RBE 549 Project 2: Buildings built in minutes - SfM and NeRF

Aabha Tamhankar
Masters in Robotics Engineering
Worcester Polytechnic Institute
astamhankar@wpi.edu

Miheer Diwan
Masters in Robotics Engineering
Worcester Polytechnic Institute
msdiwan@wpi.edu

***Abstract*—The following paper reports our implementation of a classical method of Structure for Motion (SfM) and a deep learning approach of Neural Radiance Fields (NeRF).**

I. Introduction

Structure from motion (SfM) is a photo-metric range imaging technique for estimating three-dimensional structures from two-dimensional image sequences that may be coupled with local motion signals. It is an important concept in the study of computer vision and perception. In simple terms, SfM helps us to recover 3D structure from the projected 2D (retinal) motion field of a moving object or scene.

II. Phase 1: Traditional Approach

Structure from Motion (SfM) can be defined as a method used to reconstruct a 3D scene and simultaneously obtain the camera poses of a monocular camera with respect to the given scene. Given 5 images from a camera in motion, the goal of the project is to estimate geometric values of the points in the image and represent a 3D structure out of it.

The steps taken to implement SfM in the traditional approached have been explained in detail below.

A. Data Set

A data of 5 images of Unity Hall at Worcester Polytechnic Institute captured using a camera in motion is provided. The camera calibration has already been conducted, and the images are undistorted and resized to 800×600 pixels. The data regarding matches between corresponding points of each image has also been provided using the SIFT keypoints and descriptors. This data is stored in files named "matching.X", where X is the image number, and this files stores matches of X^{th} image with the remaining images. The matches from file "*matches.1*" for image 1 and image 2 are shown in Fig.1.

B. Estimate Fundamental Matrix

The fundamental matrix, denoted by F , is a 3×3 (rank 2) matrix that relates the corresponding set of points in two images from different views (or stereo images). The F matrix can be found out using the epipolar geometry by inducing epipolar constraint or correspondence condition given by,

$$x_i'^{\,T} F x_i = 0, \tag{1}$$

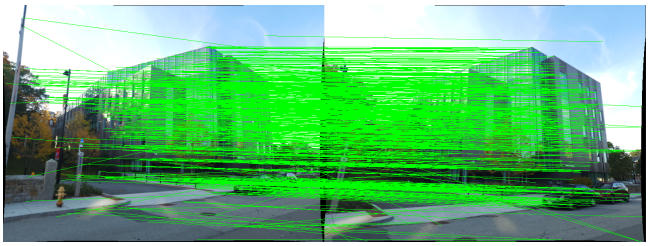

Fig. 1. Matches from Image 1 and Image 2

where x and x' are representations in first and second image of a point X in 3D space.
So, we can calculate the 3×3 fundamental matrix by,

$$\begin{bmatrix} x_i' & y_i' & 1 \end{bmatrix} * \begin{bmatrix} f11 & f12 & f13 \\ f21 & f22 & f23 \\ f31 & f23 & f33 \end{bmatrix} * \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = 0 \tag{2}$$

which is for one corresponding point in the images. Hence for m correspondences, it would become:

$$\begin{bmatrix} x_1x_1' & x_1y_1' & x_1 & y_1x_1' & y_1y_1' & y_1 & x_1' & y_1' & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_mx_m' & x_my_m' & x_m & y_mx_m' & y_my_m' & y_m & x_m' & y_m' & 1 \end{bmatrix} \begin{bmatrix} f11 \\ f21 \\ f31 \\ f12 \\ f22 \\ f32 \\ f13 \\ f23 \\ f33 \end{bmatrix} = 0 \tag{3}$$

After normalising the image points, the equation II-B was used to calculate fundamental matrices for each image. However, due to outlier correspondences, this fundamental matrix may not be accurate. For getting pure inliers, 8-point RANSAC has been performed on these sets of correspondences. These points are then further used to calculate a more accurate Fundamental Matrix. The fundamental matrix found was,

$$F = \begin{bmatrix} -5.27102668e-08 & -3.19806614e-05 & 1.38103802e-02 \\ 3.45459495e-05 & 2.50659714e-06 & -3.58130017e-02 \\ -1.57171086e-02 & 3.44113041e-02 & 1.00000000e+00 \end{bmatrix}$$

C. Estimate Essential Matrix from Fundamental Matrix

Essential matrix is used to calculate the camera poses between any two images, which are further required for triangulation and perspective. The essential matrix is calculated using the formula

$$E = K^T F K$$

, where where K is the camera calibration matrix or camera intrinsic matrix and F is the fundamental matrix. However, since we cannot get an exact value of calibration due to noise, we reconstruct the essential matrix using single value decomposition with value (1,1,0) such that,

$$E = U \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T \tag{4}$$

The essential matrix found was,

$$E = \begin{bmatrix} 0.00249426 & -0.61474787 & 0.11661446 \\ 0.66317523 & 0.04516326 & -0.73326896 \\ -0.15898169 & 0.77663055 & 0.0229181 \end{bmatrix}$$

D. Estimate Camera Pose from Essential Matrix

By geometry, four camera poses (C1, R1), (C2, R2), (C3, R3) and (C4, R4) were then obtained using the essential matrix. Though all these poses are correct in theory, practically only one camera pose is feasible, which lies in front of the image. The four camera poses found were:

$$R1 = \begin{bmatrix} 0.14380224 & 0.18499731 & 0.97216095 \\ 0.2301884 & -0.96167888 & 0.14895311 \\ 0.96246258 & 0.20236038 & -0.18087581 \end{bmatrix}$$

$$C1 = \begin{bmatrix} 0.78005122 & 0.14309272 & 0.60913428 \end{bmatrix}$$

$$R2 = \begin{bmatrix} 0.14380224 & 0.18499731 & 0.97216095 \\ 0.2301884 & -0.96167888 & 0.14895311 \\ 0.96246258 & 0.20236038 & -0.18087581 \end{bmatrix}$$

$$C2 = \begin{bmatrix} -0.78005122 & -0.14309272 & -0.60913428 \end{bmatrix}$$

$$R3 = \begin{bmatrix} 0.99722602 & 0.01775794 & 0.07228361 \\ -0.020878 & 0.99887234 & 0.04263996 \\ -0.0714449 & -0.04403082 & 0.99647223 \end{bmatrix}$$

$$C3 = \begin{bmatrix} 0.78005122 & 0.14309272 & 0.60913428 \end{bmatrix}$$

$$R4 = \begin{bmatrix} 0.99722602 & 0.01775794 & 0.07228361 \\ -0.020878 & 0.99887234 & 0.04263996 \\ -0.0714449 & -0.04403082 & 0.99647223 \end{bmatrix}$$

$$C4 = \begin{bmatrix} -0.78005122 & -0.14309272 & -0.60913428 \end{bmatrix}$$

The "correct" camera pose which is possible in practicality is the one where the 3D world point X lies in front of the camera. This can only happen under the condition,

$$r_3(X - C) > 0,$$

where r_3 is the third row of the rotation matrix (z-axis of the camera). By this cheirality condition, we find the feasible camera pose in the configuration (C,R,X).

E. Linear Triangulation

The camera pose (rotation and translation matrices) found can be used to further calculate the 3D world points from the images. This can be done by non-linear triangulation of projection matrices of every pose and image points.

$$\begin{bmatrix} \begin{bmatrix} x_1 \\ 1 \end{bmatrix} \times P_1 \\ \begin{bmatrix} x_2 \\ 1 \end{bmatrix} \times P_2 \\ \vdots \\ \begin{bmatrix} x_n \\ 1 \end{bmatrix} \times P_n \end{bmatrix} \times \begin{bmatrix} X \\ 1 \end{bmatrix} = 0 \tag{5}$$

,where x is the image points and P_x is the projection matrix of that point. SVD was performed on the first term of left side of the equation, and a singular vector of world points was obtained, which is the second term of the left side.

F. Non-Linear Triangulation

After linear triangulation, 3D points are obtained from 2D image points. However, the position of these world points can be inaccurate due to geometric error. Non-linear triangulation helps to overcome this error by optimizing the least square errors.

$$\epsilon_g = ((P_1 X)/P_3 X) - x)^2 + ((P_2 X)/P_3 X) - y)^2 \tag{6}$$

,where P_x is the projection matrix row of that image point. Thus, we can reduce the geometric error causing inaccuracy in the world point plotting.

G. Perspective-n-Points

With the given camera matrix, 3d world points and 2D image points, a 6 degree of freedom camera pose can be calculated which gives the exact position and orientation of the cameras. Linear PnP can be performed with the help of common featured already obtained from non-linear triangulation. The 2D points found are normalized with camera matrix using

$$K^-1x$$

Using 6 such 2D image points, and their corresponding 3D world co-ordinates, we are able to compute the camera pose. However, since the 2D image points may have certain outliers, the PnP thus obtained can be prone to error. Thus we can

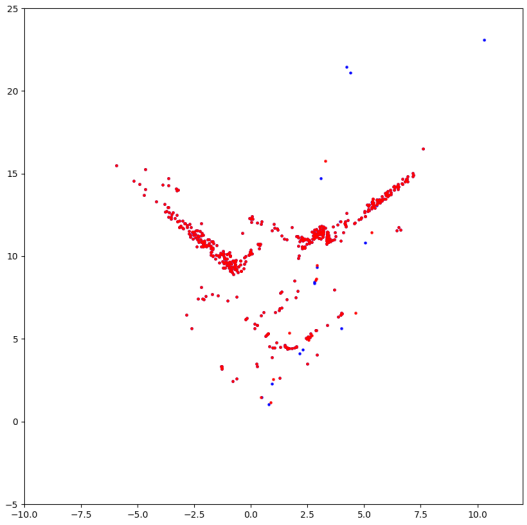

Fig. 2. Linear and Non-Linear Triangulation for Image 2 and Image 3

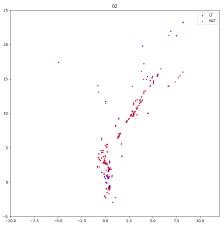

Fig. 3. Linear and Non-Linear Triangulation for Image 1 and Image 3

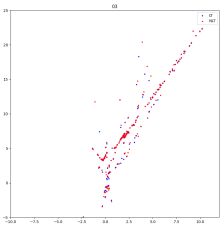

Fig. 4. Linear and Non-Linear Triangulation for Image 1 and Image 4

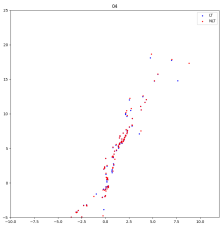

Fig. 5. Linear and Non-Linear Triangulation for Image 1 and Image 5

perform RANSAC on the undertaken 2D-3D correspondences (more than 6) to make the camera pose obtained more robust. Similar to triangulation, the linear PnP cannot account for geometric error, thus we see re-projection errors in our output. This reprojection error can be minimized using non-linear PnP.

$$\min_{C,q} \Sigma_{i=1,J}(u^j - \frac{P_1^{jT}\hat{X}_j}{P_3^{jT}\hat{X}_j})^2 + (v^j - \frac{P_2^{jT}\hat{X}_j}{P_3^{jT}\hat{X}_j})^2 \quad (7)$$

where P_j is the column of the projection matrix formed in triangulation, and $\hat{X}_j$ is the homogeneous form of world co-ordinates. This form used quarternions to optimize error and can thus be classified as non-linear PnP.

Reprojection Error				
Image	LT	NLT	L-PnP	NL-PnP
Img1-2	161.99	125.25	-	-
Img2-3	4958.75	3623.73	5221.12	4546.35

H. Bundle Adjustment and Visibility Matrix

The camera poses still need refinement initialized by minimizing reprojection error.

$$\min_{\{C_i,q_i\}_{i=1}^{i},\{X\}_{j=1}^{J}} \Sigma_{i=1,I}\Sigma_{j=1,J}V_{ij}((u^j - \frac{P_1^{jT}\hat{X}_j}{P_3^{jT}\hat{X}_j})^2 + (v^j - \frac{P_2^{jT}\hat{X}_j}{P_3^{jT}\hat{X}_j})^2) \quad (8)$$

Here,V_{ij} is the visibility matrix such that it is one if j^{th} point is visible from the i^{th} camera and zero otherwise. Thus, Visibility Matrix is of the size of all points considered after performing RANSAC. Using the above equation, we can refine the camera poses and 3D points simultaneously by minimizing the reprojection error.

III. Phase 2: Deep Learning Approach

Neural Radiance Field is a generative model of sorts, conditioned on a collection of images and accurate poses (e.g. position and rotation), that allows you to generate new views of a 3D scene shared by the images, a process often referred to as "novel view synthesis." In simple words, NeRF renders a new view of an object when given some input views.

A. Input

The basic NeRF approach represents a scene using a fully-connected (non-convolution) deep neural network, whose input is a single continuous 5D coordinate (spatial location (x,y,z) and viewing direction θ, ψ) and whose output is the volume density and view-dependent emitted radiance at that spatial location.

They synthesize views by querying 5D coordinates along camera rays and use classic volume rendering techniques to project the output colors and densities into an image. Because volume rendering is naturally differentiable, the only input required to optimize our representation is a set of images

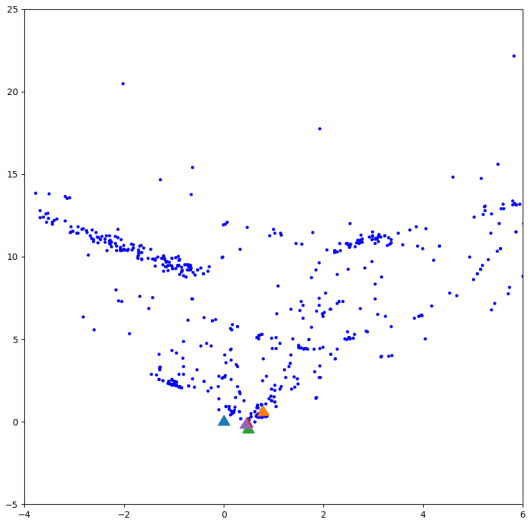

Fig. 6. Bundle Adjustment

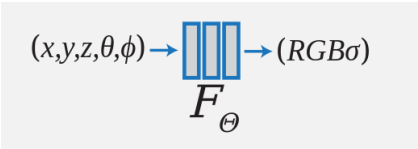

Fig. 7. Input and Output of NeRF

with known camera poses. They describe how to effectively optimize neural radiance fields to render photo-realistic novel views of scenes with complicated geometry and appearance, and demonstrate results that outperform prior work on neural rendering and view synthesis.

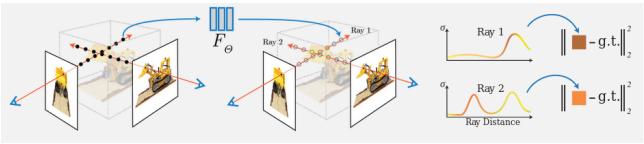

Fig. 8. NeRF Pipeline

B. Method

The following approach can be used to implement the deep learning network for the NeRF to obtain a 3D view from the given images.

1) NeRF Network: The NeRF model is 8 layers deep with feature dimension of 256 for most layers. A residual connection is placed at layer 4. After these layers, the RGB and values are produced. The RGB values are further processed with a linear layer, then concatenated with the view directions, then passed through yet another linear layer before finally being recombined with at the output.

2) Get Rays: It is clear from the input description of NeRF that it requires rays coming from each pixel of the image. Using pinhole model. direction of these rays with respect to the camera frame can be calculated. Furthermore, the rotation matrix of the camera poses converts these rays into world coordinate system. And thus the rays are generated for each pose of the object.

For accuracy, these rays need to be continuously sampled along their length. However, since that is practically impossible, feasible points along the ray lines are then consider as samples, this is called Ray Stratification.

3) Positional Encoding: NeRF model does not do well with high-frequency inputs, hence there is a high chance of getting blurred or dislocated final results. Positional Encoding can help solve this problem as it maps its continuous input to a higher-dimensional space using high-frequency functions to aid the model in learning high frequency variations in the data, which leads to sharper models. It involves into the input bands of pre-decided sin and cosine waves for the model to take as inputs.

Fig. 9. Screenshot from Rendered GIF

REFERENCES

[1] Camera Calibration and Fundamental Matrix Estimation with RANSAC
[2] Structure From Motion(SfM)
[3] https://github.com/sakshikakde
[4] https://rbe549.github.io/spring2023/proj/p2/
[5] It's NeRF From Nothing: Build A Complete NeRF with PyTorch
[6] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, Ren Ng, "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis."