RBE 549 Project 2: Buildings Built in Minutes - SfM

using 2 late days

Amrit Krishna Dayanand, Venkata Sai Krishna Bodda
MS Robotics Engineering
WPI
Email: adayanand@wpi.edu, vbodda@wpi.edu

x

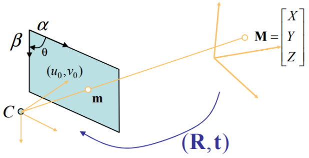
α
β
θ
(u_0, v_0)
m
C
$\mathbf{M} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$
$(\mathbf{R}, \mathbf{t})$

Fig. 1. Pinhole model of a camera is used to project a 3D point in space onto a discrete, 2D image plane

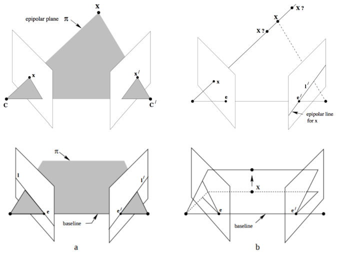

Fig. 2. The view from two cameras at arbitrary poses looking at the same world point. The world point and two camera centers lie on the epipolar plane.

I. Feature Matching

There are five input images of the same scene taken at slightly different camera poses. SIFT keypoints and descriptors can be used to perform keypoint matching across pairs of images. As discussed in the next section, keypoint matching of a point in the first image is reduced to a 1-D search along the epipolar line in the other image. In this project, we used a set of known matches, and pruned the matches using RANSAC and the fundamental matrix as our model.

II. Estimating Fundamental Matrix

The fundamental matrix describes the epipolar geometry between two image planes and a world point that lies in front of the camera pinhole ($z > 0$). The epipolar plane is described by the vector between the two camera centers (baseline) and the vector from one camera center to a world point. The intersections of the baseline with the image planes are known as the epipoles and the epipolar line in one image is the projection of the outgoing ray from the other camera center to the world point.

This forms the basic framework for estimating depth, where the coordinates of the world point, w.r.t. one camera frame, is given by the intersection of the outgoing rays from each camera center to the world point.

The fundamental matrix, F, is a 3x3 matrix that is derived using the 8-point algorithm which solves for it using algebraic least squares and at least 8 correspondences. Due to noise, the rank of F may be full, which means it has no null space, and consequently the epipolar geometry is not correctly defined. We rectify this by setting the smallest singular value of F to 0 and recomputing F.

To further improve our estimation of F, we use the known epipolar constraint and RANSAC. The epipolar constraint is a geometric constraint that states the normal to the epipolar plane is perpendicular:

$$\boldsymbol{\tilde{u_L}}^{\boldsymbol{T}} F \boldsymbol{\tilde{u_R}} = 0 \tag{1}$$

where $\boldsymbol{\tilde{u_L}}$ and $\boldsymbol{\tilde{u_R}}$ represent the homogeneous image points of the left and right cameras.

In our RANSAC model we select 8 random matches from a set of correspondences and compute the fundamental matrix. The best set of matches are those that generate the maximum number of inliers when the epipolar constraint is below some small value, $\epsilon < 0.05$. We run our RANSAC algorithm for 1000 iterations or until 99% of the inliers are recovered. The final fundamental matrix is computed from the set of inliers.

III. Estimating Essential Matrix

The essential matrix, E, describes the epipolar geometry with respect to the camera coordinate space while the fundamental matrix describes it with respect to the image space. The essential matrix encodes rotation and translation information of one camera center with respect to the other. It follows that the epipolar constraint can be written as:

$$\boldsymbol{x_L}^T E \boldsymbol{x_R} = 0 \tag{2}$$

where $\boldsymbol{x_L}$ and $\boldsymbol{x_R}$ represent projections in the camera coordinate space of the left and right cameras.

By substituting the projection equation relating the projection in the camera coordinate space to the image space into the epipolar constraint, we can derive an equation to compute E from F.

$$x = K^{-1} z\tilde{\boldsymbol{u}} \tag{3}$$

where K is the intrinsic matrix of the camera (assuming the same camera has been used in each view). Substituting this into the epipolar constraint yields:

$$\tilde{\boldsymbol{u}_{\boldsymbol{L}}}^T K^{-T} E K^{-1} \tilde{\boldsymbol{u}_{\boldsymbol{R}}} = 0 \tag{4}$$

$$K^{-T} E K^{-1} = F \tag{5}$$

Thus, the essential matrix can be derived from the fundamental matrix.

$$E = K^T F K \tag{6}$$

IV. Estimating Camera Pose from Essential Matrix

The essential matrix can be decomposed into the translation and orthonormal rotation matrices using singular value decomposition (SVD). This yields four camera poses which later need to be disambiguated:

$$\begin{aligned}
C_1 &= U(:, 3) \\
C_2 &= -U(:, 3) \\
C_3 &= C_1 \\
C_4 &= C_2 \\
R_1 &= UWV^T \\
R_2 &= R_1 \\
R_3 &= UW^TV^T \\
R_4 &= R_3
\end{aligned}$$

where C_x and R_x represent the camera center and rotation of each pose, U and V are the left and right matrix decompositions from the SVD of the essential matrix, E and W is defined as follows:

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

V. Triangulation Check for Cheirality Condition

Since there are four poses, these need to be disambiguated to identify the true orientation of the second camera with respect to the first. One simple way of doing this is the cheirality condition, which states that the reconstructed points must lie in front of the cameras. Mathematically, this means that for a world point $\boldsymbol{X}$ and camera pose defined by center $\boldsymbol{C}$ and rotation matrix R where r_3 is the third row of the rotation matrix:

$$r_3(\boldsymbol{X} - \boldsymbol{C}) > 0$$

We compute a world point through linear and non linear triangulation.

A. Linear Triangulation

Linear triangulation is performed by solving the algebraic least squares. From the relationship $\tilde{\boldsymbol{u}} = P\tilde{\boldsymbol{X}}$ where P is the projection matrix, $\tilde{X}$ is a homogeneous world point and $\tilde{u}$ is a homogeneous image point we can define the constraint, $\tilde{\boldsymbol{u}} \times (P\tilde{\boldsymbol{X}}) = 0$. This yields two constraints per image which can be stacked vertically and solved using algebraic least squares:

$$\begin{bmatrix} v\boldsymbol{P_3} - \boldsymbol{P_2} \\ -u\boldsymbol{P_3} + \boldsymbol{P_1} \end{bmatrix}$$

where $\boldsymbol{P_n}$ is a row of the projection matrix, P.

The re projection error was calculated by predicting how these points would be projected on the camera and calculating error between both the points. The Re projection error before optimization is **2.04**

B. Non-linear triangulation

Using the linear triangulation as an initial guess, we can minimize a non-linear reprojection error to find the optimal world points:

$$argmin_x \sum_{j=1}^{2} (u_j - \hat{u_j})^2 + (v_j - \hat{v_j})^2 \tag{7}$$

where j is an iterator over each image, and $\hat{u_j}$, $\hat{v_j}$ are the normalized projection of the homogeneous world point $\tilde{X}$ as defined below:

$$\hat{u_j} = \frac{P_1^{jT}\tilde{X}}{P_3^{jT}\tilde{X}}$$

$$\hat{v_j} = \frac{P_2^{jT}\tilde{X}}{P_3^{jT}\tilde{X}}$$

The re projection error was reduced to **1.9** after optimization.

VI. Estimation Camera Poses, PnP

A. Linear PnP

Upon acquiring world points using the initial two images, the subsequent step involves estimating the camera poses for the remaining images relative to the first image, which serves as the reference alignment. Leveraging the known 3D points and obtaining 2D correspondences for each image, alongside the intrinsic camera parameters encapsulated within the camera calibration matrix $\tilde{\boldsymbol{K}}$, facilitates the computation of the rotation matrix $\boldsymbol{R}$ and translation vector $\boldsymbol{C}$

Normalization of the 2D points is imperative to mitigate the intrinsic effects of the camera, achieved through the transformation $K^{-1} \cdot X$, thereby rendering the points invariant to intrinsic parameters. Given the six degrees of freedom characterizing camera pose (three Euler angles and three translations), a minimum of six 2D-to-3D correspondences is requisite to solve for the 3×4 projection matrix encompassing both rotation (R) and translation (T) parameters. This normalization step and adherence to requisite correspondence criteria are pivotal for robust camera pose estimation, underpinning subsequent stages of the reconstruction process with accuracy and reliability.

In the presented methodology, the rotational elements within the three columns of the RT matrix are ensured to be orthonormal, albeit subject to potential errors. To rectify this, a singular value decomposition (SVD) is employed, whereby only the multiplication of the left singular vectors (U) and the right singular vectors (V) is retained. Furthermore, the determinant of the resultant R matrix is calculated, and if found to be -1, the entire R matrix is multiplied by -1 to enforce proper alignment. It is noteworthy that the translation vector is situated in the third column of the RT matrix.

B. *PnPRansac*

Given the propensity of the Perspective-n-Point (PnP) algorithm to yield numerous errors, we employ the Random Sample Consensus (RANSAC) technique to mitigate outliers, leveraging re-projection error as elucidated in the preceding section.

C. *Non-Linear PnP*

Analogous to the triangulation process, wherein a linearly estimated camera pose is initially obtained, subsequent refinement of the camera pose is facilitated to minimize the reprojection error. Subsequently, a further refinement of these locations is conducted through Non-Linear Perspective-n-Point (PnP) optimization utilizing Scipy.optimize. Additionally, the rotation matrix is converted into a Quaternion representation, deemed advantageous for preserving orthogonality, along with the translation vector, during the optimization process. It is noteworthy that this minimization task entails significant nonlinearity due to the inherent divisions and parameterization with quaternions.

VII. Bundle Adjustment

With the acquisition of all camera poses and corresponding 3D points, it becomes essential to refine these points to achieve maximal accuracy and optimize both the 3D points and camera poses. To accomplish this, Bundle Adjustment is employed. Initialization of the Bundle matrix necessitates the construction of the Visibility matrix, denoted by V_{ij}, which establishes the relationship between cameras and points. Here, j signifies the j-th point visible in camera i.

Consider a scenario where there exist N image points, N_{3D} world points, and n_C cameras (where n_C is bounded by the maximum number of provided image files, which in this case is 6). Each camera is characterized by six extrinsic parameters, comprising rotation (roll, pitch, yaw) and translation (c_x, c_y, c_z). The sparsity matrix M_{ba} has dimensions $2N \times (N_{3D} \times 3 + n_C \times 6)$. In the event that an image point at index 12 in N corresponds to a world point at index 12 in N_{3D}, the elements of matrix M_{ba} pertaining to their relationship will be set to 1.

A notable advancement in the refinement process is observed through the application of the Trust Region Reflective (TRR) algorithm, a method of least squares known for its robustness in handling sparse problems. Through this method, a higher level of precision is attained in refining both the 3D points and camera poses. Consequently, the pipeline reaches its culmination with the completion of this refinement stage. A comparative analysis between the refinement outcomes before and after bundle adjustment serves to illuminate the efficacy of the refinement process.

VIII. Results and Observations

A. *Camera Pose Estimation*

There were four possible camera poses. The points generated via linear triangulation yielded two pairs of mirrored points as shown in Fig. Fig. 3. From this, it is evident that Pose 2 is the real camera pose because the Z values are positive. This was corroborated mathematically using our camera pose disambiguation function.

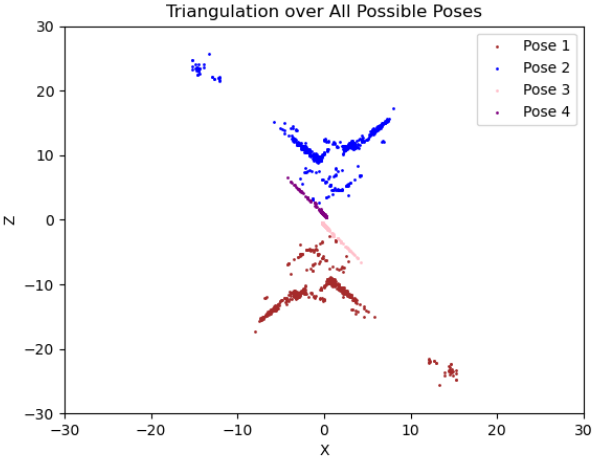

Fig. 3. The estimated 3D depth points associated with each camera pose via linear triangulation shows two sets of mirrored 3D points, of which only pose 2 is the true camera pose

After disambiguation, Fig. 4 shows the real camera pose with points generated via non-linear triangulation.

B. *Linear and Non-Linear Triangulation*

Non-linear triangulation was marginally better than linear triangulation, with a reprojection error of X and Y respectively. Since the difference was small, when both sets of 3D points were plotted on the same plot, the points overlapped, which explains why there is seemingly only one set of points plotted in Fig. 5

Type	Reproj. Error
Linear	2.04
Non-Linear	1.9
Linear PnP	N/A
Non-Linear PnP	N/A
Bundle Adjustment	N/A

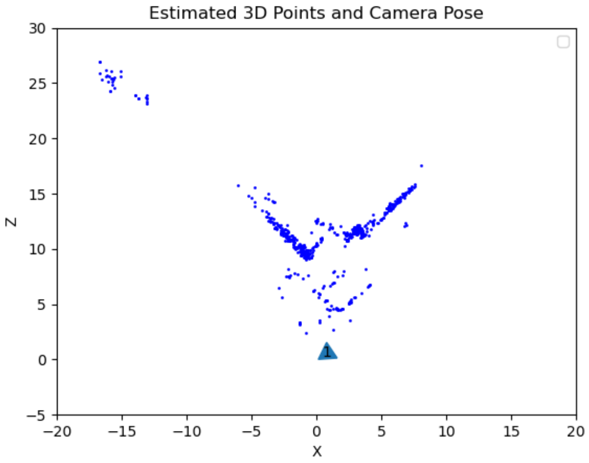

Fig. 4. The estimated 3D depth points associated with the real camera pose via non-linear triangulation

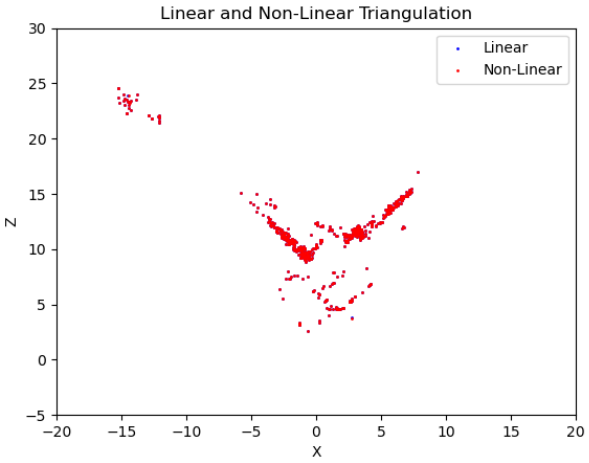

Fig. 5. The difference between the 3D points estimated via linear and non-linear triangulation varied in the order of $10^{-}5$ causing it to seem like only the non-linear points were plotted in this image

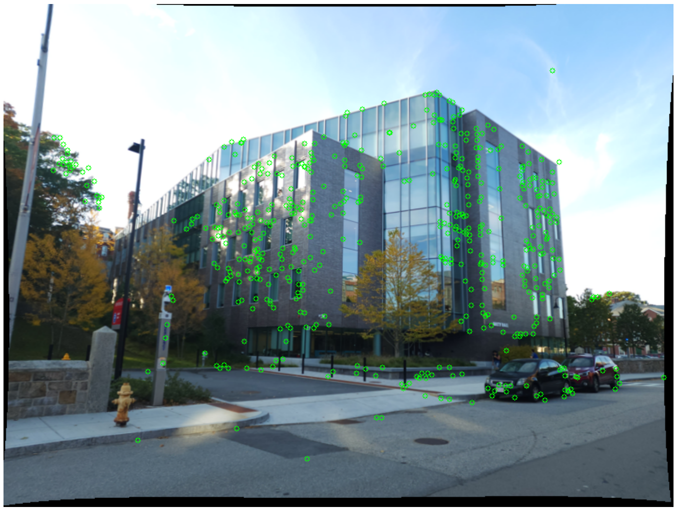

Fig. 6. Reprojection of linear triangulation depth points onto image 1

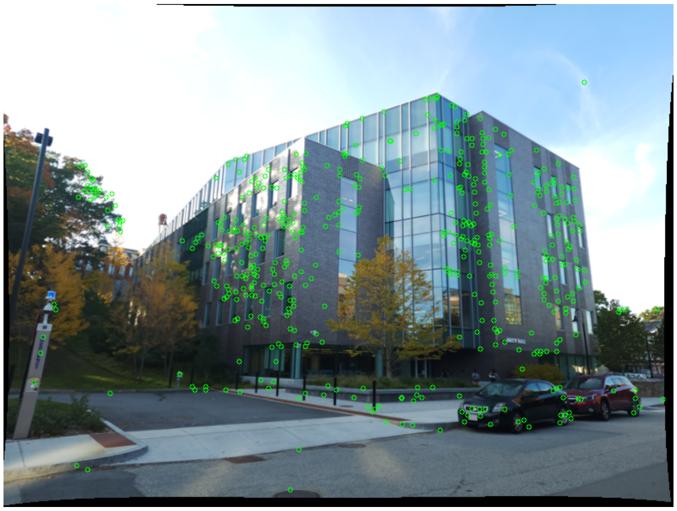

Fig. 7. Reprojection of linear triangulation depth points onto image 2

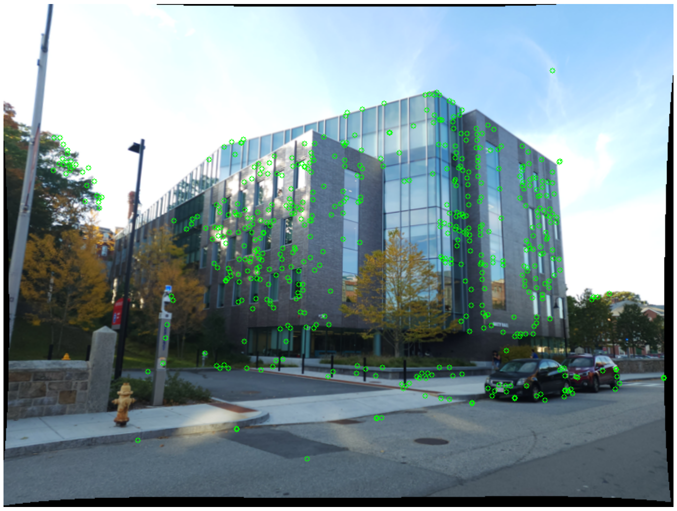

Fig. 8. Reprojection of non-linear triangulation depth points onto image 1

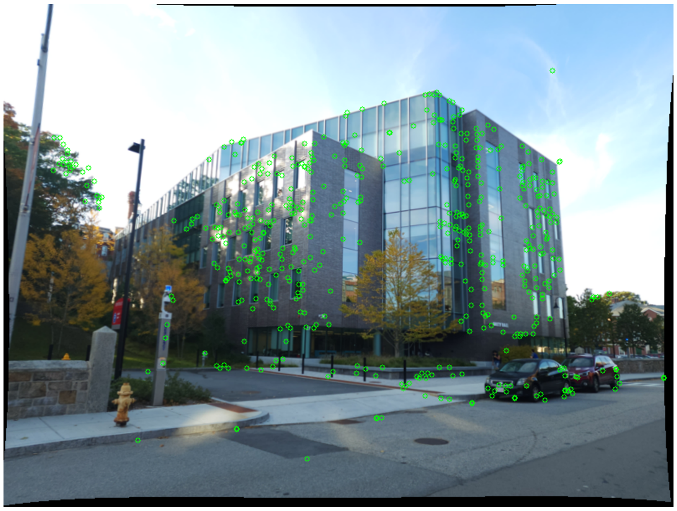

Fig. 9. Reprojection of non-linear triangulation depth points onto image 2

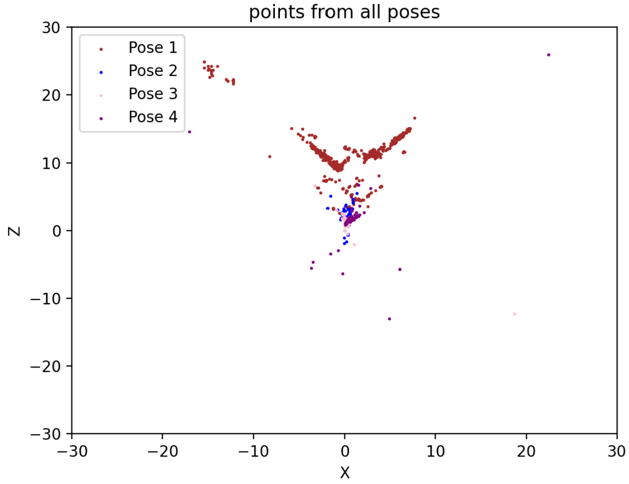

Fig. 10. The triangulation points from all the poses we're calculated and plotted in X,Z plane