

# 파노라마 스티칭 구현 및 시행착오 분석

## 0. 실행방법

1) 폴더의 `Execute.py` 열기

2) `IMG_DIR`를 `./images` → `./sampleset*` 로 바꾸면서 `Run and Execute`

## 1. 개발 환경 및 개요

본 프로젝트는 다수의 연속된 이미지를 하나의 파노라마 이미지로 결합하는 알고리즘을 직접 구현하는 것을 목표로 하였다. 단순히 기존 라이브러리를 활용하여 결과를 생성하는 것이 아니라, 영상 처리 및 컴퓨터 비전 강의에서 다룬 핵심 개념들을 실제 코드로 구현하고, 그 과정에서 발생하는 실패 원인과 한계를 분석하는 데 중점을 두었다.

개발은 Python 3 환경에서 주피터 노트북을 사용하여 진행되었다. 수치 연산과 행렬 계산에는 NumPy를 사용하였고, 이미지 로딩 및 저장에는 PIL(Pillow)을 활용하였다. 중간 결과 및 최종 결과의 시각화는 Matplotlib을 통해 수행하였다. Non-Maximum Suppression 구현을 보조하기 위해 SciPy의 `maximum_filter`를 사용하였다.

## 2. 이미지 전처리 단계

파노라마 스티칭의 첫 단계는 입력 이미지에 대한 전처리였다. Harris Corner Detector는 이미지의 미분 정보를 직접적으로 사용하므로, 입력 영상에 포함된 노이즈가 그대로 반영될 경우 불안정한 특징점이 다수 검출될 수 있다. 이러한 문제를 방지하기 위해 모든 입력 이미지는 먼저 Grayscale로 변환되었으며, 이후 Gaussian Smoothing을 적용하였다.

Gaussian 필터는 고주파 노이즈를 억제하면서도 구조적인 에지 정보를 유지하는 역할을 한다. 본 프로젝트에서는  $\sigma=1.0$ 을 사용하였으며, 이는 미분 연산 이후에도 비교적 안정적인 그래디언트 분포를 얻을 수 있는 값이었다. 색상 정보는 기하학적 정합에는 필수적이지 않다고 판단하여 사용하지 않았으며, 이는 파이썬 + 주피터 노트북 개발 환경 특유의 많은 연산량과 부족한 메모리 사용량을 줄이는 데도 기여하였다.

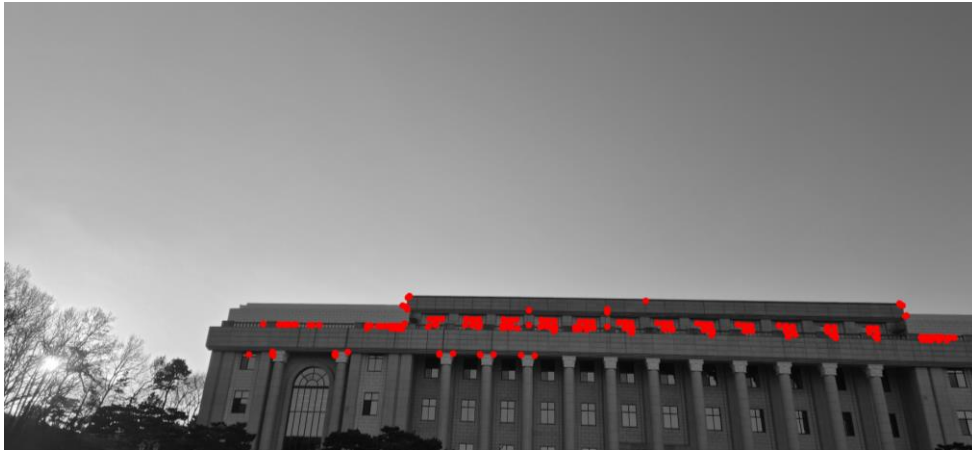
## 3. 특징점 검출: Harris Corner Detector

### 3.1 기본 Harris Detector 구현

전처리된 이미지에 대해 중앙 차분 방식을 사용하여  $x, y$  방향의 이미지 그래디언트를 계산하였다. 이를 바탕으로 Structure Tensor를 구성하였으며, 각 텐서 성분에 다시 한 번 Gaussian Smoothing을 적용하였다. 이는 국소적인 노이즈에 의해 Harris Response가 과도하게 커지는 현상을 방지하기 위함이다.

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad R = \det(M) - k \cdot (\text{trace}(M))^2$$

다음 공식을 사용하여 Harris Corner Response를 계산하였으며 k 값은 일반적으로 널리 사용되는 0.04를 채택하였다.

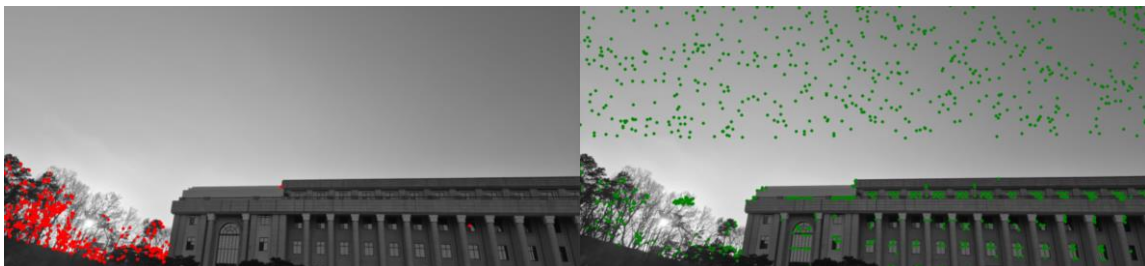


### 3.2 Non-Maximum Suppression 적용

Harris Response 맵에서 단순히 임계값만 적용할 경우, 위의 사진과 같이 하나의 실제 코너 주변에 여러 개의 후보 점이 동시에 검출되는 문제가 발생하였다. 이를 해결하기 위해 Non-Maximum Suppression을 적용하여, 국소 영역 내에서 가장 큰 응답값을 갖는 점만을 남기도록 하였다. 이 과정은 특징점의 공간적 중복을 제거하고, 이후 매칭 단계의 계산량을 줄이는 데 중요한 역할을 하였다.

### 3.3 나무 영역에서 발생한 실패와 ROI 도입

초기 구현 결과, 건물보다 나무 영역에서 훨씬 많은 특징점이 검출되는 문제가 발생하였다. 이는 나무가 가진 복잡한 고주파 텍스처 특성으로 인해 Harris Response가 과도하게 커졌기 때문이다. 그 결과, 실제 스티칭에 유의미한 건물의 코너보다 나무 영역의 특징점이 우선적으로 선택되는 현상이 발생하였다.



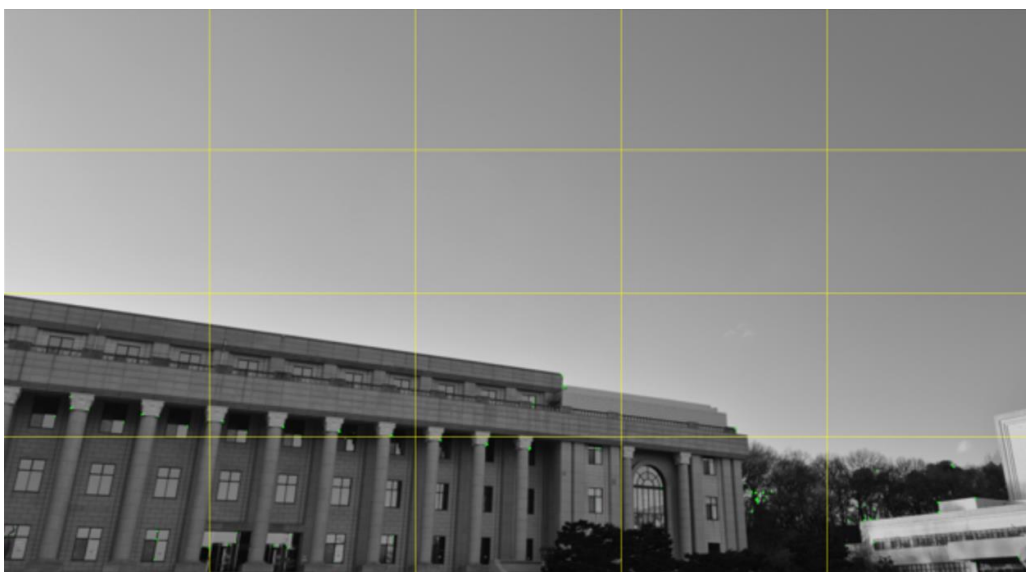
이를 해결하기 위해 관심 영역(ROI)을 도입하였다. 이미지의 하부 및 중앙 영역, 즉 건물이 위치한 영역만을 대상으로 특징점을 검출하도록 제한함으로써, 하늘이나 숲 영역에서 발생하는 불필요한 특징점을 사전에 배제하였다.



### 3.4 Local Percentile Thresholding 도입



ROI(관심 영역) 설정에도 불구하고, 이미지 내 특정 영역(예: 복잡한 텍스처가 많은 부분)에만 특징점이 과도하게 밀집되는 현상이 여전히 문제로 남아있었다. 이러한 불균형한 분포는 추후 RANSAC을 이용한 호모그래피 추정 시, 특정 영역에 대한 과적합을 유발하여 전체적인 정렬 안정성을 떨어뜨리는 주요 원인이 되었다. 따라서, 이를 해결하기 위해 이미지 전체에 걸쳐 특징점을 고르게 분포시킬 수 있는 Adaptive Thresholding 기법의 도입이 필요하였다.



따라서, Local Percentile Thresholding 방식을 도입하여, 이미지의 국소적인 특성을 반영한 임계값 적용을 수행하였다. 먼저, 전체 Harris Response Map  $R$ 을  $N \times M$  개의 균일한 격자(Grid)로 분할하고, 각 격자 내부의  $R$  값들을 독립적으로 계산한다. 다음으로, 각 격자 내에서 상위 0.5%에 해당하는  $R$  값을 가진 픽셀만을 특징점 후보로 선택한다.

이러한 방식을 통해 특징점이 많은 복잡한 영역에서는 높은 임계값이, 특징점이 적은 단순한 영역에서는 낮은 임계값이 자동으로 적용되며, 결과적으로 이미지의 전 영역에서 특징점이 균일하게 추출되도록 유도하여 매칭의 강인함을 확보하였다.

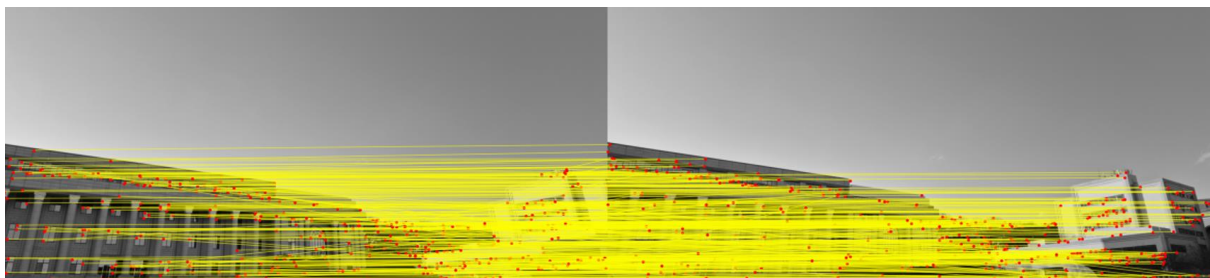
## 4. 특징점 매칭을 위한 패치 기반 접근

### 4.1 패치 추출 및 정규화

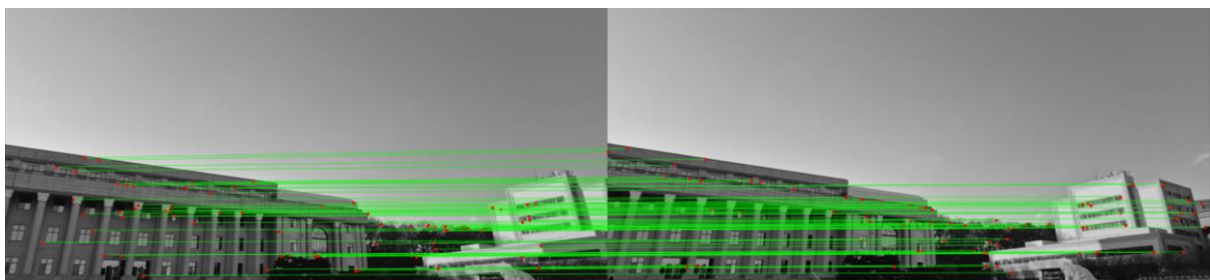
선별된 특징점 각각에 대해 주변의 국소 정보를 담은 정사각형 패치를 추출하였다. 패치 크기는  $15 \times 15$ 로 설정하였으며, 이는 국소적인 형태 정보를 충분히 포함하면서도 계산 비용이 과도하지 않은 절충점이었다. 추출된 패치는 조명 변화에 대한 민감도를 줄이고, 이후 SSD 기반 거리 계산의 안정성을 높이기 위해 Z-Score 정규화를 수행하였다.

### 4.2 SSD 기반 매칭과 Ratio Test

다음으로 모든 패치 쌍에 대해 SSD(Sum of Squared Differences)를 계산하여 유사도를 평가하였다. 그러나 단순히 가장 가까운 매칭만을 선택할 경우, 반복 패턴이 많은 건물 창문 구조로 인해 잘못된 매칭이 다수 발생하였다. 이러한 Outlier들이 RANSAC 단계로 넘어갔을 때, Inlier 비율(Ratio)을 떨어트려 정확한 모델 추정을 방해할 것으로 생각되었기 때문에 대응점의 질을 높이기 위하여 거리 기반의 Ratio Test로 필터링할 필요가 있었다. Ratio Test를 도입한 결과, 헛갈리는 창문 매칭들은 모두 탈락되고, 모양이 독특하여 확실히 구분되는 특징점들만 살아남았다. 즉, 매칭 개수는 확연히 줄었으나 정확도는 상승하였다.



Ratio\_Test 전



Ratio\_Test 후

## 5. RANSAC을 이용한 Homography 추정

### 5.1 DLT (Direct Linear Transformation) 기반 추정

매칭된 특징점 쌍 간의 사영 변환 관계를 정의하기 위해  $3 \times 3$  호모그래피 행렬  $H$ 를 추정하였다. 이를 위해 DLT 알고리즘을 사용하여  $\mathbf{x}' = H\mathbf{x}$  관계를 선형 연립방정식  $A\mathbf{h} = \mathbf{0}$  형태로 변환하였다. 이때, 잡음이 포함된 시스템에서 최적해를 구하기 위해 SVD (Singular Value Decomposition)를 수행하였으며, 행렬  $A^T A$ 의 최소 고유값에 대응하는 고유벡터를 최종  $H$ 로 선정하는 최소자승법 (Least Squares)을 적용하였다.

### 5.2 RANSAC 알고리즘을 통한 Outlier 제거

단순 최소자승법은 모든 데이터의 오차 제곱합을 최소화하려 하므로, 소수의 Outlier에도 모델 전체가 크게 왜곡되는 문제가 있었다. 특히, 다수의 Outlier가 오차 계산에 막대한 영향을 미쳐 변환 행렬이 수치적으로 불안정해지는 현상이 발생하였으며, 이로 인해 단순한 정렬 오차를 넘어, 이미지가 방사형으로 길게 늘어지거나 비정상적인 스케일로 투영되어 원본의 기하학적 구조가 완전히 붕괴되는 현상이 관찰되었다.



이를 해결하기 위해 RANSAC (Random Sample Consensus) 알고리즘을 사용하였다. 먼저, 전체 매칭 쌍 중 최소 개수인 4개의 점 쌍을 무작위로 선택하여 임시  $H$ 를 계산하여 원본 좌표를 투영 ( $Hx$ )한 뒤, 목표 좌표와의 Euclidean Distance를 계산하였다. 결과에서는 투영 오차가 임계값 (경험적; 4.0 pixel) 이내인 점들을 Inlier로 간주하였으며, 위 과정을 5,000회 반복하여 가장 많은 Inlier를 확보한  $H$ 를 최종 호모그래피로 채택하였다.

결과적으로, 구조적 정합성이 크게 개선되었으며, 건물의 직선 성분과 경계면이 정확하게 일치함을 확인하였다.



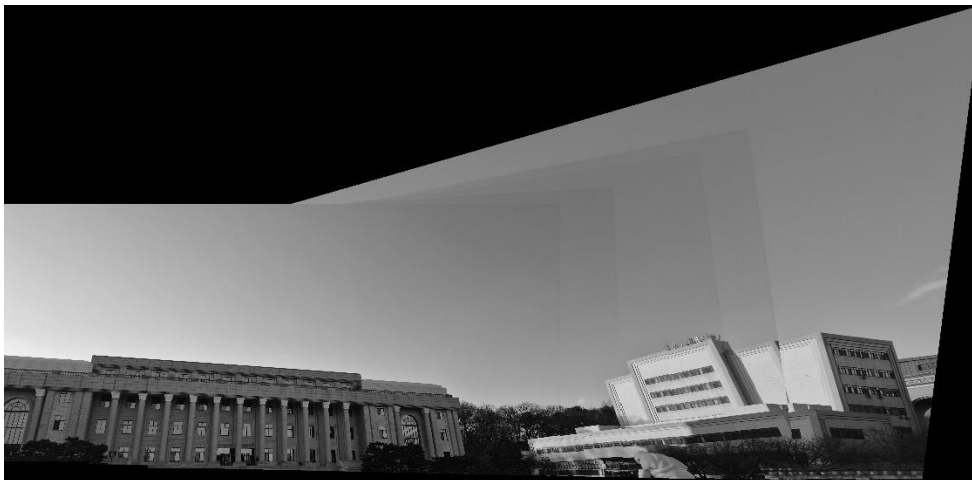
## 6. 스티칭 전략과 좌표계 안정화

### 6.1 기존 방식의 한계: 단방향 누적 왜곡

초기에는 첫 번째 이미지(Image 0)를 기준으로 순차적으로 이미지를 이어 붙이는 단방향 체인 방식을 시도하였다. 이 방식은 인접한 이미지 간의 호모그래피  $H_{\{i, i+1\}}$ 를 계속해서 곱해나가는 방식으로, N번째 이미지의 글로벌 호모그래피  $H_{\{N\}}$ 은 다음과 같이 계산된다.

$$H_{N \rightarrow 0} = H_{1 \rightarrow 0} \times H_{2 \rightarrow 1} \times \cdots \times H_{N \rightarrow N-1}$$

그러나, 이 과정에서 각 단계의 미세한 추정 오차가 누적될 뿐만 아니라, 사영 변환이 거듭될수록 이미지의 스케일이 비정상적으로 축소되거나 과도하게 늘어지는 심각한 왜곡이 발생하였다. 특히 후반부 이미지일수록 원본의 형태를 알아볼 수 없을 정도로 찌그러지는 현상이 관찰되었다.



단방향 스티칭에서 발생한 왜곡 현상

## 6.2 Center Anchor 구현 및 수치적 안정화

이러한 누적 오차를 최소화하기 위해 Center Anchor Strategy를 도입하였다. 즉, 전체 시퀀스(총 10장)의 중심에 위치한 4번 이미지를 기준 좌표계로 설정하고, 이를 중심으로 좌우로 뻗어나가는 구조로 변환 행렬을 재설계하였다.

우측 방향 (Image 5 ~ 9)은 기존과 동일하게 순방향 행렬 곱셈을 수행하지만, 좌측 방향 (Image 3 ~ 0)은 기준점인 4번으로 돌아오기 위해 역행렬을 계산하여 누적한다. 이 방식을 통해 오차가 양방향으로 분산됨에 따라 파노라마의 전체적인 기하적 균형성을 확보할 수 있었다.



중심 기준 스티칭 (testing0~4)

## 6.3 전역 좌표 보정 및 역방향 매핑



Center Anchor 전략을 통해 투영 왜곡은 완화되었으나, 다수의 이미지가 누적되어 연결될수록 일부 영상의 투영 좌표가 음수 영역으로 이동하거나 출력 시야 범위를 벗어나는 문제가 발생하였다. 이를 해결하기 위해, 3단계로 출력 좌표계를 정규화하는 과정을 도입하였다.

먼저, 계산된  $H_{global}$ 을 이용하여 모든 입력 이미지의 네 개 코너 좌표를 출력 평면으로 투영함으로써, 전체 파노라마가 차지하는 공간적 범위를 추정한다. 이후, 투영된 모든 코너 좌표 집합으로부터 최소값 ( $x_{min}, y_{min}$ )과 최대값 ( $x_{max}, y_{max}$ )을 계산하여 출력 영상의 bounding box를 정의한다. 마지막으로, 음수 좌표 영역을 제거하고 모든 투영 결과가 좌표계의 양수 부분에 위치하도록

록 하기 위해, 전역 평행 이동 행렬  $T_{shift}$  를 구성하고 이를 최종 호모그래피에 결합한다.

$$T_{shift} = \begin{bmatrix} 1 & 0 & -x_{min} + \text{padding} \\ 0 & 1 & -y_{min} + \text{padding} \\ 0 & 0 & 1 \end{bmatrix}, \quad H_{final} = T_{shift} \times H_{global}$$

이 과정을 통해 모든 영상은 공통된 출력 좌표계 내에서 안정적으로 배치될 수 있었다.

#### 6.4 Inverse Mapping 및 Bilinear Interpolation

이미지 변환(Warping) 과정에서 입력 좌표를 출력 좌표로 보내는 Forward Mapping 방식을 사용할 경우, 확대된 영역에서 픽셀 간 간격이 벌어져 Hole이 발생하는 것으로 알려져 있으므로 이를 방지하기 위해 Inverse Mapping 방식을 채택하였다.

먼저, 출력 이미지(캔버스)의 각 픽셀  $(x', y')$ 에 대해 역행렬  $H^{-1}$ 를 곱하여 원본 이미지의 대응 좌표  $(x, y)$ 를 역추적한다. 이 결과 좌표는 대부분 실수가 되므로, 인접한 4개 정수 픽셀의 값에 따라 가중 평균하는 Bilinear Interpolation을 구현하여 픽셀 값을 결정하였다. 이를 통해 커다란 왜곡 현상 없이 비교적 부드러운 파노라마 결과를 얻을 수 있었다.

### 7. 블렌딩 및 경계선 처리

#### 7.1 단순 덮어쓰기의 문제점



초기에는 픽셀 값을 단순히 덮어쓰거나 평균을 내는 방식을 사용하였으나, 이는 이미지 간의 노출 차이와 사진 촬영 시 발생한 미세한 정렬 오차로 인해 접합부에 뚜렷한 경계선이 남는 문제를 야기하였다. 특히 하늘과 같이 텍스처가 적은 영역에서는 이러한 불연속성이 더욱 두드러지게 관찰되었다.

#### 7.2 거리 기반 가중치 마스크

이를 해결하기 위해 각 픽셀의 위치에 따라 기여도를 달리하는 가중치 마스크(Weight Mask)를 생성하였고, 마스크의 값  $W(x, y)$ 는 해당 픽셀에서 가장 가까운 이미지 경계까지의 L1 거리



(Manhattan Distance)로 정의하였다.

$$W(x, y) = \min(x, W - 1 - x) \times \min(y, H - 1 - y)$$

이 수식을 통해 이미지 중심부로 갈수록 가중치가 1.0에 가까워지고, 가장자리(Edge)로 갈수록 0.0으로 부드럽게 감소하는 피라미드 형태의 마스크를 생성하였다. 이는 영상이 겹치는 영역에서 (제공된 연산으로 인한) 비선형적 감쇠를 유발하는 L2 거리와 달리, 영상이 서서히 섞이는 Feathering 효과를 제공한다.

### 7.3 누적 버퍼를 이용한 가중합

여러 장의 이미지를 순차적으로 합성하기 위해 두 개의 누적 버퍼(Accumulation Buffer)를 활용한 가중 평균 방식을 사용하였다. 먼저, Numerator Buffer (B\_num)을 사용하여 워핑된 이미지 값과 가중치를 곱하여 누적인 뒤, Denominator Buffer (B\_den)로 가중치 자체를 누적한다.

```
panorama_num[mask] += (warped_img * warped_weight)[mask]
panorama_den[mask] += warped_weight[mask]
final_panorama = panorama_num / panorama_den
```

이 방식은 이미지가 3장 이상 겹치는 영역에서도 (이미지A\*가중치 + 이미지B\*가중치 + 이미지C\*가중치) / (가중치합) 형태의 가중 평균으로 자연스러운 블렌딩을 가능하게 한다.

## 8. 최종 결과 및 고찰

### 8.1 최종 결과물 분석



본 프로젝트에서는 Harris Corner Detection을 시작으로 RANSAC 기반의 호모그래피 추정, Center Anchor 전략, 그리고 Feather Blending에 이르는 전체 파노라마 스티칭 파이프라인을 직접 구현하였다. 최종적으로 생성된 파노라마 이미지는 초기 단방향 스티칭 시도에서 발생했던 심각한 누적 왜곡이 효과적으로 제거되었고 특히 건물의 수직, 수평 라인이 비교적 잘 유지되며, 전체적인 시야각이 자연스럽게 확장되었다. 또한, 단순 덮어쓰기 방식에서 나타났던 경계선들은 가중치 마스크와 누적 버퍼를 활용한 Blending을 통해 완전히 제거되었다.

## 8.2 한계점 및 개선 방향

그럼에도 불구하고, 몇 가지 한계점과 추가적인 개선 가능성이 존재한다.

- (1) Parallax나 움직이는 물체가 있는 경우, 완벽한 정합이 어려워 잔상이 발생할 수 있다. 이를 해결하기 위해 Seam Carving이나 Graph Cut과 같은 고급 블렌딩 기법의 도입을 고려할 수 있다.
- (2) Feather Blending이 경계선을 부드럽게 처리해주지만, 이미지 전체의 노출 차이를 근본적으로 보정하지는 못한다. Gain Compensation이나 Multi-band Blending을 통해 전체적인 밝기와 색조 균형을 맞춘다면 더욱 완성도 높은 파노라마를 얻을 수 있을 것이다.

## 8.3 결론

본 프로젝트를 통해 컴퓨터 비전의 핵심 기술인 특징점 추출, 매칭, 기하학적 변환, 그리고 합성을 유기적으로 결합하여 실제적인 문제를 해결하는 경험을 하였다. 특히 이론적으로 알고 있던 알고리즘들을 구현하면서 발생한 다양한 문제들(누적 왜곡, 경계선 문제 등)을 원인 분석과 논리적인 해결책(Center Anchor, Inverse Mapping, Feather Blending 등)으로 극복해 나가는 과정에서 깊이 있는 이해를 얻을 수 있었다.

본 프로젝트는 단순한 이론의 구현을 넘어, 실제 데이터에서 발생하는 예측 불가능한 오차와 예외 상황들을 어떻게 분석하고 제어해야 하는지를 체득하는 값진 기회였다. 이는 향후 3D 복원, 자율주행과 같은 심화된 CV 분야의 공부로 확장해 나가는데 기반이 될 것이라 생각한다.