GEO-1016 Assignment 2: Triangulation

Group 9: Ming-Chieh Hu, Neelabh Singh & Daan Schlosser

March 28, 2025

Abstract

Triangulation is a fundamental technique in photogrammetry and 3D computer vision used to reconstruct 3D points from multiple 2D images. This report outlines the methodology (Section 1) for estimating the fundamental matrix, recovering the essential matrix, determining the relationship between two camera poses, and reconstructing 3D points. The process utilizes the normalized eight-point algorithm, Direct Linear Transform (DLT) method, and the Levenberg-Marquardt algorithm for further refinement. In Section 2, experiments are conducted to qualitatively and quantitatively compare the effects of varying camera intrinsics. Section 3 examines reconstruction accuracy, the impact of intrinsic camera parameters, and their estimation using Bundle Adjustment.

1 Methodology

The problem we intend to solve is a simplified triangulation problem with only two cameras having the same intrinsic matrix $K = K'$ and the following equations:

$$\begin{aligned} \text{Fundamental matrix:} \quad & F = K'^{-T} E K^{-1} \\ \text{Essential matrix:} \quad & E = [\mathbf{t}_\times] R \\ \text{Projection matrices:} \quad & M = K[I \quad \mathbf{0}], \quad M' = K[R \quad \mathbf{t}] \\ \text{Epipolar constraint:} \quad & \mathbf{p}_i'^T F \mathbf{p}_i = 0 \end{aligned} \tag{1}$$

The following sections are organized as steps that one needs to perform for a triangulation process.

1.1 Estimate fundamental matrix F using the normalized 8-point algorithm

Normalization For each image with corresponding points, with at least 8 correspondences, we compute the centroid of the points and ensure the mean distance from the centroid is $\sqrt{2}$:

$$\text{scale} = \frac{\sqrt{2}}{\frac{1}{N}\sum_{i=1}^{N} \|\mathbf{p}_i - \text{centroid}\|}, \text{ with centroid} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{p}_i \tag{2}$$

Construct the similarity transformation matrix and apply the transformation to the points:

$$\mathbf{q}_i = T \cdot \mathbf{p}_i, \text{ with } T = \begin{bmatrix} \text{scale} & 0 & -\text{scale} \times \text{centroid}_x \\ 0 & \text{scale} & -\text{scale} \times \text{centroid}_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

First linear solution Expand the epipolar constraint and reformat the equation as two matrices $W\mathbf{f} = \mathbf{0}$, where W is a $N \times 9$ matrix and $\mathbf{f}$ is a 9×1 matrix we obtained from reshaping F_{norm}:

$$W\mathbf{f} = \begin{bmatrix} u_1'u_1 & u_1'v_1 & u_1' & v_1'u_1 & v_1'v_1 & v_1' & u_1 & v_1 & 1 \\ u_2'u_2 & u_2'v_2 & u_2' & v_2'u_2 & v_2'v_2 & v_2' & u_2 & v_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_i'u_i & u_i'v_i & u_i' & v_i'u_i & v_i'v_i & v_i' & u_i & v_i & 1 \end{bmatrix} \begin{bmatrix} F_{norm,11} \\ F_{norm,12} \\ F_{norm,13} \\ F_{norm,21} \\ F_{norm,22} \\ F_{norm,23} \\ F_{norm,31} \\ F_{norm,32} \\ F_{norm,33} \end{bmatrix} = 0 \tag{4}$$

We then decomposed W as $U\Sigma V^T$ using SVD and use the last column of V as the solution (corresponds to smallest singular value).

Rank-2 constraint SVD provides an estimate of the fundamental matrix, denoted as $\hat{F}$. However, $\hat{F}$ may have full rank, while the true fundamental matrix is known to have rank 2 (because $[\mathbf{t}_\times]$ also has rank 2). To enforce this constraint, we perform SVD on $\hat{F}$ and set the smallest singular value d_3 to zero, yielding:

$$\hat{F} = UDV^T, \quad F = U \begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T. \tag{5}$$

Denormalization To use the computed fundamental matrix F_{norm} in the original coordinate space, we need to de-normalize it. This is achieved by substituting $\mathbf{q}_i$ with $T\mathbf{p}_i$ and $\mathbf{q}'_i$ with $T'\mathbf{p}'_i$:

$$\begin{aligned} \mathbf{q}_i'^T F_{norm} \mathbf{q}_i &= 0 \\ \mathbf{p}_i'^T F \mathbf{p}_i &= 0 \\ \mathbf{p}_i'^T T'^T F_{norm} T \mathbf{p}_i &= 0 \\ F &= T'^T F_{norm} T \end{aligned} \tag{6}$$

1.2 Compute essential matrix E

For our simplified case where $K = K'$, the essential matrix can be derived easily by:

$$E = K^T F K \tag{7}$$

1.3 Find the 4 candidate poses from the fundamental matrix

Recover rotation matrix R For a 3×3, real skew-symmetric matrix $[\mathbf{t}_\times]$, we'll always get one eigenvalue 0 and two imaginary eigenvalues $\pm i\lambda$. Since $[\mathbf{t}_\times]$ is real and symmetric after multiplication by i, it is diagonalizable with an orthonormal basis of eigenvectors. That means for a skew-symmetric matrix known up to scale, we can find an orthogonal matrix U and such that:

$$[\mathbf{t}_\times] = UZU^T, \quad Z = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{8}$$

Here, we introduce another matrix W to pair with Z and diagonalize ZW, defined as:

$$W = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{so that} \quad \begin{aligned} ZW &= diag(1,1,0) \\ ZW^T &= -diag(1,1,0) \end{aligned} \tag{9}$$

We know E up to scale, and by SVD we know that it has the form $E = Udiag(1,1,0)V^T$, thus we get:

$$\begin{aligned} E =& \pm Udiag(1,1,0)V^T \\ E =& UZWV^T \quad \text{or} \quad UZW^TV^T \\ E =& (UZU^T)UWV^T \quad \text{or} \quad (UZU^T)UW^TV^T \end{aligned} \tag{10}$$

And the final factorizations of E:

$$[\mathbf{t}_\times] = UZU^T, \; R_1 = UWV^T, \; R_2 = UW^TV^T \tag{11}$$

Recover translation matrix $\mathbf{t}$ Using the cross-product definition, we know that $\mathbf{t} \times \mathbf{t} = 0$, which leads to $[\mathbf{t}_\times]\mathbf{t} = UZU^T\mathbf{t} = 0$. Since U is unitary, we find that $||[\mathbf{t}_\times]|| = \sqrt{2}$. Given that E is known only up to scale, we estimate $\mathbf{t}$ as:

$$\mathbf{t} = \pm U \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \pm \mathbf{u}_3 \tag{12}$$

Where $\mathbf{u}_3$ is the third column of U. This confirms that $\mathbf{t}$ is determined up to a sign.

1.4 Triangulation

In theory, the coordinates of 3D points can be easily determined by extending the line of sight and finding the intersection point. However, in practice, various error sources—such as inaccuracies from SVD estimations and noise from low pixel resolution—can prevent us from finding an exact intersection. That's why we have to re-estimate to obtain a plausible solution.

Linear method By using the definition of the cross product, $\mathbf{p} \times (M\mathbf{P}) = 0$. By expanding the 3 dimension from the cross product result we get:

$$\mathbf{p}(M\mathbf{P}) = \begin{vmatrix} i & j & k \\ x & y & 1 \\ \mathbf{m}_1^T\mathbf{P} & \mathbf{m}_2^T\mathbf{P} & \mathbf{m}_3^T\mathbf{P} \end{vmatrix} \quad \text{and thus} \quad \begin{aligned} y\mathbf{m}_3^T\mathbf{P} - \mathbf{m}_2^T\mathbf{P} = 0 \\ x\mathbf{m}_3^T\mathbf{P} - \mathbf{m}_1^T\mathbf{P} = 0 \\ x\mathbf{m}_2^T\mathbf{P} - y\mathbf{m}_1^T\mathbf{P} = 0 \end{aligned} \tag{13}$$

Here, we only keep the first two equations per camera because the third equation is a linear combination of the first two and does not provide additional constraints (Hartley and Zisserman (2003), section 12.2, p.312). This results in a system of four equations in four homogeneous unknowns (X, Y, Z, W) which can be solved using SVD to obtain the best linear estimate of the 3D point $\mathbf{P}$.

$$A\mathbf{P} = 0 \quad \text{where} \quad A = \begin{bmatrix} x\mathbf{m}_3^T - \mathbf{m}_1^T \\ y\mathbf{m}_3^T - \mathbf{m}_2^T \\ x'\mathbf{m}_3'^T - \mathbf{m}_1'^T \\ y'\mathbf{m}_3'^T - \mathbf{m}_2'^T \end{bmatrix} \tag{14}$$

Determine the correct relative pose We ran the linear triangulation 4 times on 4 candidate poses to obtain 3D point $\mathbf{P}$ in the first camera's coordinate system and thus the $\mathbf{Q}$ in the second camera's coordinate system. The relationship between $\mathbf{P}$ and $\mathbf{Q}$ is simply:

$$\mathbf{Q} = R\mathbf{P} + \mathbf{t} \tag{15}$$

The correct R and $\mathbf{t}$ (relative pose) is simply given by the configuration with the most visible points.

Non-linear least-squares refinement After, obtaining the initial 3D points using linear triangulation method, we also applied a non-linear refinement step to minimize the reprojection error further. This is achieved using the Levenberg-Marquardt (LM) optimization algorithm, which iteratively adjusts the 3D points to better align with the observed 2D points in both images. The goal of non-linear refinement is to minimize the reprojection error, which is defined as follows:

$$\text{Reprojection Error (E)} = \sum_{i=1}^{N} \left(\|M\hat{P}_i - p_i\|^2 + \|M'\hat{P}_i - p_i'\|^2 \right) \tag{16}$$

Where:

- N is the total number of point correspondences.
- M and M' are the projection matrices for the first and second cameras, respectively.
- $\hat{P}_i$ is the reconstructed 3D point in homogeneous coordinates for the i-th correspondence.
- p_i and p_i' are the observed 2D points in the first and second images for the i-th correspondence.
- $\|\cdot\|^2$ denotes the squared Euclidean distance.

2 Experiments and Results

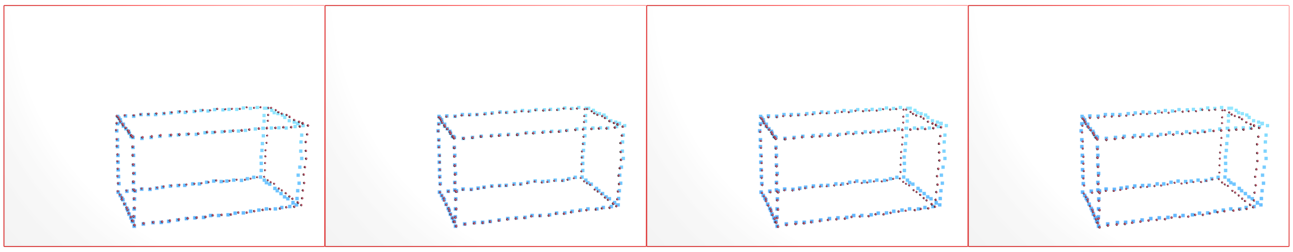
To gain a better understanding of a camera's intrinsic parameters, we synthetically changed the camera's intrinsic parameters without changing the taken images. Thereby visualizing what kind of impact that has on reconstructed 3D points. The intrinsic matrix defines how, inside a camera, 3D points are mapped to 2D image coordinates. It is typically represented as:

$$K = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{17}$$

Where: f_x, f_y represents the focal length (scaling factor in pixels), c_x, c_y represents the principal point (optical center of the camera) and s represents the skew (usually 0 for most cameras).

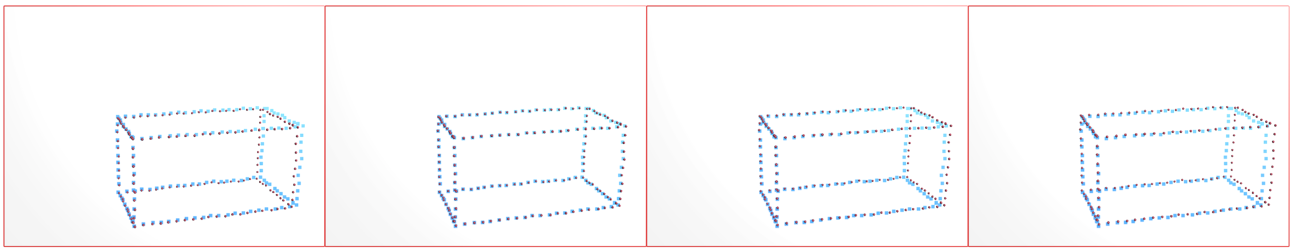
2.1 Changing f_x or f_y (focal length)

f_x, f_y define the focal length of the camera model. In physical terms, adjusting f_x or f_y simulates zooming in or out, as seen in Figure 1 and 2. If f_x, f_y **increase**, the 3D reconstruction appears closer or zoomed in (narrower field of view). If f_x, f_y **decrease**, the 3D reconstruction appears further away or zoomed out (wider field of view).



(a) $f_x = 970$ (b) $f_x = 1010$ (c) $f_x = 1030$ (d) $f_x = 1050$

Figure 1: Effect of varying f_x while keeping other parameters constant.

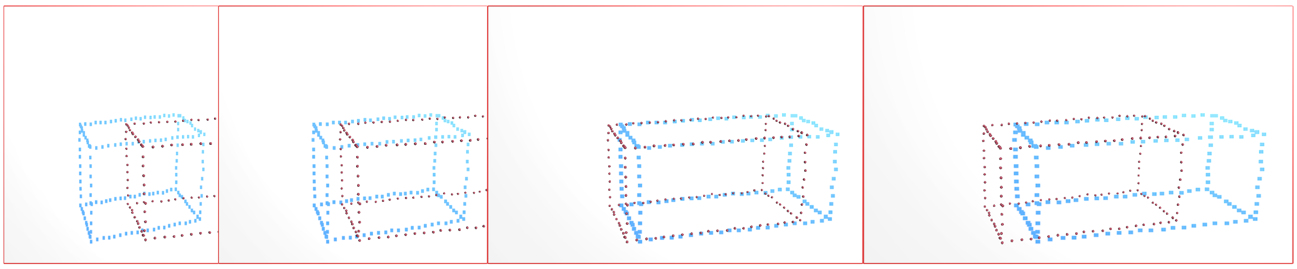


(a) $f_y = 970$ (b) $f_y = 1010$ (c) $f_y = 1030$ (d) $f_y = 1050$

Figure 2: Effect of varying f_y while keeping other parameters constant.

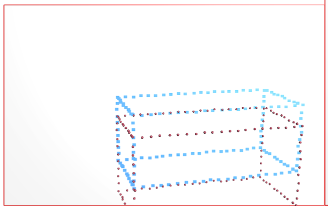
2.2 Changing c_x, c_y (principal point)

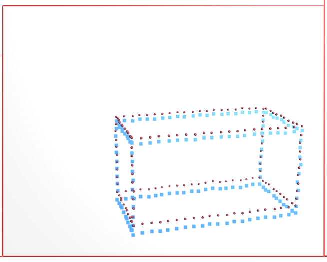
c_x, c_y represents the principal point (optical center) of the camera's intrinsic parameters. If c_x **increases**, the reconstruction shifts **right**. If c_x **decreases**, the reconstruction shifts **left** (Figure 3). If c_y **increases**, the reconstruction shifts **down**. If c_y **decreases**, the reconstruction shifts **up** (Figure 4). Additionally, since c_x, c_y are inherently related to the image size, altering these, the image size must also change.

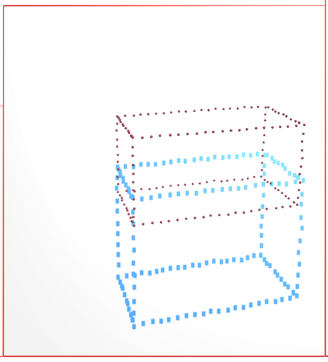


(a) $c_x = 200$ (b) $c_x = 250$ (c) $c_x = 350$ (d) $c_x = 400$

Figure 3: Effect of varying c_x while keeping other parameters constant.


(a) $c_y = 200$


(b) $c_y = 250$


(c) $c_y = 350$

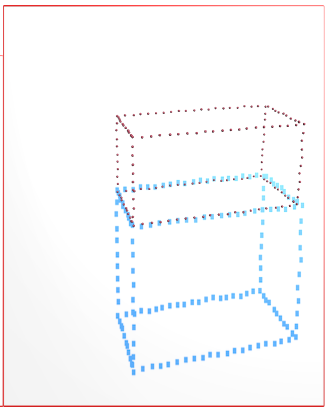
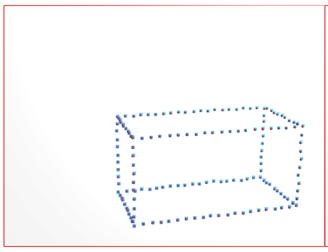
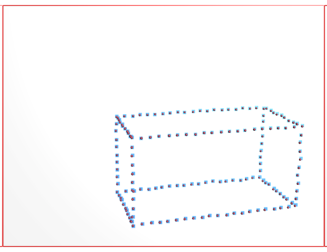

(d) $c_y = 400$

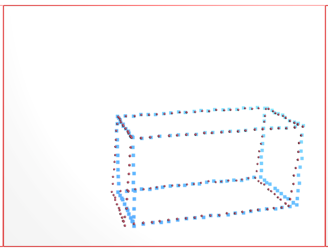
Figure 4: Effect of varying c_y while keeping other parameters constant.

2.3 Changing s (skew)

s represents the skew between the x and y axes of the image sensor. A nonzero s means that the pixels are skewed, which forms a parallelogram instead of a rectangular pixel grid. If $s \neq 0$, it introduces **shearing**, which makes the reconstruction **tilted**, as seen in Figure 5.


(a) $s = 1$


(b) $s = 10$


(c) $s = 100$

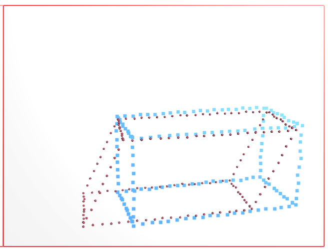

(d) $s = 500$

Figure 5: Effect of varying skew s while keeping other parameters constant.

2.4 Quantifying reconstruction accuracy

To quantify the accuracy of our 3D reconstruction, we calculate the reprojection error by projecting the reconstructed 3D points onto the two image planes using the intrinsic matrix K and extrinsic parameters (R, $\mathbf{t}$). Then we compute the squared Euclidean distance between the projected points and the corresponding ground truth 2D points in both images. The total error is the sum of these suqared distances. . We did this for both the linear optimization and the non-linear optimization, through the latter, we try to minimize this error (equation 16). In Table 1 we quantified our parameter tuning results through the reprojection error.

					Reprojection error, divided by the number of points	
f_x	f_y	c_x	c_y	s	before non-linear refinement	after non-linear refinement
970	1000	320	240	0	7.31008	7.21069
1010	1000	320	240	0	6.95709	6.81606
1030	1000	320	240	0	23.9733	23.5079
1050	1000	320	240	0	53.3184	52.2937
1000	**970**	320	240	0	22.9956	22.5886
1000	**1010**	320	240	0	1.84389	1.81324
1000	**1030**	320	240	0	8.76131	8.64064
1000	**1050**	320	240	0	28.1265	27.7015
1000	1000	**200**	240	0	12.0842	11.809
1000	1000	**250**	240	0	6.2082	6.07866
1000	1000	**350**	240	0	4.10145	4.03312
1000	1000	**400**	240	0	11.5661	11.4398
1000	1000	320	**200**	0	5.39802	5.27091
1000	1000	320	**250**	0	2.36608	2.31953
1000	1000	320	**350**	0	2.72671	2.69084
1000	1000	320	**400**	0	9.90169	9.78545
1000	1000	320	240	**1**	2.86391	2.80484
1000	1000	320	240	**10**	2.84665	2.78737
1000	1000	320	240	**100**	4.16776	4.07344
1000	1000	320	240	**500**	163.588	158.211
1000	1000	320	240	0	2.86793	2.80885

Table 1: Reprojection errors for various parameter settings

3 Discussion and Reflection

3.1 Obtaining accurate intrinsic parameters in practice

The intrinsic camera parameters—including the focal lengths (f_x and f_y), the camera's principal point (c_x and c_y), and the skew coefficient (s)—are usually obtained through camera calibration. In this assignment the ground truth intrinsic parameters were given. Yet, an accurate estimation of the intrinsic camera parameters can be obtained from Bundle Adjustment.

In the Bundle Adjustment process, the intrinsic camera parameters are incorporated into the objective function. The optimization refines both intrinsic and extrinsic parameters by minimizing the total reprojection error across all images and points. The process is highly sensitive to its initial values. Without a good starting estimate, Bundle Adjustment may fail to converge to a global minimum. Therefore, an accurate initial estimation is crucial for successful optimization. A decent first guess for intrinsic parameters can be made using reasonable approximations:

- Principal Point (c_x, c_y): Typically near the image center, approximated as $c_x \approx \frac{w}{2}$ and $c_y \approx \frac{h}{2}$, where w and h are the image width and height.
- Skew (s): Assumed to be close to zero, as modern cameras generally have nearly perpendicular axes.

For focal lengths (f_x, f_y) and distortion parameters, initial estimates can also be obtained from image metadata, camera SDKs or APIs, ensuring a more stable optimization process.

3.2 Accuracy

The reprojection error in Equation (16) is a key metric for evaluating the accuracy of 3D reconstructions as it measures the discrepancy between the observed 2D points in the images and the projected 2D points computed from the reconstructed 3D points and camera parameters. The reprojection error was evaluated after two different methods of triangulation: **linear triangulation** and **non-linear least-squares refinement** and both the methods were applied to the same set of 3D points and camera parameters to compare their accuracy. The initial 3D points were computed using linear triangulation, which solves a system of linear equations derived from the camera projection matrices and 2D correspondences. Further on, The 3D points were refined using non-linear optimization (Levenberg-Marquardt) to minimize the reprojection error.

The result for the reprojection error after linear triangulation ($E_{\text{linear}} = 458.869$) was higher , indicating initial inaccuracies in the 3D reconstruction and after non-linear refinement, the reprojection error ($E_{\text{non-linear}} = 449.416$) decreased by a certain, showing the effectiveness of the optimization process in improving the alignment between the 3D points and the observed 2D points. As we observed in our implementation, the change in value of reprojection error after non-linear refinement was not significant enough, we assume that this could be due to the use case of only 2 images and 160 correspondence points, and this method can refine the reprojection error significantly in the case of multiple images.

3.3 Reflection on feedback

We had a discussion with another group about both our code and theirs. They were quite positive about our implementation, finding it well structured and effective. In general, they did not have any comments on our code. We did ask them about the non-linear triangulation method and implementation, but they chose not to do that part.

Credit

- Ming-Chieh Hu (6186416): Methodology section, report revision, formatted the code, implementation of linear and non-linear methods.
- Neelabh Singh (6052045): Initial code for non-linear method and reprojection error, non-linear and accuracy in report
- Daan Schlosser (5726042): Experiments and results section. Wrote the code to estimate fundamental and essential matrices (which Jerry also did and we compared our implementations). Wrote parts of discussion and helped bugfixing and revised the methodology.

References

Hartley, R. and Zisserman, A. (2003). *Multiple View Geometry in Computer Vision*, volume 2. Cambridge University Press.