EE565:Mobile Robotics
Lecture 8

Welcome

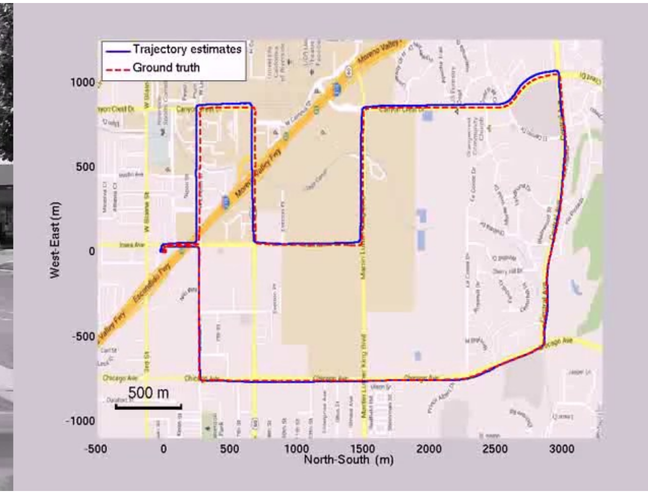
Dr. Ahmad Kamal Nasir

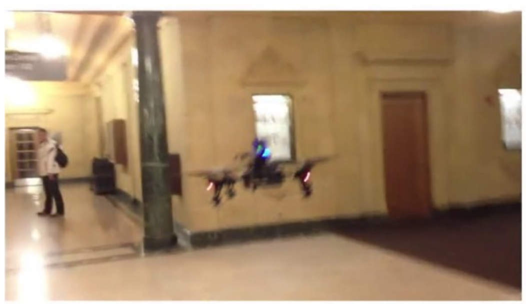
Today's Objectives

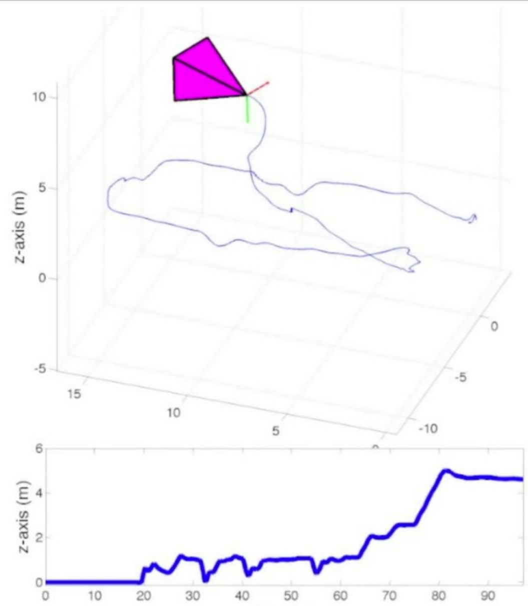
- Stereo Vision
- Stereo Rectification
- Structure From Motion (SFM) : Environment mapping (Structure), Robot/Camera pose estimation (Motion)
- Epi-polar geometry for multi-view Camera motion estimation

Last Week: Optical Flow (LKT)

Motivation


Trajectory estimates
Ground truth
West-East (m)
North-South (m)
500 m
1000
500
0
-500
-1000
-500
0
500
1000
1500
2000
2500
3000



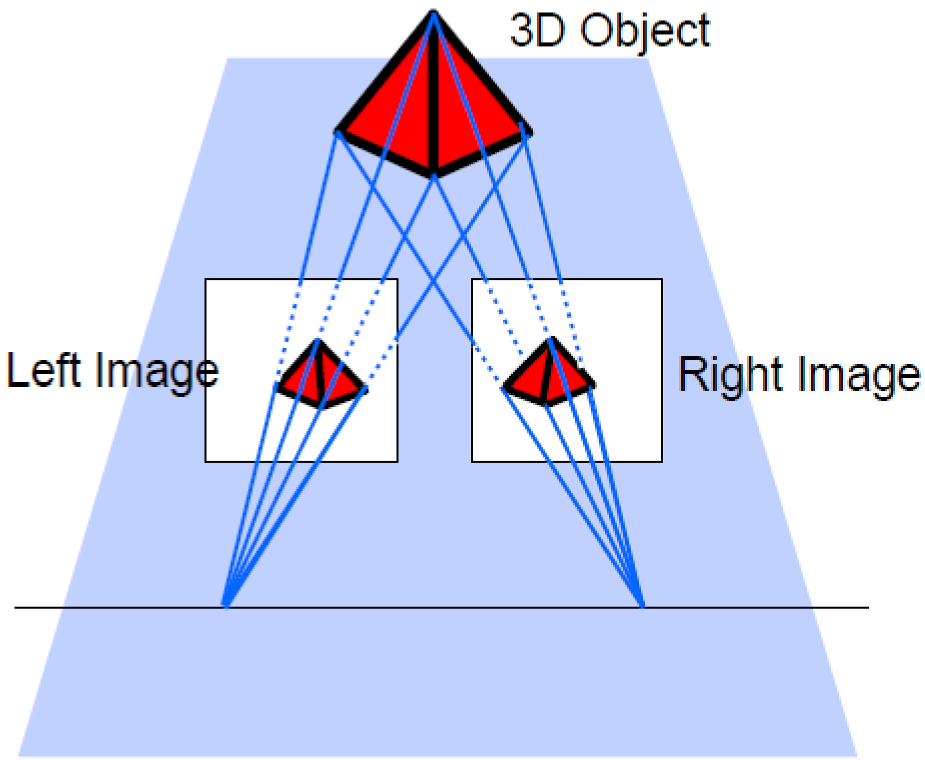

z-axis (m)
Time (s)

Stereo Vision versus Structure from Motion

- **Stereo vision:**
  - is the process of obtaining depth information from a pair of images coming from two cameras that look at the same scene from different but known positions
- **Structure from Motion:**
  - is the process of obtaining depth and motion information from a pair of images coming from the same camera that looks at the same scene from different positions

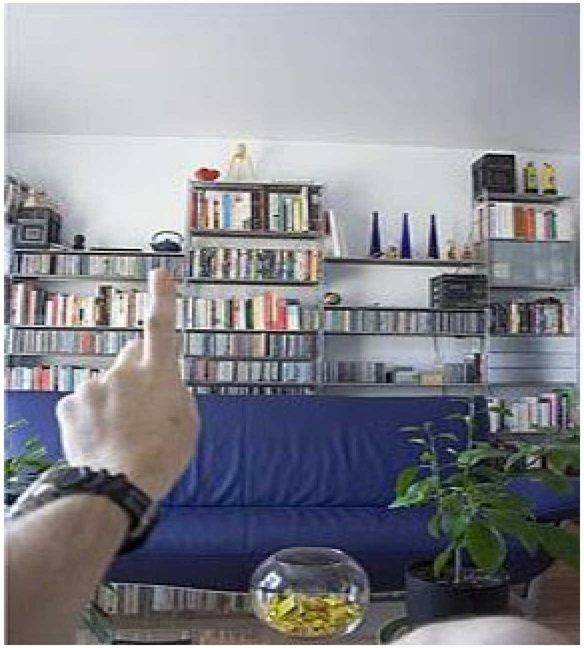
Stereo Vision: working principle

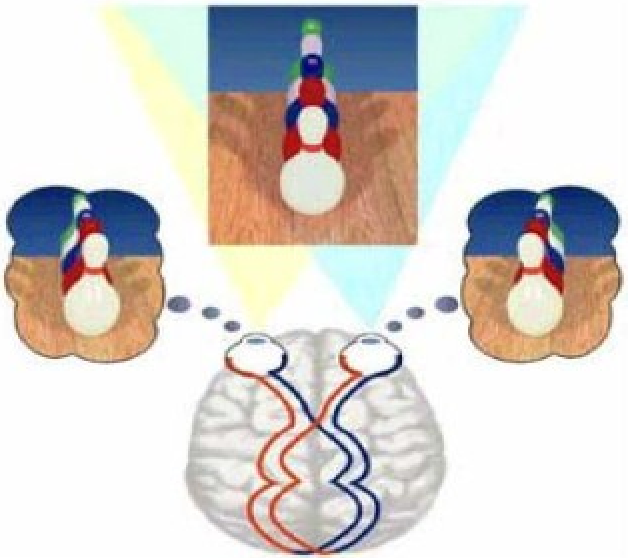
- Observe scene from two different viewpoints and solve for the intersection of the rays and recover the 3D structure



The “human” binocular system

- **Stereopsys:** the brain allows us to see the left and right retinal images as a single 3D image
- The images project on our retina up-side-down but our brains lets us perceive them as «straight». Radial disotion is also removed. This process is called «**rectification**»





The "human" binocular system

- **Stereopsys:** the brain allows us to see the left and right retinal images as a single 3D image
- The images project on our retina up-side-down but our brains lets us perceive them as «straight». Radial disotion is also removed. This process is called «**rectification**»

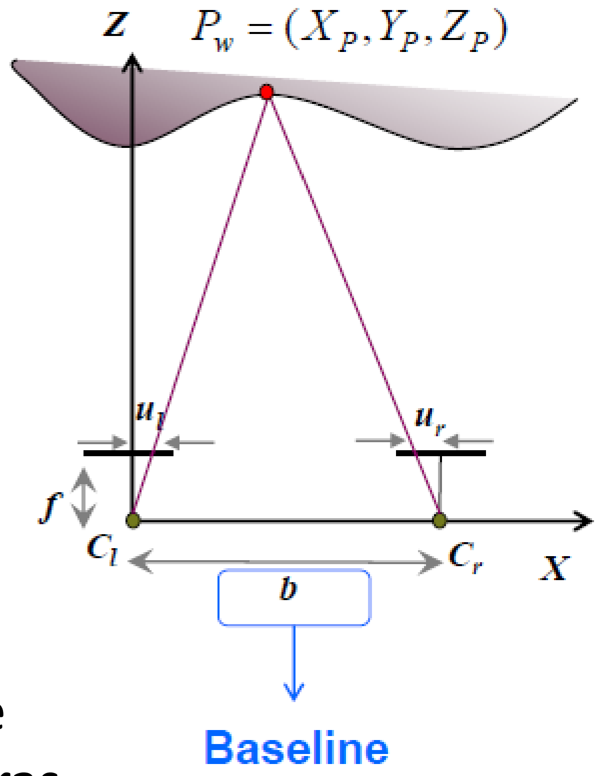
Make a simple test:
1. Fix an object
2. Open and close alternatively the left and right eyes.
- The horizontal displacement is called **disparity**
- The smaller the disparity, the farther the object

Stereo Vision: simplified case

- An ideal, simplified case assumes that both cameras are **identical** and **aligned** with the x-axis
- Can we find an expression for the depth Z_p of point P_w?
- From similar triangles:

$$\frac{f}{Z_P} = \frac{u_l}{X_P} \qquad \frac{f}{Z_P} = \frac{-u_r}{b - X_P} \quad \Rightarrow \quad Z_P = \frac{bf}{\boxed{u_l - u_r}}$$

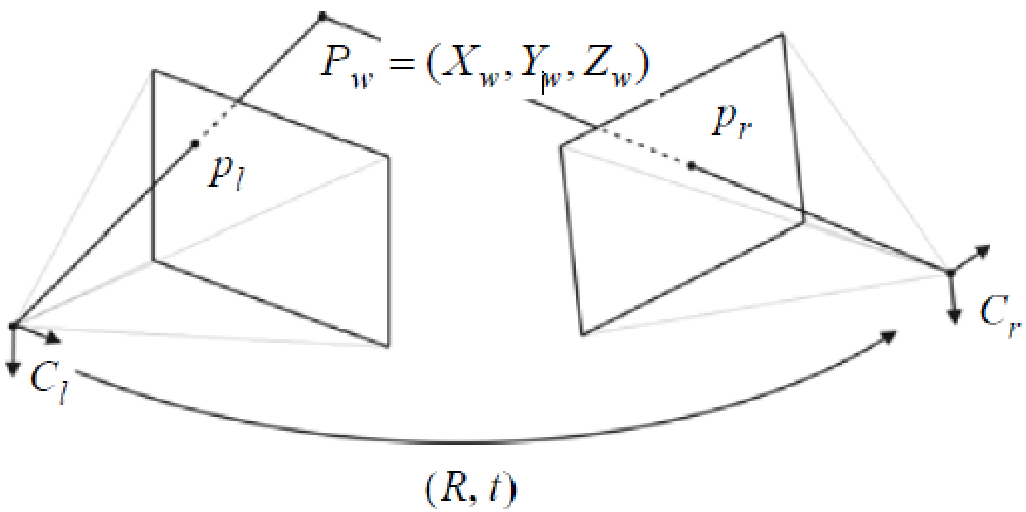
Disparity



- **Disparity** is the difference in image location of the projection of a 3D point in two image plane
- **Baseline** is the distance between the two cameras

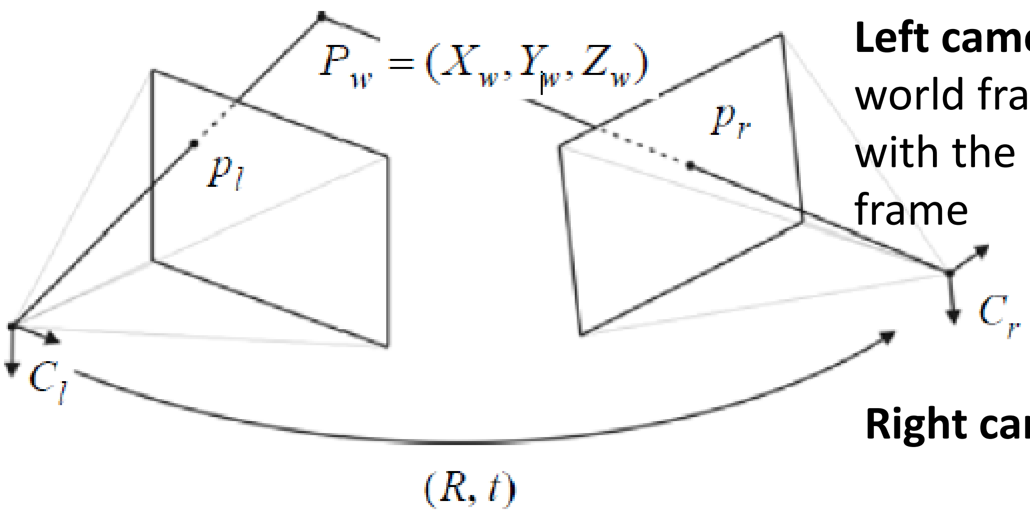
Stereo Vision: general case

- Two identical cameras do not exist in nature!
- Aligning both cameras on a horizontal axis is very difficult
- In order to use a stereo camera, we need to know the intrinsic extrinsic parameters of each camera, that is, the relative pose between the cameras (rotation, translation) ⇒ We can solve for this through camera calibration



Stereo Vision: general case

- To estimate the 3D position of P_w we can construct the system of equations of the left and right camera
- Triangulation is the problem of determining the 3D position of a point given a set of corresponding image locations and known



Left camera: set the world frame to coincide with the left camera frame

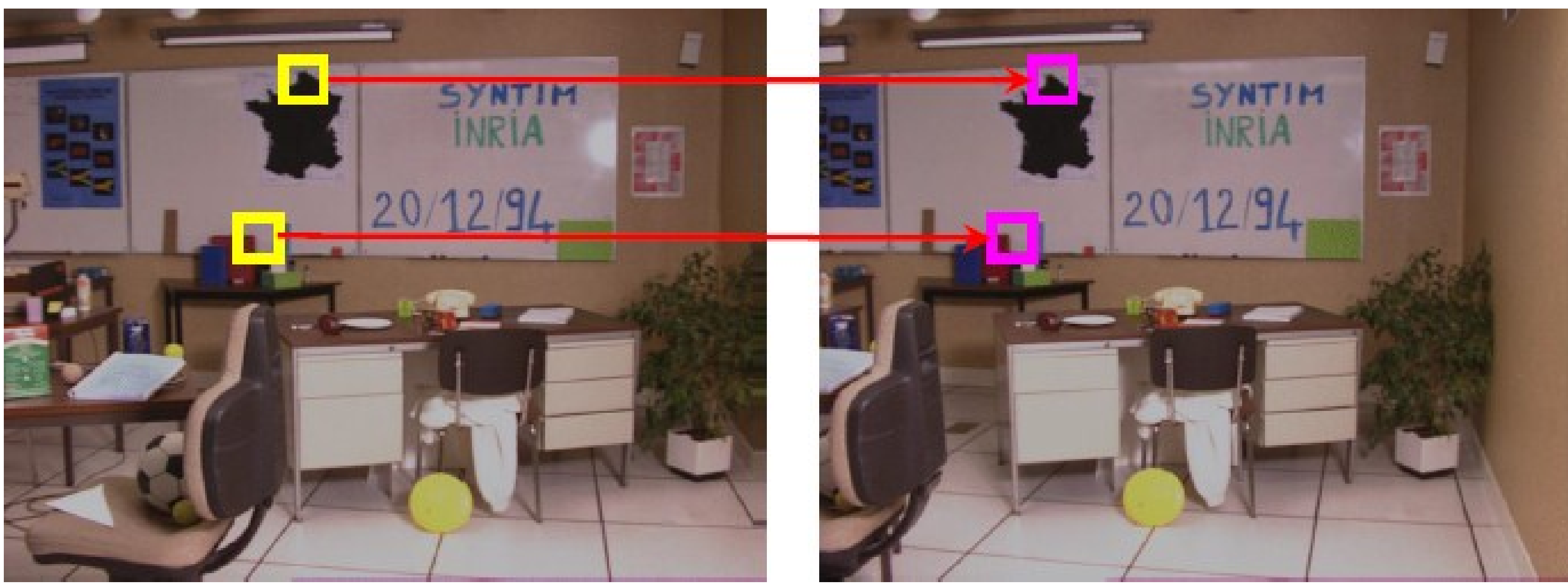
$$\tilde{p}_l = \lambda_l \begin{bmatrix} u_l \\ v_l \\ 1 \end{bmatrix} = K_l \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix}$$

Right camera:

$$\tilde{p}_r = \lambda_r \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = K_r R \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + T$$

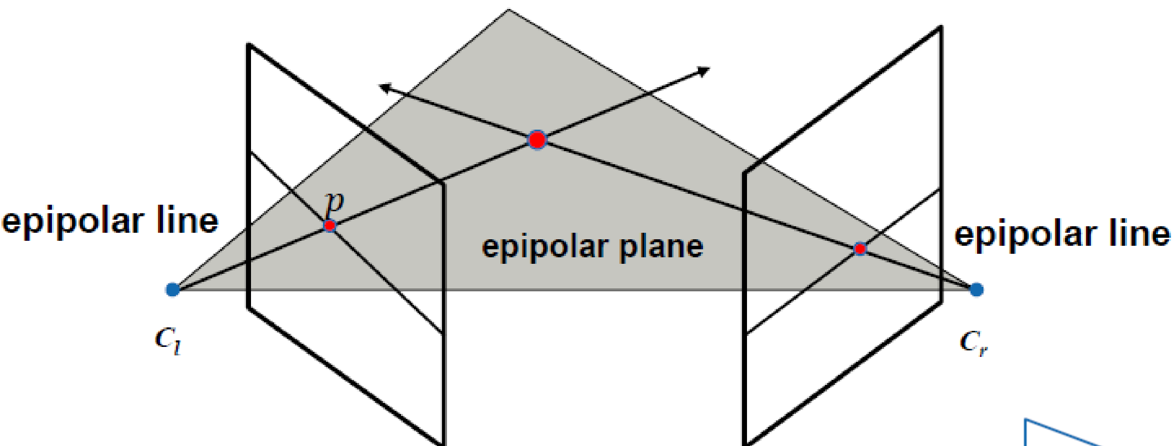
Stereo Vision: Correspondence Search

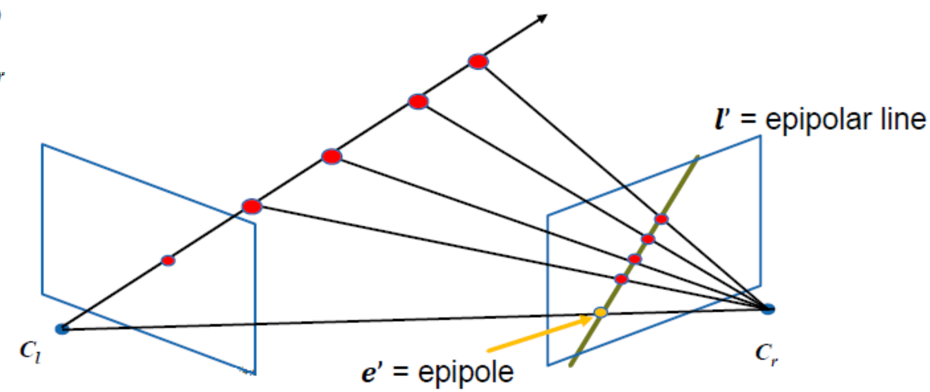
- **Goal:** identify corresponding points in the left and right images, which are the reprojection of the same 3D scene point
  - Typical similarity measures: Normalized Cross-Correlation (NCC) , Sum of Squared Differences (SSD), Census Transform
  - Exhaustive image search can be computationally very expensive! Can we make the correspondence search in 1D?



Stereo Vision: the epipolar constraint

- The epipolar plane is defined by the image point $\mathbf{p}$ and the optical centers
- Impose the epipolar constraint to aid matching: search for a correspondence along the epipolar line



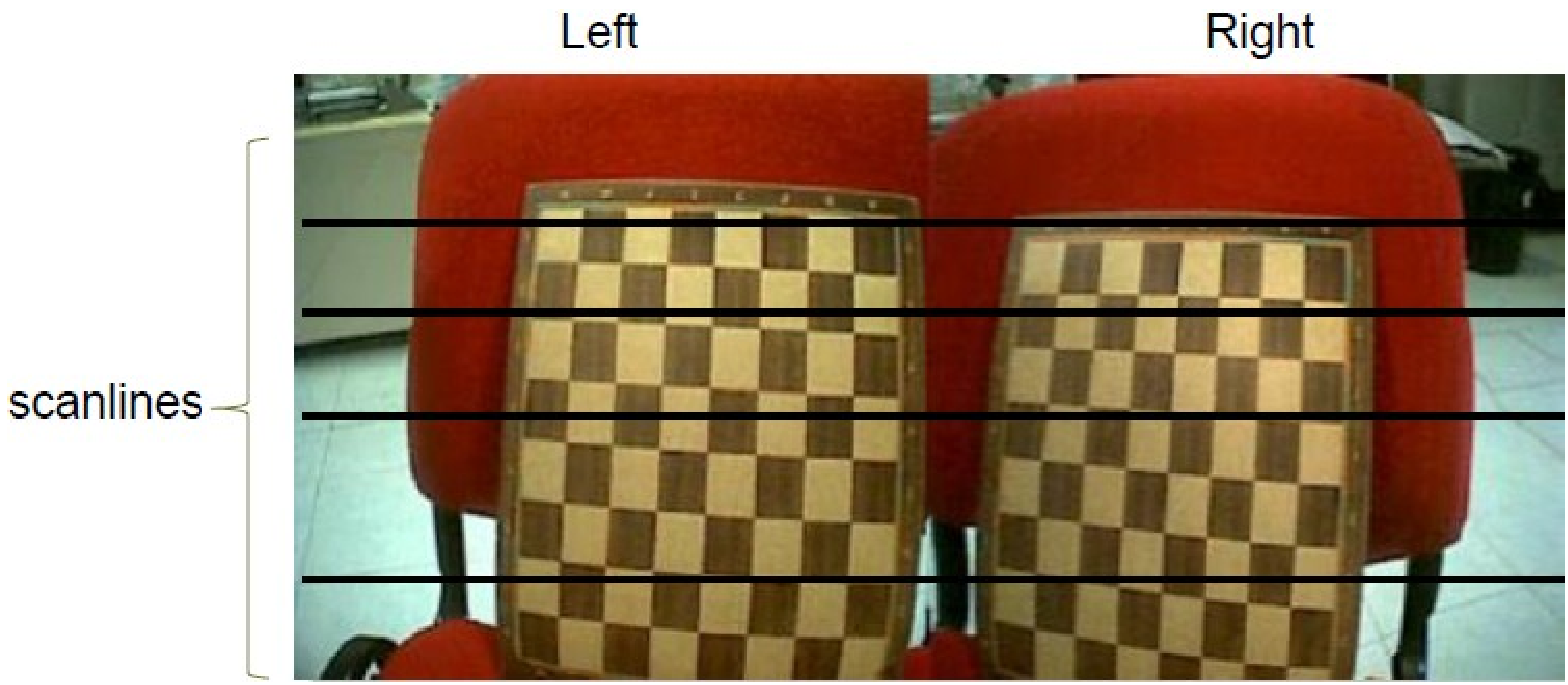


Stereo Vision: the epipolar constraint

- Using epipolar constraint, corresponding points can be searched for, along epipolar lines ⇒ computational cost reduced to 1 dimension!

Stereo Vision: Epipolar Rectification

- Goal: transform the left and right image so that pairs of conjugate epipolar lines become **collinear** and parallel to one of the image axes (usually the horizontal one)



Stereo Vision: Epipolar Rectification

- Goal: transform the left and right image so that pairs of conjugate epipolar lines become **collinear** and parallel to one of the image axes (usually the horizontal one)

Rotation

Image from Left Camera | Image from Right Camera

Stereo Vision: Epipolar Rectification

- Goal: transform the left and right image so that pairs of conjugate epipolar lines become **collinear** and parallel to one of the image axes (usually the horizontal one)

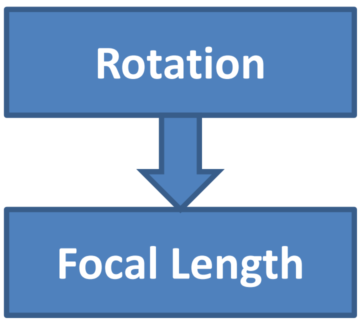
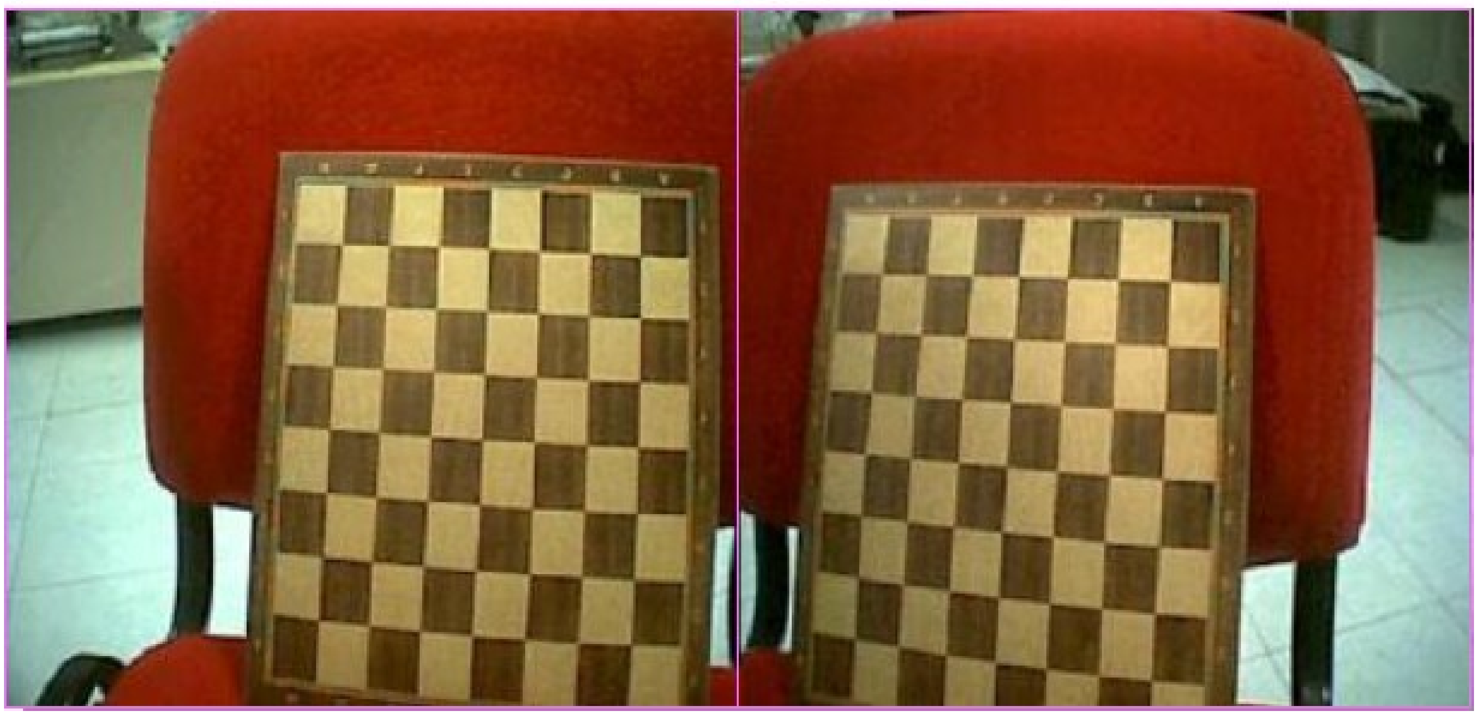
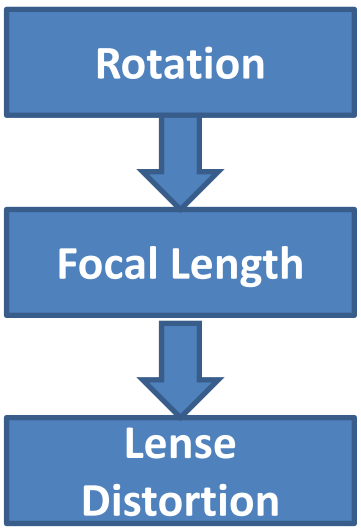


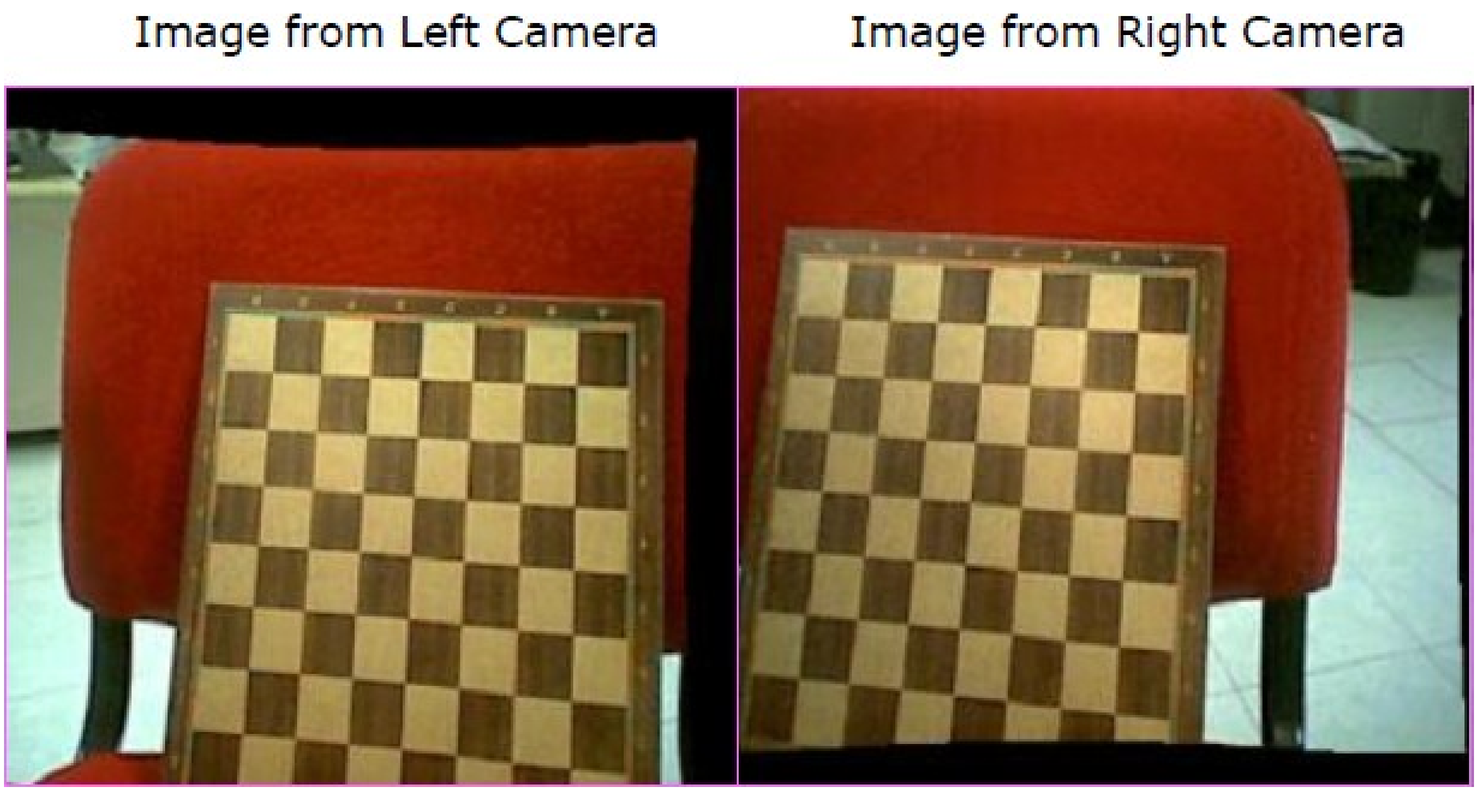
Image from Left Camera　　Image from Right Camera



Stereo Vision: Epipolar Rectification

- Goal: transform the left and right image so that pairs of conjugate epipolar lines become **collinear** and parallel to one of the image axes (usually the horizontal one)





Stereo Vision: Epipolar Rectification

- Goal: transform the left and right image so that pairs of conjugate epipolar lines become **collinear** and parallel to one of the image axes (usually the horizontal one)

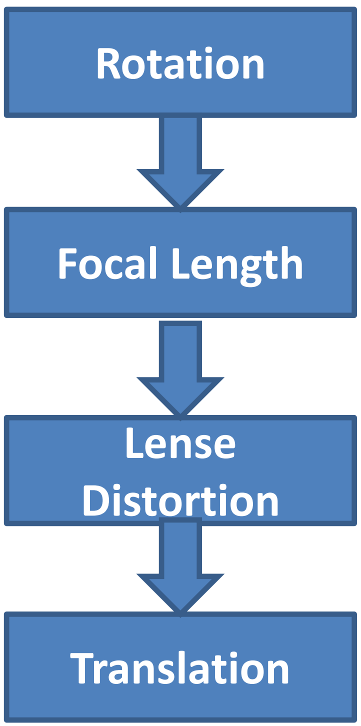
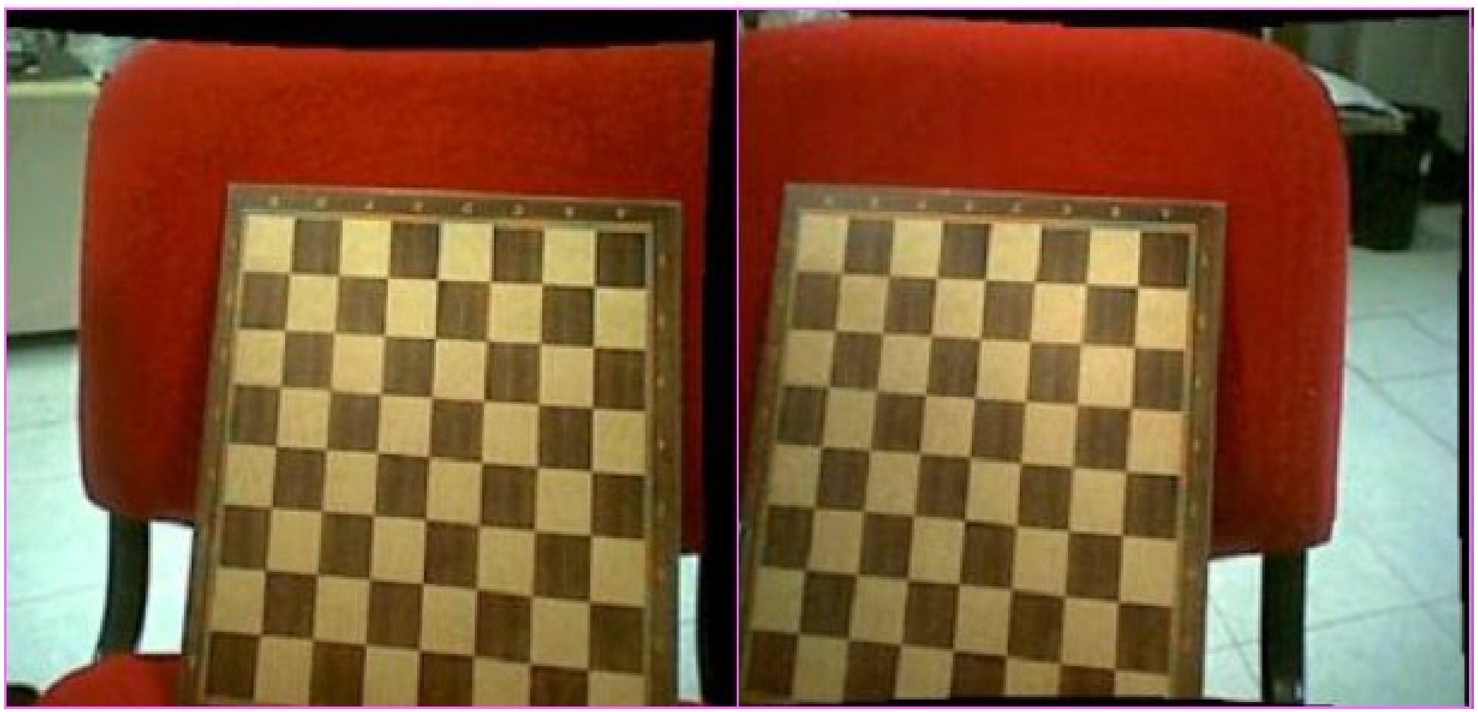


Image from Left Camera Image from Right Camera



Stereo Vision: Epipolar Rectification

- Goal: transform the left and right image so that pairs of conjugate epipolar lines become **collinear** and parallel to one of the image axes (usually the horizontal one)

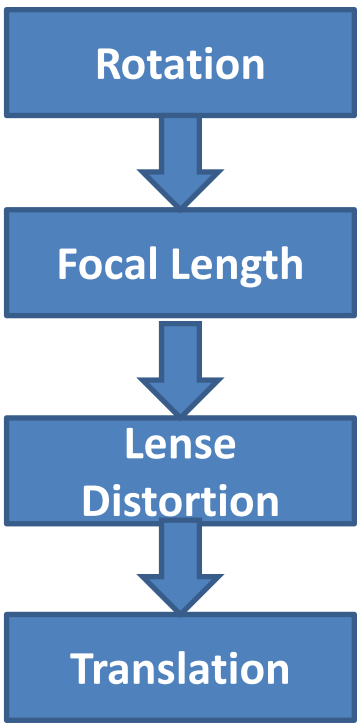
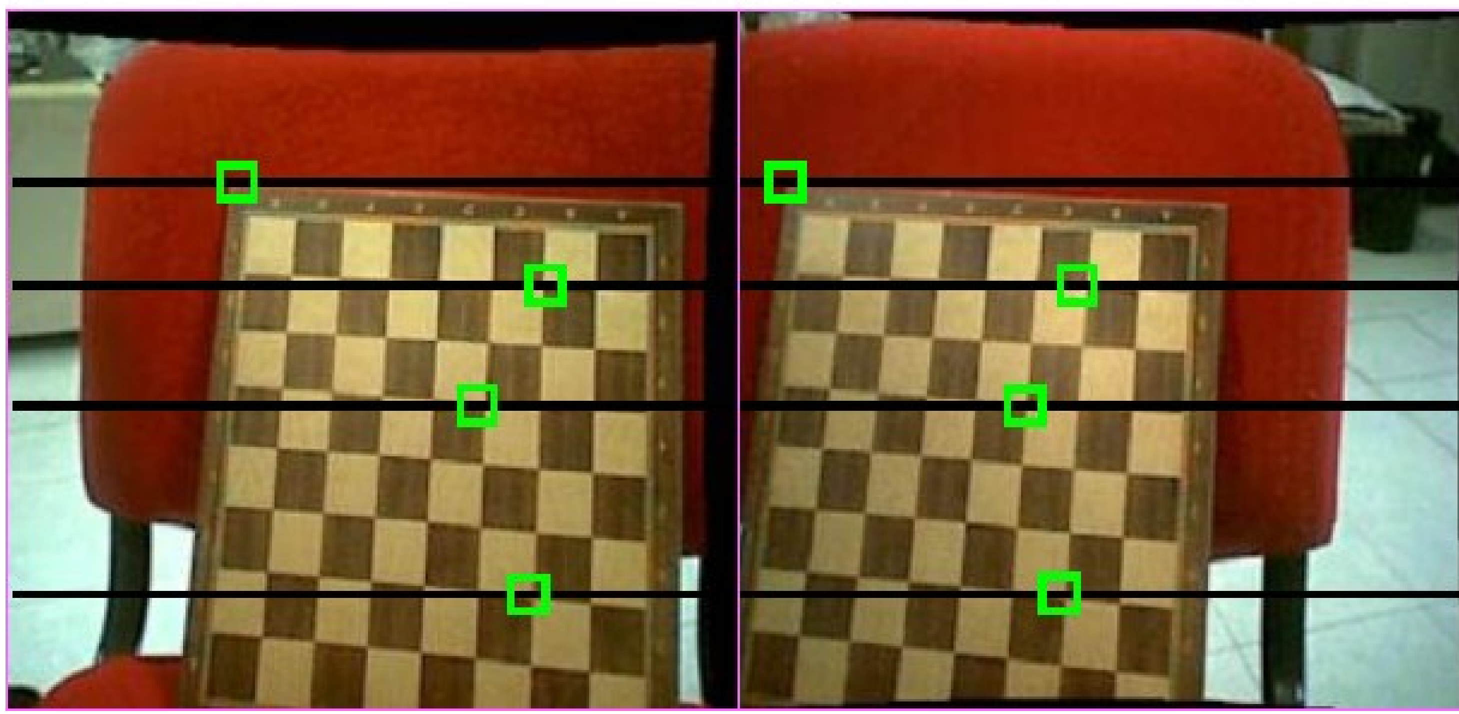


Image from Left Camera   Image from Right Camera



Stereo Vision: disparity map

- The disparity map holds the disparity value at every pixel:
  - Identify correspondent points of all image pixels in the original images
  - Compute the disparity $(ul - ur)$ for each pair of correspondences
- Usually visualized in gray-scale images
- Close objects experience bigger disparity; thus, they appear brighter in disparity map

Left image

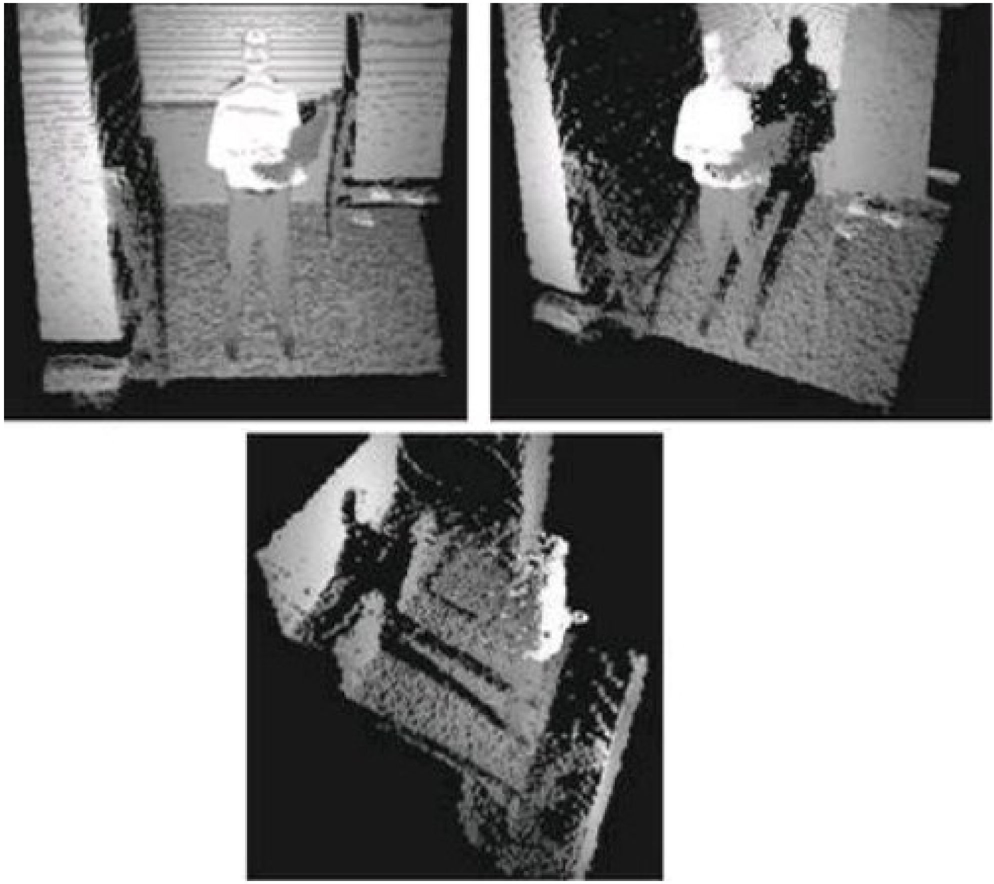
Right image

Disparity Map

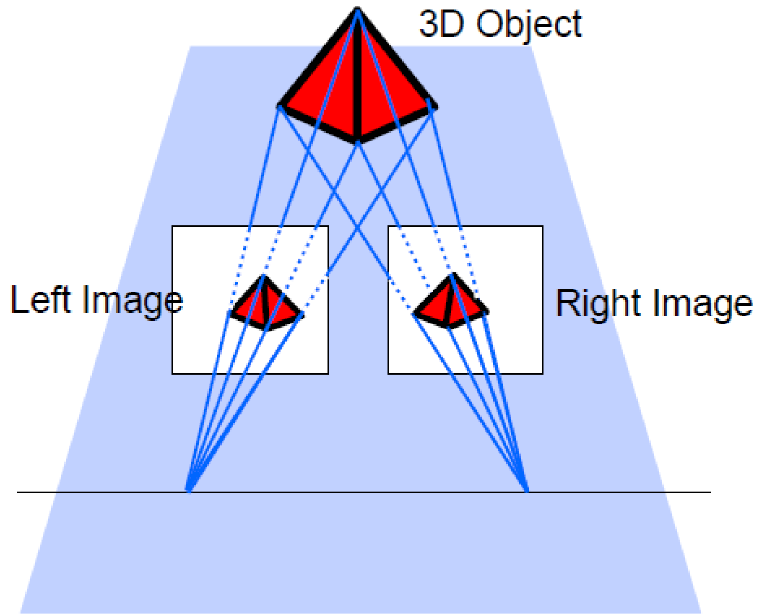
Stereo Vision: disparity map

- The disparity map holds the disparity value at every pixel:
  - Identify correspondent points of all image pixels in the original images
  - Compute the disparity $(ul - ur)$ for each pair of correspondences
- Usually visualized in gray-scale images
- Close objects experience bigger disparity; thus, they appear brighter in disparity map

$$Z = \frac{bf}{u_l - u_r}$$



Stereo Vision: Summary

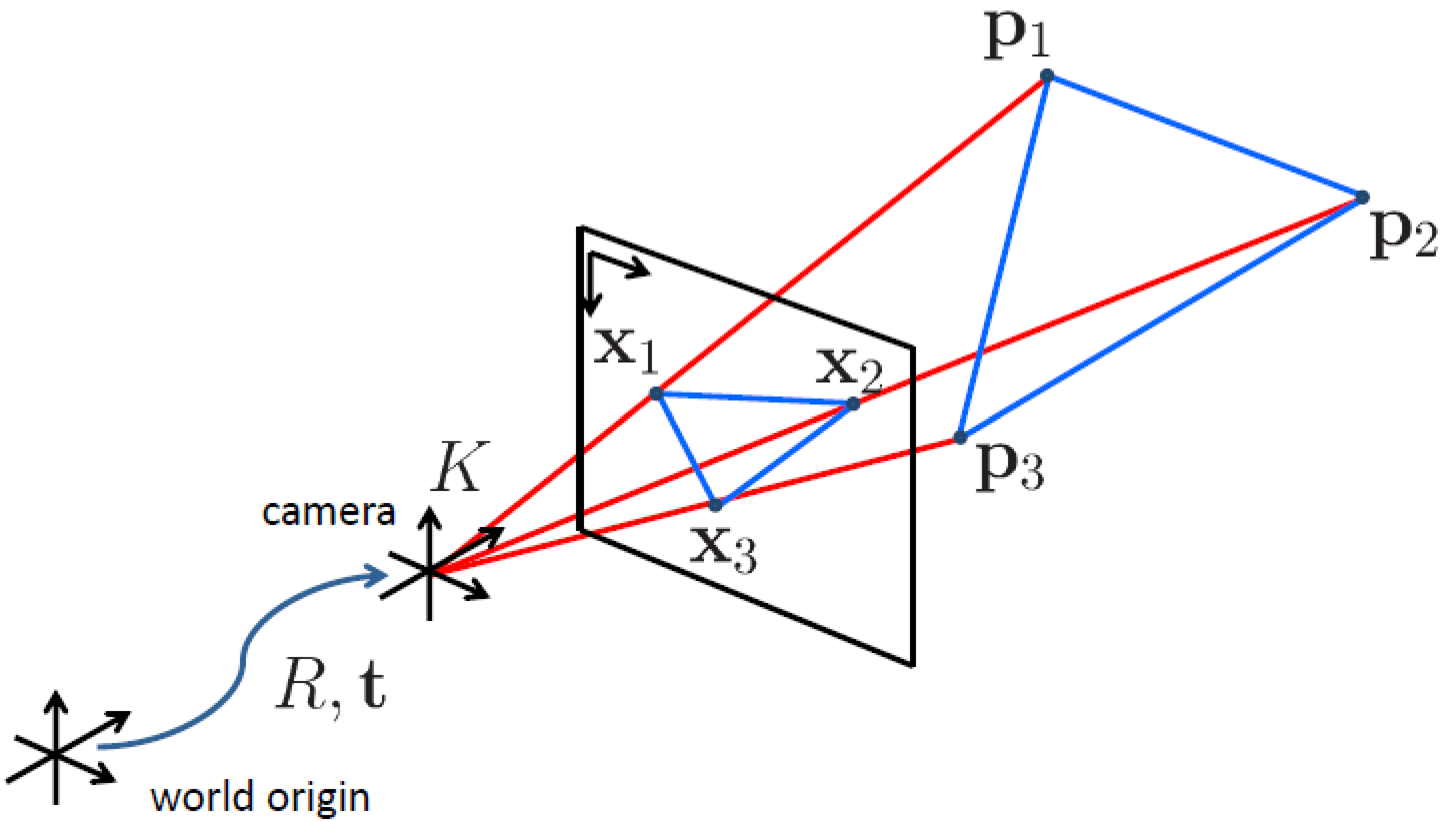


- Stereo camera calibration ⇒ compute camera relative pose
- Epipolar rectification ⇒ align images & epipolar lines
- Search for correspondences
- Output: compute stereo triangulation or disparity map
- Consider how baseline & image resolution affect accuracy of depth estimates

Structure From Motion

- **Camera calibration / resection** Known 3D points, observe corresponding 2D points, compute camera pose
- **Point triangulation** Known camera poses, observe 2D point correspondences, compute 3D point
- **Motion estimation** Observe 2D point correspondences, compute camera pose (up to scale)
- **Bundle adjustment** Observe 2D point correspondences, compute camera pose and 3D points (up to scale)

Camera Calibration (Perspective n-Point Problem)



Camera Calibration

- **Given**: n 2D/3D correspondences $x_i \leftrightarrow p_i$
- **Wanted:** $M = K \cdot [R|T]$
  such that $\widetilde{x}_i = M \cdot p_i$
- Question: How many DOFs does have?
- The algorithm has two parts:
  - Compute $M \in \mathbb{R}^{3\times4}$
  - Decompose M into K, R, T via QR decomposition

Estimate M

- $\widetilde{x_i} = M \cdot p_i$
- Each correspondence generates two equations

$$x = \frac{m_{11}X + m_{12}Y + m_{13}Z + m_{14}W}{m_{31}X + m_{32}Y + m_{33}Z + m_{34}W} \qquad y = \frac{m_{21}X + m_{22}Y + m_{23}Z + m_{24}W}{m_{31}X + m_{32}Y + m_{33}Z + m_{34}W}$$

- Re-arranged in matrix form

$$\begin{pmatrix} X & Y & Z & 1 & 0 & 0 & 0 & 0 & -xX & -xY & -xZ & -x \\ 0 & 0 & 0 & 0 & X & Y & Z & 1 & -yX & -yY & -yZ & -y \end{pmatrix} \mathbf{m} = \mathbf{0}$$

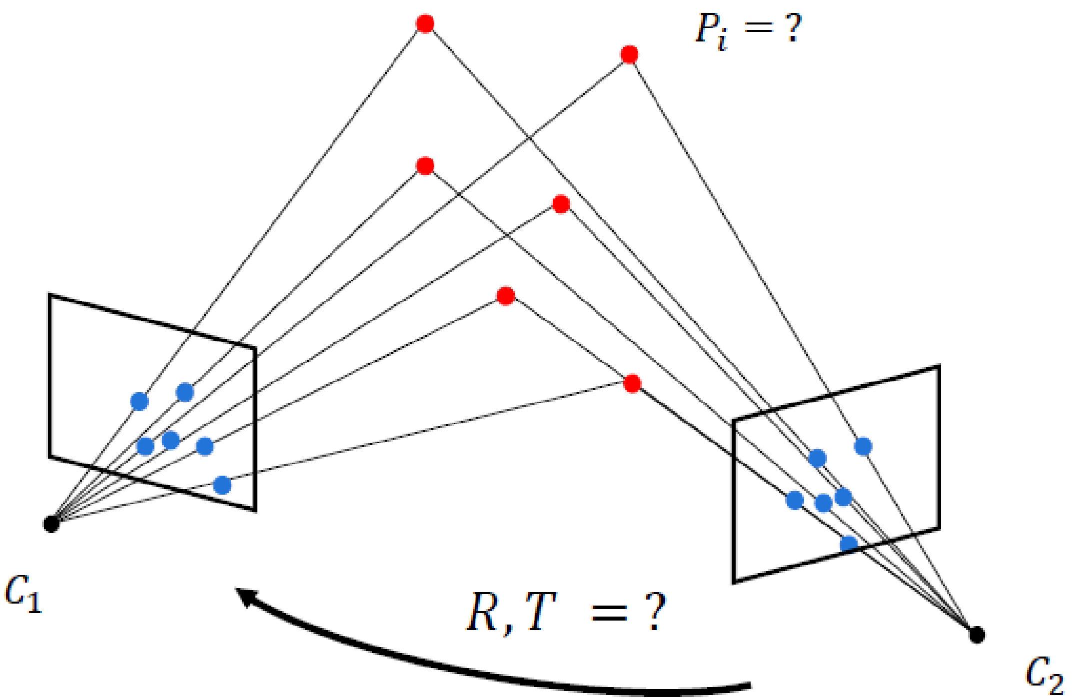
with $\mathbf{m} = (m_{11} \quad m_{12} \quad \dots \quad m_{34}) \in \mathbb{R}^{12}$

- Concatenate equations for n≥6 correspondences $\boldsymbol{A} \cdot \boldsymbol{m} = \mathbf{0}$, use **SVD**

Structure from Motion: definition

- **Problem formulation:** Given many points *correspondence* between two images, $\{(u_1^i, v_1^i), (u_2^i, v_2^i)\}$, simultaneously compute the 3D location $\boldsymbol{P}_i$, the camera relative-motion parameters $(\boldsymbol{R}, \boldsymbol{t})$, and camera intrinsic $\boldsymbol{K}_{1,2}$ that satisfy

$$\begin{cases} \lambda_1 \begin{bmatrix} u^{i}_{1} \\ v^{i}_{1} \\ 1 \end{bmatrix} = K_1 [0|0] \cdot \begin{bmatrix} X^{i}_{w} \\ Y^{i}_{w} \\ Z^{i}_{w} \\ 1 \end{bmatrix} \\ \lambda_2 \begin{bmatrix} u^{i}_{2} \\ v^{i}_{2} \\ 1 \end{bmatrix} = K_2 [R|T] \cdot \begin{bmatrix} X^{i}_{w} \\ Y^{i}_{w} \\ Z^{i}_{w} \\ 1 \end{bmatrix} \end{cases}$$



Structure from Motion: definition

- We study the case in which the camera is «calibrated» (K is known)
- Thus, we want to find R, T, Pi that satisfy

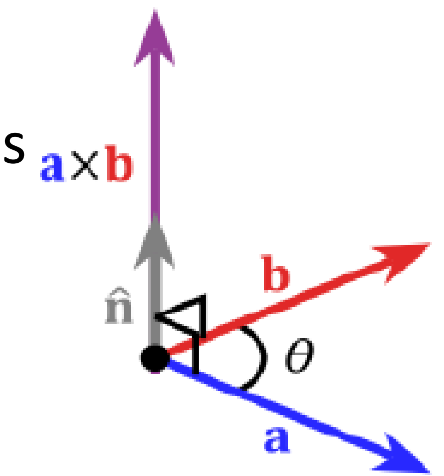
$$\begin{cases} \lambda_1 \begin{bmatrix} \bar{u}^i{}_1 \\ \bar{v}^i{}_1 \\ 1 \end{bmatrix} = [I|0] \cdot \begin{bmatrix} X^i{}_w \\ Y^i{}_w \\ Z^i{}_w \\ 1 \end{bmatrix} \\ \lambda_2 \begin{bmatrix} \bar{u}^i{}_2 \\ \bar{v}^i{}_2 \\ 1 \end{bmatrix} = [R|T] \cdot \begin{bmatrix} X^i{}_w \\ Y^i{}_w \\ Z^i{}_w \\ 1 \end{bmatrix} \end{cases}$$

Structure from Motion: how many points?

- How many knowns and unknowns?
  - $\boldsymbol{4n}$ knowns:
    - n correspondences; each one (u_1^i, v_1^i) and (u_2^i, v_2^i), $i = 1 \ldots n$
  - $\boldsymbol{5 + 3n}$ unknowns
    - 5 for the motion up to a scale (rotation $\mapsto$ 3, translation $\mapsto$ 2)
    - $3n$ = number of coordinates of the n 3D points
- Does a solution exist?
  - Yes, if and only if the number of independent equations ≥ number of unknowns $\Rightarrow 4n \geq 5 + 3n \Rightarrow \boxed{n \geq \mathbf{5}}$

Cross Product (or Vector Product): Review

$$\vec{a} \times \vec{b} = \vec{c}, \quad \|\vec{c}\| = \|\vec{a}\| \|\vec{b}\| \sin(\theta) \cdot \vec{n}$$



- Vector cross product takes two vectors and returns a third vector that is perpendicular to both inputs

$$\vec{a} \cdot \vec{c} = 0$$

$$\vec{b} \cdot \vec{c} = 0$$

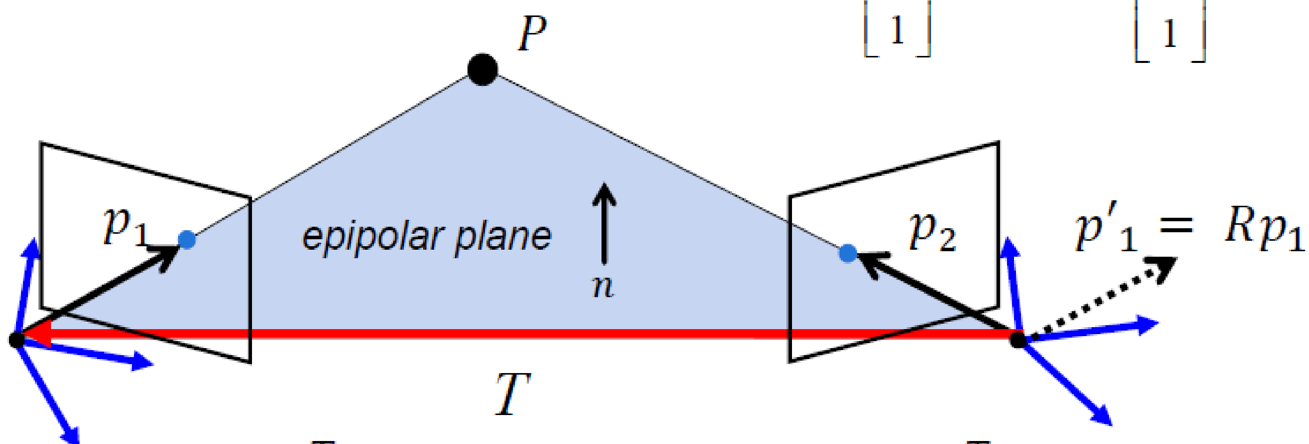
- The cross product of two parallel vectors = 0
- The vector cross product also can be expressed as the product of a skew-symmetric matrix and a vector

$$\mathbf{a} \times \mathbf{b} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = [\mathbf{a}_\times]\mathbf{b}$$

Epipolar Geometry

- $p1, p2, T$ are coplanar:

$$p_1 = \begin{bmatrix} \bar{u}_1 \\ \bar{v}_1 \\ 1 \end{bmatrix} \qquad p_2 = \begin{bmatrix} \bar{u}_2 \\ \bar{v}_2 \\ 1 \end{bmatrix}$$



$$p_2^T \cdot n = 0 \quad \Rightarrow \quad p_2^T \cdot (T \times p_1') = 0 \quad \Rightarrow p_2^T \cdot (T \times (Rp_1)) = 0$$

$$\Rightarrow p_2^T [T]_\times R\, p_1 = 0 \Rightarrow \boxed{p_2^T\, E\, p_1 = 0 \quad \textit{epipolar constraint}}$$

$$\boxed{\mathrm{E} = [T]_\times R \qquad \textit{essential matrix}}$$

Epipolar Geometry

- The Essential Matrix can be computed from 5 image correspondences [Kruppa, 1913].
  - The more points, the higher accuracy
- The Essential Matrix can be decomposed into R and T by recalling that $E = [T \times]R$ Two distinct solutions for R and T are possible (i.e., 4-fold ambiguity)

$$p_1 = \begin{bmatrix} \bar{u}_1 \\ \bar{v}_1 \\ 1 \end{bmatrix} \quad p_2 = \begin{bmatrix} \bar{u}_2 \\ \bar{v}_2 \\ 1 \end{bmatrix} \quad \textit{Normalized image coordinates}$$

$$\boxed{\begin{array}{ll} p_2^T\, E\, p_1 = 0 & \textit{Epipolar constraint} \\ \\ \mathrm{E} = [T]_\times R & \textit{Essential matrix} \end{array}}$$

How to compute the Essential Matrix?

- The Essential Matrix can be computed from 5 image correspondences [Kruppa, 1913]. However, this solution is not simple. It took almost one century until an efficient solution was found! [Nister, CVPR'2004]
- The first popular solution uses 8 points and is called **8-point algorithm** [Longuet Higgins, 1981]

Motion Estimation: The 8-point algorithm

$$\boldsymbol{p_1} = (\bar{u}_1, \bar{v}_1, 1)^T, \quad \boldsymbol{p_2} = (\bar{u}_2, \bar{v}_2, 1) \qquad p_2^T \, E \, p_1 = 0$$

$$\begin{bmatrix} \bar{u}_2 & \bar{v}_2 & 1 \end{bmatrix} \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} \begin{bmatrix} \bar{u}_1 \\ \bar{v}_1 \\ 1 \end{bmatrix} = 0 \Rightarrow \underbrace{\begin{bmatrix} u_2 u_1 & u_2 v_1 & u_2 & v_2 u_1 & v_2 v_1 & v_2 & u_1 & v_1 & 1 \end{bmatrix}}_{\text{Q (this matrix is \textbf{known})}} \underbrace{\begin{bmatrix} e_{11} \\ e_{12} \\ e_{13} \\ e_{21} \\ e_{22} \\ e_{23} \\ e_{31} \\ e_{32} \\ e_{33} \end{bmatrix}}_{\text{E (this matrix is \textbf{unknown})}} = 0$$

Minimize:

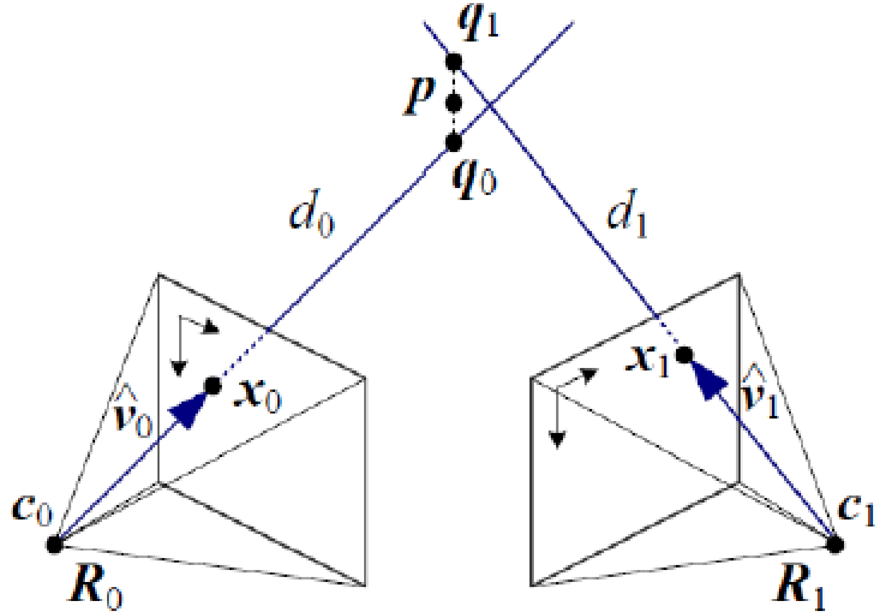
$$\sum_{i=1}^{N} (p_1^{iT} \boldsymbol{E} \, p^i{}_1)^2 :$$

under the constraint $\|\boldsymbol{E}\|^2 = 1$

- A linear least-square solution is given through Singular Value Decomposition by the eigenvector of Q corresponding to its smallest eigenvalue (which is the unit vector that minimizes $|Q \cdot E|^2$)

Structure Estimation: Triangulation

- **Given:** n cameras
  - $M_j = K_j \cdot [R_j|t_j]$
  - point correspondences x_0, x_1
- **Wanted:** Corresponding 3D point p



Structure from Motion: Summary

- **Given:** Image pair and camera Intrinsic parameters

$$K = \begin{pmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}$$

- **Find:** Camera motion R,t (up to scale)
  - Compute correspondences
  - Compute essential matrix
  - Extract camera motion
  - Extract scene structure (triangulation)

Summary

- Stereo Vision
- Stereo Rectification
- Structure From Motion (SFM) : Environment mapping (Structure), Robot/Camera pose estimation (Motion)
- Epi-polar geometry for multi-view Camera motion estimation

Questions