

# Advanced Control Laboratory (085705)

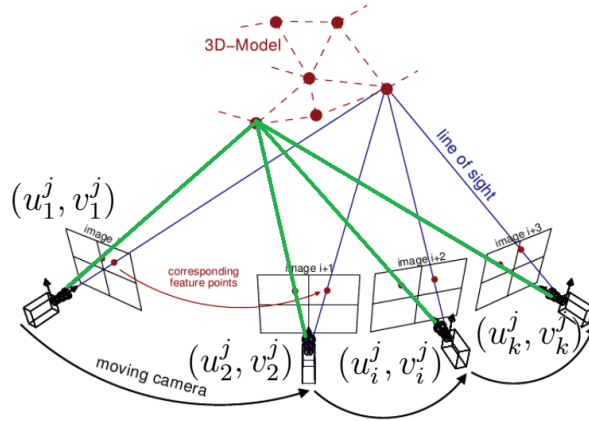
## pre-ROS #4: Triangulation & 3D Reconstruction

Daniel Engelsman 300546173

### Intuitive Intro

In previous lectures we've seen how to project a 3D landmark onto our image frame using calibration matrix ( $K$ ) and camera pose  $\pi(x, l)$ . This lecture we've learned how to use that process at a time, when the camera moves and augments frames (i). Each point in an image (i) corresponds to a line in 3D space, and the algorithm responsible for the correspondence was assumed in class as "black box" ( $p_i \leftrightarrow l_i$ ). In that pre-lab I will summarize the Triangulation process and show the algebraic tools.

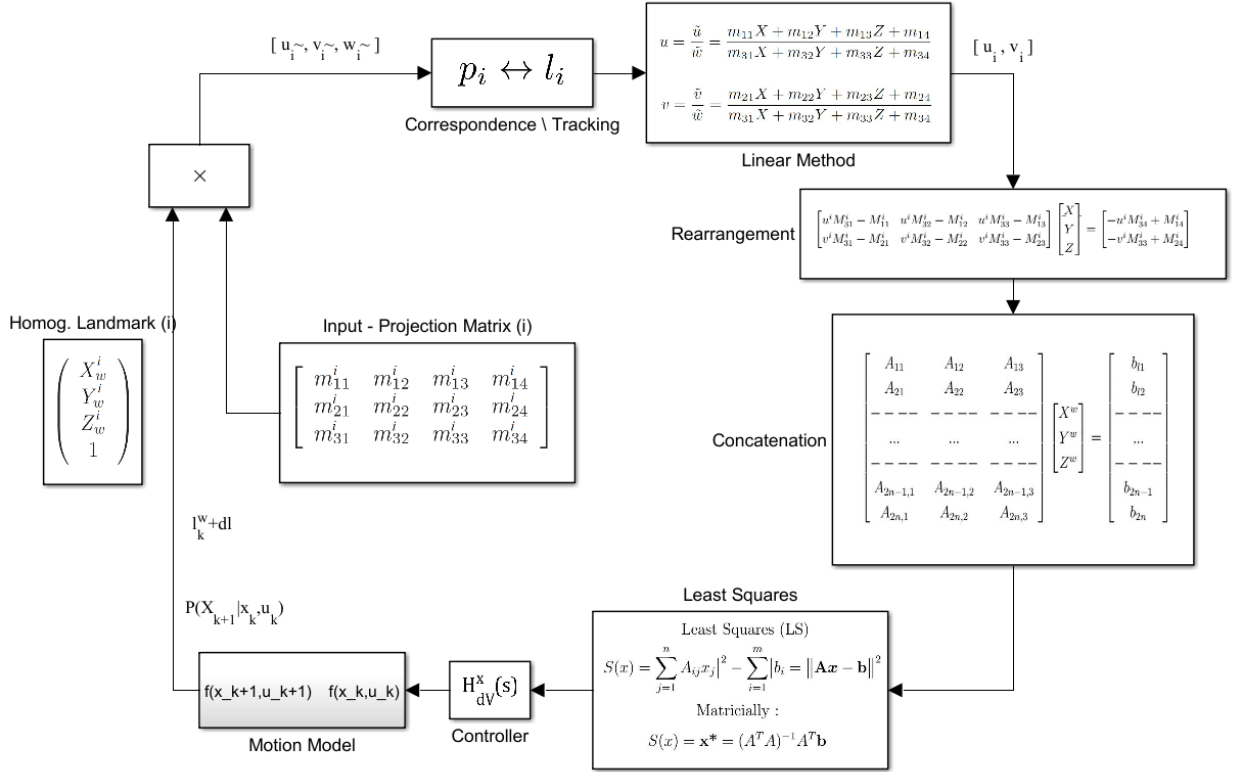
The following illustration shows the correspondence of one landmark to few different image frames via LOS. In computer vision, Triangulation refers to the process of determining a point in 3D space given its projections onto two, or more, images. In order to solve this problem it is necessary to know the parameters of the camera projection function from 3D to 2D for the cameras involved, in the simplest case represented by the camera matrices. Triangulation is sometimes also referred to as reconstruction.



# The Algorithm

Generally and Ideally<sup>[1]</sup> speaking, the process can be portrayed as :

## Triangulation & Reconstruction



The left part is the projection process. The upper part is the tracking algorithm<sup>[2]</sup> that is responsible on corresponding the landmarks between different images, and extracting the pixels<sup>[3]</sup>.

The right part is the rearrangement of the linear equations<sup>[4]</sup> and later on will be extracted via numeric methods. The lower part is the Least Squares method<sup>[5]</sup> (TBD later on), which ends up in a new motion step, that generates a new image frame<sup>[6]</sup>.

## Algebraic Formulation

Recalling that every  $(p_i \leftrightarrow l_i)$  generates a set of 2 equations  $(u, v)$ . The linearity of the problem allows us to rearrange the projection matrix matrixially, **once at a time**, and concatenate it.

$$\begin{bmatrix} u^i M_{31}^i - M_{11}^i & u^i M_{32}^i - M_{12}^i & u^i M_{33}^i - M_{13}^i \\ v^i M_{31}^i - M_{21}^i & v^i M_{32}^i - M_{22}^i & v^i M_{33}^i - M_{23}^i \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} -u^i M_{34}^i + M_{14}^i \\ -v^i M_{33}^i + M_{24}^i \end{bmatrix}$$

$$\text{Sizes : } (M_{jk}^i \in \mathbb{R}^{2 \times 3}) \quad \text{and} \quad (b_j^i \in \mathbb{R}^{2 \times 1})$$

Assume we took n images along motion, we would like to "store" it aside efficiently. Therefore, the general augmentation matrix can be denoted as :

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ \text{---} & \text{---} & \text{---} \\ \dots & \dots & \dots \\ \text{---} & \text{---} & \text{---} \\ A_{2n-1,1} & A_{2n-1,2} & A_{2n-1,3} \\ A_{2n,1} & A_{2n,2} & A_{2n,3} \end{bmatrix} \begin{bmatrix} X^w \\ Y^w \\ Z^w \end{bmatrix} = \begin{bmatrix} b_{l1} \\ b_{l2} \\ \text{---} \\ \dots \\ \text{---} \\ b_{2n-1} \\ b_{2n} \end{bmatrix}$$

$$\text{Sizes : } (A \in \mathbb{R}^{2n \times 3}) \quad \text{and} \quad (b \in \mathbb{R}^{2n \times 1})$$

One can see that the total number of images satisfies  $(n \gg 3)$ , allowing us to obtain in some of the images better predictions. Therefore we would like to make an "average" of optimization that will reduce the error as much as possible. This is where the **Least Square** Method becomes handy :

Simple solution of a linear set can be denoted as :

Least Squares (LS)

$$S(x) = \sum_{j=1}^n |A_{ij}x_j|^2 - \sum_{i=1}^m |b_i| = \|\mathbf{Ax} - \mathbf{b}\|^2$$

Here however, we are working with the aforementioned matrices that force us to compute it using matrix representation :

$$S(x) = \mathbf{x}^* = (A^T A)^{-1} A^T \mathbf{b}$$

## Assumptions made:

- [1] Ideal process - No external and internal noises.
- [2] Correspondence / Tracking Algorithm is already "given"  $\Rightarrow$  no need to know 3D coordinates of landmark.
- [3] Ideal Recovery -  $(u, v)^T = \pi(x, l)$ .
- [4] Set of Linear Equations  $\Rightarrow$  allows using Linear Method.
- [5] No computational constraints nor storage limit.
- [6] Reasonable motion rate  $\Rightarrow$  Tracking is continuous.