

Lab Course Kinect Programming for Computer Vision

Assignment 6: Human Activity Recognition

22.06.2011 - Loren Schwarz (schwarz@in.tum.de)

In this assignment, you will implement a very basic approach for human activity recognition. We will use the 3D joint location data provided by NITE that you have already used in one of the previous assignments. The goal is to write a classifier that recognizes the activity performed in every frame.

Overall, this assignment consists of recording training data for a few different activities, implementing a frame-by-frame classifier based on a multi-class support vector machine (SVM), training the classifier on your data and testing predictions on new data. For simplicity, we suggest you use Matlab for training and testing the SVM classifier.

1. Extend your NITE-based skeleton visualization application so that it can save 3D joint positions to a plain text file. Write all joint locations of one frame into a single line and create a line for each frame you capture. As a hint, you can use `fopen`, `fprintf` and `fclose` for writing plain text files in C++.
2. Record a dataset with 5-10 different movements corresponding to different activities (e.g. waving a hand, clapping, punching, etc.). Each movement should be recorded at least 3 times (so you have 2 repetitions for training and 1 for testing). Make sure you create separate files for each repetition, so that you can easily separate the movements later.
3. Download the SVM-KR Matlab toolbox¹ and familiarize yourself with the included examples. If you prefer to work in C++, you are free to select any available multi-class SVM toolbox.
4. Read your recorded files into Matlab (e.g. using `dlmread`). Center the 3D points in each frame by subtracting the mean of the 3D points (in that frame) from each of them. Then, select *one* of the repetitions you recorded as the testing dataset. Using the *other* repetitions, concatenate all frames into one big training data matrix. Create a vector of class labels that contains an integer label for each frame of your training dataset (e.g. [1111....11222.....22333333....333....]).
5. Write a Matlab script to train the multi-class SVM on your labeled training data. Hint: You should use the function `svmmulticlassoneagainstall` in the SVM-KR toolbox. The main inputs are your training data matrix and the class label vector.
6. Write a Matlab script to test your multi-class SVM on your *testing* dataset that you left out in step 4. Hint: you should iterate over all testing data frames and get the SVM prediction using the `svmmultival` function of the SVM-KR toolbox.
7. Think of how to evaluate the quality of your classification on the testing dataset. Also, think of how you could make your classifier more robust with respect to outliers.

Please submit your Matlab script files or any other code you created for this assignment through the SVN.

¹ <http://asi.insa-rouen.fr/enseignants/~arakotom/toolbox/index.html>