Project 2a: Building Built in Minutes - SfM

Ankit Mittal
Department of Robotics Engineering
Worcester Polytechnic Institute
Email: amittal@wpi.edu

Rutwik Kulkarni
Department of Robotics Engineering
Worcester Polytechnic Institute
Email: rkulkarni1@wpi.edu

***Abstract*—(utilizing 1 late day) This report explores the Structure from Motion (SfM) technique, a method that reconstructs 3D structures from sequences of 2D images taken from different viewpoints. SfM analyzes multiple images to build a complete 3D scene and determine the positions of the camera relative to the scene as if capturing the movement of a camera through space. The approach is powerful for creating detailed 3D models from photographs. The paper outlines the basic principles of SfM, including the steps involved in matching features across images, rejecting incorrect matches, estimating the camera's position, and refining the model through optimization. The goal is to explain how SfM works, and its applications, and to encourage further research in this area.**

I. Introduction

The exploration of reconstructing three-dimensional scenes from two-dimensional images has garnered significant attention, leading to the development of methodologies like Structure from Motion (SfM). SfM, a pivotal technique in this domain, enables the creation of a rigid 3D structure of a scene by analyzing a series of images taken from different viewpoints. This approach simulates the effect of a moving camera, capturing the essence and complexity of the environment in a three-dimensional space. The notable project "Building Rome in a Day"[1], which reconstructed the entire city of Rome using publicly available photos, underscores the potential of SfM. Similarly, Microsoft Photosynth[2] represents a fascinating application of these principles, further highlighting the capability of SfM to create detailed 3D models from a collection of 2D images.

The process of SfM involves several key steps[3], each critical for accurately capturing and reconstructing the scene. This paper will delve into the intricacies of these steps, offering a comprehensive guide to understanding and implementing SfM. The sections of the paper are organized as follows:

II. Outline of the Paper

The paper is organized into sections that detail the steps involved in the SfM process:

1) **Feature Matching and Dataset:** This step identifies and matches features across the images. RANSAC is used to eliminate incorrect matches, ensuring the use of only reliable matches.
2) **Fundamental Matrix and the Outlier Rejection:** This section describes the calculation of the fundamental matrix to understand the geometric relationships between the images. It also does outlier rejection in Fundamental matrix estimation using RANSAC
3) **Estimating the Essential Matrix from the Fundamental Matrix:** It involves using the fundamental matrix to derive the essential matrix, which provides insights into the camera positions and intrinsic parameters.
4) **Computing Camera Pose from the Essential Matrix:** This part determines the camera's position and orientation based on the essential matrix.
5) **Triangular Check for Chirality:** Ensures that the reconstructed points are correctly positioned relative to the camera through triangulation.
6) **Perspective-n-Point:** Solves the Perspective-n-Point problem to accurately locate the camera's position about 3D points.
7) **Bundle Adjustment:** This final step refines the camera parameters and 3D point estimates to enhance the accuracy of the reconstruction.

This project aims to explain and implement SfM, from processing the initial images to achieving a complete 3D reconstruction. The outlined approach provides a practical framework for employing SfM, showcasing its utility in converting 2D images into 3D models.

III. Methodology

A. Feature Matching and Dataset

The Structure from Motion (SfM) process heavily relies on the identification and matching of key features across a series of images to accurately reconstruct a 3D scene from 2D inputs. In our project, we focus on a set of five images depicting Unity Hall at WPI, captured using the Samsung S22 Ultra's primary camera with settings of f/1.8 aperture, ISO 50, and a shutter speed of 1/500 sec. These images were subjected to distortion correction and resized to 800x600 pixels to prepare them for the SfM analysis.

Fig. 1: Images of Unity Hall at WPI

A step in our SfM pipeline is feature matching, which establishes correspondences between different views of the scene captured in the images. For robust and reliable feature

matching, we utilized the Scale-Invariant Feature Transform (SIFT) for key points and descriptors extraction. SIFT is chosen for its proven effectiveness in detecting invariant features to image scale and rotation, as well as offering partial invariance to changes in illumination and 3D camera viewpoint. This ensures that the features used for matching are distinctive and capable of supporting the accurate reconstruction of the 3D scene from the 2D image set.

To facilitate the matching process, key points and their descriptors were extracted and matched across the images. This process identifies correspondences between the images, which are crucial for the subsequent reconstruction steps in the SfM workflow. Each pair of images within the set undergoes this matching process, allowing for the creation of a comprehensive dataset of matched points. These matched points serve as the foundation for estimating the scene's geometry and the camera's motion, essential components of the SfM algorithm.

B. Fundamental Matrix and the Outlier Rejection

Accurately computing the fundamental matrix (F) is crucial, especially when dealing with noisy data from SIFT feature descriptors. Given the presence of noise and potential outliers, the RANSAC algorithm is employed alongside the estimation of F to ensure the inclusion of the maximum number of inliers, which is essential for mitigating the impact of noise. The process starts with the normalized 8-point algorithm, chosen for its effectiveness in dealing with noisy data. This method normalizes the points to enhance the stability of the computation, addressing the issue that epipolar lines may not exactly pass through the center of point correspondences.

1) Fundamental Matrix: The fundamental matrix is linearly estimated using these normalized points. The epipolar constraint that guides this estimation is given by the equation:

$$\begin{bmatrix} x_i' & y_i' & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = 0,$$

which expands to

$$x_i x_i' f_{11} + x_i y_i' f_{21} + x_i f_{31} + y_i x_i' f_{12} + \quad (1)$$

$$y_i y_i' f_{22} + y_i f_{32} + x_i' f_{13} + y_i' f_{23} + f_{33} = 0. \quad (2)$$

When the equation is simplified for m correspondences, it is represented as follows:

$$\begin{bmatrix} x_1x_1' & x_1y_1' & x_1 & y_1x_1' & y_1y_1' & y_1 & x_1' & y_1' & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_mx_m' & x_my_m' & x_m & y_mx_m' & y_my_m' & y_m & x_m' & y_m' & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{21} \\ f_{31} \\ f_{12} \\ f_{22} \\ f_{32} \\ f_{13} \\ f_{23} \\ f_{33} \end{bmatrix} = 0 \quad (3)$$

Due to noise, F may initially be full rank (3), necessitating adjustment to rank 2 by setting the smallest singular value to zero. This step ensures F accurately reflects the epipolar geometry. Employing RANSAC with the normalized 8-point algorithm for estimating F addresses the challenges posed by noisy, outlier-filled data, ensuring a more accurate estimation crucial for 3D reconstruction in SfM applications.

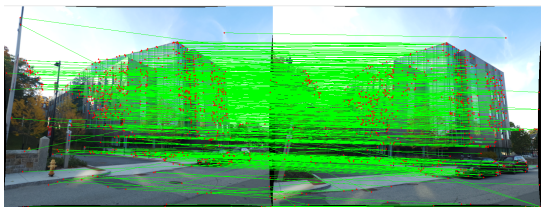

Fig. 2: Feature Matching for 2 images in the Dataset (before RANSAC)

2) Outlier Rejection in Fundamental Matrix Estimation using RANSAC: The RANSAC procedure is used to reject outliers in the computation of the fundamental matrix from image correspondences. It operates by initializing a counter for inliers and iteratively selecting random subsets of 8-point correspondences to estimate the fundamental matrix. For each iteration, it creates a set of inliers by testing all correspondences against the estimated matrix and a threshold. The set with the greatest number of inliers determines the robust fundamental matrix. This method effectively filters out inconsistent data, ensuring a reliable fundamental matrix for SfM.

```
n=0;
for i = 1:M do
    // Choose 8 correspondences, x̂1 and x̂2 randomly
    F = EstimateFundmentalMatrix(x̂1, x̂2);
    S = ∅;
    for j = 1:N do
        if |x1j^T F x2j| < ε then
            S = S ∪ {j}
        end
    end
    if n <| S | then
        n =| S |;
        S_in = S
    end
end
```

Fig. 3: Algorithm for RANSAC

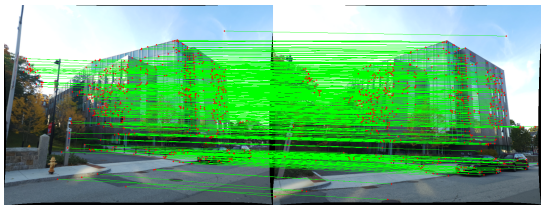

Fig. 4: Feature matching for 2 images in the dataset after outlier rejection (After RANSAC)

C. Estimating Essential Matrix using Fundamental Matrix

Given the Fundamental Matrix F computed from epipolar constraints and the camera calibration matrix K, which holds

the intrinsic parameters of the camera, we can derive the Essential Matrix E. The Essential Matrix is computed using the equation $E = K^\top FK$. This step merges the geometric relations captured by F with the internal camera characteristics, such as focal length and optical center, contained in K. The intrinsic matrix K is responsible for translating image points from pixel coordinates to normalized coordinates, centering them around the optical center of the image.

After the computation of E, Singular Value Decomposition (SVD) is applied to decompose it. The resulting singular values are then enforced to the configuration (1, 1, 0), which is a necessary adjustment due to the presence of noise in the measurements. This constraint is essential as it ensures that E correctly encapsulates the rotation and translation between two camera views. The Essential Matrix thus derived is instrumental in ascertaining the relative poses of the cameras, a pivotal step toward the three-dimensional reconstruction of the scene. Contrary to the Fundamental Matrix which is defined in the pixel coordinate space, the Essential Matrix operates within normalized image coordinates—points adjusted such that their origin coincides with the camera's optical center.

D. Computing Camera Pose from Essential Matrix

The camera pose, characterized by six degrees of freedom (rotation and translation), is estimated from the Essential Matrix E. Singular Value Decomposition (SVD) of E yields U, D, and V, leading to four possible camera poses due to ambiguity in rotation R and translation C. The rotation matrices R and translation vectors C are computed as follows, where W is a matrix used to enforce a proper rotation:

$$R_1 = UWV^T, \qquad C_1 = U(:,3) \tag{4}$$

$$R_2 = UWV^T, \qquad C_2 = -U(:,3) \tag{5}$$

$$R_3 = UW^TV^T, \qquad C_3 = U(:,3) \tag{6}$$

$$R_4 = UW^TV^T, \qquad C_4 = -U(:,3) \tag{7}$$

with $W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. To ensure $R \in SO(3)$, we enforce $\det(R) = 1$; if $\det(R) = -1$, we correct by negating C and R.

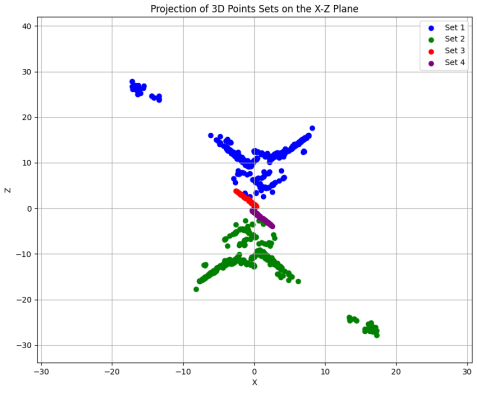

Fig. 5: All Possible Camera Poses

E. Triangular Check for Chirality

1) Linear Triangulation: In the task of triangulating 3D points from a pair of images, given two camera poses (C_1, R_1) and (C_2, R_2), and their corresponding image points $x_1 \leftrightarrow x_2$, the correct camera pose is determined by the chirality condition. This condition ensures that the reconstructed 3D points lie in front of both cameras. For a point X, its depth Z in the camera coordinate system, with respect to the camera center, must be positive when projected onto the z-axis of the camera, denoted as $r_3^T(X - C) > 0$, where r_3 is the third row of the rotation matrix R. Among the four configurations obtained from the essential matrix decomposition, the valid pose is the one for which the maximum number of triangulated points satisfy this chirality condition, thus resolving the ambiguity inherent in the reconstruction process.

2) Non-Linear Triangulation: Nonlinear triangulation refines the estimation of 3D point locations $\mathbf{X}$ by minimizing the reprojection error, a geometrically meaningful measure compared to the algebraic error minimized in linear triangulation. This error, calculated for two camera views, is the sum of squared differences:

$$\sum_{j=1,2} \left(u^j - \frac{\mathbf{P}_1^{j\top}\widetilde{\mathbf{X}}}{\mathbf{P}_3^{j\top}\mathbf{X}} \right)^2 + \left(v^j - \frac{\mathbf{P}_2^{j\top}\widetilde{\mathbf{X}}}{\mathbf{P}_3^{j\top}\mathbf{X}} \right)^2 \tag{8}$$

where j indexes each camera, $\widetilde{\mathbf{X}}$ is the homogeneous form of $\mathbf{X}$, and $\mathbf{P}_i^j$ is the i-th row of the camera projection matrix $\mathbf{P}$ for camera j. Due to the nonlinear nature of the error function, stemming from the division by the third row of the projection matrix, an initial estimate $\mathbf{X}_0$ obtained via linear triangulation is used as a starting point. The optimization is carried out using nonlinear least squares methods to iteratively adjust the 3D point estimates and converge to a solution that minimizes the reprojection error.

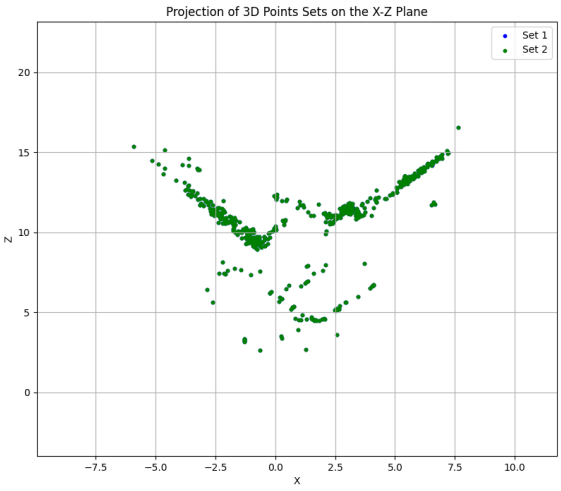

Fig. 6: Linear Triangulation(Set 1) v/s Non-Linear Triangulation(Set 2)

3) Reprojection Error Between Linear and Non-Linear Triangulation

***Reprojection Error: (–Value–)**:*

F. Perspective-n-Points (PnP)

1) PnP RANSAC: Perspective-n-Points (PnP) problems often contain outliers in the set of point correspondences. To counteract these outliers and enhance the robustness of the camera pose estimation, we utilize the RANSAC algorithm. This iterative method selects random subsets of at least six 3D-2D correspondences $\mathbf{X} \leftrightarrow \mathbf{x}$ to estimate the camera pose using a linear PnP solution, which involves solving a system of equations that relate image points and scene points via the camera's intrinsic matrix K. The estimated pose is then validated by calculating the reprojection error for all correspondences; those with errors less than a defined threshold ϵ are deemed inliers. The camera pose with the greatest number of inliers is chosen as the final estimate.

$n = 0$
for $i = 1{:}M$ **do**
  // Choose 6 correspondences, $\hat{X}$ and $\hat{x}$, randomly
  $[\mathrm{C\ R}] = \mathrm{LinearPnP}(\hat{X}, \hat{x}, \mathrm{K})$;
  $\mathcal{S} = \emptyset$;
  for $j = 1{:}N$ **do**
    // Measure Reprojection error
    $e = \left(u - \frac{P_1^T \tilde{X}}{P_3^T \tilde{X}}\right)^2 + \left(v - \frac{P_2^T \tilde{X}}{P_3^T \tilde{X}}\right)^2$;
    if $e < \epsilon_r$ **then**
      $\mathcal{S} = \mathcal{S} \cup \{j\}$
    end
  end
  if $n < |\mathcal{S}|$ **then**
    $n = |\ \mathcal{S}\ |$;
    $\mathcal{S}_{in} = \mathcal{S}$
  end
end

Fig. 7: Algorithm for PnP RANSAC

2) Non-Linear PnP: The initial camera pose obtained from linear PnP can be further refined through non-linear optimization to minimize reprojection error, which is defined geometrically and hence offers a more meaningful error metric. This process is formalized as the following minimization problem:

$$\min_{\mathbf{C},\mathbf{R}} \sum_{j=1}^{J} \left(u^j - \frac{\mathbf{P}_1^{j\top} \widetilde{\mathbf{X}}_j}{\mathbf{P}_3^{j\top} \widetilde{\mathbf{X}}_j}\right)^2 + \left(v^j - \frac{\mathbf{P}_2^{j\top} \widetilde{\mathbf{X}}_j}{\mathbf{P}_3^{j\top} \mathbf{X}_j}\right)^2 \tag{9}$$

Here, $\widetilde{\mathbf{X}}_j$ is the homogeneous representation of the 3D point $\mathbf{X}_j$, and $\mathbf{P}_i^j$ denotes the i-th row of the camera projection matrix $\mathbf{P}$, constructed as $\mathbf{P} = \mathbf{KR}[\mathbf{I}_{3\times3}| - \mathbf{C}]$. The rotation matrix $\mathbf{R}$ is parameterized using quaternions to enforce its orthogonality, expressed as $\mathbf{R}(q)$, where q represents the quaternion. The non-linear nature of this minimization arises from the division operations in the reprojection error and the quaternion representation of the rotation. Optimization begins with the initial solution $(\mathbf{C}_0, \mathbf{R}_0)$ from linear PnP.

Sr. No.	Method	Reprojection Error
1	Linear Triangulation	50.7
2	Non-Linear Triangulation	3.2
3	PnP RANSAC	1190.5
4	Non-Linear PnP	205.57

TABLE I: Comparison Between Reprojection error

G. Bundle Adjustment

Having computed all camera poses $\mathbf{P}$ and 3D points $\mathbf{X}$, the next step is to refine both to enhance accuracy and achieve optimal values through a process known as *Bundle Adjustment*. This necessitates the construction of a *Visibility Matrix* $\mathbf{V}$, which ascertains the relationship between cameras and points, denoted as V_{ij}, where j represents the j^{th} point visible to camera i.

Given N image points, N_{3d} world points, and n_C cameras (with the maximum number of cameras being 6, corresponding to the number of provided image files), each camera is characterized by 6 extrinsic parameters (Rotation: roll, pitch, yaw; Translation: cx, cy, cz). The sparsity matrix $\mathbf{M}_{ba}$, which facilitates Bundle Adjustment, has dimensions $2N \times (N_{3d} \times 3 + n_C \times 6)$. If, for instance, the image point at index 12 in N correlates to the world point at index 12 in N_{3d}, then the corresponding elements in the matrix $\mathbf{M}_{ba}$ are set to 1, indicating their relationship.

The Bundle Adjustment process refines the location points and camera poses through an optimization technique, often employing the *Trust Region Reflective Algorithm*—a method particularly suited for sparse problems. This results in refined 3D points $\mathbf{X}'$ and camera poses $\mathbf{P}'$, thus concluding the pipeline with significant improvements in accuracy. The impact of refinement can be assessed by comparing the conditions before and after the application of Bundle Adjustment.

H. Results

In the below figure blue points (set 1) refer to the 3D points before bundle adjustment, and green points (set 2) refer to the 3D after bundle adjustment.

Sr. No.	Method	Reprojection Error
1	Non-Linear PnP	1157.9
2	Bundle Adjustment	1009.6

TABLE II: Comparison Between Reprojection error

The above table shows the re-projection error for all the 3D points present in the feature map.

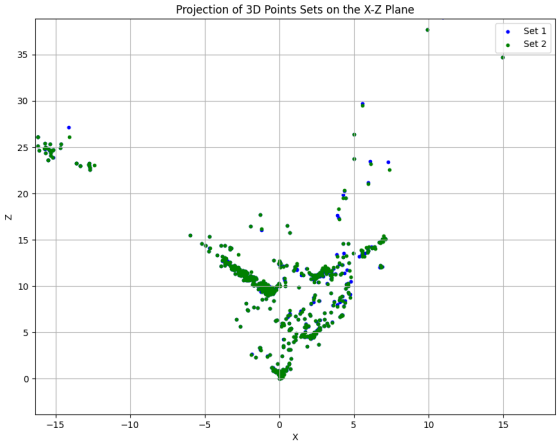

Fig. 8: Before and After Bundle adjustment

IV. Acknowledgment

The author would like to thank Prof. Nitin Sanket, Teaching Assistant, and Grader of this course RBE549- Computer Vision.

References

[1] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski, "Building rome in a day," in *2009 IEEE 12th International Conference on Computer Vision*, 2009, pp. 72–79.

[2] C. Zimmerman, "Review: Microsoft Photosynth and Stanford Humanities Lab," *Journal of the Society of Architectural Historians*, vol. 69, no. 3, pp. 463–466, 09 2010. [Online]. Available: https://doi.org/10.1525/jsah.2010.69.3.463

[3] Y. Lao, "3d vision geometry for rolling shutter cameras," Ph.D. dissertation, 05 2019.