Tutorial - Colored object tracking using HSV

e-Yantra Team

June 7, 2016

Contents

1 Colored object tracking using HSV

The objective of this tutorial is to track a colored object. We track the object whose color lies in the specified range of HSV color space values using largest contour inside the frame.

2 Prerequisites

User should have handy knowledge of following before reading this tutorial.

- Basics of Python Language.
- Introduction to OpenCV.
- Basics of Image processing in OpenCV using Python.
- Knowledge on BGR and HSV color space and interconversion.

3 Hardware Requirement

- A Computer with internal or external webcam.

4 Software Requirement

- Python 2.7.5 (with OpenCV and Numpy module)
- OpenCV 2.4.9
- numpy 1.7.1
- **Note :** These versions I had at the time of this tutorial.

5 Theory and Description

- We are tracking specific colored object. As every pixel in the frame (image) has different RGB values. So we mask the image for the detection of object we want . It means we make those pixels white which lies in the specified range and makes other pixels black.
- We are using HSV Colorspace instead of RGB Colorspace because only H (hue) value is used for color and S for Saturation and V for Brightness. So it is easy to detect color using HSV values.
- So after getting black and white frame, we find contours in the frame. Then we calculate the area of all bounded countours in the frame. Among all the bounded contours,the contour with the maximum area will be our Object.

- Now we generate a red rectangle bounding the object.
- Same process works in each frame and we track the colored object.
- **Note :** This code works only, when your object (which you want to track) is the largest area object of its color inside the frame. This is the limitation of this code.

6 Experiment

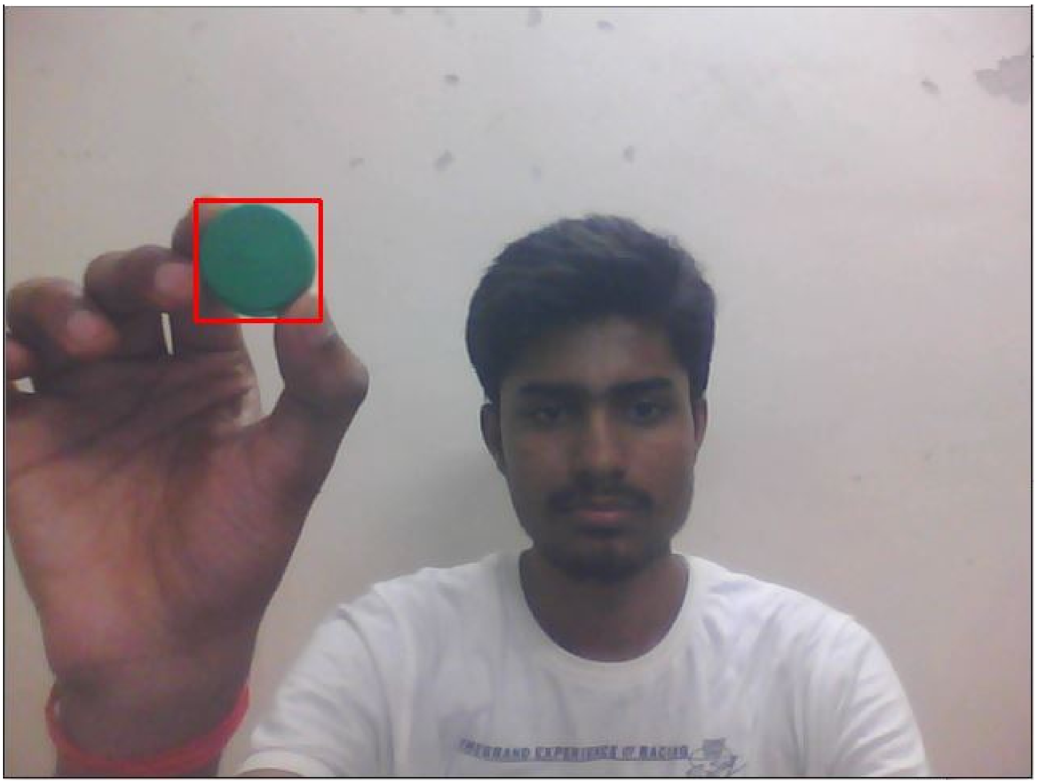
The Python code using OpenCV and Numpy is given below.

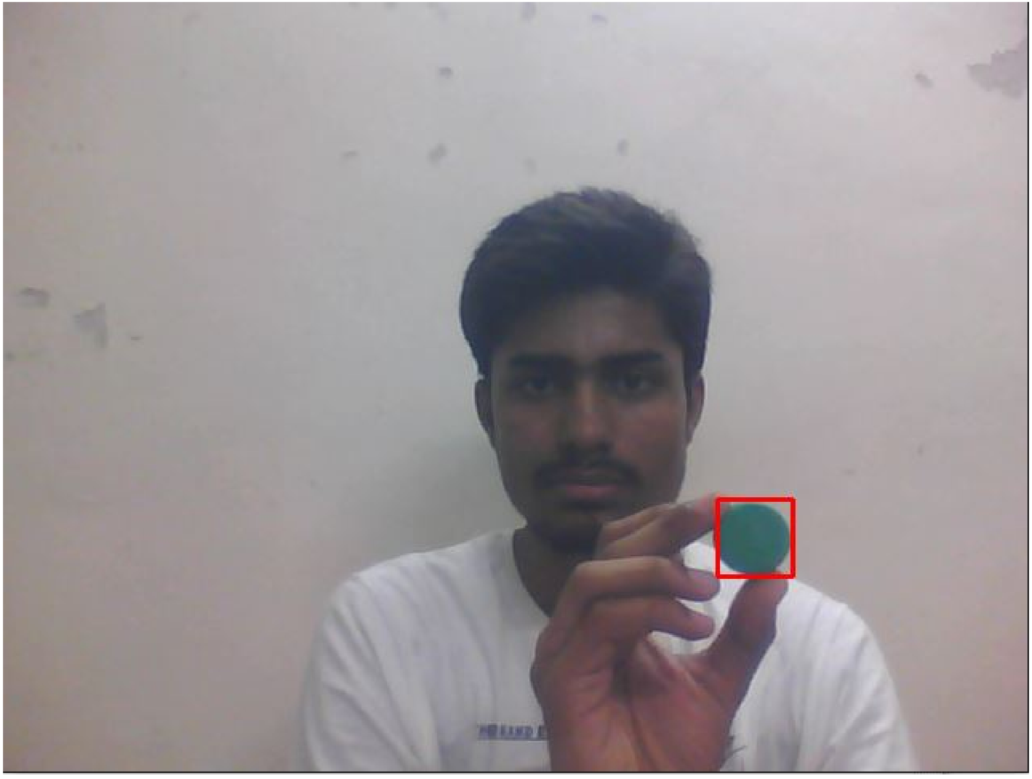
```
##########################################
## Import numpy for numerical calculations
import numpy as np

## Import OpenCV for image processing
import cv2

##########################################
## Initialize webcam
cap = cv2.VideoCapture(0) ##use 1 in parameter instead of 0 for external
                          ##camera
## You can also use a video by giving url of video in the parameter
## If video is on the same location use only name of the video as
#cap = cv2.VideoCapture('sample.mov')
##If you are using video then comment previous statement

##########################################
## param1 and param2 are minimum and maximum range of hsv values for
## green color
param1 = [50,50,50]       ## [H_min,S_min,V_min]
param2 = [90,255,255]     ## [H_max,S_max,V_max]
## You can put range of any color and track that object

##########################################
## np.array will change param1 and param2 into numpy array which
## OpenCV can understand
lower = np.array(param1)
upper = np.array(param2)

##########################################
## Video Loop
```

```
while(1):

    ## Read the frame of video
    ## frame contains frame of the video
    ## ret contains True if frame is successfully read otherwise it
    ## contains False
    ret, frame = cap.read()

    ## If frame is successfully read
    if(ret):

        ## This statement changes color space of frame from BGR to HSV
        # and stores frame into array hsv
        hsv = cv2.cvtColor(frame,cv2.COLOR_BGR2HSV)

        ## This statement changes all the pixels of the HSV frame into
        # white pixel which lies in the specified range
        ## And changes others into black pixels.
        ## And stores the new frame into mask
        mask  = cv2.inRange(hsv, lower, upper)

        ## This statement removes noise from the Masked Frame (mask)
        mask = cv2.GaussianBlur(mask,(5,5),0)

        ## This statement finds contours (white areas) in mask
        ## And returns all the contours in contours array
        contours, hierarchy = cv2.findContours(mask,cv2.RETR_TREE,cv2.CHA

        ## Let area of largest contour is zero
        max_contour_area=0

        ############################################
        ## Colored object tracking

        ## If there is specified colored object in the frame then there
        #will be atleast one contour

        ## If length of contours is atleast 1 then only it will find
        #index of largest contour

        ## And track the object
        if(len(contours) >= 1):
            ## Finding index of largest contour among all the contours
            #for color object tracking
            for i in range(0,len(contours)):
                if(cv2.contourArea(contours[i]) > max_contour_area):
                    max_contour_area = cv2.contourArea(contours[i])
                    max_contour_area_index = i

            ## This statement gives co-ordinates of North-West corner
            #in x and y
            ## And Width and Height in w and h of bounding rectangle
            #of Colored object
            ## If and only if your frame has largest thing of specified
            #color as your object which you want to track
            x,y,w,h=cv2.boundingRect(contours[max_contour_area_index])

            ## This statement will create rectangle around the object
            #which you want to track
            cv2.rectangle(frame, (x, y), (x+w, y+h), (0,0,255), 2)

        ## Showing the video which object tracking
        cv2.imshow('video',frame)

        ## Your video will stop if you press Escape (esc) key
        if cv2.waitKey(60) == 27:  ## 27 - ASCII for escape key
            break

    ## If frame is not successfully read or there is no frame to read
    #(in case of recorded video) stop video
    else:
        break

## Releasing camera
cap.release()

## Destroy all open windows
cv2.destroyAllWindows()
```

7 Exercise

Real time colored object tracking using webcam is shown below.





8 References

1. https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_gui/py_drawing_functions/py_drawing_functions.html#drawing-functions

2. https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_colorspaces/py_colorspaces.html#converting-colorspaces

3. https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_filtering/py_filtering.html#filtering

4. https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_contours/py_contours_begin/py_contours_begin.html#contours-getting-started

5. https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_contours/py_contour_features/py_contour_features.html#contour-features