

Chapter4: Color space method tracking color (HSV tracking yellow)

1.1 Extract object color

We can extract objects of a specific color and track the object within the field of view of the camera to print the center coordinates of the object in real time. The steps are as follows:

Get video stream

Color space conversion RGB→HSV Set the threshold of HSV

Identify and track objects

The source code of the program is located [/home/pi/yahboom/find_hsv/find_hsv.py](#)

The program is shown below:

```
#!/usr/bin/env python2
# -*- coding: utf-8 -*-
"""
    * @par Copyright (C): 2010-2019, Shenzhen Yahboom Tech
    * @file      face_tracking
    * @version    V1.0
    * @details
    * @par History
    * @author: longfusun
"""

import numpy as np
import cv2

yellow_lower=np.array([9,135,231])
yellow_upper=np.array([31,255,255])
cap=cv2.VideoCapture(0)

cap.set(3,320)
cap.set(4,240)

while 1:
    ret, frame=cap.read()
    frame=cv2.GaussianBlur(frame,(5,5),0)
    hsv=cv2.cvtColor(frame,cv2.COLOR_BGR2HSV)
    mask=cv2.inRange(hsv,yellow_lower,yellow_upper)
    #Image expansion swelling
    mask=cv2.erode(mask,None,iterations=2)
    mask=cv2.GaussianBlur(mask,(3,3),0)
    res=cv2.bitwise_and(frame,frame,mask=mask)
    #Find outlines and draw outlines
    cnts=cv2.findContours(mask.copy(),cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)[-2]

    if len(cnts)>0:
        #Find the largest area and draw its smallest circumscribed circle
        cnt=max(cnts,key=cv2.contourArea)
        (x,y),radius=cv2.minEnclosingCircle(cnt)
        cv2.circle(frame,(int(x),int(y)),int(radius),(255,0,255),2)
        #Find the position coordinates of the object
        print(int(x),int(y))

    else:
        pass
    cv2.imshow('frame',frame)
    cv2.imshow('mask',mask)
    cv2.imshow('res',res)
    if cv2.waitKey(5)&0xFF==27:
        break
cap.release()
cv2.destroyAllWindows()
```

Execution effect: There is noise in the image, and the corresponding filter needs to be added for filtering. The program accurately circles the yellow ball in the field of the camera.

1.2 OpenCV to achieve object tracking

1.2.1 Identifying object colors

In this lesson, we learn the use of OpenCV to identify the color of an object and use speech synthesis technology to perform speech output on the color of the recognized object.

How to use OpenCV to identify colors?

There are two common schemes:

1. Use color histogram to count the range of HSV color space of H.
2. In the HSV color space, traverse the statistical outline of HSV values of common colors. We often use color histograms for color recognition.

The color histogram can count the number of pixels in each pixel value in the image, and the histogram is the image representation of the pixel intensity distribution in the image.

In this lesson we experiment with histogram statistics.

The source code of the program is located

/home/pi/yahboom/pixel_number/pixel_number.py

The program is shown below:

```
#!/usr/bin/env python2
# -*- coding: utf-8 -*-
"""
    * @par Copyright (C): 2010-2019, Shenzhen Yahboom Tech
    * @file      Histogram
    * @version    V1.0
    * @details
    * @par History

    @author: longfuSun
"""

import cv2
import numpy as np

from scipy.misc import imread
from matplotlib import pyplot as plt

img=cv2.imread('tankCar.jpg',0)
img=imresize(img,(240,320))
#Histogram calculation function, channel 0, no mask used
hist=cv2.calcHist([img],[0],None,[256],[0,256])

hist_max=np.where(hist==np.max(hist))
print(hist_max[0])

cv2.imshow('image',img)

plt.plot(hist)
plt.xlim([0,256])
plt.show()
cv2.waitKey(0)
cv2.destroyAllWindows()
```

The result is as shown in the figure below. The histogram only shows the statistical information, and does not include the position information of the image pixel.

