

머신비전시스템 과제4

18011789 조혜수

1

원본사진



1-1) Harris corner detector

결과 영상



코드

```
[1] from google.colab import drive
drive.mount('/content/gdrive')

Mounted at /content/gdrive

[33] import cv2 as cv
import numpy as np
from google.colab.patches import cv_imshow

img_big = cv.imread('/content/gdrive/MyDrive/MachineVision/4_1_Ori.jpeg')

[34] img_big.shape

(3000, 4000, 3)

[35] img = cv.resize(img_big, dsize=(640, 480), interpolation=cv.INTER_CUBIC)

[30] img.shape

(480, 640, 3)

cv_imshow(img)
```

```
▶ gray = cv.cvtColor(img,cv.COLOR_BGR2GRAY)
gray = np.float32(gray)
corners = cv.goodFeaturesToTrack(gray, maxCorners=100, qualityLevel=0.1, minDistance=10)
for c in corners:
    x, y = c.ravel()
    cv.circle(img,(x,y),3,(0,0,255),-1)
cv.imshow('img')
```

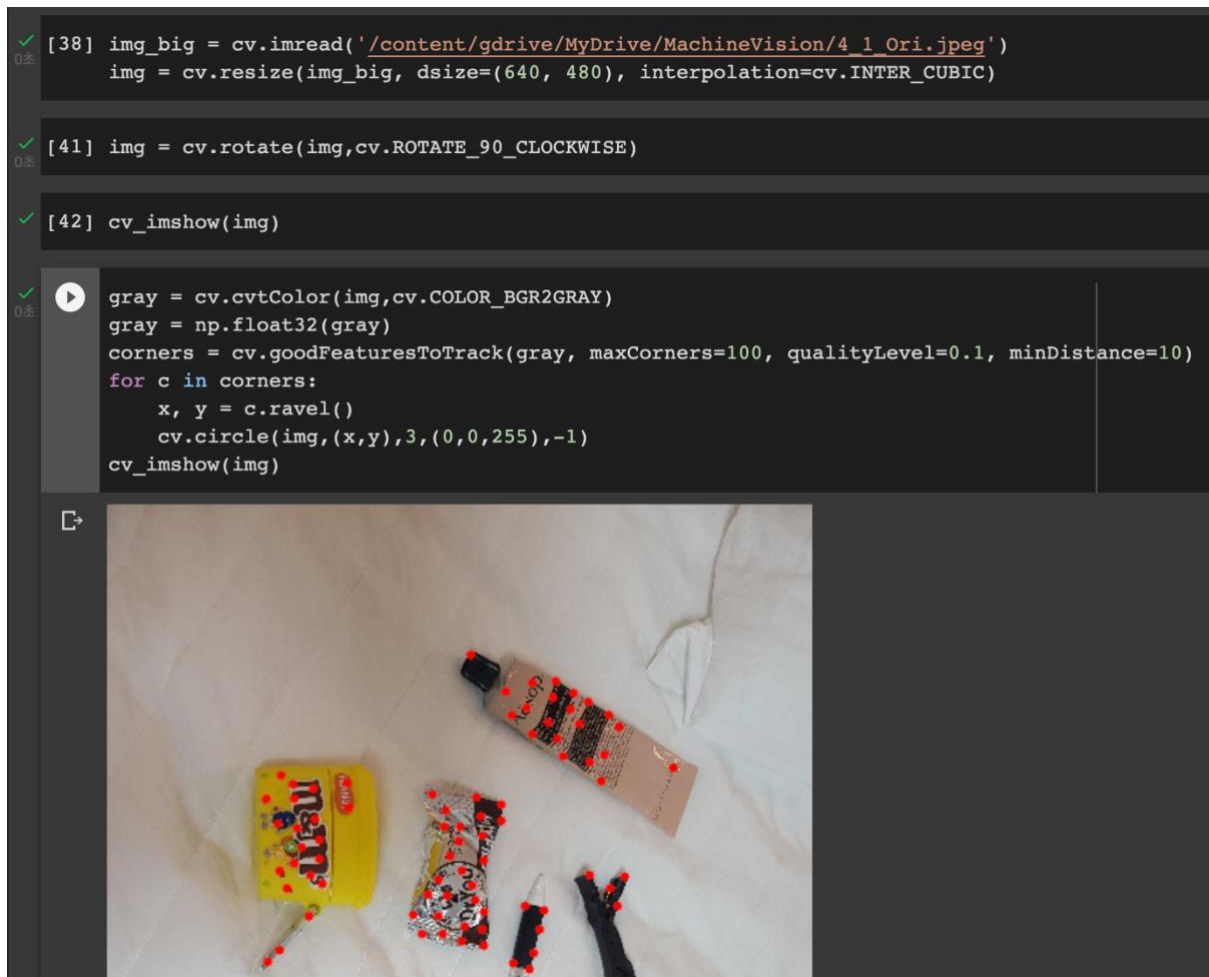


1-2)

결과영상



코드



1-3)

결과 영상



코드

```
[68] img_big = cv.imread('/content/gdrive/MyDrive/MachineVision/4_1 Ori.jpeg')
img = cv.resize(img_big, dsize=(640, 480), interpolation=cv.INTER_CUBIC)

[69] img = cv.resize(img, dsize=(1280, 960), interpolation=cv.INTER_CUBIC)

gray = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
gray = np.float32(gray)
corners = cv.goodFeaturesToTrack(gray, maxCorners=100, qualityLevel=0.1, minDistance=10)
for c in corners:
    x, y = c.ravel()
    cv.circle(img, (x, y), 3, (0, 0, 255), -1)
cv.imshow(img)
```

1-4)

1)과 2)는 정확히 같은 위치에서 코너가 검출된다.

하지만 1)에서 검출된 코너가 3)에서는 검출되지 않은 것들이 있다.

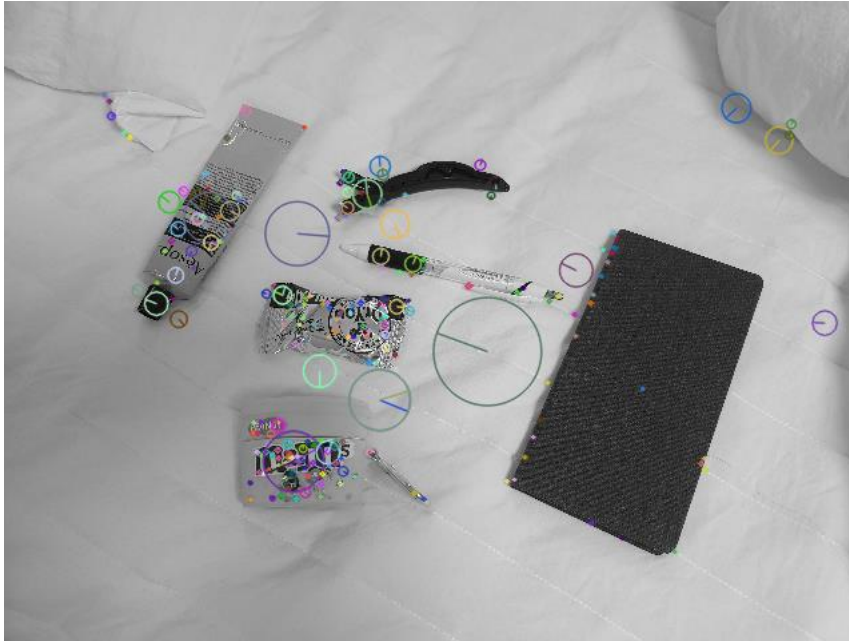
이를 통해 Harris corner detector는 rotation에는 강인하지만 Scaling에서는 한계가 있음을 알 수 있다.

2.

원본 영상 1과 같음

2-1)

결과 영상



코드

```
[3] import subprocess
subprocess.call("pip install -U opencv-python".split())

0

[4] img_big = cv.imread('/content/gdrive/MyDrive/MachineVision/4_1_Ori.jpeg')
img = cv.resize(img_big, dsize=(640, 480), interpolation=cv.INTER_CUBIC)

▶ gray= cv.cvtColor(img,cv.COLOR_BGR2GRAY)
sift = cv.SIFT_create()
kp = sift.detect(gray,None)
img=cv.drawKeypoints(gray,kp,img, flags=cv.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)
cv.imshow(img)
```

A smaller version of the original image, showing the same objects with detected keypoints and matches.

2-2)

결과 영상




코드

```
[6] img_big = cv.imread('/content/gdrive/MyDrive/MachineVision/4_1_Ori.jpeg')
img = cv.resize(img_big, dsize=(640, 480), interpolation=cv.INTER_CUBIC)

[7] img = cv.rotate(img,cv.ROTATE_90_CLOCKWISE)

gray= cv.cvtColor(img,cv.COLOR_BGR2GRAY)
sift = cv.SIFT_create()
kp = sift.detect(gray,None)
img=cv.drawKeypoints(gray,kp,img, flags=cv.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)
cv_imshow(img)
```



2-3)

결과 영상



코드

```
[9] img_big = cv.imread('/content/gdrive/MyDrive/MachineVision/4_1_Ori.jpeg')
img = cv.resize(img_big, dsize=(640, 480), interpolation=cv.INTER_CUBIC)

[10] img = cv.resize(img, dsize=(1280, 960), interpolation=cv.INTER_CUBIC)

gray= cv.cvtColor(img,cv.COLOR_BGR2GRAY)
sift = cv.SIFT_create()
kp = sift.detect(gray,None)
img=cv.drawKeypoints(gray,kp,img, flags=cv.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)
cv.imshow(img)
```

2-4)

1)에서 blob으로 검출된 것은 거의 대부분 2)에서도 검출되었다. 2)에서 일부 추가적으로 검출 되었다. 하지만 결과가 거의 동일하다.

1)에서 blob으로 검출된 것은 거의 대부분 3)에서도 검출되었다. 하지만 3)에서 아주 많이 추가적으로 검출 되었다. 1)에 있는 것은 거의 3)에 있지만 3)에 있는 것은 1)에 많이 없다.

결과가 완전히 똑같지는 않지만 1)에 있는 것들은 거의 2),3)에 있는 것으로 보아 회전과 scaling 되어있는 상태에서도 추후 Handling Scale 과 Handling Rotation 을 거치면 blob detection이 잘 될 것이다.

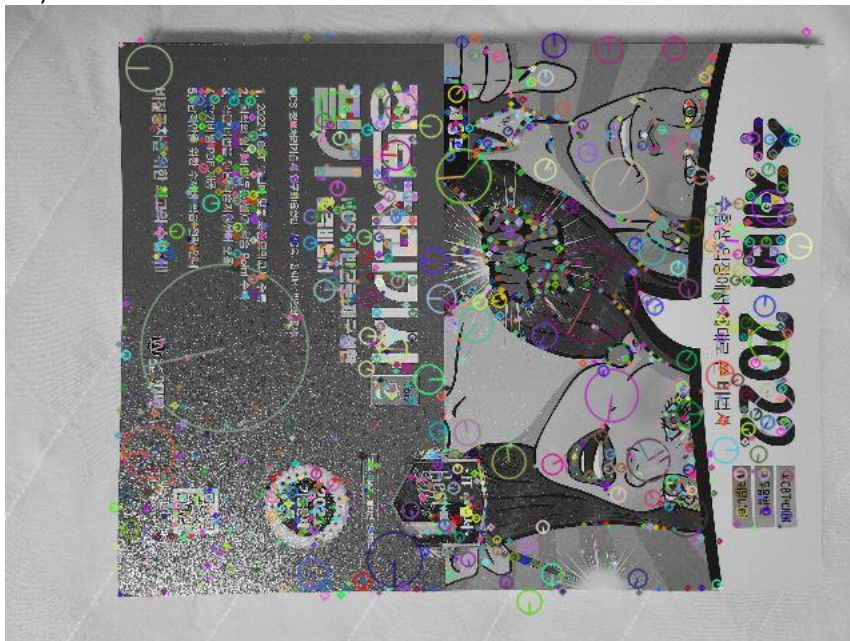
3.

원본영상





3-1)





코드

```
[ ] img_big1 = cv.imread('/content/gdrive/MyDrive/MachineVision/4_3_Ori3.jpeg')
img_big2 = cv.imread('/content/gdrive/MyDrive/MachineVision/4_3_Ori4.jpeg')

img1 = cv.resize(img_big1, dsize=(640, 480), interpolation=cv.INTER_CUBIC)
img2 = cv.resize(img_big2, dsize=(640, 480), interpolation=cv.INTER_CUBIC)

▶ gray= cv.cvtColor(img1,cv.COLOR_BGR2GRAY)
sift = cv.SIFT_create()
kp = sift.detect(gray,None)
img1=cv.drawKeypoints(gray,kp,img, flags=cv.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)
cv_imshow(img1)

▶ gray= cv.cvtColor(img2,cv.COLOR_BGR2GRAY)
sift = cv.SIFT_create()
kp = sift.detect(gray,None)
img2=cv.drawKeypoints(gray,kp,img, flags=cv.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)
cv_imshow(img2)
```

3-2)

결과 영상



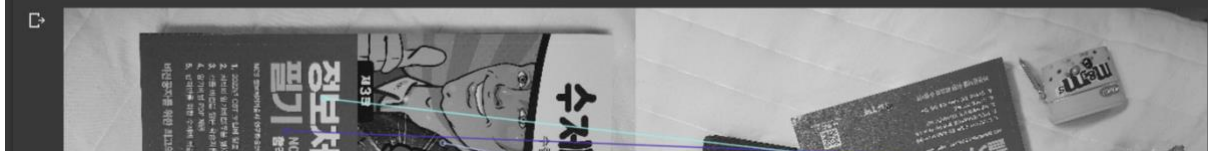
코드

```
[10] img_big1 = cv.imread('/content/gdrive/MyDrive/MachineVision/4_3 Ori3.jpeg', cv.IMREAD_GRAYSCALE)
img_big2 = cv.imread('/content/gdrive/MyDrive/MachineVision/4_3 Ori4.jpeg', cv.IMREAD_GRAYSCALE)

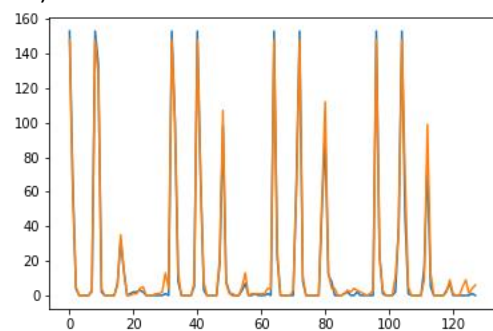
img1 = cv.resize(img_big1, dsize=(640, 480), interpolation=cv.INTER_CUBIC)
img2 = cv.resize(img_big2, dsize=(640, 480), interpolation=cv.INTER_CUBIC)

[11] kp1, des1 = sift.detectAndCompute(img1, None)
kp2, des2 = sift.detectAndCompute(img2, None)

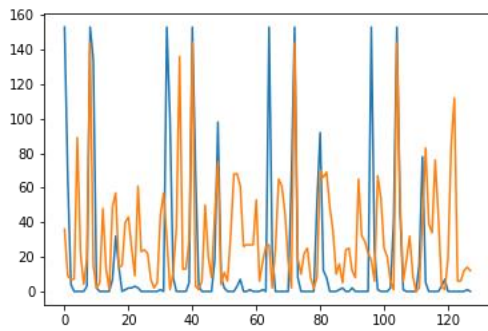
bf = cv.BFMatcher()
matches = bf.match(des1, des2)
matches = sorted(matches, key = lambda x:x.distance)
img3 = cv.drawMatches(img1, kp1, img2, kp2, matches[:15], None, flags=cv.DrawMatchesFlags_NOT_DRAW_SINGLE_POINTS)
cv.imshow('img3')
```



3-3)

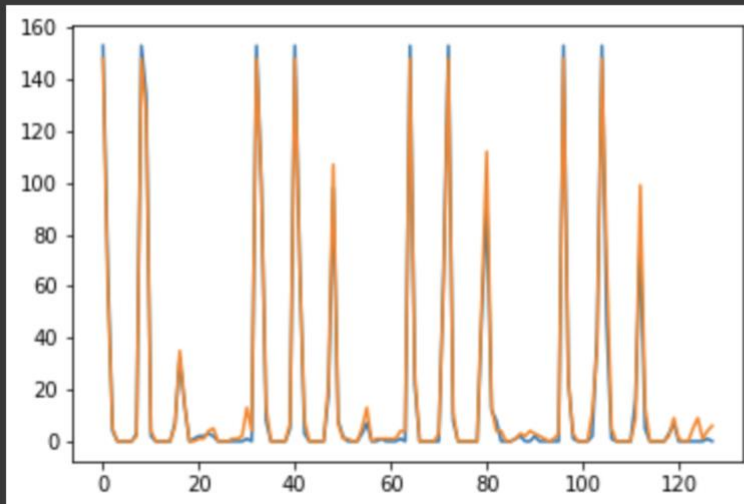


3-4)



코드

```
[13] import matplotlib.pyplot as plt
plt.plot(np.arange(128), des1[matches[0].queryIdx])
plt.plot(np.arange(128), des2[matches[0].trainIdx]), plt.show()
```



([<matplotlib.lines.Line2D at 0x7f6069f310d0>], None)



```
plt.plot(np.arange(128), des1[matches[0].queryIdx])
plt.plot(np.arange(128), des2[matches[60].trainIdx]), plt.show()
```

