

## 컴퓨터공학과 201702081 최재범 7주차\_과제

- 구현 코드 및 설명

- ◆ feature\_matching, l2\_distance 는 저번 주차에서 그대로 가져옴

- ◆ 점 찍기 : (160, 160) 에 각각의 M을 적용한 결과를 계산함

```
vec = np.dot(M, np.array([[160, 160, 1]]).T)
```

```
vec = np.dot(M_FM, np.array([[160, 160, 1]]).T)
```

```
vec = np.dot(M_FM_RANSAC, np.array([[160, 160, 1]]).T)
```

- ◆ scaling\_test ~ Forward 방식으로 결과 계산을 진행함

```
#FORWARD
h_, w_ = dst_for.shape[:2]
count = dst_for.copy()
for row in range(h):
    for col in range(w):
        ...
        #ToDo
        #과제에서 사용하지 않지만 완성해주세요
        #실습을 참고해서 완성해주세요
        ...
        vec = np.dot(M, np.array([[col, row, 1]]).T)
        x = vec[0, 0]
        y = vec[1, 0]
        x1 = int(np.floor(x))
        x2 = int(np.ceil(x))
        y1 = int(np.floor(y))
        y2 = int(np.ceil(y))

        points_list = [(y1, x1), (y1, x2), (y2, x1), (y2, x2)]
        points = set(points_list) # 중복제거

        for (row_, col_) in points:
            dst_for[min(row_, h_-1), min(col_, w_-1)] += src[row, col]
            count[min(row_, h_-1), min(col_, w_-1)] += 1

dst_for = (dst_for / count).astype(np.uint8)
```

- ◆ scaling\_test ~ Backward 방식으로 결과 계산을 진행함

```
#BACKWARD
for row_ in range(h_):
    for col_ in range(w_):
        '''
        #ToDo
        #bilinear
        #실습을 참고해서 완성해주세요
        '''

        vec = np.dot(M_, np.array([[col_, row_, 1]]).T)
        c = vec[0, 0]
        r = vec[1, 0]
        c_left = int(c)
        c_right = min(int(c + 1), w - 1)
        r_top = int(r)
        r_bottom = min(int(r + 1), h - 1)
        s = c - c_left
        t = r - r_top

        intensity = (1 - s) * (1 - t) * src[r_top, c_left] \
                    + s * (1 - t) * src[r_top, c_right] \
                    + (1 - s) * t * src[r_bottom, c_left] \
                    + s * t * src[r_bottom, c_right]

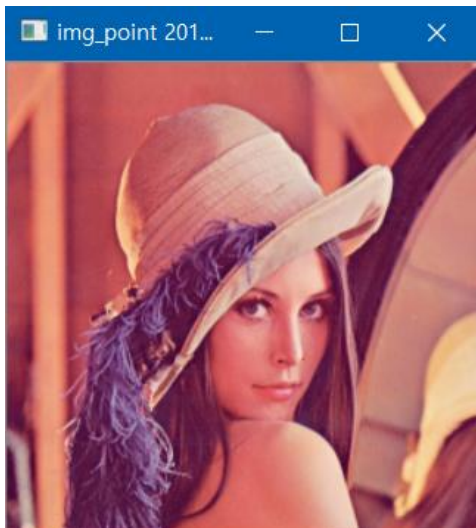
        dst_back_bilinear[row_, col_] = intensity

dst_back_bilinear = dst_back_bilinear.astype(np.uint8)

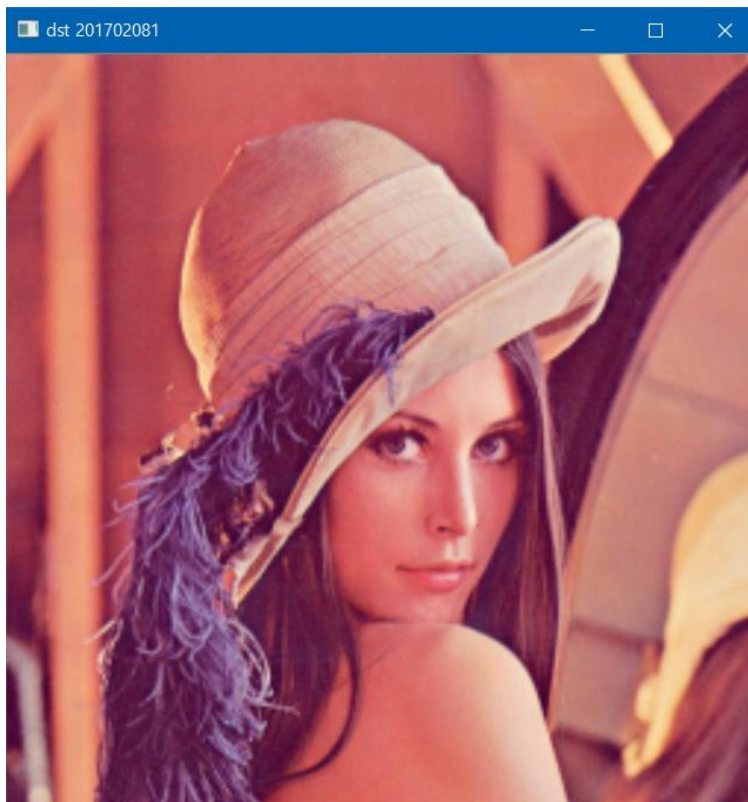
return dst_back_bilinear, M
```

- 이미지

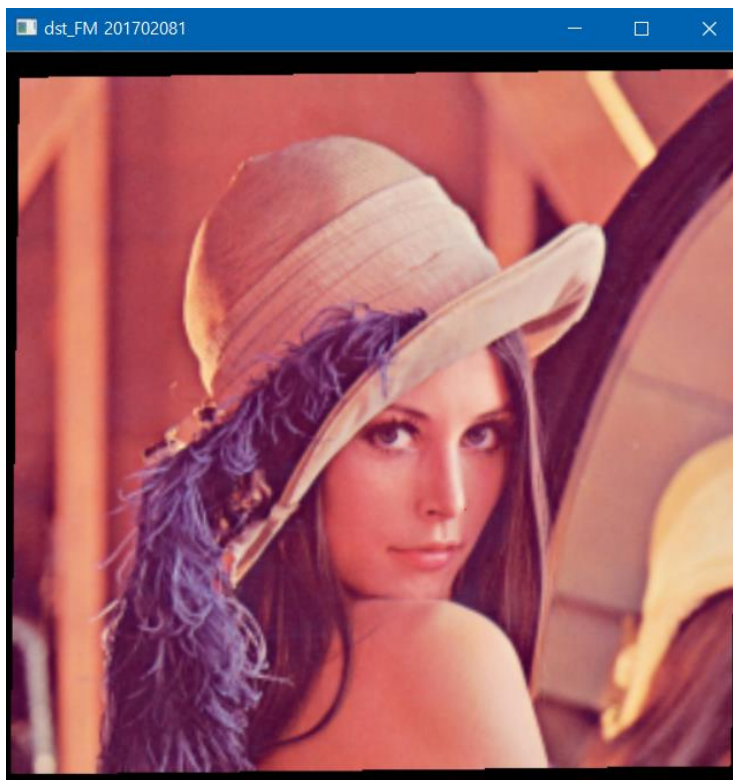
- ◆ img\_point



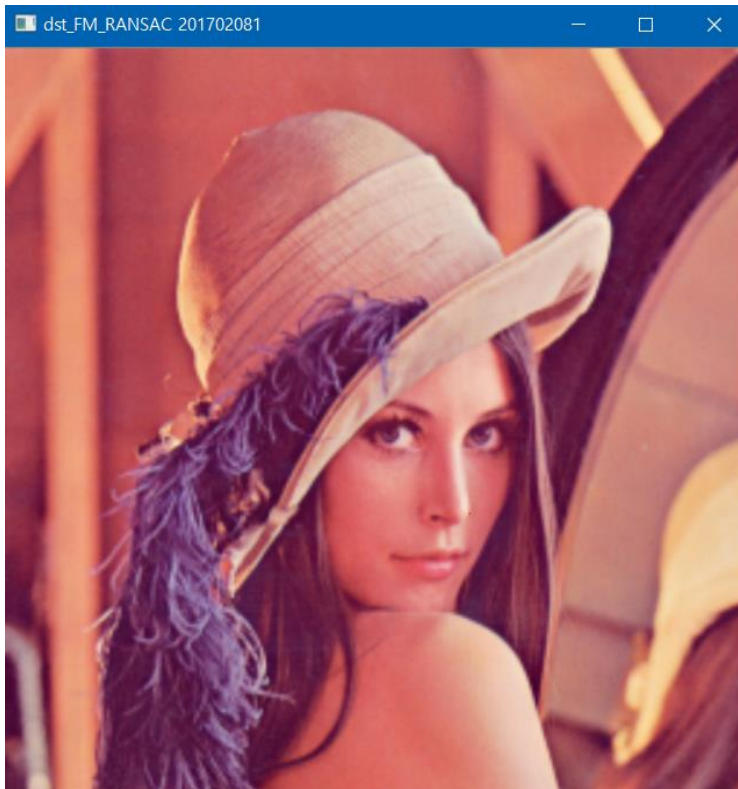
◆ dst



◆ dst\_FM



◆ dst\_FM\_RANSAC



◆ M, distance 결과

<pre>No RANSAC M [[ 1.94708535 -0.02389899 11.32937122]  [-0.02289127 1.88184856 19.09427529]  [ 0.          0.          1.          ]] RANSAC M [[ 1.99244876e+00 -1.96712989e-03 1.85309982e+00]  [-2.15749730e-03 2.01624469e+00 -1.91592106e+00]  [ 0.00000000e+00 0.00000000e+00 1.00000000e+00]]</pre>	<pre>M [[2 0 0]  [0 2 0]  [0 0 1]] M 역행렬 [[0.5 0. 0. ]  [0. 0.5 0. ]  [0. 0. 1. ]]</pre>	<pre>No RANSAC distance point : 317 319 3.1622776601683795 Use RANSAC distance point : 320 320 0.0</pre>
--	--	--

● 느낀 점

쉬어가는 느낌이라 좋았다

● 과제 난이도

이전 과제에서 많이 가져와서 할 만 했다