

코드론 II 영상처리 자율주행 전략

지도교수님 : 장정환

동아대학교 LAMBDA 2.0

전기공학과 장지원
전기공학과 권동욱
전기공학과 이예준

목차

1. 영상처리

- Gaussian Blur
- HSV 추출 , 영상 이진화
- Frame cutting
- Contour
- Contour Box 를 이용한 좌표 추출
- Math 라이브러리를 이용한 각도 추출
- OpenCV를 이용한 정보 표시
- Raspberry Pi zero 적용 시 고려사항

2. 동작 알고리즘도

Gaussian Blur

OpenCV 의 `cv2.GaussianBlur` 를 이용해 의도적으로 영상을 흐릿하게 만들어 색을 추출하는 데 더욱 용이하게 함

```
image.py - C:\Users\WLYJ\Desktop\Python\opencv\성공한 예제\image.p...
File Edit Format Run Options Window Help
import cv2

image = cv2.imread("c:\python\project.jpg", cv2.IMREAD_ANYCOLOR)
image = cv2.pyrDown(image);

result = cv2.GaussianBlur(image,(5,5),0)

cv2.imshow("image", image)
cv2.imshow("GaussianBlur", result)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



HSV 추출 , 영상 이진화

OpenCV 라이브러리의 cv2.cvtColor 와 Trackbar를 이용한 HSV 추출기 설계

```

*image.py - C:\Users\WLYJ\Desktop\Python\opencv\성공한 예제Wi...
File Edit Format Run Options Window Help
import cv2
import numpy as np

def nothing(x):
    pass

cv2.namedWindow("Trackbars")

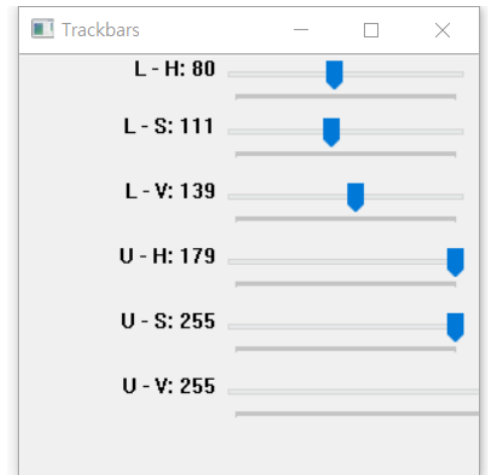
cv2.createTrackbar("L - H", "Trackbars", 0, 179, nothing)
cv2.createTrackbar("L - S", "Trackbars", 0, 255, nothing)
cv2.createTrackbar("L - V", "Trackbars", 0, 255, nothing)
cv2.createTrackbar("U - H", "Trackbars", 179, 179, nothing)
cv2.createTrackbar("U - S", "Trackbars", 255, 255, nothing)
cv2.createTrackbar("U - V", "Trackbars", 255, 255, nothing)

image = cv2.imread("c:\python\project.jpg", cv2.IMREAD_ANYCOLOR)
image = cv2.pyrDown(image);

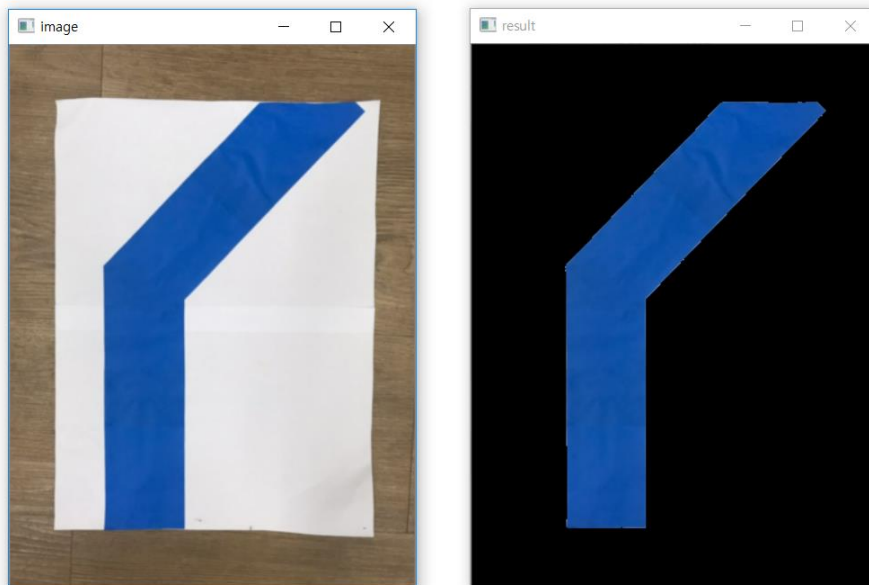
image = cv2.GaussianBlur(image,(5,5),0)
hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)

while True:
    hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)

    l_h = cv2.getTrackbarPos("L - H", "Trackbars")
    l_s = cv2.getTrackbarPos("L - S", "Trackbars")
    l_v = cv2.getTrackbarPos("L - V", "Trackbars")
    u_h = cv2.getTrackbarPos("U - H", "Trackbars")
    u_s = cv2.getTrackbarPos("U - S", "Trackbars")
    u_v = cv2.getTrackbarPos("U - V", "Trackbars")
  
```



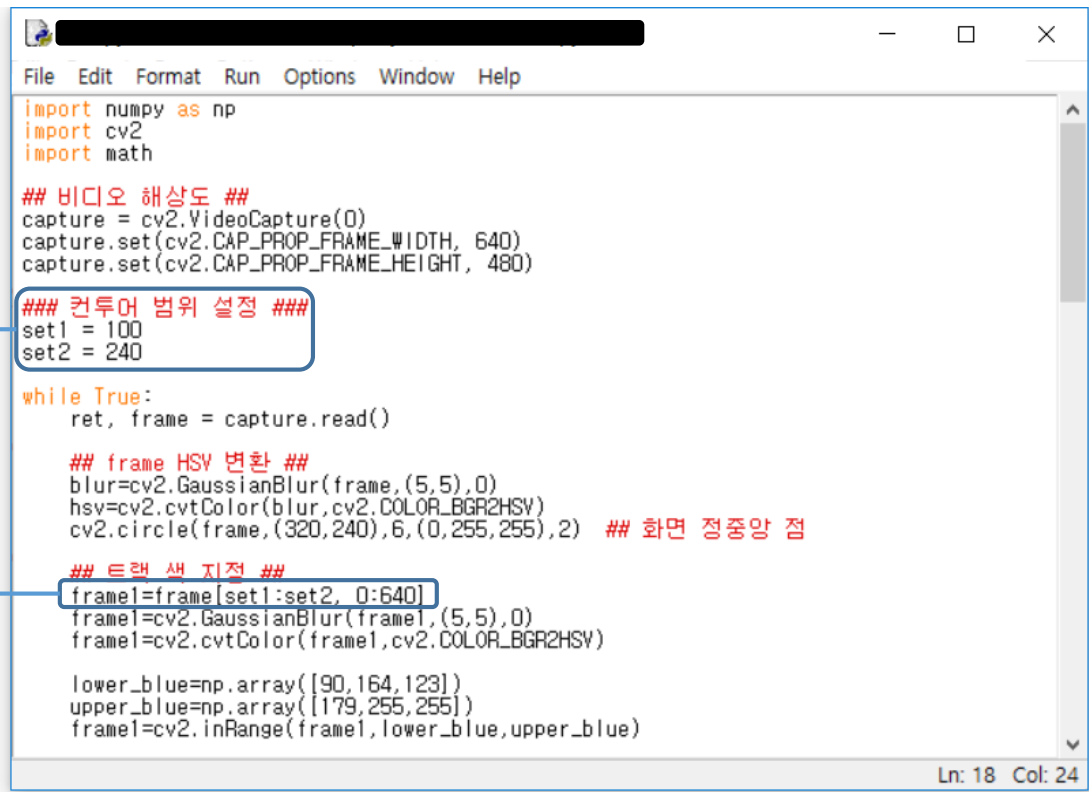
영상 이진화를 이용해 원하는 색(HSV) 이외의 값을 제외시킴



이진화를 통해 영상처리 과정에 있어 필요 없는 요소의 방해를 받지 않음. 꼭 거쳐야 하는 과정

Frame cutting

Frame 을 잘라 목표하는 색과 같은 색이 화면 가장자리에 잡히는 경우에도 오작동을 방지하기 위함



```

import numpy as np
import cv2
import math

## 비디오 해상도 ##
capture = cv2.VideoCapture(0)
capture.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
capture.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)

### 컨투어 범위 설정 ###
set1 = 100
set2 = 240

while True:
    ret, frame = capture.read()

    ## frame HSV 변환 ##
    blur=cv2.GaussianBlur(frame,(5,5),0)
    hsv=cv2.cvtColor(blur,cv2.COLOR_BGR2HSV)
    cv2.circle(frame,(320,240),6,(0,255,255),2) ## 화면 정중앙 점

    ## 트랙 색 지정 ##
    frame1=frame[set1:set2, 0:640]
    frame1=cv2.GaussianBlur(frame1,(5,5),0)
    frame1=cv2.cvtColor(frame1,cv2.COLOR_BGR2HSV)

    lower_blue=np.array([90,164,123])
    upper_blue=np.array([179,255,255])
    frame1=cv2.inRange(frame1,lower_blue,upper_blue)
  
```

`frame1 = frame[set1:set2, set3:set4]`

명령어를 이용해 세로 픽셀을 set1~set2 , 가로 픽셀을 set3~set4 만큼 자를 수 있다.

원래의 출력 화면을 자르는 것이 아니라 영상처리 하는 대상 화면만을 자름

Contour

cv2.findContours 를 이용한 윤곽선 추출 , cv2.drawContours 를 이용한 윤곽선 출력

```

image.py - C:\Users\WLYJ\Desktop\Python\opencv\성공한 예제\Win...
File Edit Format Run Options Window Help

import cv2
import numpy as np

image = cv2.imread("c:\python\project.jpg", cv2.IMREAD_ANYCOLOR)
image = cv2.pyrDown(image);

image = cv2.GaussianBlur(image, (5,5), 0)
hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)

lower_blue=np.array([90,164,123])
upper_blue=np.array([179,255,255])
image1=cv2.inRange(hsv, lower_blue,upper_blue)

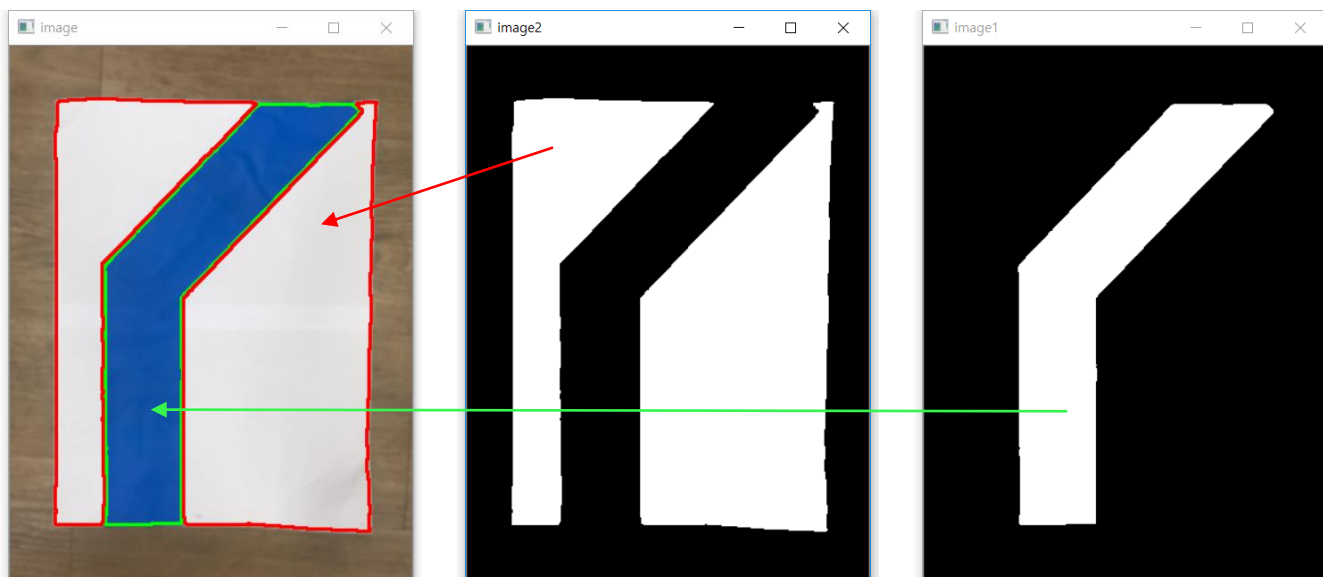
lower_white=np.array([0,0,200])
upper_white=np.array([179,255,255])
image2=cv2.inRange(hsv, lower_white,upper_white)

contours1, _ = cv2.findContours(image1, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
contours2, _ = cv2.findContours(image2, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

for contour in contours1:
    epsilon1=0.0001+cv2.arcLength(contour,True)
    approx1=cv2.approxPolyDP(contour,epsilon1,True)
    cv2.drawContours(image, [approx1], 0, (0,255,0), 2)

for contour in contours2:
    epsilon2=0.0001+cv2.arcLength(contour,True)
    approx2=cv2.approxPolyDP(contour,epsilon2,True)
    cv2.drawContours(image, [approx2], 0, (0,0,255), 2)
  
```

서로 다른 색을 가지는 물체에 다른 Contour를 부여 할 수 있음



Contour Box 를 이용한 좌표 추출

Contour 를 감싸는 Box를 출력함으로서 Box의 네 꼭지점의 좌표를 구함

```
image.py - C:\Users\LVJ\Desktop\Python\opencv\성공한 예제\image.py
File Edit Format Run Options Window Help

upper_blue=np.array([117,255,255])
img=cv2.inRange(hsv,lower_blue,upper_blue)
img1=cv2.inRange(hsv1,lower_blue,upper_blue)
img2=cv2.inRange(hsv2,lower_blue,upper_blue)

contours, _ = cv2.findContours(img, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
contours1, _ = cv2.findContours(img1, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
contours2, _ = cv2.findContours(img2, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

if len(contours1)>0:
    for i in range(len(contours1)):
        rect1=cv2.minAreaRect(contours1[i])
        box1=cv2.boxPoints(rect1)
        box1=np.int0(box1)

        cv2.drawContours(image1,[box1],-1,(0,255,0),3)

        #print(box)
        # 3 4
        # 2 1

        x1,y1=box1[0]
        x2,y2=box1[1]
        x3,y3=box1[2]
        x4,y4=box1[3]

        cv2.circle(image1,(x1,y1),4,(0,0,255),2) #빨
        cv2.circle(image1,(x2,y2),4,(255,0,0),2) #파
        cv2.circle(image1,(x3,y3),4,(0,255,255),2) #노
        cv2.circle(image1,(x4,y4),4,(0,128,255),2) #주

Ln: 41 Col: 21
```

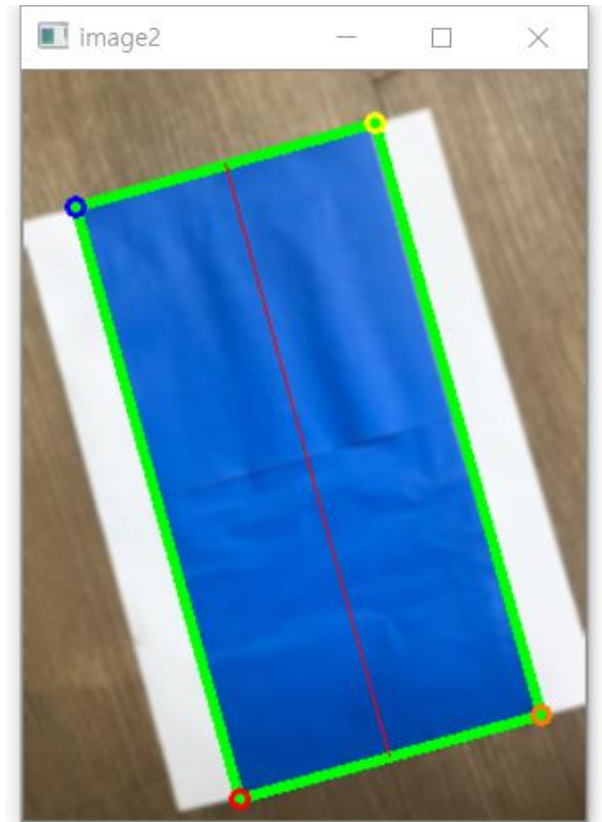
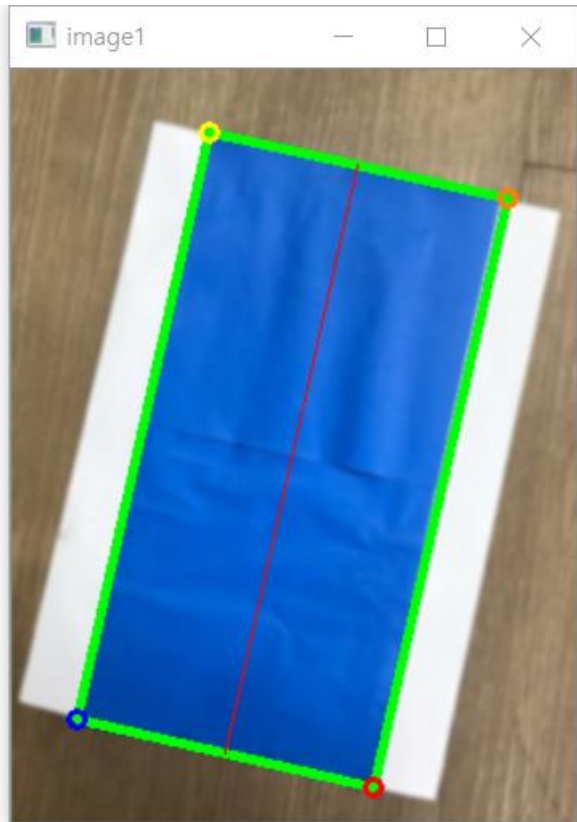


```
*Python 3.5.3 Shell*
File Edit Shell Debug Options Window Help

Python 3.5.3 (v3.5.3:1880cb95a742, Jan 16 2017, 15:51:26) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\LVJ\Desktop\Python\opencv\성공한 예제\image.py =====
==
x1= 180 y1= 358
x2= 32 y2= 324
x3= 98 y3= 31
x4= 247 y4= 64
|

Ln: 5 Col: 0
```

Contour Box 를 이용한 좌표 추출



주의사항 : 물체의 각도에 따라 네 꼭지점이 수시로 바뀜.
이 좌표를 이용하기 위해선 적절한 조건문을 작성해야 함

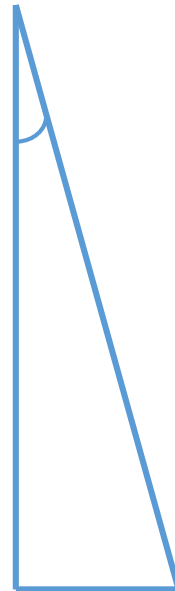
1지점 2지점 3지점 4지점

Math 라이브러리를 이용한 각도 추출

math 라이브러리를 이용해 box의 좌표 값으로 각도를 쉽게 구할 수 있음



(x_2, y_2)



(x_2, y_1)

(x_1, y_1)

math 라이브러리의 atan를 이용하여 각도 추출



```
print("angle = ", int(math.atan(((x11-x12)/2)/((y11-y12)/2))+180/math.pi))
```

```
Python 3.5.3 Shell
File Edit Shell Debug Options Window Help
Python 3.5.3 (v3.5.3:1880cb95a742, Jan 16 2017, 15:51:26) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\LYJ\Desktop\Python\opencv\성공한 예제\image.py =====
angle = 15
>>> |
```

Ln: 6 Col: 4

OpenCV 를 이용한 정보 표시

cv2.line , cv2.circle , cv2.putText 등을 이용해 화면에 필요한 정보를 실시간으로 출력할 수 있음

```
image.py - C:\Users\WLY\Desktop\Python\opencv\성공한 예제\image.py (3.5.3)
File Edit Format Run Options Window Help

cv2.circle(image1, (int((x1+x3)/2), int((y1+y3)/2)), 4, (255, 255, 0), 2)
cv2.putText(image1, '('+str(int((x1+x3)/2))+', '+str(int((y1+y3)/2))+')', (int((x1+x3)/2), int((y1+y3)/2)), cv2.FONT_HERSHEY_SCRIPT_

if len(contours2)>0:
    for i in range(len(contours2)):
        rect2=cv2.minAreaRect(contours2[i])
        box2=cv2.boxPoints(rect2)
        box2=np.int0(box2)

        cv2.drawContours(image2, [box2], -1, (0, 255, 0), 3)

        x11,y11=box2[0]
        x12,y12=box2[1]
        x13,y13=box2[2]
        x14,y14=box2[3]

        cv2.circle(image2, (x11,y11), 4, (0, 0, 255), 2) #빨
        cv2.circle(image2, (x12,y12), 4, (255, 0, 0), 2) #파
        cv2.circle(image2, (x13,y13), 4, (0, 255, 255), 2) #노
        cv2.circle(image2, (x14,y14), 4, (0, 128, 255), 2) #주

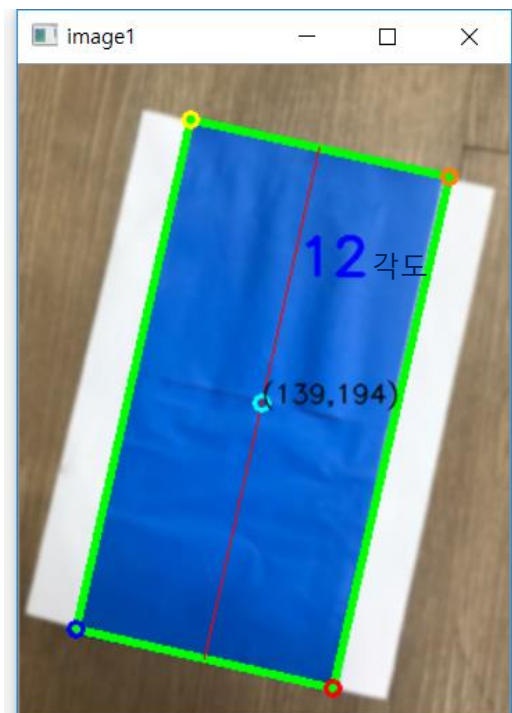
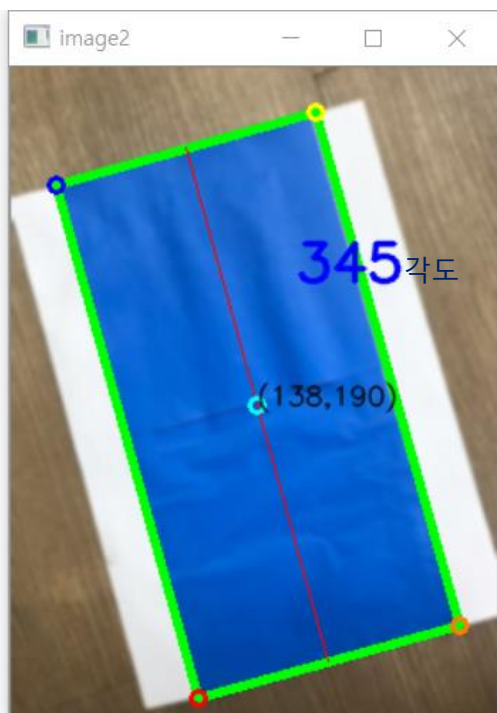
        angle11 = 360-int(math.atan((x11-x12)/(y11-y12))+180/math.pi)
        angle12 = int(math.atan((x13-x12)/(y12-y13))+180/math.pi)

        ## 일정한 각도를 출력하기 위한 조건문 ##
        if int(y12)<141:
            cv2.line(image2, (int((x12+x13)/2), int((y12+y13)/2)), (int((x11+x14)/2), int((y11+y14)/2)), (0, 0, 255), 1)
            cv2.putText(image2, str(angle11), (160, 120), cv2.FONT_HERSHEY_SCRIPT_SIMPLEX, 1, (255, 0, 0), 2, cv2.LINE_AA)
        elif int(y12)>141:
            cv2.line(image2, (int((x13+x14)/2), int((y13+y14)/2)), (int((x12+x11)/2), int((y12+y11)/2)), (0, 0, 255), 1)
            cv2.putText(image2, str(angle12), (160, 120), cv2.FONT_HERSHEY_SCRIPT_SIMPLEX, 1, (255, 0, 0), 2, cv2.LINE_AA)

        ## 트랙의 정중앙 지점 ##
        cv2.circle(image2, (int((x11+x13)/2), int((y11+y13)/2)), 4, (255, 255, 0), 2)
        cv2.putText(image2, '('+str(int((x11+x13)/2))+', '+str(int((y11+y13)/2))+')', (int((x11+x13)/2), int((y11+y13)/2)), cv2.FONT_HERSHEY_SCRIPT_

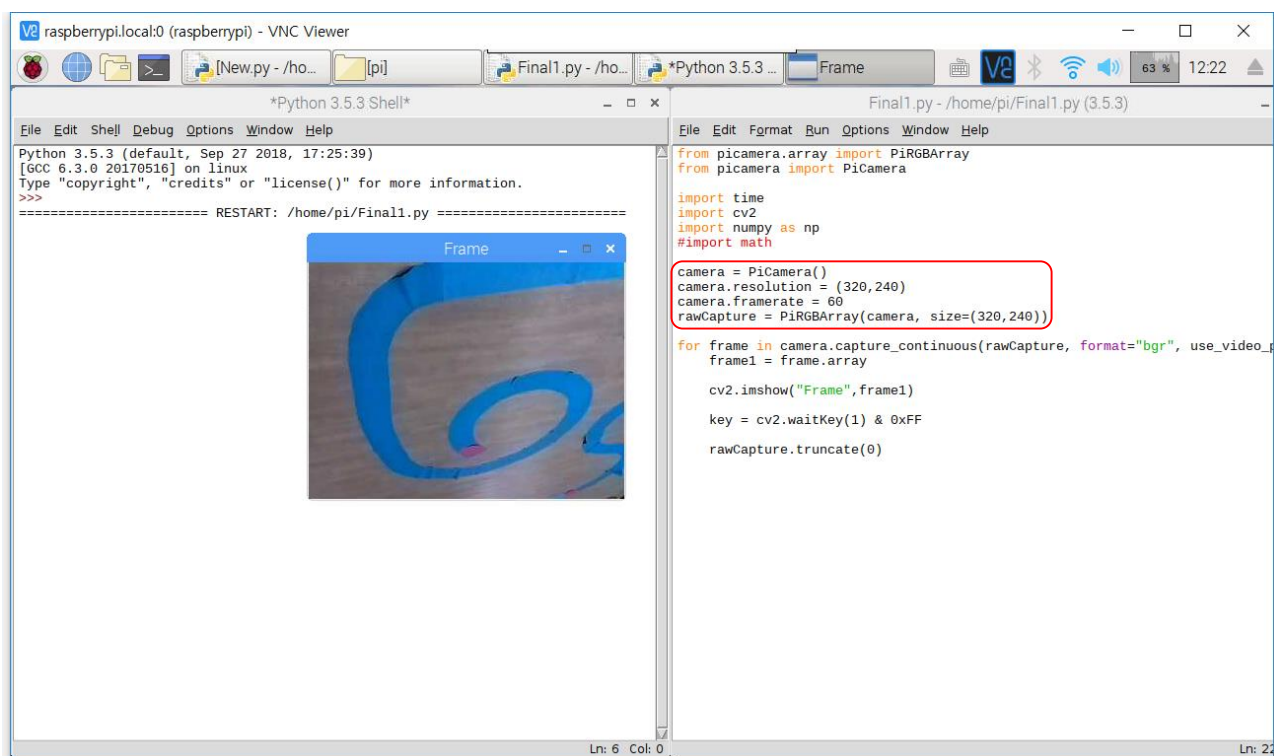
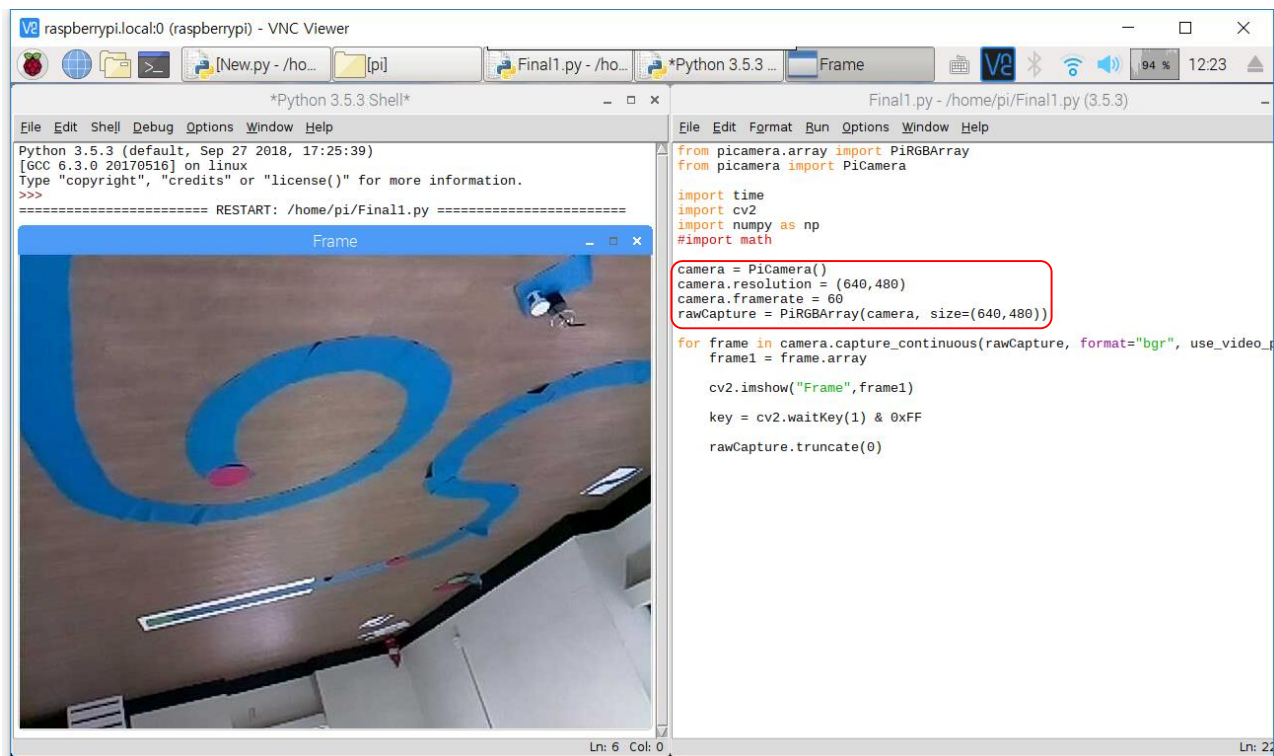
cv2.imshow("image", image)
```

Ln: 59 Col: 90



Raspberry Pi Zero 적용 시 고려사항

처리속도가 느린 Raspberry Pi Zero 특성상 작은 화면으로 출력하는 것이 실시간으로 영상을 받아오는 것에 유리함



동작 알고리즘

