

Object tracking

Object tracking

- 1.Introduction
- 2.Introduction to Object Tracking
3. Code analysis
4. Output result

Another method--Object tracking

- 1.Advantages and disadvantages of the tracker (without any data training)
- 2.Analysis code
- 3.Experimental results

website: <https://learnopencv.com/object-tracking-using-opencv-cpp-python/#opencv-tracking-api>

1.Introduction

OpenCV is a cross-platform computer vision and machine learning software library released under the BSD license (open source), which can run on Linux, Windows, Android and MacOS operating systems. [1] It is lightweight and efficient-consists of a series of C functions and a small number of C++ classes, and provides interfaces for languages such as Python, Ruby, and MATLAB, and implements many general-purpose algorithms in image processing and computer vision.

2.Introduction to Object Tracking

What is object tracking?

Simply put, locating an object in successive frames of a video is called tracking.

In our program, we want to track a blue object. Here are the few steps we have to do:

- Get every frame from the video
- Convert image to HSV space
- Set the HSV Threshold to the blue range.
- Get the blue object, of course we can do anything else we want, eg: draw a circle around the blue object

3. Code analysis

Code path: ~/astra_ws/src/astra_visual/foods_track

This document takes tracking blue objects as an example. If you want to track other color objects, please read the HSV color table.txt in the directory and modify the value of the code below.

```
lower_red=np.array([100, 43, 46])
upper_red=np.array([124, 255, 255])
```

Next, analysis food_track_base_color.py file

```
cap=cv2.VideoCapture(8)
```

The following is the code to open the RDK-X3 camera.

```
while cap.isOpened():
# 获取每一帧
    ret,frame=cap.read()
    frame_ = cv2.GaussianBlur(frame,(5,5),0)
# 转换到 HSV
    hsv=cv2.cvtColor(frame,cv2.COLOR_BGR2HSV)
# 设定颜色的阈值 - 蓝色
    lower_red=np.array([100, 43, 46])
    upper_red=np.array([124, 255, 255])
# 根据阈值构建掩模
    mask=cv2.inRange(hsv,lower_red,upper_red)
    mask = cv2.erode(mask,None,iterations=2)
    mask = cv2.dilate(mask,None,iterations=2)
    mask = cv2.GaussianBlur(mask,(3,3),0)
# 寻找轮廓
    cnts = cv2.findContours(mask.copy(),cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)[-2]
    if len(cnts) > 0:
        cnt = max (cnts, key = cv2.contourArea)
        (color_x,color_y),color_radius = cv2.minEnclosingCircle(cnt)
        if color_radius > 10:
            cv2.circle(frame,(int(color_x),int(color_y)),int(color_radius),(255,0,255),2)
# 对原图像和掩模进行位运算
    res=cv2.bitwise_and(frame,frame,mask=mask)
# 显示图像
    cv2.imshow('frame',frame)
    #cv2.imshow('mask',mask)
    #cv2.imshow('res',res)
    k=cv2.waitKey(5)&0xFF
    if k==27:
        break
```

The code in this part is: it will run only when the camera is successfully opened.

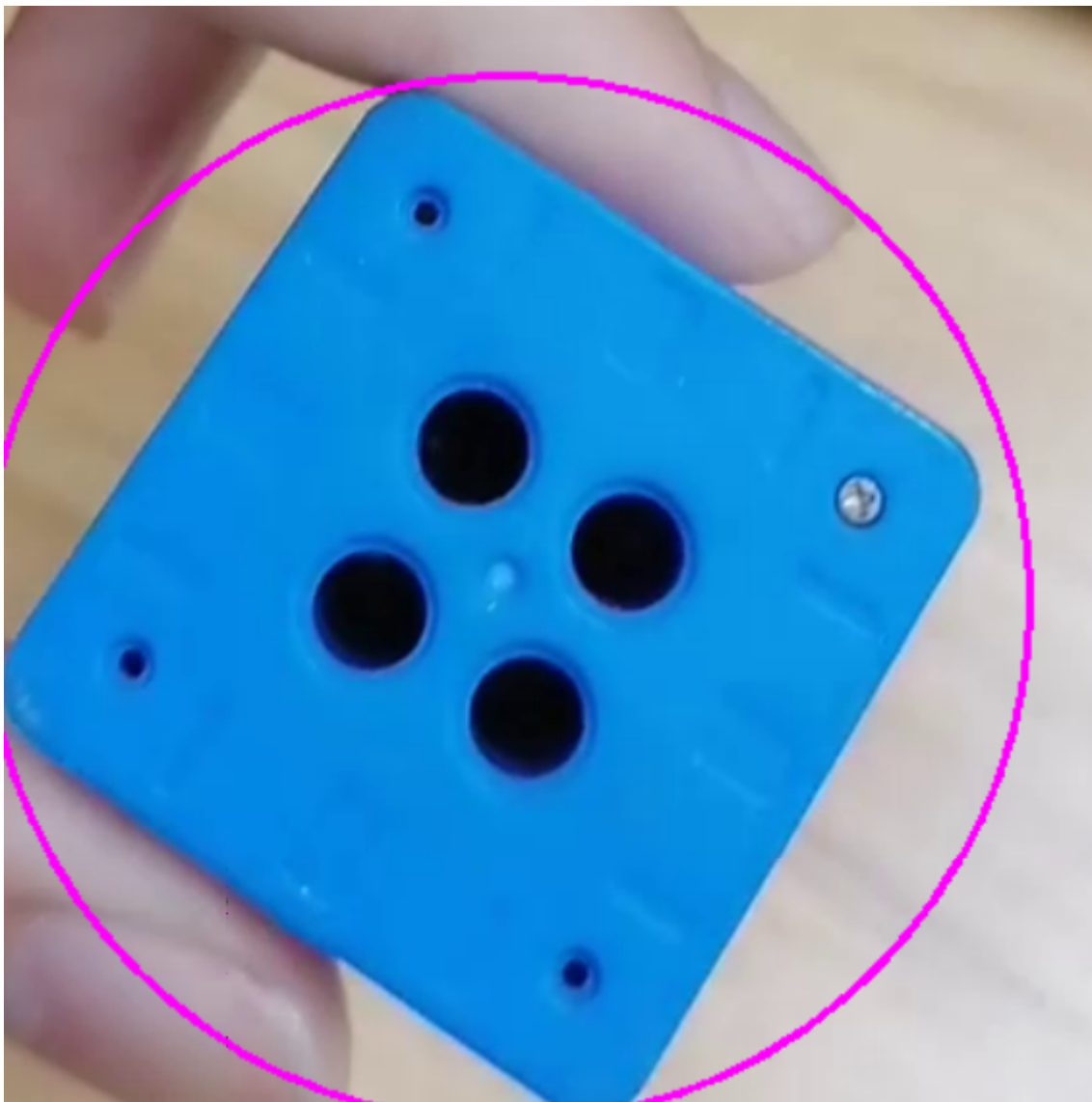
Track the blue object on the screen display of the camera, press the "ESC key" on the keyboard, and the program ends!

4. Output result

After writing the code, click Save, and enter `python3 foods_track_base_color.py` in the terminal.

You can clearly see the picture displayed by the camera, and track the blue object in the picture. The effect is as follows.





Another method--Object tracking

This method needs to install the expansion package of opencv

```
pip3 install opencv-contrib-python
```

This method is to use the tracker of the expansion pack for object tracking

1.Advantages and disadvantages of the tracker (without any data training)

Advantages: It will not be disturbed by colors, and it can track objects.

Disadvantage: Once an object is blocked by another object, the program has a high probability of tracking the new object.

2.Analysis code

The code is in the extension folder in this directory foods_track_extread.py

```
cap = cv2.VideoCapture(8)
#tracker = cv2.legacy.TrackerMOSSE_create()
tracker = cv2.legacy.TrackerCSRT_create()
success,img = cap.read()
bbox = cv2.selectROI("Tracking",img,False)
```

This part of the code is used to open the camera and wait for the user to select an object on the displayed screen. After the selection is completed, press the Enter key.

```
while True:
    timer = cv2.getTickFrequency()
    print(timer)
    success,img = cap.read()
    success,bbox = tracker.update(img)
    print(bbox)
    if success:
        drawBox(img,bbox)
    else:
        cv2.putText(img,"Lost", (75, 75), cv2.FONT_HERSHEY_COMPLEX, 0.7, (0, 0, 255), 2)
        fps = cv2.getTickFrequency()/(cv2.getTickCount()-timer)
        cv2.putText(img,str(int(fps)),(75,50),cv2.FONT_HERSHEY_COMPLEX,0.7,(0,0,255),2)
        cv2.imshow("1",img)

    if cv2.waitKey(1)&0xff == 27:
        break
```

Then, the tracker will initialize and track the object selected by the user in real time. Press the "esc" key to end the program!

3.Experimental results



Start tracking



**After the user draws the frame line,
press the Enter key**



Object tracking

