

Buildings built in minutes - An SfM Approach

John Kanu

PhD Student, Computer Science, UMD
Email: jdkanu@cs.umd.edu

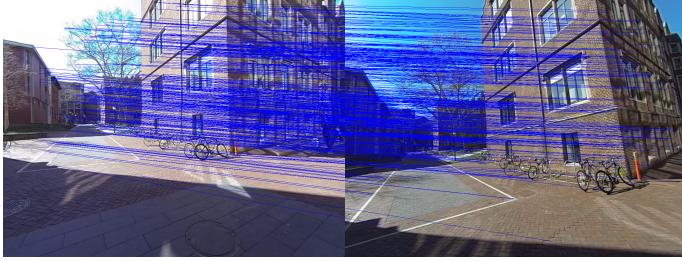
Srinidhi Sreenath

Masters Student, Robotics, UMD
Email: ssreenat@terpmail.umd.edu

Abstract—In this paper we present a structure from motion pipeline which reconstructs a 3-dimensional scene based on stereo images.

I. FEATURE MATCHING AND RANSAC

The SIFT key points are given already between the sets of images. All the matchings are visualized between image 1 and image 4 shown below:



Many of the matches are not good and they need to be refined and outliers need to be rejected. To do this, RANSAC is performed. In RANSAC, the fundamental matrix is estimated. Given 2 set of corresponding points between 2 images, the fundamental matrix is defined by $x_i'F x_i$ where x_i' is the feature point in second image and x_i is the feature point in first image. F matrix is estimated using 8 point algorithm and a minimum of 8 corresponding points is required for estimating the Fundamental matrix.

The equation system is represented as $Af = 0$ and F is solved using linear least squares method by taking SVD of A i.e $USV^T = SVD(A)$ and then taking the eigen vector associated with the least eigen value of V matrix.

Once the fundamental matrix is obtained, the rank is forced to be 2 by taking the SVD of F i.e $USV^T = SVD(F)$ and forcing the las singular value of S to be 0 i.e $S[2, 2] = 0$. F is then recomputed as $F = U * S_{mod} * V^T$ where S_{mod} is the modified S . Now, F is further divided by its norm to get to unit scale.

$$F = \begin{bmatrix} 1.8363e-06 & -2.5172e-05 & 0.0095 \\ 2.7734e-05 & -8.0892e-07 & -0.0025 \\ -0.0140 & 0.0010 & 1.0 \end{bmatrix}$$

A. RANSAC

To reject outliers, RANSAC procedure is used where 8 points are randomly sampled from corresponding feature points and F is calculated. Once F is calculated, then for all the corresponding feature points, the value of $x_i'F x_i$ is calculated. If the value is less than the threshold value of **0.1**, the corresponding feature points are considered as inliers else rejected as outliers. Different threshold values of 0.05, 0.07 and 0.5 were tested and 0.1 produced the list with most number of inliers.

For each iteration of RANSAC, a set of inliers are selected. RANSAC is run for **500** iterations and the final set of inliers are selected as the largest number of inliers over all the iterations.

The result of correspondences after RANSAC for images 1 and 4 is shown below



II. ESSENTIAL MATRIX FROM FUNDAMENTAL MATRIX

Once the fundamental matrix is obtained, we get the essential matrix by following set of equations: $E = K^T F K$ where K is the camera matrix. Due to noise in K , the singular values of E are set to be [1, 1, 0]. This is done by taking SVD of E and recalculating E with S replaced with the new singular values.

$$E = \begin{bmatrix} 0.0546 & -0.6731 & -0.0563 \\ 0.7251 & -0.0328 & 0.6871 \\ 0.0259 & -0.7297 & -0.0928 \end{bmatrix}$$

III. TRIANGULATION FOR CORRECT CAMERA POSE

Once E is obtained, 4 sets of camera poses are extracted as described in the project specs. Then triangulation check

is done to disambiguate for the correct pose, and it is implemented as described in the pseudocode, using linear least squares.

A. Reprojecting the points

Once the correct pose i.e R and C are obtained, the linear triangulation technique is used to obtain the 3D points. Once the 3D points are obtained, they are visualised. To calculate the reprojection error, the 3D points are reprojected to each image plane. This is done with the help of projection matrix i.e $P = KR[I - C]$. The projected coordinates are obtained using $x = PX$ where X is the 3D coordinates. Once the projected coordinates are obtained, then the reprojection error is calculated as the difference between the original feature position and the reprojected point.

For image 1 and 4, the reprojection error obtained was about **30.89** pixels. The reprojected points are shown in comparison with the original points, in Figures 1 and 2. The blue markers represent the original feature detected, the red markers represent the reprojected 3D points onto the image and a green line is drawn between the corresponding feature point and the reprojected point. The reprojection error for correspondences between image 2 and image 3 was much lower i.e **11.96** pixels. The reprojected points for image 2 and image 4 are shown in Figures 3 and 4.

Linear triangulation provides a reasonable initialization before refinement with nonlinear optimization. Nonlinear triangulation is performed with the Levenberg-Marquardt algorithm. Results for linear and nonlinear triangulation are shown in Figures 5 and 6. In general, we want to minimize outliers and reprojection error. We can observe that, while the reprojection error is within a typical range, there are many outliers along the left side of the reconstruction. A visual comparison can show that nonlinear triangulation yields a slightly lower reprojection error.

IV. PERSPECTIVE-N-POINTS

We performed linear and nonlinear Perspective-n-Points as described in the pseudocode. RANSAC is again performed to obtain correct pose, running for 500 iterations. The reprojection error threshold was set at **10**. Different thresholds were used ranging from 20 to 1 and 10 was selected since a very low threshold selects very less points and high threshold selects all points.

Refinement of the results of linear PnP is performed using nonlinear PnP, which we have implemented with Levenberg-Marquardt solver for optimization. The reconstruction for all images is presented in Figure 7. We can observe again that the number of outliers after performing Perspective-n-Points for all images has increased, especially on the left side. This increase in the number of outliers is expected, as we have combined the reconstruction over several images, which results in a larger spread of points.

V. BUNDLE ADJUSTMENT

We wrote code for bundle adjustment using `scipy.optimize.leastsq`. However, the optimization procedure was very slow. Results are not included in the report.

VI. REPROJECTION ERRORS AFTER EACH STEP

The error table I summarizes the best errors obtained over a set of iterations for each image. The triangulation errors for image 1 and 4 and w.r.t each other and for images 2 and 3, it's w.r.t image 1 and for images 5 and 6 it's w.r.t image 4.

Image	Linear Tri	Nonlinear Tri	PnP RANSAC	Nonlinear PnP
1	7.817	6.3641	-	-
2	19.738	19.735	182.204	2.846
3	7.968	7.9410	261.926	6.706
4	9.604	6.0444	-	-
5	15.699	15.201	218.961	71.39
6	85.165	52.917	431.724	262.05

TABLE I: Error table



Fig. 1: Reprojected points for image 1

VII. VISUALSFM OUTPUT

The images were ported into the VSfM software and the 3D reconstruction was done. The images 8, 9 and 10 show the output of the VSfM software.

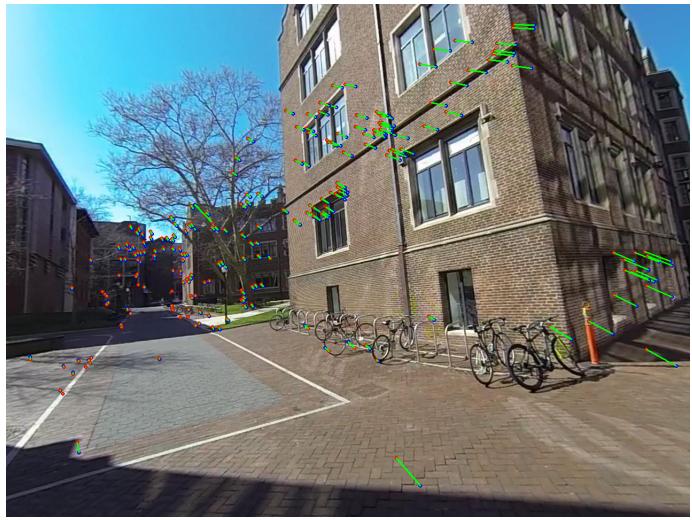


Fig. 2: Reprojected points for image 4



Fig. 4: Reprojected points for image 3

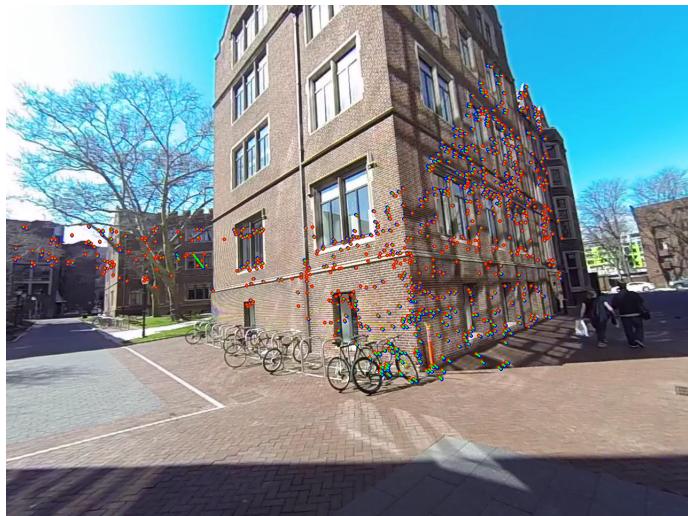


Fig. 3: Reprojected points for image 2

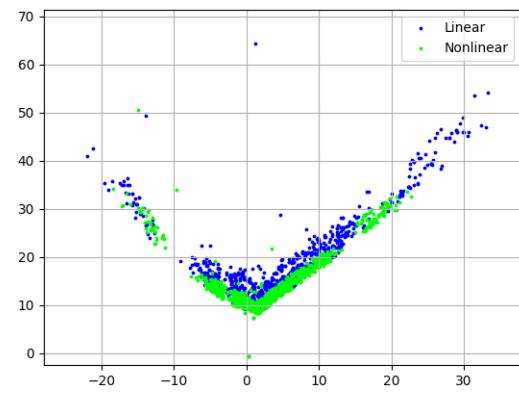


Fig. 5: Top view of reconstructed scene in Images 2 and 3

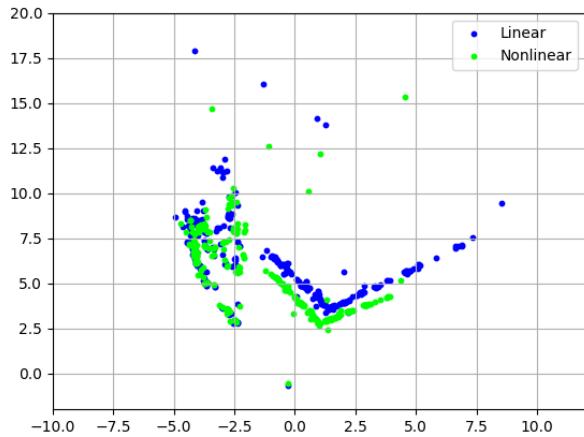


Fig. 6: Top view reconstruction for images 1 and 4 after linear and nonlinear triangulation

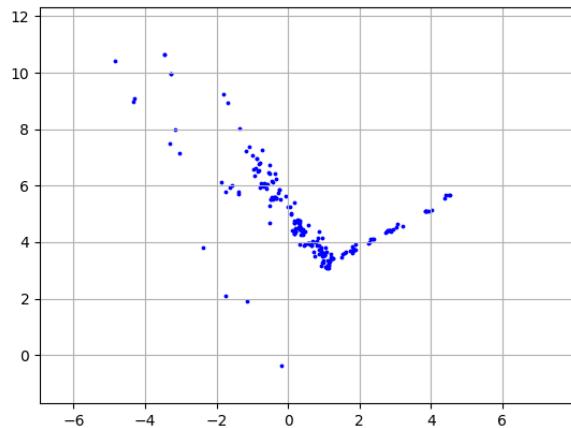


Fig. 7: Final top view reconstruction, without bundle adjustment

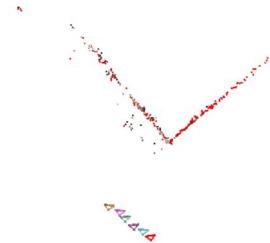


Fig. 8: VisualSFM: Top view of reconstructed scene

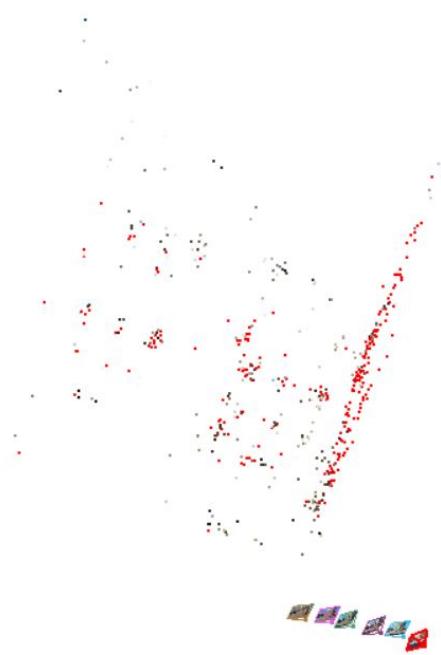


Fig. 9: VisualSFM: Oblique view of the reconstructed part

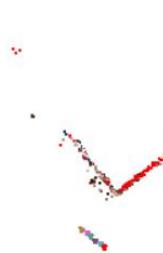


Fig. 10: VisualSFM: Outliers present in the reconstruction