Project 2 - Buildings built in minutes - SfM and NeRF
Phase 1

Manoj Velmurugan*, Rishabh Singh†
Robotics Engineering
Worcester Polytechnic Institute
Email: *v.manoj1996@gmail.com, †rsingh8@wpi.edu
USING ONE LATE DATE

***Abstract*—This project implements Structure from Motion (SfM) algorithm to reconstruct three-dimensional scenes and estimate camera poses from a set of images. SfM leverages a series of two-dimensional images captured from different viewpoints or a moving camera to come up with a cohesive, rigid structure of the scene. By employing principles found in stereoscopic photogrammetry, SfM calculates the relative three-dimensional poses of objects through triangulation methods, ultimately giving us point cloud-based 3D models.**

I. Introducion

Structure from Motion (SfM) is a computer vision technique used for reconstructing three-dimensional scenes and estimating camera poses from a sequence of images. It involves several interconnected steps that collectively enable the generation of accurate 3D point clouds (not necessarily to scale). These steps typically include feature matching and outlier rejection using methods such as Random Sample Consensus (RANSAC), estimating the Fundamental Matrix, deriving the Essential Matrix from the Fundamental Matrix, and subsequently estimating camera poses. Furthermore, we check for the Cheirality Condition through triangulation, employing techniques like Perspective-n-Point (PnP), and performing Bundle Adjustment to refine the reconstructed scene and camera parameters.

II. Novelty in this work

Pytorch is one true optimization library

- Pytorch is used for all the linear algebra work and optimization work.
- Most of our code is vectorized to increase the performance greatly. Vectorized code runs faster because of SIMD optimization.

III. Given feature matching dataset

The SIFT feature matches were provided in a text file, which we parse to extract the relevant information. The matches are then displayed as shown in Fig. 1. This contains a lot of noisy data or wrong matches.

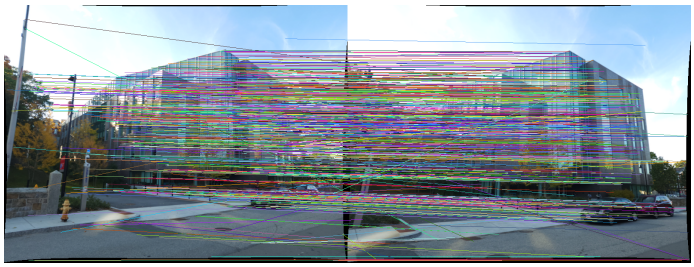

Fig. 1: Example of SIFT feature matches

IV. Estimating Fundamental Matrix

In the subject of stereo geometry, the positions of two cameras are bound by an epipolar constraint. This constraint says that if a 3D point is projected onto one camera pose, its corresponding projection onto the other pose must align along a specific line. The interconnection between these two projections is given by the Fundamental matrix. Comprising 9 unknowns, the Fundamental matrix is a representation of a system of linear equations, typically solved using Singular Value Decomposition (SVD). Following the solution of this system, the rank constraint is imposed manually. We then employ a RANSAC (Random Sample Consensus) algorithm to randomly select eight correspondences from the list of features. We then compute the fundamental matrix and evaluate the number of inliers by applying the condition $x_2^T F x_1 < thresh$. Upon identifying the maximum count of inliers, we utilize the corresponding fundamental matrix, which optimally maximizes the inlier count, to discard outliers. These outliers, are basically noisy correspondences within the provided SIFT matches, which are effectively filtered out based on this process. The final output post this process is shown in Fig. 2.

V. Estimating Essential Matrix

Next, we estimate the Essential Matrix (E), another 3x3 matrix. E can be easily extracted algebraically, $E = K^T F K$, where K denotes the camera calibration or intrinsic matrix which has already been provided to us. However, the diagonal

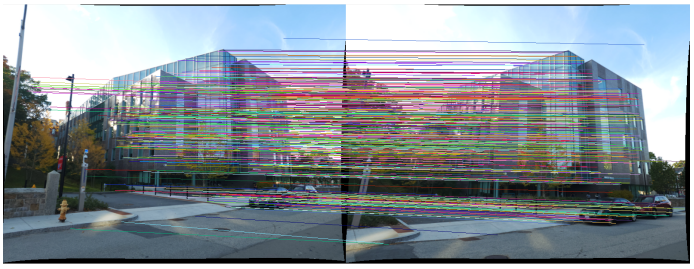

Fig. 2: Post RANSAC matches

values of E may deviate from the ideal $(1, 1, 0)$ configuration due to inherent noise within K. To solve this issue, we decompose E and then reconstruct it as expressed by $E = U \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T$ as was provided in the assignment guidelines.

VI. Estimating Camera Pose from Essential Matrix

With the calculated Essential matrix denoted as $E = UDV^T$, and defining W as:

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

We can compute the four (mathematically) possible camera configurations as follows:

1) $C_1 = U(:, 3)$ and $R_1 = UWV^T$
2) $C_2 = -U(:, 3)$ and $R_2 = UWV^T$
3) $C_3 = U(:, 3)$ and $R_3 = UW^TV^T$
4) $C_4 = -U(:, 3)$ and $R_4 = UWV^T$

VII. Disambiguate Camera Pose

After identifying the four potential camera pose configurations by decomposing the essential matrix in the preceding step, we used linear triangulation with Cheirality constraints to eliminate three impossible camera poses. Although all these poses are theoretically/mathematically valid, only one camera pose is practically feasible; i.e the "correct" camera pose is one where maximum of the 3D world point X lies in front of the camera. This circumstance occurs under the condition $r_3(X - C) > 0$, where r_3 represents the third row of the rotation matrix (the z-axis of the camera). By applying this condition, we figured out the correct camera pose.

VIII. Linear and Non-Linear Triangulation

We can first linearly find the 3D points, given sets of matching features using two camera poses, (C_1, R_1) and (C_2, R_2), and correspondences, $x_1 \leftrightarrow x_2$, triangulate 3D points using linear least squares.

The chosen 3D points serve as starting points for a nonlinear optimizer, which accounts for the error in reprojecting 3D points from images. The resulting estimate of the world points is considerably denser than the original. The reprojection error

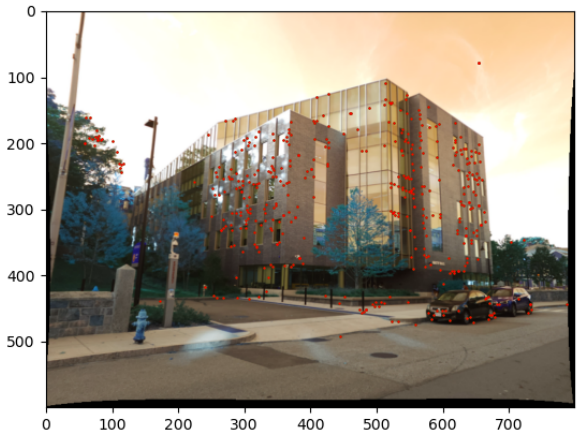

Fig. 3: Linear Triangulation

can be calculated by comparing measurements with projected 3D points using the equation:

$$\min_X \sum_{j=1,2} \left((u_j - P_j^{T_1} \tilde{X} P_j^{T_3} X)^2 + (v_j - P_j^{T_2} \tilde{X} P_j^{T_3} X)^2 \right)$$

Here, j denotes the index of each camera, $\tilde{X}$ represents the homogeneous form of X, and P_i^T signifies each row of the camera projection matrix P. The initial estimate for the solution, X_0. This optimization problem can be tackled using nonlinear optimization functions like `torch.optim.Adam`.

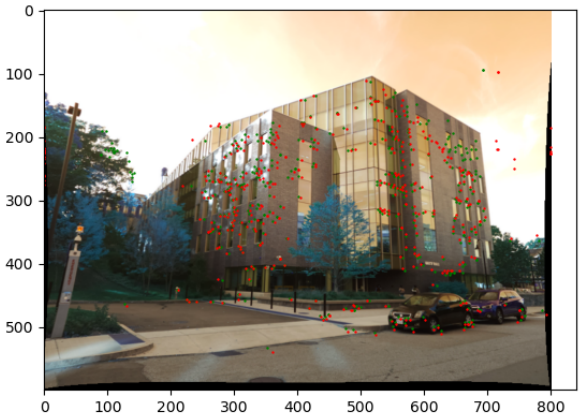

Fig. 4: Non-Linear Triangulation

IX. Perspective-n-points

The perspective-n-points (PnP) problem involves aligning the poses of new cameras based on additional images of the scene, provided that a some number of world points are visible to the original cameras used in the previous processes. It is

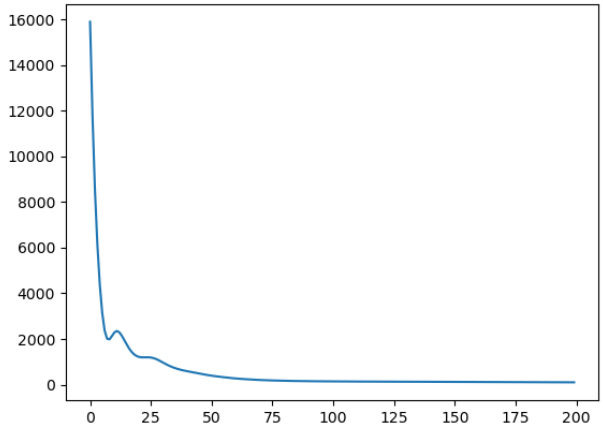

Fig. 5: Optimizer loss

presumed that there are outlier matching points from the world to the new images. Therefore, RANSAC is employed once again to filter them out.

The chosen 3D points serve as starting points for a nonlinear optimizer, which accounts for the error in reprojecting 3D points from images. The resulting estimate of the world points is considerably denser than the original, clearly showing the building's corners in the images.

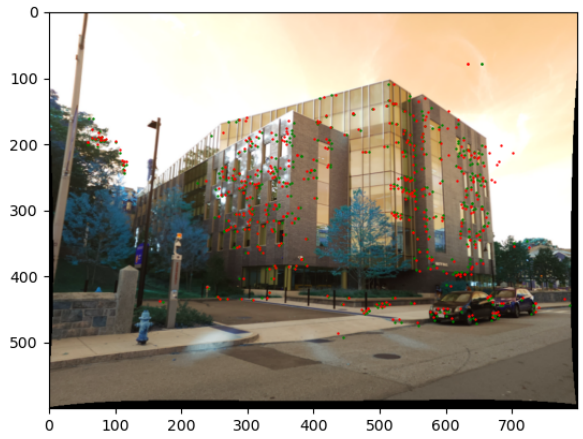

Fig. 6: Linear PNP with RANSAC

X. Bundle adjustment

After successfully aligning all five camera poses within the scene using the provided images, bundle adjustment is done to enhance the accuracy of both the camera pose estimates and the 3D world points. This process involves minimizing reprojection error once again. This iteratively refines the alignment of the cameras and the representation of the scene in three dimensions.

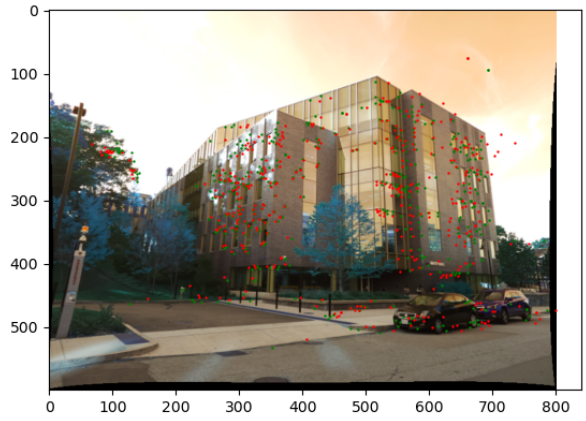

Fig. 7: Non-Linear PNP

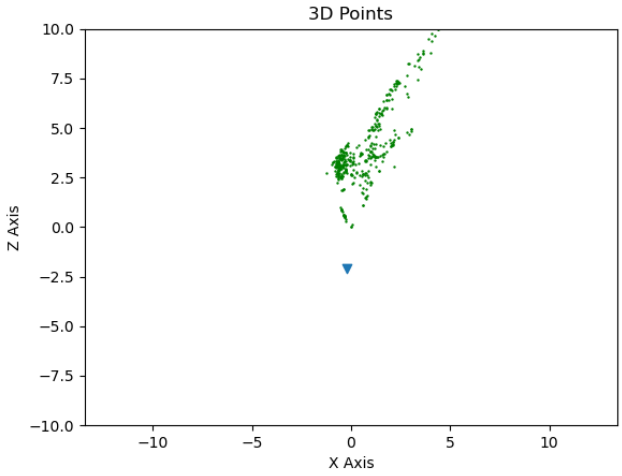

Fig. 8: Non-Linear PNP

XI. Conclusion

In conclusion, the application of Structure from Motion (SfM) algorithm has proven succesful in reconstructing three-dimensional scenes and estimating camera poses from a series of 2D images. Through the fundamental principles of epipolar geometry, triangulation, and blinear and non-linear PnP, we have been able to synthesize accurate 3D models from image data.