PanoStitch: Image Stitching Project Report - Phase 2

Team PanoStitch

December 22, 2025

Name	**ID**	**Section**
Hussien Mohamed	9230345	1
Mohamed	9230245	1
Amira Khaled Ahmed	9230513	1
Abdulrahman Medhat	9231026	2
Youssef Mohamed Noser	9231026	2

Contents

1 Introduction

PanoStitch is a modular panoramic image stitching pipeline designed to combine typically multiple overlapping images into a single, seamless panorama. This report details the algorithms used, experimental results, and the system analysis for Phase 2 of the project.

2 Work Division

The workload was divided as follows:

- **Hussein Mohamed**:
  1 Image Loading
  2 Feature Detection (Harris/SIFT)
  3 Feature Description (SIFT descriptors)
- **Abdelrahman Medhat**:
  4 Feature Matching (Brute-force + Lowe's ratio)
  5 DNN Integration of Deep Matcher (DISK+LightGlue)
- **Youssef Noser**:
  6 Homography Estimation (RANSAC + DLT)
  7 Homography Assembly (Connected components + chaining)
- **Amira Khalid**:
  8 Gain Compensation (Exposure balancing)
  9 Image Blending (Weighted composition)

3 Used Algorithms

The project implements a classic feature-based image stitching pipeline consisting of the following stages:

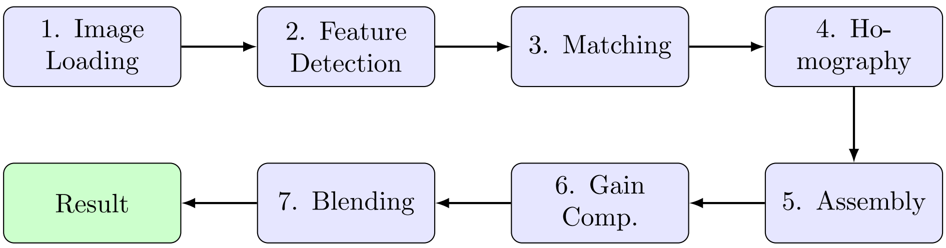

Figure 1: High-Level Processing Pipeline Flowchart

3.1 1. Image Loading

Images are loaded and optionally resized to a manageable dimension (default 800px) to ensure performance.

3.2 2. Feature Detection & Description

Two approaches were explored:

- **Harris Corner Detection**: A custom "From Scratch" implementation using Sobel operators and Gaussian smoothing to compute the Harris response $R = \det(M) - k(\text{trace}(M))^2$.
- **SIFT (Scale-Invariant Feature Transform)**: Used for robust keypoint detection and description. A custom implementation 'src/sift_implementation.py' was developed to handle scale-space extrema detection and orientation assignment.

3.3 3. Feature Matching

Matching is performed using a customized, vectorized **Brute-force Matcher** combined with **Lowe's Ratio Test** to ensure both accuracy and computational efficiency.

- **Distance Computation**: Feature descriptors are matched using the squared Euclidean distance. To improve efficiency, the implementation leverages the vectorized matrix identity:

  $$\|a-b\|^2 = \|a\|^2 + \|b\|^2 - 2ab^T$$

  This formulation enables the computation of all pairwise distances between descriptors from two images simultaneously, significantly outperforming naive Python loop-based implementations.

- **k-NN Search**: For each feature descriptor in Image A, the two nearest neighbors in Image B are identified using an optimized partition-based approach with linear time complexity $O(M)$ per query.

- **Lowe's Ratio Test**: A match is accepted only if the ratio between the distances of the closest and second-closest neighbors is below a predefined threshold (default 0.75). This criterion effectively filters out ambiguous matches and repetitive textures, improving robustness.

3.4 4. Homography Estimation

Homographies are estimated using:

- **DLT (Direct Linear Transform)**: Solves the system $Ah = 0$ using SVD to find the homography matrix.
- **RANSAC (Random Sample Consensus)**: A robust estimation algorithm that iteratively samples minimal point sets to find the homography with the most inliers, rejecting outliers and mismatched features.

3.5 5. Connected Components & Assembly

The system identifies connected components of images that share valid matches. For each component, a reference image (the most connected one) is chosen, and all other images are warped to its coordinate frame by chaining homographies.

3.6 6. Gain Compensation

To balance exposure differences:

- The system computes gain coefficients for each image.
- It minimizes an error function based on the intensity limits in the overlapping regions (σ_n and σ_g parameters).

3.7 7. Image Blending

Simple multi-band or weighted blending is applied to the warped images to create the final seamless composite.

4 Experiment Results

4.1 Test Cases

We evaluated the system on various datasets.

4.1.1 Test Case 1: Boat (Working)

- **Input**: 6 images ('imgs/boat/')
- **Result**: Successfully stitched into a single panorama.
- **Observation**: Good alignment, gain compensation handled exposure differences well.

4.1.2 Test Case 2: Dam (Working)

- **Input**: 2 images (‘imgs/dam/‘)
- **Result**: Successfully stitched.
- **Observation**: Robust matching despite fewer images.

4.2 Deep Learning Comparison (DNN Findings)

In addition to the classical pipeline, we integrated and evaluated a deep learning-based approach using **DISK** features with **LightGlue** matching. We benchmarked this against our classical SIFT and standard LoFTR implementations.

4.2.1 Quantitative Benchmark

We evaluated the methods on 7 diverse scenes (Mountain, Dam, Clock, Bicycle, Flower, Tree, River). The results are summarized below:

Table 1: Benchmark Results Summary

Scene	Method	Time (s)	Inliers	Inlier Ratio
Mountain	SIFT (Baseline)	**0.43**	1267	**99.22%**
	DISK+LightGlue	14.56	1174	98.66%
	LoFTR	31.64	**4495**	90.01%
Clock	SIFT (Baseline)	**1.03**	1389	85.42%
	DISK+LightGlue	15.82	517	**87.33%**
	LoFTR	32.76	**1541**	36.57%

4.2.2 Key Findings

- **Speed**: Classical SIFT is significantly faster (30-70×) than deep learning methods on CPU, making it the practical choice for real-time stitching without GPU acceleration.
- **Robustness**: DISK+LightGlue offered the highest average inlier ratio (86%) across all scenes, proving extremely robust even in challenging conditions where SIFT might struggle.
- **Density**: LoFTR provides very dense matches (thousands of points), which is beneficial for 3D reconstruction but less critical for homography estimation where a few high-quality matches suffice. LoFTR also struggled with repetitive patterns (e.g., the Clock scene), producing many false positives (low inlier ratio).

5 System Analysis

5.1 Classical vs. DNN Approach

Based on our experimental findings:

- **PanoStitch (Ours)**: Uses explicit geometric constraints and hand-crafted features (SIFT/Harris). It is resolution-independent and extremely fast on CPU. Ideally suited for high-resolution photography where geometric precision is paramount.
- **DNN Approaches**: Excel in specific domains (e.g., low-texture regions) but come with high computational cost. DISK+LightGlue represents the state-of-the-art for sparse matching quality, while classical SIFT remains the king of speed/efficiency trade-offs.

6 Conclusion

We have successfully implemented a modular image stitching pipeline. The system handles feature detection (both classical and deep learning), robust matching, and multi-image assembly with exposure correction, producing high-quality panoramas. Our Phase 2 comparative analysis confirms that while modern DNNs offer superior robustness, our optimized classical implementation generates high-quality results at a fraction of the computational cost.