

CMSC733-Project 3: Structure From Motion

Rohan Chandra

Swati Singhal

Abstract—Structure from Motion from multiple views using bundle adjustment. We used a different solver algorithm for non-linear triangulation. We also include a pruning algorithm after non linear triangulation that helps in reducing re-projection error.

I. VISUALSfM OUTPUT

Following is the attached result from visual sfm.



Fig. 1. Visual SfM output

As you can see, the results from the software are not perfect. In this report we demonstrate that our algorithm outperforms the results produced from this software.

II. MULTI VIEW RECONSTRUCTION

We perform the following algorithm:

A. RANSAC on matching pairs

Raw Feature Matching:



Fig. 2. Feature matching before RANSAC

After RANSAC:

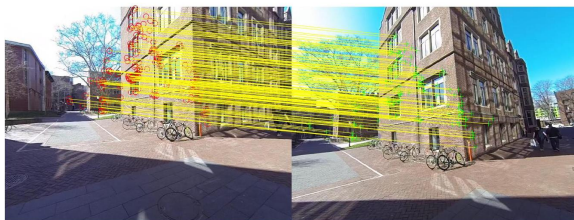


Fig. 3. Feature matching after RANSAC

B. Fundamental Matrix Computation

F =

$$\begin{bmatrix} -1.8459e^{-07} & -7.8618e^{-6} & .0024 \\ 8.9358e^{-06} & 1.2166e^{-07} & -0.0056 \\ -0.0035 & 0.0039 & 1 \end{bmatrix}$$

C. Essential Matrix Computation

E =

$$\begin{bmatrix} -0.0119 & -0.9265 & -0.2957 \\ 0.9906 & 0.0062 & 0.0425 \\ 0.1305 & -0.2226 & -0.0656 \end{bmatrix}$$

D. Re-projection Images After Non - Linear Triangulation and PnP

This was the first roadblock that we encountered. For a long time we were getting a huge re-projection error. This was solved by fixing two things:

Firstly, we observed that the suggested fundamental matrix constraint threshold of ϵ was too low and needed a larger value. Remember, the constraint $x^T F x' = 0$, ensures that the points x and x' lies on the epipolar plane and project the same 3D point. Besides this and other corrections suggested in the pdf, we further observed that normalizing the fundamental matrix was also required.

The second important fix that we made was efficient data processing.

By carefully analyzing the re-projection results, we observed that while most re-projection were perfect, few outlier/anomalous values resulted in very high re-projection error. To eliminate this, we isolated the computation of each point and filter out the results that failed to converge to a solution for non-linear least square problem. This significantly resulted in achieving correct the re-projections (error percentage is approx 5% or 0.05 pixels as shown in the figures below).

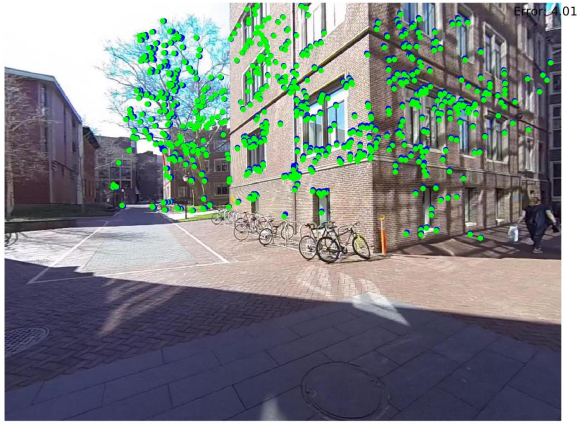


Fig. 4. 3D re-projection to 2D points. Blue indicate original points. Green indicate projected points. Error is displayed at the top

Another interesting observation was that the choice of the underneath solver for 'lsqnonlin' function can make a significant difference in the quality of results. With default solver, the solution set converged for very less points and our pruning algorithm rejects those points. Using 'lsSolver' solver, available in the TOMLAB package, we got around 700+ converged points as oppose to 20 points by default solver.

E. 3D visualization of Linear and Non Linear Triangulated Points

Figure 5 shows the initial results of our algorithm.

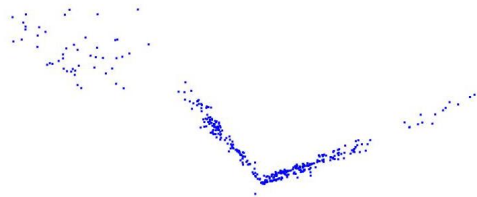


Fig. 5. Sparse Top View Reconstruction for 2 views

The two view reconstruction is sparse. Following this stage, we register the rest of the images in succession using pose estimation and apply bundle adjustment to get the following reconstruction.



Fig. 6. Dense Side View Reconstruction after Bundle Adjustment

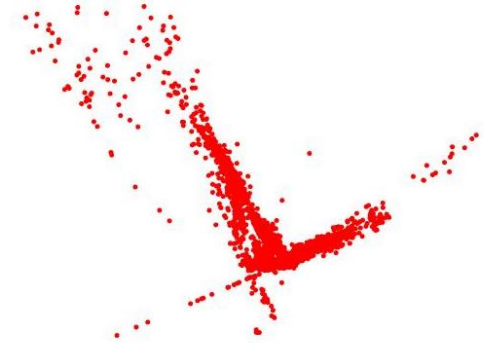


Fig. 7. Dense Top View Reconstruction after Bundle Adjustment

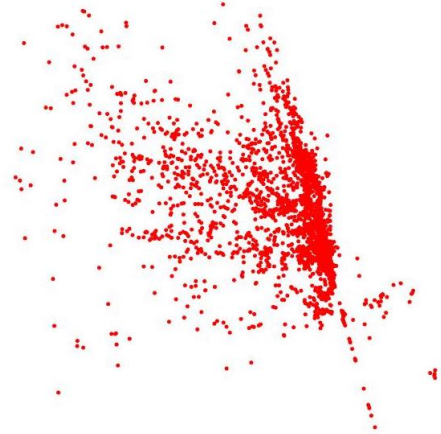


Fig. 8. Dense Oblique View Reconstruction after Bundle Adjustment

The dense point cloud is a serious improvement over the sparse one. Due to use of same color, the proper shape

might not be immediately visible, e.g., the shape in side view might appear more towards V-shape instead of U-shape. Although we achieved near perfect results, we believe some pruning after bundle adjustment could have further helped in improving the quality of the results.

It is worth mentioning that the bundle adjustment package was not easy to install and use. It took several days of painstaking software engineering to get the package to work. Due to compatibility issues with Mac computers, we installed SBA (the bundle adjustment package) on a Linux machine.

III. EXTRA CREDIT

We used iPhone 4S to capture the following images for extra credit:



Fig. 9. Images for Extra Credit

Selection of relevant and proper images is extremely important for performing reconstruction. Relevance would ensure we recover a structure that can be recognized easily from the point cloud and proper entails using images with 'good' feature points. It took us several tries to gather the images that gave us sufficient feature matches.

After securing the images that we wanted, we used the vl_feat library to perform feature detection and matching. We found that the vl_feat library considerably outperforms the SURF points from MATLAB. We procured over 2000 matching points from the former while obtaining only 400 from the latter.

The results of RANSAC can be shown below:



Fig. 10. RANSAC after feature matching using vl_feat SIFT points

Apart from feature matching, we were required to calibrate the camera before taking images. In our project, we were given both the point matches and the camera matrix, K . In order to calibrate the camera and calculate the intrinsic matrix, we followed the steps in homework 1 of CMSC 733 where we use a checkerboard of square size 21.5 mm. Following calibration, correction was made for lens distortion.

After the required processing steps of calibration and feature matching, we apply our algorithm for calculating triangulated and refined 3D points. The re-projection is shown in fig. 11. The blue points (under the green) indicate the original points and the green points indicate the re-projected points. The fact that you see so very little blue is an indicator that our non-linear triangulation worked perfectly.

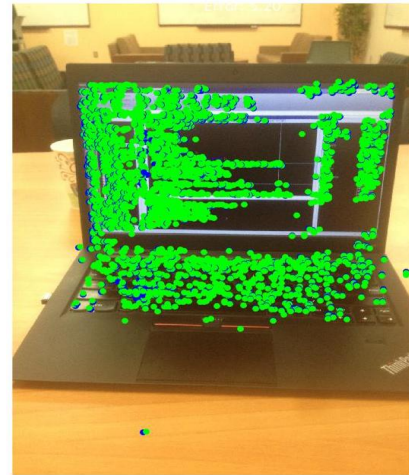


Fig. 11. 3D reprojection of 2D points. Blue indicate original points. Green indicate projected points.

For the purposes of demonstrating our algorithm we show the front view of the reconstructed laptop. The front view aligns itself in the X-Y plane. Notice the right side in fig. 12. The two red columns and the 3 little red clouds of points on top correspond to their green counterparts in fig. 11. The lines of code on the laptop screen are also quite visible in our experiment.

Similarly, in fig. 13, the 'L' shape of a laptop looked at sideways is a perfect side-view reconstruction. The two straight lines going outward from each side of the laptop is the table surface. In fig. 14, we present the top view with similar comments about the two straight lines observed in the picture.

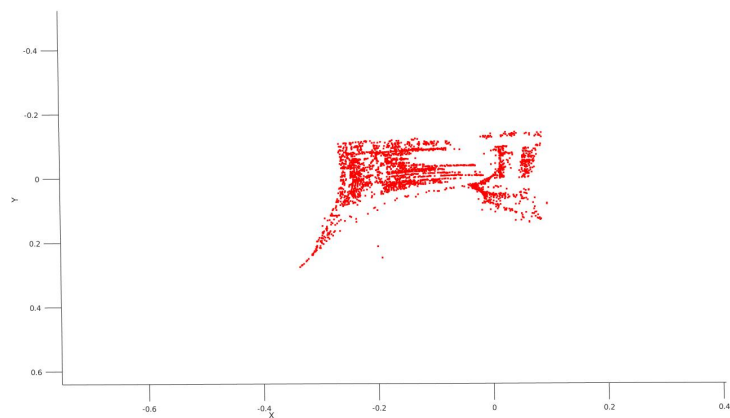


Fig. 12. Front View of an open laptop. Notice the familiarity specially on the right side of the reconstructed image

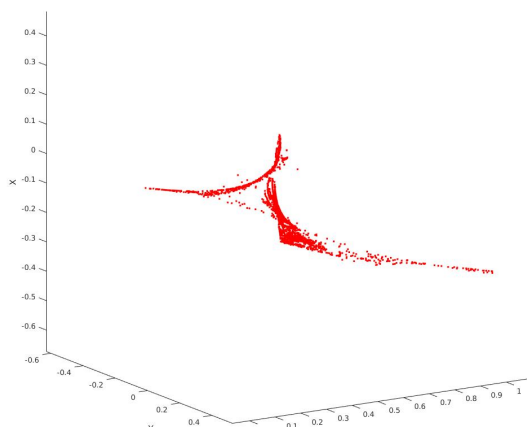


Fig. 13. Side View of an Open laptop. Notice the 'L' shape. The straight lines are the reconstruction of the table.

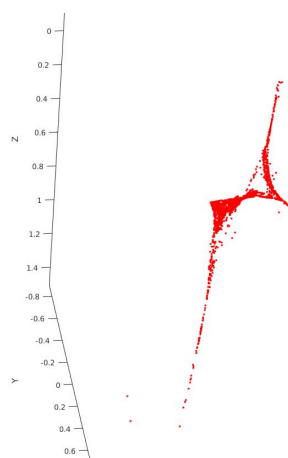


Fig. 14. Top View of an Open laptop. The straight lines going upward and downward from the laptop are the reconstruction of the table