

Extended tutorial : KNN background segmenter for moving target recognition

We choose KNN as the background selector for this use.

```
bs = cv2.createBackgroundSubtractorKNN(detectShadows=True)
```

In the lighting environment, the object will produce a shadow, but in fact the shadow is not part of the moving target. `detectShadows=True` means to calculate the shadow. By calculating the shadow, the shadow area in the image can be excluded to improve the accuracy.

At the core of this method is the `.apply()`, which returns a foreground mask to distinguish which objects are the background and which objects are the targets. Once the target is obtained, the outline of the target is detected.

The source code of the program is located at: [/yahboom/motion_detect](#)

!!!Note: Please try the actual code below on the PC.

```
fgmask = bs.apply(frame)
th = cv2.threshold(fgmask.copy(), 244, 255, cv2.THRESH_BINARY)[1]
th = cv2.erode(th, cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (3, 3)), iterations=2)
dilated = cv2.dilate(th, cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (8, 3)),
iterations=2)
image, contours, hier = cv2.findContours(dilated, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
```

```

1#!/usr/bin/env python3
2# -*- coding: utf-8 -*-
3"""
4Created on Wed Dec 26 16:32:49 2018
5
6@author: fendicloser
7"""
8
9import cv2
10import numpy as np
11
12
13
14bs=cv2.createBackgroundSubtractorKNN(detectShadows=True)
15camera=cv2.VideoCapture(0)
16camera.set(3,320)
17camera.set(4,160)
18ret,frame=camera.read()
19
20while True:
21    ret,frame=camera.read()
22    fgmask=bs.apply(frame)
23    th=cv2.threshold(fgmask.copy(),244,255,cv2.THRESH_BINARY)[1]
24    th=cv2.erode(th,cv2.getStructuringElement(cv2.MORPH_ELLIPSE,(3,3)),iterations=2)
25
26    dilated=cv2.dilate(th,cv2.getStructuringElement(cv2.MORPH_ELLIPSE,(3,3)),iterations=2)
27
28    image,contours,hier=cv2.findContours(dilated,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)
29
30    for c in contours:
31        if cv2.contourArea(c)>3000:
32            (x,y,w,h)=cv2.boundingRect(c)
33            cv2.rectangle(frame,(x,y),(x+w,y+h),(255,255,0),2)
34            if x!=0 and y!=0:
35                print('x',x,'y',y)
36
37    cv2.imshow("mog",fgmask)
38    cv2.imshow("detection",frame)
39    if (cv2.waitKey(30)&0xFF)==27:
40        break
41    if (cv2.waitKey(30)&0xFF)==ord('q'):
42        break
43 camera.release()
44 cv2.destroyAllWindows()

```

The running effect is as follows, the figure1-1 is the real scene, and the figure1-2 is the analysis of the background.

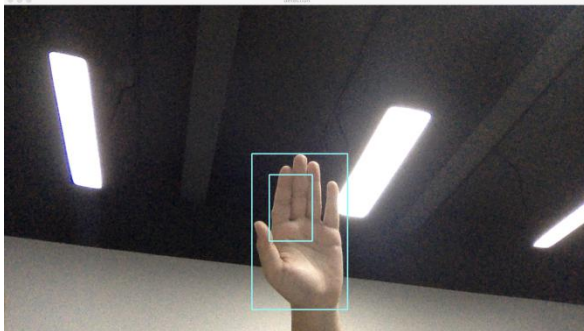


figure1-1



figure1-2