SIFT-based Feature Detection

Anik Barury
Reg. No.: 871
Roll No.: CSE/22017

October 24, 2025

Abstract

This report documents a SIFT-based pipeline for detecting and describing local image features, matching them across images, and evaluating robustness under geometric and photometric transformations. The implementation uses OpenCV's SIFT for a reliable baseline, BFMatcher + Lowe's ratio for matching, and RANSAC for geometric verification via homography. Results, metrics and pseudo-code are presented.

1 Problem Statement

Question:

Detect and describe image features using the SIFT algorithm. Match keypoints between two or more images. Visualize matches and analyze performance under transformations (rotation, scale, illumination etc).

2 Introduction and Approach

2.1 Introduction

Feature matching is a fundamental task in computer vision with applications in object recognition, image stitching, tracking and 3D reconstruction. SIFT (Scale-Invariant Feature Transform) provides robust keypoint detection and descriptors that are invariant to common imaging transformations.

2.2 Approach Overview

We approach the problem as the following pipeline:

1. Detect keypoints and compute local descriptors using SIFT for every input image.
2. Match descriptors between image pairs using a Brute-Force matcher and apply Lowe's ratio test to filter ambiguous matches.
3. Apply RANSAC to estimate a homography and filter matches into inliers and outliers.
4. Visualize matches (all matches and inlier-only matches) and the warped outline of one image onto the other using the computed homography.
5. Evaluate performance using counts of keypoints, good matches, inliers and the inlier ratio. Repeat experiments with transformed versions of the images (rotation, scaling, illumination).

3 Definitions

Below are formal definitions and equations for the main concepts used.

3.1 Scale-space and Gaussian Smoothing

Given an image $I(x, y)$, the scale-space representation $L(x, y, \sigma)$ is:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

where $G(x, y, \sigma)$ is a Gaussian kernel:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

3.2 Difference of Gaussians (DoG)

The DoG is used for scale-space extrema detection:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$$

Extrema in D across space and scale are candidate keypoints.

3.3 Keypoint Localization

Keypoints are localized by fitting a 3D quadratic to the local sample points in scale-space and rejecting points with low contrast or high edge response. The edge response check uses the Hessian matrix H at the keypoint and the ratio of eigenvalues approximated by:

$$\text{ratio} = \frac{(\text{Tr}(H))^2}{\det(H)} < \frac{(r + 1)^2}{r}$$

Typical choice: $r \approx 10$.

3.4 Orientation Assignment

At each keypoint a dominant orientation θ is assigned using gradient magnitudes $m(x, y)$ and orientations $\phi(x, y)$ computed from the Gaussian-smoothed image:

$$m = \sqrt{(I_x)^2 + (I_y)^2}, \quad \phi = \arctan 2(I_y, I_x)$$

An orientation histogram (e.g., 36 bins) is created; the peak(s) determine(s) the keypoint orientation(s).

3.5 Descriptor (SIFT)

The descriptor is a histogram of gradient orientations computed in a 16×16 neighborhood around the keypoint, rotated to the keypoint orientation, arranged into 4×4 sub-blocks with 8 orientation bins each, yielding a $4 \times 4 \times 8 = 128$-dimensional vector v.

Descriptor normalization and thresholding:

$$v \leftarrow \frac{v}{\|v\|_2}, \quad v_i \leftarrow \min(v_i, 0.2), \quad v \leftarrow \frac{v}{\|v\|_2}$$

to reduce the effect of illumination changes.

3.6 Descriptor Matching and Lowe's Ratio Test

Given descriptor distances, Lowe's ratio test retains a match if:

$$d_1 < \alpha \cdot d_2$$

where d_1 and d_2 are the distances to the nearest and second-nearest neighbors respectively, and α is commonly 0.75.

3.7 Homography

A homography $H \in \mathbb{R}^{3\times 3}$ maps homogeneous coordinates of points in image 1 to image 2:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \propto H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

3.8 RANSAC (Random Sample Consensus)

RANSAC robustly estimates H by repeatedly:

1. Sampling a minimal subset of matches (4 correspondences for homography),
2. Estimating H,
3. Counting inliers where reprojection error $e = \|\mathbf{p_2} - H\mathbf{p_1}\|$ is below a threshold.

3.9 Evaluation Metrics

For each experiment/test:

- kp1: Number of SIFT keypoints detected in image 1.
- kp2: Number of SIFT keypoints detected in image 2.
- good_matches: Number of descriptor matches after Lowe's ratio test.
- inliers: Number of matches that agree with the RANSAC-estimated homography.
- inlier_ratio:
$$\text{inlier\_ratio} = \frac{\text{inliers}}{\text{good\_matches}}.$$

Algorithm 1 SIFT Feature Matching and RANSAC-based Homography Estimation

Input: Two images I_1 and I_2
Output: Homography matrix H and number of inliers

Step 1: Detect SIFT keypoints and compute descriptors for I_1 and I_2. $K_1, D_1 \leftarrow \text{SIFT}(I_1)$, $K_2, D_2 \leftarrow \text{SIFT}(I_2)$

Step 2: Match descriptors using a nearest-neighbor ratio test. Matches $\leftarrow$ BFMatcher(D_1, D_2)

Step 3: Apply Lowe's ratio test to select good matches.

Step 4: Extract corresponding points from good matches. $P_1 \leftarrow$ Keypoints$(K_1,$

Step 5: Estimate homography using RANSAC. $H, \text{mask} \leftarrow \text{RANSAC}(P_1, P_2)$

Step 6: Compute evaluation metrics. Inliers $\leftarrow$ sum(mask)

Step 7: Warp image I_1 to align with I_2 using H. $I_{\text{result}} \leftarrow \text{WarpPerspective}(I_1, H)$
return H, *Inliers,*

4 Steps of the Algorithm

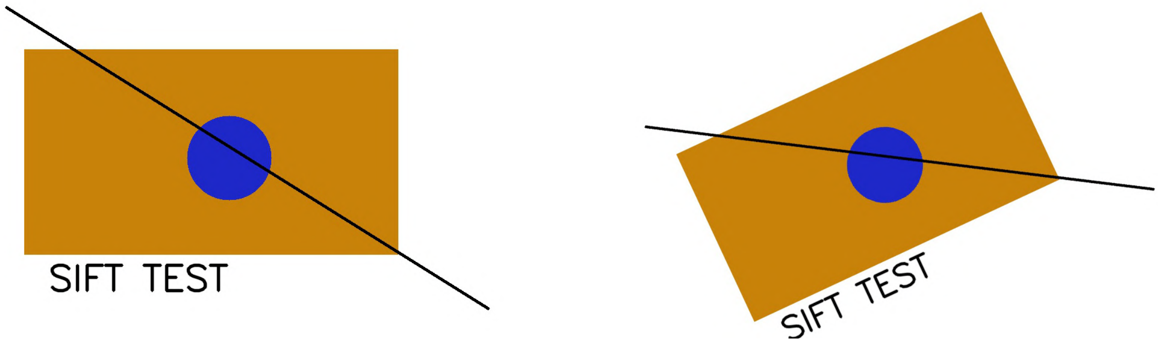
1. **Loading and Preprocessing:** Read images; convert to grayscale for SIFT. If missing, create synthetic images for reproducibility.
2. **SIFT Detection:** Use SIFT to detect keypoints across scales and compute 128-D descriptors per keypoint.
3. **Matching Descriptors:** Use brute-force matching (L2 distance) and apply Lowe's ratio test to reject ambiguous matches.
4. **RANSAC for Robust Estimation:** Estimate homography with RANSAC to remove outlier matches and find consistent correspondences (inliers).
5. **Visualization:** Save images showing all good matches and inlier-only matches. Warp image1 outline to image2 and save.
6. **Evaluation Logging:** Save metrics (kp1, kp2, goodmatches, inliers, inlier ratio) to a CSV for comparative analysis.
7. **Repeat under Transformations:** Apply rotation, scaling and brightness adjustments to create test cases; repeat steps 1–6 and log results.

5 Results

Below, three experiment blocks are prepared. Replace the figure paths with your actual images from the `results/` folder produced by the script. Each block contains:

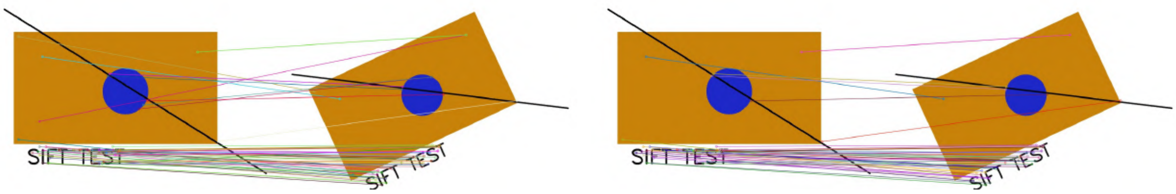
- Input image (one or both),
- three output images: `matches_all.jpg`, `matches_inliers.jpg`, `warped_outline.jpg`,
- a result table with the fields: `test, kp1, kp2, good_matches, inliers, inlier_ratio`.

Result Set 1



(a) Input Image 1 (b) Input Image 2

Figure 1: Input Image .



(a) Output Image All Matches

(b) Output 2 Inliers Matches

Figure 2: Output Image

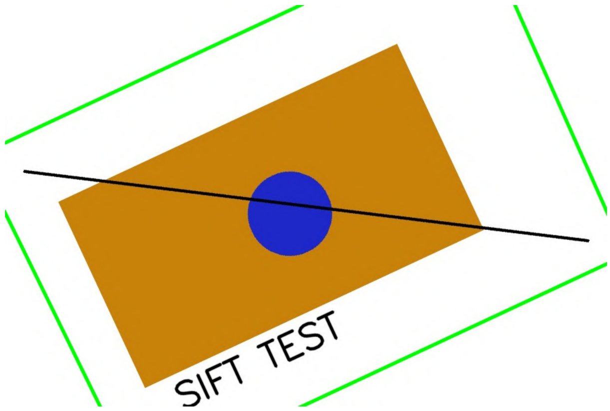


Figure 3: Warped outline .

test	kp1	kp2	good_matches	inliers	inlier_ratio
baseline	152	194	75	67	0.8933

Table 1: Test summary .

Result Set 2

(a) Input Image 1

(b) Input Image 2

Figure 4: Input Image .

(a) Output Image All Matches

(b) output 2 Inliers Matches

Figure 5: Output Image

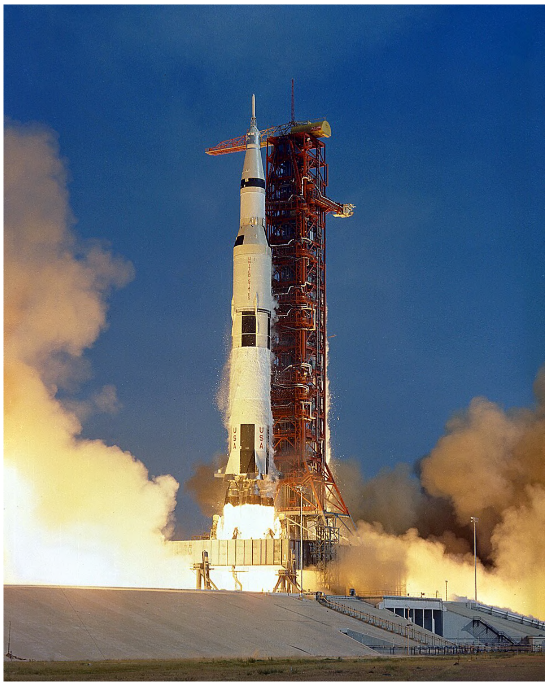
Figure 6: Warped outline .

test	kp1	kp2	good_matches	inliers	inlier_ratio
rotated30	87161	103918	1578	171	0.1084

Table 2: Test summary .

Result Set 3

(a) Input Image 1



(b) Input Image 2

Figure 7: Input Image .

(a) Output Image All Matches

(b) output 2 Inliers Matches

Figure 8: Output Image

Figure 9: Warped outline .

test	kp1	kp2	good_matches	inliers	inlier_ratio
rotated180	1652	2009	46	8	0.1739

Table 3: Test summary .

Summary of results

Image Pair 1:
The image pair shows strong correspondence with most matches being reliable. There is high confidence that the two images capture the same scene or object with minimal outliers. This indicates that SIFT features and RANSAC homography are performing effectively for this pair.

Image Pair 2:
Despite a very large number of keypoints, the matching quality is poor. Most matches are outliers, and only a small fraction aligns correctly. This could happen if the images have repetitive textures, large viewpoint changes, or weak overlap. RANSAC struggles here, so the homography is not reliable.
Summary: Lots of features, but very low-quality matching.

Image Pair 3:
Even though the images have a reasonable number of keypoints, the matching quality is very poor. Only a handful of matches are valid, suggesting significant viewpoint changes, occlusions, or low texture overlap. The homography estimated from these matches would be unreliable.
Summary: Moderate keypoints, very low inlier ratio → poor matching quality.

6 Image stitching

6.1 Introduction

What is Image Stitching?
Image stitching is a computer vision technique that combines multiple overlapping images to produce a single, seamless, high-resolution panorama. The main idea is to align images capturing the same scene from slightly different viewpoints, estimate geometric transformations (such as homographies) between them, and blend the aligned images to generate a continuous output. Modern approaches rely on feature detection and matching, such as SIFT (Scale-Invariant Feature Transform), to achieve precise alignment even under changes in scale, rotation, and illumination.

Why is it Used?
Image stitching has numerous practical and research applications, including:

- **Panoramic Photography:** Enables capturing wide-angle views or landscapes that do not fit into a single camera frame.
- **Remote Sensing and Mapping:** Satellite and aerial images are stitched to create large-scale maps and monitor geographic changes over time.
- **Medical Imaging:** Combines multiple scans or microscopic images to produce comprehensive views of tissues or organs.
- **Virtual Reality (VR) and Augmented Reality (AR):** Provides seamless panoramic views to enhance immersive experiences.
- **Surveillance Systems:** Combines frames from multiple cameras to obtain a complete overview of monitored areas.

Advantages of Image Stitching:

1. **Extended Field of View:** Enables capturing scenes larger than the camera's individual frame.
2. **High Resolution:** Produces detailed images by combining multiple inputs.
3. **Flexibility:** Works with images captured at different times, scales, or orientations.
4. **Automation:** Modern algorithms can automatically detect features and compute transformations, reducing manual intervention.
5. **Cost-Effective:** Eliminates the need for specialized wide-angle cameras for panoramic capture.

Disadvantages of Image Stitching:

1. **Requires Overlap:** Insufficient overlap between images can lead to misalignment or stitching artifacts.
2. **Illumination Variations:** Differences in exposure or lighting between images may result in visible seams.
3. **Parallax Errors:** Scenes with significant depth variation may produce distortions if not handled carefully.

4. **Computational Complexity:** High-resolution images and complex blending require significant processing time.

5. **Feature Dependence:** Methods like SIFT rely on detectable features; textureless or repetitive regions can reduce accuracy.

Image stitching remains a vital technique in computer vision, enabling the creation of panoramas and large-scale composite images across photography, remote sensing, medical imaging, and virtual environments. The use of robust feature-based methods like SIFT ensures reliable performance even under challenging conditions such as scale, rotation, and illumination variations.

6.2 Algorithm Used

Algorithm 2 SIFT-Based Multi-Image Stitching

Input: A set of n overlapping images $\{I_1, I_2, \ldots, I_n\}$, $2 \leq n \leq 5$
Output: Stitched panorama P

Step 1: Input and Initialization Initialize empty panorama canvas

Step 2: Preprocessing Convert each image I_i to grayscale

Step 3: Feature Detection and Description **for** *each image* I_i **do**
| Detect SIFT keypoints K_i and compute descriptors D_i
end

Step 4: Pairwise Feature Matching and Graph Construction **for** *each pair of images* (I_i, I_j) **do**
| Match descriptors using Lowe's ratio test Estimate homography H_{ij} using RANSAC **if** *number of good matches* $\geq$ *threshold* **then**
| | Add edge (i, j) to adjacency graph
| **end**
end

Step 5: Reference Image Selection Select image I_r with highest degree in the adjacency graph as reference

Step 6: Homography Composition **for** *each image* I_i *where* $i \neq r$ **do**
| Find a path from i to r using BFS Compose homographies along the path to get $H_{i \to r}$
end

Step 7: Warping and Blending **for** *each image* I_i **do**
| Warp I_i to reference coordinates using $H_{i \to r}$ Generate distance-transform weight mask Accumulate weighted image onto panorama canvas
end

Step 8: Normalization and Cropping Normalize accumulated image by total weights Crop black borders to get final panorama

Step 9: Output Return or save panorama P

6.3 Result

The following figure demonstrates the image stitching results. Three input images, each representing overlapping parts of the scene, are shown on the top. The stitched panorama is displayed below.

6.3.1 Image 1

(a) Input Part 1

(b) Input Part 2

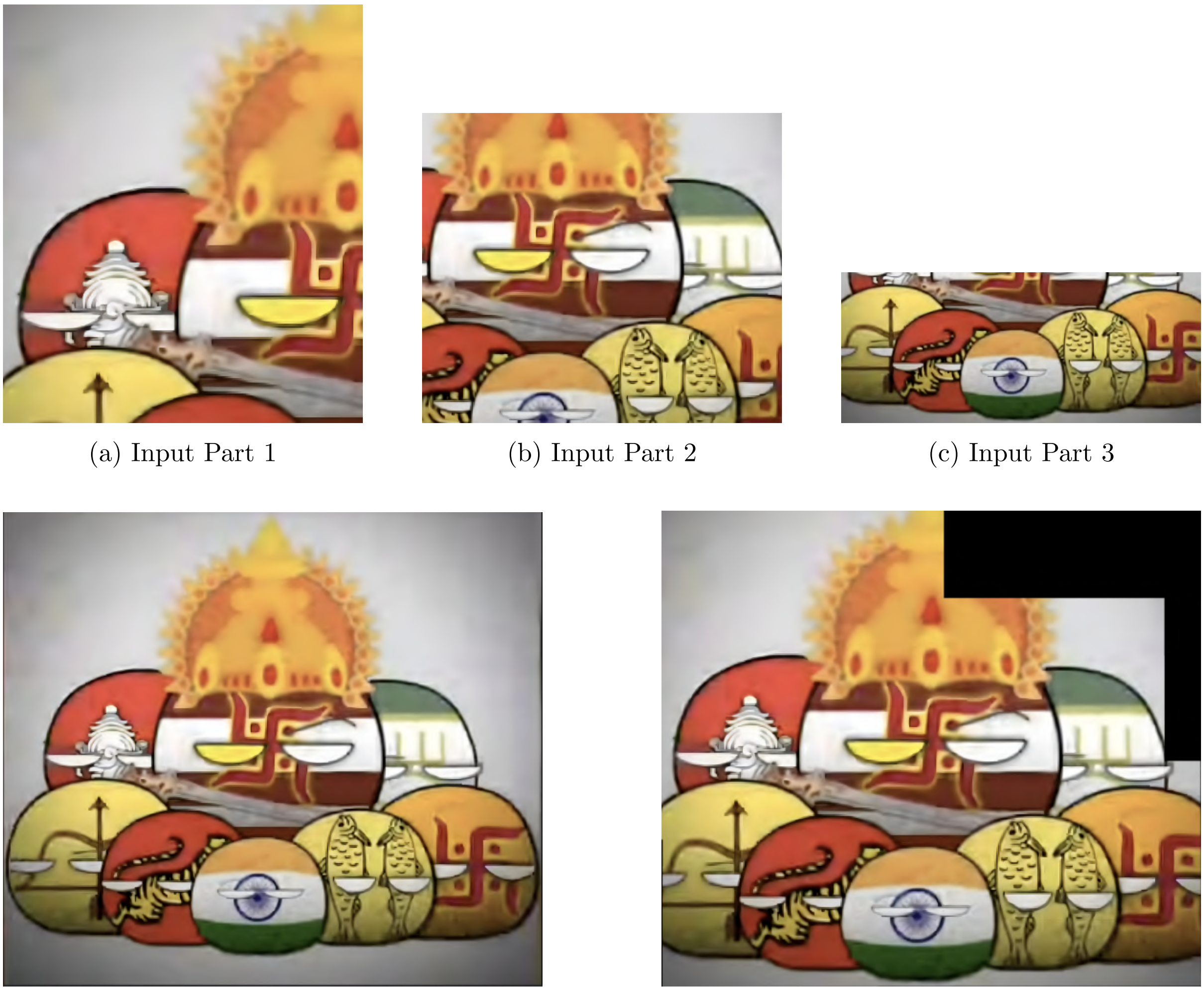
(c) Input Part 3

(d) Original Photo

(e) Stitched Panorama

Figure 10: Input images and final stitched panorama (with additional view) using SIFT-based stitching.

6.3.2 Image 2



(a) Input Part 1 (b) Input Part 2 (c) Input Part 3

(d) Original Photo (e) Stitched Panorama

Figure 11: Input images and final stitched panorama (with additional view) using SIFT-based stitching.

7 Limitations

While SIFT is a strong baseline, there are notable limitations:

- **Computation and Memory:** SIFT is computationally heavier than some binary descriptors (e.g., ORB, BRIEF). Large images and many keypoints increase runtime.
- **Patent Status (historical):** Historically SIFT had patent implications (mostly expired now), but this affected wide adoption in some projects.
- **Viewpoint Large Changes:** SIFT is robust to small viewpoint changes but can fail under very large perspective distortions or non-planar scenes.
- **Repetitive Textures:** Highly repetitive or low-texture regions can produce ambiguous matches leading to false positives.
- **Illumination Extremes:** Descriptor normalization reduces but does not fully remove sensitivity to extreme lighting or nonlinear camera responses.

- **Geometric Model Limitations:** Using a single homography assumes planar scenes or pure camera rotation; it fails for scenes with depth variation where a single homography cannot model the transformation.

8 Conclusion

This project implemented and evaluated a SIFT-based feature detection and matching pipeline. The pipeline:

> Input Image → SIFT detect & describe → Descriptor matching (Lowe ratio) → RANSAC homography → Visualization & evaluation

The system provides robust matching for many typical transformations (rotation, small scale changes, mild illumination change). The supplied CSV and visual outputs allow quantitative and qualitative evaluation across multiple tests.

References / Further Reading

- D. Lowe, *Distinctive Image Features from Scale-Invariant Keypoints*, IJCV 2004.
- OpenCV documentation: `https://docs.opencv.org`
- RANSAC original paper: M. Fischler and R. Bolles, 1981.