

Project Report 11

Mean-Shift: Single Object Tracking in Images

NAME: DHIVAKAR.R

COURSE: AI and ML

Question:

Using OpenCV implement a single object tracker. Steps to be implemented: a) Use a pre-recorded video or your webcam to have a video Capture object. b) Mark the region of interest (ROI or the object you want to track) using its coordinates in the first frame. c) Calculate the histogram of the ROI. d) Iteratively calculate the histogram at each location (using cv2.calcBackProject) and then apply mean shift to get the updated location of the ROI.

Prerequisites

What things you need to install the software and how to install them:

Python 3.6 This setup requires that your machine has latest version of python. The following url <https://www.python.org/downloads/> can be referred to download python. Once you have python downloaded and installed, you will need to setup PATH variables (if you want to run python program directly, detail instructions are below in how to run software section). To do that check this: <https://www.pythoncentral.io/add-python-to-path-python-is-not-recognized-as-an-internal-or-external-command/> . Setting up PATH variable is optional as you can also run program without it and more instructions are given below on this topic.

Second and easier option is to download anaconda and use its anaconda prompt to run the commands. To install anaconda check this url <https://www.anaconda.com/download/> You will also need to download and install below 3 packages after you install either python or anaconda from the steps above Sklearn (scikit-learn) numpy scipy if you have chosen to install python 3.6

Importing all the required libraries

```
[2]: import cv2
import numpy as np
import os
import matplotlib.pyplot as plt
import time
from IPython.display import clear_output
```

Code and output

```
[4]: video = cv2.VideoCapture(0)
_, first_frame = video.read()
# cv2.imshow("frame1", first_frame)
# cv2.waitKey(0)
print(first_frame.shape)

x = 150
y = 150
width = 200
height = 200
roi = first_frame[y:y+height, x:x+width]
# cv2.imshow("ROI", roi)
# cv2.waitKey(0)

hsv_roi = cv2.cvtColor(roi, cv2.COLOR_BGR2HSV)
roi_hist = cv2.calcHist([hsv_roi], [0], None, [256], [0, 256])
roi_hist = cv2.normalize(roi_hist, roi_hist, 0, 255, cv2.NORM_MINMAX)

term_criteria = (cv2.TERM_CRITERIA_EPS | cv2.TERM_CRITERIA_COUNT, 10, 1)

fig = plt.figure()
while True:
    _, frame = video.read()
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    mask = cv2.calcBackProject([hsv], [0], roi_hist, [0, 360], 1)

    _, track_window = cv2.meanShift(mask, (x, y, width, height), term_criteria)
    x, y, w, h = track_window
    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

    cv2.imshow("Mask", mask)

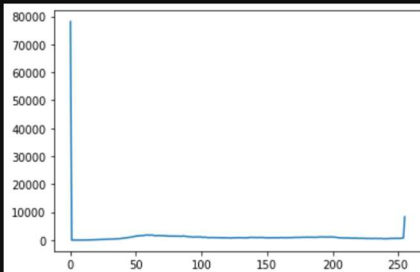
    cv2.imshow("Frame", frame)

    clear_output(wait=True)
    histr = cv2.calcHist([frame], [0], None, [256], [0, 256])

    plt.plot(histr)
    plt.show()

    key = cv2.waitKey(30)
    if key == 113: #esc
        video.release()
        cv2.destroyAllWindows()

        cv2.waitKey(1)
        break
```



Output

