MCEN 5228: Project 3 - Blob The Builder: Gaussian Splatting

Rama Chaganti
rama.chaganti@colorado.edu
Team Roronoa
using 3 late day

Srikanth Popuri
srikanth.popuri@colorado.edu
Team Roronoa
using 3 late day

***Abstract*—This project focuses on reconstructing a 3D model of a scene using Structure from Motion (SfM) techniques applied to a set of 2D images captured from varying perspectives. The dataset includes six images of a street scene containing a building, a text file describing 2D image point correspondences across all image pairs, and the calibration matrix of the camera used for capturing the images. The methodology involves leveraging these correspondences and camera calibration data to recover the 3D structure of the scene, illustrating the power of SfM in creating spatially accurate 3D reconstructions from photographic data. The results demonstrate the feasibility of transforming 2D image sets into a coherent 3D representation, offering valuable insights for applications in photogrammetry, computer vision, and related fields.**

I. Introduction

The objective of this project was to reconstruct a 3D model of a scene using multiple 2D images captured from various perspectives, ensuring overlapping regions between them for accurate correspondence. The dataset provided included six images of a street scene featuring a building, along with a text file detailing the 2D image point correspondences across all possible image pairs. Additionally, the camera's calibration matrix was provided to facilitate accurate mapping between the image coordinates and the real-world geometry. link to code

II. Methodology

A. Feature Matching and Outlier Rejection

Feature matching establishes correspondences between 2D points across multiple images, which is fundamental for 3D reconstruction. In this project, the provided 2D correspondences between image pairs were refined using RANSAC (Random Sample Consensus) to eliminate outliers and improve robustness.

RANSAC-Based Outlier Rejection: RANSAC was used to iteratively refine point correspondences:

1) A random subset of matches was selected to estimate the fundamental matrix F.
2) The epipolar constraint was applied:
$$x'^T F x = 0,$$
where x and x' represent corresponding points in the two images.
3) The residual error was calculated for all points:
$$\text{Residual Error} = |x'^T F x|.$$
Matches with residual error below a predefined threshold ϵ were classified as inliers.
4) This process was repeated for a fixed number of iterations, and the fundamental matrix with the highest number of inliers was retained.

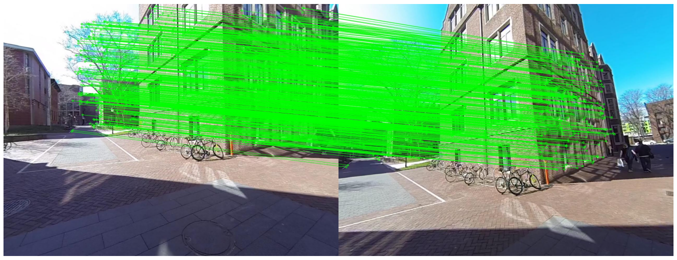

Fig. 1: After Ransac between 1 and 2 images

Normalization of Points: To ensure numerical stability during fundamental matrix computation, the image points were normalized. Each point (x, y) was transformed into normalized coordinates $(\tilde{x}, \tilde{y})$ using:

$$\tilde{x} = \frac{x - \bar{x}}{s},$$
$$\tilde{y} = \frac{y - \bar{y}}{s},$$

where $\bar{x}$ and $\bar{y}$ are the centroid coordinates of the points, and s is a scaling factor defined as:

$$s = \sqrt{\frac{1}{n}\sum_{i=1}^{n}((x_i - \bar{x})^2 + (y_i - \bar{y})^2)}.$$

Fundamental Matrix Estimation: The fundamental matrix F was computed using the normalized 8-point algorithm:

1) Construct the design matrix A based on the normalized points:
$$A = \begin{bmatrix} \tilde{x}_1\tilde{x}'_1 & \tilde{x}_1\tilde{y}'_1 & \tilde{x}_1 & \tilde{y}_1\tilde{x}'_1 & \tilde{y}_1\tilde{y}'_1 & \tilde{y}_1 & \tilde{x}'_1 & \tilde{y}'_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \tilde{x}_n\tilde{x}'_n & \tilde{x}_n\tilde{y}'_n & \tilde{x}_n & \tilde{y}_n\tilde{x}'_n & \tilde{y}_n\tilde{y}'_n & \tilde{y}_n & \tilde{x}'_n & \tilde{y}'_n & 1 \end{bmatrix}.$$
2) Solve the linear system $A \cdot \text{vec}(F) = 0$ using Singular Value Decomposition (SVD).
3) Enforce the rank-2 constraint on F by zeroing out the smallest singular value.
4) Denormalize F to transform it back to the original coordinate system.

B. Estimation of Fundamental and Essential Matrices

The fundamental and essential matrices are critical components in Structure from Motion, as they encapsulate geometric relationships between corresponding points in pairs of images. This section describes their estimation processes.

Fundamental Matrix Estimation: The fundamental matrix F defines the epipolar constraint between two images:

$$x'^T F x = 0,$$

where x and x' are corresponding points in homogeneous coordinates in the first and second images, respectively. The computation of F follows these steps:

1) **Normalization:** Image points were normalized to improve numerical stability using:

$$\tilde{x} = \frac{x - \bar{x}}{s}, \quad \tilde{y} = \frac{y - \bar{y}}{s},$$

where $\bar{x}$ and $\bar{y}$ are the centroid coordinates, and s is a scale factor.

2) **8-Point Algorithm:** Using inlier matches from RANSAC, the normalized points were used to construct the design matrix A. Solving $A \cdot \text{vec}(F) = 0$ using Singular Value Decomposition (SVD) yields F.

3) **Rank-2 Constraint:** To enforce the physical constraint of a rank-2 matrix, the smallest singular value of F was set to zero.

4) **Denormalization:** The final F was denormalized to revert it to the original coordinate system.

Computed Fundamental Matrix of Img 1 and 2: The computed fundamental matrix F is:

$$F = \begin{bmatrix} -3.6577 \times 10^{-7} & -1.0160 \times 10^{-5} & 2.5410 \times 10^{-3} \\ 1.2349 \times 10^{-5} & -4.3877 \times 10^{-7} & -4.8269 \times 10^{-3} \\ -4.0483 \times 10^{-3} & 2.6191 \times 10^{-3} & 1.0000 \end{bmatrix}.$$

Essential Matrix Estimation: The essential matrix E relates normalized image coordinates and incorporates the camera's intrinsic parameters:

$$E = K'^T F K,$$

where K and K' are the intrinsic camera matrices of the two images.

1) Using the computed fundamental matrix F, E was calculated as:

$$E = K^T F K,$$

assuming identical camera intrinsics for both images.

2) E was further refined to ensure it satisfied the constraints of the essential matrix by enforcing:

$$\det(E) = 0 \quad \text{and} \quad \text{two equal singular values.}$$

This was achieved by decomposing E using SVD:

$$E = U \Sigma V^T,$$

where $\Sigma = \text{diag}(s, s, 0)$, with s being the average of the first two singular values. The refined E was then reconstructed as:

$$E = U \Sigma V^T.$$

Computed Essential Matrix: The computed essential matrix E is:

$$E = \begin{bmatrix} -0.0208 & -0.7567 & -0.3311 \\ 0.9055 & -0.0379 & 0.3717 \\ 0.2125 & -0.5414 & -0.1395 \end{bmatrix}.$$

C. Camera Pose Estimation

The camera pose defines the rotation and translation of a camera relative to the world coordinate system. Using the essential matrix E, the relative pose (rotation and translation) between two cameras can be extracted.

Decomposition of the Essential Matrix: The essential matrix E encodes the relative motion between two views. It is decomposed into a rotation matrix R and a translation vector t as follows:

$$E = U \Sigma V^T,$$

where U, Σ, and V are obtained through Singular Value Decomposition (SVD). The matrix Σ is constrained to $\text{diag}(1, 1, 0)$ to satisfy the properties of E. From U and V, four possible solutions for R and t are derived:

$$R_1 = UWV^T, \quad R_2 = UW^TV^T, \quad t_1 = U[:,3], \quad t_2 = -U[:,3],$$

where:

$$W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

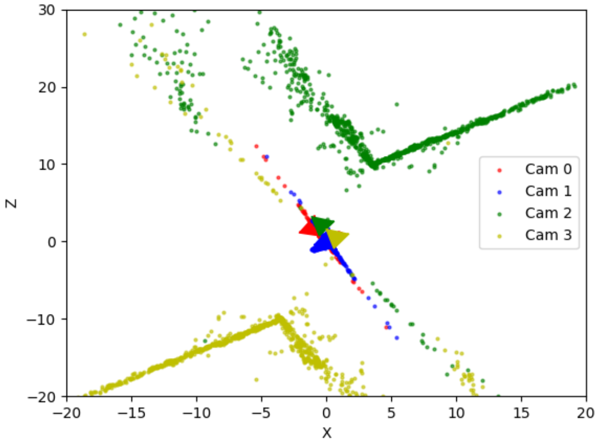

Fig. 2: four poses with 3D points

Selecting the Correct Camera Pose: To identify the correct R and t, the cheirality condition is applied. This condition ensures that 3D points reconstructed from corresponding image points lie in front of both cameras (i.e., have positive depth). The steps are:

1) For each candidate pose (R, t), 3D points are triangulated using linear triangulation (described in Section *Triangulation*).

2) For each 3D point X, the depth is calculated in the coordinate system of both cameras. The point X must satisfy:

$$Z > 0,$$

where Z is the depth component of the 3D point in the camera's local coordinate system.

3) The pose with the highest number of points satisfying the cheirality condition is selected as the correct pose.

Assumption for the First Camera: The pose of the first camera is fixed at the world origin for reference:

$$R_1 = I, \quad t_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Projection Matrix Formulation: Once R and t are determined, the projection matrix P for a camera is given by:

$$P = K[R \mid t],$$

where K is the intrinsic matrix, R is the rotation matrix, and t is the translation vector. This projection matrix is used for subsequent triangulation and reconstruction tasks.

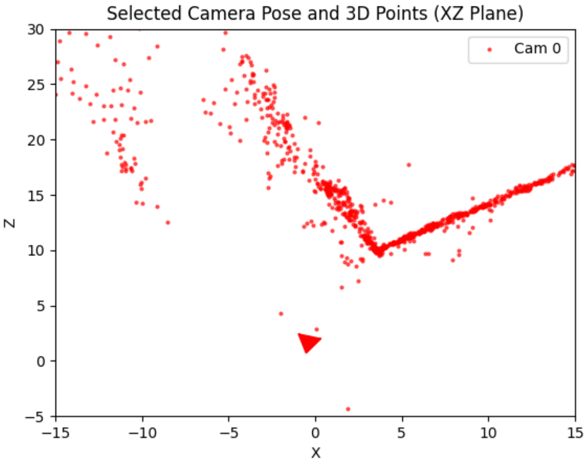

Fig. 3: Selected Pose and Points

D. Triangulation

Triangulation is the process of reconstructing 3D points from corresponding 2D points in multiple images. Using the known camera poses and projection matrices, this process computes the 3D coordinates of the scene points.

Linear Triangulation: Linear triangulation provides an initial estimate of the 3D points by solving a system of linear equations derived from the projection matrices of two cameras and their corresponding 2D points. For a 3D point X, the relation between its projection in the two images and the projection matrices P_1 and P_2 is given by:

$$x_1 = P_1 X, \quad x_2 = P_2 X,$$

where x_1 and x_2 are the homogeneous coordinates of the corresponding 2D points in the first and second images, respectively.

Rewriting the above equations, we derive a system of linear equations for X:

$$\begin{bmatrix} x_1 P_{3,1} - P_{1,1} \\ y_1 P_{3,1} - P_{2,1} \\ x_2 P_{3,2} - P_{1,2} \\ y_2 P_{3,2} - P_{2,2} \end{bmatrix} X = 0,$$

where $P_{i,j}$ represents the j-th row of the i-th projection matrix.

The solution is obtained by solving this system using Singular Value Decomposition (SVD). The 3D point X is the last column of the V matrix resulting from SVD.

Cheirality Check: To ensure the reconstructed points lie in front of both cameras, the cheirality condition is applied:

$$Z > 0,$$

where Z is the depth of the 3D point in the camera's coordinate system. Points that do not satisfy this condition are discarded or adjusted.

Non-Linear Triangulation: Linear triangulation often yields suboptimal results due to noise in the data. To refine the 3D points, non-linear optimization is employed, minimizing the reprojection error:

$$\text{Reprojection Error} = \sum_{i=1}^{N} \|x_i - \pi(P_i X)\|^2,$$

where x_i is the observed 2D point, $\pi(P_i X)$ is the reprojected 2D point, and P_i is the projection matrix for the i-th camera.

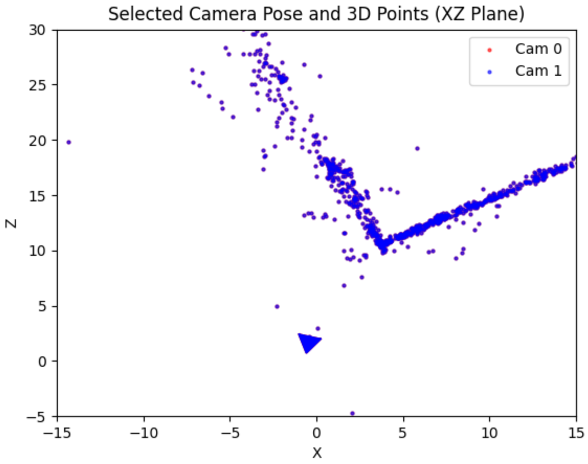

Fig. 4: After Nonlinear Triangulation

This optimization is typically performed using the Levenberg-Marquardt algorithm or similar non-linear solvers, iteratively adjusting X to minimize the error.

Visibility and Optimization: For efficient non-linear triangulation across multiple images, the visibility of each 3D point is considered. A point X is triangulated only if it is visible in a sufficient number of images. The visibility matrix V is defined as:

$$V_{ij} = \begin{cases} 1 & \text{if the point } j \text{ is visible in image } i, \\ 0 & \text{otherwise.} \end{cases}$$

The visibility matrix guides the selection of point-image correspondences for non-linear optimization.

Output: The output of the triangulation process includes the 3D coordinates of all reconstructed points. These points serve as the basis for further refinement during bundle adjustment.

E. Perspective-n-Points (PnP)

The Perspective-n-Points (PnP) problem involves estimating the pose (rotation and translation) of a camera given a set of 3D world points and their corresponding 2D image projections. This step is essential for registering new images into the 3D reconstruction pipeline.

Problem Formulation: Given $n \geq 3$ correspondences between 3D world points X_i and their 2D image projections x_i, the objective is to find the camera pose $[R \mid t]$ such that:

$$x_i = \pi(PX_i),$$

where $P = K[R \mid t]$ is the camera projection matrix, K is the intrinsic matrix, R is the rotation matrix, t is the translation vector, and π represents the perspective projection.

Linear PnP: A linear solution to the PnP problem involves constructing a system of equations using the projection relation:

$$\begin{bmatrix} X & Y & Z & 1 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix},$$

where (u, v) are the 2D image coordinates and (X, Y, Z) are the 3D world coordinates.

This relation can be rewritten in a linear least-squares form. The solution is obtained by stacking equations for all n correspondences and solving for P using Singular Value Decomposition (SVD). The rotation matrix R and translation vector t are extracted from the resulting P.

Non-Linear PnP: Linear PnP methods are sensitive to noise and may not enforce constraints like the orthogonality of R. To address this, non-linear optimization is used to refine the pose by minimizing the reprojection error:

$$\text{Reprojection Error} = \sum_{i=1}^{n} \|x_i - \pi(K[R \mid t]X_i)\|^2.$$

The optimization is performed using iterative solvers such as the Levenberg-Marquardt algorithm.

RANSAC-Based PnP: To handle outliers in the 3D-2D correspondences, a RANSAC-based approach is employed:

1) Randomly sample a minimal set of correspondences ($n = 3$).
2) Estimate the camera pose using a linear PnP method.
3) Compute the reprojection error for all correspondences.
4) Classify points as inliers if their reprojection error is below a threshold.
5) Repeat the above steps for a fixed number of iterations and select the pose with the highest number of inliers.

Output: The PnP step outputs the camera pose $[R \mid t]$ for each additional image. These poses are used to extend the 3D reconstruction and perform subsequent triangulation for new points.

F. Bundle Adjustment

Bundle Adjustment is an optimization technique used to refine the 3D structure and camera poses simultaneously by minimizing the reprojection error across all images and 3D points. This step ensures global consistency in the reconstruction and improves the accuracy of the 3D model.

Problem Formulation: The goal of Bundle Adjustment is to minimize the reprojection error:

$$\text{Reprojection Error} = \sum_{i=1}^{N} \sum_{j=1}^{M} V_{ij} \|x_{ij} - \pi(P_i X_j)\|^2,$$

where:

- N: Number of cameras.
- M: Number of 3D points.
- V_{ij}: Visibility matrix, where $V_{ij} = 1$ if the 3D point X_j is visible in camera i, and 0 otherwise.
- x_{ij}: 2D projection of the 3D point X_j in camera i.
- $P_i = K[R_i \mid t_i]$: Projection matrix for camera i.
- $\pi(P_i X_j)$: Reprojection of X_j in the image plane of camera i.

The optimization variables include:

- R_i and t_i: Rotation and translation of each camera.
- X_j: 3D coordinates of each point.

Sparse Optimization: Given the large number of parameters involved, Bundle Adjustment employs sparse optimization techniques:

- The Jacobian matrix of the error function is sparse, as each 3D point X_j affects only the cameras in which it is visible.
- Sparse solvers such as the Levenberg-Marquardt algorithm are used to efficiently compute the parameter updates.

Visibility Matrix: The visibility matrix V determines which 3D points are observed in which images. It is defined as:

$$V_{ij} = \begin{cases} 1 & \text{if the point } j \text{ is visible in image } i, \\ 0 & \text{otherwise.} \end{cases}$$

Only visible points contribute to the reprojection error, reducing the computational complexity.

Steps in Bundle Adjustment:

1) **Initialization:** The initial estimates of camera poses $[R, t]$ and 3D points X are obtained from prior triangulation and PnP steps.
2) **Error Function:** Compute the reprojection error for all visible points using:
$$e_{ij} = x_{ij} - \pi(P_i X_j).$$
3) **Jacobian Computation:** Construct the sparse Jacobian matrix of the error function with respect to all variables (R, t, X).
4) **Optimization:** Use the Levenberg-Marquardt algorithm to iteratively update the camera poses and 3D points to minimize the total reprojection error.
5) **Convergence Check:** The algorithm stops when the change in the error between iterations falls below a threshold.

Output: The output of Bundle Adjustment includes:

- Refined 3D points X_j.
- Optimized camera poses R_i and t_i for all cameras.
- Reduced reprojection error across all images.

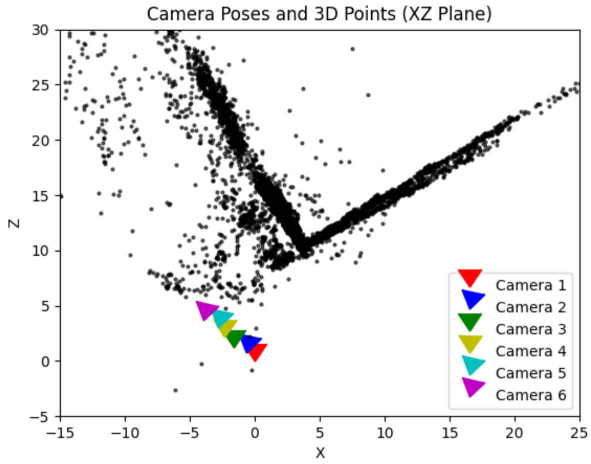

Fig. 5: Bundle Adjustment Output

III. Challenges and Observations

During the course of the project, several challenges were encountered, ranging from data quality issues to algorithmic limitations. This section outlines these challenges and the observations made during their resolution.

A. Challenges

- **Noisy Feature Matches:** Initial feature correspondences contained significant noise, especially in repetitive regions like windows or textures. This required fine-tuning the RANSAC threshold and increasing the number of iterations to obtain robust matches.
- **Sparse Correspondences:** Some image pairs had sparse matches due to a lack of distinctive features, leading to difficulty in estimating the fundamental matrix accurately.
- **Triangulation Instability:** Linear triangulation produced points behind the camera for certain poses, requiring rigorous application of the cheirality condition.
- **Bundle Adjustment Convergence:** Optimizing a large number of parameters during bundle adjustment led to slow convergence, necessitating the use of a sparse Jacobian to improve efficiency.
- **Parameter Sensitivity:** The performance of RANSAC, PnP, and non-linear optimization was highly sensitive to parameter choices, requiring extensive experimentation.

B. Observations

- Robust outlier rejection using RANSAC significantly improved the quality of matches, laying a strong foundation for subsequent processes.
- Incorporating non-linear optimization (both for triangulation and PnP) reduced the overall reprojection error and improved accuracy.
- Bundle adjustment demonstrated its effectiveness in achieving globally consistent 3D reconstructions, despite being computationally intensive.
- The intrinsic matrix K played a crucial role in ensuring the accurate computation of the essential matrix and subsequent pose estimation.
- Visualization of results, such as reprojection error and 3D point clouds, provided critical insights into algorithmic performance and errors.

IV. Conclusion

This project successfully reconstructed a 3D model of a scene using SfM techniques. The application of optimization methods such as bundle adjustment significantly enhanced the quality of the reconstruction. Future improvements could explore advanced feature matching algorithms and real-time processing.

References

[1] http://cmp.felk.cvut.cz/cmp/courses/TDV/2013W/lectures/tdv-2013-07-anot.pdf
[2] http://cis.upenn.edu/˜cis580/Spring2016/Lectures/cis580-18-coursera-2016-SfM-full.pdf
[3] https://www.uio.no/studier/emner/matnat/its/UNIK4690/v16/forelesninger/lecture73-pose-from-epipolar-geometry.pdf
[4] https://scipy-cookbook.readthedocs.io/items/bundleadjustment.html