3D Reconstruction from Two Views

Huy Bui
UIUC
huybui1@illinois.edu

Yiyi Huang
UIUC
huang85@illinois.edu

Abstract

In this project, we study a method to reconstruct a 3D scene from two views. First, we extract and match SIFT keypoints from two images, based on which we compute the fundamental matrix that relates the two images. Then we derive the camera matrices and reconstruct the 3D scene. To achieve better visual effects, we interpolate between the sparse 3D points. In order to do that, we first pick a point in one 2D image and find out the according matching point in another 2D image by searching along the epipolar line. Then the interpolation can be applied in each of the object's planes. Our results show a good 3D reconstruction from several viewpoints.

1 Introduction

Motivated by recent applications in 3D reconstruction from multiple views such as Photosynth of Microsoft (http://photosynth.net/) and the project *Building Rome in a day* from University of Washington, we decided to study about 3D reconstruction from two views, which we think is the basic thing that we need to master to realize larger applications. The core concept in our project is the epipolar geometry between two views of a scene. In our project, we implemented some basic algorithms to reconstruct a 3D scene given two different images of the scene, in order to have some insight in to the process. We first match keypoints between two views, then we compute the fundamental matrix and get the camera matrices and then reconstruct the 3D points. In order to obtain better results, we also try to interpolate the texture in each plane of the scene.

In this report, we first discuss the geometry and the basic algorithms in section 2. Results are given in section 3 with a brief summary in section 4.

2 Geometry and algorithms

In this section, we first introduce the two view geometry in 2.1, and then discuss the detailed steps on reconstructing points in $3D$ (section 2.2-2.5). Our discussion is based upon lectures in classes [3] and Richard Hartley's book [2].

2.1 Two-view geometry

Assume we have taken two images upon a same object from two different perspectives. Its epipolar geometry is showed in Figure 1.

In the figure, C and C' are the centers of two cameras, with P and P' as their projection matrices respectively. X is a point in 3D space. x and x' are the projections of X in the two image planes respectively. The line connecting C and C' is called the baseline, and it intersects the two image planes at points e and e' respectively, called epipoles. The plane π containing the baseline is called the epipolar plane, which intersects the two image planes at two epipolar lines l and l', respectively.

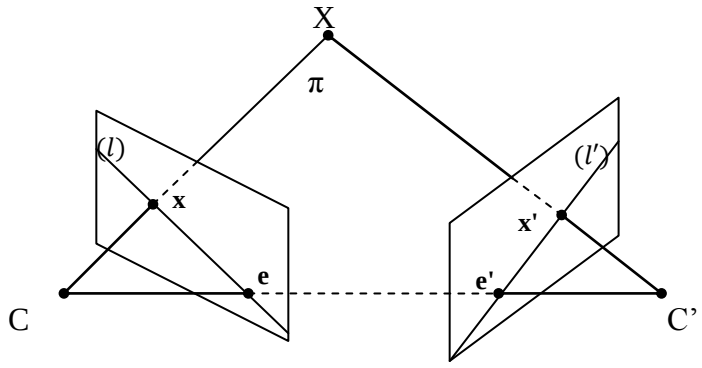

Figure 1. Epipolar Geometry between Two Views

Let F be the fundamental matrix. Some important equations are listed here:

$$\text{a) } x^{T}Fx = 0, \qquad (1)$$

$$\text{b) } l' = Fx, l = F^{T}x', \qquad (2)$$

$$\text{c) } Fe = 0, F^{T}e' = 0. \qquad (3)$$

In the case of calibrated cameras, F becomes the essential matrix E, which relates the normalized points in two images.

In our project, we try to reconstruct 3D scene from images taken by uncalibrated cameras. In order to reconstruct X in $3D$, we need to compute the fundamental matrix F (section 2.2) first, and then obtain camera projection matrices (section 2.3). After that, the $3D$ position of X can be derived with a projective ambiguity by triangulation (section 2.4). Finally, a refined solution without projective ambiguity can be obtained by using some other cues from the images (section 2.5).

2.2 Compute the fundamental matrix F

After finding a set of matching points from two images using SIFT keypoints, we find the fundamental matrix F using normalized 8-point algorithm. We combine this with RANSAC to remove outliers from the detected keypoints.

Let (x, x') be a pair of matching points in the two images. $x = [p, q, 1]^T, x' = [p', q', 1]^T$. Substitute them into Equation (1), we have

$$p'pf_{11} + p'qf_{12} + p'f_{13} + q'pf_{21} + q'qf_{22} + q'f_{23} + pf_{31} + qf_{32} + f_{33} = 0 \qquad (4)$$

where

$$F = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}.$$

So we can write Equation (4) in the following form,

$$[p'p \;\; p'q \;\; p' \;\; q'p \;\; q'q \;\; q' \;\; p \;\; q \;\; 1]f = 0,$$

where

$$f = [f_{11} \;\; f_{12} \;\; f_{13} \;\; f_{21} \;\; f_{22} \;\; f_{23} \;\; f_{31} \;\; f_{32} \;\; f_{33}]^T.$$

Since $rank(F) = 2$, we have $det(F) = 0$. We need at least eight pairs of matching points to solve for F.

From a set of n pairs of matching points, we establish the matrix:

$$Af = 0$$

The solution of f is found by using SVD of A: $A = USV^T$ and set f equal to the last column of V. In order to make the matrix A have smaller conditioning number, we first normalize these points, such that in each image, these points are centered at the origin with a RMS distance $\sqrt{2}$.

We combine this algorithm with RANSAC and Sampson error to find the best fundamental matrix from the set of detected matching points.

2.3 Determine camera matrices P and P'

The general formula for a pair of canonical camera matrices corresponding to a fundamental matrix F is give by $P = [I|0]$, and $P' = [M|t]$, where $F = [t]_\times M$ and $t = e'$.

Equation (3) gives $F^T e' = 0$. So to find e', we use SVD on F^T and e' is set to the last column of V (which corresponds to the smallest eigenvalue of F^T). According to [2] (pages 256), as Q. T. Luong and T. Viéville suggested in the paper *Canonic representations for geometrics of multiple projective views*, M can be computed by $M = [e']_\times F$.

2.4 Reconstruct 3D points by Linear Triangulation

Triangulation is the simplest method to compute the 3D point X from the matching images points x, x' given the two camera matrices. First, we have $x = PX$, but x is determined only up to scale in homogeneous coordinates. So we just require that vector x is colinear with vector PX by setting $x \times (PX) = 0$ which gives us two independent equations

$$x(P^{3T}X) - P^{1T}X = 0,$$
$$y(P^{3T}X) - P^{2T}X = 0,$$

where P^{iT} is the ith row of matrix P.

Similarly, we get another 2 equations from x' and P' and we establish an equation $AX = 0$. This equation is solved by SVD method to get X.

The reconstruction here is not unique with a projective ambiguity, which means that we can find an arbitrary projective transformation H such that another point $X_2 = HX$ and another pair of camera matrices $P_2 = PH^{-1}, P_2' = P'H^{-1}$ also satisfies $x = P_2X_2$ and $x' = P_2'X_2$. Therefore the reconstructed scene does not preserve right angles and parallel lines in the original scene.

Besides using this method, we can improve the quality of the reconstruction by using more complex method based on geometric error cost function, Sampson error, and so n. But we still only obtain a projective reconstruction, i.e. a reconstruction with projective ambiguity.

In order to resolve this ambiguity, we need more cues from the images, which will be discussed in the next section.

2.5 Resolve projective ambiguity by additional cues

In order to remove the projective ambiguity, we need more cues from the images to refine the solution. There are many approaches to do this.

In one approach, we transform directly from our projective reconstruction to metric reconstruction (or similar reconstruction) of the scene, which is just ambiguous up to scale. If we know exactly the correct coordinates of at least five scene points then we can compute a homographic matrix H of size 4×4 that maps these points to our reconstruction, and use H to correct our result.

In another approach, named stratified reconstruction, we first move from projective reconstruction to affine reconstruction, then we refine it to a metric reconstruction. This consists of two steps:

- First step: we find a plane in our reconstruction, which is supposed to be actually at infinity. To find this plane, we use some parallel lines as cues: we find the 3 intersections of 3 set of lines in our reconstruction, where each set of lines are actually parallel in real scene. These 3 intersections define a plane, which should be at infinity. After finding that plane, we find a homography H_∞ that maps this plane to the plane at infinity. Then we apply H_∞ to our reconstruction points and the camera matrices to get an affine reconstruction of the scene.

- Second step: we use scene orthogonal lines as cues in this step in order to refine our affine reconstruction above to a metric reconstruction.

Besides, we can find the calibration matrices K and K' of two cameras using three vanishing points of three directions. After that, we use these calibration matrices to normalize image points. Then we calculate the essential matrix from matching normalized points. After that, we compute a pair of canonical camera from the essential matrix and reconstruct the scene points. We may have four possible solutions and we can choose one by testing whether a reconstructed point is in front of both cameras. The cues used in this approach are the parallel lines in the images, which are used to find the vanishing points and the calibration matrices.

In our project, we use the last approach here. First we find the three vanishing points using the parallel lines in three main directions of the scene. From these vanishing points, we can solve for the internal parameters of each camera and the two camera calibration matrices K and K'. Then the essential matrix can be determined by $E = (K')^T FK$. Given the essential matrix E and let the first camera matrix be $P = (I|0)$, the second camera matrix is found from four possible choices: $P' = (UWV^T|+u_3)$ or $P' = (UWV^T|-u_3)$ or $P' = (UW^TV^T|+u_3)$ or $P' = (UW^TV^T|-u_3)$, where U and V are found from SVD decomposition of E, u_3 is the last column of U and

$$W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Only one of these four choices is possible for the second camera. We can find it by testing whether a reconstructed point lies in front of both cameras. Once we obtain the pair of camera matrices P and P' from the essential matrix, we can reconstruct the scene using triangulation discussed above and the reconstruction is a metric reconstruction.

3 Results

Here are the main results of our project. We are able to create a 3D reconstruction of the scene using two images.

3.1 Results based upon section 2

In this experiment, we use the two images in Homework4 Problem2 as shown below:

Figure 2. Two Views of a Scene

We first detect SIFT keypoints (we used the demo code of Lowe at [4]). The SIFT descriptor provides us with an 1×128 vector for each point, which can be used as a candidate for point matching. After we find out all the matching pairs, we use RANSAC to compute the fundamental matrix F. The RANSAC randomly chooses 8 pairs of matching points to compute F. The RANSAC does this process N times totally. Each time, we count the number of inliers based upon this F. In the end after N times, we choose the F with the maximum number of inliers as our fundamental matrix. In our experiment, we set $N = 766$. To decide whether a pair of matching point is an inlier, we use the Sampson distance defined as $S = \frac{\epsilon^2}{(Fx)_1^2+(Fx)_2^2+(F^Tx')_1^2+(F^Tx')_2^2}$, with the decision threshold set as 5.

The important matrices we obtained based upon section 2 are :

The fundamental matrix:

$$F = \begin{bmatrix} 0 & 0 & -0.0007 \\ 0 & 0 & -0.0088 \\ 0.0009 & 0.0091 & 0.0504 \end{bmatrix}$$

The two calibration matrices of cameras:

$$K = \begin{bmatrix} 576.2698 & 0 & 56.6585 \\ 0 & 576.2698 & 118.3710 \\ 0 & 0 & 1 \end{bmatrix}$$

$$K' = \begin{bmatrix} 576.3143 & 0 & 80.8153 \\ 0 & 576.3143 & 106.4760 \\ 0 & 0 & 1 \end{bmatrix}$$

The essential matrix:

$$E = \begin{bmatrix} -0.0134 & 0.6930 & -0.2437 \\ -1.0329 & 0.0535 & -5.1677 \\ 0.3114 & 5.3494 & 0.1879 \end{bmatrix}$$

and the two normalized camera matrices are

$$P_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

and

$$P_2 = \begin{bmatrix} -0.9977 & 0.0020 & 0.0682 & -0.9905 \\ -0.0037 & -0.9997 & -0.0246 & 0.0513 \\ -0.0681 & 0.0248 & -0.9974 & 0.1278 \end{bmatrix}$$

The 3D reconstruction of these matching points as well as the positions and looking directions of the two camera centers are plotted in the right panel below, with the left panel chapel00 as a comparison.

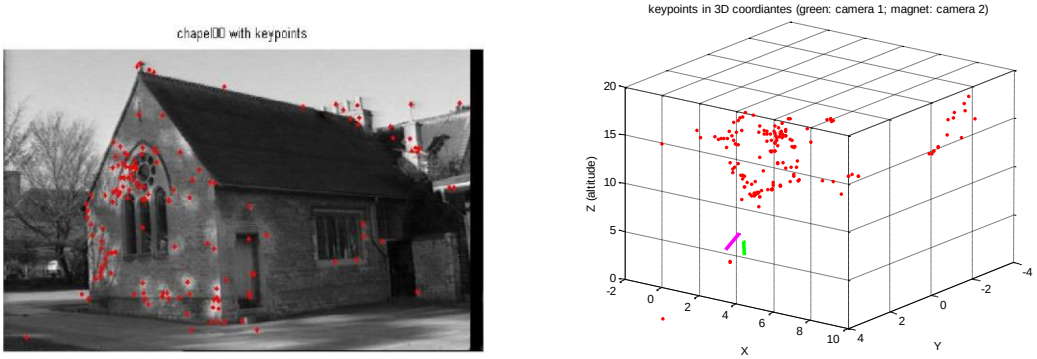

Figure 3. 3D Reconstruction (the Right) of the Keypoints (Red Dots in the Left Image)

From the plot, we can see that our reconstruction gives the relatively correct positions of camera 1(green) for Image "chapel00" and camera 2(magenta) for Image "chapel01", since camera 2 should be left to camera 1 in order to observe more left wall of the chapel. Besides, the dense dots are mostly focused in the left wall of the chapel, with some sporadic dots indicating the right wall of the chapel, the chimney behind, etc.

3.2 Results with Interpolation

In order to show the results more clearly, we can refine our results by interpolation. We first need to pick an interest point in Image 1(chapel00) and find its matching point in Image 2(chapel01) automatically. So we do the following steps:

a) Pick up an interesting point in Image 1;

b) Find out the epipolar line in Image 2 by Equation (2);

c) Now we use a 5×5 window sliding along the epipolar line, and find the minimum SSD compared to the one in Image 1. The corresponding point is our matching point.

Now we can pick up several points at the corners of one plane in a 2D image (e.g., Image 1), and obtain the matching points in Image 2, so the 3D coordinates of these points can be computed. These corners make a convex plane, so the color (or gray level) of points inside this plane is the color (or gray level) of the points $x = P_1 X$ in Image 1 or $x' = P_2 X$ in Image 2.

Here we choose 4 to 5 corner points in each of the left wall, right wall and the roof of the chapel and apply the interpolation. The results of automatically detecting matching points are:

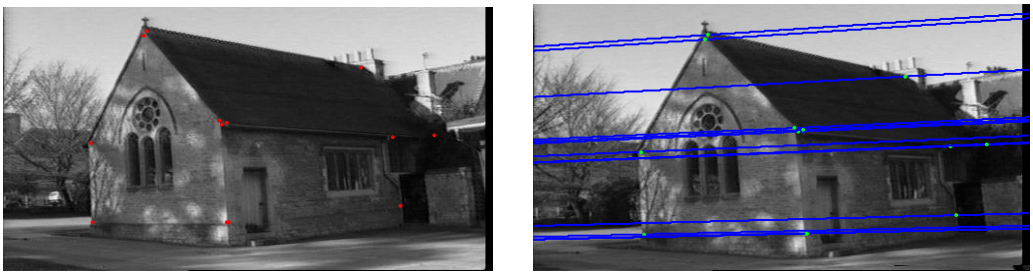

Figure 4. Automatically Detecting Matching Points by a Sliding Window along Epipolar Lines

In the left panel, the red dots are interesting points picked in each of the three planes. In the right panel, the blue lines are their epipolar lines with green dots detected as their matching points. We can see that they are in good agreement.

The final results with interpolation in four different views are:

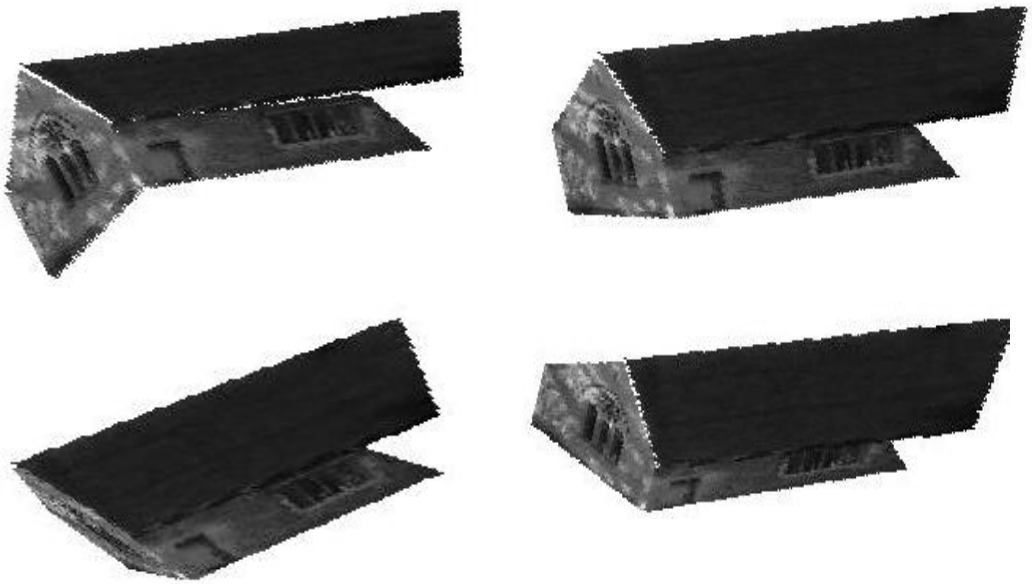

Figure 5. Results after Interpolation (Four Different Viewpoints)

4 Summary

In this project, we aim at reconstructing a 3D object based upon two views. We first detect SIFT keypoints and find out the matching pairs. Then we compute the fundamental matrix by RANSAC, retrieve the camera matrices, and compute the 3D coordinates of interest points with some projective ambiguity. To get rid of the projective ambiguity, we convert this into a calibrated reconstruction by first find the camera calibration matrices from vanishing points in each image, then we compute the essential matrix. After that, we retrieve the pair of canonical cameras from the essential matrix and obtain a 3D reconstruction by triangulation. To refine our results, we did the interpolation in each of the object's planes. The final results give a good reconstruction.

Yiyi Huang and Huy Bui wrote the report and created the poster together, besides, the collaboration is

Yiyi Huang:

1) Suggested this topic for our project and searched for several papers and book chapters as our main resources;

2) Suggested using SIFT to find interesting points and also suggested applying RANSAC for computation of F;

3) Tested Huy's codes for 3D reconstruction;

4) Implemented the Matlab code for section 3.2 (results with interpolation), including automatically detecting a matching point by sliding a window along an epipolar line, and interpolating the plane of an object to obtain better visual effects.

Huy Bui:

1) Wrote Matlab code to compute the fundamental matrix using normalized 8-point algorithm;

2) Implemented RANSAC with Sampson error to compute the fundamental matrix from set of matching points;

3) Implement 3D reconstruction algorithm (find calibration matrices from vanishing points, compute essential matrix, retrieve camera matrices and reconstruct 3D points by triangulation);

4) Tested Yiyi's codes for results with interpolation.

To further improve the results, more can be done. For example, we can remove the shadow or the light effect in the left wall, since the shadow would change when the viewpoint is different. Besides, here we interpolate the chapel in 3 planes. Actually it is not flat in any one of the three planes (e.g., the window in the right wall). It would be better if we can model these 3D planes, though complexity would rise greatly.

References

[1] Forsyth, D.A and Ponce, J. (2002) *Computer Vision: A Modern Approach*. Prentice Hall.

[2] Hartley, R. and Zesserman, A. (2001) *Multiple View Geometry In Computer Vision*. Cambridge University Press.

[3] Hoiem, D., Forsyth, D.A and Hedau, V. (2010) Slides and lecture notes in Computer Vision. UIUC. http://www.cs.illinois.edu/homes/dhoiem/

[4] Lowe, D. (2005) Demo Software: SIFT Keypoint Detector http://www.cs.ubc.ca/~lowe/keypoints/