

컴퓨터비전 과제 6 보고서

2019203021

소프트웨어학부

이승헌

참고한 자료:

(<https://velog.io/@richpin/Computer-Vision-07-2D-Transformations#2d-transformations>)

Least Squares Error

$$E_{LS} = \sum_i \|f(x_i; p) - x'_i\|^2$$

Euclidean (L2) norm squared!

predicted location measured location

$\|x\| = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$

$$E_{LLS} = \sum_i |a_i x - b_i|^2$$
$$= \|Ax - b\|^2 \quad (\text{matrix form})$$

$$\begin{aligned} E_{LLS} &= \|Ax - b\|^2 \\ &= (Ax - b)^T (Ax - b) \\ &= ((Ax)^T - b^T)(Ax - b) \\ &= (x^T A^T - b^T)(Ax - b) \\ &= x^T A^T Ax - x^T A^T b - b^T Ax + b^T b \end{aligned}$$

그런데 여기서 E_{LLS} 값이 스칼라 값이니 모든 항, 즉 $b^T Ax$ 또한 스칼라 값이 되어야 한다.
스칼라 값의 Transpose 는 같기 때문에 이를 $b^T Ax = (b^T Ax)^T = x^T A^T b$ 이므로
최종적으로 식은 아래와 같이 나온다.

$$E_{LLS} = x^T (A^T A)x - 2x^T (A^T b) + \|b\|^2$$

최솟값이 나오기 위해서는 미분한 값이 0 이여야 한다.

$$\frac{dx}{d} E_{LLS} = 2(A^T A)x - 2(A^T b) = 0$$

이 식을 풀면 결과적으로 최소가 되기 위한 x 는 다음과 같다.

$$x = (A^T A)^{-1} A^T b$$

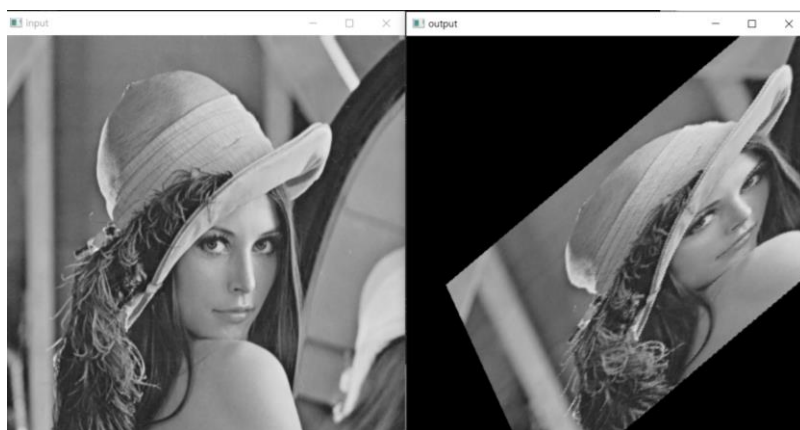
결론만 말하자면 변환 전 위치 p 의 전치행렬과 p 을 곱한 것에 대한 역행렬을 구하고,
 p 의 전치행렬과 변환 후 위치 x 를 곱한 것을 위에서 구한 역행렬과 곱해야 한다는 것이었다.
사람 손으로 푸는 문제였다면 M 의 역행렬을 그냥 구할 수는 없으므로 저렇게 해야겠지만,
opencv 의 invert 함수는 가역 행렬이 아니어도 역행렬을 구할 수 있었다.

그래서, 코드에서는 과제에서 주어진 8×6 p 행렬에 대해 6×8 p 역행렬을 구하고
 8×1 x 행렬과 그대로 곱하여 6×1 M 행렬을 구해서 변환을 구했다.

처음에는 다음과 같이 변수를 설정했다.

```
p =  
[173, 284, 0, 0, 1, 0]  
[0, 0, 173, 284, 0, 1]  
[477, 33, 0, 0, 1, 0]  
[0, 0, 477, 33, 0, 1]  
[248, 455, 0, 0, 1, 0]  
[0, 0, 248, 455, 0, 1]  
[553, 193, 0, 0, 1, 0]  
[0, 0, 553, 193, 0, 1]  
x=  
[  
100,  
100,  
412,  
100,  
100,  
412,  
412,  
412  
]
```

이렇게 설정하고 문제의 영상을 입력으로 넣었더니 문제의 영상에 적용된 변환을 한 번 더 한다는 느낌이 들었고, 목표 영상을 넣었더니 이렇게 됨을 알 수 있었다.



그래서, 다음에는 다음과 같이 변수를 설정했다.

p =

[100, 100, 0, 0, 1, 0]

[0, 0, 100, 100, 0, 1]

[412, 100, 0, 0, 1, 0]

[0, 0, 412, 100, 0, 1]

[100, 412, 0, 0, 1, 0]

[0, 0, 100, 412, 0, 1]

[412, 412, 0, 0, 1, 0]

[0, 0, 412, 412, 0, 1]

x=

[

173,

284,

477,

33,

248,

455,

553,

193

]

```

-0.001603
0.000000
0.001603
0.000000
-0.001603
0.000000
0.001603
0.000000
-----
-0.001603
0.000000
-0.001603
0.000000
0.001603
0.000000
0.001603
0.000000
-----
0.000000
-0.001603
0.000000
0.001603
0.000000
-0.001603
0.000000
0.001603
-----

```

```

0.000000
-0.001603
0.000000
-0.001603
0.000000
0.001603
0.000000
0.001603
-----
1.070513
0.000000
0.250000
0.000000
0.250000
0.000000
-0.570513
0.000000
-----
0.000000
1.070513
0.000000
0.250000
0.000000
0.250000
0.000000
-0.570513
-----

```

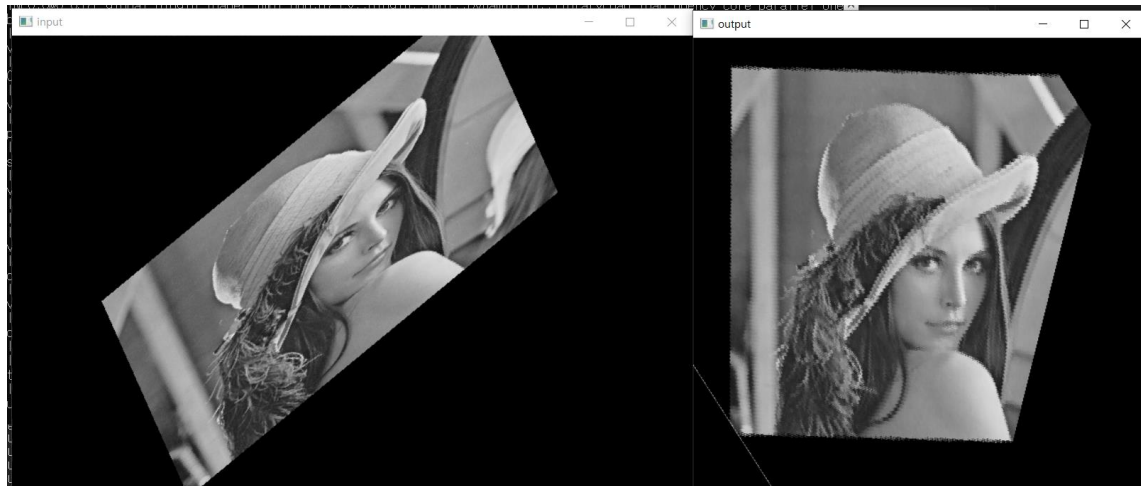
```

-----
0.975962
-----
0.241987
-----
-0.822115
-----
0.530449
-----
50.955128
-----
315.916667
-----

```

P의 6x8 역행렬 P' P'와 x를 곱한 6x1 행렬 M(위부터 m1, m2, m3, m4, t1, t2)

변환 결과



구한 계수를 통해 bilinear interpolation 한 결과이다.

출력에 대응하는 입력의 X 좌표 값은

$$(\text{출력 } x \text{ 좌표}) \times m1 + (\text{출력 } y \text{ 좌표}) \times m2 + t1$$

출력에 대응하는 입력의 Y 좌표 값은

$$(\text{출력 } x \text{ 좌표}) \times m3 + (\text{출력 } y \text{ 좌표}) \times m4 + t2$$

로 하여 계산했다.