Epipolar Geometry

Frank Cally Tabuco
fatabuco@up.edu.ph

Categories and Subject Descriptors

•Computing methodologies → Image representations;

Keywords

epipolar geomtery, triangulation, harris detector

1. INTRODUCTION

Computer vision follows the same procedures instantenously processed by humans when perceiving a certain scene or image. It involves the acquisition of an image through an eye or lens and interpretation of pixels, shapes, and colours through the brain. The main goal of computer vision is to analyze multiple images and videos to achieve human-level perception or even greater. This task would help in various fields requiring high level of image analysis such as biomedicine[4], physiology[2], and bioacoustics[3].

In this programming assignment, the task is to use the OpenCV libraries[1] to perform epipolar geometry and triangulation of two images caputered at different camera angles. We aim to fit a fundamental matrix given matched coordinates and perform the same procedure for matches obtained from putative matches from the previous assignment. The rest of the paper is structured as follows: section 2 will discuss the techniques and algorithms employed to obtain 3D plots. In section 3, the results of triangulation and 3D plotting are shown and evaluated against ground truth matches versus putative matches. The last section will discuss the conclusions for this assignment.

2. METHODOLOGY

2.1 Fundamental Matrix and Residuals

Two images of a library building captured at different angles (see Figure 1) are provided along with ground-truth matched points, *library_matches.txt* . These matched points are coordinates of pixels in the left image and their corresponding points located in the right image. Since no headers are provided for the match points matrix, we assumed that the first two columns are (x,y) coordinates of the left image and the last columns are for the right image. The matrix is sliced to form cam1_pts and cam2_pts with shapes 309 x 2, which are then used to find the fundamental Matrix. To fit a fundamental matrix from these points, we used two methodologies: 1) OpenCV built-in function for finding the fundamental matrix given by *cv.findFundamentalMat()*, and 2) built a function for estimating the Fundamental Matrix from scratch which used algorithms in [1]. We employed the normalized 9-point algorithm. The functions outputs a 3 x 3 fundamental matrix, and a mask consisting of points considered to be outliers and inliers. Only inlier points will be considered as final image coordinates to be used for triangulation and this resulted to fewer matrix points of 285 x 2.

Figure 1: Library images.

The 2D points, cam1_pts and cam2_pts, are converted into a 3D point vector by appending a values of 1.

$$\begin{pmatrix} fX/Z \\ fY/Z \end{pmatrix} \rightarrow \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

The initial shape of 285 x 2 are transformed to 285 x 3, which is the image point homogenous vector. Given these new vectors, we can form the corresponding epipolar lines by using the properties of the fundamental matrix. Epipolar lines are created using the fundamental matrix and the corresponding points of an image. These are given by:

$$l' = Fx \quad (1)$$

$$l = F.Tx' \quad (2)$$

ACM ISBN XXXXXXXX.
DOI: XXXXXXX

[1] https://www.cc.gatech.edu/classes/AY2016/cs4476_fall/results/proj3/ht

where equation 1 is the epipolar line on library2 image corresponding to points in library1 image. The seconnd equation uses the transpose of the fundamental matrix to form the epipolar line for points in image 2.

In order to evaluate the fundamental matrix in terms of the correponding epipolar lines for both image point vectors, we need to compute the residuals. Residuals are computed as the average squared distance between points on one image and its corresponding epipolar line. The distance formula is shown below:

$$distance(line, imagepoints(x_0, y_0) = \frac{|ax_0 + by_0 + c|}{\sqrt{a^2 + b^2}} \quad (3)$$

The mean square distance of the results from equation 3 constitutes the residuals for corresponding points and lines.

2.2 Camera Center

Aside from the matched points matrix and images, we are also provided by a 3 x 4 camera projection matrix. This matrix is shown in the camera calibration constraint below:

$$x = PX \quad (4)$$

$$P = \begin{bmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix} \quad (5)$$

where,

x = homogenous image point vector
P = a 3 x 4 projection matrix consisting of internal calibarion, rotaion, and translation from world to camera coordinate frame
X = homogenous world point vector [5, 6]

We can estimate the camera centers by computing for the null space of the camera projection matrix. The null space is computed by solving for the homogenous system *Ax = 0*. This can be achieved through singular value decomposition (SVD) or by reducing a matrix to its row-echelon form. For this implementation, we used Scipy's implementation of SVD using the function *null_space()* [8].

2.3 Triangulation

Triangulation is done to compute for the corresponding world point homogenous coordinates given a set of image points and camera projection matrices. For this implementation, we used the Least Square Triangulation algorithm [7] to estimate 4D points.

Given equations x = PX and x' = P'X, we need to solve for X. We know that P is a 3 x 4 matrix and can be written in short-hand by:

$$P = \begin{bmatrix} p^{1T} \\ p^{2T} \\ p^{3T} \end{bmatrix}$$

we then need to eliminate unknown scale in λx = PX by computing for the cross produc x (dot) PX = 0. After some rearrangement, we can creation a homenous system for the first camera given by:

$$\begin{bmatrix} xp^{3T} - p^{1T} \\ yp^{3T} - p^{2T} \end{bmatrix} X = 0 \quad (6)$$

the same manipulations are done for the seoncd camera which yields a system of:

$$\begin{bmatrix} x'p'^{3T} - p'^{1T} \\ y'p'^{3T} - p'^{2T} \end{bmatrix} X = 0 \quad (7)$$

combining these two systems we have 4x4 A matrix. All we need to do now is solve for X.

Finally, we can evaluate the results of the triangulation algorithm by computing for the residuals. In this case, the residual is computed by the mean-squared distance of the image points and the product of matrix multiplications for the camera projection matrix and homogenous world point coordinates.

The same procedures of fitting a fundamental matrix, computing for the residuals, and triangulation are done for the putative matches obtained using codes from programming assignment 2.

3. RESULTS

Evaluating the resulting epipolar line from the fundamental matrix requires computing for the residuals. We estimated the residuals from ground-truth matches and putative matches from programming assignment 2 (see Table 1). We can see from the results that residuals from ground-truth matches are smaller compared to matches obtained using Harris Corner Detection. One explanation for this is the lack of accuracy when using Harris Corner Detector than other state-of-the-are feature detectors. Additionally, Harris is not invariant to scale which could explain the dissimilarity in results.

Table 1: Residuals of matched points and epipolar line.

	True Matches	Putative Matches
Library 1 (OpenCV)	0.1976	1.0669
Library 1 (Scratch)	229.9445	519.6397
Library 2 (OpenCV)	0.2093	0.8963
Library 2 (Scratch)	195.45	648.6620

Obtaining the camera centers requires computing for the null space of the provided camera projection matrix. After solving the homogenous system *Ax = 0*, the camera center vectors are shown in table 2.

Table 2: Camera Centers.

	Centers
Projection Matrix 1	$\begin{bmatrix} -0.25271561 \\ 0.74619484 \\ -0.61491942 \\ -0.03467258 \end{bmatrix}$
Projection Matrix 2	$\begin{bmatrix} 0.23876263 \\ -0.53308432 \\ 0.81093406 \\ 0.03463312 \end{bmatrix}$

The triangulation algorithm allowed us to estimate the 3D points using image points and the projection matrix. To evaluate the results, residuals were also computed. Computing for the residuals uses the epipolar constraints x = PX and x' = P'X. We need to achieve a small error where the results of the right-hand side is close to the values of the left-hand side. We first computed for the dot product of PX and P'X. The points are then compared with the image points , x and x', using Euclidean distance. The results of the computations are shown in table 3.

Table 3: Camera Centers.

	True Matches	Putative Matches
Library 1	11.9415	50097.0809
Library 2	16.0346	73383.3018

We can see that the ground-truth matches yielded residuals below 20. However, for putative matches the results were too high. There are two reasons for this. The first one was already previously mentioned where the use of Harris Detector affected the accuracy of matched points. More important, however, is the lack of a projected camera matrix from the putative matches. The camera projection matrix used for the putative matches triangulation were from the ground-truth matches. This resulted to mismatching correspondences. Additionally, we cannot estimate camera projection matrices for the putative matches since we also lack corresponding ground-truth 3D points.

From the camera center vectors and triangulated 3D points obtained above we can visualize the results using a scatter plot (see Figure 2). As expected, the results for the visualization of triangulated points for the putative matches yielded an incomprehensible graph (see Figure 3).

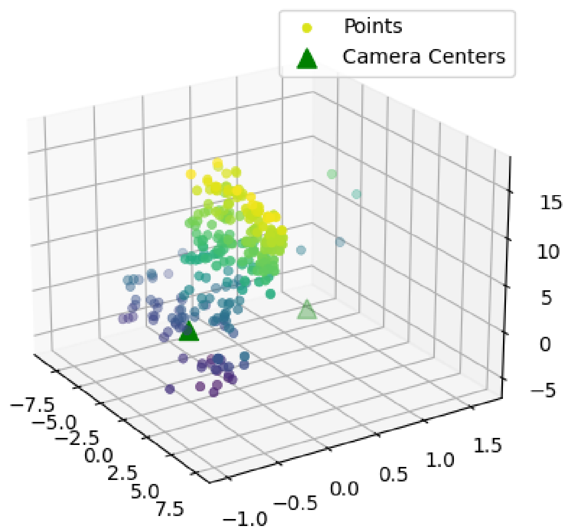

Figure 2: Visualization of Triangulated Points and Camera Centers for True Matches.

4. CONCLUSION

In this paper, we have provided a step-by-step implementation of epipolar geometry and triangulation given two images. We were able to analyze the intricancies of the different underlying matrices involved in camera calibration and image point representations. One important result for these experiments, is the need for sufficient matrices to solve certain systems. For the putative matches, we lacked information on both the projection matrix and the corresponding world point coordinates which resulted to subpar results.

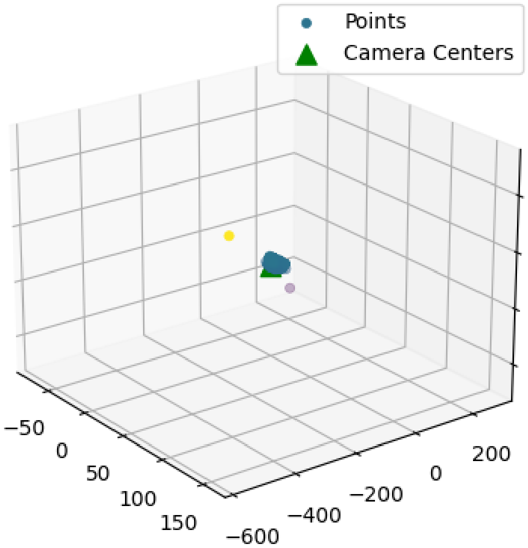

Figure 3: Visualization of Triangulated Points and Camera Centers for Putative Matches.

5. REFERENCES

[1] Bradski, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).

[2] Cho, Y. Rethinking eye-blink: Assessing task difficulty through physiological representation of spontaneous blinking. *In Proceedings of CHI Conference on Human Factors in Computing Systems (CHI'21)* (May 2021).

[3] Lasseck, M. Acoustic bird detection with deep convolutional neural networks. In *Detection and Classification of Acoustic Scenes and Events 2018* (2018).

[4] Lopez, F., Varela, A., Hinojosa, O., Mendez, M., Trinh, D.-H., ElBeze, J., Hubert, J., Estrade, V., Gonzalez, M., Ochoa, G., and Daul, C. Assessing deep learning methods for the identification of kidney stones in endoscopic images, 2021.

[5] Naval, P. Camera models, 2021.

[6] Naval, P. Epipolar geometry, 2021.

[7] Naval, P. Stereo correspondence, 2021.

[8] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods 17* (2020), 261–272.