Project 3: Buildings built in minutes: Structure from Motion

Wenyan Li
Email: wenyanli@umd.edu

Xiaoxu Meng
Email: xiaoxumeng1993@gmail.com

***Abstract*—The geometrical theory of structure from motion allows projection matrices and 3D points to be computed simultaneously using only corresponding points in each view. In this project, given six undistorted images of the building and the intrinsic parameters of the camera, a full pipeline of structure from motion is implemented. And the obtained result is compared with the output using VSfM.**

I. Introduction

The aim of this project is to reconstruct a 3D scene and simultaneously obtain the camera poses of a monocular camera with respect to this scene and this procedure is called Structure from Motion (SfM). Our task is to implement the full pipeline of structure from motion including two view reconstruction, triangulation, PnP, and bundle adjustment.

II. Data Preprocessing

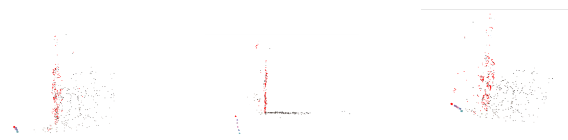
Given 6 undistorted images of the building and 5 matching files, matched feature points in each pair of images can be extracted for further calculation. Here, we extracted the matched features in each pair of the images as shown in Table.I. Moreover, we also extract common features detected in $(I_1I_2I_3)$,$(I_1I_2I_4)$,$(I_2I_3I_4)$,$(I_3I_4I_5)$,$(I_3I_5I_6)$ for calculation convenience in new camera registration.

File No.	Matched feature pairs	# of pairs
matching1	$I_1 \leftrightarrow I_2$, $I_1 \leftrightarrow I_3$, $I_1 \leftrightarrow I_4$	3
matching2	$I_2 \leftrightarrow I_3$,$I_2 \leftrightarrow I_4$	2
matching3	$I_3 \leftrightarrow I_4$,$I_3 \leftrightarrow I_5$,$I_3 \leftrightarrow I_6$	3
matching4	$I_4 \leftrightarrow I_5$,$I_4 \leftrightarrow I_5$	2
matching5	$I_5 \leftrightarrow I_6$	1

TABLE I: Matched features in images

III. Pairwise Image Feature matching

With pairwise image feature matching, we can estimate the fundamental matrix F by applying eight point algorithm, i.e. we have eight unknowns and 8 required equations that satisfy the epipolar constriants ($x_2^T F x_1 = 0$). The fundamental matrix can be estimated by solving ($Ax = 0$). Here we calculate the fundamental matrix $F$by solving linear homogeneous equation via SVD and applying rank constriant (rank(F) = 2),.



(a) Side view (b) Top view (c) Oblique view

Fig. 1: 3D reconstruction result using vsfm software

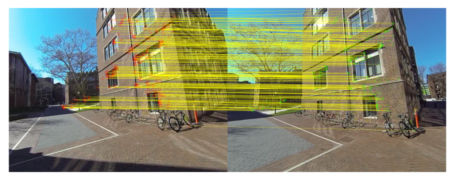


Fig. 2: Example of inlier correspondences using RANSAC

IV. Outlier Rejection via RANSAC

In this part, by simply follow the steps in RANSAC algorithm, i.e. Random sampling→Model building→Thresholding→Obtainnign inliers, we can estimate inlier correspondences using fundamental matrix.

V. Camera Pose Estimation

A. Essential Matrix Computation

With camera intrinsic parameter K provided in the calibration.txt file and using the estimated fundamental matrix, we can calculate $E = K^T F K$ as the epipolar constraint ($x_2^T F x_1 = 0$) needs to be satisfied,i.e. we have $x_2^T K^{-T} E K^{-1} x_1 = 0$. By reconstructing the essential matrix with $(1, 1, 0)$ singular values, we can obtain the corrected essential matrix.

B. Camera Pose Extraction

By applying relative transform from essential matrix, we can extract the camera poses, i.e. we can obtain the projection matrix P_2 where $P_2 = K[R|T]$. Again, applying SVD and with $W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$,we can obtain that

$$C_1 = U(:,3), R_1 = UWV^T \tag{1}$$

$$C_2 = -U(:,3), R_2 = UWV^T \tag{2}$$

$$C_3 = -U(:,3), R_3 = UW^TV^T; \quad (3)$$

$$C_4 = -U(:,3), R_4 = UW^TV^T \quad (4)$$

where $E = UDV^T$. Then, correct configuration can be resolved via point triangulation.

VI. Point Triangulation

With matched pointsx_1,x_2,P_1 and P_2, we want to compute 3D point X. However the problem is that in the presence of noise, back projected rays do not intersect, i.e. the matched feature points do not lie on corresponding epiploar lines. In order to solve the problem, we can make linear triangulation and nonlinear triangulation, where the former gives algebraic solution and the latter makes an optimization on error.

A. Linear triangulation

We use the equations $x_1 = P_1X$ and $x_2 = P_2X$ to solve for X. With the row vectors of P and by applying cross product, we can acquire that

$$\begin{bmatrix} x_1p_1^{3T} - p^{1T} \\ y_1p_1^{3T} - p^{2T} \end{bmatrix} X = 0 \quad (5)$$

and similarly we also have

$$\begin{bmatrix} x_2p_2^{3T} - p_2^{1T} \\ y_2p_2^{3T} - p_2^{2T} \end{bmatrix} X = 0 \quad (6)$$

which gives us $AX = 0$ where A is a $4*4$ matrix and we can solve for X using SVD again. Since the reconstructed point must be in front of the cameras, i.e. the image depth must be positive, the two equations $X_3 > 0$ and $r_3(X - C) > 0$ must hold for each X. This step is implemented in the function DisambiguateCameraPose() which gives us the corrected camera configuration.

B. Nonlinear Triangulation

In nonlinear triangulation, we implement the minimization of the statistical reprojection error given by

$$\underset{x}{\text{minimize}} \sum_{j=1,2} (u^j - \frac{P_1^{jT}X_{homo}}{P_3^{jT}X_{homo}})^2 + (v^j - \frac{P_2^{jT}X_{homo}}{P_3^{jT}X_{homo}})^2 \quad (7)$$

where j is the index of each camera, X_{homo} is the homogeneous representation of X, and P_i^T is the row vector of P. lsqnonlin is used in matlab for implementing the minimization.

C. PnP RANSAC

Given N $\geq$ 6 3D-2D correspondences, X and x, implement the following function that estimate camera pose (C;R) via RANSAC.

D. Nonlinear PnP

Given N $\geq$ 6 3D-2D correspondences, X and x, and linearly estimated camera pose, (C;R), refine the camera pose that minimizes reprojection error. Use the lsqnonlin function and the use the result of PnP RANSAC as the initials.

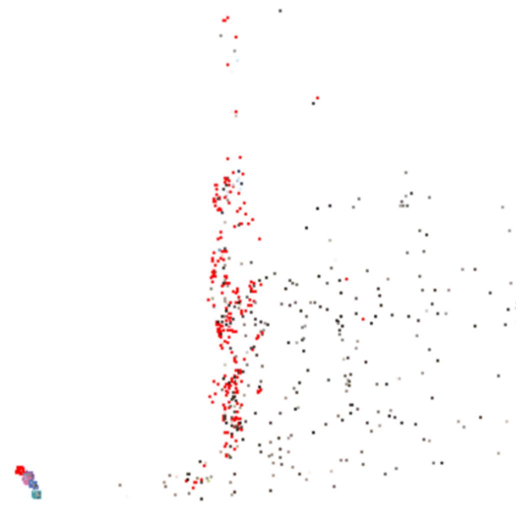


Fig. 4: Side view of Vsfm.

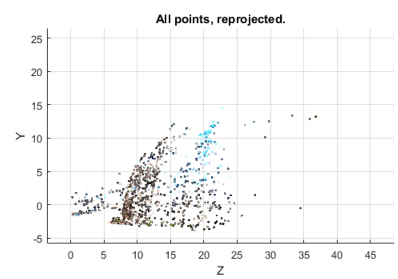


Fig. 5: Side view of our program.

E. Bundle Adjustment

We should do bundle adjustment by using SBA. I tried but I got dimension error. I also tried to use $lsqnonlin()$ and got memory outage. So my bundle adjustment function didn't succeed.

VII. Result Analysis

Compare with the reconstruction result using Vsfm software as shown below, the results are similar. A comparison between our results and sample results is shown in Fig. 3. Side view: result of Vsfm is shown in Fig. 4, result of our program is shown in Fig. 5.

Top view: result of Vsfm is shown in Fig. 6, result of our program is shown in Fig. 7.

Oblique view: result of Vsfm is shown in Fig. 8, result of our program is shown in Fig. 9.

References

[1] Tutorial of UCI: A tutorial of stereo reconstruction.

[2] Totorial from UPENN: A tutorial of structure from motion Pipeline.

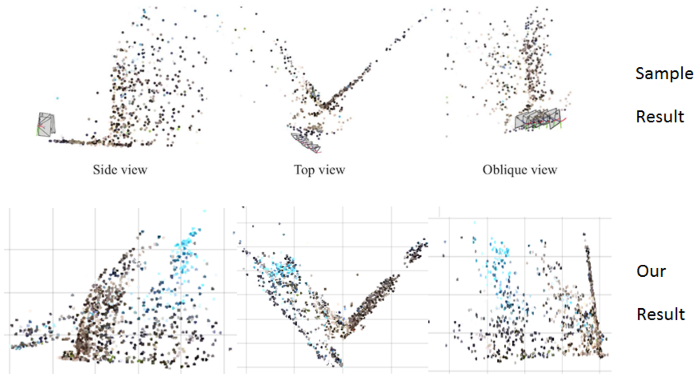


Fig. 3: A comparison between our results and sample results.

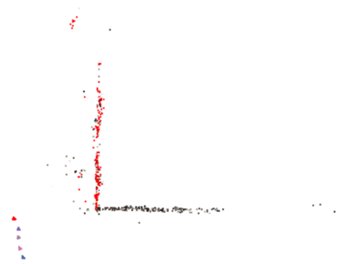


Fig. 6: Top view of Vsfm.

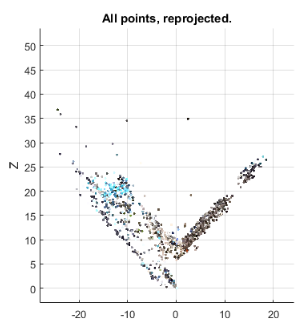


Fig. 7: Top view of our program.

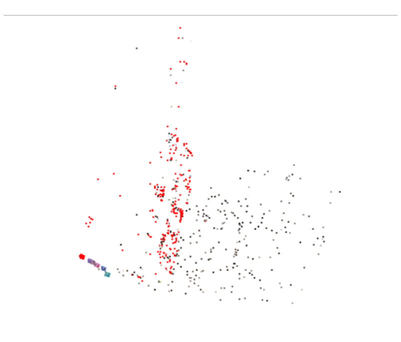


Fig. 8: Oblique view of Vsfm.

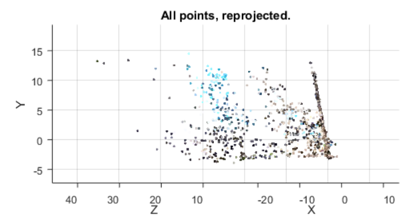
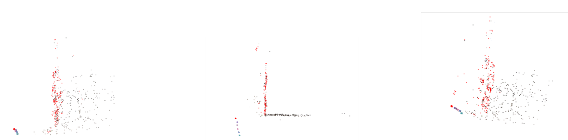


Fig. 9: Oblique view of our program.



(a) Side view (b) Top view (c) Oblique view

Fig. 10: 3D reconstruction result using vsfm software

[3] Tutorial of structure pipeline: A tutorial of structure from motion Pipeline.