

영상처리 실습 보고서

13주차: Backward warping

학번	201802170
이름	하 상 호

1. 과제의 내용

Backward warping 구현

2. 과제의 해결 방법

과제의 내용을 해결하기 위해 어떠한 방법을 사용했는지 자세하게 기술한다.

실습 및 이론 수업을 통한 것을 바탕으로 backward warping 을 구현한다.

```
x_ = x * inv_M[0, 0] + y * inv_M[0, 1] + inv_M[0, 2]
y_ = x * inv_M[1, 0] + y * inv_M[1, 1] + inv_M[1, 2]

s_f = x_ % 1
t_f = y_ % 1

x_ = int(x_)
y_ = int(y_)
```

Inverse matrix 를 이용하여 각각의 matrix 연산을 한 것을 구해야 한다.

예를 들어

M inv	temp	
[a, b, c]	[x,	[v1,
[a2, b2, c2]	y,	= v2,
[a3, b3, c3]]	1]	v3]

를 통해 연산을 진행할 때 $v1 = a * x + b * y + c$ / $v2 = a2 * x + b2 * y + c2$ 임을 알 수 있다. 여기서 v1 은 위 코드에서 x_ , v2 는 위 코드에서 y_ 가 된다.

위 변수들은 1 이상의 값이므로 % 1 을 통해 소수 값을 얻을 수 있다. 따라서 s_f , t_f 로 각각 변수에 저장해 준다. 위 과정을 해주는 이유는 bilinear interpolation 을 시행하기 위함이다.

또한 x_ y_ 는 float 이므로 int 를 통해 좌표값에 접근할 수 있도록 만들어 준다.

위 과정을 마친다면 5주차에 시행했던 bilinear interpolation 을 해준다.

```
if 0 < x_ + 1 < src_w and 0 < y_ + 1 < src_h:

    f_m_n = (1 - s_f) * (1 - t_f) * img[y_, x_]
    f_m_n1 = s_f * (1-t_f) * img[y_, x_ + 1]
    f_m1_n = (1 - s_f) * t_f * img[y_+1, x_]
    f_m1_n1 = s_f * t_f * img[y_ + 1, x_ + 1]

    dst[y, x] = f_m_n + f_m_n1 + f_m1_n + f_m1_n1
```

3. 결과물

결과물이 잘 보이도록 화면을 캡처해 보고서에 올린다.

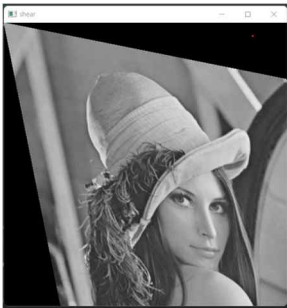
Org



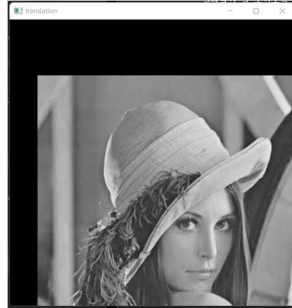
rotation



Shear



translation



Scaling (translation 과 크기 차이)

