P2 Phase1: Structure from Motion
Used 1 Late day

Sarthak Mehta
Department of Robotics Engineering
Worcester Polytechnic Institute
smehta1@wpi.edu

Prasham Soni
Department of Robotics Engineering
Worcester Polytechnic Institute
psoni@wpi.edu

***Abstract*—This report presents a Structure from Motion (SfM) pipeline to reconstruct a 3D scene from multiple 2D images. The process includes feature extraction and matching, camera pose estimation, triangulation, and PnP-based image registration, with refinement using Nonlinear Triangulation and Bundle Adjustment to minimize reprojection error. Key visualizations compare Fundamental Matrix enforcement, feature correspondences, triangulation accuracy, and PnP refinement, demonstrating the effectiveness of the reconstruction.**

I. Phase1: SfM

A. Robust Feature Correspondence Selection Using RANSAC

In Structure-from-Motion (SfM), we first need to identify reliable feature correspondences between images. These correspondences are used to estimate the fundamental matrix, which in turn helps recover the relative motion of cameras and reconstruct 3D points. However, real-world images often contain outliers due to noise, mismatches, occlusions, and reflections. If we directly compute the fundamental matrix using all detected features, these outliers can severely degrade the accuracy of the reconstruction.

RANSAC is used to robustly estimate the fundamental matrix (F) while rejecting incorrect feature matches. By selecting only inliers, we ensure a more stable and accurate estimation of camera motion and 3D structure.

We run multiple iterations and apply an error threshold to two sets of matched keypoints (*coord1, coord2*). In each iteration, we randomly select 8 correspondences and compute the fundamental matrix (*F*) using the 8-point algorithm. The inlier check is based on the epipolar constraint:

$$x_2^T F x_1 = 0$$

where x_1 and x_2 are homogeneous coordinates. Matches with errors below the threshold are classified as inliers. The best fundamental matrix is the one with the highest inlier count.

B. Estimating the Fundamental Matrix

The fundamental matrix (F), defines how points in one image correspond to lines in the other image thus giving the Epipolar geometry between two images. Estimating F accurately is crucial in Structure-from-Motion (SfM) because it allows us to compute the essential matrix, which then helps in recovering camera motion. However, because pixel coordinates are not naturally scaled, we need to normalize them before estimation and enforce constraints to obtain a valid rank-2 matrix.

The fundamental matrix (*F*) is estimated using the normalized 8-point algorithm, following these steps:

1) **Normalize Image Coordinates**
   - Compute the centroid of the points and shift them to the origin.
   - Scale them so the average distance from the origin is $\sqrt{2}$.
   - Construct the transformation matrix T:

$$T = \begin{bmatrix} s & 0 & -s \cdot u_{\text{mean}} \\ 0 & s & -s \cdot v_{\text{mean}} \\ 0 & 0 & 1 \end{bmatrix}$$

   - Transform the points using T.
2) **Construct the Equation Matrix**
   - Use corresponding points (u_1, v_1) from Image 1 and (u_2, v_2) from Image 2 to build the equation matrix:

$$A = \begin{bmatrix} u_2u_1 & u_2v_1 & u_2 & v_2u_1 & v_2v_1 & v_2 & u_1 & v_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

3) **Compute the Fundamental Matrix**
   - Solve $Af = 0$ using SVD:

$$A = UDV^T$$

   - The last column of V gives the flattened F, which is reshaped into a 3×3 matrix.
4) **Enforce Rank-2 Constraint**
   - Perform SVD on F:

$$F = UDV^T$$

   - Set the smallest singular value to zero to force rank-2:

$$D' = \begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

   - Recompute F':

$$F' = UD'V^T$$

5) **Denormalize the Fundamental Matrix**
   - Transform F' back to original pixel coordinates:

$$F = T_2^T F' T_1$$

   - Normalize F so that $||F|| = 1$.

C. Essential Matrix from the Fundamental Matrix

The essential matrix (*E*) encodes the relative rotation and translation between two cameras in calibrated coordinates. It is derived from the fundamental matrix (*F*) by using the camera's intrinsic parameters. The process follows these steps:

1) **Compute the Essential Matrix:**
   - Given the fundamental matrix F and the camera intrinsic matrix K, compute E as:

$$E = K^T F K$$

2) **Enforce the Singular Value Constraint:**
   - The essential matrix must have two equal singular values and one zero singular value. We apply Singular Value Decomposition (SVD):

$$E = USV^T$$

   - Modify the singular values to enforce the constraint:

$$S' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

   - Recompute the corrected essential matrix:

$$E' = US'V^T$$

D. Extracting Camera Pose from the Essential Matrix

Once the essential matrix E is computed, the next step is to extract the possible camera poses. This helps determine the second camera's position and orientation relative to the first, which is crucial in SfM. To achieve this, SVD is applied to E, followed by matrix operations using a predefined matrix W.

This results in four possible solutions for the camera center C and rotation matrix R. However, only one of these solutions is correct, as the true camera pose must ensure that the reconstructed 3D points lie in front of both cameras. The correct pose is identified in the next step. Extracting the camera pose is essential for reconstructing 3D scene points, and estimating motion across multiple views.

E. Disambiguating Camera Pose

After obtaining four possible camera poses from the decomposition of the essential matrix, the next step is to determine which one represents the actual motion of the camera. Since not all four solutions correspond to a valid real-world scenario, we use a method called the cheirality check to find the correct pose.

To select the correct camera pose, we perform triangulation using each of the four candidates and check how many reconstructed 3D points have positive depth in both camera views. The pose with the highest number of valid 3D points is selected as the correct one.

1) Cheirality Check: For a triangulated 3D point X to be physically valid, its depth must be positive in both cameras. This means that when transformed into the coordinate frame of each camera, the Z-coordinate must satisfy:

$$Z_1 > 0 \quad \text{and} \quad Z_2 > 0$$

where Z_1 is the depth in the world frame, and Z_2 is the depth after transforming X into the camera coordinate system:

$$X_{\text{cam}} = R(X - C)$$

where R and C are the rotation matrix and camera center of the candidate pose. The correct pose is the one that maximizes the number of 3D points satisfying this cheirality condition.

Determining the correct camera pose is essential for accurate 3D reconstruction. If an incorrect pose is chosen, the entire scene geometry will be inverted or misaligned. By ensuring that the selected pose places the maximum number of reconstructed points in front of both cameras, we obtain a correct and consistent 3D structure that aligns with the real-world scene.

F. Linear Triangulation

1) Computing Final 3D Point Positions: Now that we have determined the correct camera pose, we use Linear Triangulation again to compute the final 3D positions of points in space.

In SfM, the goal is to reconstruct the 3D structure of a scene from multiple 2D images. Given a pair of corresponding feature points in two images and their known camera positions, Linear Triangulation allows us to compute the 3D coordinates of those points.

To recover 3D points, we first compute the camera projection matrices using the known rotation matrices R_1, R_2 and camera centers C_1, C_2. These define the projection matrices as:

$$P_1 = K[R_1 \mid -R_1 C_1]$$

$$P_2 = K[R_2 \mid -R_2 C_2]$$

Using corresponding feature points (x_1, x_2) from two images, a linear system of equations is constructed to enforce the

constraint that the 3D point must lie along the camera rays. This system is then solved using Singular Value Decomposition (SVD):

$$AX = 0$$

where A is the coefficient matrix formed from the projection equations, and X represents the homogeneous 3D point. The solution is obtained by taking the last column of the right singular vector matrix from SVD. The resulting homogeneous 3D point is then converted into its non-homogeneous form:

$$X = \frac{X_h}{X_h^4}$$

to obtain its final 3D coordinates.

After determining the correct camera pose, linear triangulation provides an initial 3D reconstruction. However, due to noise and errors, these points are not perfectly accurate. To refine the reconstruction, later steps involve nonlinear triangulation, which optimizes the 3D points by minimizing reprojection error.

G. Nonlinear Triangulation

Linear triangulation provides an initial estimate of 3D points by solving a linear system, but it minimizes algebraic error rather than the actual geometric reprojection error. Since algebraic error does not directly correspond to the Euclidean distance between the observed 2D feature points and their projected 3D locations, the reconstructed 3D points may be inaccurate. While linear triangulation is computationally efficient, its accuracy is often suboptimal, especially in the presence of noise. To improve precision, we refine the initial estimates using nonlinear triangulation, which directly minimizes the reprojection error.

1) Minimizing Reprojection Error: Given an initial estimate X_0 from linear triangulation, we refine its position by optimizing the reprojection error, which is defined as the difference between the observed 2D feature points x_{obs} and the reprojected 3D points x_{proj}:

$$\text{error} = x_{\text{obs}} - x_{\text{proj}}$$

where the projected point is computed as:

$$x_{\text{proj}} = \frac{PX_h}{PX_h^3}$$

where P is the camera projection matrix, and X_h is the homogeneous 3D point. The total reprojection error across both images is minimized using a nonlinear least-squares optimization:

$$X_{\text{refined}} = \arg\min_X \sum \left\| x_1^{\text{obs}} - x_1^{\text{proj}} \right\|^2 + \sum \left\| x_2^{\text{obs}} - x_2^{\text{proj}} \right\|^2$$

where x_1^{obs} and x_2^{obs} are the observed 2D feature locations in both images, and $x_1^{\text{proj}}, x_2^{\text{proj}}$ are their corresponding projected positions from the estimated 3D point.

Nonlinear triangulation significantly improves the accuracy of 3D reconstruction by ensuring that the computed 3D points minimize geometric error rather than just algebraic error. This leads to better alignment of the reconstructed scene with the original image features, reducing distortion caused by noise. The refined 3D points obtained through nonlinear triangulation serve as a more precise input for later optimization steps such as bundle adjustment.

H. P-n-P with RANSAC

The Perspective-n-Point (PnP) problem involves estimating the camera pose (rotation and translation) given a set of 3D points and their corresponding 2D projections. Since real-world feature correspondences often contain outliers due to mismatches, PnP alone may yield inaccurate results. To make the estimation robust, we use RANSAC to filter out outliers and obtain a more reliable pose estimate.

1) PnP RANSAC Algorithm: Given a set of N 3D-to-2D correspondences, PnP RANSAC follows these steps:

1) **Random Sampling:** Randomly select a subset of 6 correspondences to estimate the camera pose using the PnP algorithm.
2) **Pose Estimation:** Solve for the candidate rotation R and translation t using the *linear PnP* method.
3) **Reprojection Error Check:** For each 3D point X, reproject it onto the image plane using the estimated pose:

$$X_{\text{cam}} = RX + t$$

$$x_{\text{proj}} = KX_{\text{cam}}$$

$$u_{\text{proj}} = \frac{x_{\text{proj}}^1}{x_{\text{proj}}^3}, \quad v_{\text{proj}} = \frac{x_{\text{proj}}^2}{x_{\text{proj}}^3}$$

The Euclidean reprojection error is computed as:

$$e = \sqrt{(u_{\text{proj}} - u_{\text{obs}})^2 + (v_{\text{proj}} - v_{\text{obs}})^2}$$

Points with errors below a predefined threshold are considered inliers.
4) **Best Model Selection:** The pose with the highest number of inliers is chosen as the best estimate.
5) **Refinement:** Optionally, after selecting the best inliers, the final pose is re-estimated using all inliers to improve accuracy.

In Structure-from-Motion, camera pose estimation is crucial for incrementally registering new images to the reconstructed 3D structure. PnP with RANSAC ensures robustness against mismatches and noisy feature correspondences, leading to more stable camera localization. By selecting the pose with the highest inlier count and refining it with all inliers, we improve accuracy, ensuring consistent alignment between 3D points and images.

I. Nonlinear Perspective-n-Point (PnP)

Linear PnP provides an initial estimate of the camera pose by minimizing algebraic error, but it does not directly optimize the reprojection error. Since the reprojection error is the Euclidean distance between the observed 2D image points and their reprojected 3D counterparts, minimizing it leads to a more accurate pose estimation. Nonlinear PnP refines the initial pose by solving an optimization problem that minimizes this geometric reprojection error.

1) Optimization Using Nonlinear PnP: Given an initial estimate of the camera center C and rotation matrix R, we refine them using nonlinear optimization. The cost function for the optimization is defined as the sum of squared reprojection errors:

$$\sum_i \left\| x_i^{\text{obs}} - x_i^{\text{proj}} \right\|^2$$

where x_i^{obs} are the observed 2D points and x_i^{proj} are the corresponding projected points computed as:

$$X_{\text{cam}} = RX + t$$

$$x_{\text{proj}} = KX_{\text{cam}}$$

$$u_{\text{proj}} = \frac{x_{\text{proj}}^1}{x_{\text{proj}}^3}, \quad v_{\text{proj}} = \frac{x_{\text{proj}}^2}{x_{\text{proj}}^3}$$

where K is the camera intrinsic matrix and X_{cam} is the 3D point in camera coordinates. Instead of representing R as a matrix, we parameterize it as a unit quaternion q, which avoids redundancy and ensures a valid rotation.

2) Optimization Process:

1) Convert the initial rotation matrix R into a unit quaternion q.
2) Define the cost function based on the reprojection error.
3) Use nonlinear least squares optimization to minimize the cost function.
4) Extract the refined camera pose by converting the optimized quaternion back to a rotation matrix.

Nonlinear PnP significantly improves camera pose accuracy compared to linear PnP by directly minimizing reprojection error. This ensures better alignment between the estimated camera pose and the actual image observations. In the Structure-from-Motion pipeline, refining the camera pose through nonlinear optimization reduces errors in subsequent steps like triangulation and bundle adjustment, leading to a more accurate and stable 3D reconstruction.

J. Building the Visibility Matrix

In Structure-from-Motion, different cameras observe different subsets of 3D points. The visibility matrix encodes this information by indicating which cameras can see which 3D points. This is crucial for optimizing reprojection errors efficiently, as it allows computations to be restricted to only observed points, leveraging the sparse nature of the observations.

1) Defining the Visibility Matrix: The visibility matrix V is an $I \times J$ binary matrix, where I represents the number of cameras and J represents the number of reconstructed 3D points. Each entry is defined as:

$$V_{ij} = \begin{cases} 1, & \text{if camera } i \text{ observes 3D point } j \\ 0, & \text{otherwise} \end{cases}$$

To construct V, we extract a subset of feature correspondences based on valid indices and convert them into a binary matrix where 1 indicates visibility and 0 indicates non-visibility.

The visibility matrix enables efficient computation of reprojection errors by only considering points visible to a given camera. This is particularly useful in bundle adjustment, where sparse optimization techniques can be used to significantly reduce computational complexity. By leveraging the structure of V, we improve both the accuracy and efficiency of 3D reconstruction.

K. Bundle Adjustment and Chirality Correction

In our SfM pipeline, bundle adjustment plays a critical role in refining both the camera poses and the 3D point estimates simultaneously. The goal is to minimize the overall reprojection error, which is the discrepancy between the observed 2D feature locations and the 2D projections of the estimated 3D points. This non-linear optimization step ensures that the reconstruction is globally consistent across all views.

L. Formulation

Our bundle adjustment is formulated as a non-linear least-squares problem. The parameter vector, $\mathbf{p}$, consists of:

- Camera parameters: For each camera, we use a minimal representation consisting of a Rodrigues rotation vector and a translation vector. If there are n_c cameras, these parameters account for $6 \times n_c$ elements.
- 3D point coordinates: For each 3D point, we include its (x, y, z) coordinates, contributing $3 \times n_p$ parameters for n_p points.

Thus, the total number of parameters is $6n_c + 3n_p$.

Given a set of observations (i.e., the 2D feature locations in each image), the cost function is defined as the sum of squared reprojection errors:

$$\min_{\mathbf{p}} \sum_{i=1}^{n_{\text{obs}}} \|\mathbf{u}_i - \pi(\mathbf{R}_{c(i)}, \mathbf{t}_{c(i)}, \mathbf{X}_{p(i)}, K)\|^2,$$

where:

- $\mathbf{u}_i$ is the observed 2D point.
- $\pi(\cdot)$ denotes the projection function given the camera parameters (rotation $\mathbf{R}_{c(i)}$ and translation $\mathbf{t}_{c(i)}$, where $\mathbf{t} = -\mathbf{R}\,\mathbf{C}$) and the intrinsic matrix K.
- $\mathbf{X}_{p(i)}$ is the corresponding 3D point.

We employ `scipy.optimize.least_squares` with a sparse Jacobian to efficiently solve the optimization. The Huber loss is used to mitigate the influence of outliers. The optimization terminates when the cost function's tolerance (set

to 1×10^{-6}) is reached or after a maximum of 200 function evaluations.

M. Chirality Check and Correction

During our initial reconstruction, we observed that many of the projected 3D points appeared to lie behind the camera. This issue, often manifesting as a nearly linear (collinear) structure in the recovered scene, is indicative of errors in the initial camera pose estimates.

To address this, we incorporated a chirality check. The chirality check involves:

- Transforming the 3D points into the camera coordinate system using the estimated camera pose.
- Evaluating the depth (the third coordinate) of each point. A valid reconstruction requires that most points have positive depth (i.e., lie in front of the camera).

Bundle adjustment helps correct this by refining the camera parameters such that the reprojection error is minimized. As a result, the majority of the 3D points are reprojected in front of the camera. Figure 1 shows the chirality check after bundle adjustment, where points in front of the camera are clearly distinguished.

N. Visualization and Results

To qualitatively assess the improvements achieved by bundle adjustment, we generate several visualizations:

1) **Chirality Check After Bundle Adjustment:** Figure 1 displays the chirality check, highlighting that most 3D points are now correctly located in front of the camera.
2) **Reconstruction After Bundle Adjustment:** Figure 2 presents a top-view of the 3D reconstruction after bundle adjustment, illustrating the overall structure.
3) **Comparison of Pre- and Post-Bundle Adjustment:** Figure 3 overlays the pre-bundle adjustment (pre-BA) point cloud (displayed in blue) with the refined (post-BA) point cloud (displayed in green), enabling a direct visual comparison of the improvement.

These visualizations, combined with the quantitative mean reprojection error metrics printed at various stages (for linear triangulation, non-linear triangulation, and both linear and non-linear PnP), confirm that the bundle adjustment significantly improves the accuracy of the camera poses and the 3D structure. The correction of the chirality issue (i.e., ensuring points lie in front of the camera) was critical to obtaining a realistic V-shaped reconstruction.

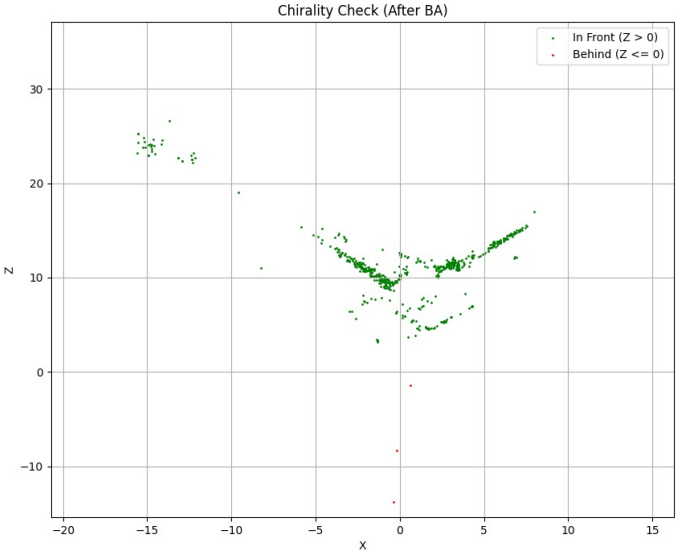

Fig. 1. Chirality check after bundle adjustment. The majority of the 3D points are now correctly projected in front of the camera.

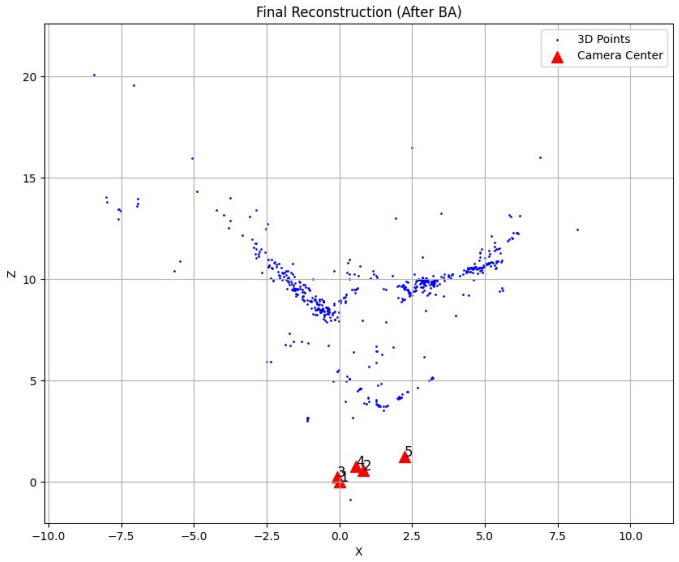

Fig. 2. Top view of the 3D reconstruction after bundle adjustment.

II. Conclusion

In this project, we developed a complete Structure-from-Motion pipeline that integrates feature extraction, fundamental matrix estimation, camera pose recovery, triangulation, and bundle adjustment. Our pipeline employs both linear and non-linear methods at various stages to obtain an initial reconstruction and then refine it for improved accuracy. One of the critical challenges we encountered was the chirality issue—where many 3D points were initially being projected behind the camera. We addressed this problem through a robust chirality check and by incorporating bundle adjustment, which corrected the camera poses and resulted in a more realistic 3D reconstruction.

The experimental results demonstrate a significant reduction in the mean reprojection error after applying bundle adjustment, as summarized in Table I. While the initial PnP methods yielded high errors, the final reconstruction error was considerably lower after refinement, indicating that our optimization strategy and the inclusion of bundle adjustment were effective in enhancing the overall reconstruction quality.

Overall, our approach successfully reconstructs a 3D scene from multiple images, with bundle adjustment playing a crucial role in ensuring that the camera poses are accurate and the reconstructed 3D points are consistent with the observed data.

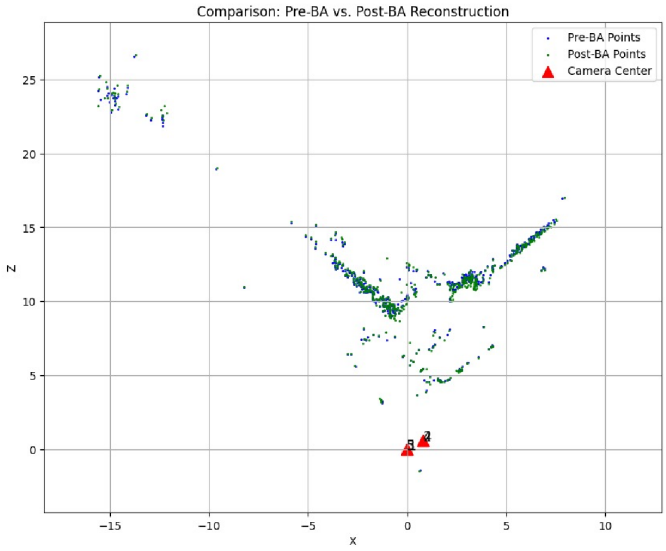

Fig. 3. Comparison of the reconstruction before bundle adjustment (blue) and after bundle adjustment (green).

Method	**Mean Reprojection Error (pixels)**
Linear Triangulation	2.28
Non-Linear Triangulation	2.27
Linear PnP	113.17
Non-Linear PnP	91.23
Bundle Adjustment	12.97

TABLE I
COMPARISON OF MEAN REPROJECTION ERRORS FOR DIFFERENT STAGES OF THE SFM PIPELINE.

The significant decrease in reprojection error, as evidenced by our experimental results, validates the effectiveness of our methodology.