

20233009 김난슬

왜 두 이미지에서 각각 세 개의 점을 찍어야 하나요? —

cv2.getAffineTransform 관점에서의 설명

2D 아핀 변환은 6개의 자유도(회전, 스케일, 전단, 평행이동)를 가지며 2×3 행렬로 표현된다. 한 쌍의 대응점은 x, y 두 식 (= 2개의 방정식)을 제공하므로, 6개의 미지수(행렬 4개 + 이동 2개)를 풀기 위해서는 최소 3쌍 (= 6식)의 대응점이 필요하다. 따라서 왼쪽/오른쪽 이미지에서 각각 3개의 대응점을 같은 순서로 찍어야 `cv2.getAffineTransform` 이 유일한 아핀 변환을 계산할 수 있는 것이다. 이때, 세 점은 서로 일직선이 아니어야 한다.(비공선)

gui.py의 흐름과 연결

gui.py에서 사용자는 왼쪽과 오른쪽 이미지에 대해 대응점 3개를 클릭한다. 코드에서는 이 점들을 `np.float32` 배열로 정리한 뒤, 다음과 같이 아핀 변환을 추정한다.

```
mapping = cv2.getAffineTransform(src_points, dst_points)
```

프로젝트 코드에서는 오른쪽 이미지의 점들을 `src_points`, 왼쪽 이미지의 점들을 `dst_points` 로 넣어 `mapping = cv2.getAffineTransform(right, left)` 를 얻고,

이어서

```
aligned_right = cv2.warpAffine(right_image, mapping, (w, h), borderMode=cv2.BORDER_REFLECT)
```

로 오른쪽 이미지를 왼쪽 이미지에 정렬하면 된다. 여기서 아핀 변환 행렬 `mapping` 을 신뢰성 있게 추정하려면 최소 3쌍의 정확한 대응점이 필요한 것이다.

왜 정확히 "3쌍"인가?

아핀 변환은 입력 좌표를 출력 좌표로 바꾸는 과정에서, 회전,스케일,전단 같은 선형 성분과 가로,세로 평행이동 성분을 함께 가진다. 이때 필요한 값(미지수)은 선형 성분 네 가지와 평행이동 두 가지를 합쳐 총 여섯 가지이다. 한 쌍의 대응점은 가로 방향과 세로 방향에서 각각 하나씩, 총 두 가지 조건을 제공한다. 따라서 여섯 가지 값을 정확히 정하려면 두 조건을 주는 점쌍이 세 개 필요하며, 그래서 최소 세 쌍의 대응점을 클릭해야 한다. 이 세 점은 서로 일직선이 아니어야 한다. 일직선에 가까우면 회전,전단,스케일을 서로 구분하기 어려워 계산이 불안정해질 수 있다.

주의: 세 점이 모두 일직선(collinear)인 경우에는 전단/스케일/회전 정보를 충분히 구별하지 못해 변환이 퇴화할 수 있다. 따라서 세 점은 서로 비공선이어야 한다.

아핀 변환과 동차 좌표계 관점

동차 좌표계에서는 아핀 변환을 하나의 3×3 표로 표현한다. 위부분의 2×2 영역은 회전,스케일,전단처럼 선형 성분을, 오른쪽 열의 두 값은 가로,세로 평행이동을, 마지막 행은 형태 유지를 위해 고정된 값을 담는다. 이렇게 표현하더라도 결국 필요한 값은 여섯 가지이며, 세 쌍의 대응점이 최소 요건이라는 사실은 변하지 않는다.