

ESE650 Project 3: Gesture Recognition

Due Date: 2/25/2016 at 1:20pm on Canvas, and in class

In this project, you will implement Hidden Markov Models to recognize motion gestures. Using IMU sensor readings from gyroscopes and accelerometers, you will learn a set of HMM models to recognize different arm motion gestures. You should then be able to classify unknown motions in real-time using your algorithms.

1. **DOWNLOAD DATA.** A set of IMU data will be available for you at <https://upenn.box.com/2016ese650-project3-train>. These data sets contain the raw IMU readings and will be provided with corresponding labels that describe the arm motions associated with the movements. Download these files and be sure you can load and interpret the file formats. This dataset were collected from a consumer mobile device so there is no need to consider bias/sensitivity issues as you may have done in the last project. The coordinate system used is: <http://developer.android.com/reference/android/hardware/SensorEvent.html> in case you wonder.

*Data format for each column in the IMU data is

TS Ax Ay Az Wx Wy Wz
(Time (millisecond), 3x Accelerometer (m/s²), 3x Gyroscope (rad/sec))

2. **GETTING INTUITION.** Experiment with filtering and quantizing the raw sensor information using Kalman filters. This will allow you to visualize the underlying orientation information as you did in the previous project. Given your filtered observations, you can then train HMMs for each motion gesture class using the Expectation-Maximization procedure.
3. **IMPLEMENT HMM.** You will use HMM models to describe the corresponding motion gestures. You should initially use a simple left-to-right HMM's in the dynamical models. Write down the corresponding model parameters that will need to be learned from the training data.

4. TESTING. You should then make sure that your program can take input sensor readings from unknown gestures. You can then compute the log likelihood under the different HMM models, and then show how certain you can classify the unknown motion as a gesture.
5. SUBMISSION. You will upload to Canvas a written description of your algorithm in PDF form and a zip file of your code. Use the naming convention “project3_[YourPennKey].pdf” and “project3_[YourPennKey].zip”.
6. PRESENTATION. For the presentation in class you are expected to bring your own laptop or use the classroom computer. The classroom computer has MATLAB installed but no compatibility is guaranteed. The projector has a VGA port and you may need a VGA adaptor for your laptop. During the presentation, you will be asked to present your algorithm and run your code on a set of test data. The test data will be released both online and on a USB flash disk prior to the presentations. Clearly presenting your approach and having good algorithm performance are equally important.