

2020년 컴퓨터그래픽스

- HW 04 -

제출일자	2020.10.15.
이름	장수훈
학번	201402414
분반	00

구현코드

```
def calc_derivatives(src):  
    """  
    #ToDo  
    3x3 sobel 필터를 사용해서 Ix Iy 구하기  
    :param src: 입력 이미지 (흑백)  
    :return: Ix, Iy  
    """  
  
    sobel_x, sobel_y = get_my_sobel()  
    # calculate Ix, Iy  
    Ix = my_filtering(src, sobel_x)  
    Iy = my_filtering(src, sobel_y)  
    return Ix, Iy
```

위에 구현되어 있는 mu_filtering을 이용하여 sobel필터를 적용하였다.

```
"""  
#ToDo  
IxIx = Ix^2  
IyIy = Iy^2  
IxIy = Ix * Iy  
#구하기  
"""  
  
# Square of derivatives  
IxIx = np.square(Ix)  
IyIy = np.square(Iy)  
IxIy = np.multiply(Ix, Iy)
```

np 내장함수를 통해 제곱과 곱셈을 하였다.

```

#가우시안 필터 적용하기
#G_IxIx = IxIx에 가우시안 필터 적용
#G_IyIy = IyIy에 가우시안 필터 적용
#G_IxIy = IxIy에 가우시안 필터 적용
"""
G_IxIx = GaussianFiltering(IxIx, fshape=(3,3), sigma=1)
G_IyIy = GaussianFiltering(IyIy, fshape=(3,3), sigma=1)
G_IxIy = GaussianFiltering(IxIy, fshape=(3,3), sigma=1)

```

위에 구현되어 있는 가우시안 필터를 이용해 $I_x I_x$, $I_y I_y$, $I_x I_y$ 를 각각 적용 시켰다.

```

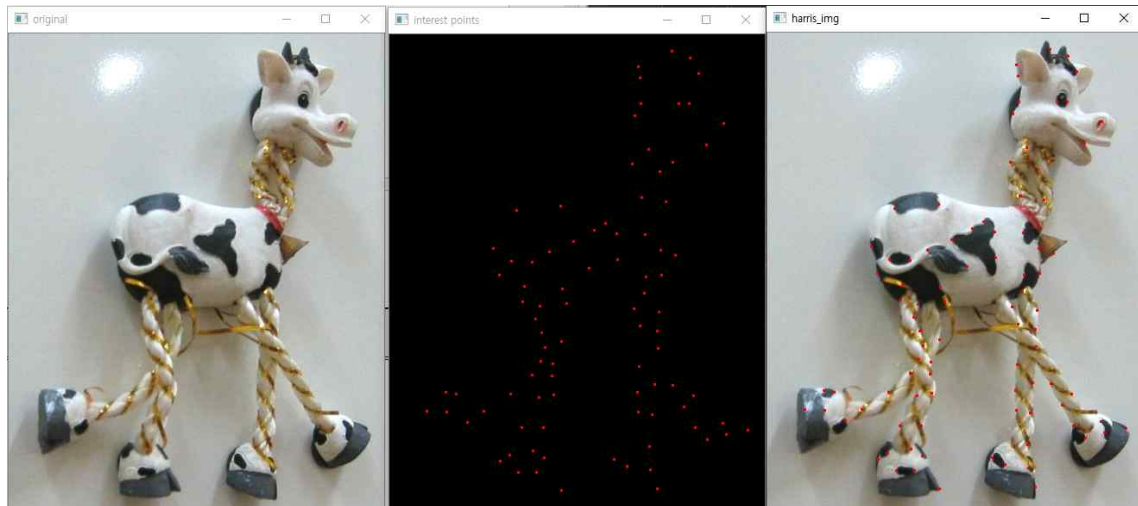
#ToDo
# har 구하기 교수님 이론 pdf 67page 참고
"""
har = G_IxIx * G_IyIy - np.square(G_IxIy) - 0.12*np.square(G_IxIx + G_IyIy)

```

$$g(I_x^2)g(I_y^2) - [g(I_x I_y)]^2 - \alpha [g(I_x^2) + g(I_y^2)]^2$$

교수님 이론을 참고하여 그대로 적용하였다.

결과값



느낀점

함수가 여러개 미리 정의되었고 그 함수를 활용해 구현해야 할 함수 빈칸을 채우는 과정이 전체적인 흐름이 어떻게 돌아가는지 파악할 수 있어서 이해하기 편했다.

난이도

기존의 영상처리 지식과 python 지식이 없던 나로서는 이 정도의 난이도가 적당한 것 같았다.