Hand Gesture Recognition

Name: Ziran Min
Email: minziran@bu.edu
Date: 04/02/2018

Problem Definition

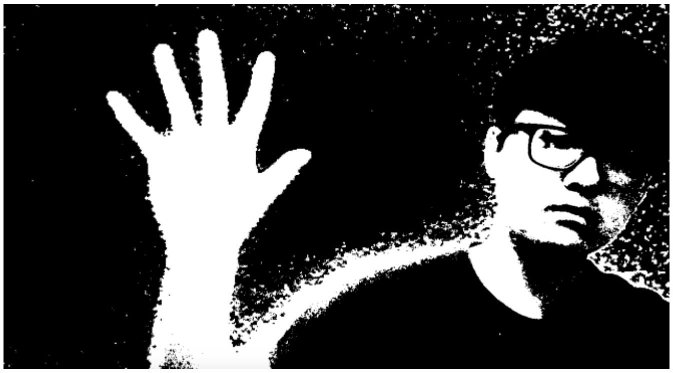
In this assignment, I will design and implement algorithms that recognize 3 static hand shapes ("High Five", "V Yeah", and "Fist") and 1 dynamic hand gesture (waving horizontally). In my system, I use OpenCV library functions and several computer vision techniques that were discussed in lab/class, such as background differencing, motion energy templates, and skin-color detection. Furthermore, I also use some functions to find the contour, convex hull and convex defect of a hand to count finger tips.

Method and Implementation

Part 1: Static Hand Shapes Recognition

The three hand shapes my system will recognize are "High Five", "V Yeah", and "Fist", or you can think they are the three gestures from "Rock, Paper, Scissors". The goal is to detect a hand through camera and distinguish shapes by counting the number of fingers. Then main function I implement are the following:

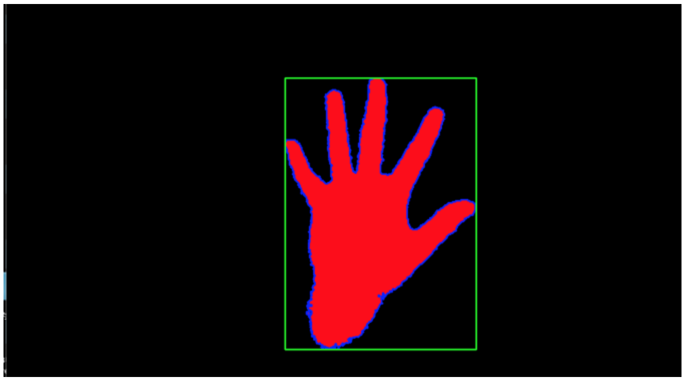
1. **mySkinDetect** Function: It turns a normal frame into binary image where pixels that belongs to the skin color based on RGB values become white and all other pixels become black.



2. **findContours** Function: Its find the contours in the binary image.

3. **contourArea** Function: Its calculates the area of the each contour found in the binary image. Then I only focus on the contour that has the largest area and I regard that as a hand. (This means I have to put hand close enough to the camera to make it has the largest contour).

4. **drawContours** Function & **rectangle** Function: They better visualize the hand found the in the window as shown by the following:



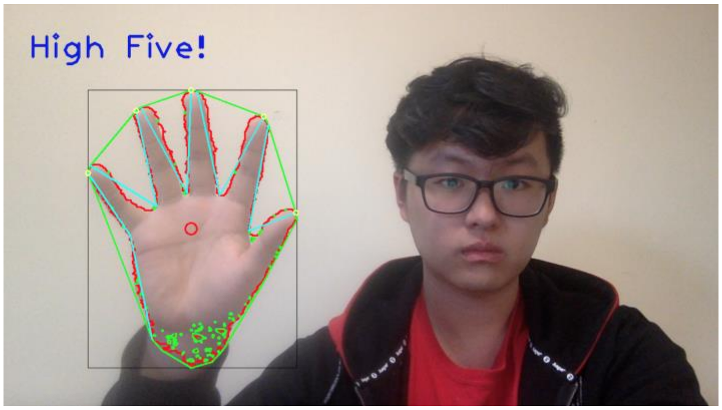
5. **convexHull** Function & **convexityDefects** Function: Theses two function helps me to locate the convex hulls and defects of a hand and then later I can calculate the distance between a hull vertex and a defect valley point. By comparing those distance, I can further count number of fingers.

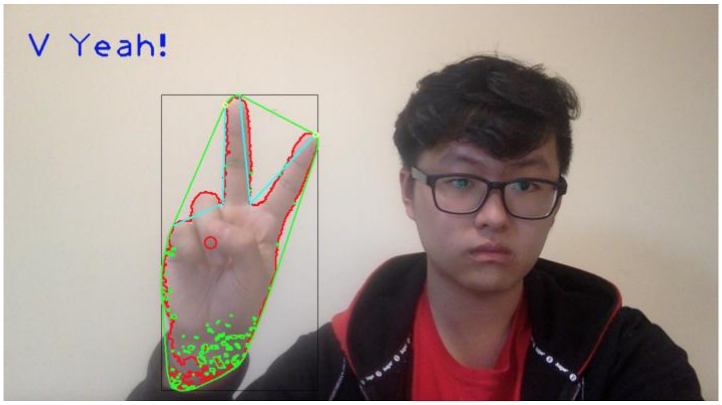
6. **minEnclosingCircle** Function: It turns out the smallest enclosing circle of a hand. Then further I can find the center of a hand.

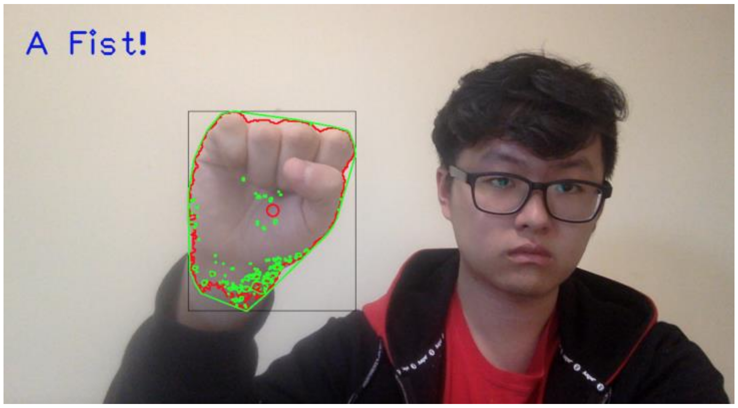
7. Now we can find fingers by finding all hull vertices that are above the center of a hand and finding all pairs of hull vertex and a defect valley point that has distance greater than a specific value (I use 15 to be the threshold after trying many numbers).

8. Then I use **line** and **circle** functions to visualize those qualified points and lines, and make a simple skeleton of a specific hand shape.

9. A "High Five" should have 5 fingers, a "V Yeah" should have 3 fingers, and a "Fist" should have 0 fingers when there is a hand contour is detected by the camera.





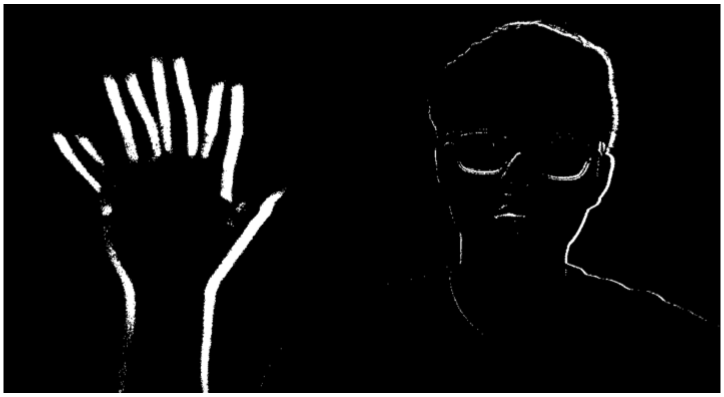


Part 2: Dynamic Hand Gesture Recognition

In the second part, I want my system can recognize that my hand is waving horizontally in front of the camera. Then main function I implement are the following:

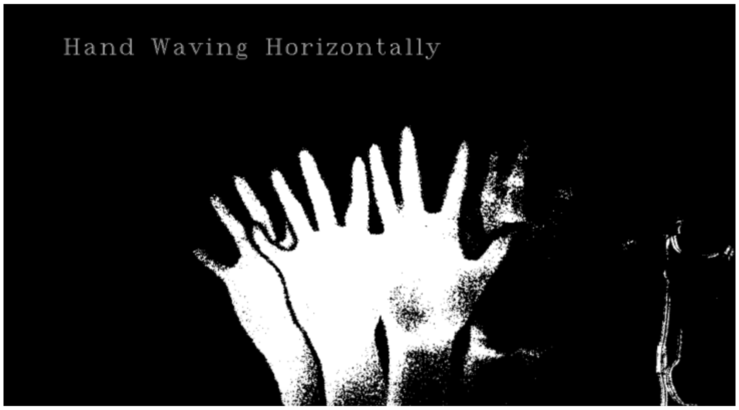
1. **mySkinDetect** Function: Same as part 1, it turns a normal frame into binary image where pixels that belongs to the skin color based on RGB values become white and all other pixels become black.

2. **myFrameDifferencing** Function: Based on the binary image produced by **mySkinDetect** function, this **myFrameDifferencing** function does frame differencing between the current frame and the previous frame, or in other worlds, it make pixels white if the corresponding pixel intensities in the current and previous image are not the same. If I sit in front of the camera and make a small move, the image looks like the following:



   If I don't move at all in front of the camera, the frame image will be whole black.

3. **myMotionEnergy** Function: Based on **myFrameDifferencing** function, this one accumulates the frame differences for a certain number of pairs of frames. This why you can see the previous images overlap in one frame ("movement lagging") in the following image. Furthermore, to detect whether a hand is moving horizontally, I built a nasty loop to count the accumulated horizontal frame differences. When the accumulated differences break a threshold (I set it as 350), a hand waving horizontally will be detected.



Result: Confusion Matrices

	"High Five" (actual)	"V Yeah" (actual)	"A Fist" (actual)	None (actual)
"High Five" (predicted)	44	5	2	1
"V Yeah" (predicted)	2	43	6	1
"A Fist" (predicted)	0	0	40	3
None (predicted)	4	2	2	45

	Waving Horizontally (actual)	Not Waving Horizontally (actual)
Waving Horizontally (predicted)	44	3
Not Waving Horizontally (predicted)	2	1

Note: For the confusion matrix of three static hand shape, I add an element “None”. “None (actual)” means I don’t show any of three hand shapes but only show my face or nothing in front of the camera. “None (predicted)” will counted as the moment that none of the three hand shapes be detected.

The accuracy for my three static hand shapes system is 0.86, and the accuracy for my dynamic gesture system is 0.9.

Conclusion

In all, my system has pretty good accuracy for recognizing both static and dynamic hand gestures. However, the method of counting number of fingers may not be the best way to detect hand shapes. The “V Yeah” only means you show your index finger and middle finger. However, according to my system, showing any two finger could possibly be detected as a “V Yeah” shape. I also need more threshold or criteria to distinguish a finger. Sometimes when I show a “High Five”, my thumb is below the center of my hand, then it won’t detect five fingers. Maybe considering the angle between fingers is another way.

Furthermore, for detecting dynamic movement, my program is not perfect. Since I only accumulate the frame difference, not matter my hand waving horizontally or my head moving horizontally, it will show “Hand waving horizontally”. So instead of counting the frame differences, I also need to keep tracking the status of gesture.

There are more I can improve in the future, but this project is just the start for me to explore computer vision.