

2020년 컴퓨터그래픽스

- HW 05 -

제출일자	2020.10.27.
이름	장수훈
학번	201402414
분반	00

구현코드

```
# for row in range(h):
#     for col in range(w):
#         M[row, col, 0, 0] = np.sum(IxIx_pad[row:row+fsize, col:col+fsize])
#         M[row, col, 0, 1] = np.sum(IxIy_pad[row:row+fsize, col:col+fsize])
#         M[row, col, 1, 0] = M[row, col, 0, 1]
#         M[row, col, 1, 1] = np.sum(IyIy_pad[row:row+fsize, col:col+fsize])

for row in range(h):
    for col in range(w):
        ixix = 0
        ixiy = 0
        iyiy = 0
        for f_row in range(fsize):
            for f_col in range(fsize):
                ixix = ixix + IxIx_pad[row + f_row][col + f_col]
                ixiy = ixiy + IxIy_pad[row + f_row][col + f_col]
                iyiy = iyiy + IyIy_pad[row + f_row][col + f_col]
        M[row, col, 0, 0] = ixix
        M[row, col, 0, 1] = ixiy
        M[row, col, 1, 0] = ixiy
        M[row, col, 1, 1] = iyiy

return M
```

위의 2중포문을 4중포문으로 교체하였다. 시간을 확인하기 위하여 2중포문을 이용해 np.sum을 이용하는게 아니고 4중포문을 이용해 직접 더한값을 넣어주었다.

```

integral_image = dst
for row in range(0, h):
    summation = 0
    for col in range(0, w):
        summation += src[row][col]
        integral_image[row][col] = summation

    if row > 0:
        integral_image[row][col] += integral_image[row - 1][col]

# dst = integral_image
# return dst

dst2 = np.zeros(src.shape)

for y in range(h):
    for x in range(w):

        min_row, max_row = max(0, y - 1), min(h - 1, y + 1)
        min_col, max_col = max(0, x - 1), min(w - 1, x + 1)

        dst2[y][x] = integral_image[max_row][max_col]

        if min_row > 0:
            dst2[y][x] -= integral_image[min_row - 1][max_col]

        if min_col > 0:
            dst2[y][x] -= integral_image[max_row][min_col - 1]

        if min_col > 0 and min_row > 0:
            dst2[y][x] += integral_image[min_row - 1][min_col - 1]

return dst2

```

이미지 전제 적분을 해준 뒤 영역적분을 해 시간을 줄어들게 해주었다

```

M_harris = calc_M_harris(IxIx, IxIy, IyIy, fsize)
end = time.perf_counter() # 시간 측정 끝
print('M_harris time : ', end-start)

R = np.zeros((h, w))
for row in range(h):
    for col in range(w):
        #####
        # ToDo
        # det_M 계산
        # trace_M 계산
        # R 계산 Harris & Stephens (1988), Nobel (1998) 어떤걸로 구현해도 상관없음
        #####
        det_M = M_harris[row, col, 0, 0] * M_harris[row, col, 1, 1] - (M_harris[row, col, 0, 1] * M_ha
        trace_M = M_harris[row, col, 0, 0] + M_harris[row, col, 1, 1]
        R[row, col] = det_M - k*trace_M*trace_M

```

harris&stephens 공식을 사용해 작성하였다.

```

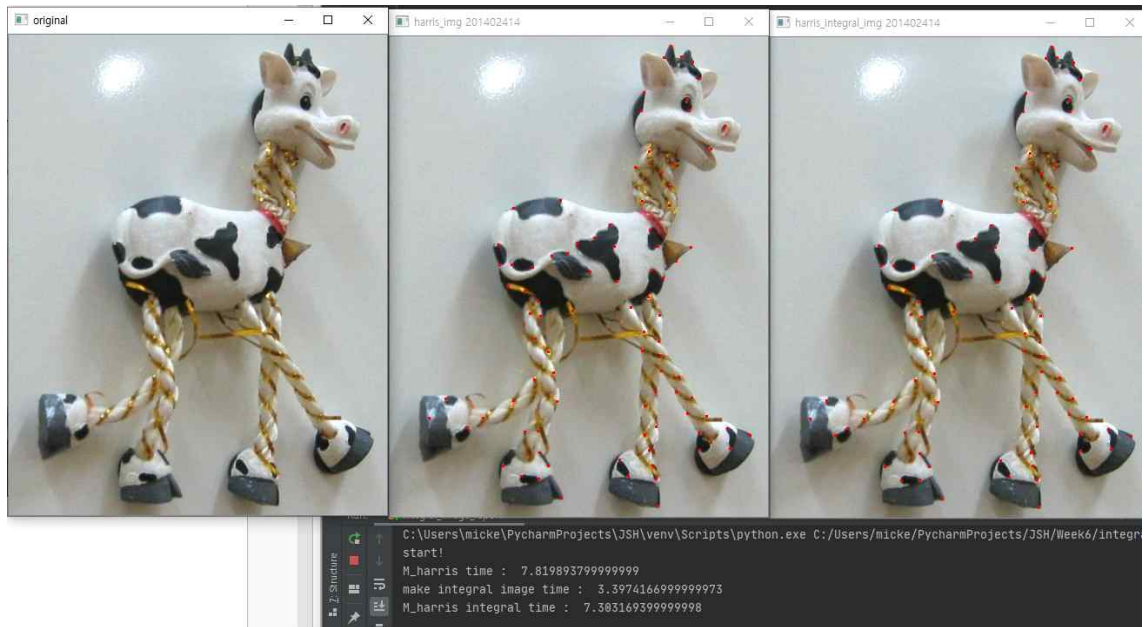
M_integral = calc_M_harris(IxIx_integral, IxIy_integral, IyIy_integral, fsize)
end = time.perf_counter() # 시간 측정 끝
print('M_harris_integral time : ', end-start)

R = np.zeros((h, w))
for row in range(h):
    for col in range(w):
        #####
        # ToDo
        # det_M 계산
        # trace_M 계산
        # R 계산 Harris & Stephens (1988), Nobel (1998) 어떤걸로 구현해도 상관없음
        #####
        det_M = M_integral[row, col, 0, 0] * M_integral[row, col, 1, 1] - (M_integral[row, col, 0, 1]
        trace_M = M_integral[row, col, 0, 0] + M_integral[row, col, 1, 1]
        R[row, col] = det_M - k * trace_M * trace_M

```

integral도 harris&stephens 공식을 사용해 작성하였다.

결과값



느 낯점

채워야 할 함수에 대한 자세한 설명이 없어서 어려웠다.

난이도

저번주에 비해서 어려웠던거 같다.