

컴퓨터비전 보고서

20212998 공윤호

Q. 왜 세 개의 점을 각각 왼쪽 이미지와 오른쪽 이미지에서 찍어야 할까요? gui.py의 getAffineTransform 함수와 관련지어 설명하세요.

처음 과제를 수행하기 위해 양 쪽 사진의 점을 3개를 찍고 사진을 합성하면서 왜 하필 3개의 점을 찍는 것인지 그리고 왜 대응되는 점들을 양쪽 사진에 찍어야 하는지 궁금하였다. 이 궁금증의 힌트는 아핀변환에 있다는 설명을 보고 아핀변환에 대한 공부를 진행하였다.

2차원과 3차원에서 각각 2x2 행렬, 3x3 행렬로 된 선형 변환 행렬로 scaling, shear, rotation 등 형태를 변경할 수 있다. 그러나 앞에서 언급한 선형 변환으로는 평행이동이 불가능하다. 일반적인 행렬의 곱 만으로는 벡터간의 덧셈을 표현할 수 없기 때문이다. 벡터의 형태 변화와 함께 평행이동을 구현하는 방법으로 2x2 사이즈의 형태 변화 행렬(ex. scaling)을 우선 입력값 벡터에 곱해주고 그 값에 평행이동 시켜주는 값(bias)를 더해주는 방식이 있다. 그러나 이렇게 곱셈과 덧셈을 둘 다 하는 방식보다 더 편리하게 행렬곱 한 번으로 평행이동과 형태 변환을 하는 방식이 있다.

$$\begin{bmatrix} x_{new} \\ y_{new} \\ 1 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & b_1 \\ A_{21} & A_{22} & b_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

2차원 벡터에 평행이동을 시켜주는 값(bias)에 해당하는 차원(b1, b2)을 오른쪽에 추가하고 3행에 [0, 0, 1]을 추가하여 3x3 행렬을 만든다. 이러한 연산을 아핀 변환이라고 한다.

아핀 변환에 대한 이해를 토대로 getAffineTransform 함수와 그 함수를 리턴값으로 하는 함수 getMapping 함수에 대해 공부하였다. 이미지 합성을 수행하는 과정을 담은 gui.py에서 getMapping 함수는 이름 그대로 오른쪽 이미지를 왼쪽 이미지에 덮기 위해 특정 규칙, 즉 아핀변환을 리턴하는 함수이다. 그리고 자연스러운 두 이미지의 합성을 위해서는 오른쪽 이미지를 왼쪽 이미지 기준으로 변환하는 함수가 getAffinetransform이다. 이 아핀변환을 만들기 위해

서는 위의 이미지에 나온 2차원에서 형태변화를 야기하는 2x2 행렬 속 4개의 좌표(A11, A12, A21, A22)와 평행이동을 위한 2개의 좌표(b1, b2)가 필요하다.

두 개의 이미지를 자연스럽게 맵핑하기 위해서는 오른쪽 이미지의 특정 좌표(x,y)가 왼쪽 이미지의 특정 좌표(x', y')로 이동해야 한다. 이 것을 식으로 나타내면 다음과 같다.

$$x' = a*x + b*y + c$$

$$y' = d*x + e*y + f$$

위 두 방정식의 계수들이 바로 아핀변환을 완성하기 위한 미지의 6개의 점이 된다. 이 방정식을 풀어 6개의 계수들을 알기 위해서는 대응되는 왼쪽 이미지의 특징점 (x,y)와 오른쪽 이미지의 특징점(x', y')의 쌍이 총 3개가 필요하다. 따라서 우리가 점을 6개 찍게 되는 것이다.