

## 1. 실습 개요

본 실습의 주제는 Image Transformation 및 Video Stabilization이다. 실습1에서는 Feature Detection을 이용해 연속된 프레임 사이의 Homography를 추정하고, 이 누적 Homography의 역행렬을 각 프레임에 적용하여 기본적인 영상 안정화(Stabilization)를 수행하였다. 하지만 이 방식은 카메라 Motion을 그대로 반대로 적용할 뿐이므로, 카메라 흔들림에 포함된 고주파(Jitter) 성분까지 그대로 반대로 보상하여 결과 영상에 잔잔한 떨림이 남는다는 한계가 있다.

실습2에서는 실습1의 코드(feature\_detection\_실습\_1.cpp)를 기반으로, 강의자료에서 제시된 Stabilization 과정 중 Path Planning 단계를 추가 구현한다. 구체적으로는 프레임별로 추정된 Homography 시퀀스를 시간축 그래프로 보고, 이를 Gaussian Blur로 부드럽게(저역통과 필터링) 만든 후, 이 Smooth Path와 원래 카메라 Path의 차이를 영상에 적용하여 보다 안정적인 동영상을 얻는 것이 목표이다.

## 2. 코드 구현 설명

### 2.1 Motion Estimation (Feature + Homography)

Motion Estimation 단계는 실습1과 동일하게 SIFT 특징점 + Homography 추정으로 구성하였다.

1. 각 프레임에 대해 SIFT로 keypoint 및 descriptor를 추출한다.
2. 이전 프레임과 현재 프레임의 descriptor를 FLANN 기반 매처로 매칭하고, Lowe의 비율 테스트를 통해 좋은 매칭 쌍만 선택한다.
3. 선택된 매칭 쌍에 대해 RANSAC을 적용하여 이상치를 제거하면서 Homography 행렬을 추정한다.
4. 충분한 매칭 쌍이 없거나 Homography 계산에 실패하면 항등 행렬(Identity)을 사용하여 Motion이 없다고 가정한다.

코드 상에서는 `estimateHomography()` 함수가 이 과정을 수행하며, 입력은 그레이스케일 두 프레임, 출력은  $3 \times 3$  Homography 행렬이다.

### 2.2 카메라 궤적(Camera Trajectory) 계산

Stabilization에서는 단순히 프레임 간 상대 Motion만이 아니라, 시간에 따라 누적된 카메라 위치(카메라 궤적)가 중요하다. 이를 위해 다음과 같이 Homography를 누적한다.

- 첫 프레임을 기준 좌표계로 두고,  $H_0 = I$ (항등 행렬)로 설정한다.
- $H_i$ 를 프레임  $i-1$ 에서  $i$ 로 가는 Homography라고 할 때
  - 누적 Homography  $C_i = H_i * C_{i-1}$ 으로 정의하여, 프레임 0에서  $i$ 까지의 카메라 Motion을 표현한다.

코드에서는 `buildCameraTrajectory()` 함수가 이 역할을 담당한다. 이 함수는 전체 프레임 시퀀스를 입력 받아, 각 프레임에 대응하는 누적 Homography 목록  $H\_list[i]$ 를 반환한다. 이  $H\_list$ 가 곧 원본 카메라 Path에 해당한다.

### 2.3 Path Planning: Homography 시퀀스 Gaussian Smoothing

강의자료에서 Path Planning 단계는 카메라 Motion 그래프를 완만하게 만드는 과정으로 설명된다.

Translation만 고려하는 경우에는 X, Y 이동량에 대해 각각 그래프를 만들고 저역통과 필터(예: Gaussian Filter)를 적용한다. Homography의 경우에는 8개의 자유도를 가지므로, 각 요소를 따로 그래프로 보고 동일한 작업을 수행할 수 있다.

이를 코드로 구현한 것이 `smoothHomographies()` 함수이다.

1. Homography 시퀀스  $H_s$ 에서 각 행렬의
  - $H(0,0), H(0,1), H(0,2), H(1,0), H(1,1), H(1,2), H(2,0), H(2,1)$8개 요소를 추출하여 (프레임 수 x 8) 크기의 행렬  $data$ 에 저장한다.

2. 이 `data` 를 시간축 방향(세로 방향)으로 GaussianBlur 처리한다.
  - 커널 크기 `kernelSize` 가 클수록 더 긴 구간을 평균 내므로 카메라 Path가 더 부드러워지고
  - `sigma` 가 클수록 고주파 성분이 더 강하게 억제되어 미세한 흔들림이 크게 줄어든다.
3. Blur 처리된 `data` 를 다시 Homography 행렬로 복원하면서, `H(2,2)` 는 1로 고정한다.

결과적으로 `H_smooth[i]` 는 시간적으로 부드럽게 만든 Smooth Camera Path를 나타내며, 원래의 누적 Homography `H_list[i]` 와 비교했을 때 고주파 성분이 제거된 형태이다.

## 2.4 Warping: Path 보상(Compensation)을 통한 Stabilization

강의자료의 Warping 부분에서는, 우리가 만든 완만한 그래프(카메라 Motion)를 실제 영상에 적용하는 방법을 설명한다. Translation만 고려할 때는 `Smooth Path - Original Path` 를 실제 보상 모션으로 적용하고, Homography의 경우에는 `Smooth_H * Original_H-1` 를 적용해야 한다고 명시되어 있다.  
코드에서는 각 프레임 i에 대해 다음과 같이 보상 Homography를 계산한다.

- `H_list[i]` : 원본 카메라 누적 Homography (Original Path)
- `H_smooth[i]` : Gaussian smoothing을 거친 Smooth Path
- 보상 Homography `H_comp = H_smooth[i] * H_list[i].inv()`

이 `H_comp` 를 `warpPerspective()` 에 전달하여 각 프레임을 새로운 좌표계로 변환하면, 결과적으로 카메라 궤적이 `H_smooth` 에 가깝도록 보정된 안정화 영상을 얻을 수 있다.

코드에서는 `warpPerspective()` 호출 시 보간 방식은 `INTER_LINEAR`, 경계는 `BORDER_REPLICATE` 를 사용하였다. 또한 `VideoWriter` 를 통해 `stabilized_practice2.avi` 라는 이름으로 결과 영상을 저장하며, `imshow` 로 원본/안정화 화면을 동시에 출력하도록 구성하였다.

## 3. Stabilization 실습2 내용 분석

### 3.1 Stabilization 전체 파이프라인 정리

강의자료에서 제시한 Stabilization의 전체 과정은 다음 3단계로 요약된다.

1. Motion Estimation
  - Feature & Homography, Optical Flow, Block Matching 등 다양한 방식으로 카메라 Motion을 추정한다.
  - 본 실습에서는 SIFT 특징점과 RANSAC 기반 Homography를 사용하여 프레임 간 Motion을 추정하였다.
2. Path Planning
  - 추정된 카메라 Motion을 시간축 상의 그래프로 보고, 이 그래프를 얼마나 완만하게(Smooth) 만들지 결정한다.
  - Gaussian Filter나 주파수 영역에서의 Low-pass filter 등을 적용하여 고주파 성분(빠른 흔들림)을 제거한다.
3. Warping
  - Path Planning을 통해 얻은 Smooth Path와 원래 Path의 차이를 각 프레임에 적용한다.
  - Translation 기반 모델에서는 `Smooth Path - Original Path`, Homography 기반 모델에서는 `Smooth_H * Original_H-1` 를 실제 보상 모션으로 사용한다.

실습1은 이 중 Motion Estimation + Warping(Original Path의 역행렬만 적용) 단계까지 구현한 것이고, 실습2는 여기에 Path Planning(그래프 smoothing)을 추가한 것이라 볼 수 있다.

### 3.2 Path Planning에서 Gaussian Blur의 역할

Homography 시퀀스를 Gaussian Blur로 처리하는 것은, 각 Homography 요소를 시간에 대한 신호로 간주하고 여기에 저역통과 필터(Low-pass Filter)를 적용하는 것과 같다.

- 카메라 흔들림에서 고주파 성분은 손떨림이나 작은 진동처럼 짧은 시간 간격에 발생하는 빠른 변화에 해당한다.
- Gaussian Blur는 인접한 프레임들의 값을 가중 평균하여 이러한 빠른 변화를 완화하고, 비교적 느린 Motion(저주파)만 남긴다.
- 결과적으로 Smooth Path는 원래 Path에 비해
  - 급격한 이동이 줄어들고
  - 전반적으로 완만한 Camera trajectory를 형성하여 안정화된 영상을 만들어 내는 기반이 된다.

실습2에서 `kernelSize`와 `sigma`를 조절하면서 다양한 결과를 확인할 수 있다.

- `kernelSize` 증가
  - 더 긴 시간 구간을 평균 내므로, 장시간에 걸친 흔들림까지 크게 완화된다.
  - 하지만 너무 크게 설정하면 카메라가 원래 움직여야 할 큰 Motion까지 과도하게 평탄화하여, 영상이 느리게 따라가는 듯한 느낌을 줄 수 있다.
- `sigma` 증가
  - 동일한 커널 크기 내에서 고주파 성분을 더 강하게 억제하여 세밀한 흔들림이 더 잘 제거된다.
  - 너무 크게 설정하면 작은 Motion까지 모두 제거되어, 원래 의도된 카메라 움직임(팬/틸트 등)이 부자연스럽게 변형될 수 있다.

### 3.3 Homography 기반 Path Planning의 특징

Translation만을 고려하는 모델과 달리 Homography는 회전, 스케일 변화, 시점 변화 등 더 복잡한 카메라 Motion을 표현할 수 있다. 이 장점 덕분에

- Hand-held 카메라의 2D 평면 상 흔들림뿐 아니라,
- 약간의 시야 변화나 줌 효과까지도 어느 정도 보정할 수 있다.

그러나 Homography 기반 Path Planning은 다음과 같은 특성을 가진다.

- 각 요소를 독립적인 시계열로 보고 smoothing을 수행하기 때문에, 이론적으로는 기하학적 제약(예: 정규화된 Rotation/Scale)이 완전히 보장되지 않을 수 있다.
- 그럼에도 불구하고 실제 영상에서는 Gaussian smoothing이 시각적으로 자연스러운 Stabilization을 만드는 데 충분히 효과적이며, 구현이 간단하다는 장점이 있다.

## 4. 결과 및 고찰

구현한 실습2 코드를 사용해 Hand-held로 촬영한 흔들리는 영상을 입력으로 사용하면, 다음과 같은 경향을 관찰할 수 있다.

1. 실습1 방식(단순 역행렬 적용)에 비해, 실습2의 Path Planning을 적용한 결과는
  - 작은 떨림이 상당 부분 제거되고
  - 카메라 움직임이 보다 부드럽고 연속적으로 느껴진다.
2. 커널 크기와 sigma를 너무 작게 설정하면
  - Smooth Path가 원본 Path와 크게 다르지 않아, 안정화 효과가 제한적이다.
3. 반대로 값을 너무 크게 설정하면

- 카메라의 큰 이동이 과도하게 평탄화되어, 화면이 천천히 따라오는 느낌 또는 과도하게 부드러운(느린) 카메라처럼 보일 수 있다.
- 또한 보상 Motion이 커질수록 warp 시에 프레임 가장자리에서 검은 영역 또는 늘어난 패턴이 발생할 수 있어, 후처리로 크롭(crop)이나 스케일 조정이 필요해질 수 있다.

4. Homography 기반 Stabilization은 나무, 건물 등 정적인 배경 물체를 기준으로 카메라 Motion을 보정 한다. 따라서 프레임 내에서 큰 물체가 빠르게 움직이는 경우(예: 사람이 화면을 가로지르는 경우), 그 움직임이 안정화 결과에 영향을 줄 수 있다는 점도 고려해야 한다.

종합하면, 실습2는 실습1의 기본 Stabilization 파이프라인에 Path Planning(카메라 궤적 smoothing)을 추가 함으로써, 실제 활용 가능한 수준의 영상 안정화 결과를 얻는 과정을 보여준다. 특히 Homography 시퀀스를 Gaussian Blur로 처리하는 간단한 방법만으로도, 카메라의 불규칙한 흔들림을 효과적으로 줄일 수 있음을 확인할 수 있다. 실제 응용에서는 촬영 환경과 원하는 안정화 정도에 맞추어

- Feature 추정 방식,
- Path Planning 필터 종류 및 파라미터,
- Warping 후 크롭/리사이즈 전략

등을 적절히 조합하여 품질과 계산량 사이의 균형을 맞추는 것이 중요하다.