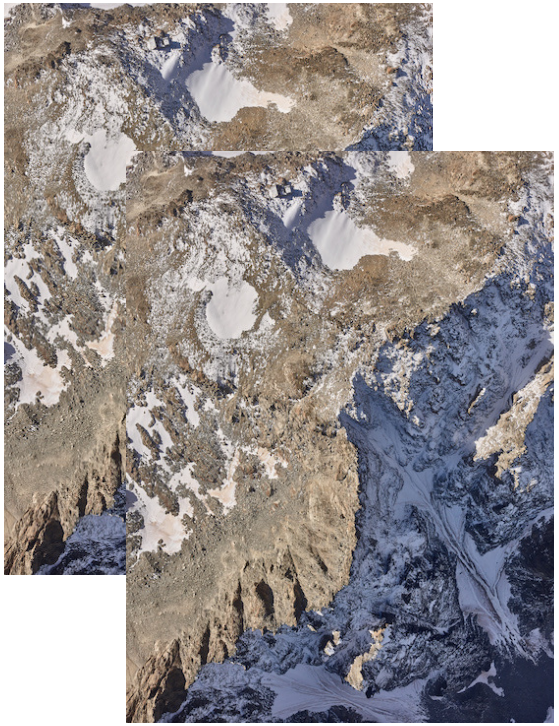
Lecture 4
Stereo Vision

ENV408: Optical Sensing & Modeling for Earth Observations

Jan Skaloud

Stereo reconstructions



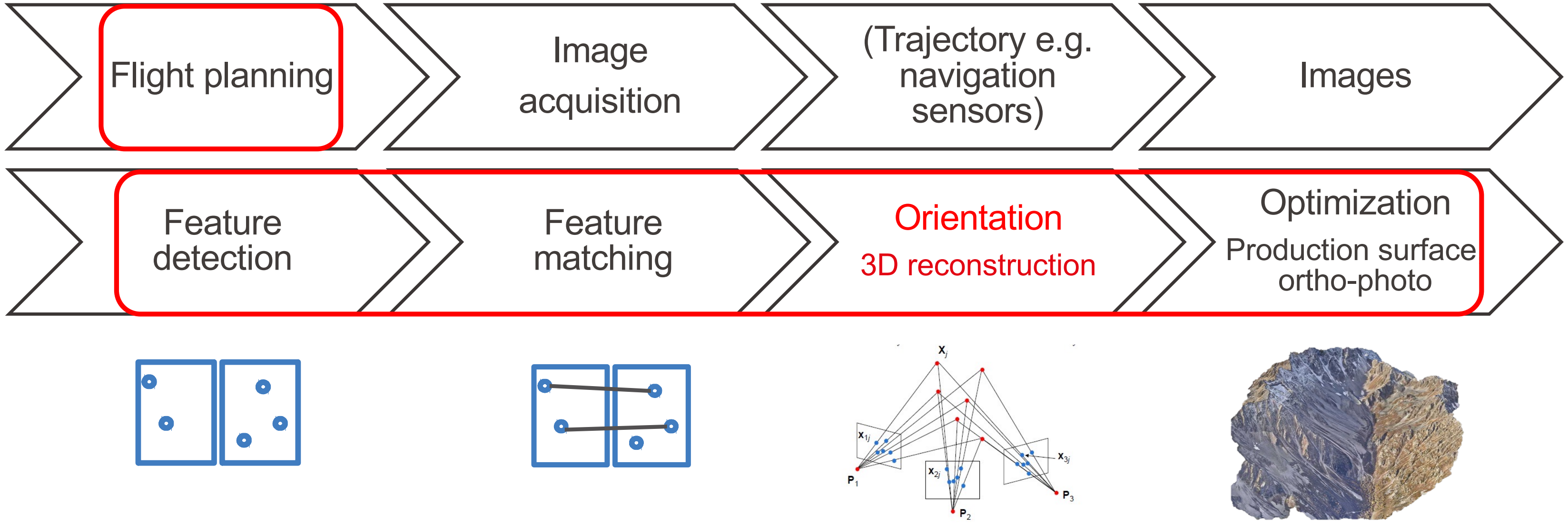
Today

Understand how to reconstruct simultaneously 3D scene structure and camera pose from two + multiple images

Tomorrow

Use undistorted (Lab01) matched key-points (Lab02) to orient *relatively* two images and triangulate key-point coordinates in 3D.

Apply solution of absolute orientation (Lab03) to express the scene in world coordinates.

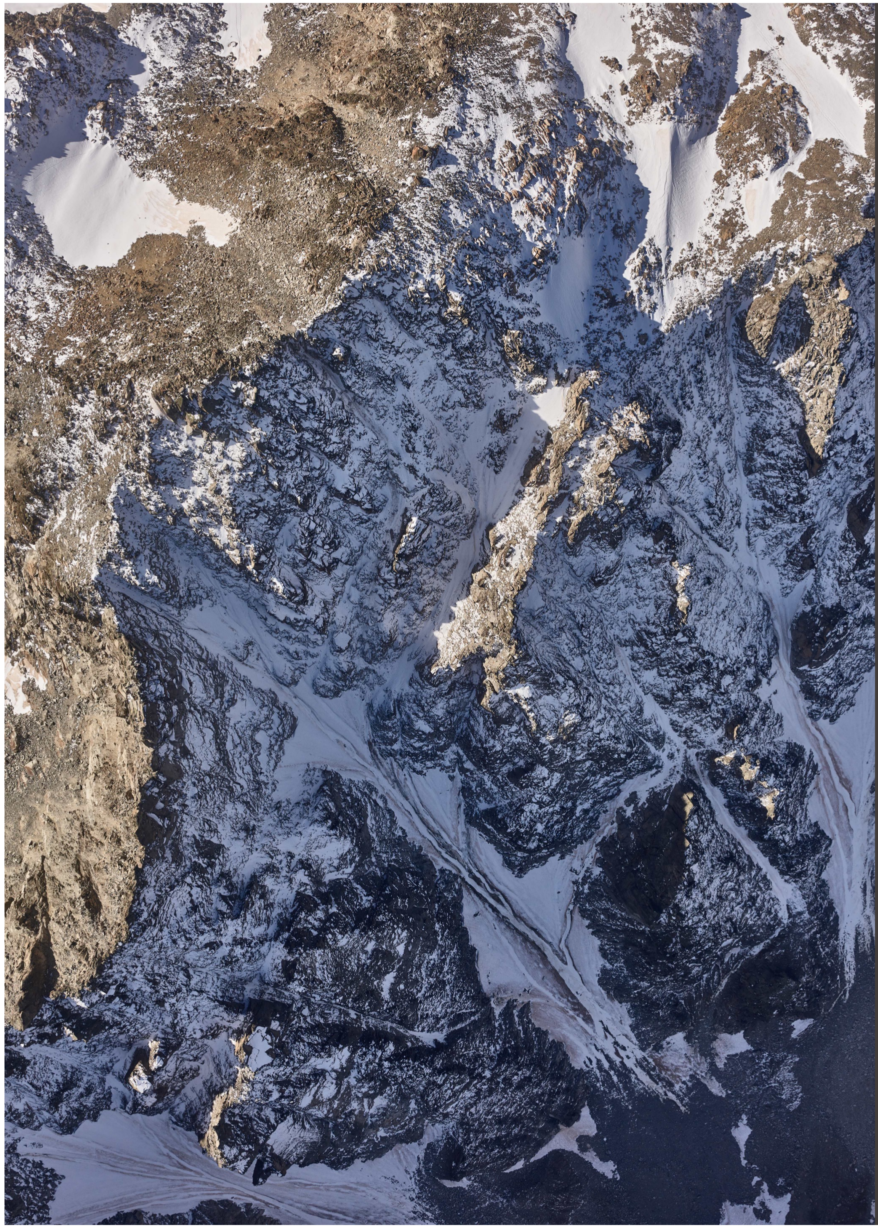


Lectures

- Image primes (L1)
- Salient *features* (L2)
- Image *orientation* (L3)
- *Stereo* vision (L4)
- Many photos, mapping products (L5)

Exercises

- Image 'corrections' (Lab01)
- Detection & matching (Lab02)
- Approx. absolute orientation (Lab03)
- Approx. relative orientation (Lab04)
- Optimization, DEM, ortho-photo (Lab05)



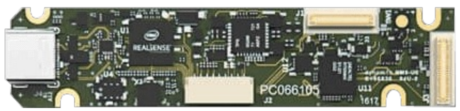
Stereo vision - principle

Triangulation (depth)

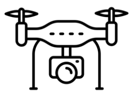
Stereo vision (pose & depth)

Examples

- Short-base stereo cameras

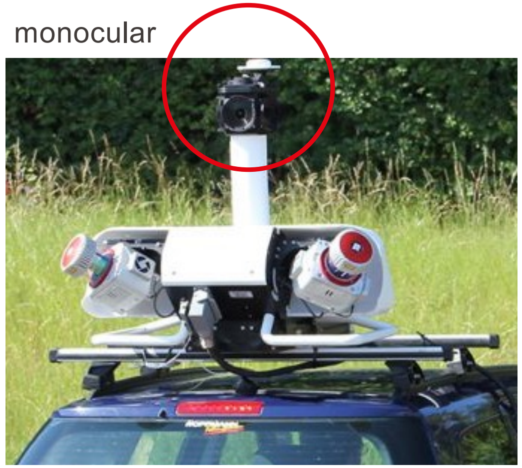


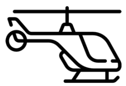
- Motion-based stereo



stereo-pair

monocular





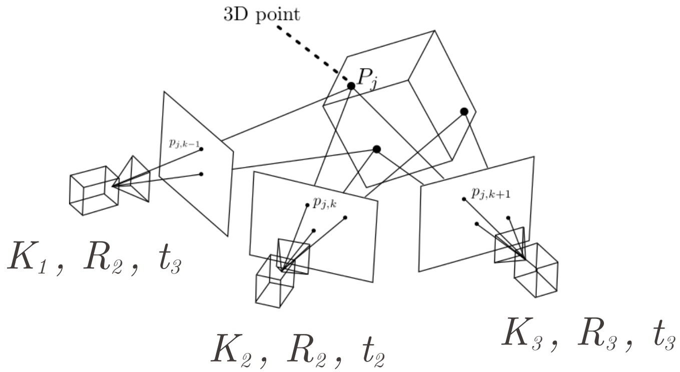
Two or Multiple view geometry

A. Depth from stereo vision (3D reconstruction)

Goal: recover the 3D structure from images

Assumption: **known** K_i , R_i , t_i (i.e. camera calibrated & oriented)

- simplier, we start with it

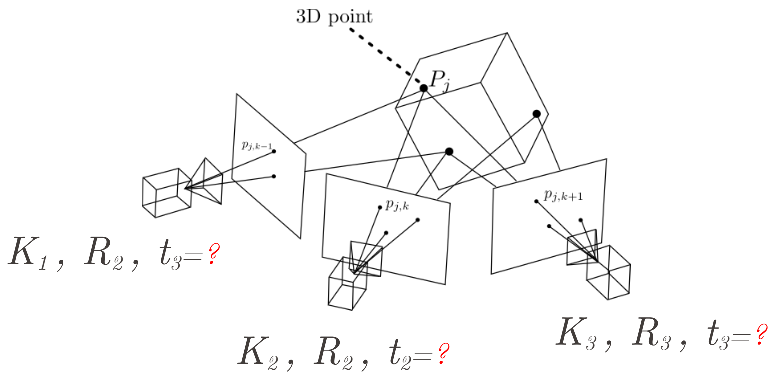


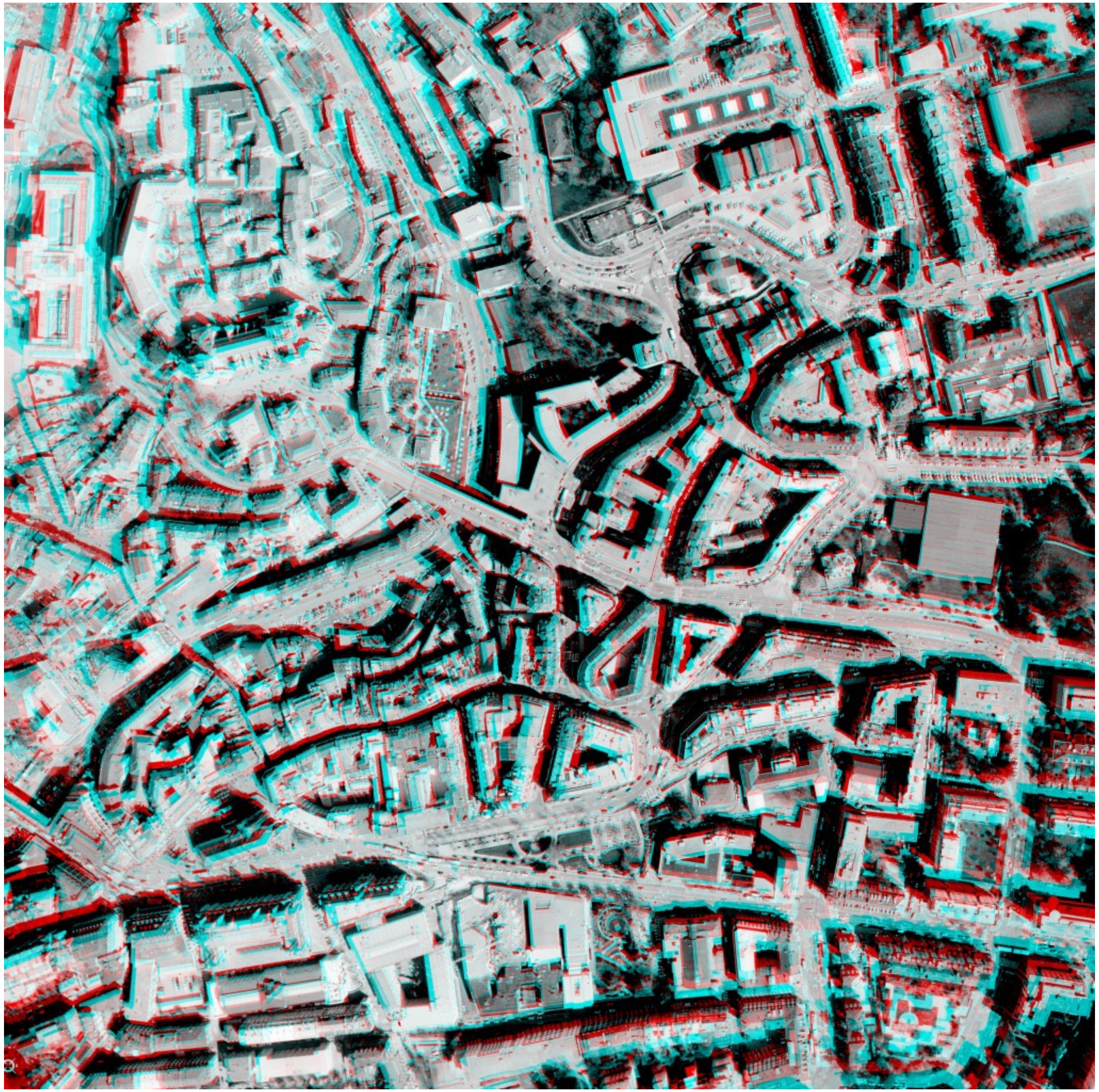
B. Structure from motion (SFM)

Goal: recover simultaneously scene structure (3D) and camera pose (up to scale)

Assumption: **unknown** K_i , R_i , t_i

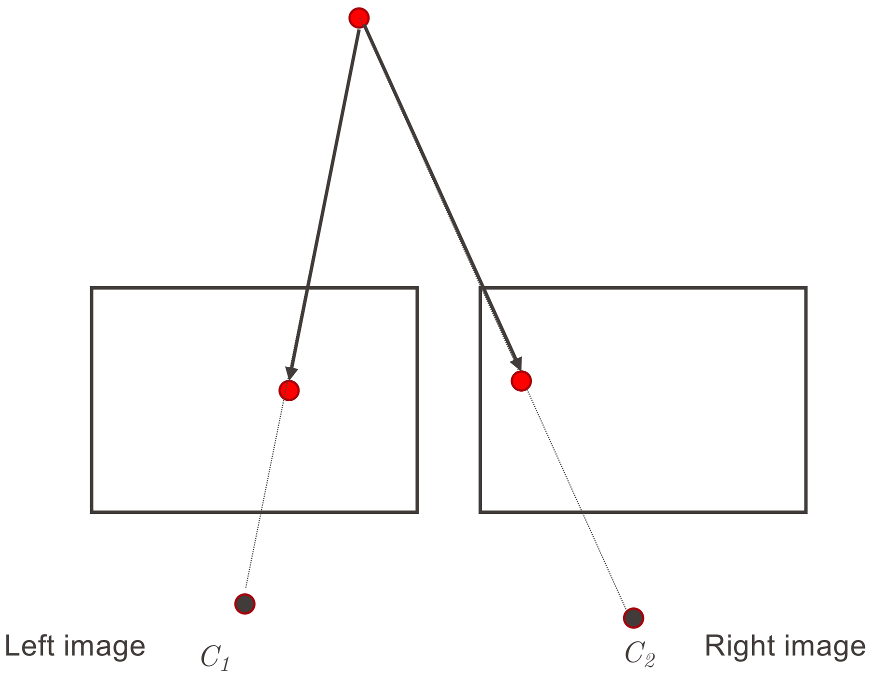
- once we know how to determine the depth in the previous part, we show how to solve also this problem

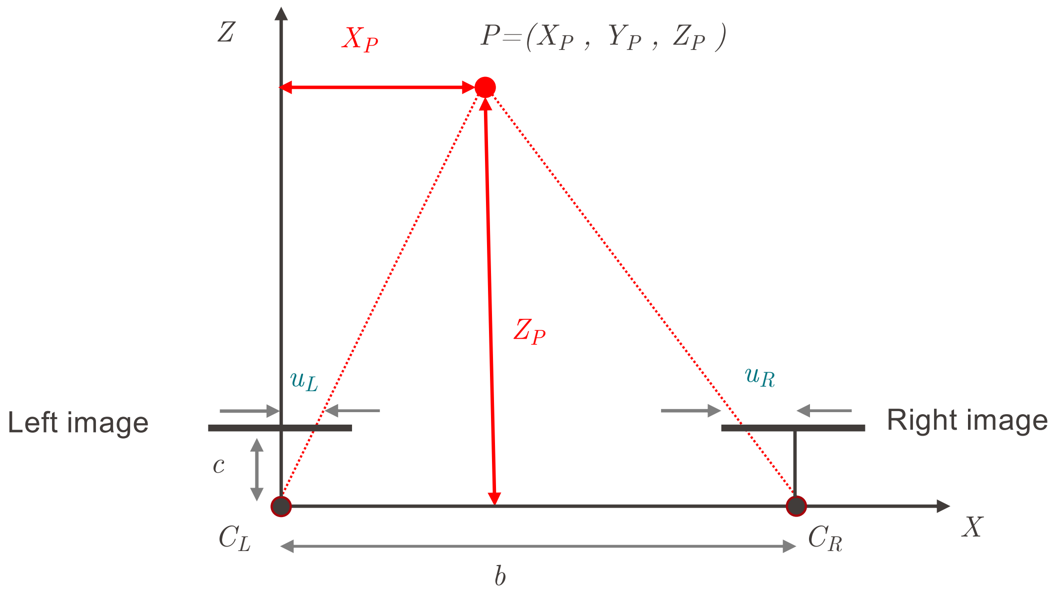




Stereo vision – simplified case

- Aligned (on x-axis) identical cameras

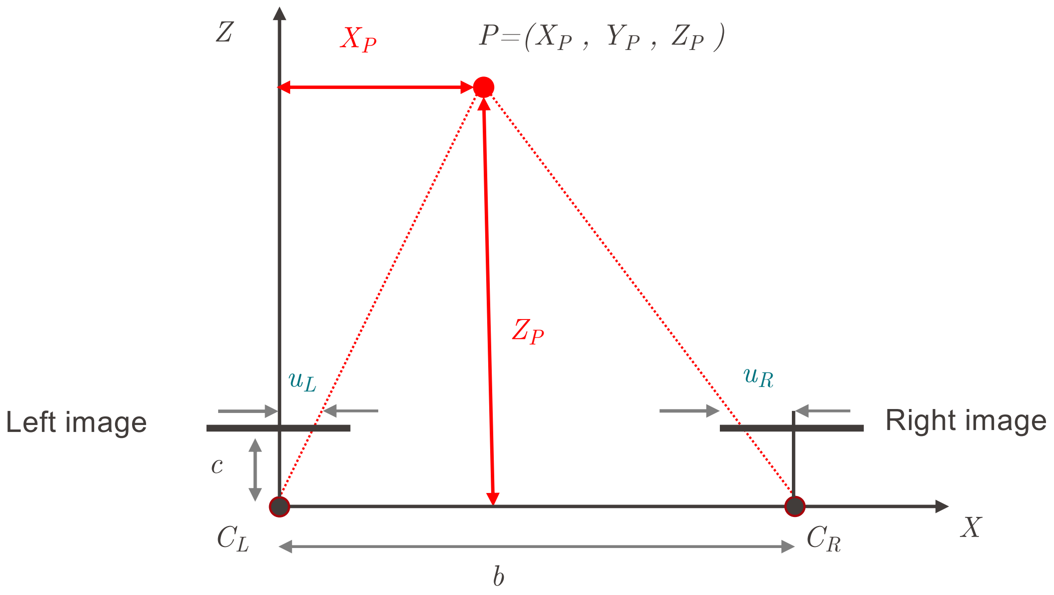




Baseline b = distance between the optical centers of the two cameras

Stereo vision – simplified case

- Aligned (on x-axis) identical cameras (top view)



Baseline b = distance between the optical centers of the two cameras

$$\frac{c}{Z_p} = \frac{u_L}{X_P}$$

$$\frac{c}{Z_p} = \frac{-u_R}{b - X_P}$$

$$Z_p = \frac{b \cdot c}{u_L - u_R}$$

$(u_L - u_R)$ is a disparity, a difference in the location of a 3D point on two image planes

Q: What is the disparity of a point in infinity?

Q: What is the maximum disparity of a stero camera?

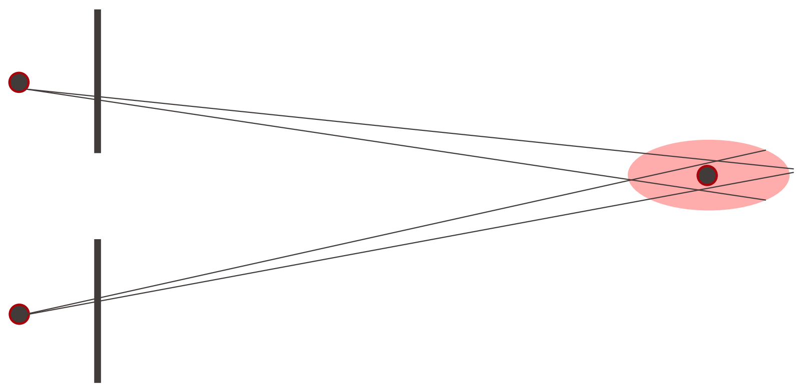
How to choose a baseline?

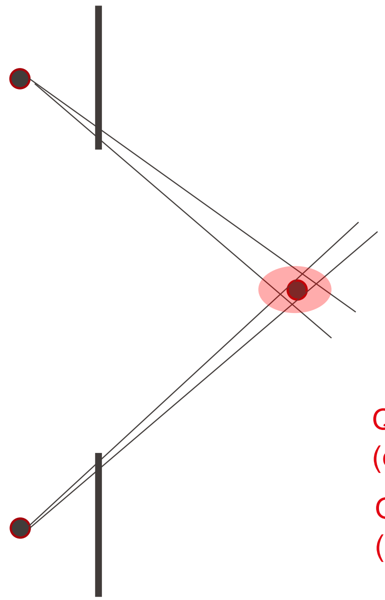
Short baseline

- Small error in image obs. => large error in depth
- Close objects are observable
- Automated matching easier (less change)

Large baseline

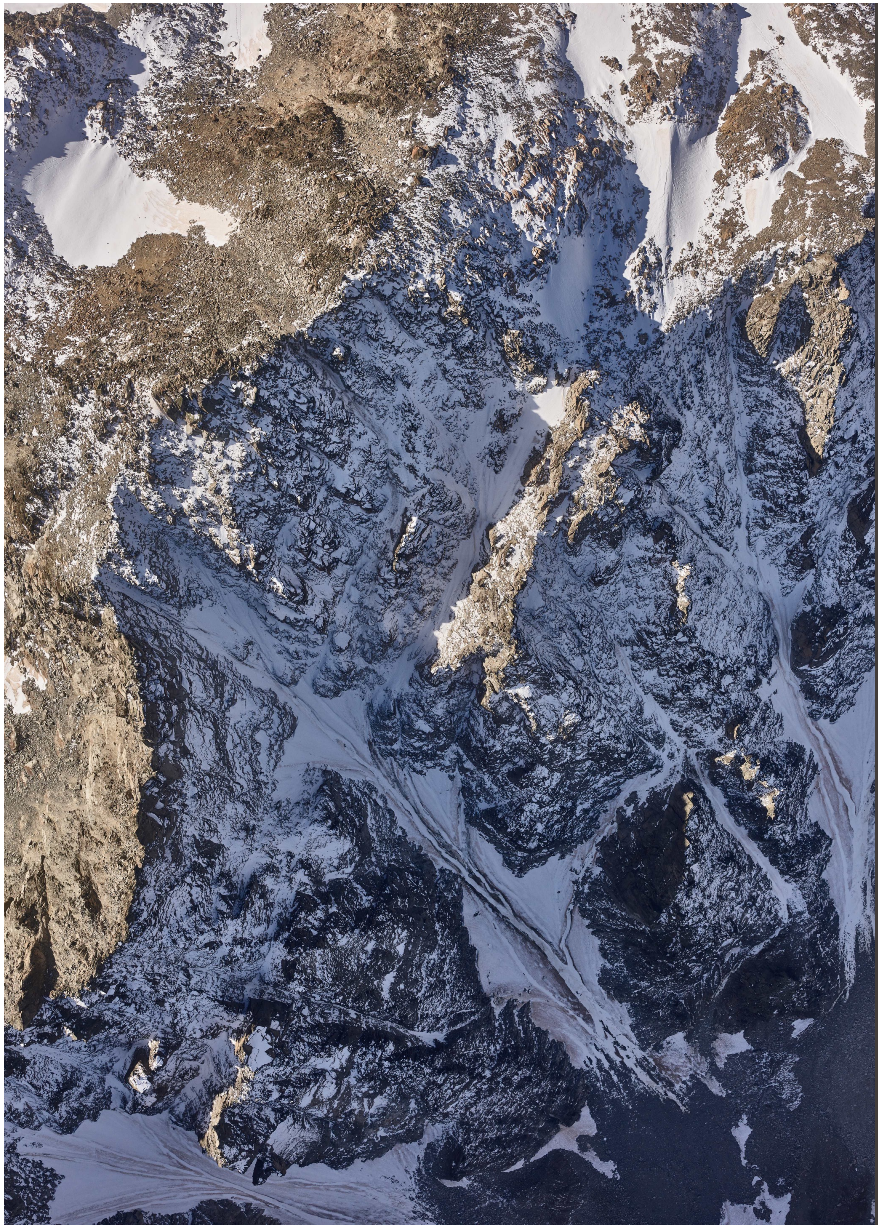
- Smaller error in depth
- Close objects are *not observable*
- *Challenging automated matching* for closer objects





Q: How to increase the accuracy (of object determination)?

Q: How to compute depth uncertainty (as a function of disparity)?



Stereo vision - principle

Triangulation (depth)

Stereo vision (pose and depth)

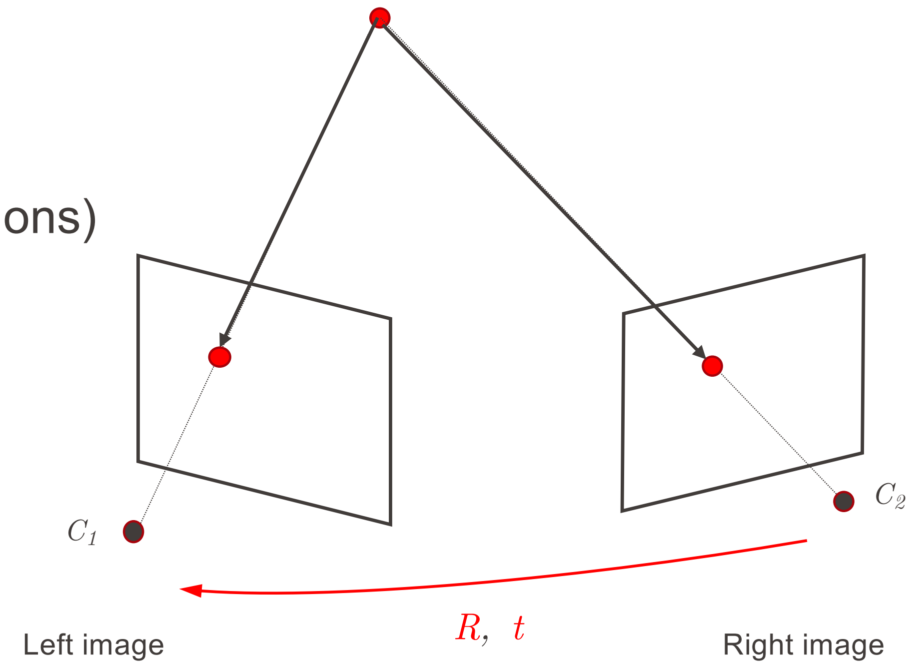
Stereo vision – general case

I. Cameras are not identical (in a fixed system)

II. Cameras are not aligned (at least horizontally)

Usage of stereo camera requires:

- Relative pose*
- Intrinsic param.** (c, x_0 , y_0, lens distortions)



- Approach:
  - 1st : camera calibration
  - 2nd: How to get R, t ?

(later in Lecture 4 and Lab 04)

* pose = position + attitude = exterior orientation (EO) or extrinsic parameters

** intrinsic parameters = interior orientation (IO)

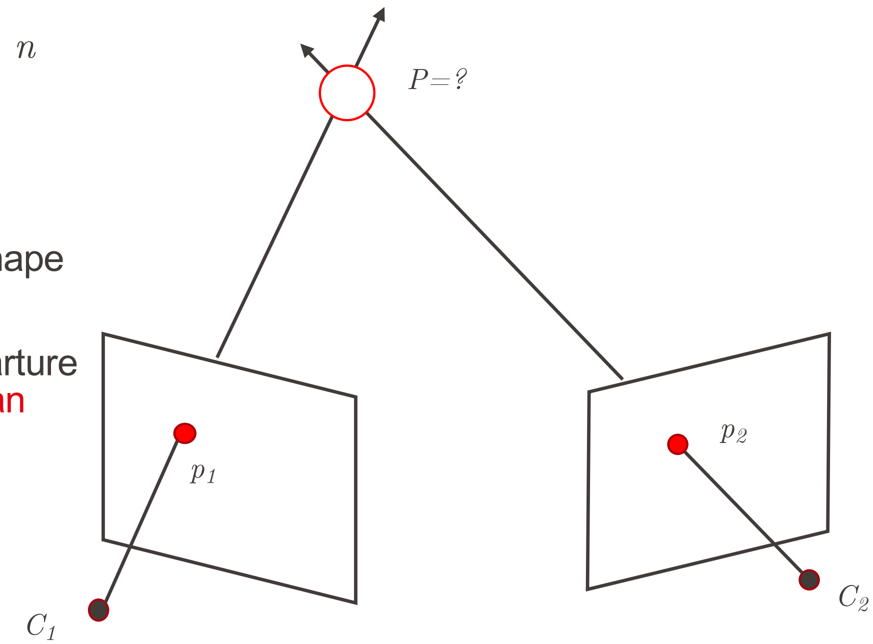
Stereo vision – triangulation (particular case)

Prerequisites

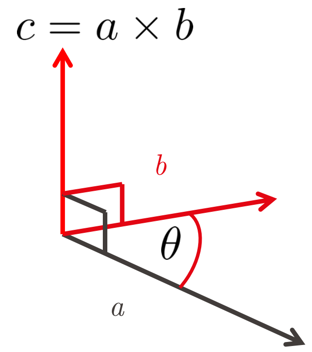
- Pose $(R,\ t)$ is known (at least relatively)
- Image correspondences exist for a set of points $P_i\ ,\ \ i=1\ \dots\ n$

Goal

- Intersect correspondences p_1 & p_2 spatially to recover 3D shape (coordinates of P)
- With noisy observations and imperfect modelling (e.g. departure from colinarity) the intersection is not perfect - search first an approximation.



Vector cross product (refresher for triangulation)



Cross product of two vectors $(a,\ b)$

Gives a third vector (c) that is perpendicullar to (both of) them:

$$c \cdot a = 0$$
$$c \cdot b = 0$$

The vector cross product can be expressed as a product of a *skew-symetric matrix* $[\ ]_x$:

$$c = a \times b = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = [a_\times]\, b$$

For parallels vectors (a, b) their cross product = ?

Triangulation – a least squares (LS) approximation

Approach

- Create a system of equations for the left and right cameras
- Solve it

Left camera

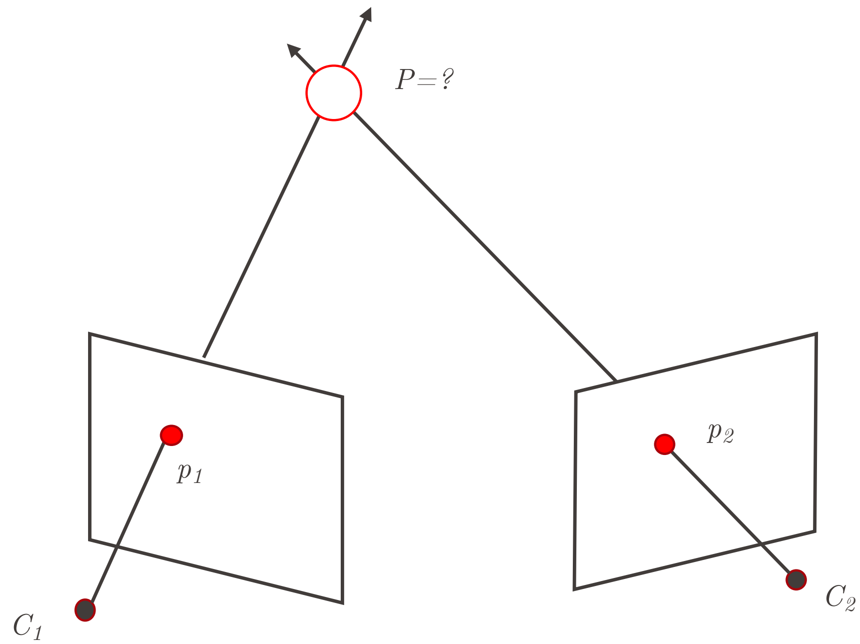
(Often) assumed to represent a mapping (world) frame

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

Right camera

(Often) expressed relatively to left camera

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



Left camera frame considered as a mapping (world) frame

Q: How to modify the 2nd equations if the right image is taken by displacing the left camera?

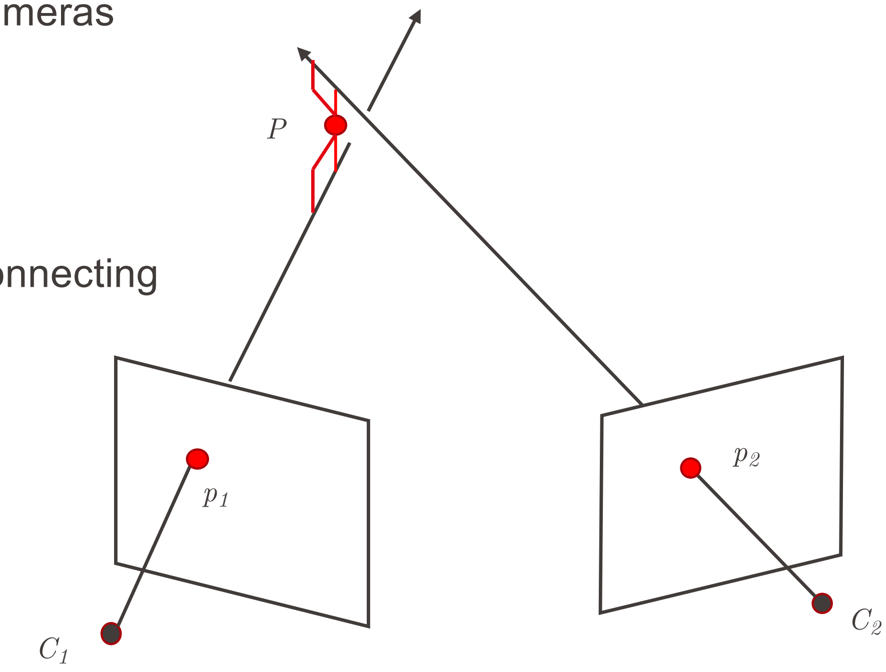
Triangulation – an approximation (least-square)

Approach

- Create a system of equations for the left and right cameras
- Solve it (system of homogeneous equations)

Solution (SVD)

- P is found as a mid point of the shortest segment connecting both spatial lines.



Left camera frame considered as mapping /world frame

Triangulation – an approximation (LS)

Left camera

$$\mu_1 \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} = \underbrace{K_1[I\,|\,0]}_{\Pi_1'} \cdot \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \implies \mu_1 p_1 = \Pi_1' \cdot P \implies p_1 \times \mu_1 p_1 = p_1 \times \Pi_1' \cdot P \implies 0 = [p_1]_\times \cdot \Pi_1' \cdot P$$

Right camera

$$\mu_2 \begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} = \underbrace{K_2[R\,|\,t]}_{\Pi_2'} \cdot \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \implies \mu_2 p_2 = \Pi_2' \cdot P \implies p_2 \times \mu_2 p_2 = p_2 \times \Pi_2' \cdot P \implies 0 = [p_2]_\times \cdot \Pi_2' \cdot P$$

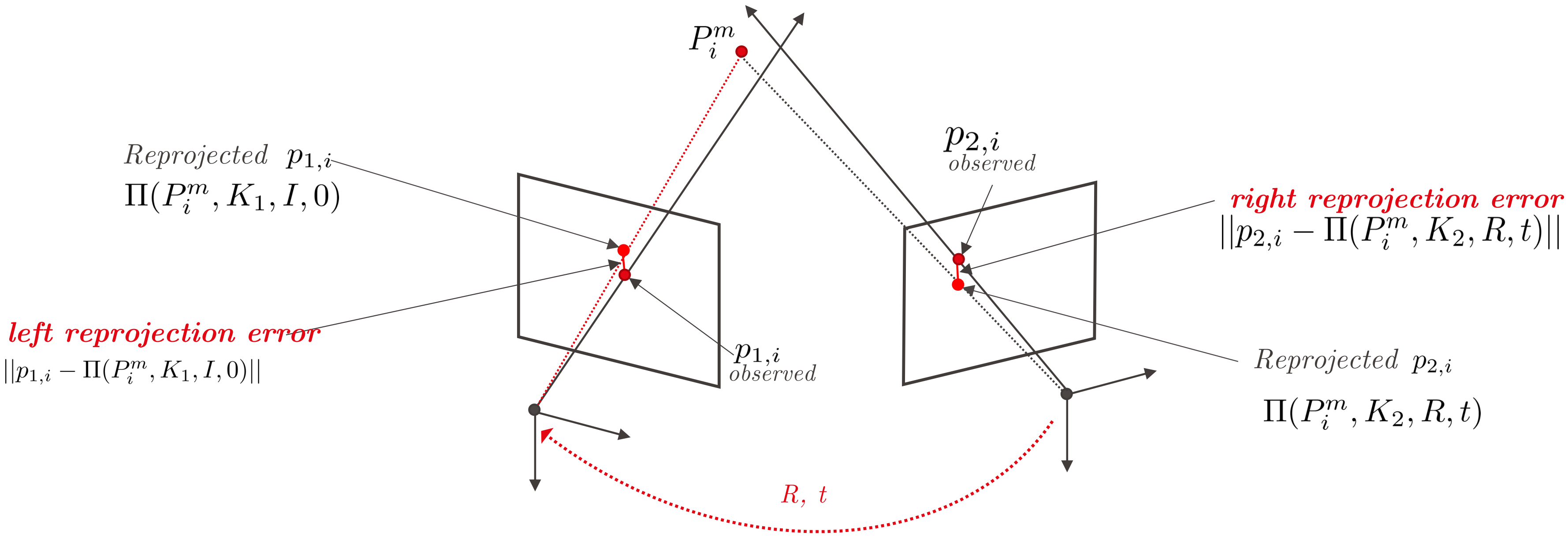
A system of homogenous equations
P can be determined using SVD (similarly to DLT, Lab 03)
[more details in Lab04 handlout]

Triangulation – non-linear optimization

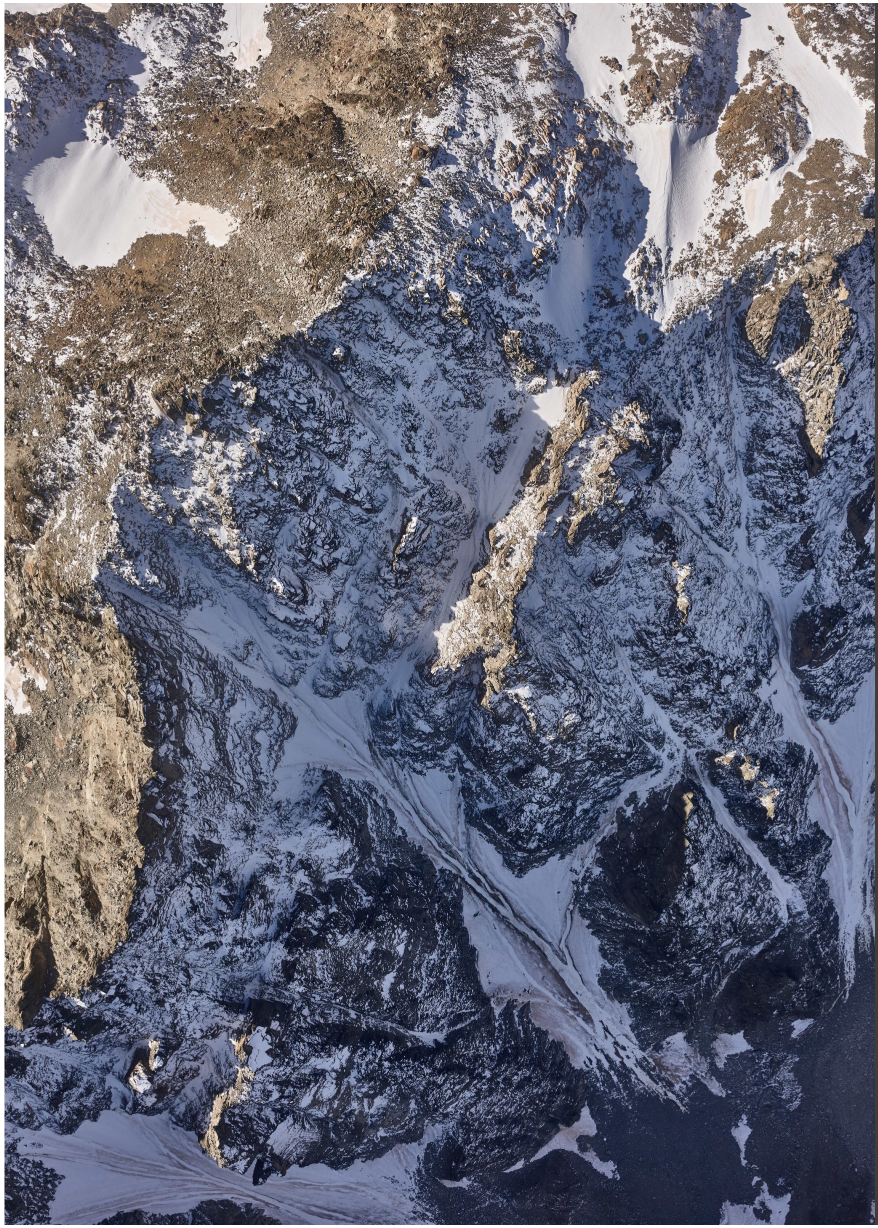
The coordinates of points are:

- first **initiated** by the linear (LS) approximation
- then **refined** via (iterative) non-linear **optimization**

$$argmin_P \sum_{i=1}^{n} \|p_{1,i} - \Pi(P_i^m, K_1, I, 0, \|^2 + \|p_{2,i} - \Pi(P_i^m, K_2, R, t, \|^2$$



EPFL



Stereo vision (pose & depth)

Epipolar Geometry

Essential and Fundamental Matrices

8 points algorithm

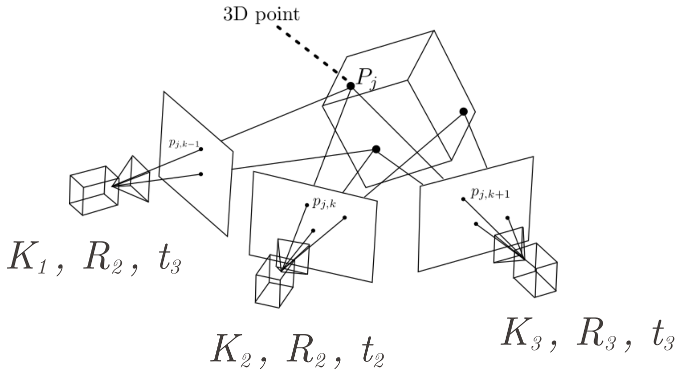
Two or Multiple view geometry

A. Depth from stereo vision (3D reconstruction)

Goal: recover the 3D structure from images

Assumption: **known** K_i , R_i , t_i (i.e. camera calibrated & oriented)

- simplier, we start with it

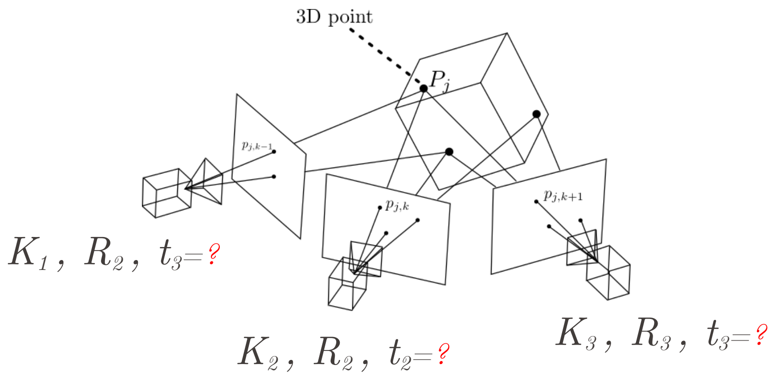


B. Structure from motion (SFM)

Goal: recover simultaneously scene structure (3D) and camera pose (up to scale)

Assumption: **unknown** K_i , R_i , t_i

- if we can triangulate!
- let's see what is common on 2 images (epipolar geometry)
- let's put it all together & solve it !



Correspondence problem

Prerequisits

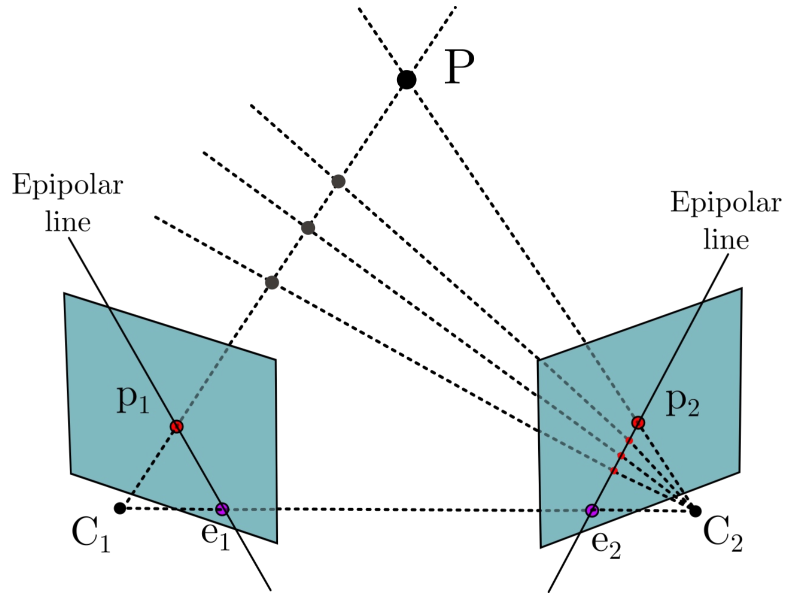
- Assume that pose (R and t) is known (at least relatively)
- Image correspondences exist for a set of points $P_i \quad i=1 \ldots n$

Questions

- Given a point on the left-image, p_L , where is its correspondence, p_R , on the right image?
- Note: 2D exhaustive search is very expensive (computationally)

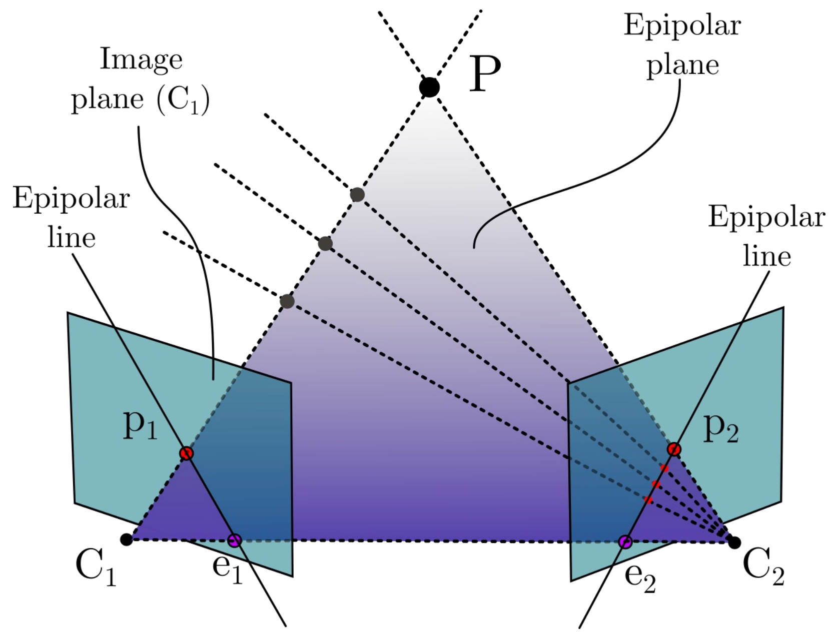
Answer:

Potential matches have to lie on an epipolar line! (see after)



Epipolar Geometry

- **Epipolar plane**: 3D plane formed by $\mathbf{C_1}, \mathbf{C_2}$ (cam. centers) & $\mathbf{P}$
- **Epipoles $\mathbf{e_1}, \mathbf{e_2}$:** intersection of the line $\mathrm{C_1}, \mathrm{C_2}$ with image planes
- **Epipolar line:** Intersection of epipolar plane with image plane
- **Epipolar constraint:** given P, corresponding points $\mathrm{p1}, \mathrm{p2}$ **must** lie on their respective epipolar lines



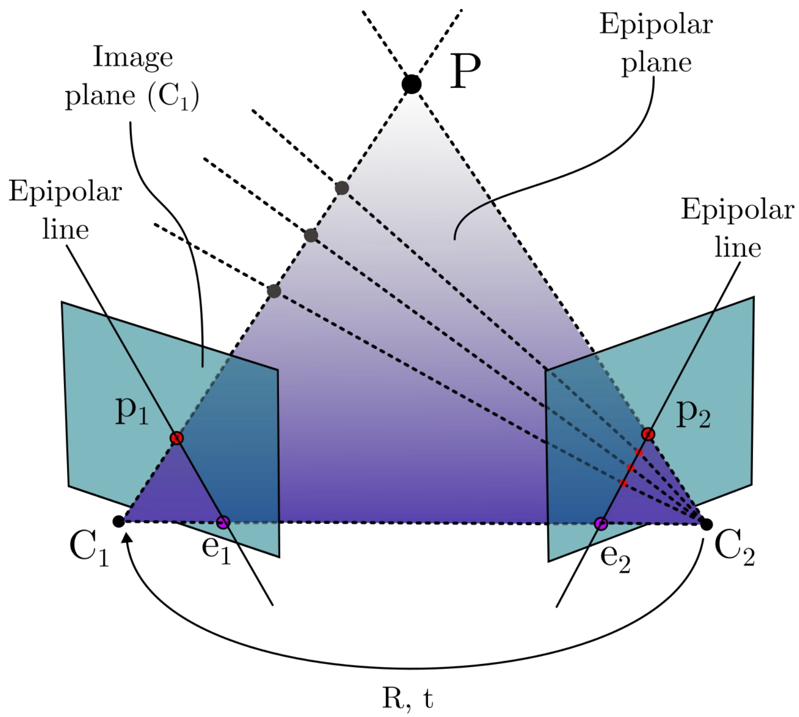
Epipolar Constraint 1/3

Formulation via epipolar lines

- $\mathbf{R}, \mathbf{t}$: rotation and translation relative from C_2 to C_1
- Point P in:
  - Camera 1 frame $\overrightarrow{C_1, P} = P_1 = \mu_1 p_1$
  - Camera 2 frame $\overrightarrow{C_2, P} = P_2 = \mu_2 p_2$

$$\Downarrow$$

$$\mu_1 p_1 = R\mu_2 p_2 + t$$



Epipolar Constraint 2/3

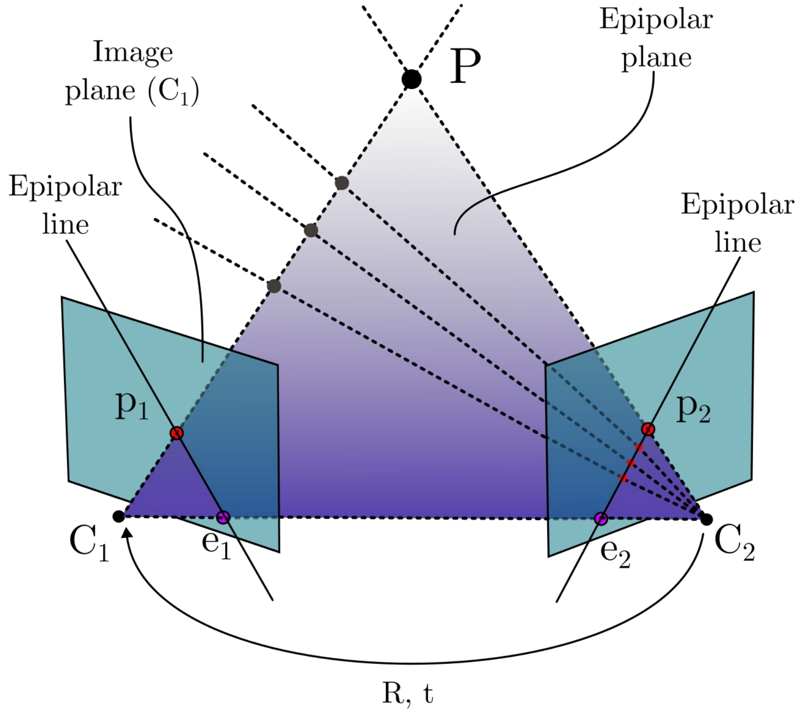
Formulation via epipolar lines

- The 3 vectors $\overrightarrow{C_1, P_1}$, $\overrightarrow{C_2, P_2}$ and $\overrightarrow{C_1, C_2} = t$ must be coplanar

- This constraint is mathematically equivalent to null vector triple product $a \cdot (b \times c) = 0$

- This gives: $P_1 \cdot (t \times \underbrace{RP_2}_{}) = 0$

  P_2 in camera 1 frame



Epipolar Constraint 3/3

Reminder: vector *cross* product equivalent to skew-symmetric matrix multiplication $[t_\times]$

$$P_1 \cdot (t \times RP_2) = 0$$

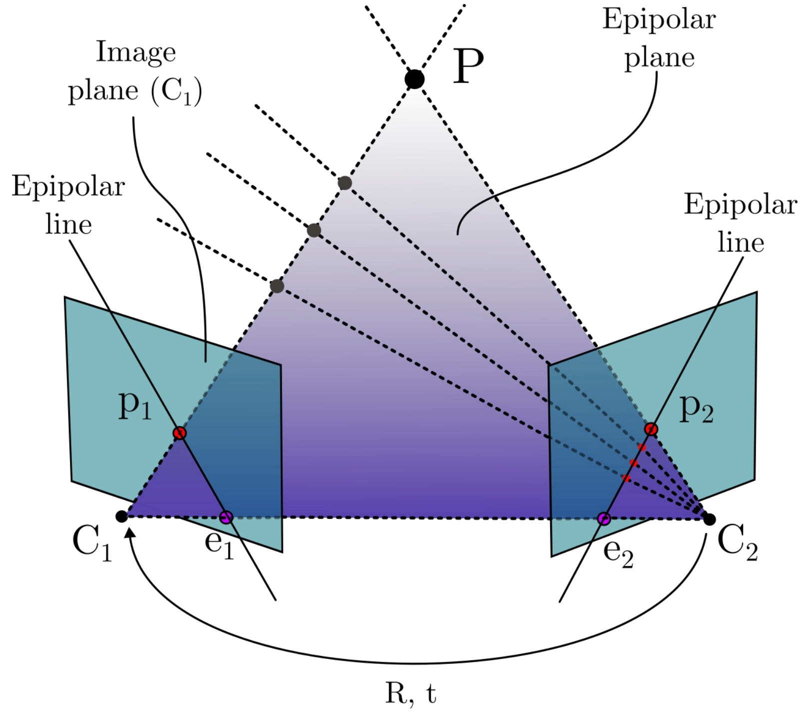
$$\Updownarrow$$

$$P_1^T [t_\times] R P_2 = 0$$

$$\Updownarrow$$

Only vector direction counts

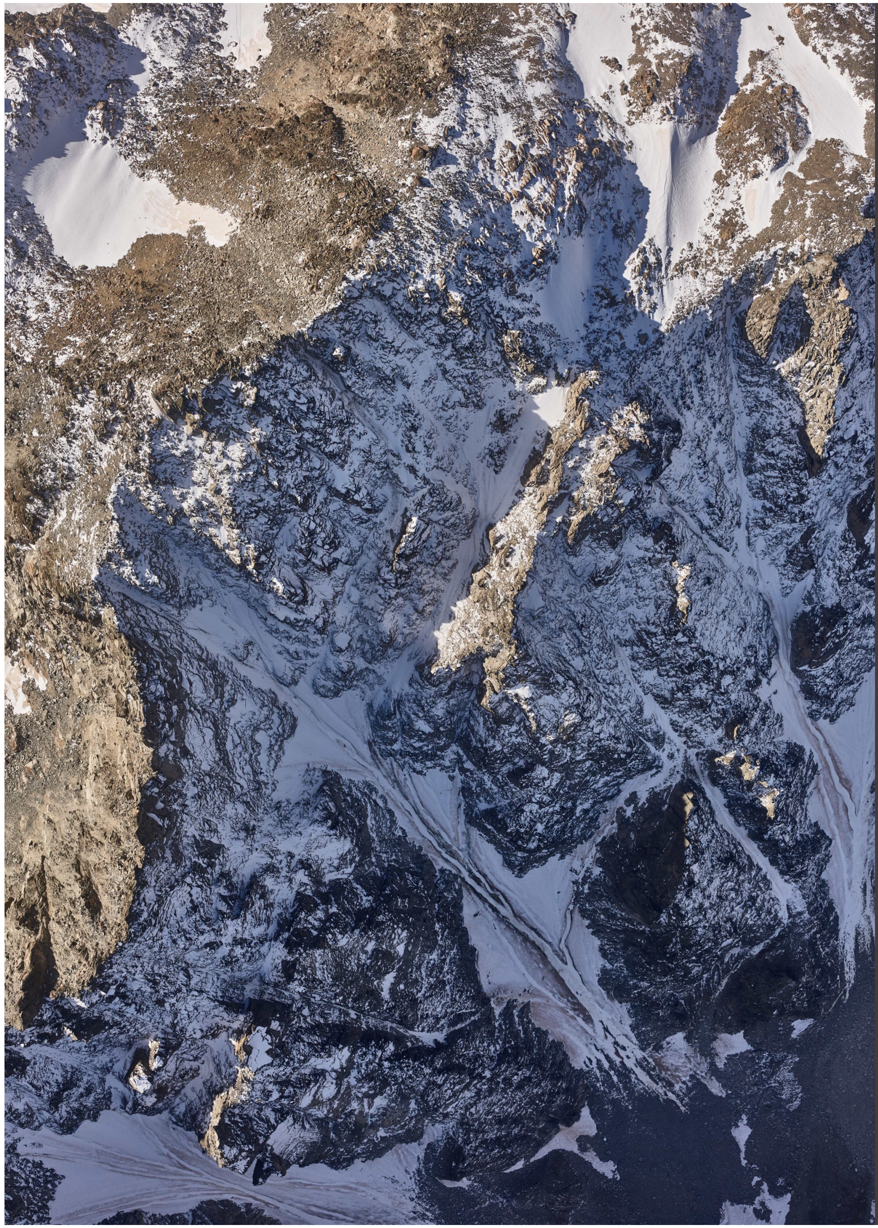
$$p_1^T [t_\times] R\, p_2 = 0$$



Defining Essential matrix $\mathbf{E}$:

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

$$p_2^T E\, p_1 = 0$$



Stereo vision (pose & depth)

Epipolar Geometry

Essential and Fundamental Matrices

8 points algorithm

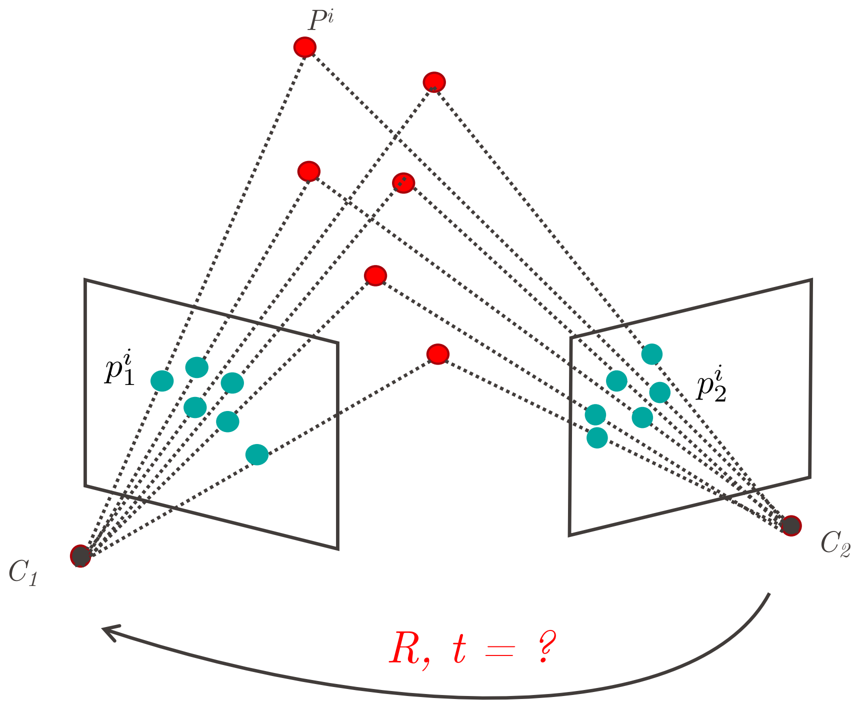
Relative orientation (SFM)

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

- 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 undistorted coordinates

$$\begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} = K_1^{-1} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} \qquad \begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} = K_2^{-1} \begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} \qquad K = \begin{pmatrix} c & 0 & x_{PPS} + c_x \\ 0 & c & y_{PPS} + c_y \\ 0 & 0 & 1 \end{pmatrix}$$

$$x_2^T E x_1 = 0$$ (notation of complementary reading / book chapter)

$$p_2^T E p_1 = 0$$ (notation slides + labs, same thing)

$$\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix}^T E \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} = 0$$

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

-

Epipolar geometry – uncalibrated camera

Without the knowledge of $\mathbf{K}$: p_i can only be defined by u,v since x,y are unknown

$$\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix}^T E \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} = 0$$

$$\Downarrow$$

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

Fundamental matrix $F = (K_2^T)^{-1} E K_1^{-1} \implies \begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix}^T \boxed{F} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} = 0$

Epipolar geometry – system of equations

Each pair of point correspondences $p_1 = (u_1, v_1, 1)^T, \quad p_2 = (u_2, v_2, 1)^T$ provides a linear equation:

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

$$\Downarrow$$

$$u_2 u_1 e_{11} + u_2 v_1 e_{12} + u_2 e_{13} + v_2 u_1 e_{21} + v_2 v_1 e_{22} + v_2 e_{23} + u_1 e_{31} + v_1 e_{32} + e_{33} = 0$$

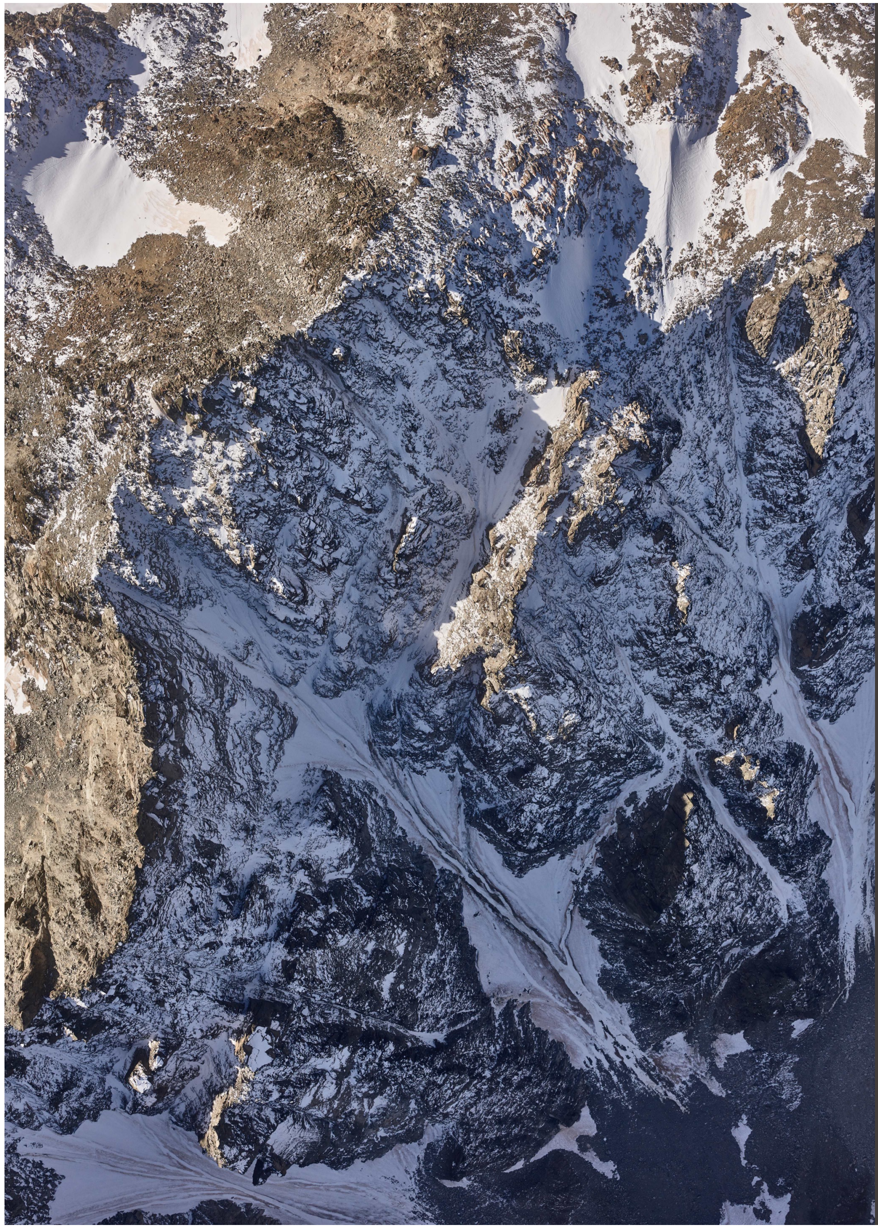
Given enough correspondences, E (or F) can be obtained.

We will investigate the following questions :

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

Epipolar geometry – inverse problem for E

- How many knowns per n ?
  - per correspondence:
  - per n :
- How many unknowns per n ?
  - correspondences:
  - general:
  - together:
- When a solution exist?



Stereo vision (pose & depth)

Epipolar Geometry

Essential and Fundamental Matrices

8 points algorithm

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.

The 8-point algorithm – formation of constraints

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

$$u_2 u_1 e_{11} + u_2 v_1 e_{12} + u_2 e_{13} + v_2 u_1 e_{21} + v_2 v_1 e_{22} + v_2 e_{23} + u_1 e_{31} + v_1 e_{32} + e_{33} = 0$$

- For n points (when omitting bars)

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

-

The 8-point algorithm – finding E

Minimum solution $Q \cdot \mathtt{vec}(E) = 0$

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

Over-determined solution (n > ?)

- By minimizing $||Q \cdot \mathtt{vec}(E)||^2 = (\mathtt{vec}(E))^T \cdot Q^T Q \cdot \mathtt{vec}(E)$
- Subject to constraint $||\mathtt{vec}(E)||^2 = 1$
- Solution $\mathtt{vec}(E)$ is an eigenvector corresponding to the smallest eigen value of Q
- Via SVD of $Q^T Q$ matrix that is in this case equivalent to SVD of Q* (+ see the lecture 3 and Lab03)
- Implementation hints: *see slides in appendix and Lab04 handout*

** K. Inkilä, 2005, Homogeneous least square problem, Photogrammetric Journal of Finland.*

Extracting R, t from E

I. Enforcing E to be in the "E-space"

- Singular value decomposition $E = U\Sigma V^T$
- "In case of no-errors", perfect correspondences: $\Sigma = diag\,(\sigma, \sigma, 0)$
- Due to errors: $\tilde{E} = U\,diag\,(\sigma_1, \sigma_2, \sigma_3)\,V^T, \quad \sigma_1 \geq \sigma_2 \geq \sigma_3$
- Choosing $\hat{E} = U\,diag\,(\sigma, \sigma, 0)\,V^T, \quad \sigma = (\sigma_1 + \sigma_2)/2$
- … satisfies E-space, but there could be another E leading to a smaller $||Q \cdot \texttt{vec}(E)||^2$
- Python … see appendix and Lab04 handout for hints

Extracting R, t from E

II. Finding t

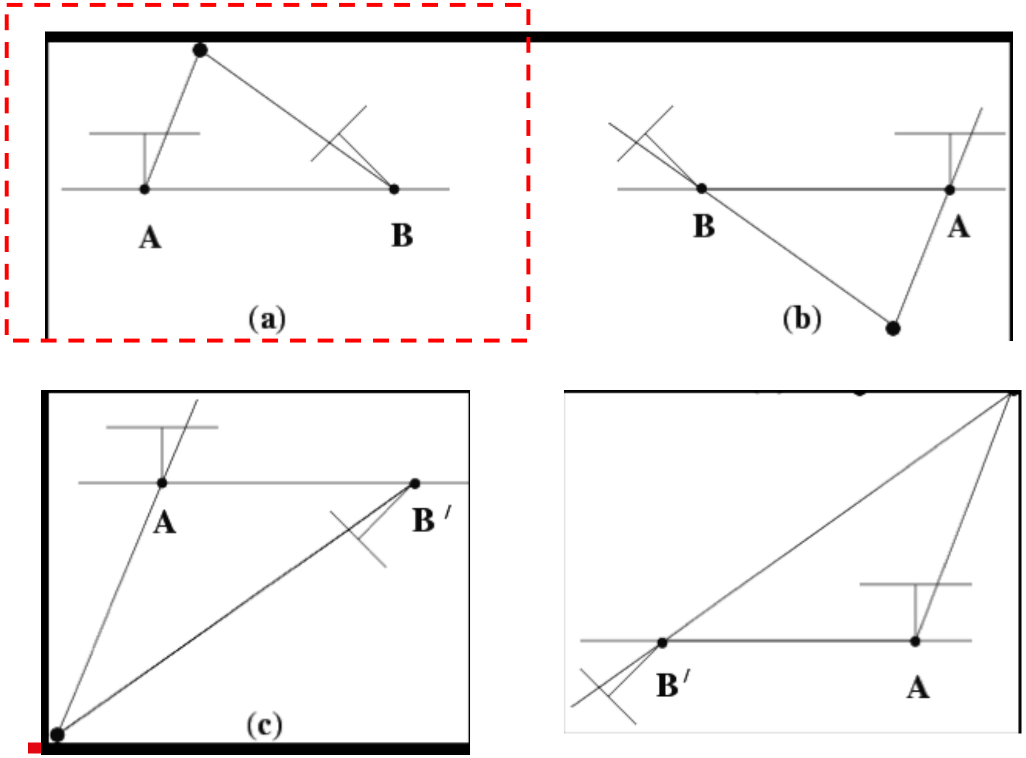
- $RR^T=1$, thus: $EE = [t_\times]RR^T[t_\times]^T = [t_\times][t_\times]^T = [t_\times][-t_\times] = -[t_\times]^2$
- Reminder: $[t_\times] = \begin{pmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{pmatrix}$ and $||t||_2 = 1$ (scale is not recovered from E)
- Thus $-[t_\times]^2 = \begin{pmatrix} -t_z^2 - t_y^2 & t_x t_y & t_x t_z \\ t_x t_y & -t_z^2 - t_x^2 & t_z t_z \\ t_x t_z & t_y t_z & -t_y^2 - t_x^2 \end{pmatrix}$
- 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

Proof in Appendix

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



- The are 4 possibilities to test"

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

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

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

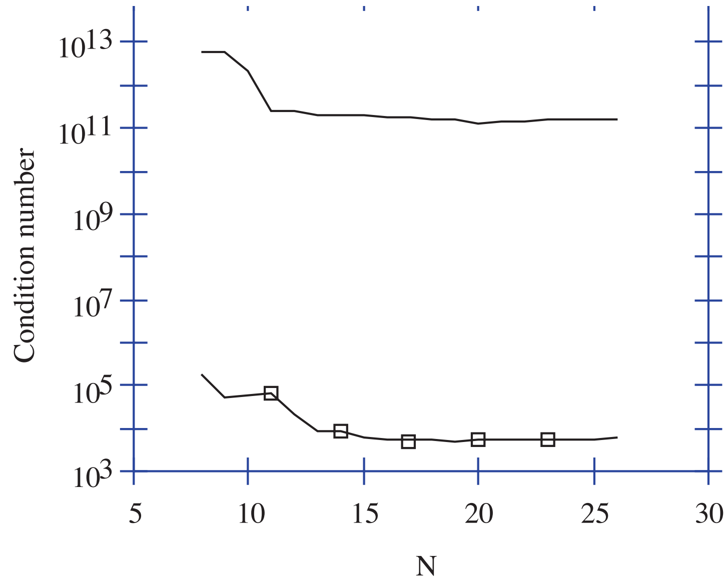
Remaining problem:

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

Practical issue correction

“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. $p_i = (1, 1, 1)$
- Improves condition number – solution stability!

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

Understanding - self assessment 1/2

- What is the difference between structure from motion (SFM) and 3D reconstruction?
- How do you define disparity in a *simple* case?
- How do you define disparity in a *general* case?
- How to express / derive the mathematical relation between the depth, disparity, baseline and camera constant?
- How to apply error propagation to express uncertainty of depth?
- How to analyze the effect of short or large base-line?
- What is the closest depth observable by a stereo camera?
- How to compute mathematically the intersection of two lines (linear approx.)?

Understanding - self assessment 2/2

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

Appendix

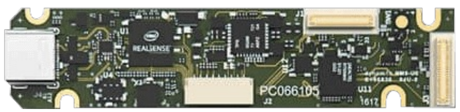
Examples

Short-base stereo cameras

- Ideal triangulation < 10 m
- Global shutter, onboard processing
- Integrated IMU, other sensors
- (visual odometry, SLAM, ROS ready)

https://www.intelrealsense.com

https://www.stereolabs.com



Mobile mapping stereo cameras

- Ideal triangulation < 50 m or laser-color
- Global shutter
- Integrated IMU, GNSS, other sensors
- Services, e.g. https://www.inovitas.ch
- Products, e.g. https://www.igi-systems.com/streetmapper.html

Triangulation – expressed in Python

Per point p1(3,i), p2(3,i) correspondences and projections Pi1(3,4), Pi2(3,4)

```
#1) Build matrix of linear homogeneous system of equations
        A1 = skewSymMtrx(p1[:, i]) @ Pi1
        A2 = skewSymMtrx(p2[:, i]) @ Pi2
        A = np.r_[A1, A2]

#2) Solve the homogeneous system of equations
        _, _, v = np.linalg.svd(A, full_matrices=False)
        P[:, i] = v.T[:,-1]

#3)De-homogenize (P is expressed in homogeneous coordinates)
    P /= P[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( p2[:,i], p1[:,i] ).T

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

The 8-point algorithm – E ”correction” in 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
```

4 possible solutions for R, t – the proof

- 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
- and $R_z\Sigma = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$
- R is orthogonal (product of 3 orthogonal matrices) if $[t_\times]^T = -[t_\times]$ then

$$E = -E$$

$$\texttt{det}(R) = -1$$