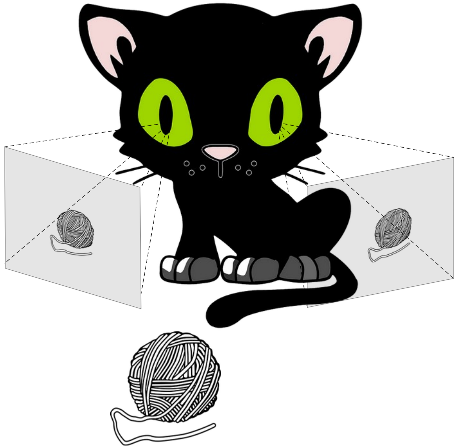
Computer Vision

Class 07

Raquel Frizera Vassallo

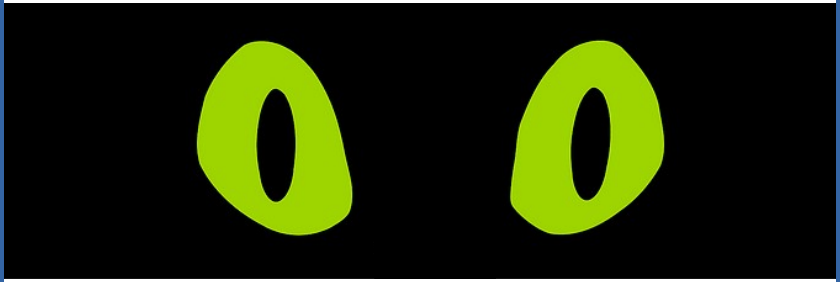
Stereo Vision
and
Epipolar Geometry



Summary

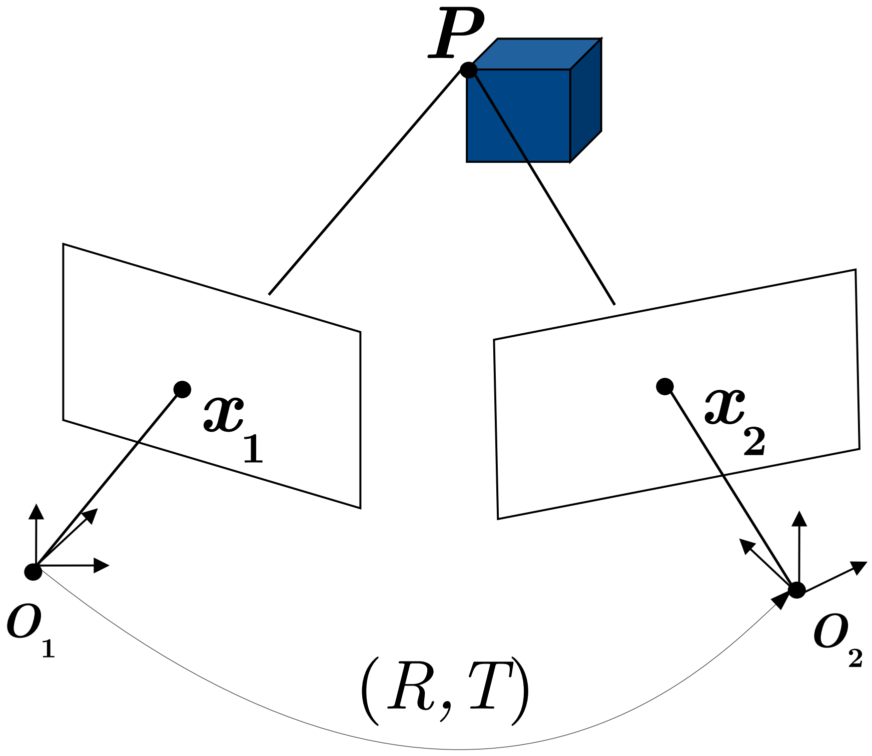
- Stereo Vision
- Camera Model
- Calibrated System
- Partially Calibrated System
- Epipolar Geometry
- Essential Matrix

Stereo Vision

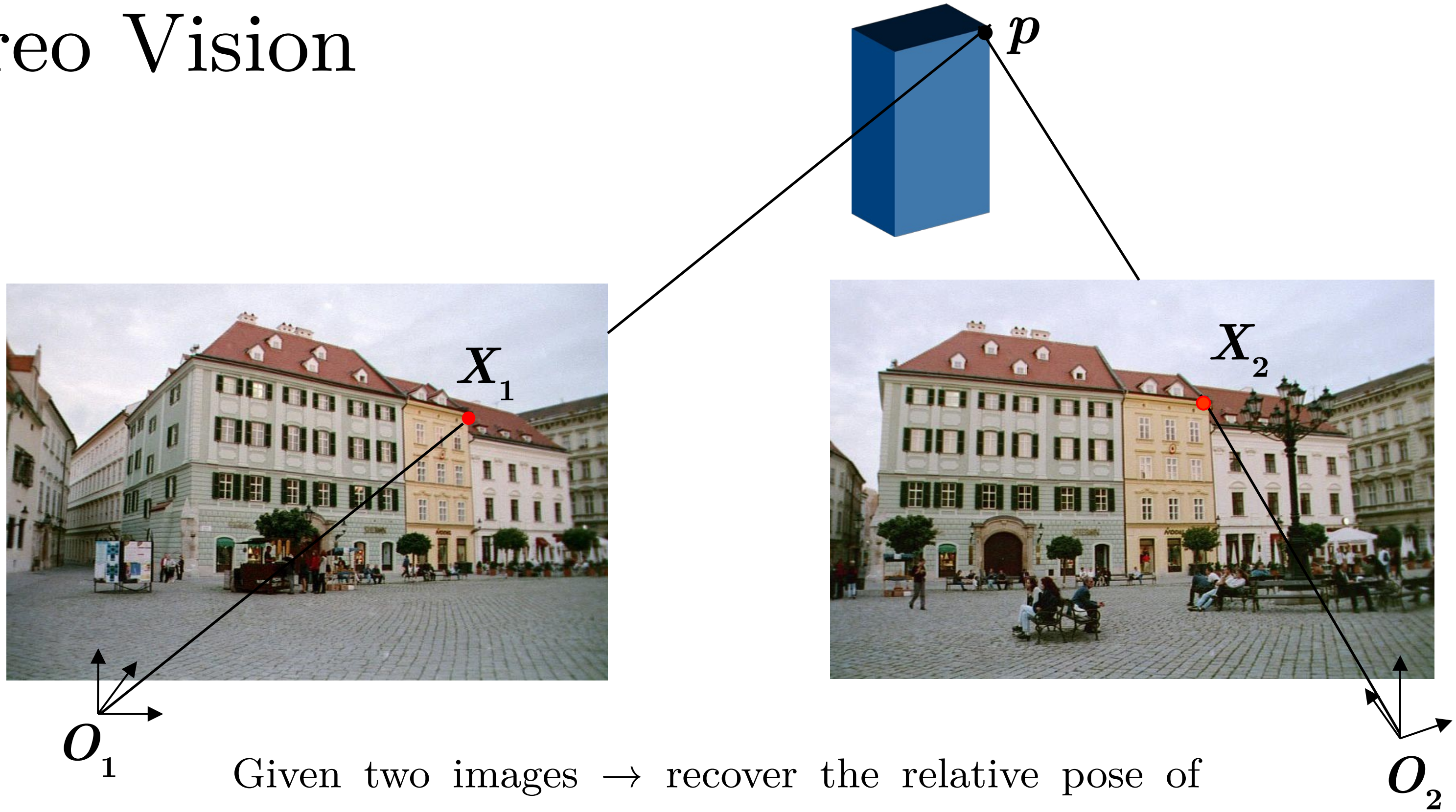


Stereo Vision

- Two images
- Matching of points between images
- Distance between two corresponding points in the left and right image: **disparity**
- Allow recovering 3D information



Stereo Vision



Given two images → recover the relative pose of the cameras and the 3D structure of the scene

Stereo Vision

Usually the problem is:

- Given two images, recover the tridimensional structure of the scene and the relative pose of the cameras.

The vision system can be:

- Calibrated (intrinsic and extrinsic parameters are known)
- Partially calibrated (just intrinsic parameters are known)
- Not calibrated (all parameters are unknown)

Pinhole Camera Model

For each camera, the image points are obtained as:

$$\lambda \underbrace{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}_{\text{Point on the 2D Image}} = \underbrace{\begin{bmatrix} fs_x & fs_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix}}_{\text{Intrinsic Parameter Matrix}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{Projection Matrix}} \underbrace{\begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}}_{\text{Extrinsic Parameter Matrix}} \underbrace{\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}}_{\text{Point in 3D World}}$$

Pinhole Camera Model

For each camera, the image points are obtained as:

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} fs_x & fs_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix}}_{\text{Intrinsic Parameter Matrix}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{Projection Matrix}} \underbrace{\begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}}_{\text{Extrinsic Parameter Matrix}} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

Point on the 2D Image

Point in 3D World

Knowing the intrinsic parameters, we can convert pixels to metric points → remove the intrinsic parameter matrix.

Pinhole Camera Model

For each camera, the image points are obtained as:

$$\lambda \underbrace{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}_{\text{Point on the 2D Image}} = \underbrace{\xcancel{\begin{bmatrix} fs_x & fs_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix}}}_{\text{Intrinsic Parameter Matrix}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{Projection Matrix}} \underbrace{\begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}}_{\text{Extrinsic Parameter Matrix}} \underbrace{\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}}_{\text{Point in 3D World}}$$

Knowing the intrinsic parameters, we can convert pixels to metric points → remove the intrinsic parameter matrix.

Considering the camera frame as the world frame → remove the extrinsic parameters matrix.

Pinhole Camera Model

For each camera, the image points are obtained as:

$$\underbrace{\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}_{\text{Point on the 2D Image}} = \underbrace{\cancel{\begin{bmatrix} fs_x & fs_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix}}}_{\text{Intrinsic Parameter Matrix}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{Projection Matrix}} \underbrace{\cancel{\begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}}}_{\text{Extrinsic Parameter Matrix}} \underbrace{\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}}_{\text{Point in 3D World}}$$

Knowing the intrinsic parameters, we can convert pixels to metric points $\rightarrow$ remove the intrinsic parameter matrix.

Considering the camera frame as the world frame $\rightarrow$ remove the extrinsic parameters matrix.

Pinhole Camera Model

For each camera, the image points are obtained as:

$$\lambda \underbrace{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}_{\text{Point on the 2D Image}} = \underbrace{\xcancel{\begin{bmatrix} fs_x & fs_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix}}}_{\text{Intrinsic Parameter Matrix}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{Projection Matrix}} \underbrace{\xcancel{\begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}}}_{\text{Extrinsic Parameter Matrix}} \underbrace{\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}}_{\text{Point in 3D World}}$$

$$\boxed{\lambda \boldsymbol{x} = \boldsymbol{X}}$$

Camera 1

$$\underbrace{\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}_{\text{Image Point}} = \underbrace{\begin{bmatrix} fs_x & fs_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix}}_{\text{Intrinsic Parameter}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{Projection Matrix}} \underbrace{\begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}}_{\text{Extrinsic Parameter}} \underbrace{\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}}_{\text{3D Point}}$$

Knowing the intrinsic parameters $\rightarrow$ remove the intrinsic parameter matrix.

First camera frame as the world frame $\rightarrow$ remove the extrinsic parameters matrix.

Camera 1

$$\underbrace{\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}_{\text{Image Point}} = \underbrace{\cancel{\begin{bmatrix} fs_x & fs_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix}}}_{\text{Intrinsic Parameter}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{Projection Matrix}} \underbrace{\cancel{\begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}}}_{\text{Extrinsic Parameter}} \underbrace{\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}}_{\text{3D Point}}$$

Knowing the intrinsic parameters $\rightarrow$ remove the intrinsic parameter matrix.

First camera frame as the world frame $\rightarrow$ remove the extrinsic parameters matrix.

$$\boxed{\lambda_1 \boldsymbol{x}_1 = \boldsymbol{X}}$$

Camera 2

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} fs_x & fs_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix}}_{\text{Intrinsic Parameters}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{Projection Matrix}} \underbrace{\begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}}_{\text{Extrinsic Parameters relate cameras 1 and 2}} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

Image Point

3D Point

Knowing the intrinsic parameters → remove the intrinsic parameter matrix.

First camera frame as the world frame → the extrinsic parameter matrix relates the frames of cameras 1 and 2.

Camera 2

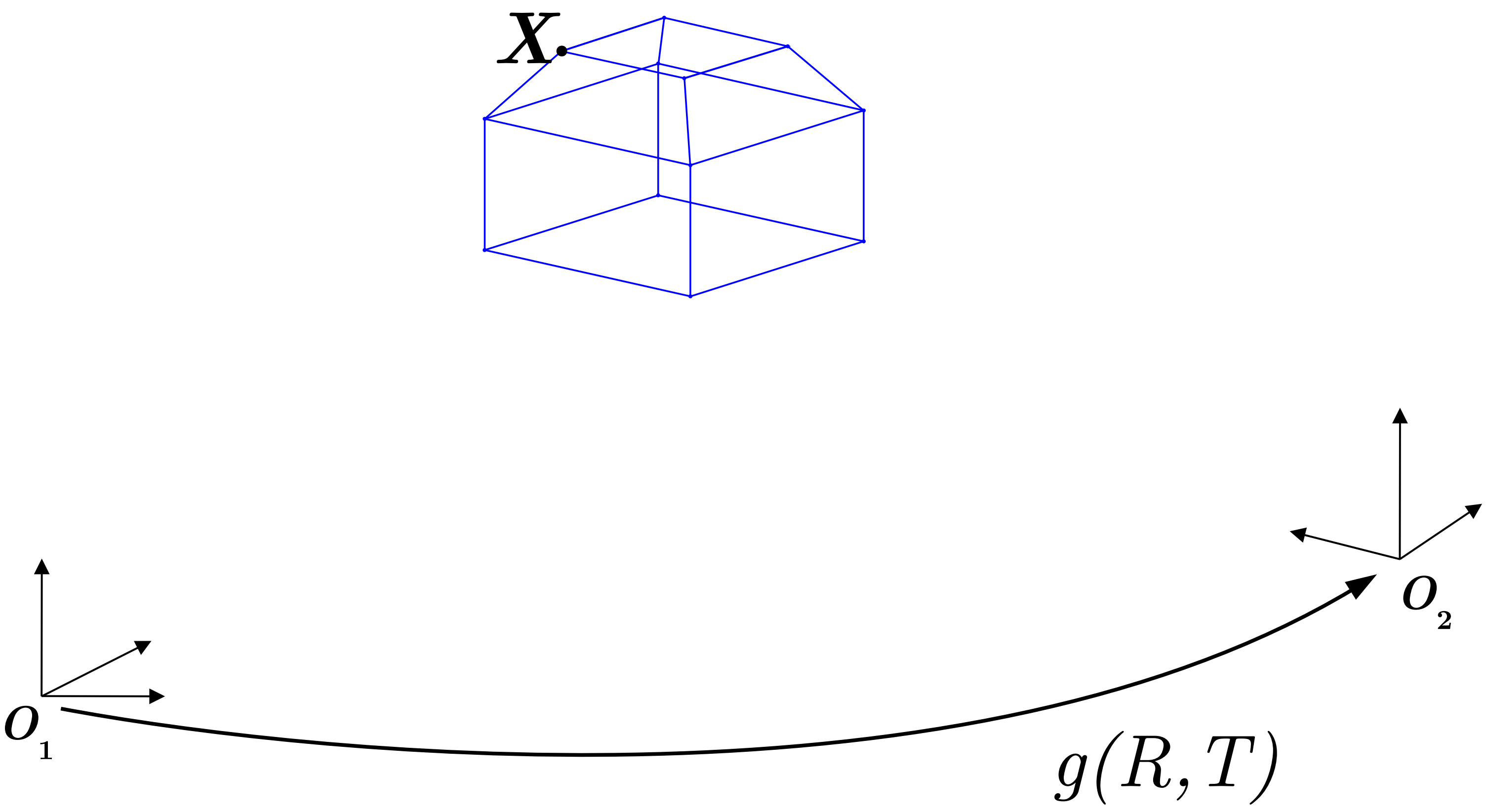
$$\underbrace{\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}_{\text{Image Point}} = \underbrace{\xcancel{\begin{bmatrix} fs_x & fs_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix}}}_{\text{Intrinsic Parameters}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{Projection Matrix}} \underbrace{\begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}}_{\boxed{\text{Extrinsic Parameters relate cameras 1 and 2}}} \underbrace{\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}}_{\text{3D Point}}$$

Knowing the intrinsic parameters → remove the intrinsic parameter matrix.

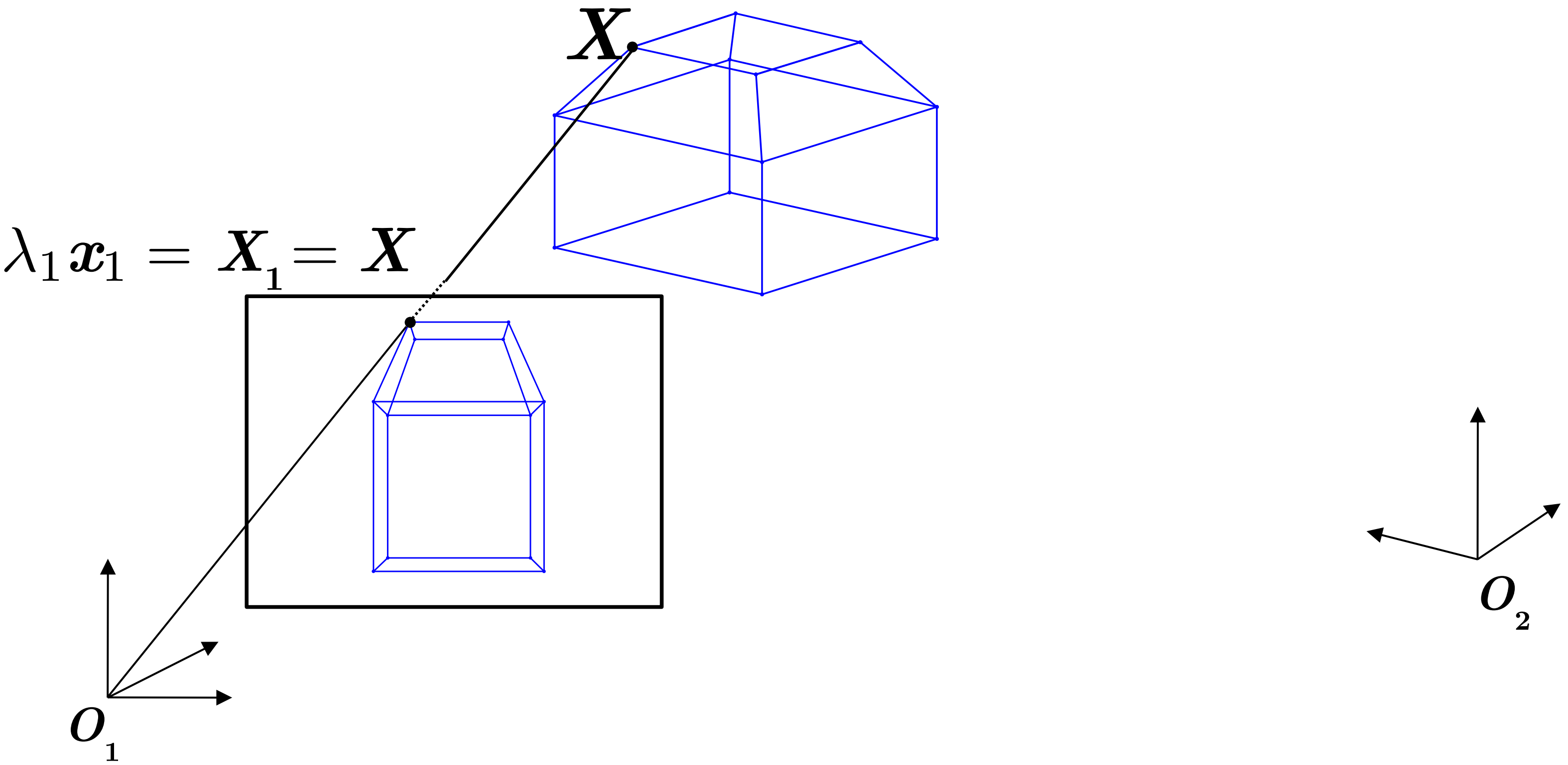
First camera frame as the world frame → the extrinsic parameter matrix relates the frames of cameras 1 and 2.

$$\boxed{\lambda_2 \boldsymbol{x}_2 = R\lambda_1 \boldsymbol{x}_1 + T}$$

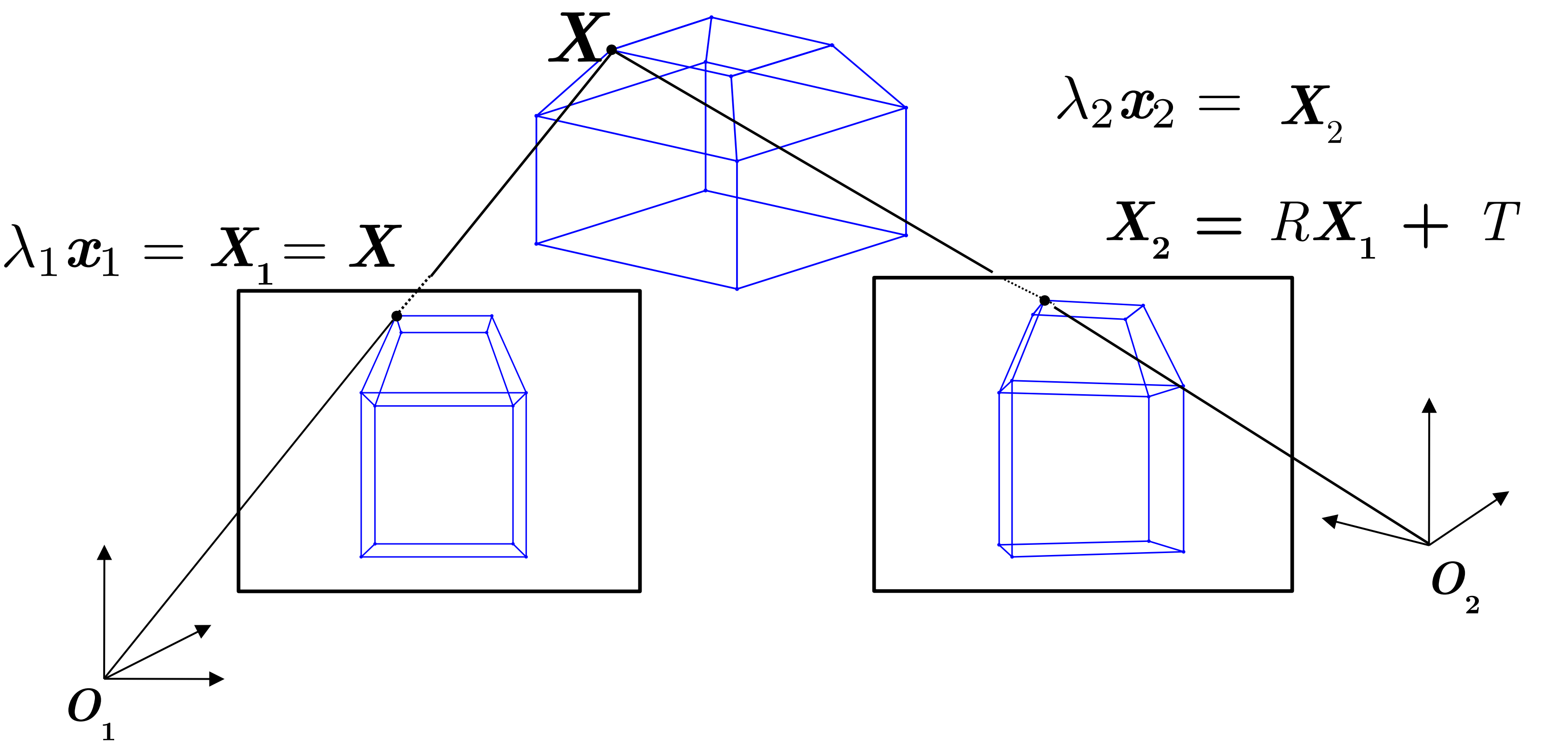
Relation between the points of the two images



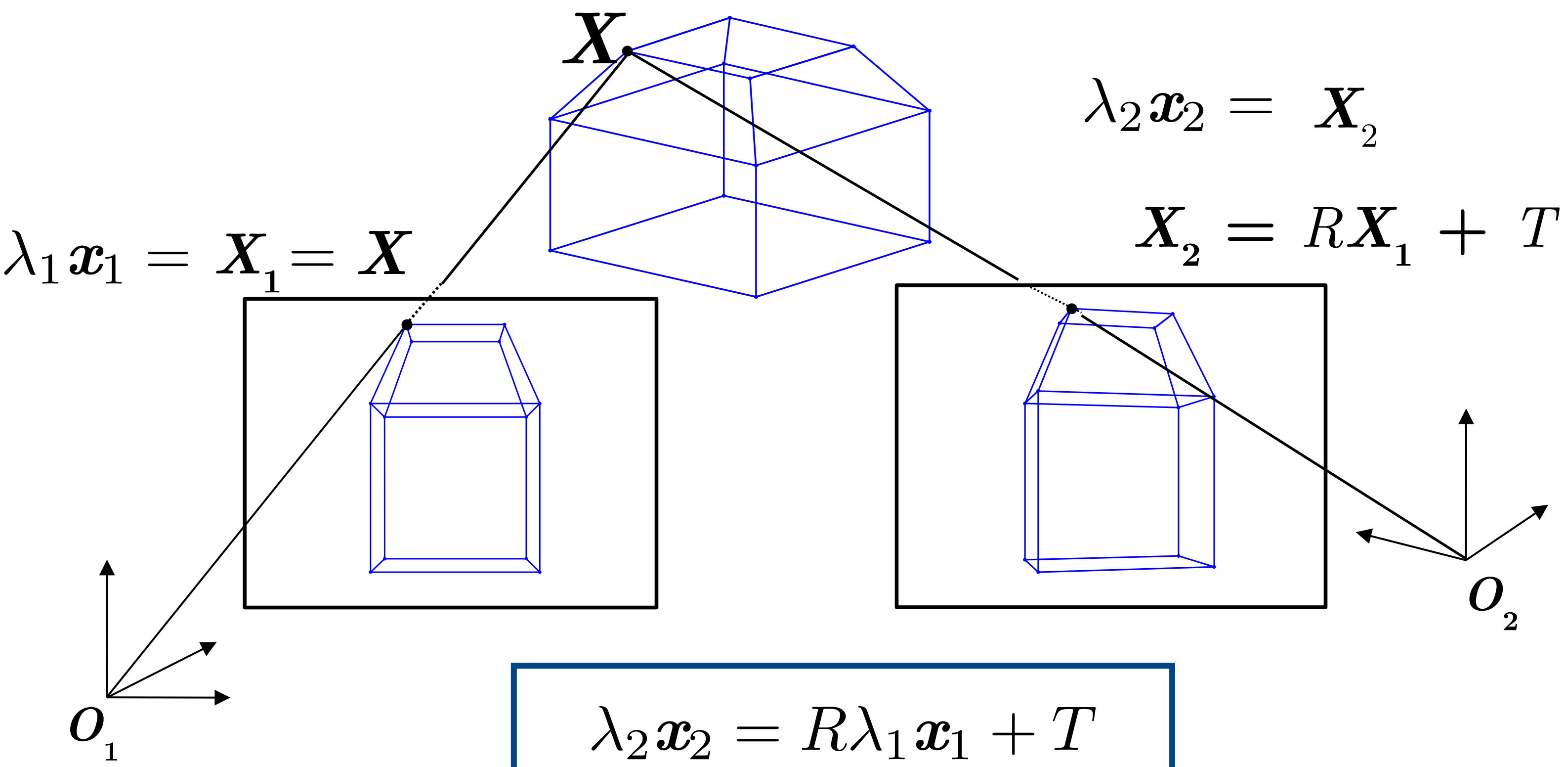
Relation between the points of the two images



Relation between the points of the two images



Relation between the points of the two images



$$\boxed{\lambda_2 \boldsymbol{x}_2 = R\lambda_1 \boldsymbol{x}_1 + T}$$

Calibrated System

- If the stereo system is fully calibrated, the intrinsic parameters of both cameras as well as the relative pose of the cameras (R, T: extrinsic parameters) are known. The 3D position of the points can be obtained by triangulation.

$$\lambda_1 \boldsymbol{x}_1 = \boldsymbol{X} \longrightarrow \begin{aligned} \lambda_1 x_1 &= X \\ \lambda_1 y_1 &= Y \\ \lambda_1 &= Z \end{aligned}$$

$$\lambda_2 \boldsymbol{x}_2 = R \lambda_1 \boldsymbol{x}_1 + T$$

4 unkowns: (X, Y, Z, λ_2)
and 4 equations

Partially Calibrated

Intrinsic Parameters known

unknowns
measurements

Rigid-body transformation $\lambda_2 \boldsymbol{x}_2 = R\lambda_1 \boldsymbol{x}_1 + T$

Minimize the reprojection error:

$$\sum_{j=1}^{n} \left\| \boldsymbol{x}_1^j - \pi(R_1, T_1, \boldsymbol{X}) \right\|^2 + \left\| \boldsymbol{x}_2^j - \pi(R_2, T_2, \boldsymbol{X}) \right\|^2$$

Partially Calibrated

Intrinsic Parameters known

Rigid-body transformation $$\underbrace{\lambda_2}\underbrace{\boldsymbol{x}_2} = \underbrace{R\lambda_1}\underbrace{\boldsymbol{x}_1} + \underbrace{T}$$

unknowns

measurements

Even if one of the cameras is taken as the main reference frame (R_1=0 and T_1=0) and just the pose with respect to the other camera (R_2, T_2) need to be estimated, we must find the Rotation, the Translation and the Depth that minimize the reprojection error.

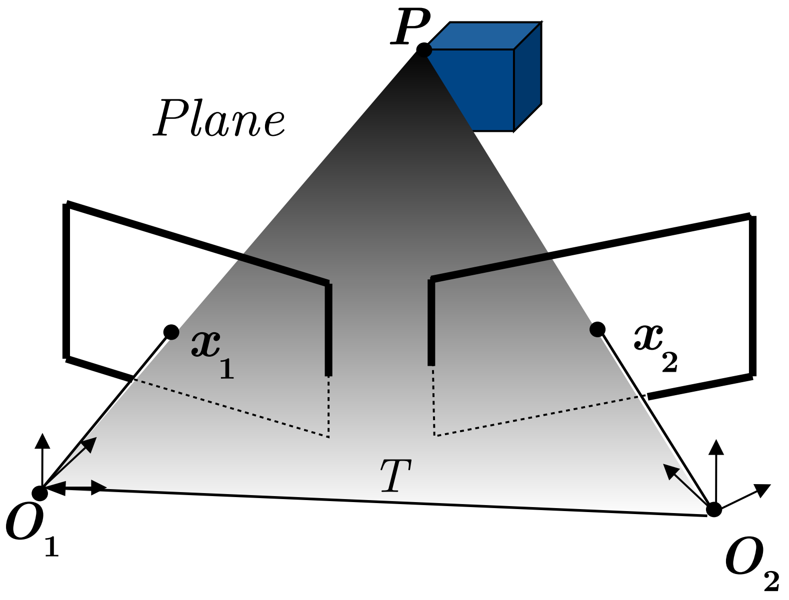
Usually:

- 2 views ~ Number of correspondences (N): from 200 to 200.000 points
- 6 unkowns - Motion: Rotation (3), Translation (3)
  - Structure: N×3 coordinates
  - Scale factor

Hard Optimization Problem!

Epipolar Geometry

$$\lambda_2 \boldsymbol{x_2} = \lambda_1 R \boldsymbol{x_1} + T$$

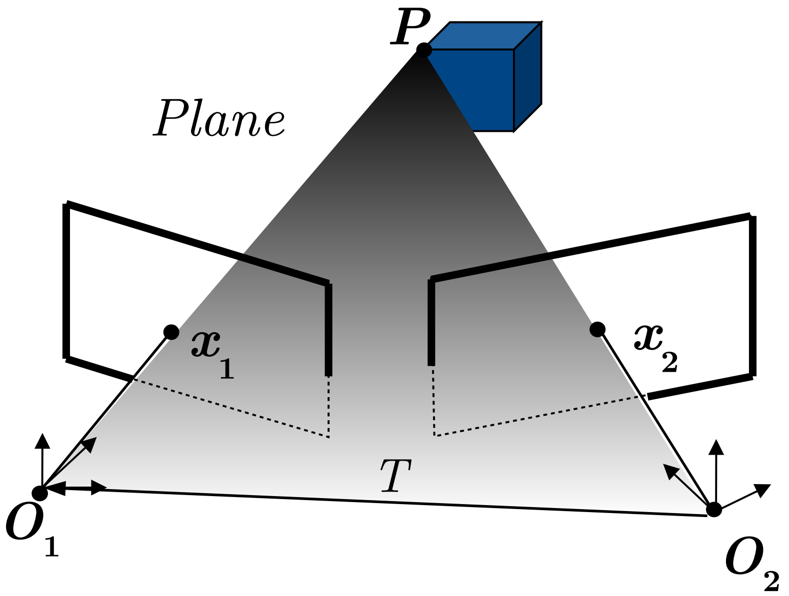


Epipolar Geometry

$$\lambda_2 \boldsymbol{x_2} = \lambda_1 R\boldsymbol{x_1} + T$$

Performing the cross product with T on both sides:

$$\lambda_2 T\times\boldsymbol{x_2} = \lambda_1 T\times R\boldsymbol{x_1} + \cancelto{\mathbf{0}}{T\times T}$$



Epipolar Geometry

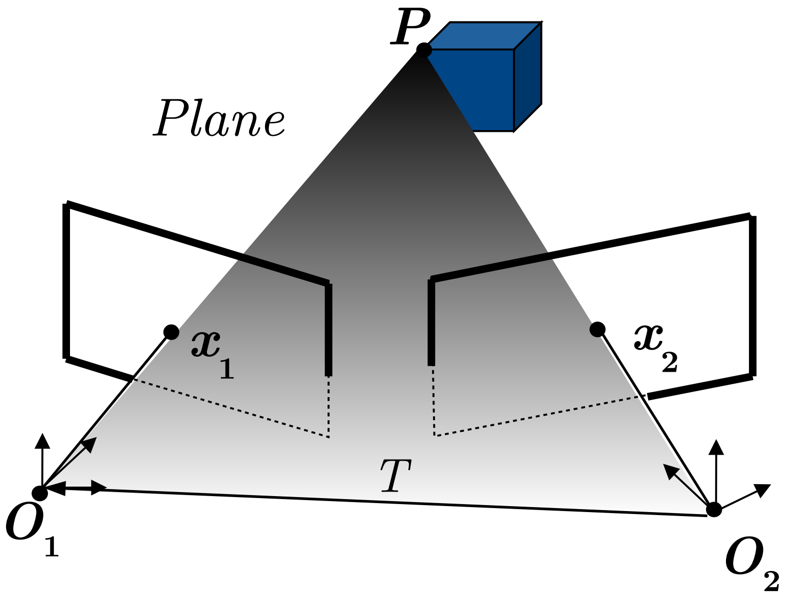
$$\lambda_2 \boldsymbol{x_2} = \lambda_1 R\boldsymbol{x_1} + T$$

Performing the cross product with T on both sides:

$$\lambda_2 T\times\boldsymbol{x_2} = \lambda_1 T\times R\boldsymbol{x_1} + \underbrace{\cancel{T\times T}}_{\mathbf{0}}$$

Performing the inner product with $\boldsymbol{x_2}$:

$$\underbrace{\cancel{\lambda_2 \boldsymbol{x_2}^T T\times\boldsymbol{x_2}}}_{\mathbf{0}} = \lambda_1 \boldsymbol{x_2}^T T\times R\boldsymbol{x_1}$$



Epipolar Geometry

$$\lambda_2 \boldsymbol{x}_2 = \lambda_1 R\boldsymbol{x}_1 + T$$

Performing the cross product with T on both sides:

$$\lambda_2 T\times \boldsymbol{x}_2 = \lambda_1 T\times R\boldsymbol{x}_1 + \cancelto{\mathbf{0}}{T\times T}$$

Performing the inner product with $\boldsymbol{x}_2$:

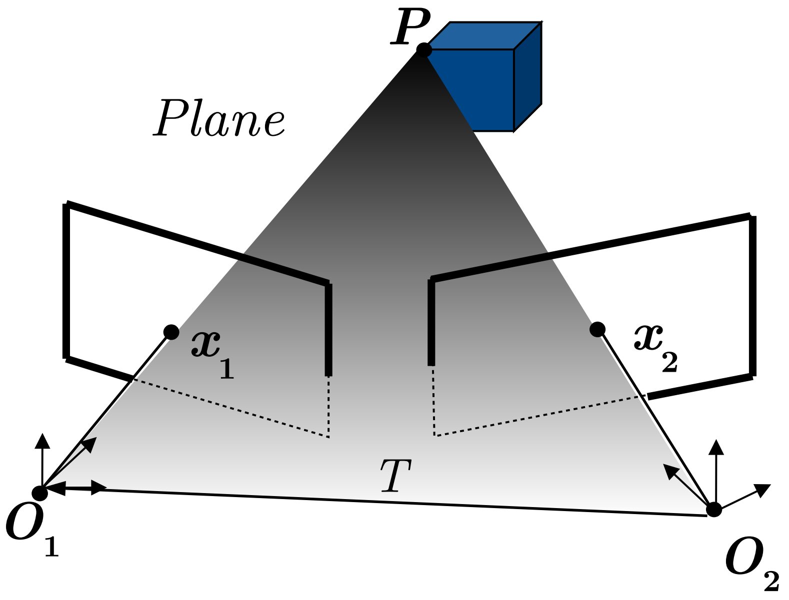
$$\cancelto{\mathbf{0}}{\lambda_2 \boldsymbol{x}_2^T T\times \boldsymbol{x}_2} = \lambda_1 \boldsymbol{x}_2^T T\times R\boldsymbol{x}_1$$

$$\boldsymbol{x}_2^T T\times R\boldsymbol{x}_1 = 0$$

Essential Matrix

$$E = T\times R$$

or $E = \widehat{T}R$



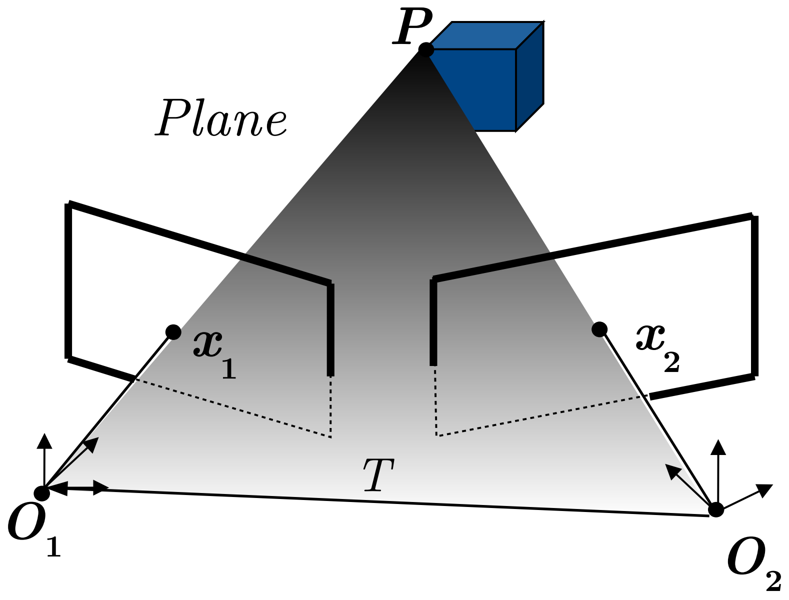
$$\boxed{\boldsymbol{x}_2^T E\, \boldsymbol{x}_1 = 0}$$

Epipolar Geometry

The 3D point (P) and its projections on the images ($\boldsymbol{x}_1$ and $\boldsymbol{x}_2$) are on the same plane.

$$\boldsymbol{x}_2{}^T E\,\boldsymbol{x}_1 = 0$$

$$\boldsymbol{x}_2{}^T\, T \times R\,\boldsymbol{x}_1 = 0$$



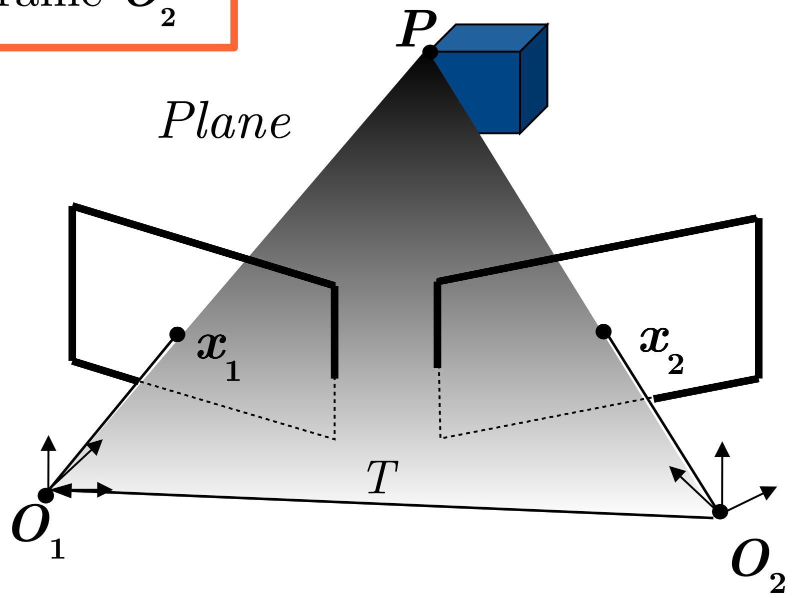
Epipolar Geometry

The 3D point (P) and its projections on the images ($\boldsymbol{x}_1$ and $\boldsymbol{x}_2$) are on the same plane.

$$\boldsymbol{x}_2{}^T E\, \boldsymbol{x}_1 = 0$$

Projection of $\boldsymbol{x}_1$ on the frame $\boldsymbol{O}_2$

$$\boldsymbol{x}_2{}^T\, T \times R\, \boldsymbol{x}_1 = 0$$



Epipolar Geometry

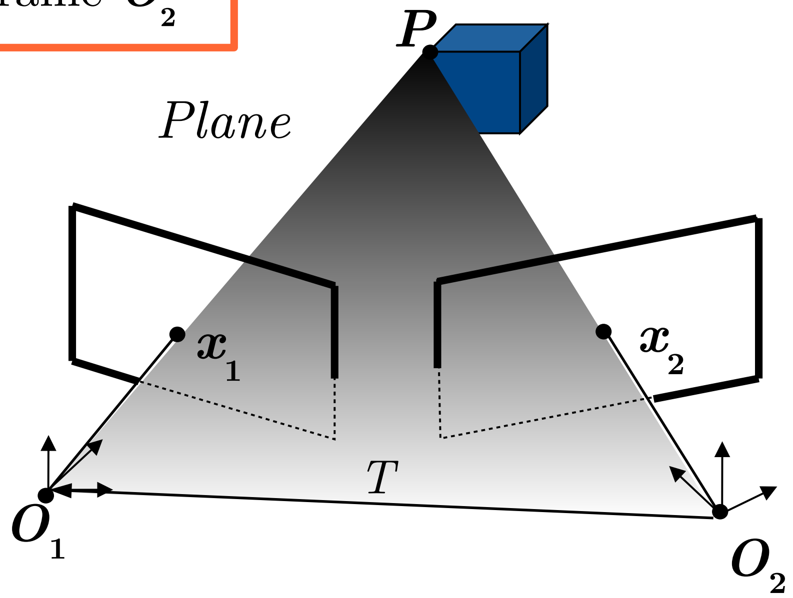
The 3D point (P) and its projections on the images ($\boldsymbol{x}_1$ and $\boldsymbol{x}_2$) are on the same plane.

$$\boldsymbol{x}_2{}^T E\, \boldsymbol{x}_1 = 0$$

Projection of $\boldsymbol{x}_1$ on the frame $\boldsymbol{O}_2$

$$\boldsymbol{x}_2{}^T T \times R\, \boldsymbol{x}_1 = 0$$

Vector perpendicular to $R\,\boldsymbol{x}_1 \rightarrow$ Normal of the plane



Epipolar Geometry

The 3D point (P) and its projections on the images ($\boldsymbol{x}_1$ and $\boldsymbol{x}_2$) are on the same plane.

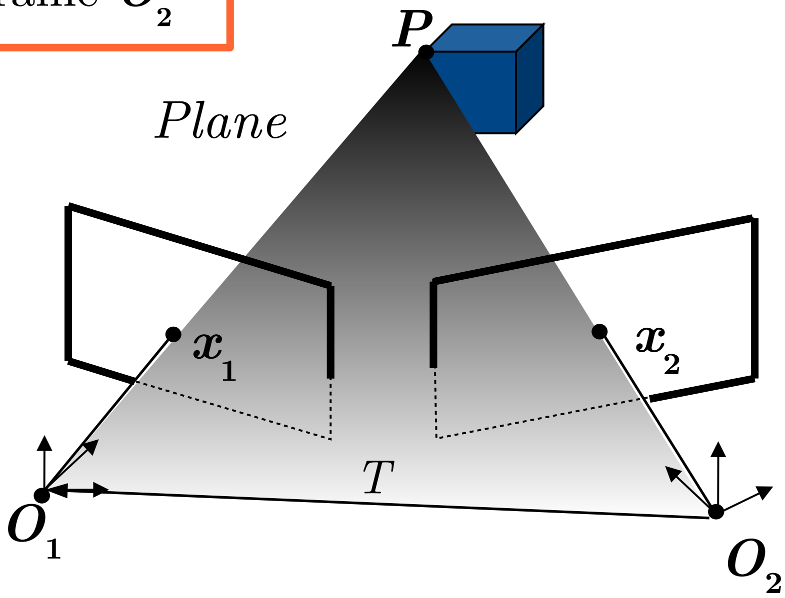
$$\boldsymbol{x}_2^{\,T} E\, \boldsymbol{x}_1 = 0$$

Projection of $\boldsymbol{x}_1$ on the frame $\boldsymbol{O}_2$

$$\boldsymbol{x}_2^{\,T}\, T \times R\, \boldsymbol{x}_1 = 0$$

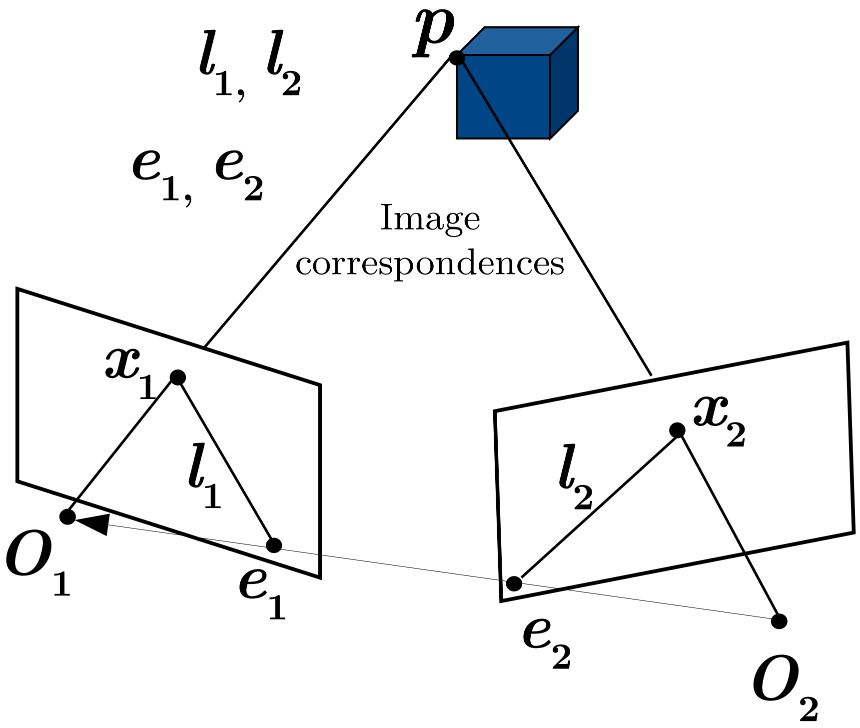
Since $\boldsymbol{x}_2$ is on the plane, the inner product with $T \times R\, \boldsymbol{x}_1$ is zero

Vector perpendicular to $R\, \boldsymbol{x}_1 \rightarrow$ Normal of the plane



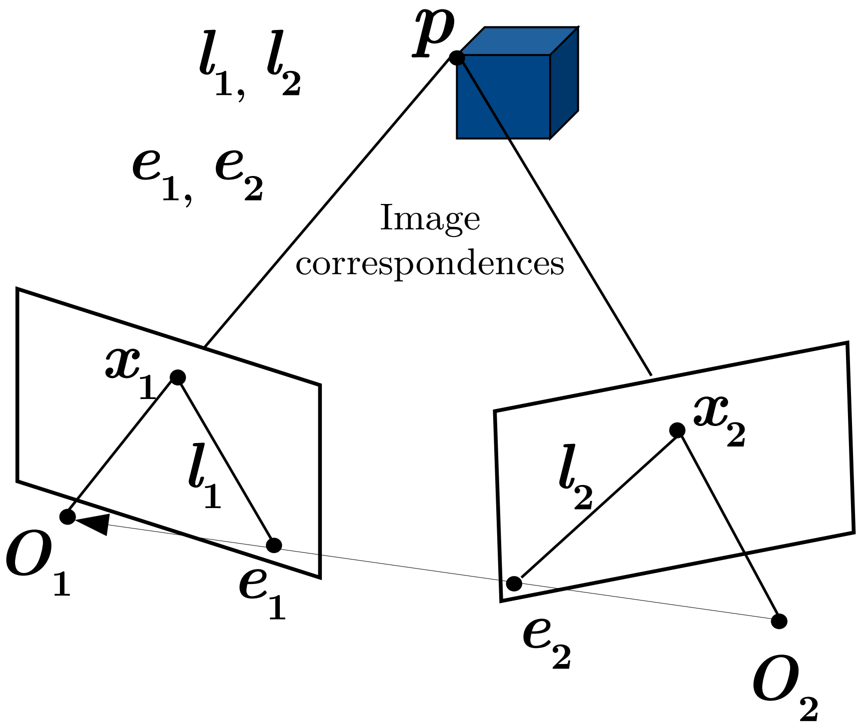
Epipoles and Epipolar Lines

- The plane $(\boldsymbol{O_1}, \boldsymbol{O_2}, \boldsymbol{p})$ is called epipolar plane associated with the point $\boldsymbol{p}$. There is one epipolar plane for each point $\boldsymbol{p}$.



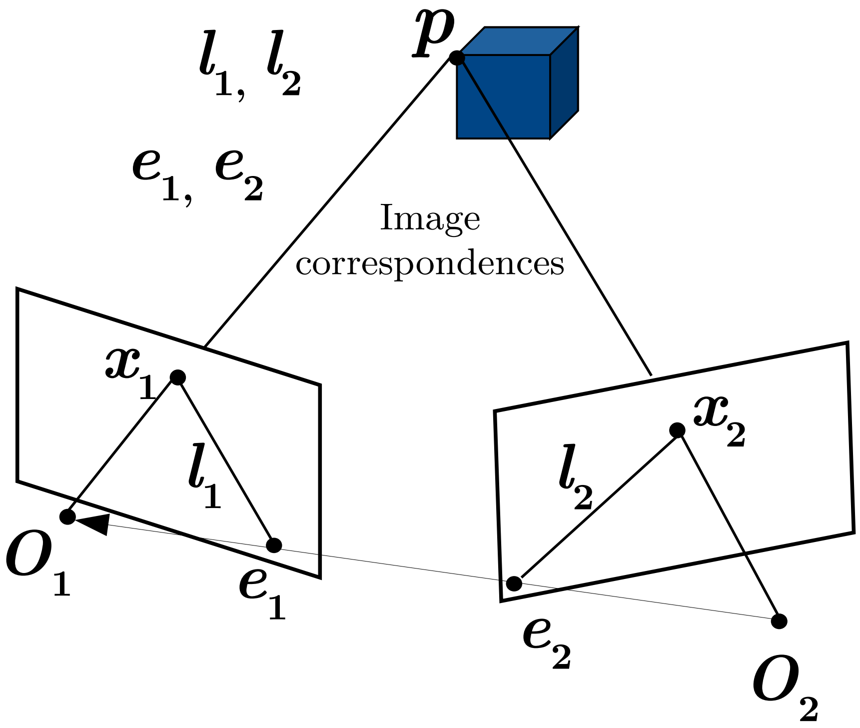
Epipoles and Epipolar Lines

- The plane $(\boldsymbol{O}_1, \boldsymbol{O}_2, \boldsymbol{p})$ is called epipolar plane associated with the point $\boldsymbol{p}$. There is one epipolar plane for each point $\boldsymbol{p}$.
- The projection of one camera center onto the image plane of the other camera frame is called epipole. So $\boldsymbol{e}_1$ is the projection of $\boldsymbol{O}_2$, and $\boldsymbol{e}_2$ of $\boldsymbol{O}_1$.



Epipoles and Epipolar Lines

- The plane $(\boldsymbol{O}_1, \boldsymbol{O}_2, \boldsymbol{p})$ is called epipolar plane associated with the point $\boldsymbol{p}$. There is one epipolar plane for each point $\boldsymbol{p}$.
- The projection of one camera center onto the image plane of the other camera frame is called epipole. So $\boldsymbol{e}_1$ is the projection of $\boldsymbol{O}_2$, and $\boldsymbol{e}_2$ of $\boldsymbol{O}_1$.
- The intersection of the epipolar plane of $\boldsymbol{p}$ with one image plane is a line, which is called epipolar line of $\boldsymbol{p}$. There is an epipolar line on each image, $\boldsymbol{l}_1$ e $\boldsymbol{l}_2$.



Properties

1. The two epipoles $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ are the left and right null spaces of the essential matrix E.

$$E\boldsymbol{e}_1 = 0 \qquad \boldsymbol{e}_2^T E = 0$$

$$\boxed{\begin{array}{c} \boldsymbol{x}_2^T E \boldsymbol{x}_1 = 0 \\ E = \hat{T}R \end{array}}$$

$\boldsymbol{e}_2 \sim T$ and $\boldsymbol{e}_1 \sim R^T T$ (equality up to a scalar factor).

2. The epipolar lines $\boldsymbol{l}_1$ and $\boldsymbol{l}_2$ associated with the two image points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, can be expressed as :

$$\boldsymbol{l}_1 \sim E^T \boldsymbol{x}_2 \qquad \boldsymbol{l}_2 \sim E\boldsymbol{x}_1$$

where $\boldsymbol{l}_1$ and $\boldsymbol{l}_2$ are the normal vectors to the epipolar plane expressed with respect to the two camera frames, respectively.

3. In each image, both the image point and the epipole lie on the epipolar line.

$$\boldsymbol{l}_i^T \boldsymbol{e}_i = 0 \qquad \boldsymbol{l}_i^T \boldsymbol{x}_i = 0$$

Summary and Discussion

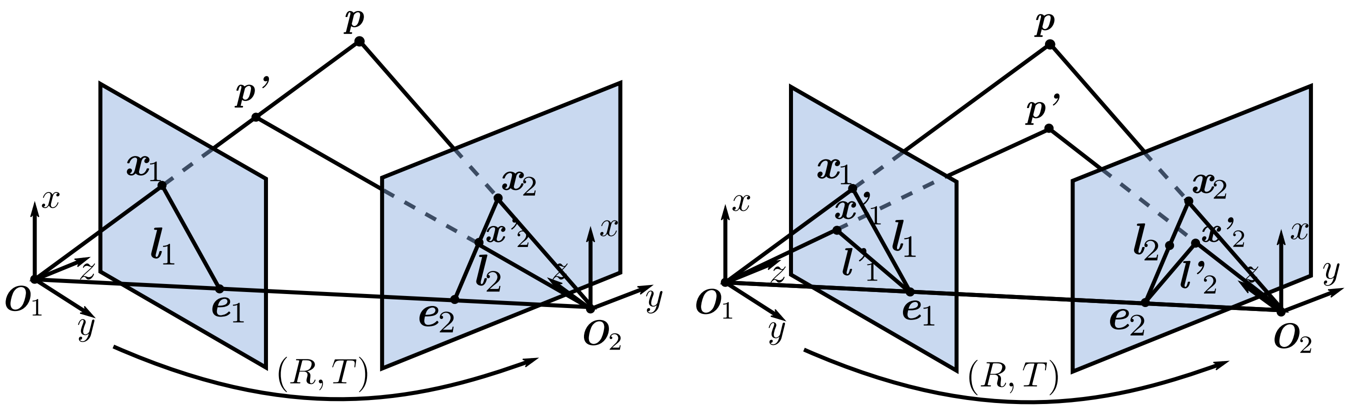


Figure 5.2 Left: the essential matrix E associated with the epipolar constraint maps an image point $\boldsymbol{x}_1$ in the first image to an epipolar line $\boldsymbol{l}_2 = E\boldsymbol{x}_1$ in the second image; the precise location of its corresponding image ($\boldsymbol{x}_2$ or $\boldsymbol{x}'_2$) depends on where the 3D point ($\boldsymbol{p}$ or $\boldsymbol{p}'$) lies on the ray $(\boldsymbol{O}_1, \boldsymbol{x}_1)$; Right: When $(\boldsymbol{O}_1, \boldsymbol{O}_2, \boldsymbol{p})$ and $(\boldsymbol{O}_1, \boldsymbol{O}_2, \boldsymbol{p}')$ are two different planes, they intersect at the two image planes at two pairs of epipolar lines $(\boldsymbol{l}_1, \boldsymbol{l}_2)$ and $(\boldsymbol{l}'_1, \boldsymbol{l}'_2)$, respectively, and these epipolar lines always pass through the pair of epipoles $(\boldsymbol{e}_1, \boldsymbol{e}_2)$.

$$\boldsymbol{x}_2^T E \boldsymbol{x}_1 = 0$$
$$E = \hat{T} R$$

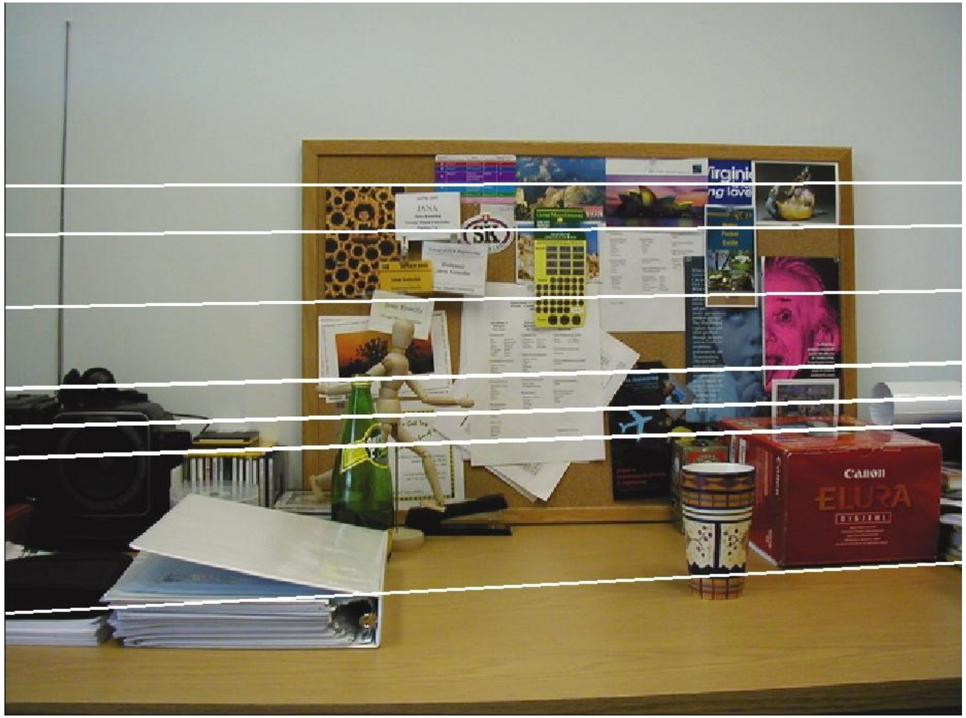
$$\boldsymbol{l}_1 \sim E^T \boldsymbol{x}_2$$
$$\boldsymbol{l}_2 \sim E \boldsymbol{x}_1$$

$$E \boldsymbol{e}_1 = 0$$
$$\boldsymbol{e}_2^T E = 0$$

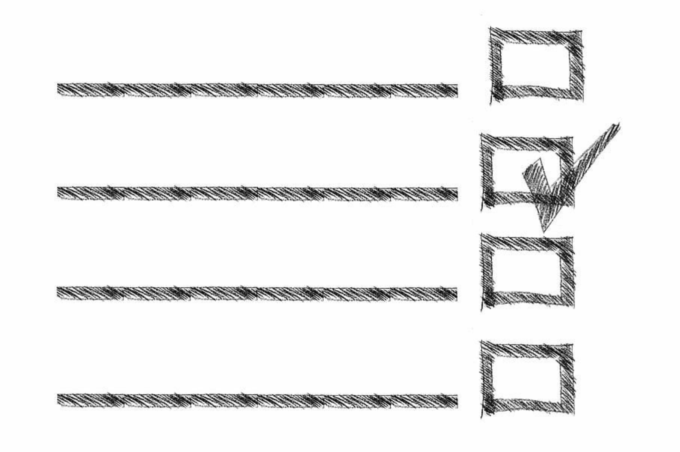
$$\boldsymbol{l}_i^T \boldsymbol{x}_i = 0$$
$$\boldsymbol{l}_i^T \boldsymbol{e}_i = 0$$

Example - Point Feature Matching

Example – Epipolar Lines



Properties of the Essential Matrix



Elementary properties of the Essential Matrix

$$E = T \times R \longrightarrow E = \begin{bmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{bmatrix}$$

A nonzero matrix E is an Essential Matrix if and only if E has a Singular Value Decompos (SVD):

$$E = U\Sigma V^T \begin{cases} \boxed{\Sigma = diag\{\sigma, \sigma, 0\}} \\ \boxed{U, V \in SO(3)} \end{cases}$$

U and V are Special orthogonal matrices: determinant is $+1$.

To understand why ...

Lemma 5.1 (The hat operator). *If* $T \in \mathbb{R}^3$, $A \in SL(3)$ *and* $T' = AT$, *then* $\widehat{T} = A^T \widehat{T'} A$.

Proof. Since both $A^T \widehat{(\cdot)} A$ and $\widehat{A^{-1}(\cdot)}$ are linear maps from $\mathbb{R}^3$ to $\mathbb{R}^{3\times 3}$, one may directly verify that these two linear maps agree on the basis $[1,0,0]^T, [0,1,0]^T$ or $[0,0,1]^T$ (using the fact that $A \in SL(3)$ implies that $\det(A) = 1$).

- For any essential matriz there is (at least one pair) $(R,\ T)$, $R \in SO(3)$, $T \in \mathbb{R}^3$, such that $\widehat{T}R = E$.

- For any essential matriz there is (at least one pair) $(R,\ T)$, $R \in SO(3)$, $T \in \mathbb{R}^3$, such that $\widehat{T}R = E$.
- For T there exists a rotation matrix R_0 such that $R_0 T = [0;\ 0;\ \|T\|]^T$.

Takes T to the Z-axis

- For any essential matriz there is (at least one pair) $(R,\ T)$, $R \in SO(3)$, $T \in \mathbb{R}^3$, such that $\widehat{T}R = E$.
- For T there exists a rotation matrix R_0 such that $R_0 T = [0;\ 0;\ \|T\|]^T$.
- Let's consider $a = R_0 T$ and since $\det(R_0) = 1 \rightarrow \widehat{T} = R_0^T \widehat{a} R_0$

Takes T to the Z-axis

- For any essential matriz there is (at least one pair) $(R,\ T)$, $R \in SO(3)$, $T \in \mathbb{R}^3$, such that $\widehat{T}R = E$.
- For T there exists a rotation matrix R_0 such that $R_0 T = [0;\ 0;\ \|T\|]^T$.
- Let's consider $a = R_0 T$ and since $\det(R_0) = 1 \rightarrow\ \widehat{T} = R_0^T \widehat{a} R_0$
- Then:

$$EE^T = \widehat{T}RR^T\widehat{T}^T = \widehat{T}\widehat{T}^T = R_0^T \widehat{a} R_0 R_0^T \widehat{a}^T R_0$$

Takes T to the Z-axis

- For any essential matriz there is (at least one pair) $(R,\ T)$, $R \in SO(3)$, $T \in \mathbb{R}^3$, such that $\widehat{T}R = E$.
- For T there exists a rotation matrix R_0 such that $R_0 T = [0;\ 0;\ \|T\|]^T$.
- Let's consider $a = R_0 T$ and since $\det(R_0) = 1 \rightarrow \widehat{T} = R_0^T \widehat{a} R_0$

Takes T to the Z-axis

- Then:

$$EE^T = \widehat{T}RR^T\widehat{T}^T = \widehat{T}\widehat{T}^T = R_0^T\widehat{a}\cancel{R_0}\cancel{R_0^T}\widehat{a}^T R_0 = R_0^T\,\widehat{a}\widehat{a}^T R_0$$

- And:

$$\widehat{a}\widehat{a}^T = \begin{bmatrix} 0 & -\|T\| & 0 \\ \|T\| & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & \|T\| & 0 \\ -\|T\| & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} \|T\|^2 & 0 & 0 \\ 0 & \|T\|^2 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Following...

OBS:

About SVD of a matrix M:

- The non-zero singular values of M (found on the diagonal entries of Σ) are the square roots of the non-zero eigenvalues of both M^TM and MM^T
- The left-singular vectors of M are a set of orthonormal eigenvectors of MM^T.
- The right-singular vectors of M are a set of orthonormal eigenvectors of M^TM.

Thus, from the previous slide: $EE^T = \widehat{T}RR^T\widehat{T}^T = \widehat{T}\widehat{T}^T = R_0^T\widehat{a}\widehat{a}^T R_0$.

$$\widehat{a}\widehat{a}^T = \begin{bmatrix} 0 & -\|T\| & 0 \\ \|T\| & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & \|T\| & 0 \\ -\|T\| & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} \|T\|^2 & 0 & 0 \\ 0 & \|T\|^2 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

The singular values of the essential matrix: $E = \widehat{T}R$ are $(\|T\|, \|T\|, 0)$.

A little bit more...

Now knowing that $E = \widehat{T}R = R_0^T \widehat{a} R_0 R$.

And considering a rotation around the Z-axis: $R_Z(+\frac{\pi}{2}) = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$.

We can write:

$$\widehat{a} = R_Z(+\tfrac{\pi}{2}) R_Z^T(+\tfrac{\pi}{2}) \widehat{a} = R_Z(+\tfrac{\pi}{2}) \mathrm{diag}\{\|T\|, \|T\|, 0\}.$$

Remember that

$$\widehat{a} = \begin{bmatrix} 0 & -\|T\| & 0 \\ \|T\| & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

A little bit more...

Now knowing that $E = \widehat{T}R = R_0^T \widehat{a} R_0 R$.

And considering a rotation around the Z-axis: $R_Z(+\frac{\pi}{2}) = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$.

We can write:

$$\widehat{a} = R_Z(+\tfrac{\pi}{2}) R_Z^T(+\tfrac{\pi}{2}) \widehat{a} = R_Z(+\tfrac{\pi}{2}) \mathrm{diag}\{\|T\|, \|T\|, 0\}.$$

Remember that

$$\widehat{a} = \begin{bmatrix} 0 & -\|T\| & 0 \\ \|T\| & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Therefore $E = \widehat{T}R = R_0^T R_Z(+\frac{\pi}{2}) \mathrm{diag}\{\|T\|, \|T\|, 0\} R_0 R$.

So, in the SVD of $E = U\Sigma V^T$ we can choose $U = R_0^T R_Z(+\frac{\pi}{2})$ and $V^T = R_0 R$.

Since U and V are constructed by the product of matrices in $SO(3)$, they are also in $SO(3)$, that is, they are rotation matrices.

Then...

If a given matrix $E \in \mathbb{R}^{3\times 3}$ has SVD: $E = U\Sigma V^T \begin{cases} U, V \in SO(3) \\ \Sigma = \mathrm{diag}\{\sigma, \sigma, 0\} \end{cases}$

We can define:

$$U = R_0^T R_Z(+\tfrac{\pi}{2}) \qquad V^T = R_0 R$$

$$R_0^T = U R_Z^T(+\tfrac{\pi}{2}) \qquad R = R_0^T V^T$$

Then...

If a given matrix $E \in \mathbb{R}^{3\times 3}$ has SVD: $E = U\Sigma V^T$ $\begin{cases} U, V \in SO(3) \\ \Sigma = \mathrm{diag}\{\sigma, \sigma, 0\} \end{cases}$

We can define:

$$U = R_0^T R_Z(+\tfrac{\pi}{2}) \qquad V^T = R_0 R$$

$$R_0^T = U R_Z^T(+\tfrac{\pi}{2}) \qquad R = R_0^T V^T$$

$$\widehat{T} = R_0^T \widehat{a} R_0 \longrightarrow \widehat{T} = U R_Z^T(+\tfrac{\pi}{2}) R_Z(+\tfrac{\pi}{2}) \Sigma R_Z(+\tfrac{\pi}{2}) U^T$$

$$\widehat{a} = R_Z(+\tfrac{\pi}{2})\Sigma$$

Then...

If a given matrix $E \in \mathbb{R}^{3\times 3}$ has SVD: $E = U\Sigma V^T \begin{cases} U, V \in SO(3) \\ \Sigma = \mathrm{diag}\{\sigma, \sigma, 0\} \end{cases}$

We can define:

$$U = R_0^T R_Z(+\tfrac{\pi}{2}) \qquad V^T = R_0 R$$

$$R_0^T = U R_Z^T(+\tfrac{\pi}{2}) \qquad R = R_0^T V^T$$

$$\widehat{T} = R_0^T \widehat{a} R_0 \longrightarrow \widehat{T} = U \cancel{R_Z^T(+\tfrac{\pi}{2}) R_Z(+\tfrac{\pi}{2})} \Sigma R_Z(+\tfrac{\pi}{2}) U^T$$

$$\widehat{a} = R_Z(+\tfrac{\pi}{2})\Sigma \qquad \widehat{T} = U\Sigma R_Z(+\tfrac{\pi}{2})U^T$$

$$\boxed{\widehat{T} = U R_Z(+\tfrac{\pi}{2})\Sigma U^T}$$

Then...

If a given matrix $E \in \mathbb{R}^{3\times 3}$ has SVD: $E = U\Sigma V^T$ $\begin{cases} U, V \in SO(3) \\ \Sigma = \mathrm{diag}\{\sigma, \sigma, 0\} \end{cases}$

We can define:

$$U = R_0^T R_Z(+\tfrac{\pi}{2}) \qquad V^T = R_0 R$$

$$R_0^T = U R_Z^T(+\tfrac{\pi}{2}) \qquad R = R_0^T V^T$$

$$\widehat{T} = R_0^T \widehat{a} R_0 \longrightarrow \widehat{T} = U \cancel{R_Z^T(+\tfrac{\pi}{2}) R_Z(+\tfrac{\pi}{2})} \Sigma R_Z(+\tfrac{\pi}{2}) U^T$$

$$\widehat{a} = R_Z(+\tfrac{\pi}{2})\Sigma \qquad \widehat{T} = U\Sigma R_Z(+\tfrac{\pi}{2}) U^T$$

$$\boxed{\widehat{T} = U R_Z(+\tfrac{\pi}{2}) \Sigma U^T} \qquad \boxed{R = U R_Z^T(+\tfrac{\pi}{2}) V^T}$$

Finally

What we have done with $+\frac{\pi}{2}$ also works with $-\frac{\pi}{2}$.

Therefore we have two possible solutions for (R, T).

$$E = U\Sigma V^T \qquad \begin{cases} (\widehat{T}_1, R_1) = (UR_Z(+\frac{\pi}{2})\Sigma U^T, UR_Z^T(+\frac{\pi}{2})V^T), \\ (\widehat{T}_2, R_2) = (UR_Z(-\frac{\pi}{2})\Sigma U^T, UR_Z^T(-\frac{\pi}{2})V^T). \end{cases}$$

that $\widehat{T}_1 R_1 = \widehat{T}_2 R_2 = E$.

Example 5.8 (Two solutions to an essential matrix). It is immediate to verify that $\widehat{e_3} R_Z\left(+\frac{\pi}{2}\right) = \widehat{-e_3} R_Z\left(-\frac{\pi}{2}\right)$, since

$$\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

These two solutions together are usually referred to as a "twisted pair", due to the manner in which the two solutions are related geometrically, as illustrated in Figure 5.3. A physically correct solution can be chosen by enforcing that the reconstructed points be visible, i.e. they have positive depth. We discuss this issue further in Exercise 5.11.

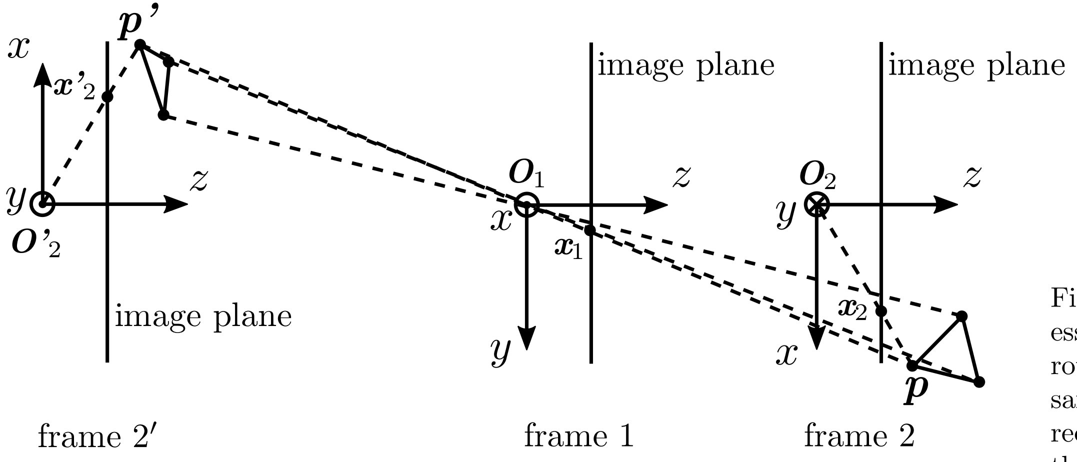


Figure 5.3. Two pairs of camera frames, i.e. $(1, 2)$ and $(1, 2')$, generate the same essential matrix. The frame 2 and frame 2′ differ by a translation and a 180° rotation (a twist) around the Z-axis, and the two pose pairs give rise to the same image coordinates. For the same set of image pairs $\boldsymbol{x}_1$ and $\boldsymbol{x}_2 = \boldsymbol{x}'_2$, the recovered structures $\boldsymbol{p}$ and $\boldsymbol{p}'$ might be different. Notice that with respect to the camera frame 1, the point $\boldsymbol{p}'$ has a negative depth.

Is there only two solutions?

- Given a rotation matrix R and a translation vector T, it is immediate to construct an essential matrix $E = \widehat{T}R$.
- The inverse problem, that is how to retrieve T and R from a given essential matrix E, is less obvious.
- To proof the properties of E, we have used the SVD to construct two solutions for (R, T).
- **Are these the only solutions?**

Because the sign of E is arbitrary...

- It is assumed that in the SVD of E both matrices U and V are rotation matrices in $SO(3)$. This is not always true when E is estimated from noisy data.
- In fact, standard SVD routines do not guarantee that the computed U and V have $\det = +1$.
- The sign of the essential matrix E is also arbitrary (even after normalization).
- It can operate either on $+E$ or -E.
- **One of the matrices $\pm E$ will always have an SVD that satisfy the conditions**

$$E = U\Sigma V^T \quad \begin{cases} \boxed{\Sigma = diag\{\sigma, \sigma, 0\}} \\ \boxed{U, V \in SO(3)} \end{cases}$$

Thus...

- Each normalized essential matrix E gives two possible poses (R, T).
- From E, we can recover the pose up to four solutions.
- In fact, three of the solutions can be eliminated by imposing the positive depth constraint.
- **To account for the possible sign change with E, the "+" and "-" signs in the equations for R and T should be arbitrarily combined so that all four solutions can be obtained.**

$$\boxed{R = U R_Z^T\left(\pm\tfrac{\pi}{2}\right) V^T, \qquad \widehat{T} = U R_Z\left(\pm\tfrac{\pi}{2}\right) \Sigma U^T}$$

Example

Suppose that

$$R = \begin{bmatrix} \cos\left(\frac{\pi}{4}\right) & 0 & \sin\left(\frac{\pi}{4}\right) \\ 0 & 1 & 0 \\ -\sin\left(\frac{\pi}{4}\right) & 0 & \cos\left(\frac{\pi}{4}\right) \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \\ 0 & 1 & 0 \\ -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \end{bmatrix}, \qquad T = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}$$

The essential matrix is $E = \widehat{T}R = \begin{bmatrix} 0 & 0 & 0 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 0 & 2 & 0 \end{bmatrix}$

Since $\|T\| = 2$,the E obtained here is not normalized. It is also easy to see this from its SVD,

$$E = U\Sigma V^T \doteq \begin{bmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -\frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} \\ 0 & 1 & 0 \\ \frac{\sqrt{2}}{2} & 1 & -\frac{\sqrt{2}}{2} \end{bmatrix}^T,$$

where the nonzero singular values are 2 instead of 1. Normalizing E is equivalent to replacing the above Σ by

$$\Sigma = \mathrm{diag}\{1, 1, 0\}.$$

It is then easy to compute the four possible decompositions $(R, \widehat{T})$ for E:

1. $UR_Z^T\left(+\frac{\pi}{2}\right)V^T = \begin{bmatrix} \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \\ 0 & -1 & 0 \\ \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} \end{bmatrix}, \; UR_Z\left(+\frac{\pi}{2}\right)\Sigma U^T = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix};$

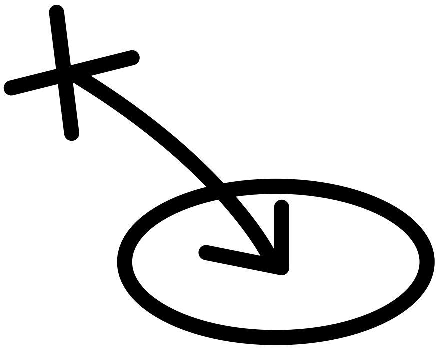
2. $UR_Z^T\left(+\frac{\pi}{2}\right)V^T = \begin{bmatrix} \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \\ 0 & -1 & 0 \\ \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} \end{bmatrix}, \; UR_Z\left(-\frac{\pi}{2}\right)\Sigma U^T = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix};$

3. $UR_Z^T\left(-\frac{\pi}{2}\right)V^T = \begin{bmatrix} \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \\ 0 & 1 & 0 \\ -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \end{bmatrix}, \; UR_Z\left(-\frac{\pi}{2}\right)\Sigma U^T = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix};$

4. $UR_Z^T\left(-\frac{\pi}{2}\right)V^T = \begin{bmatrix} \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \\ 0 & 1 & 0 \\ -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \end{bmatrix}, \; UR_Z\left(+\frac{\pi}{2}\right)\Sigma U^T = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}.$

Clearly, the third solution is exactly the original motion $(R, \widehat{T})$ except that the translation T is recovered up to a scalar factor (i.e. it is normalized to unit norm).

Estimating the Essential Matrix



Partially Calibrated Stereo System

- The intrinsic parameters of both cameras are known.
- Estimate the Essential Matrix and then recover the 3D structure.

Estimate the Essential Matrix E (Rotation and Translation between cameras) that minimizes the epipolar error.

Ideal $\boldsymbol{x}_2^T E \boldsymbol{x}_1 = 0$ $\longrightarrow$ $\min_E \sum_{j=1}^{n} \boldsymbol{x}_2^{j\,T} E \boldsymbol{x}_1^j$

$$E = \begin{bmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{bmatrix}$$

Eight unkowns up to a scalar factor.
Each pair of correspondent points provides an epipolar equation $\rightarrow$ at least eight pairs of points are needed.

Eight point algorithm

1. Compute a first approximation of the essential matrix E that minimizes the epipolar error.
   - For a set of correspondent points (n>= 8) write down the epipolar equation for each pair and stack into a matrix.

$$\boldsymbol{a} = \boldsymbol{x}_1 \otimes \boldsymbol{x}_2 \longrightarrow \text{Kronecker Product}$$

$$\boldsymbol{a} = [x_1x_2, x_1y_2, x_1z_2, y_1x_2, y_1y_2, y_1z_2, z_1x_2, z_1y_2, z_1z_2]^T$$

$$E^S = [e_1, e_4, e_7, e_2, e_5, e_8, e_3, e_6, e_9]^T$$

$$\boldsymbol{x}_2^T E \boldsymbol{x}_1 = 0 \xrightarrow[\text{inner product}]{\text{Represented as an}} \boldsymbol{a}^T E^S = 0$$

For n pairs arranged into a matrix X.

$$\boxed{XE^S = 0}$$

Eight point algorithm

$$\min_{E} \sum_{j=1}^{n} \boldsymbol{x}_2^{jT} E \boldsymbol{x}_1^j \xrightarrow{\text{That is the same of}} \min_{E^S} \|X E^S\|^2$$

Find the eigenvector associated with the smaller eigenvalue of $X^T X$

Problem: $\mathrm{rank}(X^T X) < 8$

Compute the SVD of $X = U \Sigma V^T$ and $\longrightarrow$ First estimation of E

E^s is the 9th column of V

Eight point algorithm

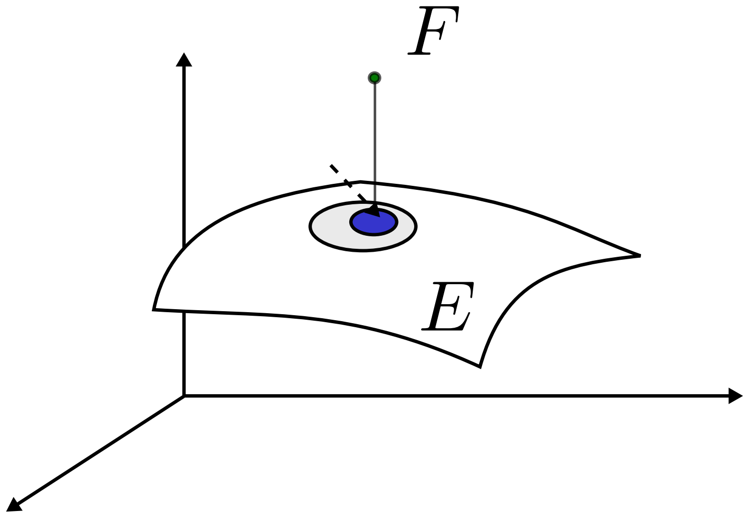
2. SVD of E $\longrightarrow$ $\Sigma = \mathrm{diag}([\sigma_1, \sigma_2, \sigma_3])$

$$\Sigma' = \mathrm{diag}(1, 1, 0)$$

Projection onto the normalized Essential Space

$$E = U\Sigma' V^T$$

Normalized Essential Matrix E



Eight point algorithm

3. Recover R and T from E

$$R = U R_Z^T\left(\pm\tfrac{\pi}{2}\right) V^T, \qquad \widehat{T} = U R_Z\left(\pm\tfrac{\pi}{2}\right) \Sigma U^T$$

- Four possible mathematical solutions: 4 pairs (R, T)
  - 2 pairs associated with -E
  - 2 pairs associated with $+E$

- Positive Depth Constraint: only one solution leads to 3D reconstruted points with positive depths with respect to both camera frames $\rightarrow$ (positive Z coordinates)

- Translation must not be zero
- Points must be in general positions

Problem $\rightarrow$

Degenerate Configurations:
- Coplanar Points
- Quadratic Surfaces

Algorithm 5.1 (The eight-point algorithm).

For a given set of image correspondences $(\boldsymbol{x}_1^j, \boldsymbol{x}_2^j), j = 1, 2, \ldots, n (n \geq 8)$ this algorithm recovers $(R, T) \in SE(3)$, which satisfy

$$\boldsymbol{x}_2^{jT} \widehat{T} R \boldsymbol{x}_1^j = 0, \quad j = 1, 2, \ldots, n.$$

1. **Compute a first approximation of the essential matrix**

   Construct $X = \left[\boldsymbol{a}^1, \boldsymbol{a}^2, \ldots, \boldsymbol{a}^n\right]^T \in \mathbb{R}^{n \times 9}$ from correspondences $\boldsymbol{x}_1^j$ and $\boldsymbol{x}_2^j$ as in (5.12), namely,
   $$\boldsymbol{a}^j = \boldsymbol{x}_1^j \otimes \boldsymbol{x}_2^j \quad \in \mathbb{R}^9.$$

   Find the vector $E^S \in \mathbb{R}^9$ of unit length such that $\|XE^S\|$ is minimized as follows: compute the SVD of $X = U_X \Sigma_X V_X^T$ and define E^S to be the ninth column of V_X. Unstack the nine elements of E^S into a square 3×3 matrix E as in (5.10). Note that this matrix will in general *not* be in the essential space.

2. **Project onto the essential space**

   Compute the singular value decomposition of the matrix E recovered from data to be
   $$E = U \mathrm{diag}\{\sigma_1, \sigma_2, \sigma_3\} V^T,$$

   where $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq 0$ and $U, V \in SO(3)$. In general, since E may not be an essential matrix, $\sigma_1 \neq \sigma_2$ and $\sigma_3 \neq 0$. But its projection onto the normalized essential space is $U \Sigma V^T$, where $\Sigma = \mathrm{diag}\{1, 1, 0\}$.

3. **Recover the displacement from the essential matrix**

   We now need only U and V to extract R and T from the essential matrix as
   $$R = U R_Z^T\left(\pm\frac{\pi}{2}\right) V^T, \quad \widehat{T} = U R_Z\left(\pm\frac{\pi}{2}\right) \Sigma U^T.$$

   where $R_Z^T\left(\pm\frac{\pi}{2}\right) \doteq \begin{bmatrix} 0 & \pm 1 & 0 \\ \mp 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$.

3D Structure Reconstruction

$$\lambda_2^j \boldsymbol{x}_2^j = \lambda_1^j R\, \boldsymbol{x}_1^j + \gamma T$$

Translation scale factor that is unkown

Eliminate one of the structural scales (depths) $\rightarrow$ cross product with $\boldsymbol{x}_2^j$

$$\cancelto{0}{\lambda_2^j \hat{\boldsymbol{x}}_2^j \boldsymbol{x}_2^j} = \lambda_1^j \hat{\boldsymbol{x}}_2^j R \boldsymbol{x}_1^j + \gamma \hat{\boldsymbol{x}}_2^j T \qquad \text{with} \quad j = 1 \ldots n$$

$$\lambda_1^j \hat{\boldsymbol{x}}_2^j R \boldsymbol{x}_1^j + \gamma \hat{\boldsymbol{x}}_2^j T = 0$$

The equation $\lambda_1^j \widehat{\boldsymbol{x}_2^j} R \boldsymbol{x}_1^j + \gamma \widehat{\boldsymbol{x}_2^j} T = 0$

Is equivalent to $M^j \vec{\lambda_1^j} = \left[\widehat{\boldsymbol{x}_2^j} R \boldsymbol{x}_1^j, \widehat{\boldsymbol{x}_2^j} T \right] \begin{bmatrix} \lambda_1^j \\ \gamma \end{bmatrix} = 0$

With $M^j = \left[\widehat{\boldsymbol{x}_2^j} R \boldsymbol{x}_1^j, \widehat{\boldsymbol{x}_2^j} T \right]$ and $\vec{\lambda_1^j} = \begin{bmatrix} \lambda_1^j \\ \gamma \end{bmatrix}$

$$\vec{\lambda} = \left[\lambda_1^1, \lambda_1^2, \ldots, \lambda_1^n, \gamma \right] \in \mathbb{R}^{n+1} \text{ and a matrix } M \in \mathbb{R}^{3n \times (n+1)} \text{ as}$$

$$M\vec{\lambda} = 0 \qquad M \doteq \begin{bmatrix} \widehat{\boldsymbol{x}_2^1} R \boldsymbol{x}_1^1 & 0 & 0 & 0 & \widehat{\boldsymbol{x}_2^1} T \\ 0 & \widehat{\boldsymbol{x}_2^2} R \boldsymbol{x}_1^2 & 0 & 0 & \widehat{\boldsymbol{x}_2^2} T \\ 0 & 0 & \ddots & 0 & \vdots \\ 0 & 0 & \widehat{\boldsymbol{x}_2^{n-1}} R \boldsymbol{x}_1^{n-1} & 0 & \widehat{\boldsymbol{x}_2^{n-1}} T \\ 0 & 0 & 0 & \widehat{\boldsymbol{x}_2^n} R \boldsymbol{x}_1^n & \widehat{\boldsymbol{x}_2^n} T \end{bmatrix}$$

We can now find the Linear Least-Squares Estimate for the parameter vector $\vec{\lambda}$, which is the eigenvector of $M^T M$ that corresponds to its smallest eigenvalue. Or we can apply another optimization method to find a better solution.

3D Structure Reconstruction

Another way is to compute the inner product of two parallel vectors with the same direction, so the result is positive, and use that to estimate λ_1^j or λ_2^j, with j = 1 .. n.

For λ_1^j :

$$\lambda_2^j \boldsymbol{x}_2^j = \lambda_1^j R \boldsymbol{x}_1^j + T$$

In this case we do not recover the translation scale factor. It is considered 1.

$$\lambda_2^j \overbrace{\hat{\boldsymbol{x}}_2^j \boldsymbol{x}_2^j}^{\mathbf{0}} = \lambda_1^j \hat{\boldsymbol{x}}_2^j R \boldsymbol{x}_1^j + \hat{\boldsymbol{x}}_2^j T$$

Cross product with $\boldsymbol{x}_2^j$

$$\lambda_1^j \hat{\boldsymbol{x}}_2^j R \boldsymbol{x}_1^j = -\hat{\boldsymbol{x}}_2^j T$$

Inner product with $\hat{\boldsymbol{x}}_2^j T$

$$\lambda_1^j (\hat{\boldsymbol{x}}_2^j R \boldsymbol{x}_1^j)^T (\hat{\boldsymbol{x}}_2^j T) = -(\hat{\boldsymbol{x}}_2^j T)^T (\hat{\boldsymbol{x}}_2^j T)$$

$$\boxed{\frac{1}{\lambda_1^j} = \frac{-(\hat{\boldsymbol{x}}_2^j R \boldsymbol{x}_1^j)^T (\hat{\boldsymbol{x}}_2^j T)}{\|\hat{\boldsymbol{x}}_2^j T\|^2}}$$

That should be done for all correspondences.

For $\lambda_2{}^{j}$:

$$\lambda_2^j \boldsymbol{x}_2^j = \lambda_1^j R\boldsymbol{x}_1^j + T$$

$$\lambda_2^j \widehat{R\boldsymbol{x}_1^j}\boldsymbol{x}_2^j = \lambda_1^j \cancelto{0}{\widehat{R\boldsymbol{x}_1^j} R\boldsymbol{x}_1^j} + \widehat{R\boldsymbol{x}_1^j} T$$

$$\lambda_2^j \widehat{R\boldsymbol{x}_1^j}\boldsymbol{x}_2^j = \widehat{R\boldsymbol{x}_1^j} T$$

$$\lambda_2^j (\widehat{R\boldsymbol{x}_1^j}\boldsymbol{x}_2^j)^T (\widehat{R\boldsymbol{x}_1^j}\boldsymbol{x}_2^j) = (\widehat{R\boldsymbol{x}_1^j} T)^T (\widehat{R\boldsymbol{x}_1^j}\boldsymbol{x}_2^j)$$

In this case we do not recover the translation scale factor. It is considered 1.

Cross product with $R\boldsymbol{x}_1{}^{j}$

Inner product with $\widehat{R\boldsymbol{x}_1^j}\boldsymbol{x}_2^j$

$$\boxed{\lambda_2^j = \frac{(\widehat{R\boldsymbol{x}_1^j} T)^T (\widehat{R\boldsymbol{x}_1^j}\boldsymbol{x}_2^j)}{\|\widehat{R\boldsymbol{x}_1^j}\boldsymbol{x}_2^j\|^2}}$$

That should be done for all correspondences.

3D Structure Reconstruction

- Finally, the solutions obtained from all four pairs of (R, T) are tested.
- Only one guarantees the positive depths for all 3D reconstructed points.
- Atention: scale ambiguity is instrinsic, since without any prior knowledge about the scene and camera motion, one cannot disambiguate whether the camera moved twice the distance while looking at a scene twice larger but two times further away.

Credits

- Yi Ma, Stefano Soatto, Jana Kosecka e S. Shankar Sastry.
  An Invitation to 3D Vision: From Images to Geometric Models.
  Springer, ISBN 0387008934