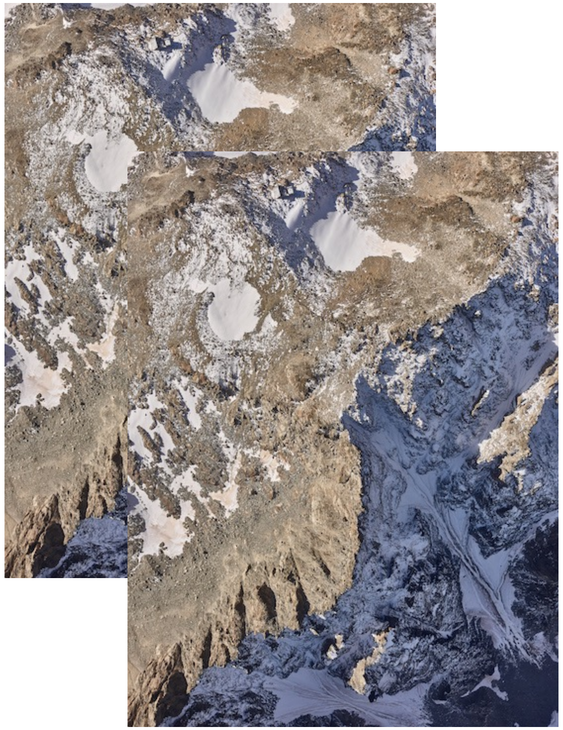
Lecture 07
Two View Geometry

ENV408: Optical Sensing & Modeling for Earth Observations

Jan Skaloud

Ex 4 – Relative orientation

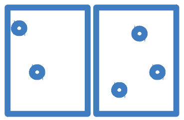


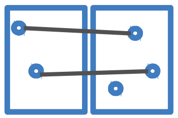
- Tomorrow: use matched key-points (Ex.2) corrected for image distortion (Ex.1) to orient two images (relative pose, rel. EO p.) and triangulate key-point coordinates in 3D.

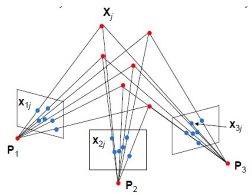
- Today: understand how to reconstruct simultaneously 3D scene structure and camera pose (up to scale) from multiple images.

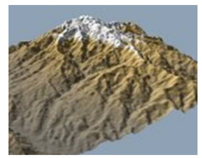
flight plan → flight → trajectory (ins/gnss) → images

feature detection → feature matching → optimization (orientation, calibration) → product(s) (e.g. surface, ortho-photo)









in the course
+ info extraction

- Lectures
  - Image primes (L1)
  - Salient features (L2)
  - Image orientation (L3)
  - Multiple views, optimization (L4)
  - Mapping products (L5)

- Exercises
  - Image 'corrections' (Ex1)
  - Detection & matching (Ex2)
  - Absolute pose (Ex3)
  - Relative pose (Ex4)
  - Calibration, DEM, ortho-photo (Ex5)

Outline

- Epipolar geometry
- Essential and fundamental matrices
- 8-point algorithm

Terminology - Computer Vision (CV)

- Structure from motion (SFM) – pose, 3D, calib.
- Multiple view geometry (matching + SFM)
- Optimization

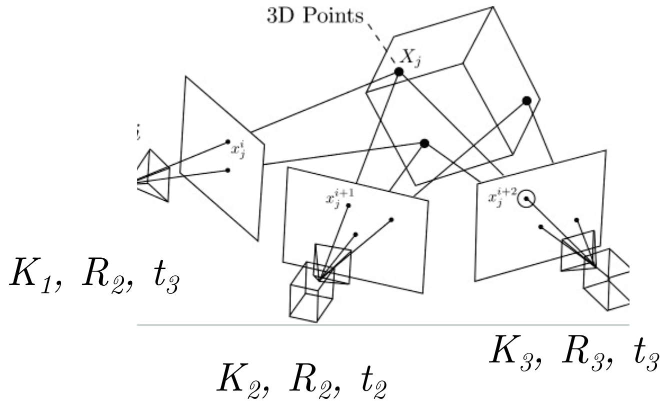
Terminology - Photogrammetry

- Relative orientation (with calibration)
- Triangulation, scene reconstruction
- Bundle adjustment

Two or Multiple view geometry - recapitulation

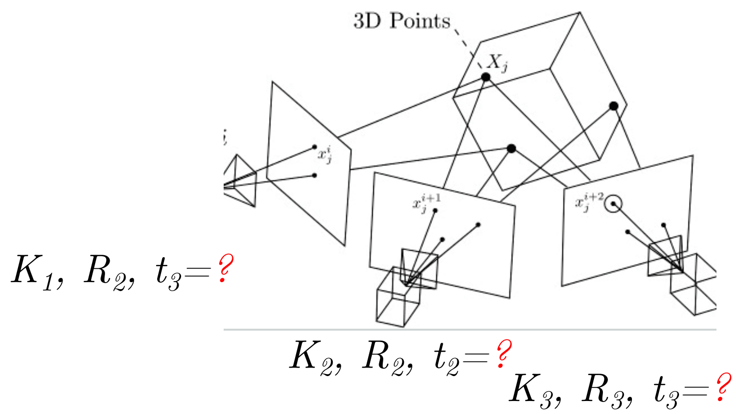
<u>Depth from stereo vision (3D reconstruction)</u>

- Assumptions: camera calibrated & oriented (i.e. **known** $K_i,\ R_i,\ t_i$)
- Goal: recover the 3D structure from images



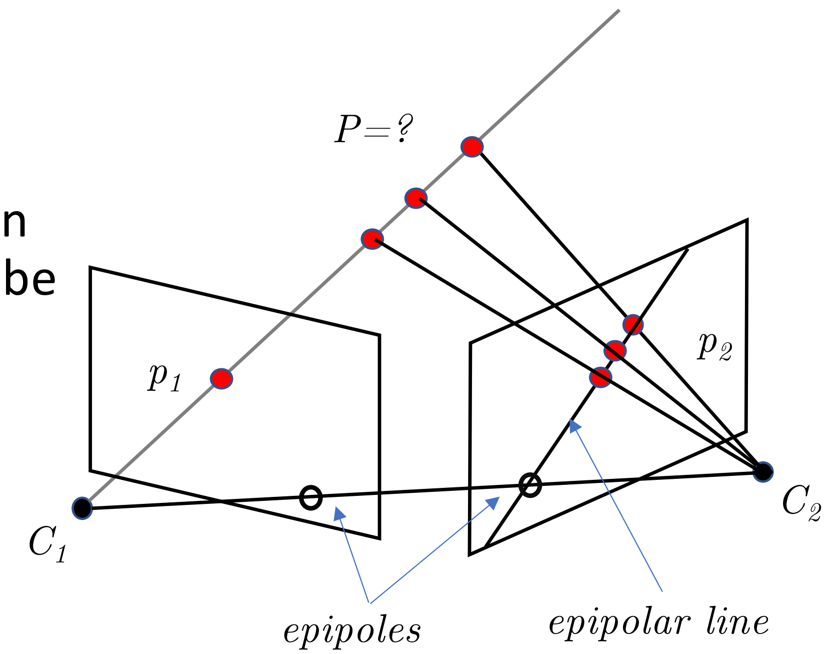
<u>Structure from motion</u> (SFM)

- Assumpitons: **unknown** $K_i,\ R_i,\ t_i$
- Goal: recover simultaneously scene structure (3D) and camera pose (up to scale)



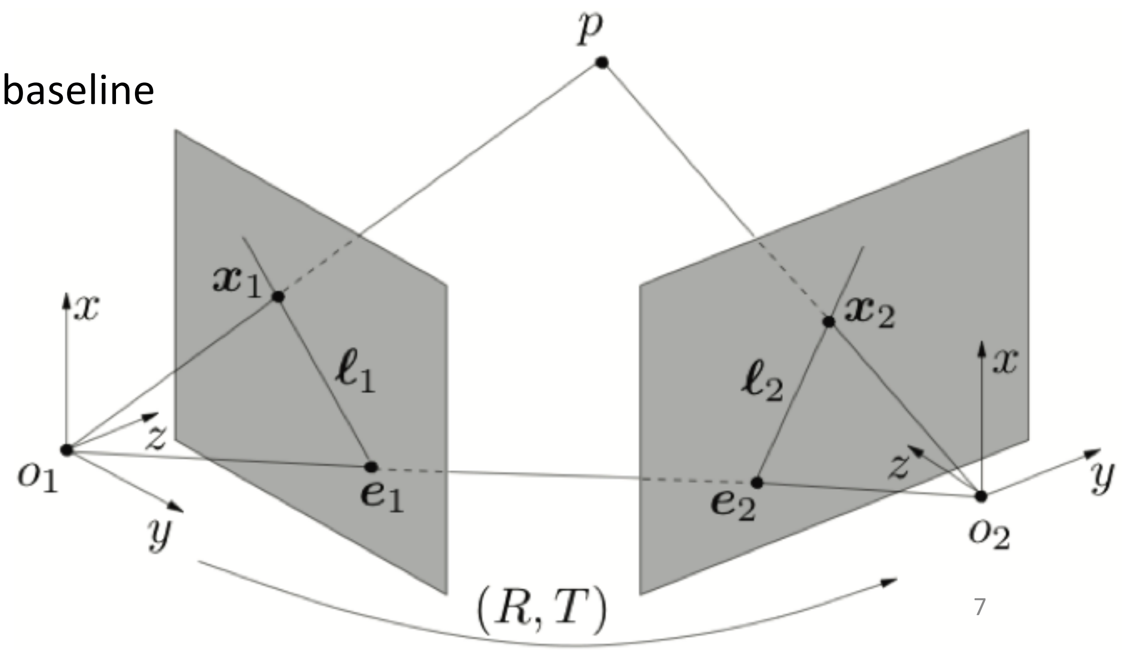
Correspondence problem

- Triangulation prerequisits
  - the pose (R and t) is known (at least relatively)
  - Image correspondences exist for a set of points $P_i \quad i=1 \ \ldots \ n$
- Problem
  - Given a point on the left-image, p_L , how can its correspondence, p_R , on the right image be determined?
  - 2D exhaustive search is very expensive (computationally)
  - Potential matches have to lie on the corresponding epipolar line!



Coplanarity constraint (epipolar geometry)

- Camera centers and one image point defines: **epipolar plane**
- Intersection of epipolar plane with 2 image planes are: **epipolar lines**
- **Epipolar constraint**: a fact that corresponding point lies on epipolar line
- Formulation:
  - via epipolar lines
  - Coplanarity between image vectors and baseline



Coplanarity constraint (epipolar geometry)

- The 3 vectors t, x_1 , x_2 , must be coplanar
- Volume of parallelepiped spanned by them (vector triple product) = 0
- Considering skew-symmetric matrix $[t_\times]$ and:

$$X_1 = \mu_1 x_1\,,\;\; X_2 = \mu_2 x_2$$

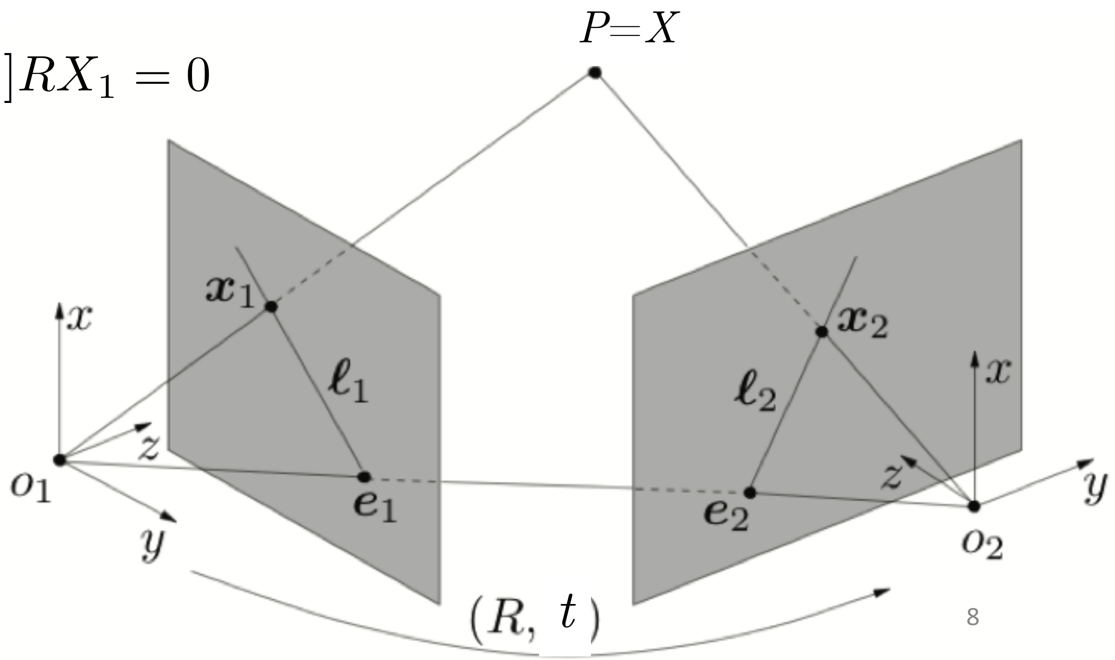
$$X_2 \cdot ([t\times]RX_1) = X_2^T\,([t_\times]RX_1) = X_2^T[t_\times]RX_1 = 0$$

$$x_2^T[t_\times]Rx_1 = 0$$

$$x_2^T E x_1 = 0$$

with

$$E = [t_\times]R$$



Outline

- Epipolar geometry
- Essential and fundamental matrices
- 8-point algorithm

Terminology - Computer Vision (CV)

- Structure from motion (SFM) – pose, 3D, calib.
- Multiple view geometry (matching + SFM)
- Optimization

Terminology - Photogrammetry

- Relative orientation (with calibration)
- Triangulation, scene reconstruction
- Bundle adjustment

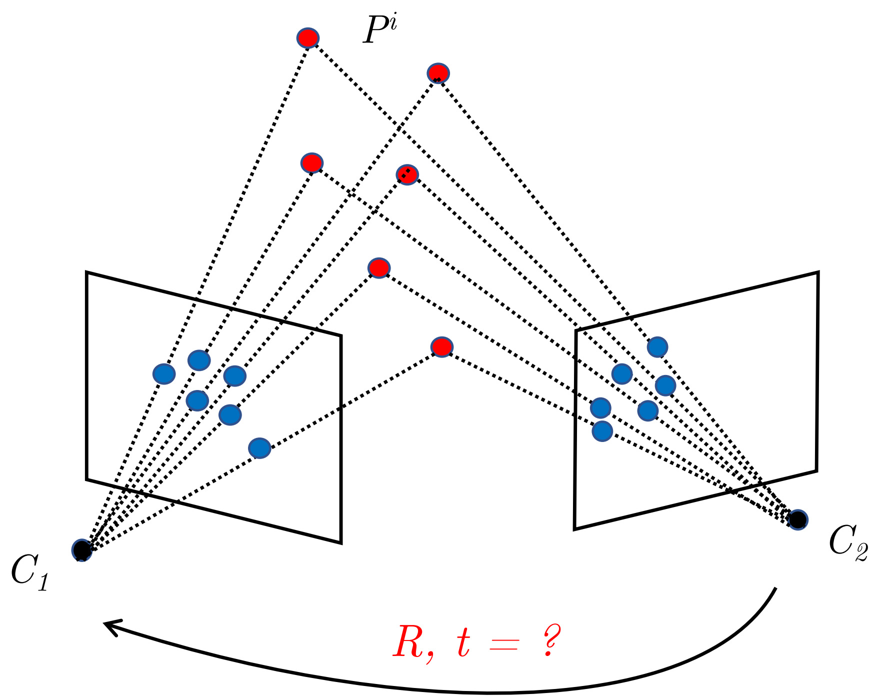
Relative orientation (Structure from Motion-SFM)

Given a set of i=(1..n) point correspondences for 2 images, $p_1^i = (u_1^i,\, v_1^i),\; p_2^i = (u_2^i,\, v_2^i)$ <u>estimate simultaneously</u>:

- The 3D points P^i
- The camera relative-orientation/pose ($R,\; t$)
- Camera intrinsic $K_1,\; K_2,$ satisfying:

$$\mu_1^i \begin{pmatrix} u_1^i \\ v_1^i \\ 1 \end{pmatrix} = K_1[I\,|\,0] \cdot \begin{pmatrix} X^i \\ Y^i \\ Z^i \\ 1 \end{pmatrix}$$

$$\mu_2^i \begin{pmatrix} u_2^i \\ v_2^i \\ 1 \end{pmatrix} = K_2[R\,|\,t] \cdot \begin{pmatrix} X^i \\ Y^i \\ Z^i \\ 1 \end{pmatrix}$$



Epipolar geometry – calibrated camera

- Normalized coordinates

$$\begin{pmatrix} \bar{u}_1 \\ \bar{v}_1 \\ 1 \end{pmatrix} = K_1^{-1} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} \qquad \begin{pmatrix} \bar{u}_2 \\ \bar{v}_2 \\ 1 \end{pmatrix} = K_2^{-1} \begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} \qquad K = \begin{pmatrix} c_u & 0 & u_0 \\ 0 & c_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}$$

$x_2^T E x_1 = 0$ (notation polycopié)

$p_2^T E p_1 = 0$ (same thing)

$$\begin{pmatrix} \bar{u}_2 \\ \bar{v}_2 \\ 1 \end{pmatrix}^T E \begin{pmatrix} \bar{u}_1 \\ \bar{v}_1 \\ 1 \end{pmatrix} = 0$$

Essential matrix $E = [t_\times]R$

Epipolar geometry – uncalibrated camera

- Previously

$$\begin{pmatrix} \bar{u}_1 \\ \bar{v}_1 \\ 1 \end{pmatrix} = K_1^{-1} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} \qquad \begin{pmatrix} \bar{u}_2 \\ \bar{v}_2 \\ 1 \end{pmatrix} = K_2^{-1} \begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix}$$

$$p_2^T E p_1 = 0$$

- Without the knowledge of K

$$\begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix}^T \boxed{K_2^{-T} E K_1^{-1}} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} = 0$$

$$\begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix}^T F \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} = 0$$

- Fundamental matrix

$$F = \left(K_2^T\right)^{-1} E K_1^{-1}$$

Epipolar geometry – system of equations

Each pair of point correspondences $\bar{p}_1 = (\bar{u}_1, \bar{v}_1, 1)^T, \quad \bar{p}_2 = (\bar{u}_2, \bar{v}_2, 1)^T$
provides a linear equation*:

$$p_2^T E p_1 = 0 \qquad\qquad E = \begin{pmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{pmatrix}$$

$$\bar{u}_2\bar{u}_1 e_{11} + \bar{u}_2\bar{v}_1 e_{12} + \bar{u}_2 e_{13} + \bar{v}_2\bar{u}_1 e_{21} + \bar{v}_2\bar{v}_1 e_{22} + \bar{v}_2 e_{23} + \bar{u}_1 e_{31} + \bar{v}_1 e_{32} + e_{33} = 0$$

Given ‘enough’ correspondences, E (or F) can be obtained

- What is the minimum number of correspondences ?
- Can R, t can be recovered from E ?
- (In more general case, can R, t, K_1, K_2 be recovered from F ?)

* Omitting the bar symbol over p

Epipolar geometry – inverse problem for E

- How many <u>knowns</u> per n ?
  - per correspondence:
  - per n :

- How many <u>unknowns</u> per n ?
  - per correspondence:
  - general:
  - together:

- When a solution exist?

The 8-point algorithm – formation of constraints

- If for 1 point, we have from $p_2^T E p_1 = 0$

$$\bar{u}_2\bar{u}_1 e_{11} + \bar{u}_2\bar{v}_1 e_{12} + \bar{u}_2 e_{13} + \bar{v}_2\bar{u}_1 e_{21} + \bar{v}_2\bar{v}_1 e_{22} + \bar{v}_2 e_{23} + \bar{u}_1 e_{31} + \bar{v}_1 e_{32} + e_{33} = 0$$

- For n points (when omitting bars)

$$\underbrace{\begin{pmatrix} u_2^1 u_1^1 & u_2^1 v_1^1 & u_2^1 & v_2^1 u_1^1 & v_2^1 v_1^1 & v_2^1 & u_1^1 & v_1^1 & 1 \\ u_2^2 u_1^2 & u_2^2 v_1^2 & u_2^2 & v_2^2 u_1^2 & v_2^2 v_1^2 & v_2^2 & u_1^2 & v_1^2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_2^n u_1^n & u_2^n v_1^n & u_2^n & v_2^n u_1^n & v_2^n v_1^n & v_2^n & u_1^n & v_1^n & 1 \end{pmatrix}}_{Q \text{ (known)}} \begin{pmatrix} e_{11} \\ e_{12} \\ e_{13} \\ e_{21} \\ e_{22} \\ e_{23} \\ e_{31} \\ e_{32} \\ e_{33} \end{pmatrix} = 0 \qquad Q \cdot E_s = 0$$

E_s (stacked E - unknown)

The 8-point algorithm – finding E

Minimum solution $Q \cdot E_s = 0$

- $Q_{(n \times 9)}$ - a unique (up to a scale) solution is possible if matrix rank = …. ?
- Each correspondence gives 1 independent equation.
- Hence, … correspondences (non-planar) needed

Over-determined solution ($n >$?)

- By minimizing $||Q \cdot E_s||^2 = E_s^T Q^T Q E_s$ subject to constraint $||E_s||^2 = 1$
- Solution E_s is an eigenvector corresponding to the smallest eigen value of $Q^T Q$
- Singular value decomposition (SVD) – in Matlab:

```
[U,S,V] = svd(Q^2);
Es = V(:,9);
E = reshape(Es,3,3)';
```

The 8-point algorithm – SVD of Q in Python

```
Q = np.zeros((num_points,9))
    for i in range(num_points):
        Q[i,:] = np.kron( p1[:,i], p2[:,i] ).T

_, _, Vt= np.linalg.svd(Q, full_matrices = False)
     E = np.reshape(Vt[-1,:], (3,3)).T
```

Extracting $R, \ t$ from E

1) Enforcing E to be in the "E-space"

- Singular value decomposition $E = U \Sigma V^T$
- "In case of no-errors": $\Sigma = diag\left(\sigma, \sigma, 0\right)$
- Due to errors: $\tilde{E} = U\, diag\left(\sigma_1, \sigma_2, \sigma_3\right) V^T, \quad \sigma_1 \geq \sigma_2 \geq \sigma_3$
- Choosing $\hat{E} = U\, diag\left(\sigma, \sigma, 0\right) V^T, \quad \sigma = (\sigma_1 + \sigma_2)/2$
- … satisfies E-space, but there could be another E leading to a smaller $||Q \cdot E_s||^2$
- Python

```
# Enforce det(E)=0 by projecting E on a set of 3x3 orthogonal matrices
U, S, Vt = np.linalg.svd(E)
S[0] = s[1] = (s[0]+s[1])/2
S[2] = 0
Ehat = U @ np.diag(S) @ Vt
```

Extracting $R,\ t$ from E

$$[t_\times] = \begin{pmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{pmatrix}$$

2) Finding t

- Reconstruction of the scene can be found up to a scale factor, hence OK to use "normalized E", where : $||t||_2 = 1 \implies \sigma = 1$
- Since $RR^T{=}1$, squaring: $EE = [t_\times]RR^T[t_\times]^T = [t_\times][t_\times]^T = [t_\times][-t_\times] = -[t_\times]^2$

$$-[t_\times]^2 = \begin{pmatrix} -t_z^2 - t_y^2 & t_x t_y & t_x t_z \\ t_x t_y & -t_z^2 - t_x^2 & t_z t_z \\ t_x t_z & t_y t_z & -t_y^2 - t_x^2 \end{pmatrix} \qquad -tr\left([t_\times]^2\right) = 2\left(t_x^2 + t_y^2 + t_z^2\right) = 2||t||_2$$

- Since $||t||_2 = 1$, we obtain a matrix, from which diagonal we can obtain the absolute entries of t

$$-[t_\times]^2 = \begin{pmatrix} 1 - t_x^2 & t_x t_y & t_x t_z \\ t_x t_y & 1 - t_y^2 & t_z t_z \\ t_x t_z & t_y t_z & 1 - t_z^2 \end{pmatrix}$$

4 possible solutions for R, t

- Recall that $\Sigma = diag(1,1,0)$ in $E = U\Sigma V^T$
- Defining

$$R_z(\pi/2) = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- The relative rotation is

$$R = UR_z^T V^T$$

- The relative translation (unitary scale) is

$$[t_\times] = UR_z^T \Sigma U^T$$

- As the same is valid for $R_z(-\pi/2)$ there are 4 possible solutions: 2 permutations of

$$R_z(\pm\pi/2) = \begin{pmatrix} 0 & \mp 1 & 0 \\ \pm 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Proof

$$E = [t_\times]R = \underbrace{UR_z\Sigma U^T}_{t}\underbrace{UR_z^T V^T}_{R}$$
$$= UR_z\Sigma R_z^T V^T = U\Sigma V^T$$

- as $R_z\Sigma$ is a skew-symmetric

$$R_z\Sigma = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

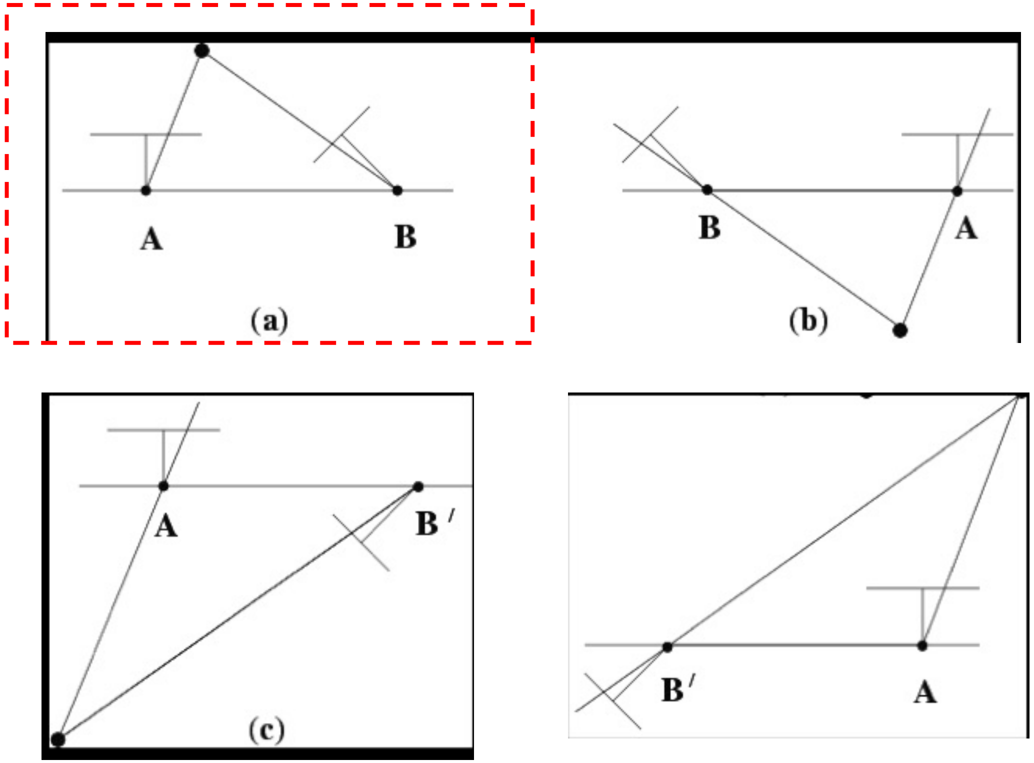
- and

$$[t_\times]^T = -[t_\times]$$

- R is orthogonal (product of 3 orthogonal matrices) if $\texttt{det}(R) = -1$ then $E = -E$

4 possible solutions for R, t

- However, <u>the only plausible solution</u> is the one when P lies in front-view of both cameras



- The 4 possibilities to test are

$$\hat{R} = U \begin{pmatrix} 0 & \mp 1 & 0 \\ \pm 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} V^T$$

$$[\hat{t}_\times] = U \begin{pmatrix} 0 & \mp 1 & 0 \\ \pm 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \Sigma U^T$$

$$[\hat{t}_\times] = \begin{pmatrix} 0 & -\hat{t}_z & \hat{t}_y \\ \hat{t}_z & 0 & -\hat{t}_x \\ -\hat{t}_y & \hat{t}_x & 0 \end{pmatrix} \Longrightarrow \begin{pmatrix} \hat{t}_x \\ \hat{t}_y \\ \hat{t}_z \end{pmatrix}$$

Remaining problem:

- Can $R,\ t,\ K_1,\ K_2$ be recovered from F ?

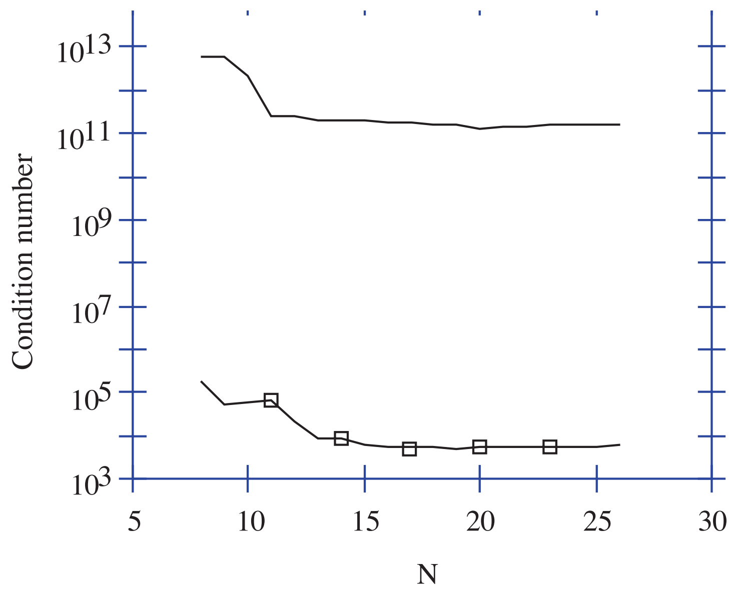
Practical challenges

“Noise” in data

- E matrix near singular – points lying on the same 2D plane, small parallax (disparity)

Solution

- Translate all image points coordinates to a centroid
- Scale them so that the average distance from center is sqrt(2), i.e. p_i =(1, 1, 1)
- Improvement of condition number



Hartley, R.I., 2012: In defense of the 8-point algorithm. *IEEE Trans. Pattern Analysis*, 19(6), 580-593

Historical development

- 1913 **Kruppa** – Determined the min. no. of correspondences (five), 11 solutions
- 1981 **Longuet-Higgins** – Easy implementation, 8-point algorithm (NASA-rover)*
- 1988 **Demazure** – Showed that there is at most 10 distinct solutions
- 1996 **Philipp** – Described and iterative algorithm to find the solutions
- 2004 **Nister** – 1st efficient and non iterative solution (basis decomposition)**

* H. Christopher Longuet-Higgins, A computer algorithm for reconstructing a scene from two projections, Nature, 1981
**D. Nister, An Efficient Solution to the Five-Point Relative Pose Problem, PAMI, 2004.

Understanding - self assessment

- What is the minimum number of correspondences between 2 images to perform relative orientation with a calibrated camera and why?
- Can you provide a geometrical interpretation of the epipolar constraint?
- How would you derive the epipolar constraint?
- How is the essential matrix defined?
- What are the properties of the essential matrix?
- How are the essential and fundamental matrices related?
- Is it always important to normalize the point coordinates in the 8-point algorithm?
- How is the normalization of image coordinates performed?
- How is the quality of the derived translation and rotation measured?