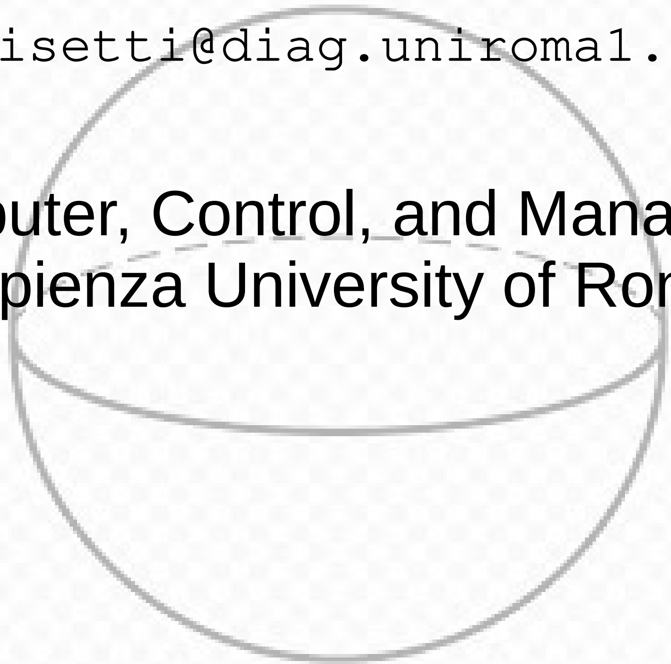
Probabilistic Robotics Course

Mini Recap of Epipolar Geometry on Unit Sphere

Giorgio Grisetti

grisetti@diag.uniroma1.it

Department of Computer, Control, and Management Engineering
Sapienza University of Rome



Courtesy of Li et. al

Lines

Line in the space described as an offset **p**, and a direction **d**. The points in the line are spanned by scalar ascissa *s*.

$$\mathbf{l}_0 = \mathbf{p}_0 + \mathbf{d}_0 s_0$$

Transforming a line by T: R,t, transforms the point and rotates the direction:

$$\mathbf{T} = [\mathbf{R}\ \mathbf{t}]$$

$$\mathbf{l}_1 = \mathbf{T} \cdot \mathbf{l}_0 \ = \ \underbrace{\mathbf{R}\mathbf{p}_0 + \mathbf{t}}_{\mathbf{p}_1} + \underbrace{\mathbf{R}\mathbf{d}_0}_{\mathbf{d}_1} s_0$$

Lines, closest point

Lines in the space do not always intersect

Given two lines, can find the closest point by looking for the pair of ***s*** for which the corresponding points in the line are the closest

$$\mathbf{\Delta}(s_0, s_1) = \mathbf{p}_0 + \mathbf{d}_0 s_0 - \mathbf{p}_1 - \mathbf{d}_1 s_1$$

$$= \underbrace{(\mathbf{p}_0 - \mathbf{p}_1)}_{\mathbf{\Delta p}} + \underbrace{(\mathbf{d}_0 \mid -\mathbf{d}_1)}_{\mathbf{\Delta D}} \underbrace{\begin{pmatrix} s_0 \\ s_1 \end{pmatrix}}_{\mathbf{s}}$$

$$\mathbf{s}^* = \underset{\mathbf{s}}{\operatorname{argmin}} \|\mathbf{\Delta p} + \mathbf{\Delta D s}\|^2$$

$$= -(\mathbf{\Delta D}^T \mathbf{\Delta D})^{-1} \mathbf{\Delta D}^T \mathbf{\Delta p}$$

Line Intersection

Two lines

$$\mathbf{l}_0 : \mathbf{p}_0, \mathbf{d}_0$$

$$\mathbf{l}_1 : \mathbf{p}_1, \mathbf{d}_1$$

The intersection condition specifies that the projection of

- the difference vector between any two points in the line (so we choose **p0** and **p1**)
- along the vector **d0 x d1** orthogonal to both lines

should be 0.

$$0 = (\mathbf{p}_0 - \mathbf{p}_1) \cdot (\mathbf{d}_0 \times \mathbf{d}_1)$$

Line Intersection

Both lines pass through a camera center in **p=0**

$$\mathbf{l}_0 : \mathbf{p}_0 = 0, \mathbf{d}_0$$

$$\mathbf{l}_1 : \mathbf{p}_1 = 0, \mathbf{d}_1$$

The second camera is at position **T = [R,t]**

$$\text{l'}_1 : \mathbf{t}, \mathbf{R}\mathbf{d}_1$$

$$0 = \mathbf{t} \cdot (\mathbf{R}\mathbf{d}_1 \times \mathbf{d}_0)$$

Intersection condition

$$= \mathbf{d}_0 \cdot (\mathbf{t} \times (\mathbf{R}\mathbf{d}_1))$$

cross prod. identity $\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c}) = \mathbf{c} \cdot (\mathbf{a} \times \mathbf{b})$

$$= \mathbf{d}_0 \cdot \lfloor \mathbf{t} \rfloor_\times \mathbf{R}\mathbf{d}_1$$

cross prod. to skew

$$= \mathbf{d}_1^T \underbrace{\left(\mathbf{R}^T \lfloor \mathbf{t} \rfloor_\times\right)}_{\mathbf{E}} \mathbf{d}_0$$

transpose and negate

$$0 = \mathbf{d}_1 \mathbf{E} \mathbf{d}_0$$

Epipolar Contstraint

Extracting R and t from E

E contains both **R** and **t**. To get them we need to decompose **E** in the product of a rotation matrix and a skew symmetric matrix.

Done by:

- Computing SVD
- Using the rearraging matrix **W**
- Extract **R** and t by identification
- 4 solutions for: **t**, -**t**, **W** and **W**^{T}
- Choose one by triangulating the points and pick the one with most points in front of camera

$$\mathbf{E} = \mathbf{R}^T \lfloor \mathbf{t} \rfloor_\times \quad \mathrm{rank}(\mathbf{E}) \leq 2$$

$$= \mathbf{USV}^T \, \mathrm{s.t.} \mathbf{S} = \begin{pmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & 0 \end{pmatrix}$$

$$\mathbf{W} = \begin{pmatrix} & 1 & \\ -1 & & \\ & & 1 \end{pmatrix}$$

$$\mathbf{WS} = \begin{pmatrix} & \sigma_2 & \\ -\sigma_1 & & \\ & & 0 \end{pmatrix}$$

$$\mathbf{WSW}^T = \mathbf{S}$$

$$\mathbf{E} = \mathbf{UWSW}^T\mathbf{V}^T$$

$$= \underbrace{\mathbf{UWV}^T}_{\mathbf{R}} \underbrace{\mathbf{VSW}^T\mathbf{V}^T}_{\lfloor \mathbf{t} \rfloor_\times}$$

Estimating E (8 point)

Find the matrix that better satisfies the intersection constraint between all correspondences

$$\mathbf{E} = \begin{pmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{pmatrix} \qquad \mathbf{e} = \begin{pmatrix} e_{11} \\ e_{12} \\ \vdots \\ e_{33} \end{pmatrix}$$

$$\mathbf{d}'^T \mathbf{E} \mathbf{d} = 0$$

$$\mathbf{e}^* = \operatorname*{argmin}_{\mathbf{e}} \sum_i \| \mathbf{d}'^{[i]\,T} \mathbf{E} \mathbf{d}^{[i]} \|^2$$

$$\mathbf{E} = \mathbf{0} \;\; \text{trivial solution, enforce } \|\mathbf{e}\| = 1$$

Estimating E (8 point)

$$\mathbf{d}'^T\mathbf{E}\mathbf{d} = \underbrace{(x'x \;\; x'y \;\; x'z \;\; y'x \;\; y'y \;\; y'z \;\; z'x \;\; z'y \;\; z'z)}_{\mathbf{A}} \underbrace{\begin{pmatrix} e_{11} \\ e_{12} \\ \vdots \\ e_{33} \end{pmatrix}}_{\mathbf{e}}$$

$$\mathbf{e}^* = \operatorname*{argmin}_{\mathbf{e}} \sum_i \mathbf{e}^T \mathbf{A}^{[i]T} \mathbf{A}^{[i]} \mathbf{e}$$

$$= \operatorname*{argmin}_{\mathbf{e}} \mathbf{e}^T \underbrace{\left(\sum_i \mathbf{A}^{[i]T} \mathbf{A}^{[i]} \right)}_{\mathbf{H}} \mathbf{e}$$

$$\text{s.t} \quad \mathbf{e}^T\mathbf{e} = 1$$

Estimating E (8 point)

Constrained minimization using Lagrange multipliers

$$\mathcal{L}(\mathbf{e}, \lambda) = \mathbf{e}^T \mathbf{H} \mathbf{e} - \lambda(\mathbf{e}^T \mathbf{e} - 1)$$

$$\frac{\partial \mathcal{L}(\mathbf{e}, \lambda)}{\partial \mathbf{e}} = 2\mathbf{H}\mathbf{e} - 2\lambda\mathbf{e} = \mathbf{0}$$

$$\frac{\partial \mathcal{L}(\mathbf{e}, \lambda)}{\partial \lambda} = \mathbf{e}^T \mathbf{e} - 1 = \mathbf{0}$$

$$\mathbf{H}\mathbf{e} = \lambda\mathbf{e} \quad \mathbf{e} : \text{is an eigenvector of } \mathbf{H}$$

$$\mathbf{e}_j^T \mathbf{H} \mathbf{e}_j = \mathbf{e}_j^T \lambda_j \mathbf{e}_j = \lambda_j \text{ cost}$$

Considerations

Using the 8 points algorithm, we need 8 correspondences.

- The E matrix has only 5 DOF.
- There are more complex solutions that require less points (7, 6 and 5)to compute E
- In fact the algorithm is commonly used to estimate the Fundamental matrix **F**, that incorporates also the camera matrix **K**

How to get T?

From a set of points in the image compute the direction vectors (or something proportional to them), by undoing the effects of the camera matrix **K**

Estimate the essential from these directions

Compute the 4 solutions for R and t and for each triangulate the points. Choose the solution with most admissible points

- In front of camera
- In camera frustum

Concusions

Short recap of how to get a reconstruction (up to a scale) from a set of directions.

Improvements include

- Use an algorithm requiring less correspondences to compute **E**
- Normalize the points in the image plane to get a better conditioned system

This lesson does not substitute in any way a proper computer vision course. Its main purpose is to provide those students that had no contact with the subject with some basic knowledge within one hour.