RBE 549 Project 2: Buildings Built in Minutes

UdayGirish Maradana
Robotics Engineering (MS)
Worcester Polytechnic Institute
Worcester, MA 01609
Email: umaradana@wpi.edu
Using 2 Late Days

Pradnya Sushil Shinde
Robotics Engineering (MS)
Worcester Polytechnic Institute
Worcester, MA 01609
Email: pshinde1@wpi.edu
Using 2 Late Days

***Abstract*—The following report consists of a detailed analysis of a classical and deep learning approach to 3D reconstruction of a scene and simultaneously obtaining the camera poses of a monocular camera w.r.t. the given scene. This procedure is known as Structure from Motion.**
Keywords: Camera Intrinsic Matrix, Fundamental Matrix, Feature Matching, Epipolar Geometry, Triangulation, Perspective-n-Points(PnP), Bundle Adjustment,

I. Introduction

A general approach to the SfM problem is described below:

- Feature Matching and Outlier Rejection using RANSAC
- Estimating Fundamental Matrix
- Estimating Essential Matrix from Fundamental Matrix
- Estimate Camera Pose from Essential Matrix
- Check for Cheirality Condition using Triangulation
- Perspective-n-Point
- Bundle Adjustment

II. Methodology

Concerning the workflow of the classical approach described above, the following is a detailed description of each step.

A. Feature Matching, Fundamental Matrix, and RANSAC

We have a set of images and their corresponding feature matches in the form of text files. We will begin by extracting the relevant features for a pair of two images but first, we need tp understand the concept of the Fundamental Matrix. We have performed two feature matching and outlier elimination. One uses homography-based RANSAC and then refined with RANSAC from the fundamental matrix.

Feature Matches of Image 1 and 2 are shown in Figures.

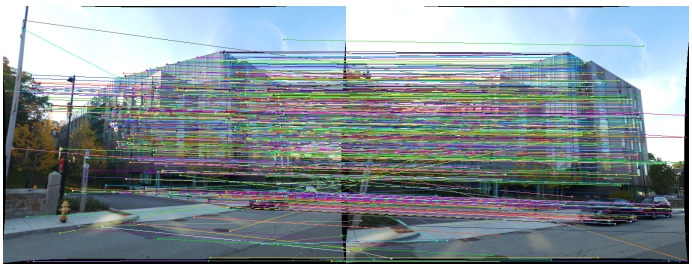

Fig. 1: Feature Matches Before

Fig. 2: Feature Matches after Homography RANSAC

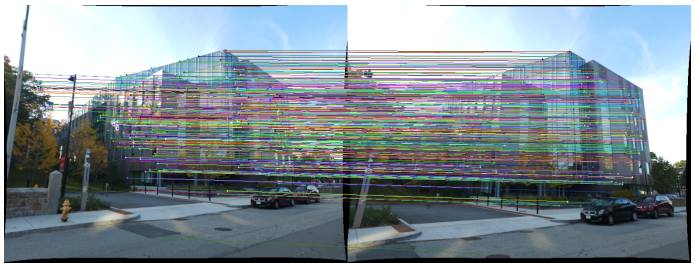

Fig. 3: Feature Matches After RANSAC using Fundamental Matrix

1) Estimating Fundamental Matrix: The primary step is to define **Fundametal 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 is only an algebraic representation of epipolar geometry(intrinsic projective geometry between two views). It can be algebraically defined as $\mathbf{x_i^{'T} F x_i} = 0$ where $i = 1, 2, ...m$. The fundamental matrix can be estimated by solving the linear least squares $Ax = 0$.

$$\begin{bmatrix} x_i' & y_i' & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{21} & f_{31} \\ f_{12} & f_{22} & f_{32} \\ f_{13} & f_{23} & f_{33} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = 0$$

$$x_i x_i' f_{11} + x_i y_i' f_{21} + x_i f_{31} + y_i x_i' f_{12} + y_i y_i' f_{22} + y_i f_{32} + x_i' f_{13} + y_i' f_{23} + f_{33} = 0$$

With $N \geq 8$ correspondences between two images, the fundamental matrix, F can be obtained as: By stacking the above equation in a matrix A, the equation $Ax = 0$ is obtained. This system of equations can be answered by solving

$$\begin{bmatrix} x_1x_1' & x_1y_1' & x_1 & y_1x_1' & y_1y_1' & y_1 & x_1' & y_1' & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_mx_m' & x_my_m' & x_m & y_mx_m' & y_my_m' & y_m & x_m' & y_m' & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{21} \\ f_{31} \\ f_{12} \\ f_{22} \\ f_{32} \\ f_{13} \\ f_{23} \\ f_{33} \end{bmatrix} = 0$$

the linear least squares using Singular Value Decomposition (**SVD**). When applying SVD to matrix A, the decomposition USV^T would be obtained with U and V orthonormal matrices and the last column of V is the true solution. However, due to noise in the correspondence, the estimated F matrix can be of rank 3. So, to enforce the rank 2 constraint, the last singular value of the estimated F must be set to zero.

Fundamental Matrix Results

Optimized Fundamental Matrix:

$$\begin{bmatrix} -9.490 \times 10^{-07} & -4.111 \times 10^{-06} & 0.009 \\ 0.00000708 & 0.00000411 & -0.034 \\ -0.010 & 0.032 & 1.000 \end{bmatrix}$$

CV2 Fundamental Matrix:

$$\begin{bmatrix} -9.935 \times 10^{-07} & -2.997 \times 10^{-06} & 0.009 \\ 5.992 \times 10^{-06} & 4.162 \times 10^{-06} & -0.034 \\ -0.010 & 0.032 & 1.000 \end{bmatrix}$$

2) Outlier Rejection using RANSAC: The features will have noisy correspondences that can be removed using the RANSAC algorithm. Further, once we complete outlier rejection using RANSAC, we will fix a set of inliers that can be used to calculate the Fundamental Matrix. Following is the algorithm that describes this section:

Algorithm 1 Get inliers RANSAC

```
r = 0
for i = 1 : M do  ▷ Choose 8 correspondences, (x̂1, x̂2)
    F = EstimateFundamentalMatrix(x̂1, x̂2)
    S = 0
    for j = 1 : N do
        if |x1^T F x2^T| < ε then
            S = S ∪ j
        end if
    end for
    if n < |S| then
        n = |S|
        S_in = S
    end if
end for
```

B. Estimate Essential Matrix from Fundamental Matrix

The relative camera poses between two images can be computed using the Essential Matrix E, which can be defined as $E = K^T F K$ where K is the camera calibration matrix. After extracting E from the above equation, we need to reconstruct it in the following way:

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

Epipolar Matrix:

$$\begin{bmatrix} -0.0176 & -0.0650 & 0.2396 \\ 0.1243 & 0.0594 & -0.9600 \\ -0.2628 & 0.9628 & 0.0383 \end{bmatrix}$$

C. Estimate Camera Pose from Essential Matrix

Since the E matrix is identified, the four camera pose configurations: $(C_1, R_1), (C_2, R_2)(C_3, R_3)and(C_4, R_4)$ where $C \epsilon R3$ is the camera center and $R \epsilon SO(3)$ is the rotation matrix, can be computed from E matrix. Let $E = UDV^T$ and $W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

The four configurations can be written as:

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 $R_4 = UW^TV^T$

Another condition to check while calculating the above configurations is if $det(R) = -1$, then the camera poses should be corrected as $C = -C$ AND $R = -R$.

Please find the plots for Epipolar lines in the Figure 4 and 5.

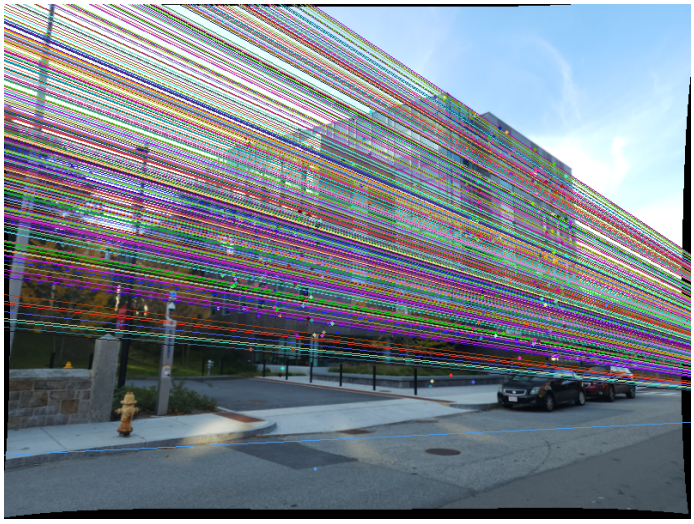

Fig. 4: Epipolar lines of Image 1

D. Triangulation Check for Cheirality Condition

Since we have the camera poses, we can now determine the 3D position of points in the scene. This is known as **Triangulation**. The set of 3D points obtained can be valid or invalid depending on their relevance in depth as seen from two different camera views. This can be accomplished by checking

Fig. 5: Epipolar lines of Image 2

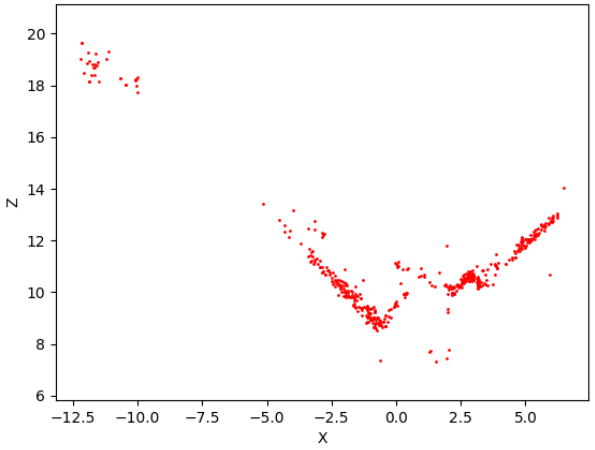

Fig. 7: Correct Pose Triangulation Output of Image 1,2

the cheirality condition i.e. the reconstructed points must be in front of the cameras. A 3D point X is in front of the camera if: $r_3(X - C) > 0$ where r_3 is the third row of the rotation matrix ($z-$axis of the camera). Essentially we want to know if the depth of the given point is positive. If the depth is obtained to be negative, then the point is behind the camera and can be neglected. The best camera configuration (C, R, X) is the one that produces the maximum number of points satisfying the cheirality condition.

After performing triangulation we get four poses for which the output is shown in Figure 6. Further after disambiguation of the pose, we can identify correct pose for which the triangulation output is shown in Figure 7.

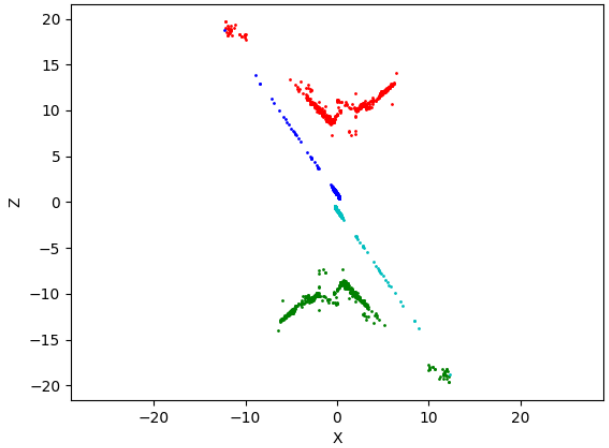

Fig. 6: Points Viewed from Four Poses - Only One pose is correct (z¿0)

1) Non-Linear Triangulation: Apart from minimizing the algebraic error (achieved by linear triangulation), we are also concerned with minimizing geometric error or in other words, the projection error. The projection error can be calculated as the difference between the measurement and the projected 3D point.

$$min_x \sum_{j=1,2} (u^j - \frac{P_1^{jT}\tilde{X}}{P_3^{jT}X})^2 + (v^j - \frac{P_2^{jT}\tilde{X}}{P_3^{jT}X})^2$$

j is the index of each camera.// $\tilde{X}$ is the homogeneous representation of X. P_i^T is each row of the camera projection matrix P. After performing triangulation we optimize the error and get the output as shown in Figure 8.

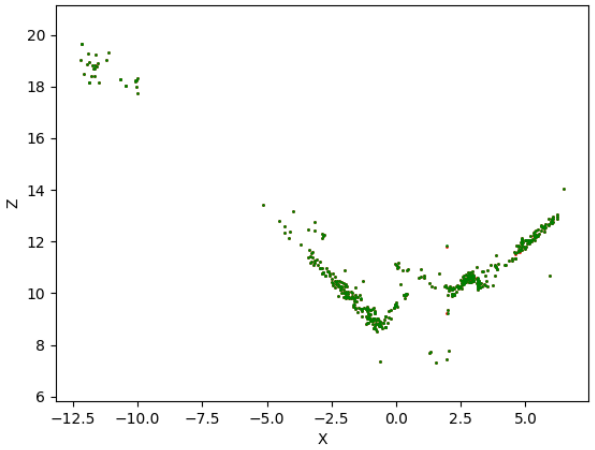

Fig. 8: Correct Pose Non-Linear Triangulation Output of Image 1,2

E. Perspective-n-Points

Now that we have a set of n 3D points in the world, their 2D projections in the image, and the intrinsic parameter; the 6 DOF camera pose can be estimated using linear least

squares. This fundamental problem, in general, is known as Perspective-n-Point (PnP).

1) Linear Camera Pose Estimation: To estimate the camera pose, given 2D-3D correspondences, we will enforce linear least squares. The 2D points can be normalized by using the intrinsic parameter to isolate camera parameters, (C, R), i.e. $K^{-1}x$.

2) PnP RANSAC: PnP is prone to error as there are outliers in the given set of point correspondences and hence we will use RANSAC for outliers rejection. The algorithm is defined below:

Algorithm 2 PnP RANSAC

$n = 0$
for $i = 1 : M$ **do** ▷ Choose 6 correspondences, $(\hat{X}, \hat{x})$ randomly.
  $[C, R] = LinearPnP(\hat{X}, \hat{x})$
  $S = 0$
  for $j = 1 : N$ **do**
    $e = (u^j - \frac{P_1^{jT}\tilde{X}}{P_3^{jT}X})^2 + (v^j - \frac{P_2^{jT}\tilde{X}}{P_3^{jT}X})^2$
    if $e < \epsilon_r$ **then**
      $S = S \cup j$
    end if
  end for
  if $n < |S|$ **then**
    $n = |S|$
    $S_{in} = S$
  end if
end for

3) Nonlinear PnP: We must refine the camera poses, (C, R) by minimizing the projection error. Linear PnP deals with minimizing the algebraic error. The geometric error which is calculated as a difference between the measurement and projected 3D point, is minimized using Non-Linear PnP.

$$min_{(C,R)} \sum_{i=1,J} (u^j - \frac{P_1^{jT}\tilde{X}}{P_3^{jT}\tilde{X}_j})^2 + (v^j - \frac{P_2^{jT}\tilde{X}}{P_3^{jT}X_j})^2$$

To enforce the orthogonality of the rotation matrix, the rotation matrix R can be represented using quaternion,$R = R(q)$. The above equation can be modified as:

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

After performing the pair level calculation and optimization using Non Linear PnP and Non Linear Triangulation, we got the below errors:

So the average reprojection error in pixels for the pipeline till non linear triangulation are:

1) Linear Triangulation : 18.254
2) Non Linear Triangulation : 17.1
3) Linear PnP : 90.85
4) Non Linear PnP: 78.35

Output Before Bundle Adjustment

TABLE I: Reprojection Error - In Pixels (W.r.t Image 1)

Method	Img2	Img3	Img4	Img5
Linear Triangulation	1.516	20.24	14.85	36.41
Non Linear Triangulation	1.45	19.59	14.01	33.35
Linear PnP (RANSAC)	-	81.88	71.72	118.95
Non Linear PnP	-	72.81	56.07	106.19

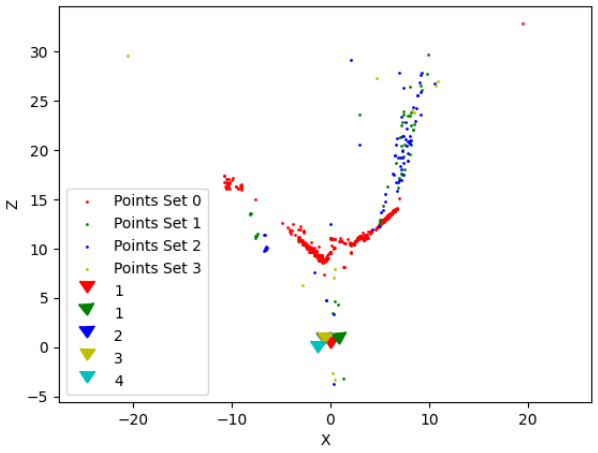

Fig. 9: Points in the World before bundle estimation

F. Bundle Adjustment

To refine the camera poses and 3D points obtained previously, we need to minimize the projection error and do a final reconstruction of the scene.

1) Visibility Matrix: To achieve a final reconstruction, we will first define a visibility matrix. $V(I \times J)$ is a binary matrix where V_{ij} is one if j^{th} point is visible form i^{th} camera and zero otherwise.

After Bundle adjustment there were some issues because of some outliers or optimization didnt happen properly. the output is shown in Figure.10

2) Bundle Adjustment: The bundle adjustment refines camera poses and 3D points simultaneously by minimizing the following reprojection error over $C^I_{i_{i=1}}, q^I_{i_{i=1}} and X^J_{j_{j=1}}$.

III. Putting the pipeline Together

The complete pipeline we have followed is shown in the SfM-Overview algorithm.

IV. Comparison - Our Pipeline and Visual SFM

Our Pipeline - Figure .
Visual SFM Software (Only Point based Output)

V. Discussion

Even though the current pipeline produces decent results, there is more work needs to be done.

Some of the major issues that we faced initially and might be the reason for randomness in accuracy sometimes:

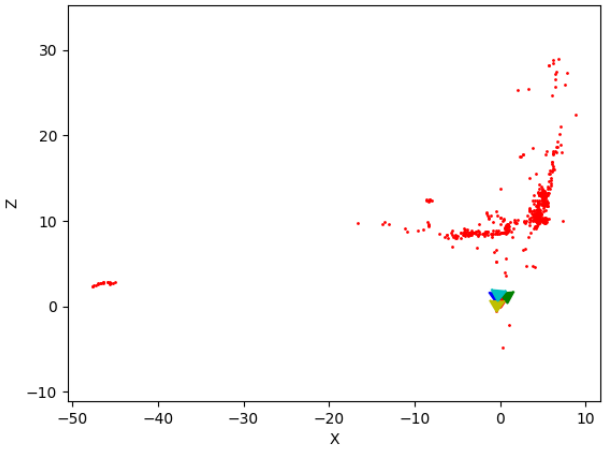

Fig. 10: Points in the World after Bundle Estimation

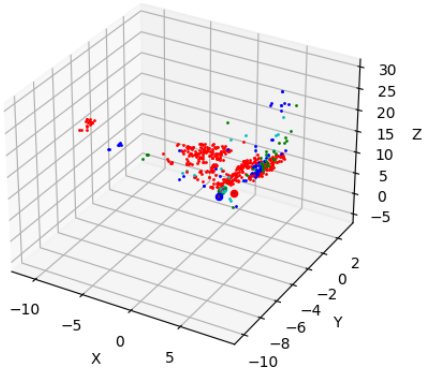

Fig. 11: Pipeline Output in 3D

Algorithm 3 SfM - Overview

for all possible pair of images **do** ▷ Reject outlier correspondences.
  $[x1, x2] = GetInliersRANSAC(x1, x2)$
end for

$F = EstimateFUndamentalMatrix(x1, x2)$

$E = EssentialMatrixFromFundamentalMatrix(F, K)$

$[C_{set}, R_{set}] = ExtractCameraPose(E)$ ▷ Perform Linear Triangualtion

for $j = 1 : 4$ **do**
  $Xseti = LinearTriangulation(K, zeros(3, 1), eye(3), Cseti, Rseti, x1, x2)$
end for ▷ Check cheirality condition.

$[C, R] = DisambiguateCameraPose(Cset, Rset, Xset)$ ▷ Perform Non-Linear Triangualtion

$X = NonLinearTriangulation(K, zeros(3, 1), eye(3), C, R, x1, x2, X0)$

$Cset = C, Rset = R$ ▷ Register camera and add 3D points for the rest of the image.

for i=3:I **do** ▷ Register the i^{th} image using PnP.
  $[Cnew, Rnew] = PnPRANSAC(X, x, K)$
  $[Cnew, Rnew] = NonlinearPnP(X, x, K, Cnew, Rnew)$
  $Cset = Cset \cup Cnew, Rset = Rset \cup Rnew$ ▷ Add new 3D points.
  $Xnew = LinearTriangulation(K, C0, R0, Cnew, Rnew, x1, x2)$
  $Xnew = NonLinearTriangulation(K, C0, R0, Cnew, Rnew, x1, x2, X0)$
  $X = X \cup Xnew$ ▷ Build Visibility Matrix
  $V = BuildVisibilityMatrix(traj)$ ▷ Perfrom Bundle Adjustment.
  $[Cset, Rset, X] = BundleAdjustment(Cset, Rset, X, K, traj, V)$
end for

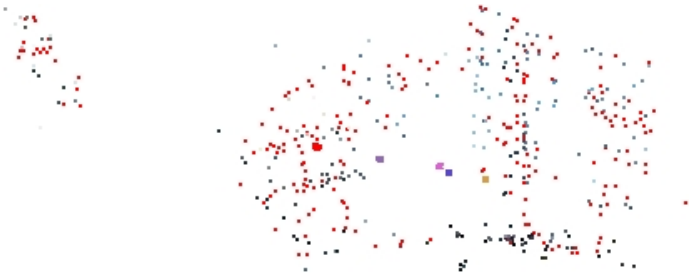

Fig. 12: Output using VSM - Sparse Reconstruction

- Even though SIFT is a good scalable feature matcher, it gave some bad matches at least between images 1-4, and 1-5. We tried to remove some of the outliers by strict thresholding using the methods (Homography RANSAC and Fundamental matrix RANSAC). This helped but because of this, the points may be concentrated on one region which might affect the overall Essential matrix calculation.
- Some other issues include Non linear optimization not working every time like sometimes the optimization is stuck on local minima, these issues can further be investigated and resolved using different parameters of the optimizer.
- The Bundle adjustment implementation we did is having some issues related to non accounting of Visibility matrix as we handled the visibility a bit differently.
- Understanding and handling the data properly is also a major concern as this affects both the stability of the algorithm and also the performance.

Limitations of the Implementations:

- Currently our implementation lacks a solid visibility matrix concept as we did some sort of hash map based handling but it is not very proper.
- Our implementation Non Linear approaches can further be tuned with different parameters of the optimizers.

- Bundle Adjustment needs to be improved further as currently we are using scipy.optimize but this can be upgraded with Solvers such as pySBA or cares solvers.
- Our implementation does not take into account of the other features from other image correspondences. This is important, event though our implementation can be modified it requires a bit effort. These addition of correspondences ensure all the depth points taken into account and also helps in constructing a proper visibility matrix.

Future Directions

- Adding the Multi Image correspondences and optimizing the handling of the data.
- Working on Deep learning based feature extractors such as SuperGlue, LoFTR etc. These can help for better correspondences in turn reduces the outliers.
- Improving the Speed of Bundle Adjustment.

One of the sample deep learning output from Superglue is shown in Fig.13. We are currently trying to combine this into our pipeline. If you see the Figure those matches seem more robust without any need for refinement using RANSAC, especially some of the models like superglue are trained for scenarios like this (Buildings, Outdoor environment etc.)

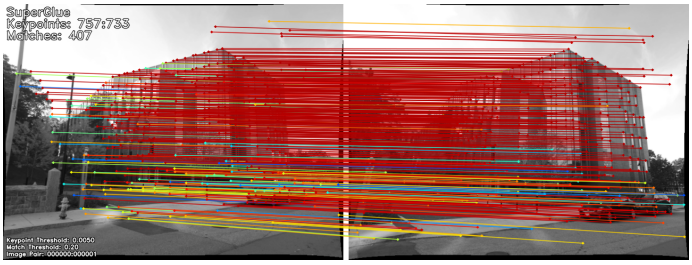

Fig. 13: SuperGlue Output - Matches

REFERENCES

[1] Z. Zhang, "A flexible new technique for camera calibration," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, pp. 1330-1334, Nov. 2000, doi: 10.1109/34.888718.

[2] https://github.com/naitri/SFM/ (One such Implementation from Uni. of Maryland)