

이름 : 강시온

학번 : 201802045

학과 : 컴퓨터공학과

구현 코드

```
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
```

```
    """  
    #ToDo  
    IxIx = Ix^2  
    IyIy = Iy^2  
    IxIy = Ix * Iy  
    #구하기  
    """  
    # Square of derivatives  
    IxIx = Ix ** 2  
    IyIy = Iy ** 2  
    IxIy = Ix * Iy
```

```
# Gaussian filter
"""
#ToDo
#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)
G_IyIy = GaussianFiltering(IyIy)
G_IxIy = GaussianFiltering(IxIy)
```

```
# cornerness function - both eigenvalues are strong
"""
#ToDo
# har 구하기 교수님 이론 pdf 67page 참고
"""
har = G_IxIx * G_IyIy - G_IxIy ** 2 - alpha * (G_IxIx + G_IyIy) ** 2
```

코드 설명

Get_my_sobel() 함수를 사용하여 적용할 필터를 만들고 그 필터를 my_filtering() 함수를 사용하여 해당 이미지에 적용하였습니다.

그렇게 구한 Ix, Iy를 각각 연산하여 IxIx, IyIy, IxIy 3이미지를 만들어주고 각각 GaussianFiltering() 함수를 이용해 가우시안필터를 적용하여줍니다.

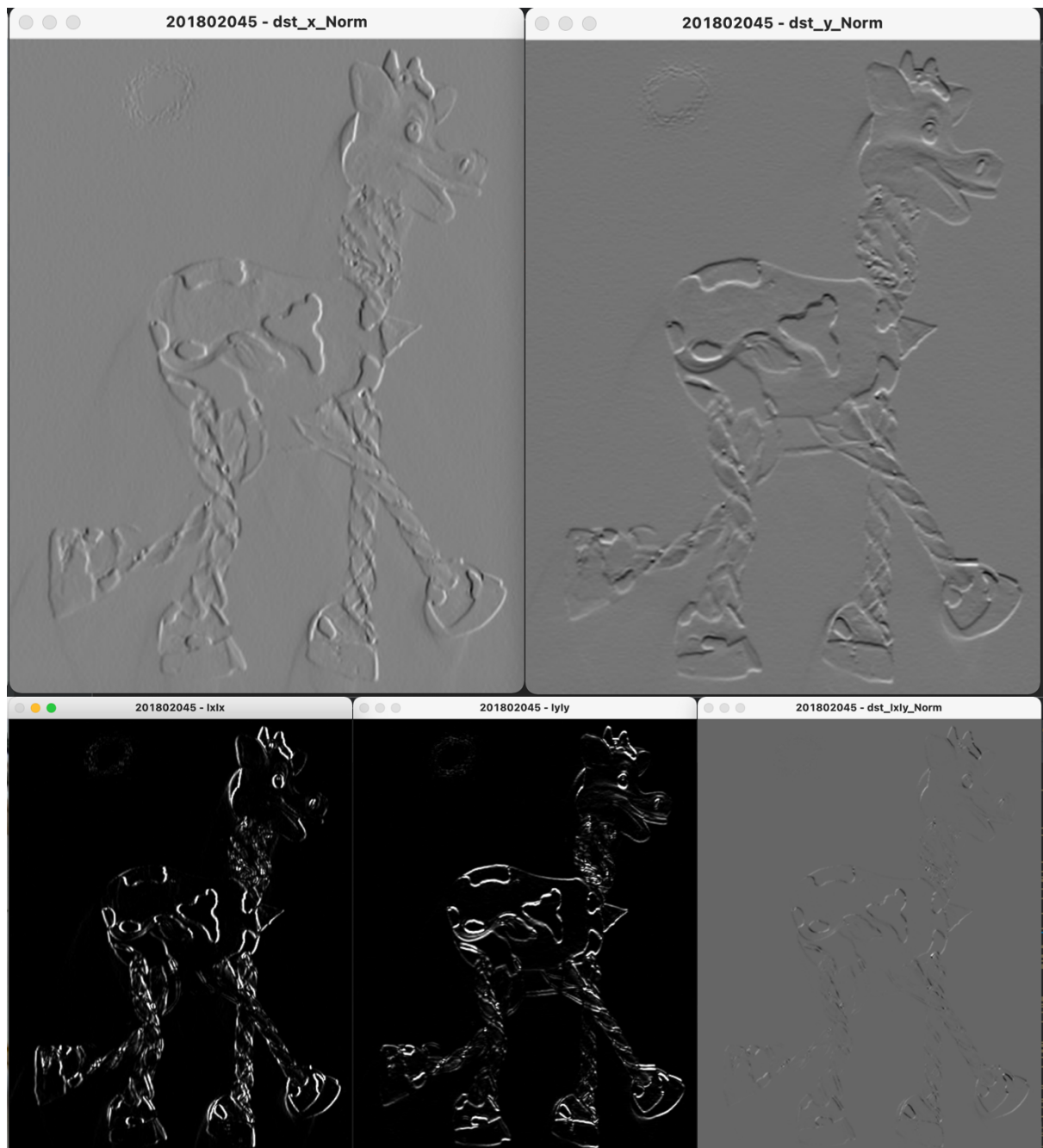
그리고 har 는

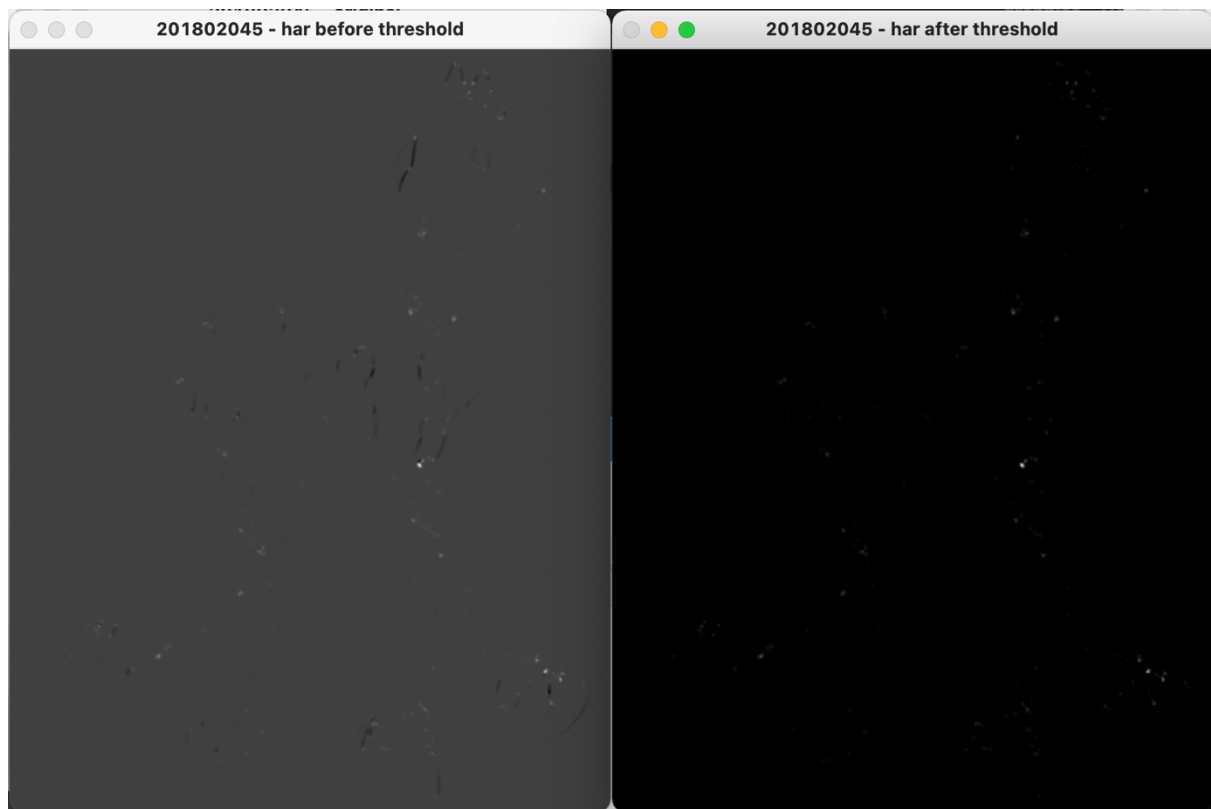
4. Cornerness function – both eigenvalues are strong

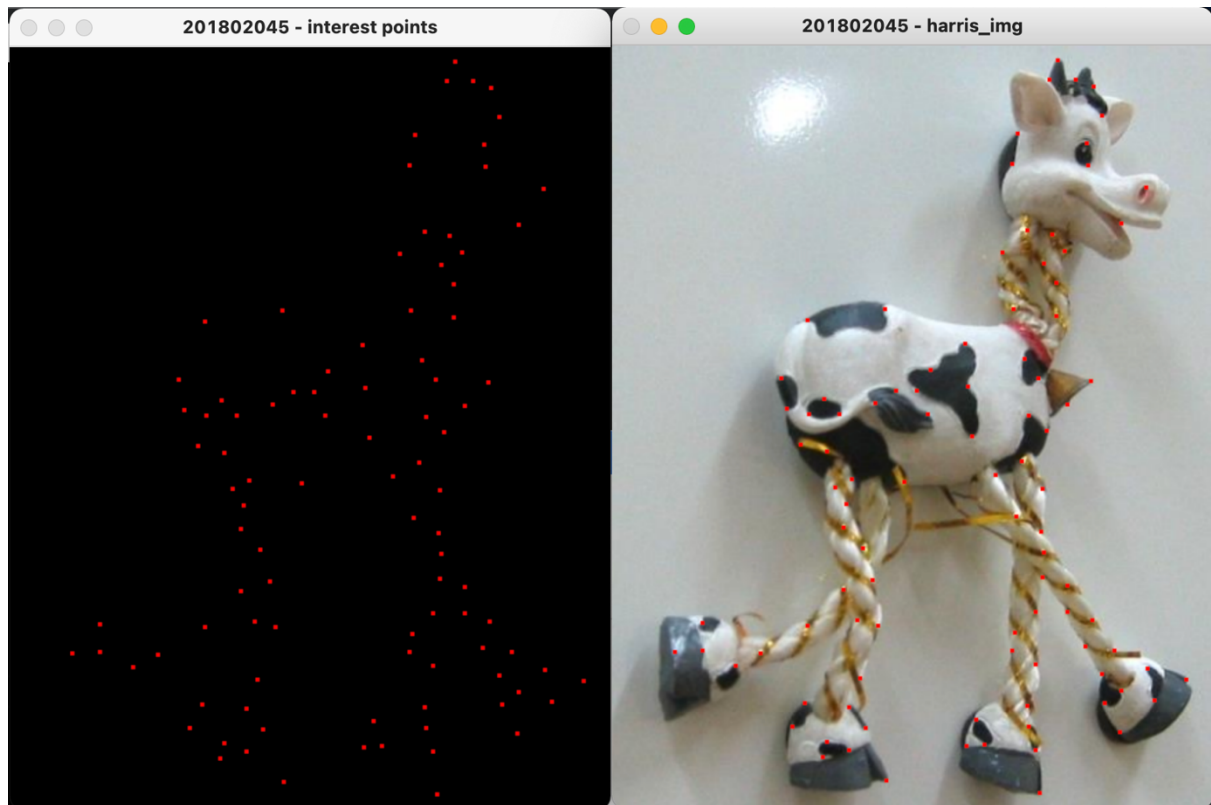
$$har = \det[\mu(\sigma_I, \sigma_D)] - \alpha [\text{trace}(\mu(\sigma_I, \sigma_D))^2] = g(I_x^2)g(I_y^2) - [g(I_x I_y)]^2 - \alpha [g(I_x^2) + g(I_y^2)]^2$$

해당 수식을 참고하여 수식과 같이 작성하였습니다.

결과 이미지







난이도

첫과제여서 쉽게 내주셔서 쉽게 풀이할 수 있었습니다.

특히 윗부분에 함수로 구현된 부분이 많아 그부분을 활용하여 풀어서 편했습니다.