

CS 283 Final Project Proposal - Hand Gesture Actions

Kenny Yu, HUID: 30798260

Fall 2012

1 Motivation

Inspired by the Kinect and recent hand gesture technology (for an example, see http://news.cnet.com/8301-11386_3-57548636-76/bringing-minority-report-touchless-gestures-to-w) I want to explore how one can use a simple webcam to detect hand gestures and convert these into concrete actions. I also want to explore how to train a classifier to detect objects and explore how to use optical flow to track objects.

2 Goal

The goal of this project will be to use a webcam to detect simple hand gestures and convert these into concrete actions in an application, such as Google Maps or a drawing app. Examples of possible hand gestures include actions such as scrolling left/right/up/down, zooming in and out, rotating, and pointing. Hopefully, I will be able to detect simple motion of the hands (like in this video: http://www.youtube.com/watch?v=CvEa_92JW44) and ultimately produce something like this: <http://www.youtube.com/watch?v=U4taMDEozCs>.

3 Proposed Method

I will use OpenCV's optical flow capabilities to perform video tracking, and I will train a classifier to detect different orientations of the hand and/or multiple hands. I will most use a cascade classifier to train hands: http://docs.opencv.org/doc/tutorials/objdetect/cascade_classifier/cascade_classifier.html.

3.1 Training the Classifier

To train the classifier, I will gather data in the following way:

1. Use the Cambridge Hand Gesture Data set http://www.iis.ee.ic.ac.uk/icvl/ges_db.htm to train the initial Haar classifier. I will initially attempt to build a classifier for a closed and/or open palm.

2. Gather my own training data by taking videos in different lighting and backgrounds and annotating it, using the procedure outlined here:

(a) <http://www.21csi.com/?q=blog/buildingtrainingdata>

(b) <http://note.sonots.com/SciSoftware/haartraining.html>

(c) <http://www.andol.info/hci/1991.htm>. However, this post suggests the the last technique does not work very well: <http://cwyalpha.wordpress.com/2012/05/07/thought-this-was-cool-a-failed-example-of-hand-gesture-recognition-using-op>
I will explore this.

3. Once I have the training set, I will build my own Haar classifier for a closed and/or open palm, and compare the results from training it with just the Cambridge Hand Gesture Data set. If this works well, I will attempt to train other hand positions too (e.g. one finger up, a first).

3.2 Using Optical Flow to Track the Hand

Once I am able to detect a hand gesture, I will use this to find better salient points to track during optical flow. I will use this process:

1. Detect the hand (e.g. in some fixed position such as an open palm).
2. Draw a bounding box around the hand, and detect the salient points within this box.
3. Use optical flow (http://docs.opencv.org/modules/video/doc/motion_analysis_and_object_tracking.html) to track these salient points and compute a final velocity vector to indicate the overall motion of the hand. Use this vector to actualize a hand gesture (e.g. a left pointing vector would correspond to an application action of scrolling left x amount).

I will experiment with several techniques that may improve recognition, such as background subtraction: <http://www.cse.ohio-state.edu/~busaryev/Projects/Gesture%20Recognition%20with%20Applications/Report.pdf>

3.3 Actualizing the Hand Gesture

Once I am able to accurately track the position and orientation of the hands, I will use Chrome's webRTC (<http://www.webrtc.org/>) capabilities to send a video stream from the client to a server that performs the video tracking and hand detection, and then convert this back into concrete actions in the client. A example of such a server-client setup is here: <http://www.smartjava.org/content/face-detection-using-html5-javascript-webrtc-websockets-jet>
An example of an application using hand gestures would be navigating Google Maps or drawing an image.

4 Plan for Evaluation

I will evaluate my project based on these guidelines:

1. Accuracy of the hand/finger classifier, subject to different orientations, lighting, background, and other noise factors of the Cambridge data set vs. my own generated data set.
2. Number of images required to train the classifier.
3. Number of different hand gestures that can be detected.
4. Accuracy of tracking hand detection + optical flow vs. simply tracking using salient points of the entire frame without any notion of hands.
5. How well these different hand gestures translate into concrete actions in some application.