAndAcquireLighthouseImu vid: 28de, pid: 2300, desc: IMU, serial: LHR-71D20826
Acquired Lighthouse IMU.
Attempting HID Open IMU: LHR-71D20826
hid_open_nths
        vid=0x28de, pid=0x2300, sn=LHR-71D20826
        vid=0x28de, pid=0x2300, sn=LHR-71D20826
        vid=0x28de, pid=0x2300, sn=LHR-71D20826
HID opened: VID 28de PID 2300 serial LHR-71D20826 seq 1 | if 0
Lighthouse IMU HID opened
LHR-71D20826: Firmware Version 1464829396 seanke@wiseeye11-w 2016-06-01 FPGA 0.6
LHR-71D20826: Successfully fetched gyro/accelerometer range modes from the device.
GyroRangeMode:3 AccelRangeMode:2
Estimated IMU rate 1000Hz
Attempting HID Open Optical: LHR-71D20826

Press enter to sample, 'q' to quit.
hid_open_nths
        vid=0x28de, pid=0x2300, sn=LHR-71D20826
        vid=0x28de, pid=0x2300, sn=LHR-71D20826
        vid=0x28de, pid=0x2300, sn=LHR-71D20826
HID opened: VID 28de PID 2300 serial LHR-71D20826 seq 2 | if 1
Lighthouse Optical HID opened
Attempting HID Open VrController: LHR-71D20826
hid_open_nths
        vid=0x28de, pid=0x2300, sn=LHR-71D20826
        vid=0x28de, pid=0x2300, sn=LHR-71D20826
        vid=0x28de, pid=0x2300, sn=LHR-71D20826
HID opened: VID 28de PID 2300 serial LHR-71D20826 seq 3 | if 2
Lighthouse VrController HID opened

Recording 5000 samples...
    Accel Average = (-0.03243, -9.854, 0.1275)
    Accel Variance = (0.003163)
    Gyro Average = (0.06223, 0.01031, -0.02125)
    Gyro Variance = (9.465e-006)
ACCEPTED. Sample 1 / 6.

Press enter to sample, 'q' to quit.

Recording 5000 samples...
    Accel Average = (0.1524, 9.774, -0.7174)
    Accel Variance = (0.002327)
    Gyro Average = (0.06255, 0.01053, -0.02096)
    Gyro Variance = (8.942e-006)
ACCEPTED. Sample 2 / 6.

Press enter to sample, 'q' to quit.

Recording 5000 samples...
```

```
    Accel Average = (-9.774, 0.09511, -0.3012)
    Accel Variance = (0.002878)
    Gyro Average = (0.0625, 0.01059, -0.02108)
    Gyro Variance = (9.478e-006)
ACCEPTED. Sample 3 / 6.

Press enter to sample, 'q' to quit.

Recording 5000 samples...
    Accel Average = (9.875, -0.1822, -0.3791)
    Accel Variance = (0.002393)
    Gyro Average = (0.06266, 0.01053, -0.02145)
    Gyro Variance = (8.818e-006)
ACCEPTED. Sample 4 / 6.

Press enter to sample, 'q' to quit.

Recording 5000 samples...
    Accel Average = (-0.09916, -0.4699, -10.11)
    Accel Variance = (0.002386)
    Gyro Average = (0.06296, 0.01092, -0.02189)
    Gyro Variance = (8.582e-006)
ACCEPTED. Sample 5 / 6.

Press enter to sample, 'q' to quit.

Recording 5000 samples...
    Accel Average = (0.08173, 0.5885, 9.645)
    Accel Variance = (0.002718)
    Gyro Average = (0.0623, 0.01034, -0.02108)
    Gyro Variance = (1.028e-005)
DONE.

Calibrating to gravity sphere, radius 9.8066
0.04417 accelerometer fit error (6 sample vectors x 8 subsamples per vector)

"acc_scale" : [ 0.998, 0.9983, 0.9915 ],
"acc_bias" : [ 0.05089, -0.03676, -0.2253 ],
"gyro_scale" : [ 1.0, 1.0, 1.0 ],
"gyro_bias" : [ 0.06253, 0.01054, -0.02128 ],
```