A Real-time Hand Gesture Recognition System

Arun Ganesan
University of Michigan, Ann Arbor

Caoxie (Michael) Zhang
University of Michigan, Ann Arbor

Abstract

Hello World

1 Introduction

Natural user interface (NUI) is a new way for human to interactive with machines. Among numerous NUIs which include multi-touch screen, eye tracking and many others, hand gesture seems to be one promising candidate. In this paper, we design and evaluate a novel hand gesture recognition system to demonstrate how much farther we are away from a actual production-level system. The reader should be noticed that we do not claim hand gesture is THE way user interface, and in fact there are some limitations such as users may feel fatigues (in the movie Minority Report, Tom Cruise has to take breaks many time due to fatigue). Similar to many other new UI systems, we propose our system as one (maybe interesting) way to interact with the computer. We do not claim that the system would replace mouse and keyboard and we leave the usability problem to future research as building hand gesture recognition system alone is quite challenging.

1.1 Design Goals

Our system is designed to maximize user experience in the sense that the user should not feel comfortable to use the system. Moreover, our system differs other existing systems in the following ways.

Just hands. The users do not need any additional physical objects to use our systems. They just need to show up their hands. Many existing system such as [SixSense, MIT Minority Report, MIT color glove] require user to wear any gloves or markers for the RGB camera to capture. We eliminate this since the user should not do anything more than showing their hands.

Real-time. Our system should run smoothly on a modern machine with a graphical card not just high-end machines. The system should also recognize hand gesture in a high frame rate. Our desired frame rate 30Hz, although the current version has around 5Hz. Therefore in our design and implementation, we try our best to save every milliseconds.

No calibration. Our system should not require a new user to do anything to calibrate the system to be used to the user's shape or etc.

Robust and Accurate Our system should have an accurate estimation of where the users' hands are and what gestures they use. Moreover, the system should be insensitive to various background, user's location, camera position and other noise.

Arbitrary gestures Our system should be able to easily incorporate new types of gestures that any developers would like to add. By training new gestures, the system can recognize arbitrary gestures, for example the American sign language.

1.2 Main Ideas

Our system would not be possible without the use of Microsoft Kinect for PC, which we are probably among the first to obtain it in February 2012. Kinect is a multi-purpose sensor providing RGB camera, depth camera and audio sensor. The SDK has offered skeletal recognition, which is far away from hand gesture recognition. The SDK also provides raw pixels for the RGB image

and depth image on a maximum frame rate of 30Hz. We use the depth image pixels for gesture recognition and both RGB and depth image for generating training samples. The depth image is the key factor that distinguishes our system most existing systems. The advantage of the depth image is that it offers an addition dimension, i.e. depth that is not present in the RGB image.

1.3 Contributions

* A data-driven approach for hand gesture recognition * An computational insight about random forest and support vector machine (SVM)

1.4 Related Work

2 System

2.1 Overview

There are three main stages to our system. First is acquiring training samples using a colored glove. Second is training a classifier on the samples. Third is live prediction and pooling.

The approach is pixel-level classification, and future pooling. The motivation for pixel-level classification is the potential of parallelizing the task and using the GPU to achieve real-time prediction.

2.2 Acquiring training samples

Color glove approach. To simplify the data acquisition step, we used cropping, depth thresholding, and a single colored glove. We used 4 gestures, giving a total of 5 classes including the background. We collected 400 training samples evenly spread across the gestures.

2.3 Training classifier

We trained the classifier on different subsets of the 400 training samples to study the effect of the training sample size. Also, we set aside 50 samples for testing purposes. Because our prediction is at the pixel level, we sample 1000 pixels from each of the training image, trying to get a good balance between the background and the gesture.

We used a depth-invariant feature vector for each pixel in our training sample. Each feature in the feature vector is the difference in the depth value of two offset pixels from the current pixel. We experimented with different number of such offset pairs for the feature vector.

We trained two classifiers - an SVM classifier for baseline testing, and a random forest classifier with multiple configurations. The random forest classifier can have multiple trees. We explored the effect of different number of trees in the forest.

2.4 Prediction

We aimed for real-time prediction of the hand gesture. The trained model from the classifier is used for per-pixel classification. That is, each pixel is classified as belonging to one of the four gestures or to the background. From here we have to first identify the likely gesture, and then find the location of that gesture, a process we called 'pooling'.

2.4.1 GPU

Running the prediction algorithm using the CPU proved to be very slow. For a 640×480 image, our system took 2.5 minutes to classify all the pixels. Given that this problem is highly parallelizable, we turned to the GPU. Re-implementing the prediction algorithm with the GPU reduced the prediction time from 2.5 minutes to 400 milliseconds - a 99.7% speedup!

2.4.2 Pooling

- mean versus median – mean is prone to outliers messing up the location. median is more resistant to that.
- majority gesture – if the hand is far away, this leads to the noisy labels being
- k-meniods – can be used to support multiple hands at once. and can be used to filter out the noise.

3 Experimental results

3.1 GPU performance enhancements

3.2 Classification algorithm experiments

4 Experience

5 Conclusion

References

[1] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, A. Blake. Real-time human pose recognition in parts from single depth images. CVPR, 2011.

[2] R. Wang and J. Popović. Real-time hand-tracking with a color glove. In Proc. ACM SIGGRAPH, 2009.

[3] I. Oikonomidis, N. Kyriazis, and A. Argyros. Efficient model-based 3D tracking of hand articulations using kinect. In BMVC, Aug 2011.

[4] V. Lepetit, P. Lagger, and P. Fua. Randomized trees for real-time keypoint recognition. In Proc. CVPR, pages 2:775-781, 2005.

[5] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification Journal of Machine Learning Research 9(2008), 1871-1874.

[6] B. Schneiderman. The eyes have it: a task by data type taxonomy for information visualizations. Visual Languages, 1996.

[7] D.A. Keim. Information visualization and visual data mining. Visualization and Computer Graphics, IEEE Transactions. Vol 8, no.1, pp. 1-8, Jan 2002.

[8] W. Stuerzlinger, C Wingrave. The value of constraints for 3D user interfaces. Dagstuhl Seminar on VR, 2010.

[9] M. Hoffman, P. Varcholik, and J. LaViola. Breaking the status quo: improving 3D gesture recognition with spatially convenient input devices. IEEE VR, 2010.

[10] V. Bystritsky. ALGLIB. 14 Aug 1999. Web. `http://www.alglib.net`.