Contents

This archive contains the 3D Gesture database described in the 2013 Entertainment Computing paper (text file containing bibtex citation included with this download)

"Salman Cheema, Michael Hoffman, Joseph J. LaViola Jr., 3D Gesture classification with linear acceleration and angular velocity sensing devices for video games, Entertainment Computing, Volume 4, Issue 1, February 2013, Pages 11-24, ISSN 1875-9521, 10.1016/j.entcom.2012.09.002"

The gesture database is split up into three directories.

1. '\data_all': Contains all gestures collected in our user study (17888 gesture samples). These are further subdivided into directories containing training and gameplay gestures for each user.
2. '\data_pruned': Contains a subset of '\data_all' that removes instances of complete failure. These are instances when, during gameplay, the user was unable to get the game system to correctly recognize the gesture within the allowed 5 attempts. These are discarded on the assumption that if it could not be recognized correctly in the maximum given attempts, it is highly likely that the user had forgotten the gesture motion. (17,557 gesture samples)
3. '\data_arff': Represents the entire gesture dataset's feature vectors exported as ARFF files. Using the provided code samples, you can either export '\data_pruned' or '\data_all' to this directory. Since we only export the feature vector for the gestures in ARFF, this directory contains 3 files per user, each corresponding to a different category of gesture samples collected in our user study. (3 files per each of the 25 users in our study)
4. '\results': Is where the results of the recognition accuracy experiments will be reported.

This archive also contains several C# code samples that cover the following functionality:

1. Load the provided gesture database from the disk (Path can be set in the Config.cs file).
2. Compute feature vectors from loaded gestures.
3. Write the feature vectors for each gesture to disk in ARFF format (which can then be analyzed using the WEKA framework).
4. An implementation of our Linear Classifier (used in the experiments described in the paper)
5. Our entire experiment setup using the Linear Classifier. You can use this to verify our results. Experiment settings can also be controlled via the Config.cs file.

Notes on running the included C# project:

1. The provided project was written in C# and built using Visual Studio 2010 Ultimate.
2. It runs as a C# console application. The entry point is in Program.cs.
3. The Experiment setup can be altered from Config.cs.
4. Each run of the experiment generates a log file which is put into the '\results' directory. Depending on the experiment setup, the log file can contain a lot of data. The most pertinent results are usually at the end of the file.
5. The 'ExperimentControl.cs' class is responsible for initialization. It also runs the experiment as well as reports results.

Author: Salman Cheema
Interactive Systems and User Experience Lab (http://eecs.ucf.edu/isuelab/)
University of Central Florida
Email: salmanc@cs.ucf.edu
If you have any further questions, drop me an email.