IIIT Hyderabad **Scribed By: Nikhil Bansal, Mohit Sharma**
Mobile Robotics (CSE-483) **Topic: Triangulation**
Instructor: Dr. K Madhava Krishna **September 10, 2019**

Outline *In this paper, we consider the problem of finding the position of a point in space given its position in two images taken with cameras with known calibration and pose. This process requires the intersection of two known rays in space, and is commonly known as triangulation. In the absence of noise, this problem is trivial. When noise is present, the two rays will not generally meet, in which case it is necessary to find the best point of intersection.*

Contents

1 Recap

In the previous lectures, we derived that two perspective cameras observing the same point must satisfy the epipolar constraints.

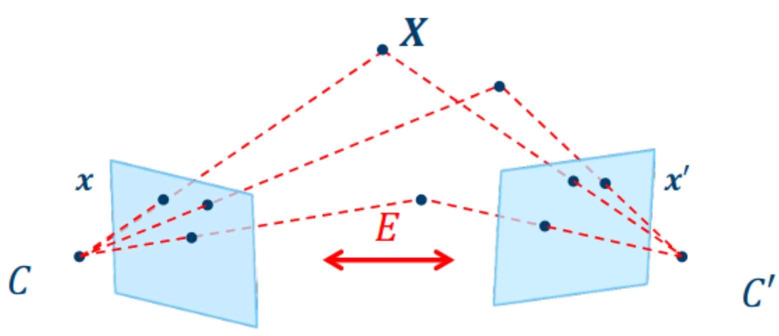

Figure 1: Epipolar Geometry

If x and x' are image points of world point X on the two cameras, then, they must satisfy:

$$x'_T F x = 0$$

$$x'_T E x = 0$$

where E and F are Essential matrix and Fundamental matrix.

And, if P and P' are the camera matrices of the two cameras, then, X must satisfy:

$$PX = x$$

$$P'X = x'$$

Now, we will look at how we can estimate $3D$ point X from known camera matrices P and P' and $2D$ correspondences $x \leftrightarrow x'$.

2 Introduction

Let's assume that we know the camera matrices P and P' and a set of correspondences $x_i \leftrightarrow x'_i$. In order to determine 3D point X_i, one can project the two image points and their intersection should give X_i. But, due to noise in determining correspondences and the $3D$ points, the two rays from x and x' may not intersect in $3D$. And so, we need to determine X_i using optimization methods.

There are thus, several ways to approach the problem depending on what we choose to optimize over. We can optimize over:

1. Errors in correspondences i.e. $x_i \leftrightarrow x'_i$, or
2. Errors in both $x_i \leftrightarrow x'_i$, P and P'.

3 The $3D$ mid-point[1]

Naturally, one would try to minimize the $3D$ error between actual point X_i and estimated point $\hat{X}_i$. For this, we need to take the midpoint of the shortest line between the two back-projected rays.

There are two disadvantage of this approach:

1. It does not minimize the re-projection error: $\sqrt{(x_i - (P\hat{X}_i)) + (x'_i - (P'\hat{X}_i))}$.
2. It does not extend to situations when object is observed by more than two cameras.

This would lead us to the next method **'Linear Triangulation'**, which tries to minimize the re-projection error.

4 Linear triangulation[2]

This method is analogous to **DLT**. Given the correspondence $x_i \leftrightarrow x'_i$ and lets assume the estimation for X_i is $\hat{X}_i$, then:

$$x_i = P\hat{X}_i$$

$$\implies x_i \times P\hat{X}_i = 0$$

$$\implies \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} \times \begin{bmatrix} P_1^T \\ P_2^T \\ P_3^T \end{bmatrix} \hat{X}_i = 0$$

where $P = [P_1 P_2 P_3]$, P_i is the ith column of projection matrix P.

$$\implies \begin{bmatrix} 0 & -1 & v_i \\ 1 & 0 & -u_i \\ -v_i & u_i & 0 \end{bmatrix} \begin{bmatrix} P_1^T \\ P_2^T \\ P_3^T \end{bmatrix} \hat{X}_i = 0$$

$$\implies \begin{bmatrix} v_i P_3^T - P_2^T \\ P_1^T - u_i P_3^T \\ u_i P_2^T - v_i P_1^T \end{bmatrix} \hat{X}_i = 0$$

In the above matrix, only 2 rows are linearly independent (i.e rank of above matrix = 2), which implies:

$$\implies \begin{bmatrix} v_i P_3^T - P_2^T \\ u_i P_3^T - P_1^T \end{bmatrix} \hat{X}_i = 0 \tag{1}$$

Similarly, we can derive the equation for $2nd$ camera with Projection matrix P':

$$\implies \begin{bmatrix} v_i' P_3'^T - P_2'^T \\ u_i' P_3'^T - P_1'^T \end{bmatrix} \hat{X}_i = 0 \tag{2}$$

Combining equations (1) and (2), we get an over determined homogeneous system of linear equations that we can solve with **SVD**.

$$\begin{bmatrix} v_i P_3^T - P_2^T \\ u_i P_3^T - P_1^T \\ v_i' P_3'^T - P_2'^T \\ u_i' P_3'^T - P_1'^T \end{bmatrix} \hat{X}_i = 0$$

$$\implies A\hat{X}_i = 0 \tag{3}$$

The above optimization is not geometrically meaningful. But, we can extend this method to multi camera setting:

Suppose, we have m cameras with projection matrices P, P', P''... and so on, and each of them is looking at a world point X_i, then we can naturally extend the above method as:

$$\begin{bmatrix} v_i P_3^T - P_2^T \\ u_i P_3^T - P_1^T \\ v_i' P_3'^T - P_2'^T \\ u_i' P_3'^T - P_1'^T \\ v_i'' P_3''^T - P_2''^T \\ u_i'' P_3''^T - P_1''^T \\ \vdots \end{bmatrix} \hat{X}_i = 0$$

where, x_i is the image of X_i in camera 1, and x_i' is the image of X_i in camera 2... and so on.

5 Non-Linear triangulation[3]

Following the same notations, and let

1. K and K' be the the intrinsic matrices associated with camera-1, 2.
2. r and r' be the direction vector from camera centers C and C' such that:

$$r = K^-1x$$
$$r' = R_2^1 K'^-1x'$$

3. l and l' be the lines from two camera centers and x and x'.

$$l = C + \lambda r$$
$$l' = C' + \mu r'$$

The next step is to find the points p and p' on l and l' such that the distance between those points is minimum. For the distance to be minimum we need the line pp' to be perpendicular to both lines l and l'.

$$(p - p').r = 0$$
$$(p - p').r' = 0$$

Replacing, equation of line l and l':

$$(C + \lambda r - (C' + \mu r')).r = 0$$
$$(C + \lambda r - (C' + \mu r')).r' = 0$$

The above equations can be written in matrix form as:

$$\begin{bmatrix} r.r & -r'.r \\ r.r' & -r'r' \end{bmatrix} \begin{bmatrix} \lambda \\ \mu \end{bmatrix} = \begin{bmatrix} (C - C').r \\ (C - C').r' \end{bmatrix}$$

$$\implies \begin{bmatrix} \lambda \\ \mu \end{bmatrix} = \begin{bmatrix} r.r & -r'.r \\ r.r' & -r'r' \end{bmatrix}^{-1} \begin{bmatrix} (C - C').r \\ (C - C').r' \end{bmatrix} \tag{4}$$

The last step is to find λ and μ, using equation 4. Then, using these we can find p and p'. Then,

$$\hat{X}_i = \frac{1}{2}(p + p')$$

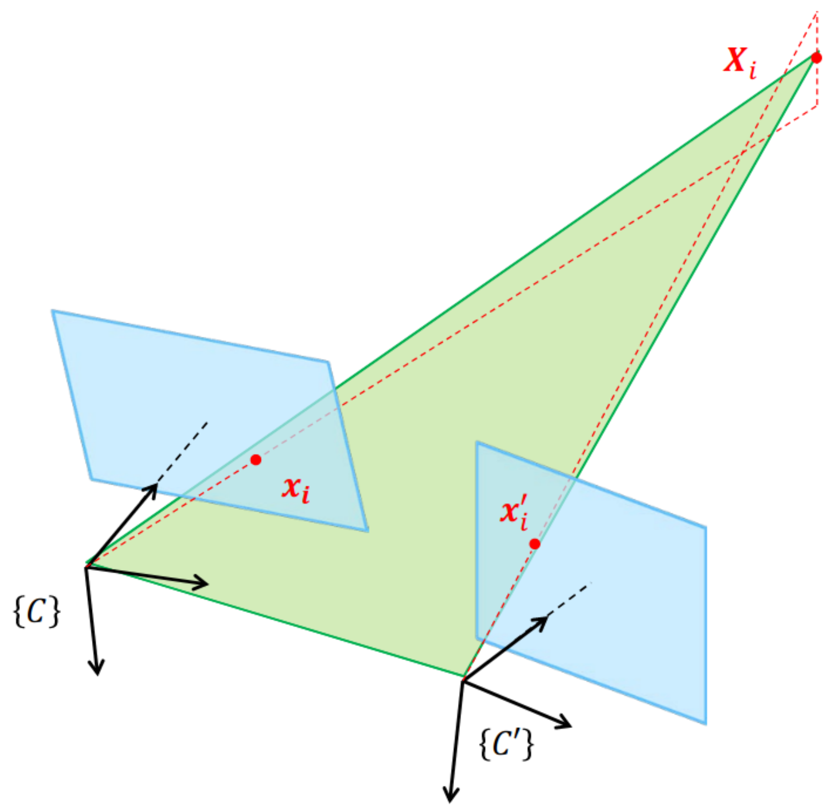

Figure 2: Non-Linear Triangulation

References

[1] Andrew Zisserman Richard Hartley. *Multiple View Geometry in computer vision.* Cambridge University Press, New York, 2009.

[2] Thomas Opsahl. Triangulation, 2009.

[3] Cyrill Stachniss. Triangulation, 2019.