Computer Vision HW 1

Elia Mirafiori VR537643

December 22, 2025

Abstract

This project aims to perform 3D reconstruction of an object from two viewpoints using a Structure-from-Motion pipeline. I first describe the experimental setup, including camera calibration and image acquisition. The designed pipeline for feature detection, matching, and epipolar geometry estimation is then presented. Various feature extraction methods and geometric estimation approaches are compared, and the results are evaluated both quantitatively and qualitatively to assess reconstruction accuracy and completeness.

1 Setup

The experimental setup used in this project was intentionally kept simple and easily reproducible. To this end, a consumer-grade smartphone camera was selected, allowing other users to replicate and extend the proposed pipeline without requiring specialized hardware. Specifically, an *Oppo Reno 6* smartphone was used, featuring a 64 MP main camera with an aperture of $f/1.7$, a field of view of $81°$, a 6P lens system, and autofocus supported by a closed-loop focus motor.

Camera calibration was performed using a well-established and widely adopted approach based on a planar checkerboard pattern. The checkerboard consisted of 8×8 squares, each with a side length of 24 mm, providing a sufficient number of corner points for accurate intrinsic parameter estimation.

As the target object for 3D reconstruction, a Moai statue (used as a tissue holder) was selected. This object was chosen due to its rich surface texture and geometric details, which are beneficial for robust feature detection and matching. Compared to objects with fewer visual features, this choice resulted in a more stable and reliable reconstruction.

Image acquisition was conducted in a controlled indoor environment, specifically a bedroom. Different lighting conditions were tested during the photo capture process, and an appropriate configuration was selected to minimize shadows and reflections, ultimately contributing to improved feature detection and overall reconstruction quality.

2 Methodology

2.1 Feature Detection and Matching

2.1.1 Feature Detection

In this assignment, several feature detectors and descriptors were evaluated to analyze their performance and robustness under different conditions. The tested methods include ORB, AKAZE, BRISK, and SIFT, which differ in terms of invariance properties, computational cost, and descriptor type.

- ORB (Oriented FAST and Rotated BRIEF) is a fast and efficient feature detector that combines the FAST keypoint detector with the BRIEF descriptor, enhanced with orientation and rotation invariance. It produces binary descriptors, making it particularly suitable for real-time applications due to its low computational cost.
- AKAZE detects features in a non-linear scale space, which improves robustness to scale changes while maintaining good performance. Similar to ORB, AKAZE produces binary descriptors, allowing fast matching using Hamming distance.

- BRISK (Binary Robust Invariant Scalable Keypoints) is a scale- and rotation-invariant detector and descriptor that relies on a circular sampling pattern. It also generates binary descriptors, offering a good trade-off between robustness and computational efficiency.
- SIFT (Scale-Invariant Feature Transform) is a classic and highly robust method that detects keypoints in a Gaussian scale space and computes floating-point descriptors. It is invariant to scale and rotation and more robust to illumination changes, but it is computationally more expensive compared to binary-based methods.

2.1.2 Feature Matching

After extracting keypoints and descriptors, feature matching was performed to establish correspondences between images. Different matching strategies were adopted depending on the descriptor type.

2.1.3 Brute-Force Matcher

The Brute-Force (BF) matcher computes distances between all pairs of descriptors and selects the best match according to a chosen norm:

- For ORB, AKAZE, and BRISK, the Hamming distance was used, as these methods produce binary descriptors.
- For SIFT, the L2 norm was used, which is appropriate for floating-point descriptors.

The BF matcher guarantees exact nearest-neighbor matching but can become computationally expensive for large numbers of features.

2.1.4 FLANN-Based Matcher

To improve efficiency, especially for large descriptor sets, the FLANN (Fast Library for Approximate Nearest Neighbors) matcher was also tested:

- For ORB, FLANN was configured with LSH (Locality-Sensitive Hashing), which is specifically designed for approximate nearest-neighbor search on binary descriptors.
- For SIFT, FLANN was configured with a KD-Tree index, which is well suited for high-dimensional floating-point descriptors.

FLANN provides approximate matches with significantly reduced computational cost compared to brute-force matching, often with negligible loss in accuracy.

2.2 Epipolar Geometry

To estimate the geometric relationship between two views, two different pipelines were implemented: an indirect approach, based on the computation of the Fundamental matrix, and a direct approach, where the Essential matrix is estimated directly from point correspondences.

2.3 Fundamental Matrix Estimation

To estimate the Fundamental Matrix, I first converted the matched feature points into keypoint coordinates, in order to represent them in the standard 2D image coordinate system required by the estimation algorithm. Then, I employed the OpenCV function `findFundamentalMatrix` with the RANSAC method to robustly handle outliers. The RANSAC reprojection threshold was set to 1.0, while the confidence parameter was set to 0.999, which allowed for a precise and reliable estimation of the Fundamental Matrix.

The `findFundamentalMatrix` function also returns a mask indicating which points are inliers and which are outliers. I used this mask to filter out outlier correspondences, retaining only the inlier points for subsequent processing steps, thereby improving the robustness of further geometric computations.

2.3.1 Indirect Pipeline: Fundamental Matrix to Essential Matrix

In the indirect approach, the epipolar geometry is first modeled in pixel coordinates by estimating the **Fundamental matrix F**. This matrix encapsulates the epipolar constraint between corresponding points in two uncalibrated images.

Given a set of matched feature points, the Fundamental matrix is computed using a robust estimation method, **RANSAC**, to handle outliers. Once **F** is obtained, camera calibration information is introduced to recover the **Essential matrix E**, which relates normalized image coordinates.

The conversion is performed as:

$$\mathbf{E}_{raw} = \mathbf{K}^\top \mathbf{F} \mathbf{K} \tag{1}$$

where **K** is the intrinsic calibration matrix of the camera.

However, the matrix $\mathbf{E}_{raw}$ obtained from this algebraic product does not automatically satisfy the internal constraints of a valid Essential matrix (specifically, having two equal non-zero singular values and one zero singular value). To ensure geometric consistency, I explicitly **enforced these constraints** using Singular Value Decomposition (SVD). Given the decomposition $\mathbf{E}_{raw} = \mathbf{U}\text{diag}(\sigma_1, \sigma_2, \sigma_3)\mathbf{V}^\top$, I projected the matrix onto the Essential space by setting the singular values to $(1, 1, 0)$:

$$\mathbf{E} = \mathbf{U} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{V}^\top \tag{2}$$

The resulting Essential matrix **E** properly encodes the relative rotation and translation between the two camera views, up to a scale factor.

This pipeline clearly separates feature-based geometry estimation from camera calibration, making it flexible in scenarios where calibration is applied as a post-processing step.

2.3.2 Direct Pipeline: Essential Matrix Estimation

In the direct approach, the **Essential matrix** is estimated directly from the matched feature points using known camera intrinsics. Instead of first computing the Fundamental matrix, point correspondences are normalized using the intrinsic matrix **K**, and the epipolar geometry is modeled directly in normalized coordinates.

This is achieved using algorithms such as **Nistér's 5-point algorithm**, implemented in OpenCV via `cv.findEssentialMat`. This method requires a minimum of five point correspondences and inherently uses RANSAC to reject outliers based on the calibrated geometry.

To further enhance the quality of the reconstruction, I implemented a **two-stage filtering strategy**. Rather than estimating the Essential matrix from the raw set of feature matches, I used the set of **inliers previously validated by the Fundamental Matrix estimation** (as described in the previous section).

By pre-filtering the correspondences using the epipolar constraint of **F**, I removed gross outliers before enforcing the stricter metric constraints of **E**. This "polished" input set allowed the `cv.findEssentialMat` RANSAC loop to converge on a more optimal solution, effectively reducing the noise in the final geometry. The function returned a refined mask, which was used to perform a final filtering pass, ensuring that only the most geometrically consistent points were used for triangulation and 3D reconstruction.

This approach proved to be more numerically stable than the indirect method and resulted in a cleaner final point cloud.

2.3.3 Comparison of the Two Pipelines

- The indirect pipeline is conceptually simpler and applicable even when camera intrinsics are not initially known, but it may accumulate numerical errors due to the two-step estimation.
- The direct pipeline leverages known calibration parameters and directly estimates the Essential matrix, often resulting in improved accuracy and robustness.

Both approaches ultimately yield the Essential matrix, which can then be decomposed to recover the relative camera pose (rotation and translation), as you will see later.

2.4 Relative Pose Results

To estimate the Rotation Matrix and the Translation vector, I employed the built-in OpenCV function `recoverPose`, which requires the inlier points obtained from the previous steps, the Essential Matrix, and the Camera Matrix as inputs. The function returns four outputs: the number of valid points, the Rotation Matrix, the Translation vector, and a mask indicating inliers. Similar to the previous masks, this mask was used to further refine the set of points, ensuring that only geometrically consistent correspondences were retained for subsequent processing. This iterative refinement improves the accuracy and robustness of the relative pose estimation.

2.5 Triangulation

After estimating the relative pose between the two camera views, 3D points were reconstructed from 2D point correspondences through triangulation. This process recovers the spatial position of scene points by exploiting their projections onto multiple images.

Given the camera intrinsics and the relative rotation $\mathbf{R}$ and translation $\mathbf{t}$ obtained from the Essential matrix and the Camera matrix $\mathbf{K}$, the projection matrices for the two views were defined as:

$$\mathbf{P}_1 = \mathbf{K}\,[\mathbf{I} \mid \mathbf{0}]\,, \quad \mathbf{P}_2 = \mathbf{K}\,[\mathbf{R} \mid \mathbf{t}]\,. \tag{3}$$

Using these projection matrices and the corresponding 2D feature points in both images, triangulation was performed with the OpenCV function `cv.triangulatePoints`, which computes homogeneous 3D points by solving a linear least-squares problem. The resulting points were then converted from homogeneous coordinates to Euclidean 3D coordinates, yielding a sparse 3D point cloud of the scene.

2.6 3D Point Filtering

The initial triangulated point cloud may contain geometrically inconsistent or noisy points due to mismatches and numerical inaccuracies. To improve the quality of the reconstruction, several filtering steps were applied.

First, a *positive depth constraint* was enforced, retaining only points with positive depth in both camera coordinate systems. This ensures that reconstructed points lie in front of both cameras and satisfy the physical constraints of the imaging process.

Subsequently, extreme depth outliers were removed by applying a percentile-based filtering strategy. Points with depth values above the 97th percentile were discarded, effectively removing points with unrealistically large depths while preserving the main structure of the scene.

3 Experimental Setup

3.1 Dataset

The experimental evaluation was conducted on a pair of images depicting a Moai statue, captured from two different viewpoints. These images constitute the input to the entire reconstruction pipeline and are used for feature detection, matching, epipolar geometry estimation, and triangulation.

Both images have a resolution of *[2608]* $\times$ *[4624]* pixels. Figure 1 shows the two input images used in the experiments.

3.2 Camera Calibration

Camera calibration proved to be one of the most challenging stages of the entire project, as the accuracy of all subsequent steps strongly depends on the quality of the estimated intrinsic parameters. To this end, several calibration image sets were acquired using a checkerboard pattern.

Specifically, three different acquisition strategies were tested: a set of images captured with the checkerboard in predominantly horizontal orientations, a second set with vertical orientations, and a

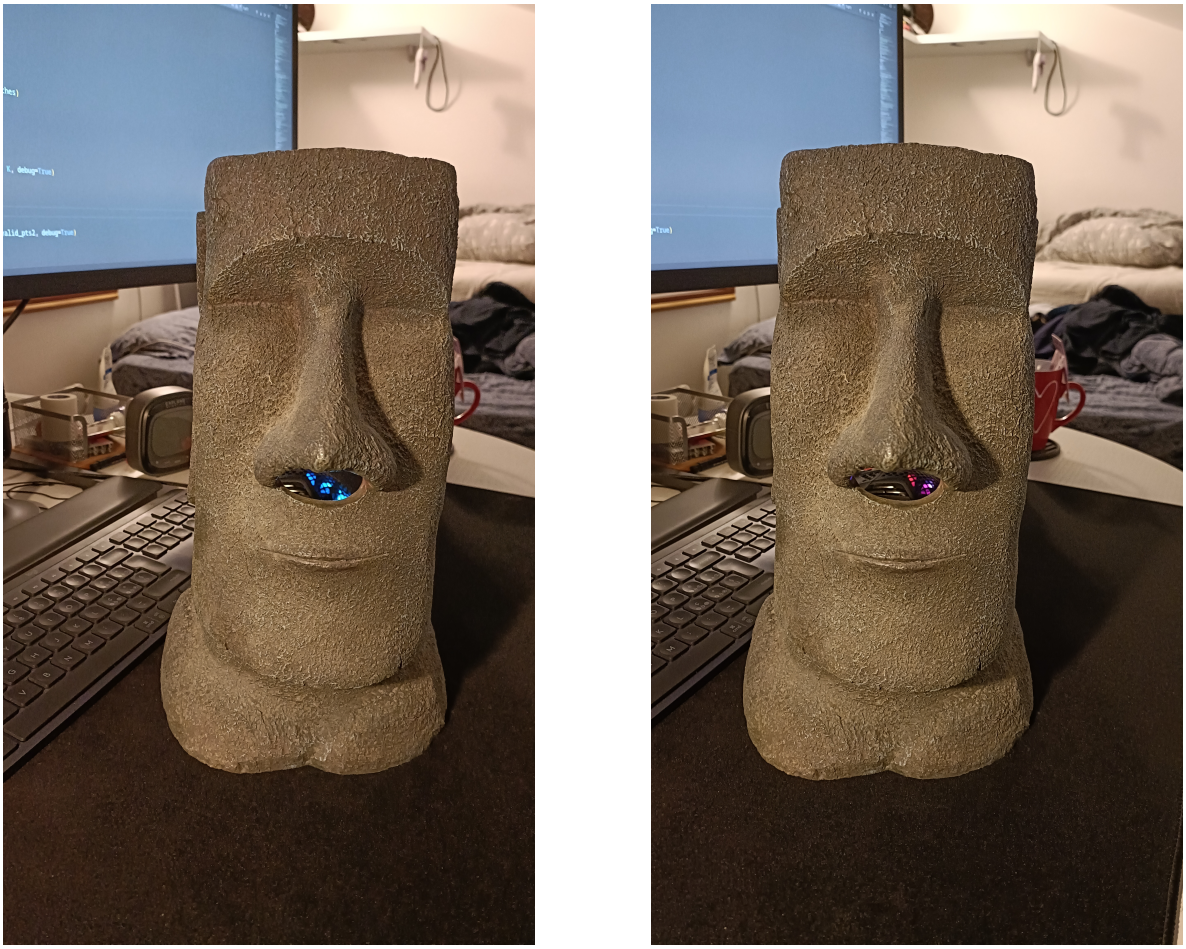

Figure 1: Input image pair of the Moai statue used in the experiments.

third set acquired under low-light conditions using a tablet to display a virtual checkerboard. The latter approach yielded poor results, as the corner detection algorithm (`findChessboardCorners`) consistently failed to detect valid corner points during image iteration. Consequently, this strategy was discarded.

While both the horizontal and vertical checkerboard image sets produced mathematically valid calibration results with low reprojection errors, experimental validation revealed significant differences in practical performance.

The vertical image set was ultimately selected as it provided robust performance across all detector types. In contrast, the horizontal image set proved problematic for sparse detectors (e.g., ORB, BRISK), failing to yield a stable reconstruction despite a comparable calibration error. Interestingly, dense detectors (SIFT, AKAZE) were able to overcome the limitations of the horizontal calibration and produce a reconstruction, albeit with a significantly reduced number of 3D points.

This discrepancy suggests that while the horizontal calibration parameters were numerically sound, they likely introduced geometric biases that only the high-density feature matching of SIFT and AKAZE could accommodate. Therefore, the vertical set was chosen to ensure consistent and comparable results across all tested algorithms.

The resulting camera intrinsic matrix $\mathbf{K}$ is:

$$\mathbf{K} = \begin{bmatrix} 3369.83 & 0 & 1330.55 \\ 0 & 3377.15 & 2358.32 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4}$$

The corresponding lens distortion coefficients were estimated as:

$$\mathbf{d} = \begin{bmatrix} -0.0164 & 0.3087 & 0.0014 & 0.0013 & -0.6379 \end{bmatrix}, \tag{5}$$

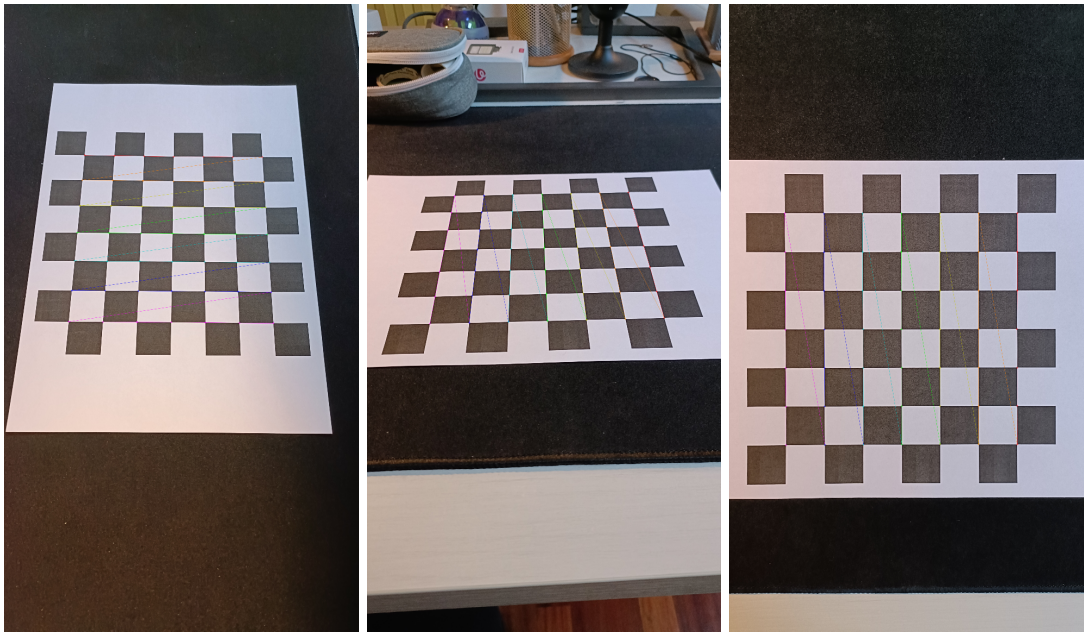

Figure 2: Example chessboard calibration images with detected corner points.

where the parameters follow the standard OpenCV distortion model.

Calibration accuracy was evaluated using the reprojection error measured in pixels. The obtained values are:

- Mean reprojection error: 0.560 px
- Standard deviation: 0.443 px
- Maximum observed error: 0.714 px

These results indicate a satisfactory calibration quality, with sub-pixel mean reprojection error, suitable for reliable epipolar geometry estimation and 3D reconstruction.

Figure 2 shows example calibration images with the detected chessboard corners overlaid.

3.3 Parameters

To ensure a fair comparison across different methods, several parameters were kept fixed throughout the experiments.

For robust model estimation, RANSAC was used with a reprojection threshold of:

$$\texttt{RANSAC\_THRESH} = 1.0, \tag{6}$$

and a confidence level of:

$$\texttt{CONFIDENCE} = 0.999. \tag{7}$$

For feature extraction, the ORB detector was configured with a maximum number of features set to:

$$\texttt{nfeatures} = 10000. \tag{8}$$

AKAZE was used with a detector threshold of:

$$\texttt{threshold} = 5 \times 10^{-4}. \tag{9}$$

To limit computational complexity, the maximum number of matches retained after matching was set to 5000. When using FLANN-based matching, Lowe's ratio test was applied with a ratio threshold of 0.70.

3.4 Estimated Geometric Primitives

Corresponding to the epipolar geometry visualized in Figure 3, I report the estimated geometric matrices obtained using the **BRISK** detector.

1. **Fundamental Matrix (F)**: Encodes the epipolar geometry without calibration.
2. **Essential Matrix (E)**: Encodes the geometry with calibration, satisfying $E \approx K^T F K$.
3. **Motion (R, t)**: The rotation and translation between the two views.
4. **Projection Matrices (P_1, P_2)**: The full camera projection matrices where $P_1 = K[I|0]$ and $P_2 = K[R|t]$.

$$F = \begin{bmatrix} 5.07 \times 10^{-8} & -5.77 \times 10^{-6} & 0.0162 \\ 4.95 \times 10^{-6} & 8.50 \times 10^{-8} & -0.0831 \\ -0.0147 & 0.0831 & 1.0000 \end{bmatrix}$$
$$E = \begin{bmatrix} 0.0021 & -0.1741 & 0.0393 \\ 0.1495 & 0.0035 & -0.6900 \\ -0.0426 & 0.6840 & 0.0044 \end{bmatrix} \tag{10}$$

From the Essential Matrix, the relative camera pose was recovered. The rotation matrix R is close to the identity matrix, and the translation vector t indicates a dominant horizontal movement ($t_x \approx -0.97$), confirming the correct lateral "crab-walk" motion during acquisition.

$$R = \begin{bmatrix} 0.9994 & 0.0039 & -0.0356 \\ -0.0038 & 1.0000 & 0.0043 \\ 0.0356 & -0.0042 & 0.9994 \end{bmatrix}, \quad t = \begin{bmatrix} -0.9676 \\ -0.0566 \\ -0.2460 \end{bmatrix} \tag{11}$$

Finally, the full Projection Matrices P_1 (canonical) and P_2 (transformed) used for triangulation are:

$$P_1 = \begin{bmatrix} 3369.83 & 0 & 1330.55 & 0 \\ 0 & 3377.15 & 2358.32 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$
$$P_2 = \begin{bmatrix} 3.415 \times 10^3 & 7.686 & 1.210 \times 10^3 & -3.588 \times 10^3 \\ 71.11 & 3.367 \times 10^3 & 2.371 \times 10^3 & -771.29 \\ 0.0356 & -0.0042 & 0.9994 & -0.2460 \end{bmatrix} \tag{12}$$

3.5 Epipolar Geometry Visualization

To verify the correctness of the estimated geometry, I visualized the epipolar lines for a subset of matched keypoints. Figure 3 displays the epipolar constraints computed using the **BRISK** detector with the Direct Essential Matrix pipeline.

In a correct estimation, every feature point in the right image (second view) must lie on the corresponding epipolar line in the left image (first view), and vice versa. As seen in the figure, the epipolar lines accurately pass through the centers of the corresponding feature points (marked with circles). The convergence of these lines towards a point outside the image frame indicates the position of the epipoles, consistent with the horizontal translation of the camera during the acquisition.

4 Results

4.1 Reconstruction Comparison

Figure 4 presents a qualitative comparison between a *sparse* and a *dense* 3D reconstruction of the scene. The sparse reconstruction was obtained using the ORB detectors, while the denser reconstructions were produced using AKAZE and SIFT.

As expected, the ORB-based reconstruction results in a sparser point cloud due to the lower number of stable inliers and triangulated points. In contrast, AKAZE and SIFT produce significantly denser point clouds, generally capturing finer geometric details of the Moai statue. This improvement is

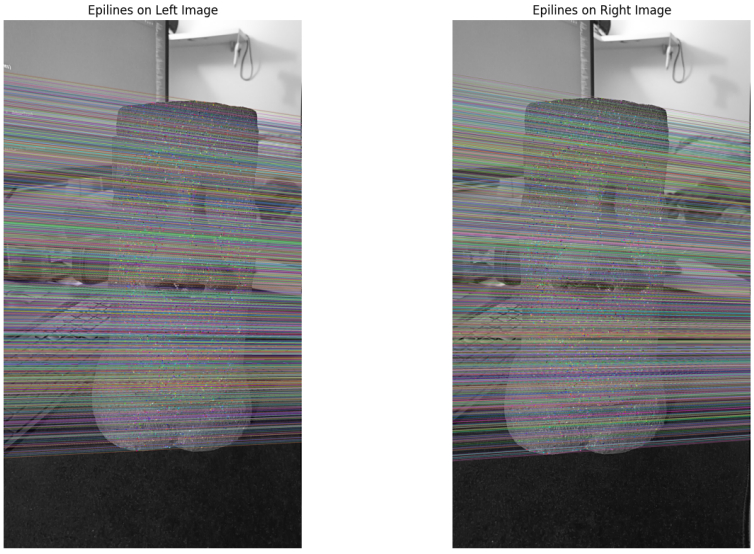

Figure 3: Epipolar lines visualization using BRISK matches. The left image shows the epipolar lines corresponding to points in the right image, and vice versa. The precise alignment of points on the lines confirms the accuracy of the estimated Fundamental Matrix.

Table 1: Quantitative comparison of feature detectors and matchers.

Detector	Matcher	#Keypoints	#Matches	#Inliers	#3D Points	Reproj. Error	Time
ORB	BFMatcher	10000	4827	2475	2400	0.25 px	0.96 s
ORB	FlannBasedMatcher	10000	2707	1662	1612	0.26 px	0.70 s
AKAZE	BFMatcher	55428	5000	3998	3875	0.22 px	16.57 s
BRISK	BFMatcher	84693	5000	4090	3967	0.23 px	32.27 s
SIFT	BFMatcher	89573	5000	3929	3811	0.30 px	95.69 s
SIFT	FlannBasedMatcher	89573	5000	3547	3387	0.27 px	8.21 s

primarily attributed to the higher number of detected keypoints and more robust feature matching, at the cost of increased computational time.

However, a notable discrepancy was observed with the **SIFT + FLANN** pipeline. While it performed exceptionally well from a quantitative perspective (achieving a high number of inliers and low reprojection error), the qualitative reconstruction did not meet expectations. As illustrated in Figure 5, the resulting point cloud is geometrically incomplete, resolving only a narrow, vertical "column" of the statue rather than its full volume. This suggests that while the features were mathematically consistent with the epipolar geometry, they were spatially concentrated along specific high-contrast vertical textures, failing to cover the smoother peripheral areas of the face.

In contrast, the **BRISK** detector (Figure 6) demonstrated a notable balance between density and accuracy. Although technically classified as a sparse detector like ORB, BRISK's scale-invariant corner detection appeared to be more robust to the specific texture of the statue. This suggests that for this specific object, BRISK provided a higher ratio of "high-quality" structural points, avoiding the spatial clustering issues observed in SIFT and the sparsity of ORB.

Overall, this comparison highlights that quantitative metrics alone are insufficient to judge reconstruction quality. While dense reconstructions offer improved visual completeness, robust sparse variants like BRISK can remain highly effective for applications where geometric precision and global shape fidelity are prioritized over surface density.

Figure 4: Side-by-side comparison of sparse (ORB) and dense (SIFT) 3D reconstructions.

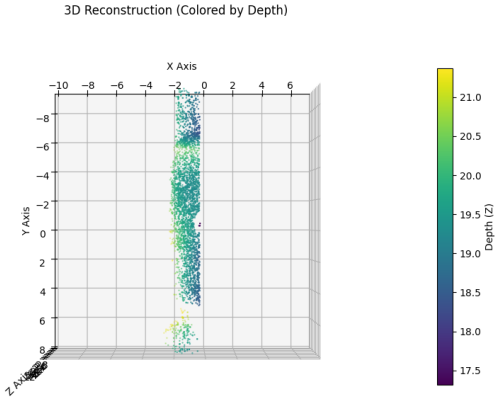

Figure 5: Qualitative failure of the SIFT+FLANN pipeline. Despite valid metrics, the reconstruction captures only a central vertical slice of the object.

4.2 Pipeline Comparison

Two different epipolar geometry estimation pipelines were evaluated: an indirect approach based on the estimation of the Fundamental matrix followed by its conversion to the Essential matrix ($\mathbf{F} \rightarrow \mathbf{E}$), and a direct approach in which the Essential matrix was estimated directly from point correspondences.

From a qualitative perspective, no visible differences were observed between the reconstructions obtained using the indirect and direct pipelines. Both approaches produced comparable camera poses, point cloud structures, and reprojection error distributions.

From a quantitative perspective, the main differences between the direct and indirect approaches are observed in the number of reconstructed 3D points and the reprojection error. Specifically, the indirect approach produces a larger number of 3D points, but at the cost of a higher reprojection error compared to the direct approach.

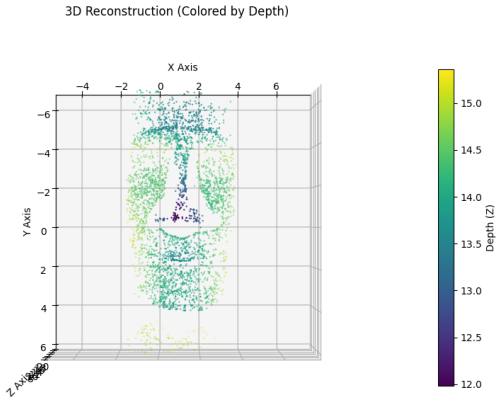

Figure 6: 3D reconstruction using the BRISK detector. Note the high accuracy and reduced noise compared to denser methods.

This similarity suggests that, given accurate camera calibration and sufficiently good feature correspondences, both pipelines are equally effective for the considered scenario. Consequently, the direct estimation of the Essential matrix can be preferred due to its simpler formulation and improved numerical stability, without compromising reconstruction quality.

5 Conclusions

This project successfully implemented and evaluated various Structure-from-Motion (SfM) pipelines for the 3D reconstruction of an object from stereo views. The experiments highlighted that the physical acquisition process is the single most critical factor in reconstruction quality; specifically, selecting a highly textured object and ensuring precise lateral camera translation (as opposed to rotation) were prerequisites for a valid geometric estimation.

Regarding the algorithmic comparison, the choice of feature detector and matcher proved decisive for the final application context. **SIFT**, **AKAZE** and **BRISK** combined with Brute-Force matching demonstrated superior performance for high-fidelity tasks, producing dense point clouds that captured fine surface details. In contrast, **ORB** proved to be robust alternatives for real-time scenarios. While it yielded sparser reconstructions, its computational efficiency and ability to retrieve the essential geometric structure make it ideal for latency-constrained applications.

Ultimately, this study confirms that there is no single "best" pipeline; rather, the optimal choice depends on balancing the trade-off between surface density (visual completeness) and computational speed. The source code for this project is available at: https://github.com/eliamirafiori/cv_hw_01.