Université Grenoble Alpes

3D Vision – Geometry2

Edmond Boyer

Université Grenoble Alpes

Contents

1. Panoramic mosaics: Stiching images into panoramas.
2. Two view geometry: point reconstruction, epipolar geometry.

Panoramic Mosaics

Images and panaroma taken from the Mathworks documentation website

Panoramic Mosaics

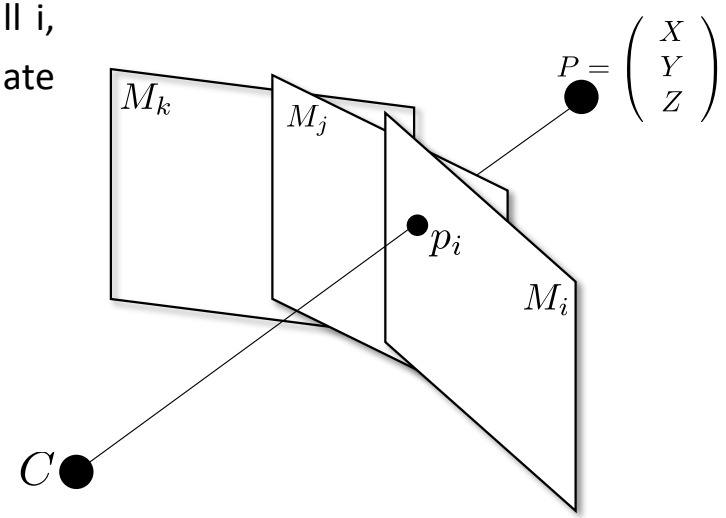
The 3x4 projection matrix for an image is: $M \sim [K \cdot R \quad K \cdot T]$.

When several images *i* are taken from the same viewpoint, i.e. T_i=T for all i, and assuming the world coordinate frame center to be the camera coordinate frame center ($T=(0,0,0)^t$) :

$$M_i \sim [K \cdot R_i \quad K \cdot (T_i = 0)] \sim [K \cdot R_i \quad 0].$$

Consider the 3D point $P = (X, Y, Z, 1)^t$, it projects onto p_i in image i such that:

$$M_i \cdot P \sim p_i \sim [K \cdot R_i \quad 0] \cdot \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \sim K \cdot R_i \cdot \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$



Panoramic Mosaics

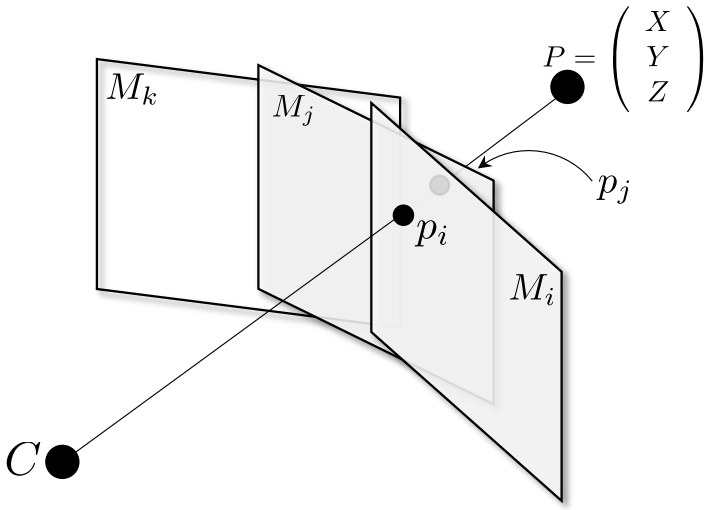
And its projection in image *j:*

$$p_j \sim M_j \cdot P \sim K \cdot R_j \cdot \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

$$p_j \sim K \cdot R_j \cdot (K \cdot R_i)^{-1} \cdot (K \cdot R_i) \cdot \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

$$p_j \sim K \cdot R_j \cdot (K \cdot R_i)^{-1} \cdot p_i$$

$$p_j \sim K \cdot R_j \cdot R_i^t \cdot K^{-1} \cdot p_i$$



Panoramic Mosaics

And its projection in image *j:*

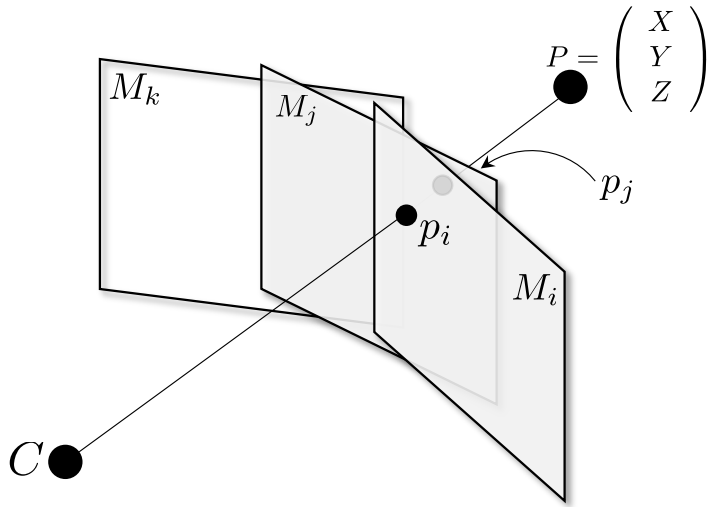
$$p_j \sim M_j \cdot P \sim K \cdot R_j \cdot \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

$$p_j \sim K \cdot R_j \cdot (K \cdot R_i)^{-1} \cdot (K \cdot R_i) \cdot \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

$$p_j \sim K \cdot R_j \cdot (K \cdot R_i)^{-1} \cdot p_i$$

$$p_j \sim K \cdot R_j \cdot R_i^t \cdot K^{-1} \cdot p_i$$

Homography or projective transformation in the image plane



Panoramic Mosaics

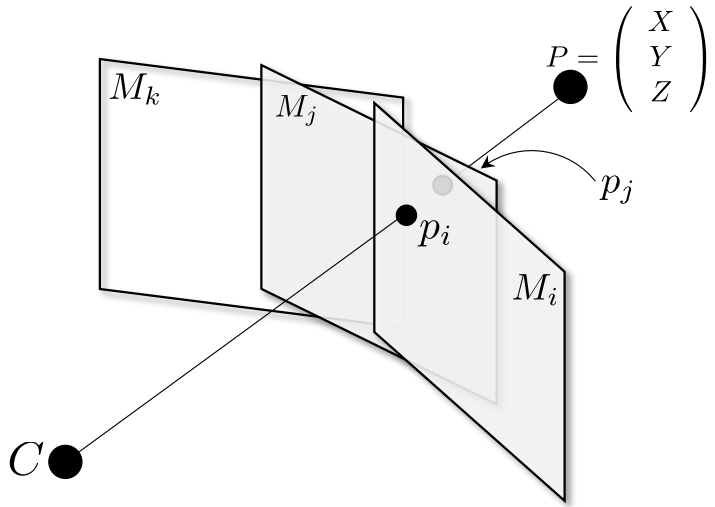
And its projection in image *j:*

$$p_j \sim M_j \cdot P \sim K \cdot R_j \cdot \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

$$p_j \sim K \cdot R_j \cdot (K \cdot R_i)^{-1} \cdot (K \cdot R_i) \cdot \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

$$p_j \sim K \cdot R_j \cdot (K \cdot R_i)^{-1} \cdot p_i$$

$$p_j \sim K \cdot R_j \cdot R_i^t \cdot K^{-1} \cdot p_i$$



Homography or projective transformation in the image plane

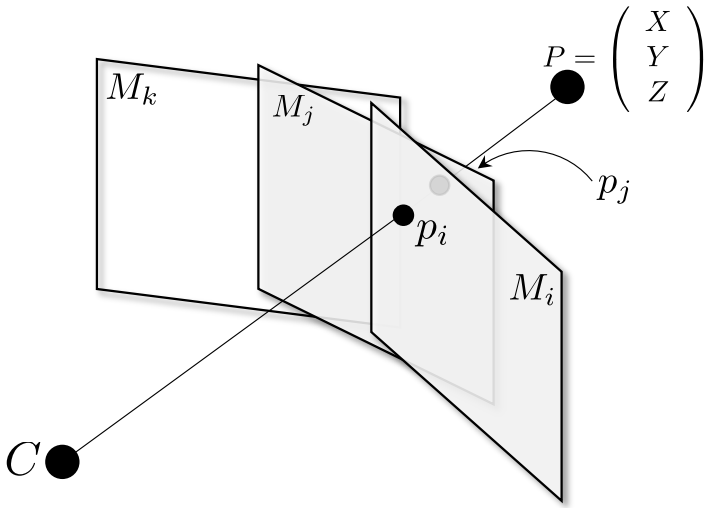
Panoramic Mosaics

Therefore the homography H between 2 image projections *i* and *j* when the camera is in rotation around the projection center is:

$$H_{ji} \sim K \cdot R_j \cdot R_i^t \cdot K^{-1}$$

Relative rotation between image plane *i* and *j*

$$H_{ij} \sim H_{ji}^{-1} \sim K \cdot R_i \cdot R_j^t \cdot K^{-1}$$



Panoramic Mosaics

Estimating the homography:

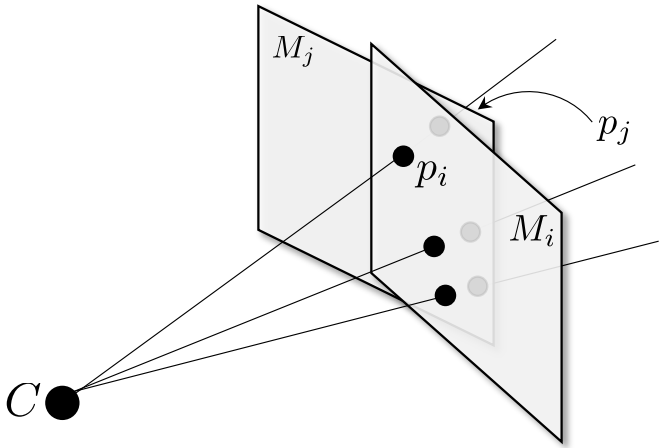
1. Find correspondences between images *i* and *j* using e.g. image features
2. For each corresponding pair, the homogeneous equation is :

$$p_j \sim H_{ji} \cdot p_i$$

$$\begin{pmatrix} u_j \\ v_j \\ 1 \end{pmatrix} \sim H_{ji} \cdot \begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix}$$

And the inhomogeneous form is:

$$\begin{pmatrix} \gamma u_j \\ \gamma v_j \\ \gamma \end{pmatrix} = H_{ji} \cdot \begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix}$$



Panoramic Mosaics

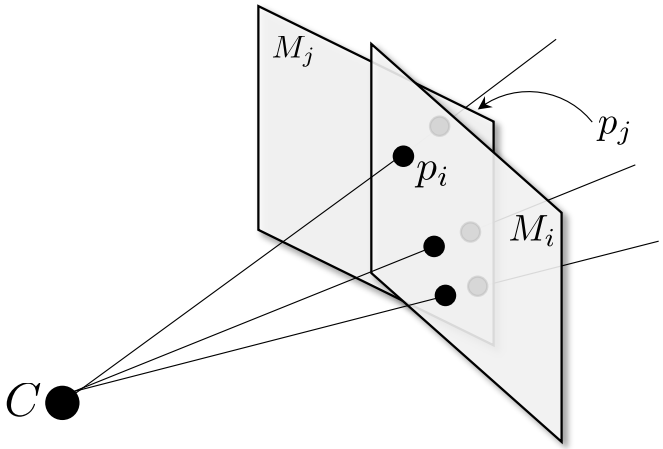
Estimating the homography:

2. For each corresponding pair :

$$\begin{pmatrix} \gamma u_j \\ \gamma v_j \\ \gamma \end{pmatrix} = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix} \cdot \begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix}$$

$$\iff \begin{cases} \gamma u_j = h_1 u_i + h_2 v_i + h_3 \\ \gamma v_j = h_4 u_i + h_5 v_i + h_6 \\ \gamma = h_7 u_i + h_8 v_i + h_9 \end{cases}$$

$$\iff \begin{cases} u_i u_j h_7 + v_i u_j h_8 + u_j h_9 = u_i h_1 + v_i h_2 + h_3 \\ u_i v_j h_7 + v_i v_j h_8 + v_j h_9 = u_i h_4 + v_i h_5 + h_6 \end{cases}$$



Panoramic Mosaics

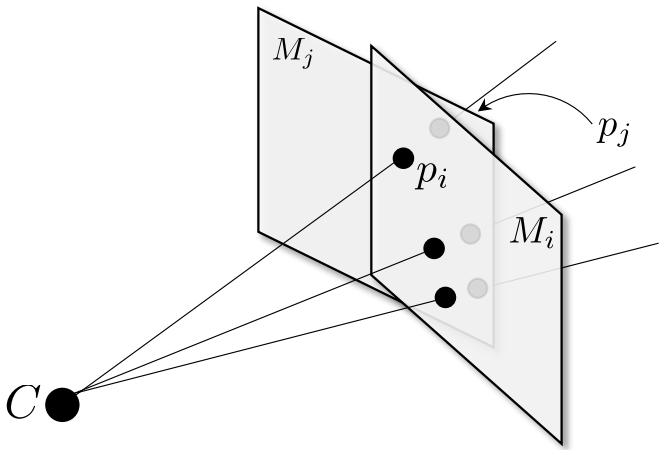
Estimating the homography:

2. For each corresponding pair :

$$\iff \begin{pmatrix} u_i & v_i & 1 & 0 & 0 & 0 & -u_iu_j & -v_iu_j & -u_j \\ 0 & 0 & 0 & u_i & v_i & 1 & -u_iv_j & -v_iv_j & -v_j \end{pmatrix} \cdot \begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \end{pmatrix} = 0$$

2. For n pairs, n ≥ 4 (least squares when n > 4):

$$\iff \begin{pmatrix} u_1 & v_1 & 1 & 0 & 0 & 0 & -u_1u_2 & -v_1u_2 & -u_2 \\ 0 & 0 & 0 & u_1 & v_1 & 1 & -u_1v_2 & -v_1v_2 & -v_2 \\ . & . & . & . & . & . & . & . & . \\ & & & & . & & & & \\ & & & & . & & & & \\ u_i & v_i & 1 & 0 & 0 & 0 & -u_iu_j & -v_iu_j & -u_j \\ 0 & 0 & 0 & u_i & v_i & 1 & -u_iv_j & -v_iv_j & -v_j \end{pmatrix} \cdot \begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \end{pmatrix} = 0$$



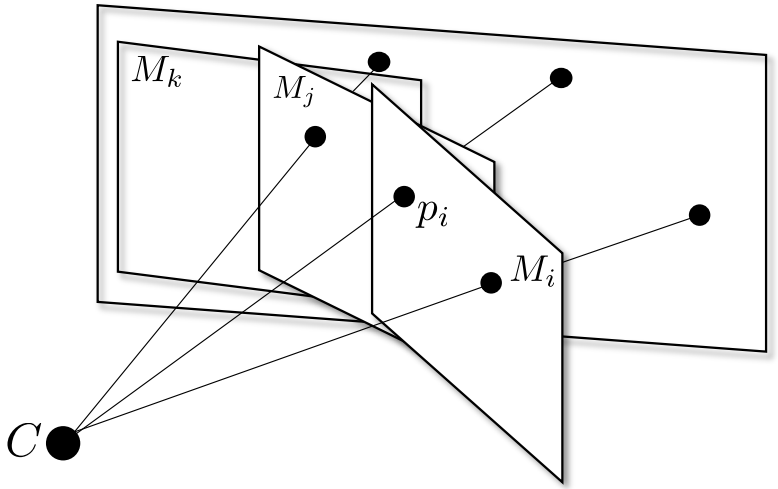
Panoramic Mosaics

Application:

Given n images, we can remap all of them onto one image plane by combining transformations:

$$H_{ki} \sim H_{kj} \cdot H_{ji}$$

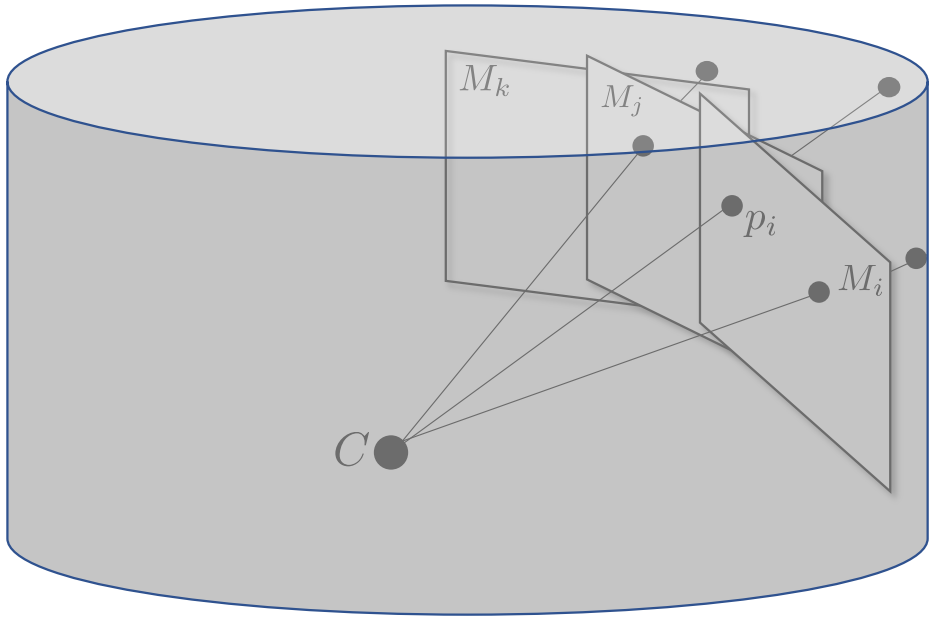
- Useful since matched pairs may not exist between images *k* and *i.*
- Note that in addition to this geometric mapping, a photometric mapping must be applied to make colors consistent.
- Artefacts are usually still present as a result of geometric errors: matching, parallax (shift in apparent object positions) with camera motion not exactly rotation, etc).
- We do not account for camera distorsions in the geometric model.



Panoramic Mosaics

Application:

Mapping onto a cylinder around *C:* use matrix K to find consistent directions for viewing rays in all images and then mapped pixels onto the cylinder using the viewing ray directions.



Two View Geometry

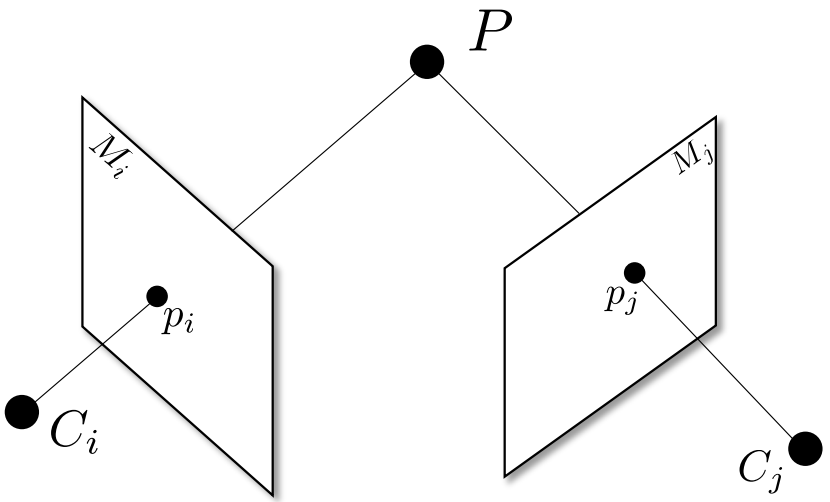
Reconstruction:

Given 2 calibrated image projections of the same 3D point *P* we can, in principle, estimate its position by triangulation: *P* is at the intersection of the viewing lines from the 2 projections centres.

In practice, 3D lines do not intersect since geometry (camera parameters, image point locations, matching) is not exact.

Solutions:

1. Geometric approach: Find the closest point to the 2 viewing lines.



Two View Geometry

Reconstruction:

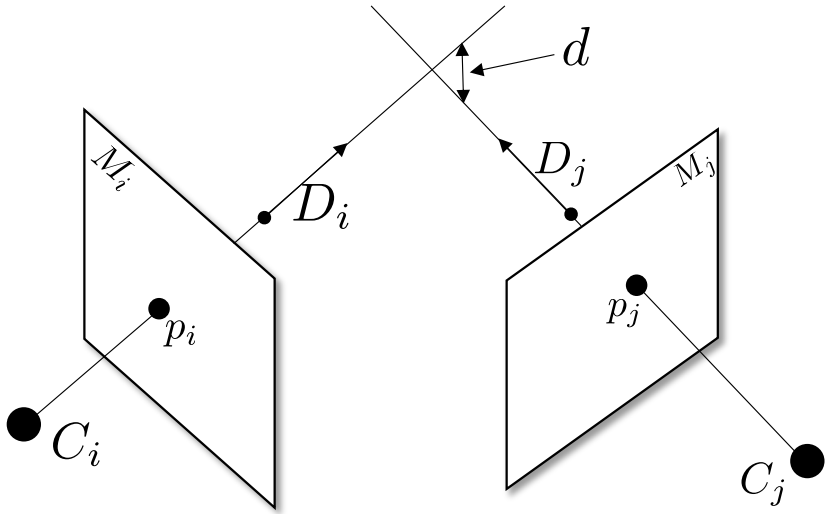
1. Geometric approach: Find the closest point to the 2 viewing lines:

$$\begin{cases} (X,Y,Z)^t = C_i + \lambda_i D_i + \frac{d}{2} D_j \times D_i, \\ (X,Y,Z)^t = C_j + \lambda_j D_j - \frac{d}{2} D_j \times D_i. \end{cases}$$

$$\iff \quad C_i - C_j + \lambda_i D_i - \lambda_j D_j + d(D_j \times D_i) = 0,$$

-> 3 equations in 3 unknowns $(\lambda_i, \lambda_j, d)$.

1. We can solve for $(\lambda_i, \lambda_j, d)$.
2. Geometric reasoning gives also closed form solutions.



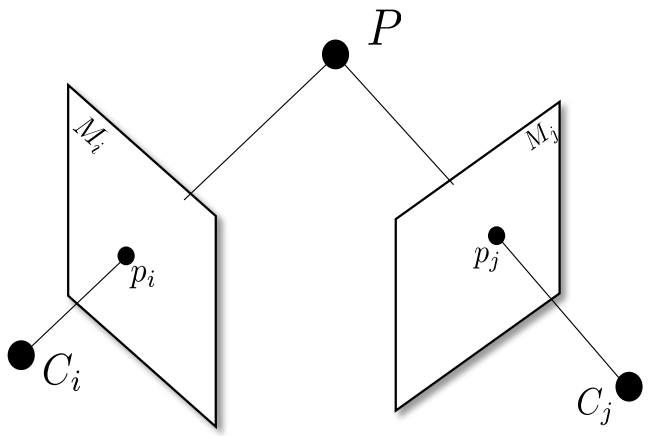
Two View Geometry

Reconstruction:

2. Algebraic approach: each image point gives 2 independent equations in the coordinates of *P* (X,Y,Z)*:*

$$\begin{pmatrix} \gamma_i u_i \\ \gamma_i v_i \\ \gamma_i \end{pmatrix} = \begin{pmatrix} m_1^i & m_2^i & m_3^i & m_4^i \\ m_5^i & m_6^i & m_7^i & m_8^i \\ m_9^i & m_{10}^i & m_{11}^i & m_{12}^i \end{pmatrix} \cdot \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

$$\iff \begin{cases} (m_9^i X + m_{10}^i Y + m_{11}^i Z + m_{12}^i) u_i = m_1^i X + m_2^i Y + m_3^i Z + m_4^i \\ (m_9^i X + m_{10}^i Y + m_{11}^i Z + m_{12}^i) v_i = m_5^i X + m_6^i Y + m_7^i Z + m_8^i \end{cases}$$



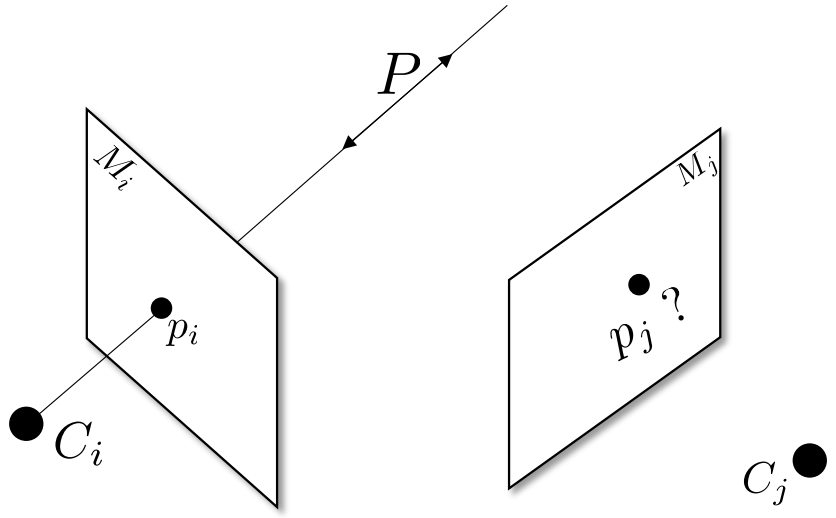
And with 2 or more images, we obtain an overdetermined system of equations that can be solved with the least squares:

$$\begin{cases} (m_9^i X + m_{10}^i Y + m_{11}^i Z + m_{12}^i) u_i = m_1^i X + m_2^i Y + m_3^i Z + m_4^i \\ (m_9^i X + m_{10}^i Y + m_{11}^i Z + m_{12}^i) v_i = m_5^i X + m_6^i Y + m_7^i Z + m_8^i \\ (m_9^j X + m_{10}^j Y + m_{11}^j Z + m_{12}^j) u_i = m_1^j X + m_2^j Y + m_3^j Z + m_4^j \\ (m_9^j X + m_{10}^j Y + m_{11}^j Z + m_{12}^j) v_i = m_5^j X + m_6^j Y + m_7^j Z + m_8^j \end{cases}$$

Two View Geometry

Epipolar Geometry: link between two image projections of the same 3D point.

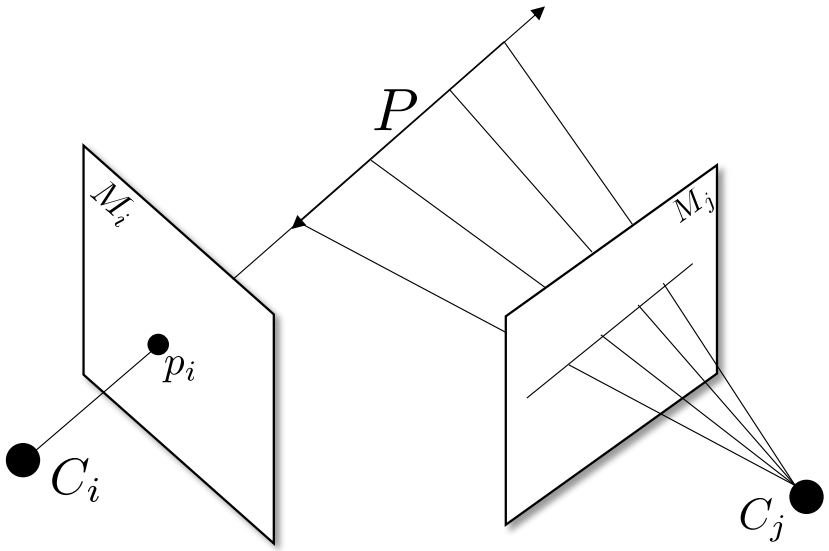
Given a point in an image, the source 3D point lies along the viewing line and the corresponding point in another image should satisfy a geometric constraint



Two View Geometry

Epipolar Geometry: link between two image projections of the same 3D point.

The constraint, called the *epipolar constraint*, is that the correspondent lies on a line, the epipolar line, in the second image.



Two View Geometry

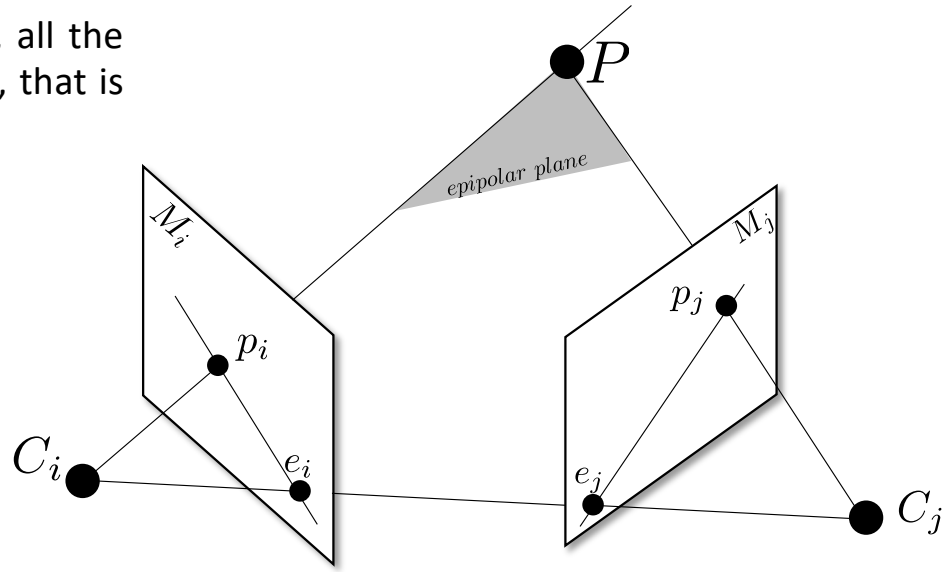
Epipolar Geometry

- The epipolar constraint is symmetric.
- Since the epipolar line is the projection of a viewing ray, all the epilolar lines in an image go through a point, the epipole, that is the projection of the other image center:

$$e_j \sim M_j \cdot \begin{pmatrix} C_i \\ 1 \end{pmatrix},$$

$$e_i \sim M_i \cdot \begin{pmatrix} C_j \\ 1 \end{pmatrix}.$$

Where C_i, C_j are 3x1 coordinate vectors.



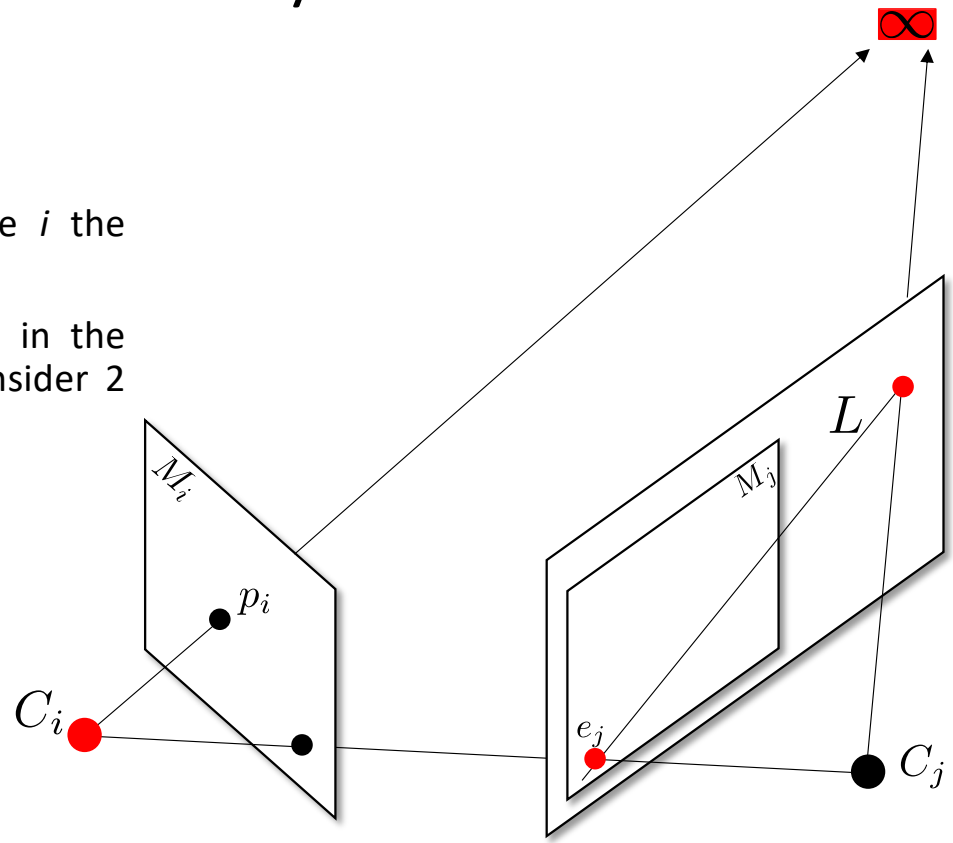
Two View Geometry

Epipolar Geometry: Algebraic constraint

- The epipolar constraint is that given a point in image *i* the correspondent in image *j* lies on a line *L.*
- To compute the homogeneous coordinates of this line in the image plane, given the projection matrices, we will consider 2 particular points along the viewing ray and project them:

1. C_i that projects onto $e_j \sim M_j \cdot \begin{pmatrix} C_i \\ 1 \end{pmatrix}$

2. The point at infinity along the viewing ray that can be estimated for any image point.



Two View Geometry

Epipolar Geometry: Algebraic constraint

1. C_i that projects onto $e_j \sim M_j \cdot \begin{pmatrix} C_i \\ 1 \end{pmatrix}$

By construction: $M_i \cdot \begin{pmatrix} C_i \\ 1 \end{pmatrix} = 0$

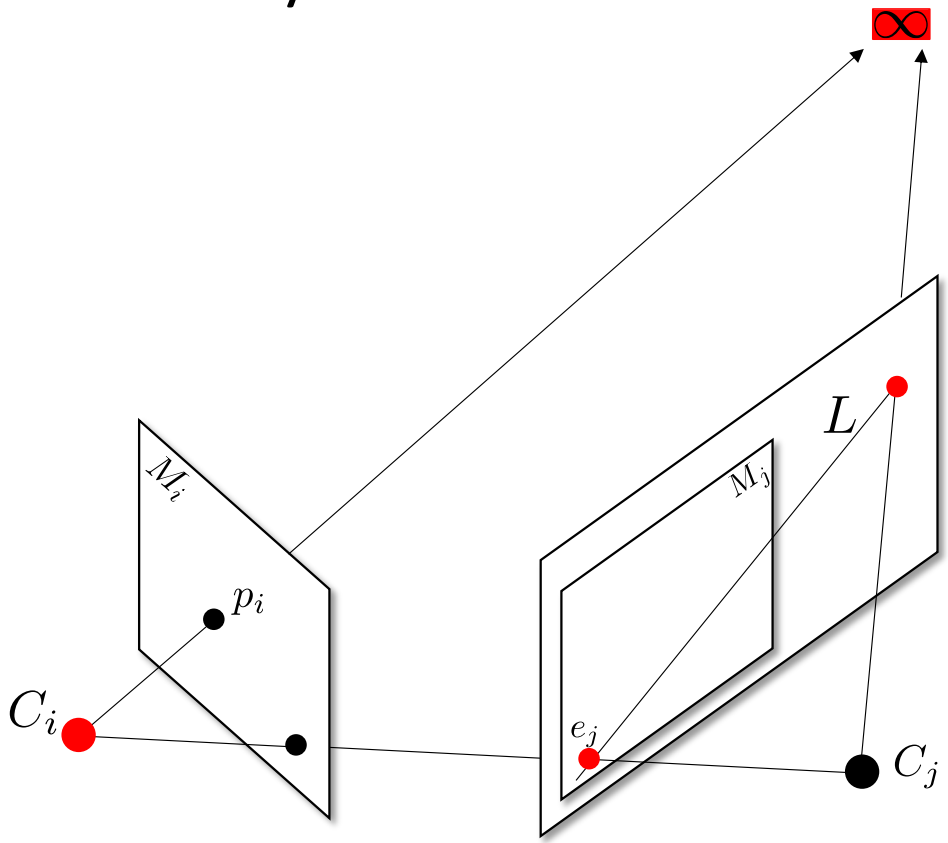
$$(\overline{M_i}\ m_i) \cdot \begin{pmatrix} C_i \\ 1 \end{pmatrix} = 0.$$

Where: $\overline{M_i}$ are the first 3 columns of M_i

and m_i is the last column of M_i

Thus: $C_i = -\overline{M_i}^{-1} \cdot m_i$ and $C_j = -\overline{M_j}^{-1} \cdot m_j$.

Hence: $e_j \sim \overline{M_j} \cdot C_i + m_j \sim \overline{M_j} \cdot (C_i - C_j)$



Two View Geometry

Epipolar Geometry: Algebraic constraint

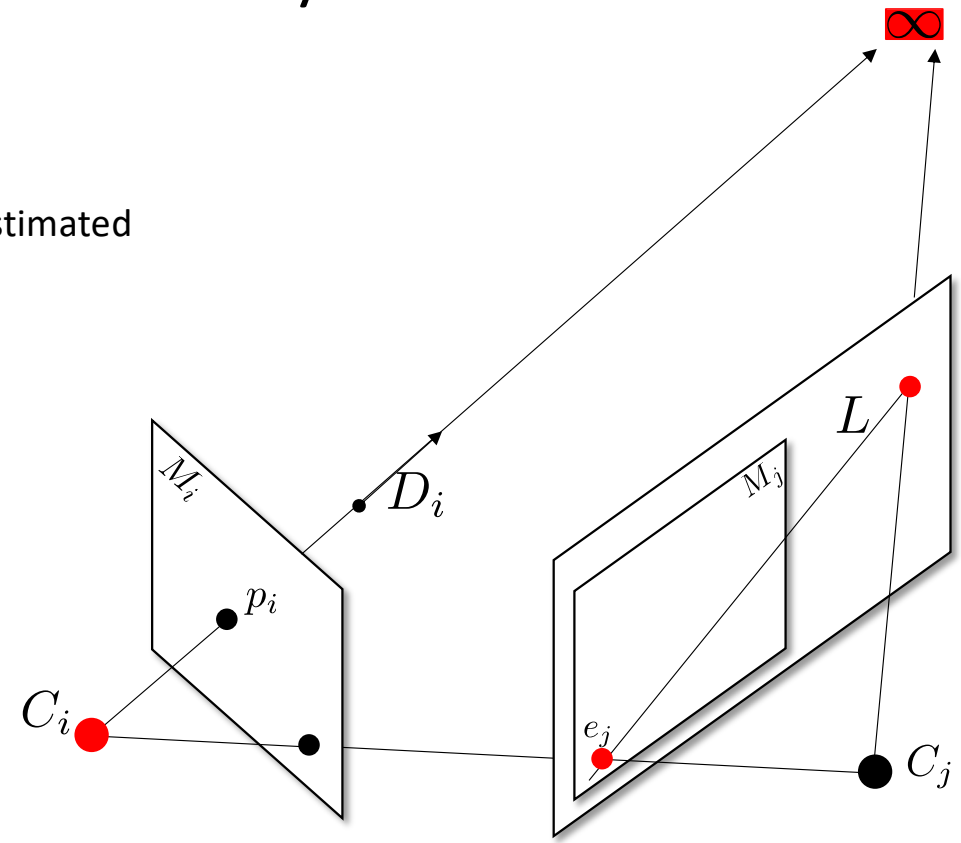
2. The point at infinity along the viewing ray that can be estimated for any image point:

$$p_i \sim M_i \cdot P \sim (\overline{M_i}\ m_i) \cdot P,$$

$$p_i \sim (\overline{M_i}\ m_i) \cdot \begin{pmatrix} D_i \\ 0 \end{pmatrix} \sim \overline{M_i} \cdot D_i.$$

Thus: $D_i \sim \overline{M_i}^{-1} \cdot p_i.$

And its projection in image *j* is: $\overline{M_j} \cdot \overline{M_i}^{-1} \cdot p_i$



Two View Geometry

Epipolar Geometry: Algebraic constraint

The epipolar line *L* in image *j* is by construction:

$$L \sim e_j \times \overline{M_j} \cdot \overline{M_i}^{-1} \cdot p_i$$
$$L \sim \overline{M_j} \cdot (C_i - C_j) \times \overline{M_j} \cdot \overline{M_i}^{-1} \cdot p_i$$

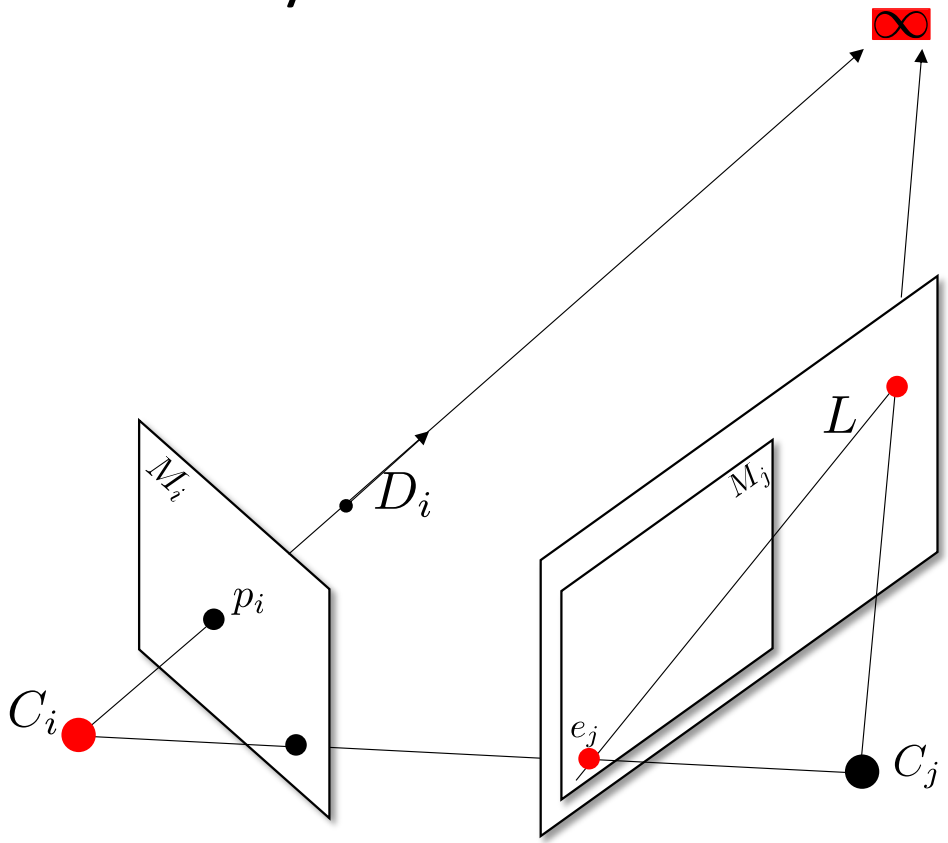
Given that: $M \cdot X \times M \cdot Y \sim M^{-t} \cdot (X \times Y)$,

$$L \sim \overline{M_j}^{-t} \cdot (C_i - C_j) \times \overline{M_i}^{-1} \cdot p_i$$

The above expression is linear and can be rewritten as:

$$L \sim F_{ji} \cdot p_i .$$

Where *F* is a 3x3 matrix called the fundamental matrix.



Two View Geometry

Epipolar Geometry: Algebraic constraint

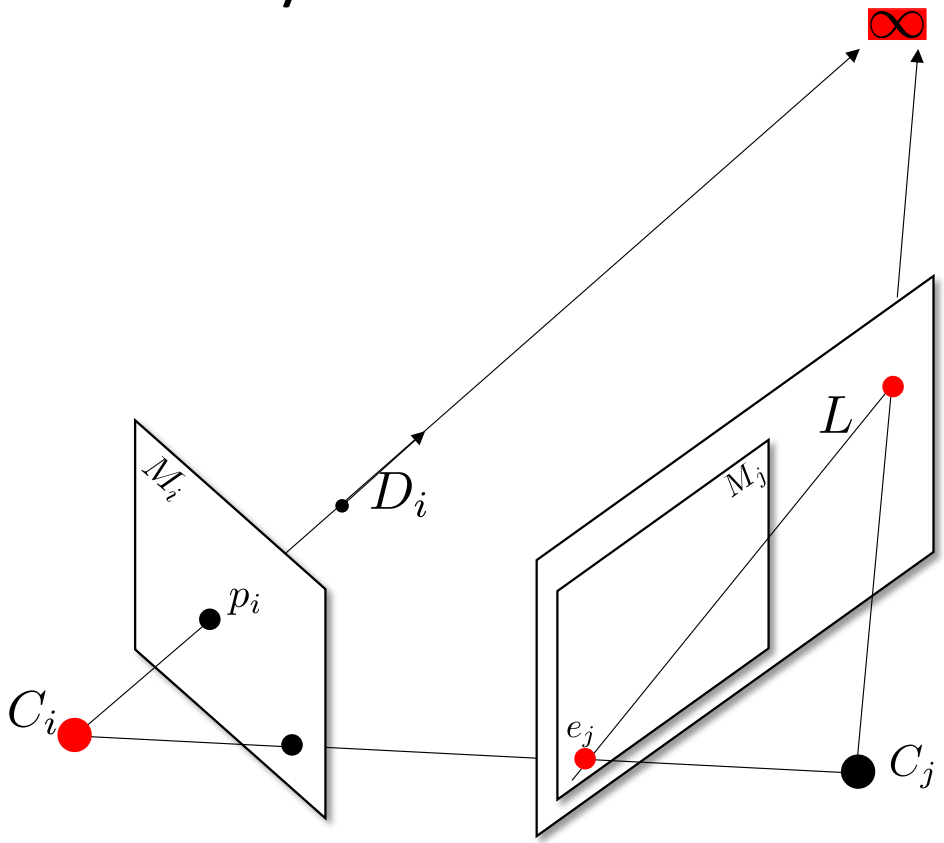
The *epipolar constraint* between 2 corresponding points is then:

$$p_j^t \cdot F_{ji} \cdot p_i = 0.$$

- Symmetry: Taking the transpose

$$(p_j^t \cdot F_{ji} \cdot p_i)^t = p_i^t \cdot F_{ji}^t \cdot p_j = 0$$

where $F_{ji}^t = F_{ij}$ is the fundamental matrix defining epipolar lines in image *i* for points in image *j*.



Two View Geometry

Epipolar Geometry: Fundamental matrix

- The fundamental matrix captures all the geometric information between 2 images.
- It is a singular matrix, i.e. its determinant is zero.
  - It maps point to lines, hence it is one-to-many.
  - It has a kernel: $F_{ij} \cdot e_j = 0$ and

$$F_{ij}^t \cdot e_i = F_{ji} \cdot e_i = 0$$

  - In practice it has rank 2.

- 8, or more, correspondences are required to estimate the fundamental matrix. However, without any additional constraint, the estimated matrix will not be singular.

Two View Geometry

Epipolar Geometry: The calibrated case with the Essential matrix

- Given the calibration $K_i, K_j,$ we can rewrite the epipolar line in the calibrated case:

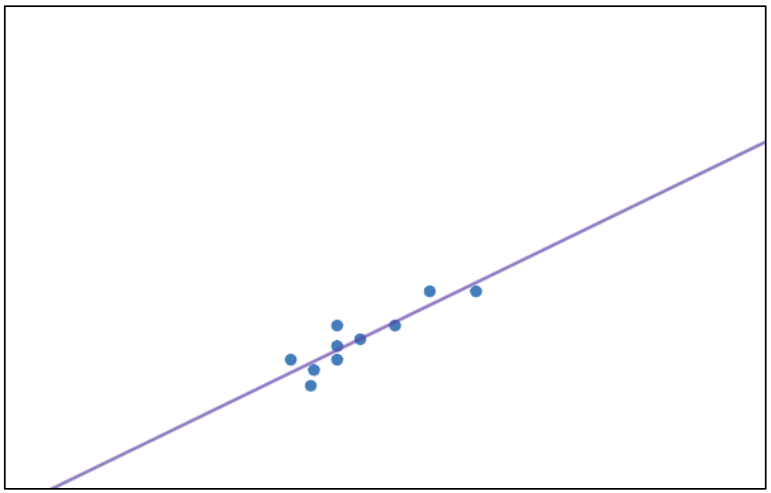
$$L \sim \overline{M_j}^{-t} \cdot (C_i - C_j) \times \overline{M_i}^{-1} \cdot p_i$$
$$L \sim (K_j \cdot R_j)^{-t} \cdot (C_i - C_j) \times (K_i \cdot R_i)^{-1} \cdot p_i$$
$$L \sim K_j^{-t} \cdot \underbrace{R_j^{-t} \cdot (C_i - C_j) \times R_i^{-1}}_{E_{ji}} \cdot K_i^{-1} \cdot p_i$$

Essential matrix: $E_{ji} \sim R_j \cdot (C_i - C_j) \times R_i^t \sim K_j \cdot F_{ji} \cdot K_i$

Robust Estimation

Issue with parametric model estimation (e.g. homography, epipolar geometry) : non relevant data or outliers.

Example with 2D line regression using least squares:

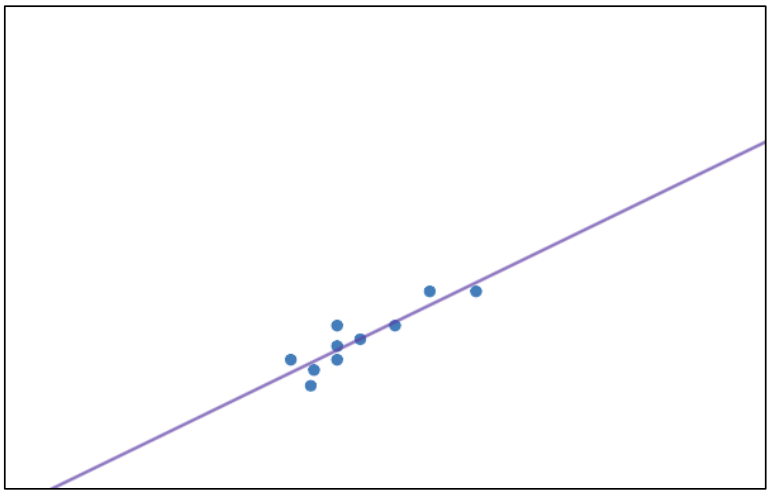


Least square solution given the blue points

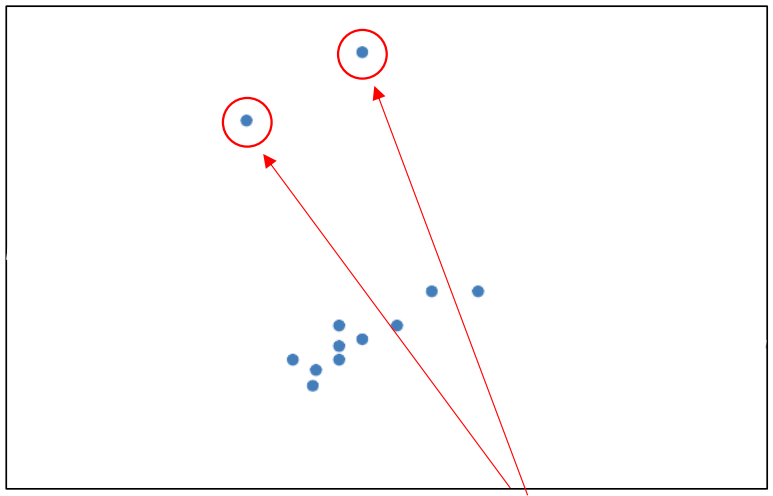
Robust Estimation

Issue with parametric model estimation (e.g. homography, epipolar geometry) : non relevant data or outliers.

Example with 2D line regression using least squares:



Least square solution given the blue points

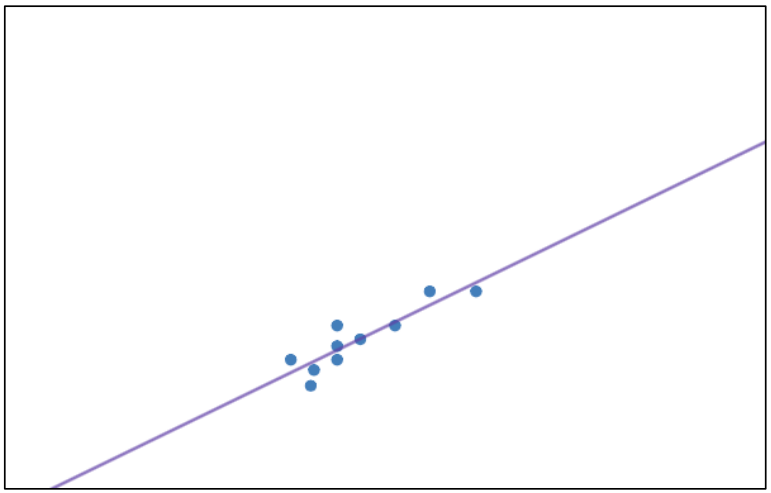


Adding 2 outliers

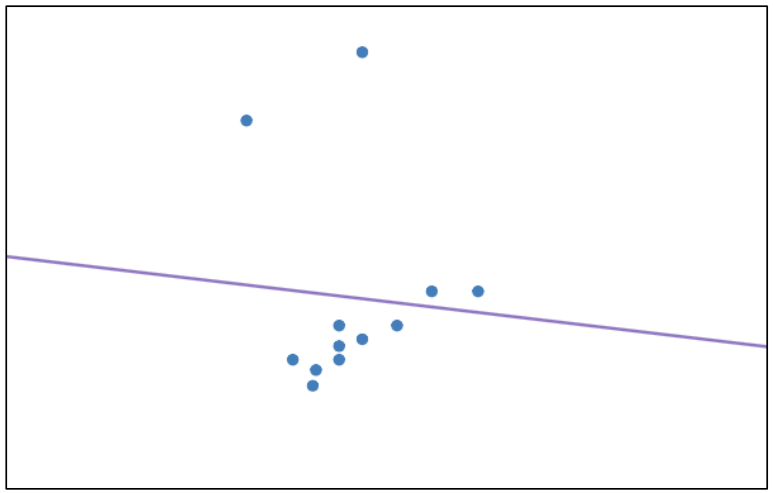
Robust Estimation

Issue with parametric model estimation (e.g. homography, epipolar geometry) : non relevant data or outliers.

Example with 2D line regression using least squares:



Least square solution given the blue points



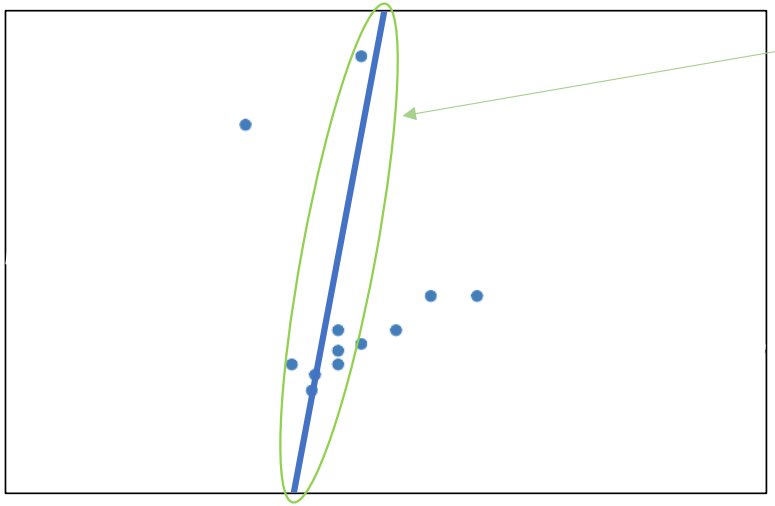
Estimation with 2 outliers

Robust Estimation

RANSAC: Random Sample Consensus – A very popular strategy in computer vision and beyond.

Idea: Instead of using all observations randomly select minimal sets and check how good they are with respect to outliers.

For the 2D linear regression:



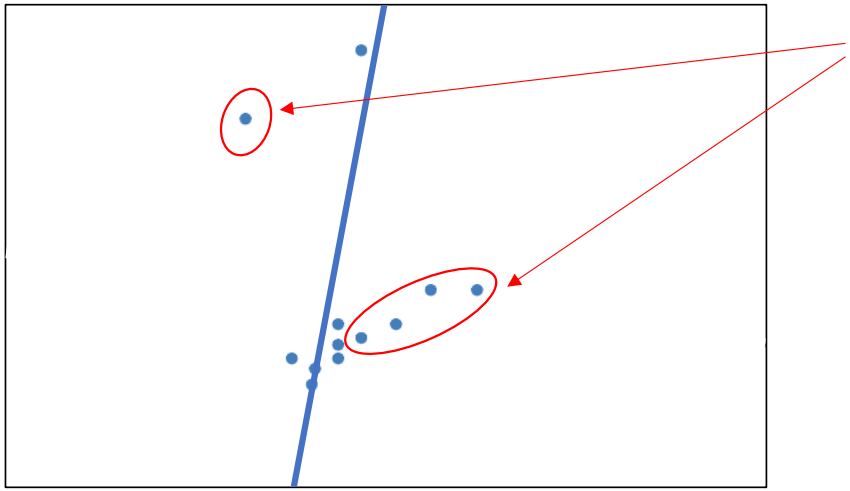
Inliers: point within error Ei

Robust Estimation

RANSAC: Random Sample Consensus

Idea: Instead of using all observations randomly select minimal sets and check how good they are with respect to outliers.

For the 2D linear regression:

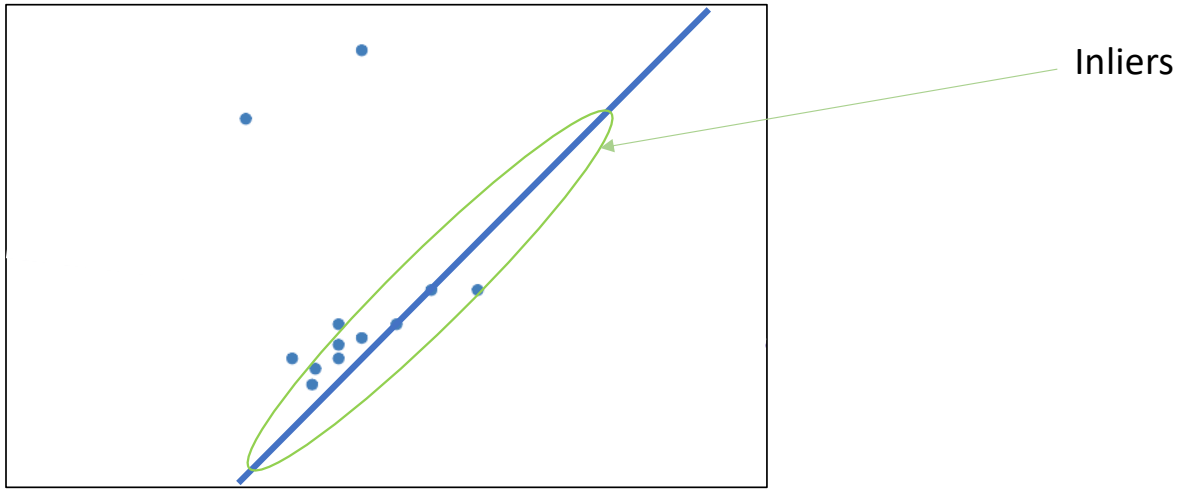


Robust Estimation

RANSAC: Random Sample Consensus

Idea: Instead of using all observations randomly select minimal sets and check how good they are with respect to outliers.

For the 2D linear regression:

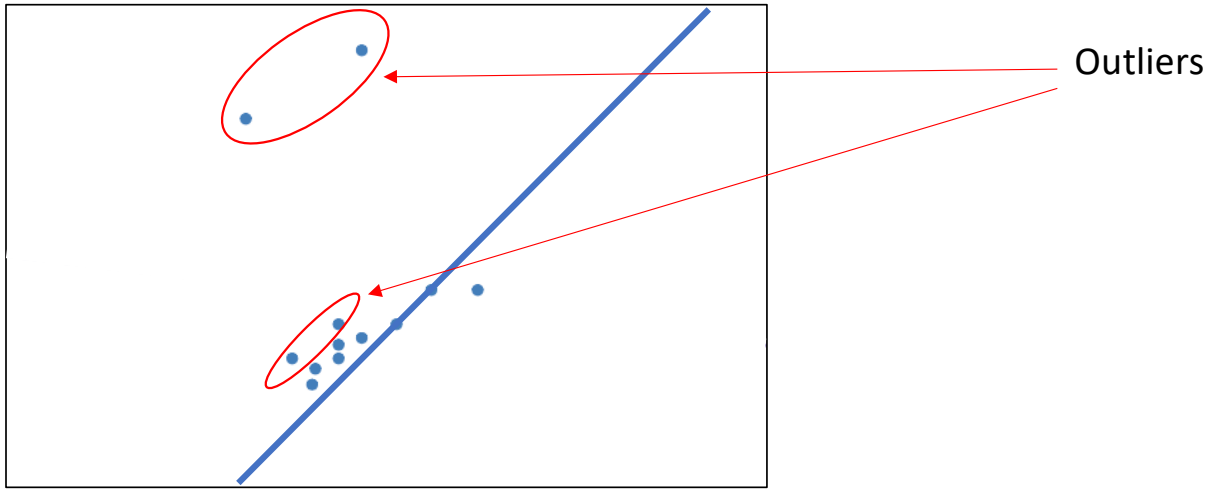


Robust Estimation

RANSAC: Random Sample Consensus

Idea: Instead of using all observations randomly select minimal sets and check how good they are with respect to outliers.

For the 2D linear regression:

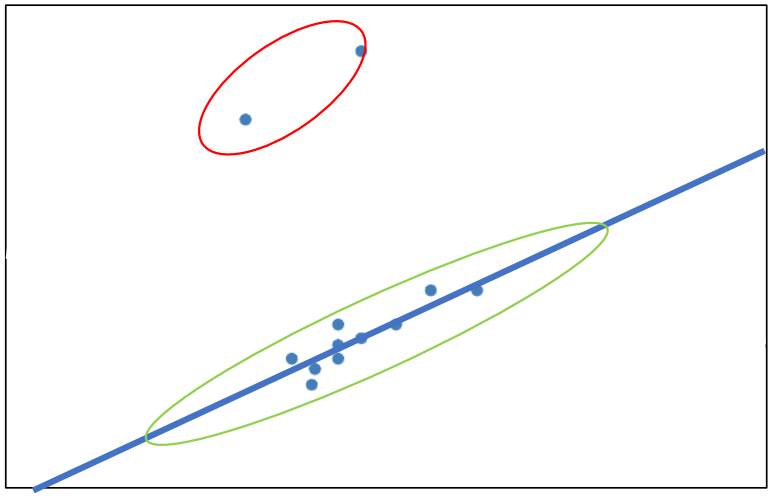


Robust Estimation

RANSAC: Random Sample Consensus

Idea: Instead of using all observations randomly select minimal sets and check how good they are with respect to outliers.

For the 2D linear regression:



Robust Estimation

RANSAC: Random Sample Consensus

Algorithm:

Input: Data, sample size : n; distanceseuil: D; minInliers: Ni

While (iterations < MaxIteration)

1. Sample = random sample of size n in Data
2. estimate Model given Sample
3. Inliers = Points in Data within distance D from Model
4. If #Inliers > Ni

```
        Estimate Model given Inliers
        Error  = Sum over point in Inliers of distance(point, model)
        If Error < BestError
            BestModel = Model
            BestError = Error
        endif
    endif
endwhile
Return BestModel
```

Robust Estimation

RANSAC: Random Sample Consensus

Assuming that w is the proportion of good data, n the sample size and k the number of samples

- w^n is the probability to get n good points (assuming independent sampling).
- (1-w^n) is the probablity that one point is not good within a sample.
- (1-w^n)^k is the probability that k samples are not good.
- 1-(1-w^n)^k is the probality that at least one sample is good.

Thus the number of iteration k required to get a good sample with a probability higher than p is:

- k > ln(1-p)/ln(1-w^n).

Can ransac succeed with more than 50% outliers ?