Assignment #2 - OpenCV Two Part Assignment

Due February 24th @ 7pm (VIDEO demo with explanation linked today's lecture notes)

Two part assignment.....using Python and OpenCV you need to do the following two parts (Should be two different applications).

60 points total, 25 for color, 35 for motion

Part One:

Tracking an object (25 points)

Steps to do it properly.....

1. Capture live video from Webcam
2. Display live video
3. convert BGR to HSV
4. Display the HSV video (5 points)
5. Click on HSV video and capture the HSV values at the location clicked, and a few other local values of the item you are going to track. (3 points)
    1. use a MouseCallback to print the HSV values. (Shown on Monday)
6. Using Sliders create scalers for the min and max values you want to track (Shown on Monday) (10 points)
    1. a Scalar will be a numpy array (np.array) that takes 3 values for minH, minS, and minV.....then a second scalar to catch the other three Max values
    2. create 6 trackbars, createTrackbar with callback methods to set your six variables
7. Us the OpenCV inRange method to find the values between the scalars from HSV image and the result will go to a grayscale image (make it a binary image, white/black).
8. Dilate, erode the grayscale image to get a better representation of the object you are tracking. (5 points)
9. Display the original image and the binary image where everything is black except for the object you are tracking. The tracked object will be white. (

Points Breakdown (25 points)

- 3 points Show a live video feed from a camera
- 3 points convert to HSV and Display
- 4 points click on an object in the HSV image to print out HSV values at mouse click location (Red is always the easiest value to track)
- 4 points Sliders to control the min and max Scalar track for each HSV channel
- 3 points Dilate and Erode the image for better tracking
- 8 points Display the original video feed and the binary image that shows the object being tracked working.

Done Part One....

Part Two

Detect Motion (35 Points)

The Steps I took in the example I showed in class.

Video Explanation (this shows it working in Visual Studio, but we're using Python, I'll show that in class.

1. Capture an image
2. Create blank images:
    1. grayscale image with proper dimensions
    2. 32f, 3-channel image
    3. A capture clone we'll call image1
    4. One to hold the result from the absDiff function. (absolute difference)
3. while loop to capture image after image.....
4. grab new frame
    1. I brightened image a bit first which helped
5. blur the image
6. take running average of frame: **accumulateWeighted**
7. swap running average result from step 6 to same bits as frame: **convertScaleAbs**
8. Take difference, built in OpenCV function to do a diff between two images.
9. convert to grayscale
10. Threshhold grayscale (using a low number)
11. Blur grayscale image
12. Threshhold grayscale again (using a high number)
13. find contours: **findContours**
14. Use contours to find significant blobs,
15. draw polygons of blobs
16. use dimensions of blobs to draw bounding boxes and center on original image

Points Breakdown (35 points)

- (3 points) Create 4 images in step 2 properly
- (3 points) Steps 4, 5 brighten and blur
- (5 points) Taking the accumulated weight image and converting it from 32bits back to what is needed
- (3 points) Steps 8, 9,Take difference, turn to grayscale, display that image
- (5 points) Steps 10,-12 take a threshold, blur and threshhold again, then display that image
- (6 points Steps 13, 14, 15 doing the contours and displaying them on a white image
- (10 points) Step 16 find a suitable cutoff for the amount of points in a contour and draw a box in the original image showing the movement