LAB #4 KINECT AND DATA ANALYSIS

OBJECTIVE

- Learn how to acquire and display Kinect data in MATLAB;
- Get familiar with the SMART MATALB software;
- Learn how to quantify inertial sensor noises.

INTRODUCTION

I. Microsoft Kinect

Microsoft Kinect is a motion sensing device that was initially designed for gaming but later found many other applications. It is composed of a 3D depth sensor, a regular RGB camera, and an array of four microphones. A photo of the Kinect is shown in Figure 1.

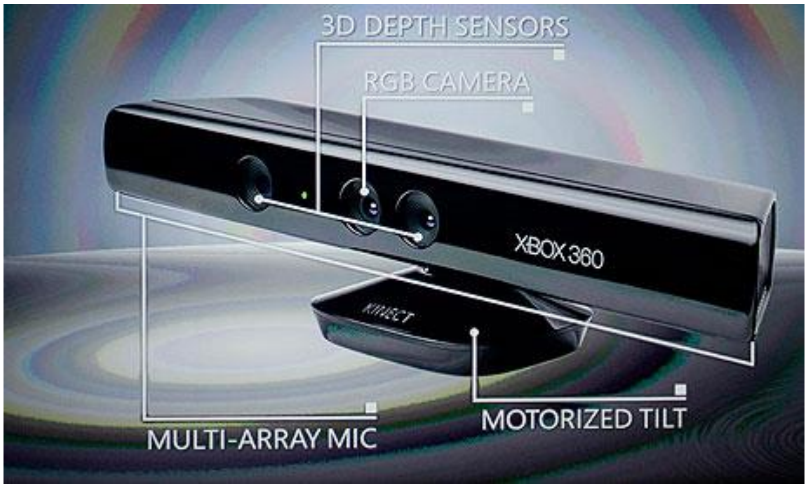

Fig.1. Kinect and Its Layout

The Kinect can provide real-time RGB-Depth data at 30Hz with a spatial resolution of 640 × 480. The sensor has an angular field of view of 57° horizontally and 43° vertically. The range of depth sensing is approximately between 80cm and 400cm. Kinect has been used widely in robotics mapping applications; therefore it's included in the senor suite of the SMART robot design.

Microsoft provided a Software Development Kit (SDK) for Kinect and a large number of samples. During this experiment, we will be looking at some of the samples.

II. Acquiring Kinect Data in MATLAB

Kinect is supported in MATLAB (from version 2013a on) and its data can be acquired using the MATLAB Image Acquisition Toolbox. A detailed discussion about how to acquire data from Kinect is provided in the link below: http://www.mathworks.com/help/imaq/examples/using-the-kinect-r-for-windows-r-from-image-acquisition-toolbox-tm.html

In short, Kinect will show up in MATLAB as two independent sensors, an color sensor and a depth sensor. We need to first define a video input object that links to the sensor we want to use (color or depth). After setting parameters and starting the device, every time we will need to use a trigger to acquire a frame of image (color or depth). The image will just show up as a matrix of data in the memory so we can use later for different purposes. After done using the sensor, the Kinect object needs to be properly terminated.

To illustrate this process, two sample programs are provided. The first one, 'Kinect_Data_Visualization.m', will show you the raw data collected from Kinect. The second one, 'Kinect_To_Laser_Scanner.m' converts the Kinect 3D depth data into a 2D laser scanner. You can use this scanner for the Simultaneous Localization and Mapping (SLAM) algorithm if you are working on the mapping subtask of the final project.

III. SMART MATLAB Software

Several pieces of MATLAB code was written for the SMART robot. The code 'SMART_RUN.m' contains the main robot sensing and control loop. It initialize the serial ports and the Kinect, and then starts a (approximately) 10 Hz main loop. Within the loop several functions are called, such as 'Read_Create_2' for reading the sensor data from the iRobot Create; 'Read_Logger_2' for reading the IMU and rangfinder data from the sensor interface board; 'Attitude_Estimation' for estimating the robot pitch, roll, and yaw angles with an Extended Kalman Filter (EKF); and 'SetDriveWheelsSMART' for controlling the robot wheel speed while checking the cliff and bump sensors. All collected data are saved in a structure called 'SD' in the memory. They will also be saved in the file 'SMART_DATA.mat' after finishing the code. You can use the program 'SMART_PLOT' to review all saved robot data.

IV. Warnings

1. Do not keep the Create powered on unattended. The robot may drive itself off the table. Only turn on the robot power when you are ready for a test and turn it off right away when you are done with the test.
2. Flip the laptop screen up when you are driving the robot. Do not use the laptop as a bumper!
3. If your laptop needs an update when restarting, let it finish. Do not turn off the laptop in the middle of an update.
4. Make sure to properly shutdown the laptop after each session. Do not put the laptop in sleep! It will just drain the battery after sitting idle for a week!
5. Kinect use LASER for mapping. This is a low power laser that is projected at many directions. Microsoft claim it's harmless to our eyes but we all know how much we can trust Microsoft. My recommendation is not to stare at the laser window (when it's red) for an extended period of time. At least we are not those kids that play video games all day long!

PROCEDURE

I. Get Familiar with Kinect

1. The instructor will first show a demo about how Kinect works; Try a few demos within the Kinect Toolkit (pinned to the task bar) yourself;
2. Close all the demo programs before starting the next task.

II. Use Kinect in MATLAB

1. Run the provided MATLAB demo programs 'Kinect_Data_Visualization.m' and 'Kinect_To_Laser_Scanner.m'; You can find the code in the directory: D:SMART Software: SMART MATLAB Code;
2. Read line by line in the code and understand the purpose of each function;
3. Change parameters in 'Kinect_To_Laser_Scanner.m' (save as a new name!) so that the Kinect will do a vertical scan instead of a horizontal scan;
4. Calibrate the distance measurements from the Kinect center points using a multi-point linear calibration approach;
5. Determine what is the depth measurement range of the Kinect;
6. Determine if the Kinect will interfere with another Kinect from a different team;

7. Show the instructor your results.

III. Analyze the IMU Data

1. Try to understand the provided SMART MATLAB codes;
2. Collect 60 seconds of data from the IMU;
3. Determine the mean and standard deviation of each accelerometer and gyroscope channels with the robot sitting still;
4. Determine if the noises are white and Gaussian;
5. Determine the covariance matrix for the measurements $\begin{bmatrix} a_x & a_y & a_z & p & q & r \end{bmatrix}$ under a static condition, where p, q, and r are roll, pitch and raw rates respectively;
6. Show the instructor your results.

DELIVERABLE

An abbreviated lab report of your experiments and answer the following questions:

1. Is Kinect a linear depth measurement device?
2. What is the minimum and maximum depth sensing range for Kinect?
3. Are the Kinect measurements accurate? What is the maximum error that you recorded?
4. What is the covariance matrix for the measurement noises of the following channels: $\begin{bmatrix} a_x & a_y & a_z & p & q & r \end{bmatrix}$? Are there any two channels that are highly correlated?
5. Are the accelerometer and gyroscope noises white and Gaussian? How could you tell?