P2: Building Built in Minutes: SfM (Phase 1)

1st Venkateshkrishna
Masters in Robotics
Worcester Polytechnic Institute
Worcester, MA 01609
vparsuram@wpi.edu

2nd Mayank, Bansal
Masters in Robotics
Worcester Polytechnic Institute
Worcester, MA 01609
mbansal1@wpi.edu

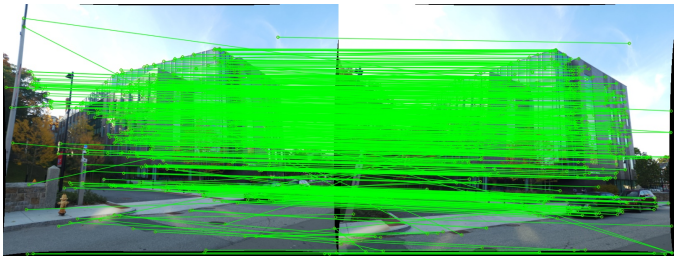

Fig. 1: Matching Features

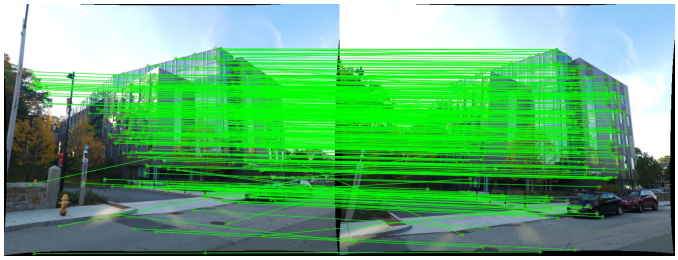

Fig. 2: Feature Matching After Ransac

***Abstract*—This project seeks to leverage computer vision technologies for the purpose of reconstructing three-dimensional scenes and determining the positioning of cameras through a process known as Structure from Motion (SfM). SfM is a computational approach that employs a sequence of two-dimensional images to extrapolate the three-dimensional structure of a scene, resulting in the generation of 3D models based on point clouds. Utilizing the foundational principles of stereoscopic photogrammetry, SfM computes the relative three-dimensional positions of objects from pairs of stereo images by applying triangulation techniques.**

I. Phase 1: Classical Approach

A. Estimating Fundamental Matrix

In stereo geometry, two camera poses are subject to an epipolar constraint. This means that if a 3D point is projected onto one of the camera poses, its corresponding projection on the other pose must lie on a line. The relationship between the two projections is captured by the Fundamental matrix.

The Fundamental matrix represents a system of linear equations with 9 unknowns, which can be solved using Singular Value Decomposition (SVD). After solving the system, the rank constraint is enforced by setting the last singular value to zero and recomputing the Fundamental matrix.

B. RANSAC

RANSAC stands for Random Sample Consensus, and it's a powerful technique often used in computer vision for identifying and removing outliers, particularly to fine-tune matches between features. The method involves selecting random subsets of the original data, then building a model from each subset. The model that fits the most data points (inliers) best is chosen, and these inliers are used to refine the model further. This procedure is carried out repeatedly until a model that meets the desired criteria is found. In this scenario, RANSAC was employed to eliminate incorrect pairings, thereby enhancing the precision of the feature matching process.

C. Estimating Essential Matrix from Fundamental Matrix

The Essential matrix takes what the Fundamental matrix does and goes a step further. It changes the focus from just looking at images to considering the actual positions and angles of the cameras in a 3D space. This means it helps us understand how the two cameras in a stereo setup relate to each other in the real world, not just how they capture images.

This extra information is really useful for figuring out the 3D layout of a scene and where exactly the cameras are, by looking at how certain points match up in both camera views. The Essential matrix also helps solve the puzzle of finding these matching points in each pair of images.

In short, the Essential matrix is a key tool for anyone working with 3D images and trying to understand or recreate the space where the photos were taken. It's very helpful for building 3D models of scenes and for many other computer vision projects.

D. Estimating Camera Pose from Essential Matrix

We applied Singular Value Decomposition (SVD) along with several mathematical methods, as outlined in the problem statement, to break down the Essential matrix into matrices representing rotation and translation. Moreover, we conducted a procedure known as SVD cleanup to verify that the resulting rotation matrix truly represented a rotation.

Decomposing the Essential matrix is a vital task for calculating the relative positions and orientations of two cameras that capture the same scene. With the rotation and translation matrices in hand, we're able to figure out how one camera is

Fig. 3: Epipolar Lines in first image

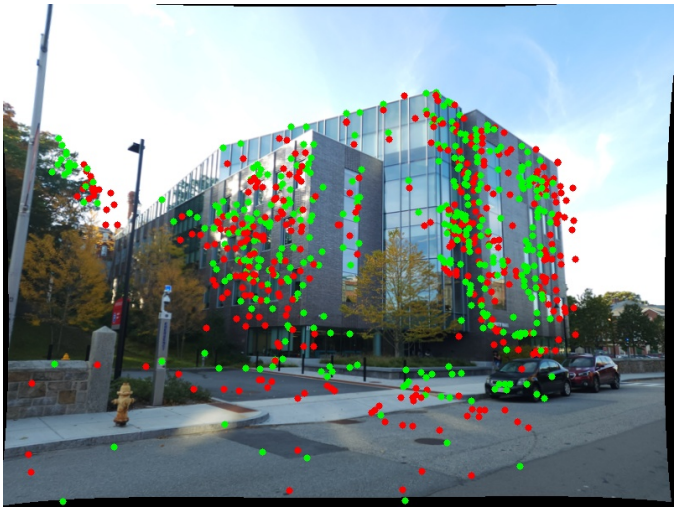

Fig. 5: Linear Triangulation Re-projection

Fig. 4: Epipolar Lines in second image

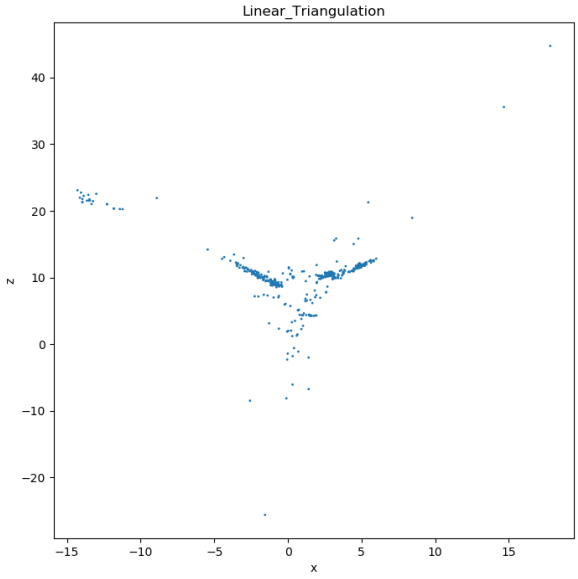

Fig. 6: Linear Triangulation

positioned and angled in relation to the other. This enables us to create 3D points from matching 2D points in images, a fundamental operation in 3D computer vision. This technique is widely used in several areas, including robotics, augmented reality, and the development of self-driving vehicles.

E. Linear Triangulation

Linear Triangulation involves determining the 3D locations of points in the world by utilizing correspondences between features, the camera matrix, and the positions and orientations of the cameras. For each pair of corresponding 2D and 3D points, two equations emerge from the projections of each camera. Ideally, the point where the lines—from a 3D point to its projections in both images—meet should give us the exact location of the 3D point. However, these lines often do not lie in the same plane, making it difficult to find a precise meeting point. To work around this, we use Singular Value Decomposition (SVD) to find the closest possible solution by solving a set of linear equations. We remove the scale factors from these equations by taking advantage of the property that the cross product of a vector with itself is zero. Additionally, to make sure we have the right solution, we perform a chirality check to confirm that the points are indeed located in front of the camera, indicated by their positive Z-coordinates.

F. Nonlinear Triangulation

Once we've established the precise orientation and location of the camera, our next objective is to reduce the discrepancy between the world point as it's re-projected and its observed position in the image. Although minimizing this error linearly can be beneficial, it often falls short of addressing the geo-

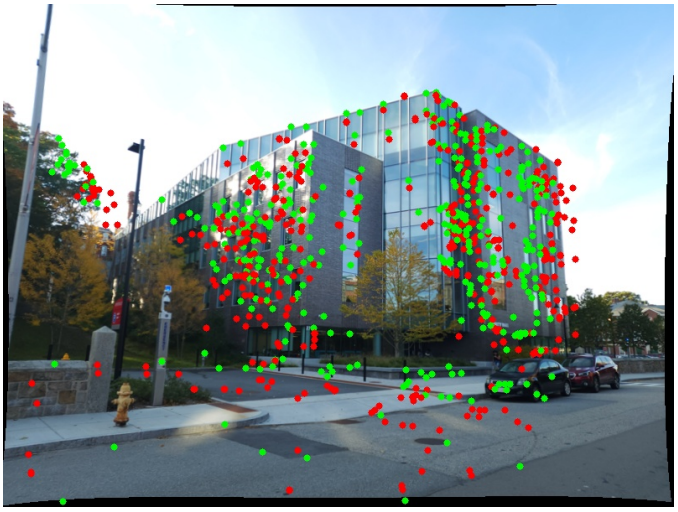

Fig. 7: Non-Linear Triangulation Re-projection

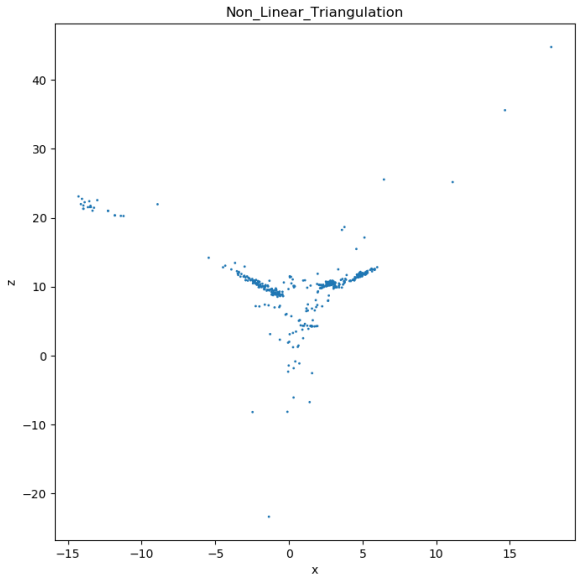

Fig. 8: Non-Linear Triangulation

metric discrepancies inherent in three-dimensional space. As a result, we focus on refining the non-linear error to better align the re-projected point with its true position in the world, utilizing the least squares method available in the scipy library for this purpose.

It's important to highlight that the non-linear triangulation method yields more precise outcomes than its linear counterpart. This enhanced accuracy stems from the non-linear technique's ability to incorporate the complexities of the camera's projection model—something the linear approach overlooks. Moreover, the non-linear method's outcomes are more closely aligned with the genuine geometry of the 3D world, affirming its superiority and reliability for accurate 3D reconstruction.

G. Perspective-n-Point (PnP)

After optimizing the world coordinates for two camera frames, the next step involved estimating the poses for the additional four frames using all images collected. The methods employed were:

1) **Linear Perspective-n-Point (PnP)**: We addressed this by solving a linear least squares problem using either 12 or 6 point correspondences to estimate the new camera pose (R and T). The extraction of R and T was facilitated through the use of SVD, specifically from the last row of V^T.
2) **PnP RANSAC**: Given PnP's susceptibility to outliers potentially skewing camera pose accuracy, we implemented RANSAC. This method enhances robustness by including points in the inlier set based on a reprojection error threshold ϵ.
3) **Nonlinear PnP**: Echoing the triangulation approach, we minimized geometric loss via least squares to secure a more accurate camera pose estimation.

Utilizing nonlinear PnP, we derived each camera's pose relative to the X points, and plotted these alongside triangulated points for both Unity hall and another building dataset. This nonlinear approach was pivotal in estimating camera poses for all remaining frames, each considered in relation to triangulated 3D points, thereby ensuring a consistent and precise 3D scene reconstruction.

H. Bundle Adjustment

Following the acquisition of camera poses and world coordinates, refinement through Bundle Adjustment is the next crucial step. This procedure optimizes the positions of cameras and points simultaneously, involving:

1) **Visibility Matrix**: Constructed upon data file analysis, this matrix tracks point visibility across cameras. As "matching" files for each image are processed, a matrix (with entries approximating 10,000 points in the case of Unity Hall datasets) is filled, marking a 1 for each point visible from a specific camera. This yields an $n \times m$ matrix for n points and m cameras.
2) **Bundle Adjustment**: Employing the Scipy least squares optimizer, as in prior steps, the visibility matrix plays a crucial role in computational efficiency by delineating necessary Jacobian calculations for expedited processing. The objective is to minimize the reprojection error, representing the variance between observed image points and their 3D projections. Adjustments to camera poses and 3D points are iteratively made to reduce this discrepancy to a minimal level. This process not only diminishes reprojection error but also enhances the precision and consistency of 3D reconstruction outcomes, facilitating the removal of outliers from initial reconstructions.

Bundle Adjustment is indispensable for refining 3D reconstruction results, ensuring the accuracy and reliability of camera poses and world coordinates.

REFERENCES

[1] Building Build in Minutes-SfM: link