CS 283 Final Project
Pipeline for Improving Accuracy of Hand Tracking

Kenny Yu, HUID: 30798260

Fall 2012

1 Introduction

What if you had the power of Microsoft's Kinect in your webcam? What if you could control applications simply with a swipe of your hand? Detecting and tracking hands typically requires a good detector, which requires large amounts of good data (typically at least 5000 images !!!CITE!!!) and a tremendous amount of computation time to train the classifier. In this paper, we trained our hand detector with only 250 images, generated from 10 positives and 100 negatives. Despite this poorly trained Haar cascade classifier, we provide a pipeline to improve accuracy during hand tracking, providing comparable results to tracking with a well-trained hand detector.

2 Methods

2.1 Generating our Hand Detector

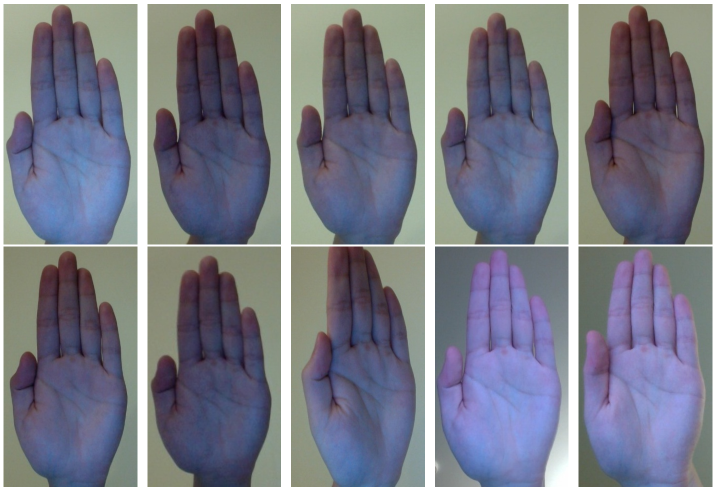

Figure 1: The 10 positive images used to train our hand detector. We generated 250 samples from these, each of size 90 pixels high and 50 pixels wide.

To generate our hand detector, we took 10 pictures of our right hand using a webcam from a Macbook Pro 2011 model, and we cropped the images to leave only our hand in the images. Our pictures consisted of our right hand with the palm facing forward and all fingers together. See Figure 1 for the the images we used as positives. We used the first 100 images from !!!!CITE!!!!! as negatives. We used OpenCV's `opencv_createsamples` !!! CITE !!!!! utility to generate 250 90 × 50 pixel samples from this set of 10 positives and 100 negatives, resulting in images where our positive images have been warped by homographies and placed randomly into our negative images. We used OpenCV's `opencv_haartraining` tool to train our Haar cascade classifier, setting our min hit rate to 0.999, our max false alarm rate to 0.5 and with 14 stages in our cascade. This took about a day to complete.

2.2 Building the Pipeline

3 Results

4 Conclusions

5 References