Robot Vision

TTK4255

Lecture 07 – Multiple-View Geometry

Annette Stahl

(Annette.Stahl@ntnu.no)

Department of Engineering Cybernetics – ITK

NTNU, Trondheim

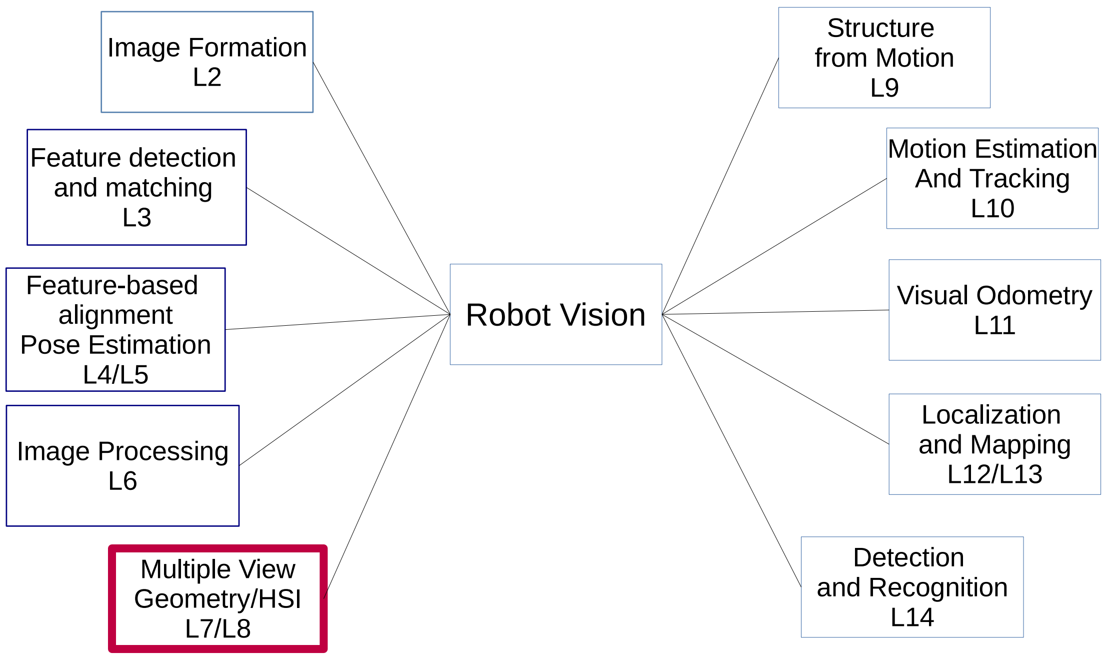
Spring Semester

17. February 2020

Lecture 07 – Multiple View Geometry

Annette Stahl (Annette.Stahl@ntnu.no)

Simen Haugo (Simen.Haugo@ntnu.no)



Outline of the fifth lecture:

- Two-view Geometry
- Triangulation
- Structure from Motion
- Epipolar Geometry
- Correspondence Problem
- Disparity and Depth

Recap L06

Image Processing

- Image Enhancement
- Thresholding
- Convolution and Correlation
- Filtering
- Edge Detection
- Line Detection (Hough-Transform)

Multiple-View

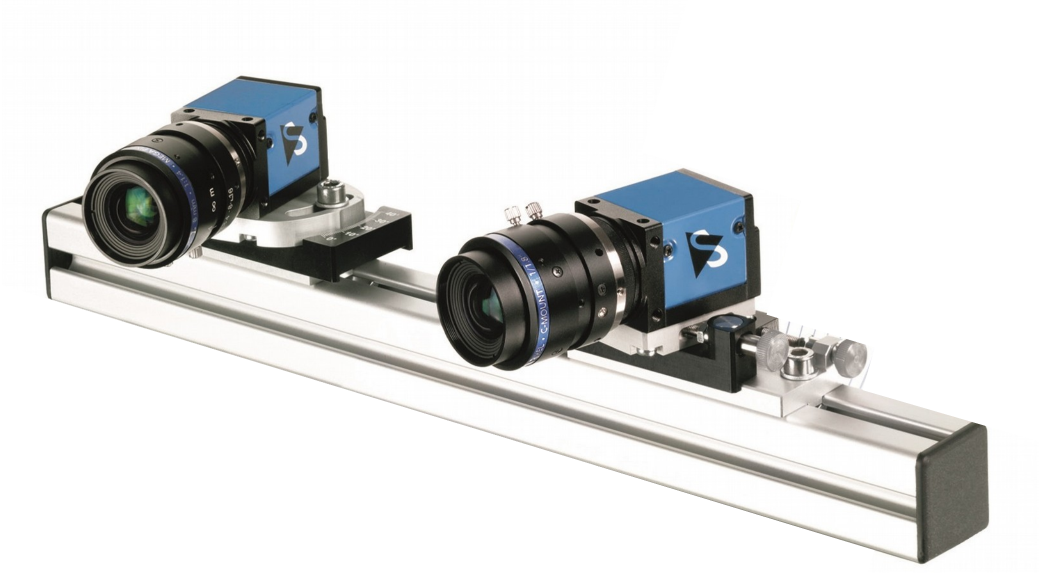
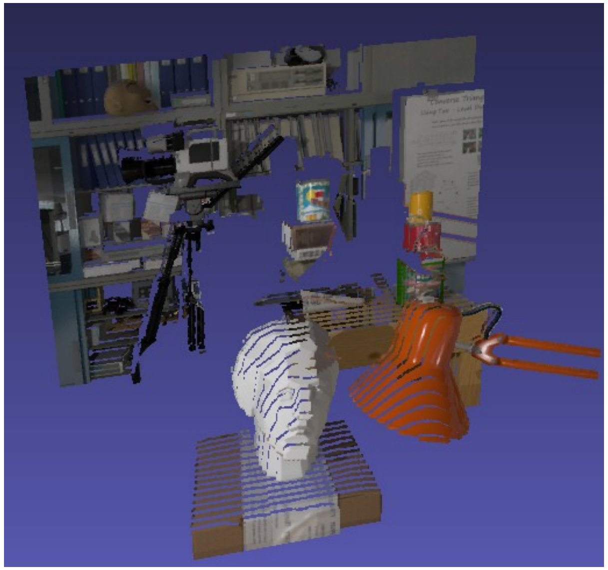


Image courtesy: opli.net

Image courtesy: breezesys.com

Multiple View - Motivation

- 3D perception of our/robot surrounding → 3D (object/scene) reconstruction
- Calculate object geometry
- Manipulate objects with robotic arms
- Avoid obstacles during robotic navigation
- Many others applications…



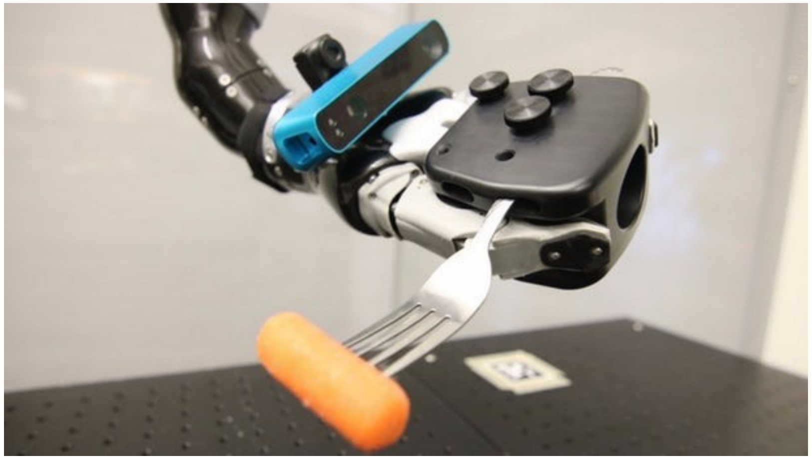
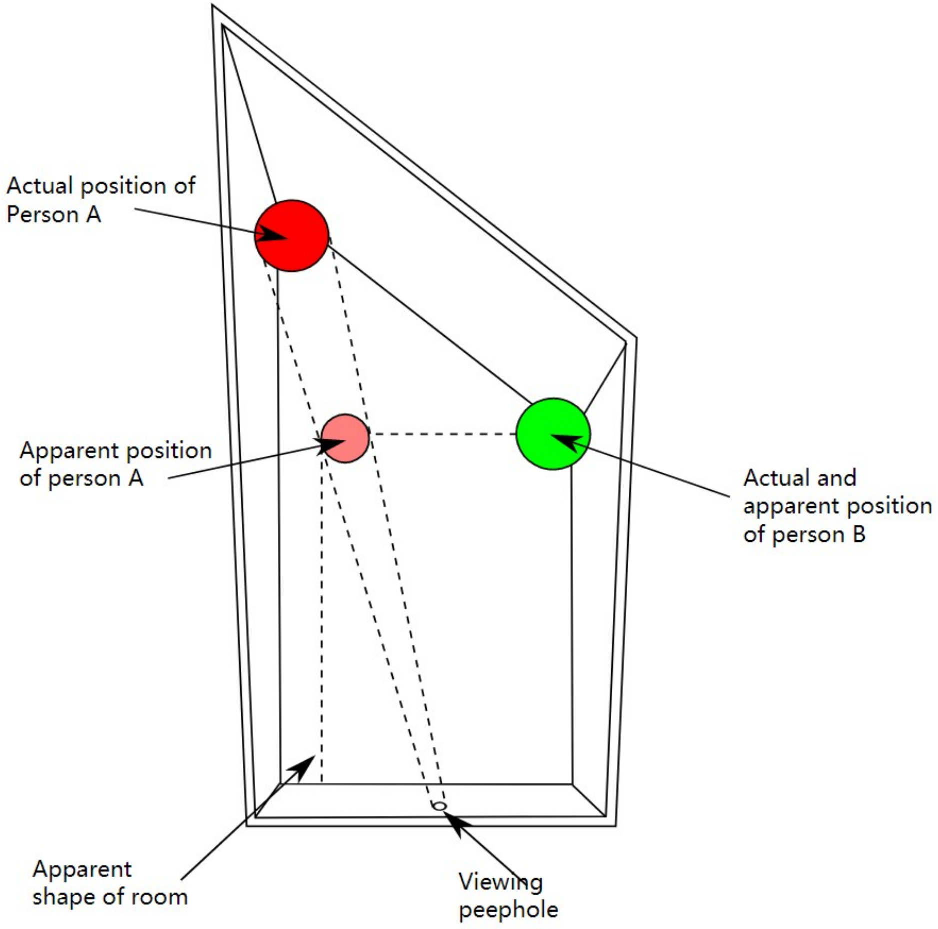


Image courtesy: “Carnegie Mellon University“

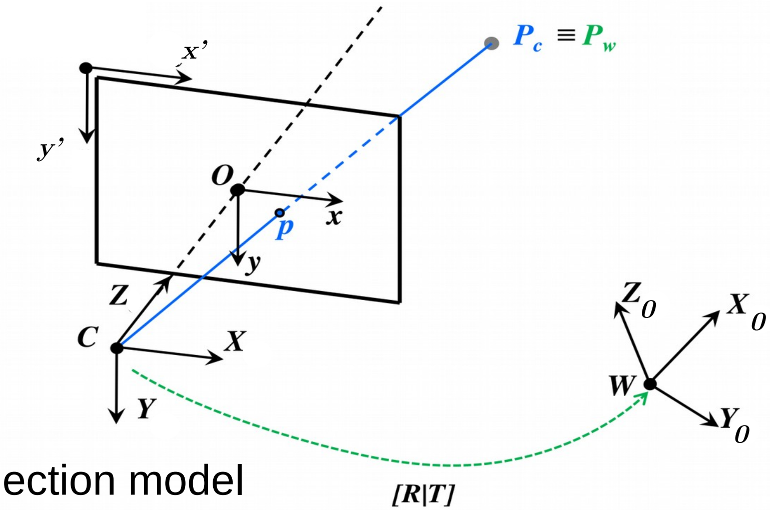
Stereo Vision - Motivation

3D measurements from one camera?

Ames room



Two-View Geometry



Single-View Geometry:

- Calibration and undistortion
- Given 3D – 2D correspondences $\mathbf{x_i} \leftrightarrow \mathbf{x_i'}, i = 1, 2, ..., n$
- Estimate projection matrix P that satisfies the perspective projection model

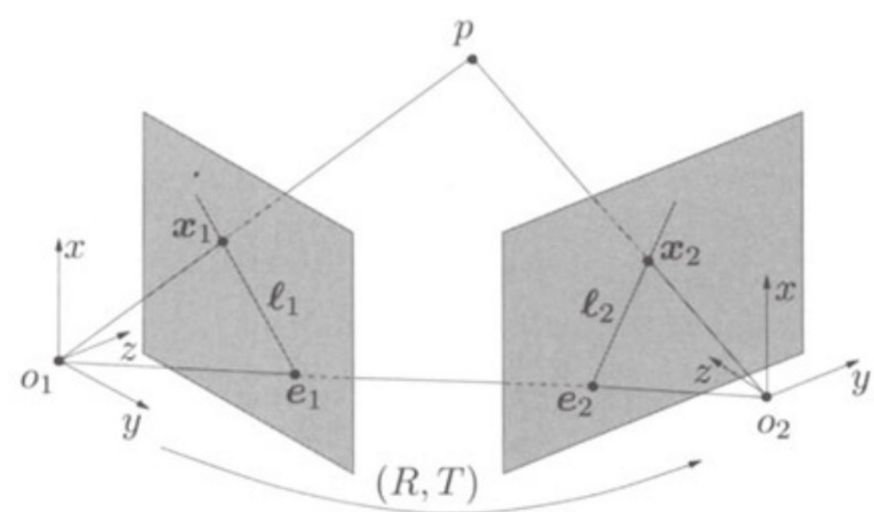
$$\tilde{\mathbf{x}}_\mathbf{i}' = K[R|T]\tilde{\mathbf{x}}_\mathbf{i} \qquad P = K[R|T]$$

$$\tilde{\mathbf{x}}' = P\tilde{\mathbf{x}}$$

$$\lambda \begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix} = \begin{bmatrix} fs_x & fs_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}$$

Two-View Geometry:

- Single camera calibration and undistortion
- Stereo calibration and rectification
- Epipolar Geometry describing the geometric relationship between two perspective cameras
- Estimation of $\tilde{\mathbf{x}}_\mathbf{1}' = P_1\tilde{\mathbf{x}} \qquad \tilde{\mathbf{x}}_\mathbf{2}' = P_2\tilde{\mathbf{x}}$
- Essential and Fundamental Matrix



Typical Stereo Vision Systems

Stereo vision system can be composed into four main processing steps:

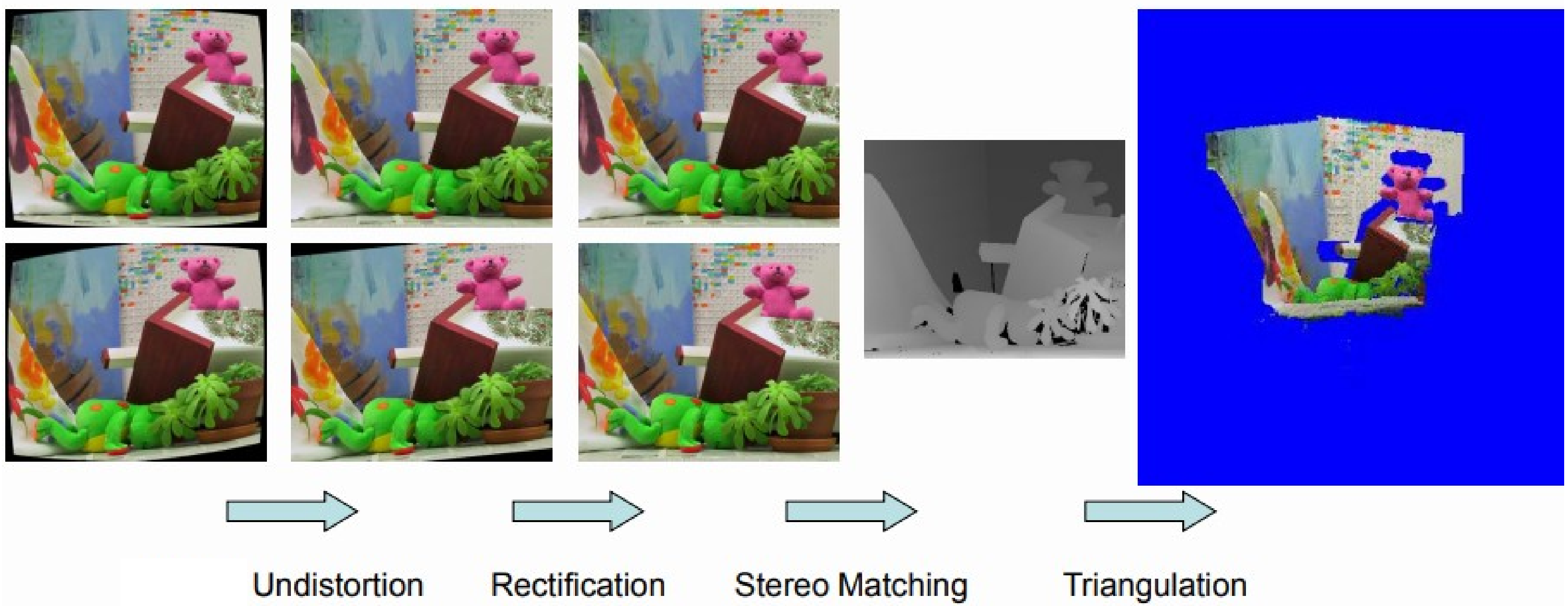


Image courtesy: Christian Unger, TUM

1. Single Camera Calibration (left and right camera): (Radial) Undistortion
2. Image Rectification: Minimize Image Distortion
3. Stereo Matching: Compute Disparity
4. Triangulation: 3D reconstruction

Two-View Geometry

Depth from stereo vision - Triangulation

Given: Projection matrices, 2D point correspondence between two images

Aim: Recover the 3D structure from images

Two-View Structure from Motion

Given: n 2D point correspondences between two images

Aim: Reconstruct simultaneously

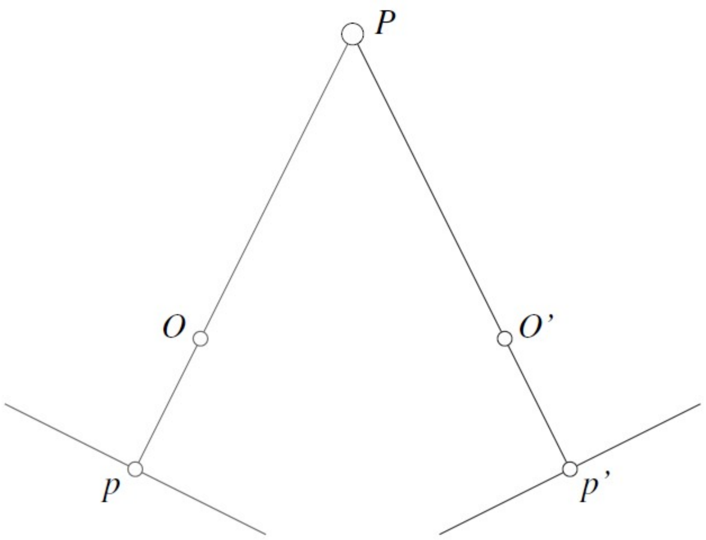
- 3D scene structure,
- camera poses (up to scale) and
- intrinsic parameters

from two different views of the scene

Stereo Vision

Eliminate ambiguity by two cameras → No ambiguity, reconstruction is simple

Main Idea:



Triangulation

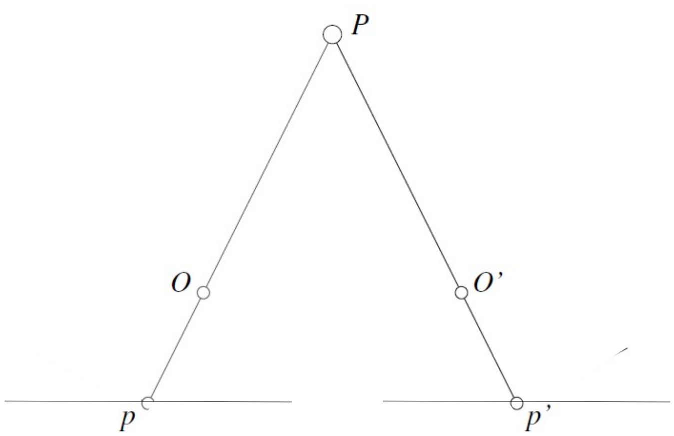
→ Reconstruction by intersection of two rays

Requires:

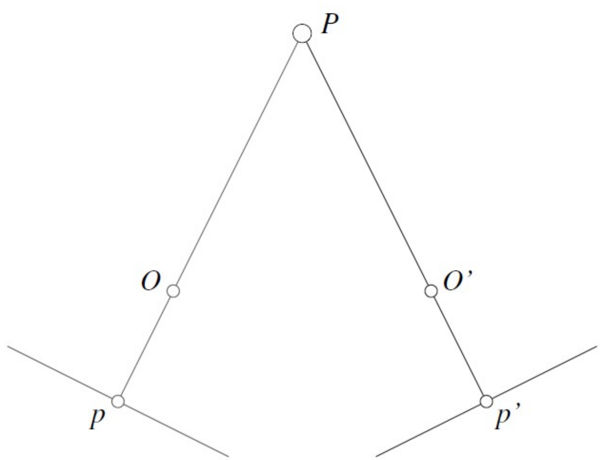
- Calibrated camera system
- Known camera pose
- Point correspondence

Stereo vision

Simplified case: identical cameras are aligned

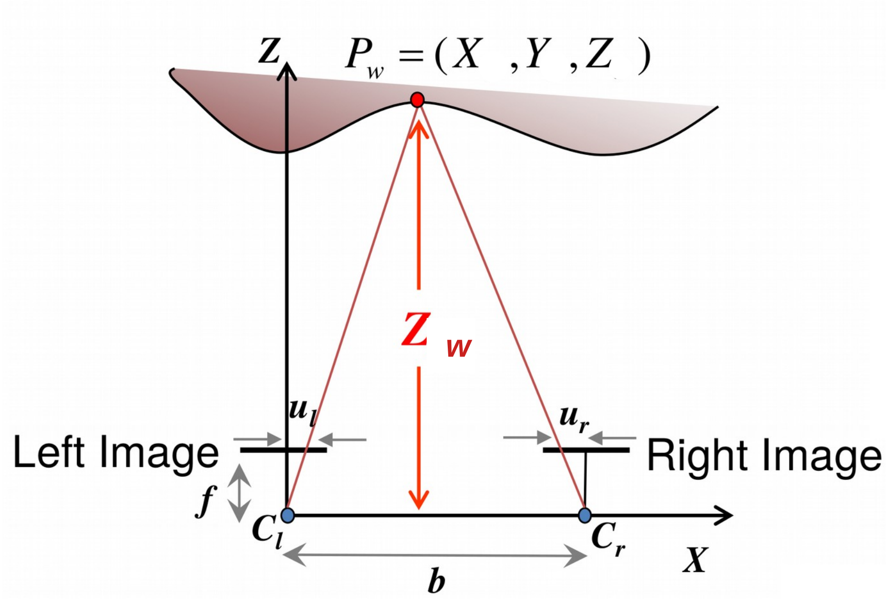


General case: non-identical cameras are not aligned



Stereo Vision - Simplified

Cameras are identical and aligned with the x-axis.



From similar triangles we receive:

$$\frac{f}{Z} = \frac{u_l}{X} \qquad \frac{f}{Z} = \frac{-u_r}{b - X} \qquad \Rightarrow Z_w = \frac{bf}{u_l - u_r}$$

Where b defines the **baseline** (distance between the optical centers of the two cameras) and

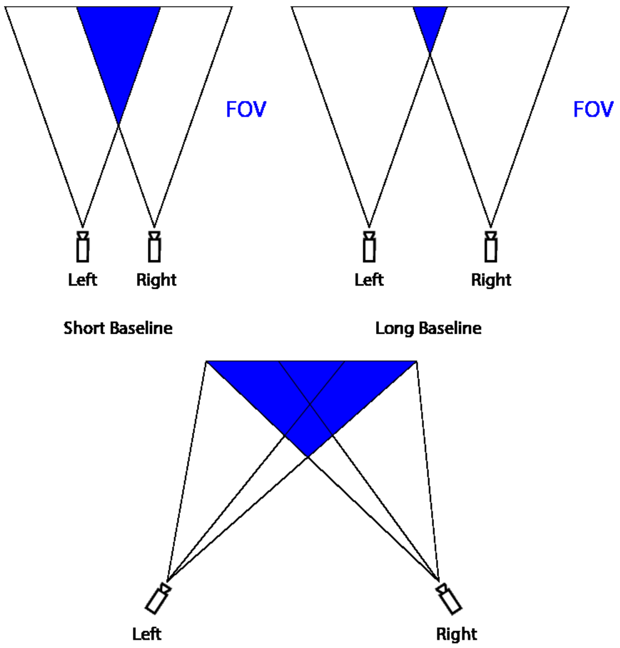
$\|u_l - u_r\|$ defines the **disparity** (difference in image location of the projection of a 3D point on two image planes)

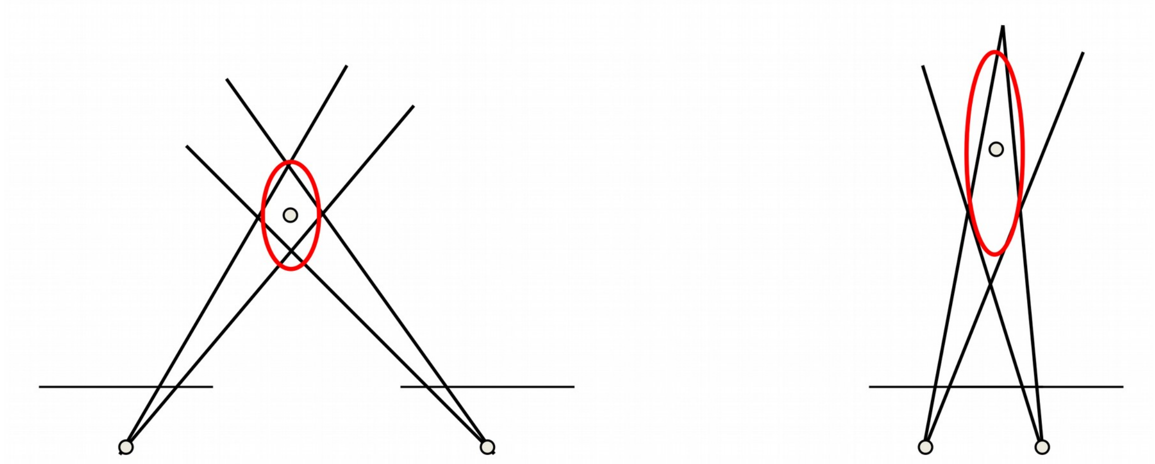
Baseline

Is the baseline

too large:
• Minimum measurable depth increases
• Difficult search problem for close objects

too small:
• Large depth error
• Can you quantify the error as a function of the disparity?





Uncertainty of Reconstruction

Baseline is not the only parameter that determines the 3D point probability distribution:

“A good rule of thumb is that the angle between the rays determines the accuracy of reconstruction. This is a better guide than simply considering the baseline, which is the more commonly used measure.”

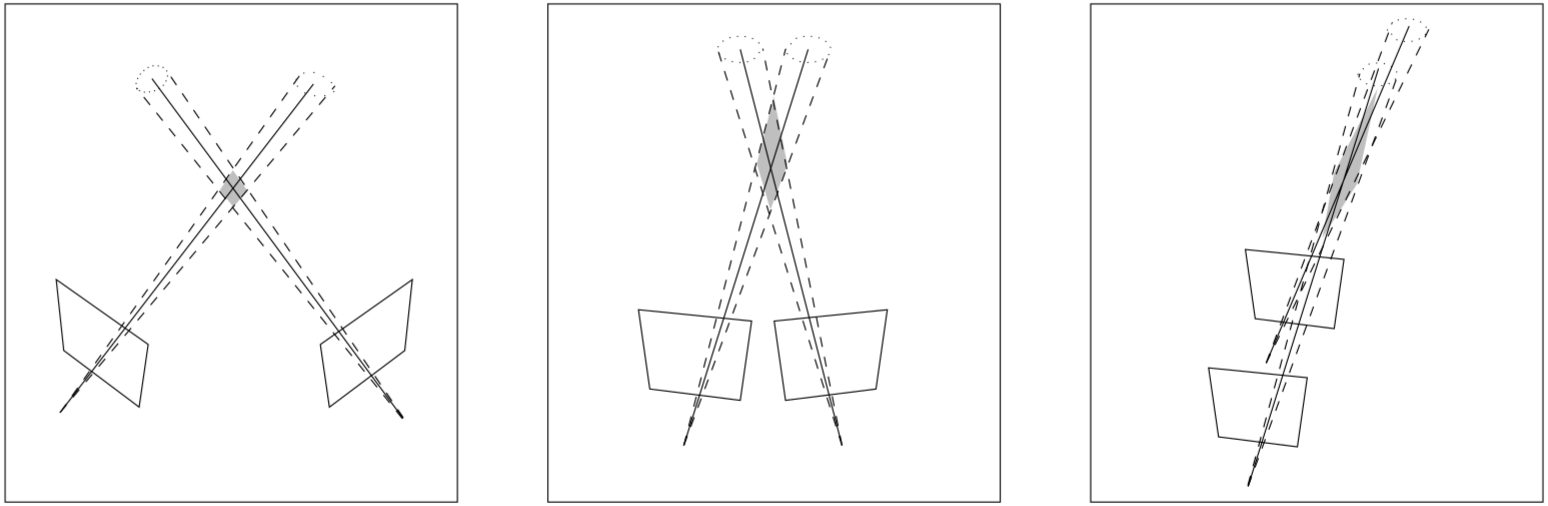


Fig. 12.6. **Uncertainty of reconstruction.** *The shaded region in each case illustrates the shape of the uncertainty region, which depends on the angle between the rays. Points are less precisely localized along the ray as the rays become more parallel. Forward motion in particular can give poor reconstructions since rays are almost parallel for much of the field of view.*

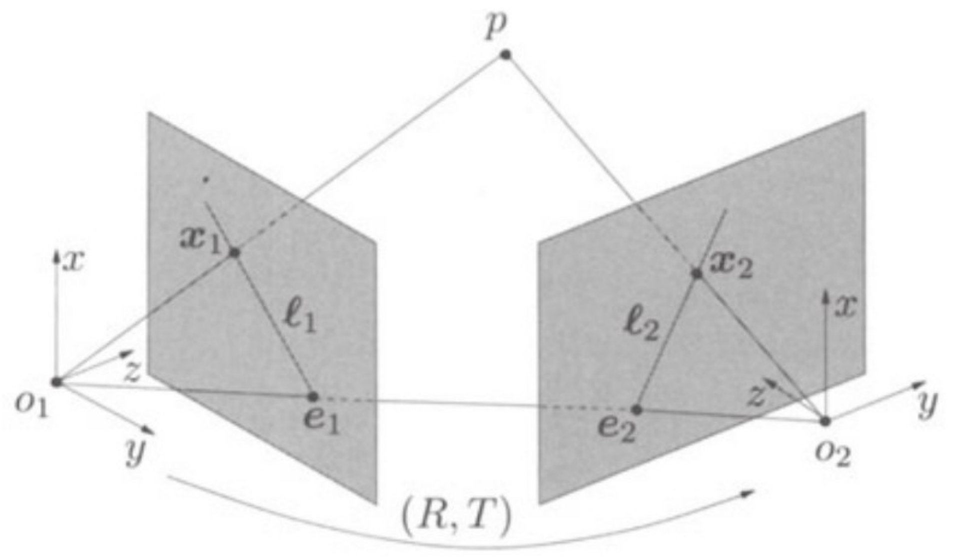
Stereo Vision

In the general, two cameras are not-aligned and non-identical.

In order to be able to use a stereo camera setup, we need to know the

- **Extrinsic parameters** (relative rotation and translation)
- **Instrinsic parameters** (focal length, optical center, radial distortion of each camera)

→ Use a calibration method (Camera calibration from planar grids using homographies, Zhang, see L03)

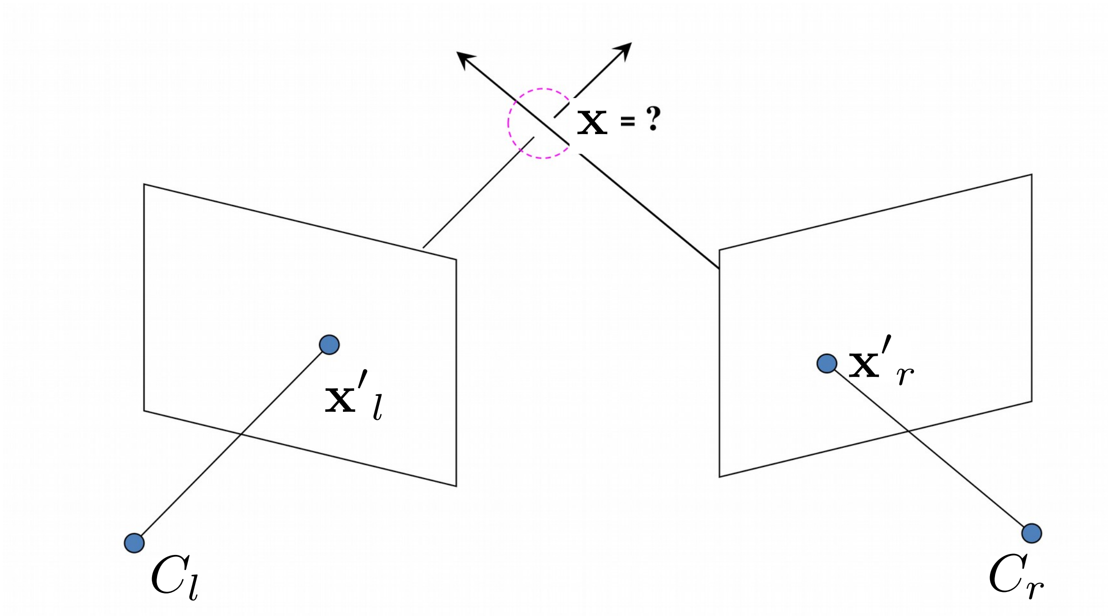


Triangulation

Determine 3D position of a point $\mathbf{x}$ given a set of corresponding image locations and known camera poses.

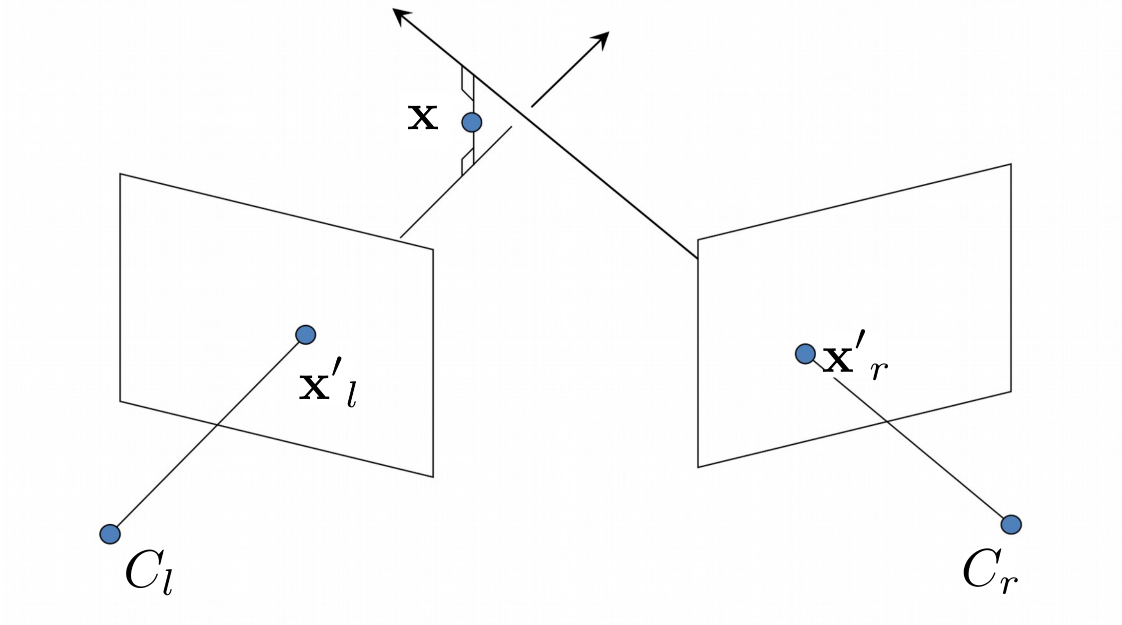
→ intersect two rays corresponding to $\mathbf{x}'_l$ and $\mathbf{x}'_r$

→ only approximation of intersection point $\mathbf{x}$, because of noise



Triangulation: Mid-point Approximation

An intuitive solution for $\mathbf{x}$ would be the mid-point on the shortest line between the two back projected rays



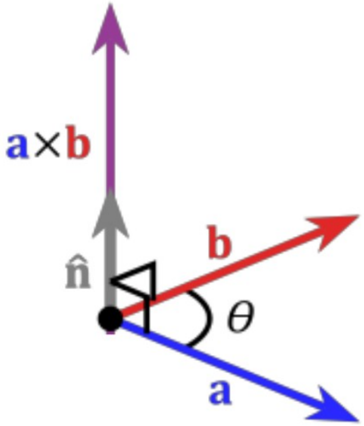
→ minimizes the 3D error

→ small disturbances in the image points $\mathbf{x}'_l$ and $\mathbf{x}'_r$ may lead to an large error in position of $\mathbf{x}$

→ Mid-point approximation is not recommended

Recall: Cross Product (Vector Product)

$$\vec{a} \times \vec{b} = \vec{c}$$



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

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

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

- So here, ***c*** is perpendicular to both ***a*** and ***b***, which means the dot product = 0
- Also, **recall that the cross product of two parallel vectors = 0**

- The vector **cross product** can also be expressed as the product of a **skew-symmetric matrix** and a vector

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

Triangulation – Linear Approximation

Construct the system of equations of the left and right cameras

Left camera (assumed as world frame)

$$\tilde{\mathbf{x}}'_l = \lambda_l \begin{bmatrix} x'_l \\ y'_l \\ 1 \end{bmatrix} = K_l \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\lambda_l \begin{bmatrix} x'_l \\ y'_l \\ 1 \end{bmatrix} = K_l [I|0] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\Rightarrow \tilde{\mathbf{x}}'_l = P_l \cdot \tilde{\mathbf{x}}$$

$$\tilde{\mathbf{x}}'_l \times P_l \cdot \tilde{\mathbf{x}} = 0$$

$$[\tilde{\mathbf{x}}'_{l\times}] P_l \cdot \tilde{\mathbf{x}} = 0$$

$$\begin{bmatrix} y'_l p_l^3 - p_l^2 \\ p_l^1 - x'_l p_l^3 \\ \cancel{x'_l p_l^2 - y'_l p_l^1} \end{bmatrix} \tilde{\mathbf{x}} = 0$$

Right camera

$$\tilde{\mathbf{x}}'_r = \lambda_r \begin{bmatrix} x'_r \\ y'_r \\ 1 \end{bmatrix} = K_r R \begin{bmatrix} x \\ y \\ z \end{bmatrix} + T$$

$$\lambda_r \begin{bmatrix} x'_r \\ y'_r \\ 1 \end{bmatrix} = K_r [R|T] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\Rightarrow \tilde{\mathbf{x}}'_r = P_r \cdot \tilde{\mathbf{x}}$$

$$\tilde{\mathbf{x}}'_r \times P_r \cdot \tilde{\mathbf{x}} = 0$$

$$[\tilde{\mathbf{x}}'_{r\times}] P_r \cdot \tilde{\mathbf{x}} = 0$$

$$\begin{bmatrix} y'_r p_r^3 - p_r^2 \\ p_r^1 - x'_r p_r^3 \\ \cancel{x'_r p_r^2 - y'_r p_r^1} \end{bmatrix} \tilde{\mathbf{x}} = 0$$

$$\begin{bmatrix} y'_l p_l^3 - p_l^2 \\ p_l^1 - x'_l p_l^3 \\ y'_r p_r^3 - p_r^2 \\ p_r^1 - x'_r p_r^3 \end{bmatrix} \tilde{\mathbf{x}} = 0$$

$$A\tilde{\mathbf{x}} = 0$$

Triangulation – Linear Approximation

With

$$\begin{bmatrix} y_l' p_l^3 - p_l^2 \\ p_l^1 - x_l' p_l^3 \\ y_r' p_r^3 - p_r^2 \\ p_r^1 - x_r' p_r^3 \end{bmatrix} \tilde{\mathbf{x}} = 0$$

$$A\tilde{\mathbf{x}} = 0$$

- We receive for each camera system two independent equations each in terms of the three unknown elements of $\mathbf{x}$.
- The resulting overdetermined homogeneous system of linear equations can be solved using the SVD (see L04: DLT) to find the 3D point $\mathbf{x}$ that minimizes the **algebraic error**

$$\epsilon = \|A\tilde{\mathbf{x}}\|$$

- The linear approximation provides a “good” 3D estimate

→ utilize as starting point for an iterative non-linear estimation method

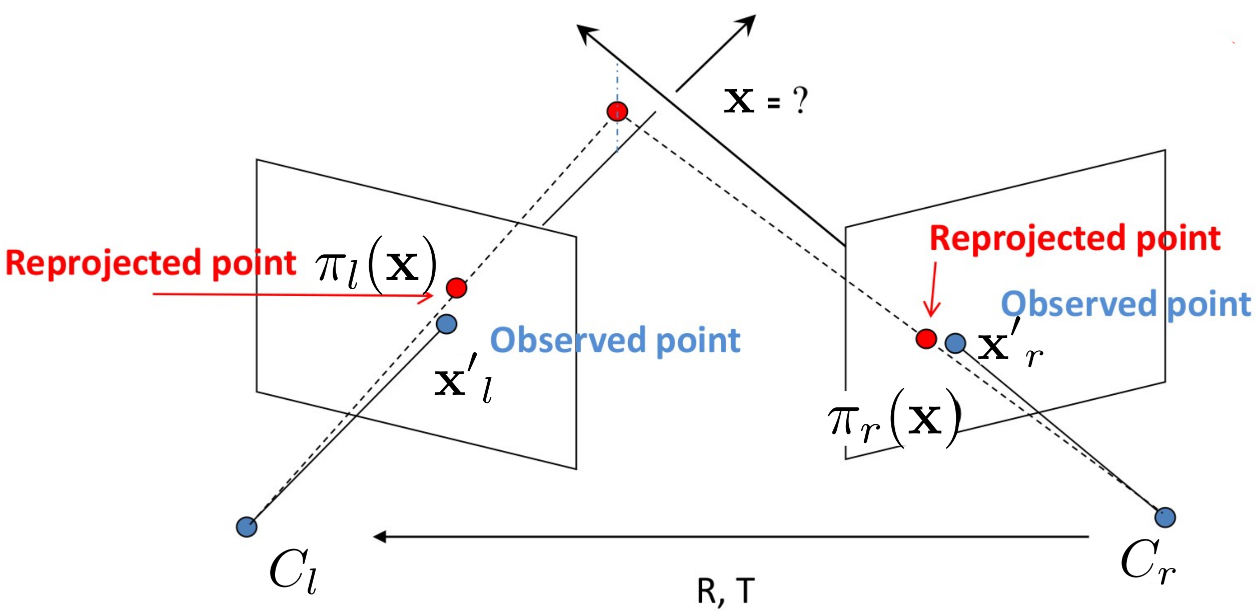
Triangulation: Non-linear Approximation

Find $\mathbf{x}$ that minimizes the sum of squared reprojection (SSR) errors:

$$\varepsilon_{SSR} = \|\mathbf{x}'_l - \pi_l(\mathbf{x})\|^2 + \|\mathbf{x}'_r - \pi_r(\mathbf{x}, R, T)\|^2$$

where $\pi_l(\mathbf{x})$ is the projection of $\mathbf{x}$ onto the left camera plane.

In practice, we initialize $\mathbf{x}$ using the linear approximation and then we refine it by minimizing the SSR error (using Gauss-Newton or Levenberg-Marquardt).



Stereo-View Geometry

Depth from stereo vision - Triangulation

Given: Projection matrices, 2D point correspondence between two images

Aim: Recover the 3D structure from images

Two-View Structure from Motion

Given: n 2D point correspondences between two images

Aim: Reconstruct simultaneously

- 3D scene structure,
- camera poses (up to scale) and
- intrinsic parameters

from two different views of the scene

Structure from Motion (SfM)

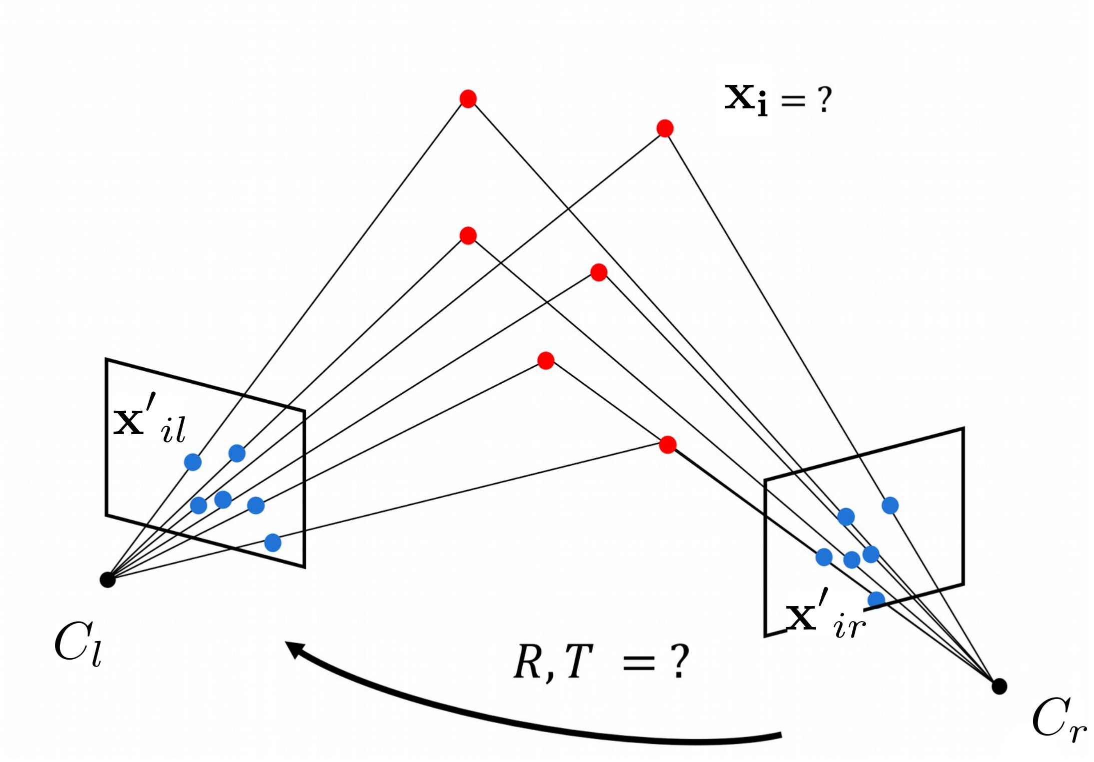
Given: n 2D point correspondences between two images $\mathbf{x'}_{il} = [x'_{il}\ y'_{il}]^\top \leftrightarrow \mathbf{x'}_{ir} = [x'_{ir}\ y'_{ir}]^\top$

Aim: Estimate simultaneously

- 3D points $\mathbf{x_i}$
- Camera relative-motion parameters R, T - (poses up to scale)
- Camera intrinsic parameters $K_l,\ K_r$

$$\lambda_l \begin{bmatrix} x'_{il} \\ y'_{il} \\ 1 \end{bmatrix} = K_l [I|0] \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}$$

$$\lambda_r \begin{bmatrix} x'_{ir} \\ y'_{ir} \\ 1 \end{bmatrix} = K_r [R|T] \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}$$



Structure from Motion

Two cases:

Calibrated case: $K_l,\ K_r$ are known.

Uncalibrated case: $K_l,\ K_r$ are unknown.

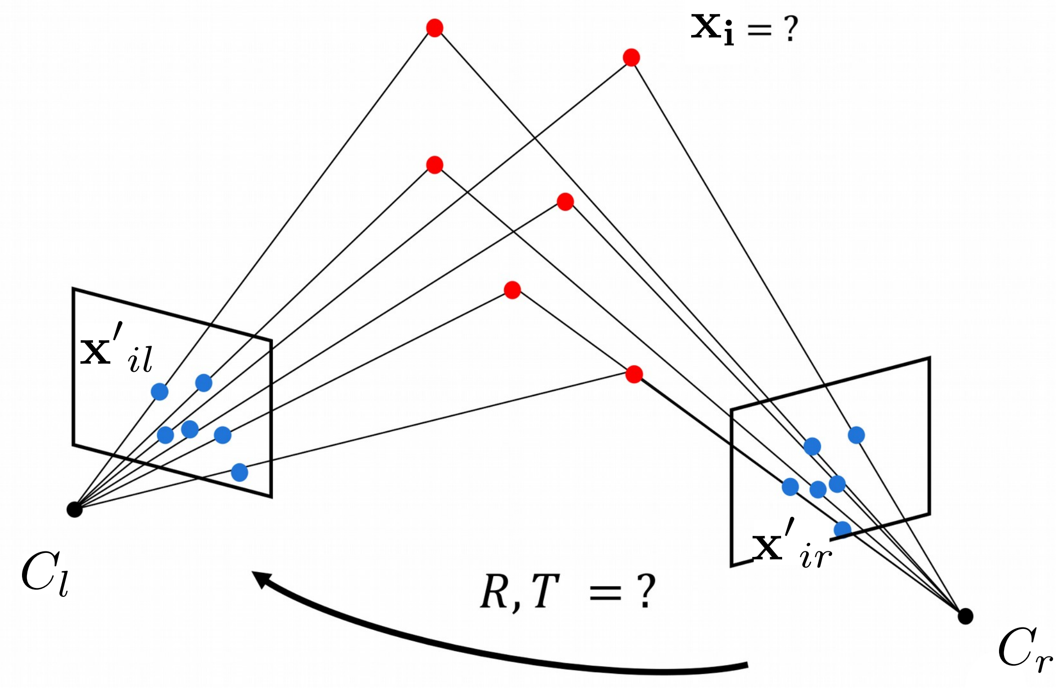
Calibrated Case

- Cameras are calibrated
- Use normalized image coordinates
- Find $R,\ T$ and $\mathbf{x_i}$ that satisfy

$$\begin{bmatrix} x_{il} \\ y_{il} \\ 1 \end{bmatrix} = K^{-1} \begin{bmatrix} x'_{il} \\ y'_{il} \\ 1 \end{bmatrix}$$

$$\lambda_l \begin{bmatrix} x_{il} \\ y_{il} \\ 1 \end{bmatrix} = [I|0] \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}$$

$$\lambda_r \begin{bmatrix} x_{ir} \\ y_{ir} \\ 1 \end{bmatrix} = [R|T] \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}$$



Reconstruction Ambiguity

If we rescale the entire scene and camera views by a constant factor (i.e., similarity transformation), the projections (in pixels) of the scene points in both images remain exactly the same (calibr. case):

Calibrated case

Uncalibrated case

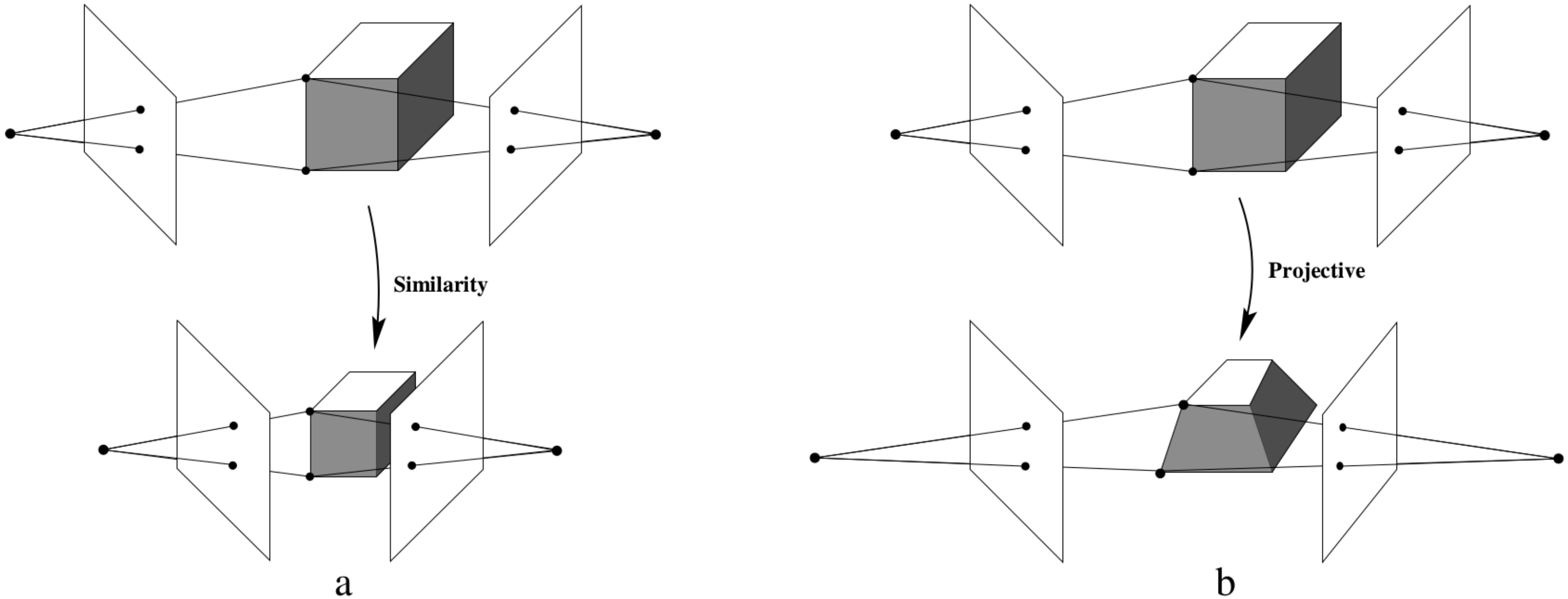


Fig. 10.2. **Reconstruction ambiguity.** *(a) If the cameras are calibrated then any reconstruction must respect the angle between rays measured in the image. A similarity transformation of the structure and camera positions does not change the measured angle. The angle between rays and the baseline (epipoles) is also unchanged. (b) If the cameras are uncalibrated then reconstructions must only respect the image points (the intersection of the rays with the image plane). A projective transformation of the structure and camera positions does not change the measured points, although the angle between rays is altered. The epipoles are also unchanged (intersection with baseline).* (Courtesy: Hartly&Zissermann)

Calibrated Case: Reconstruction Ambiguity

Monocular vision: It is not possible to recover the absolute scale of the scene!

Stereo vision: Only 5 degrees of freedom are measurable:

- 3 parameters to describe the rotation
- 2 parameters for the translation up to a scale (we can only compute the direction of translation but not its length)

→ **Amount of knowns and unknowns:**

- $4n$ knowns: n 2D correspondences between two images
- $5 + 3n$ unknowns: 5 for the motion up to a scale (3 for rotation, 2 for translation), 3n = number of coordinates of the n 3D points

Solution

- If and only if the number of independent equations ≥ number of unknowns: ´$4n \geq 5 + 3n$ ´$n \geq 5$ [Kruppa1913]

Calibrated Case

- Estimate at first R and T
- If R and T are known, estimate the 3D structure (triangulation: non-linear approximation)

→ Solution: Utilize Epipolar Geometry

Epipolar Geometry

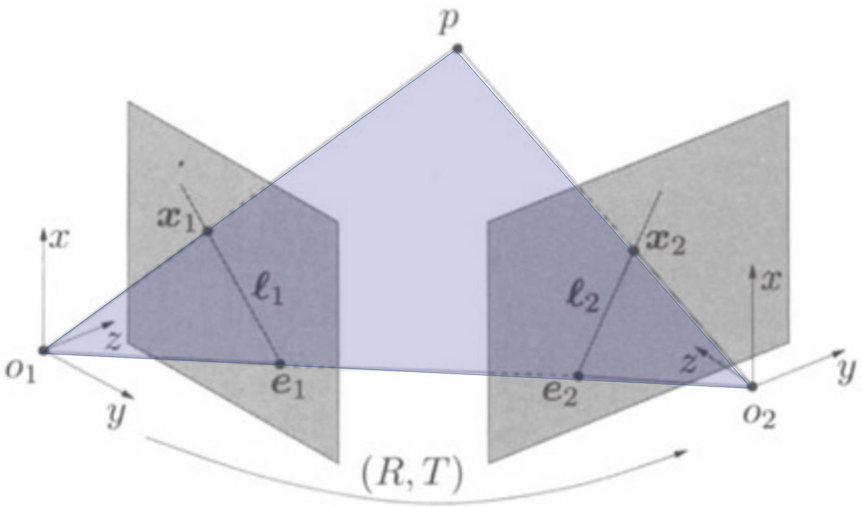
Starting point: Image pair of a 3D scene recorded with two cameras at centers o_1 and o_2

Epipolar geometry describes the geometric relation of two views of a 3D scene.

Lit.: Richard Hartley and Andrew Zisserman. Multiple View Geometry in computer vision. Cambridge University Press, First published 2000.

Epipolar Geometry - Terminology

- Cameras at centers $o_1 \text{ and } o_2$
- p point in 3D
- $x_1 \text{ and } x_2$ are projections of p into the images

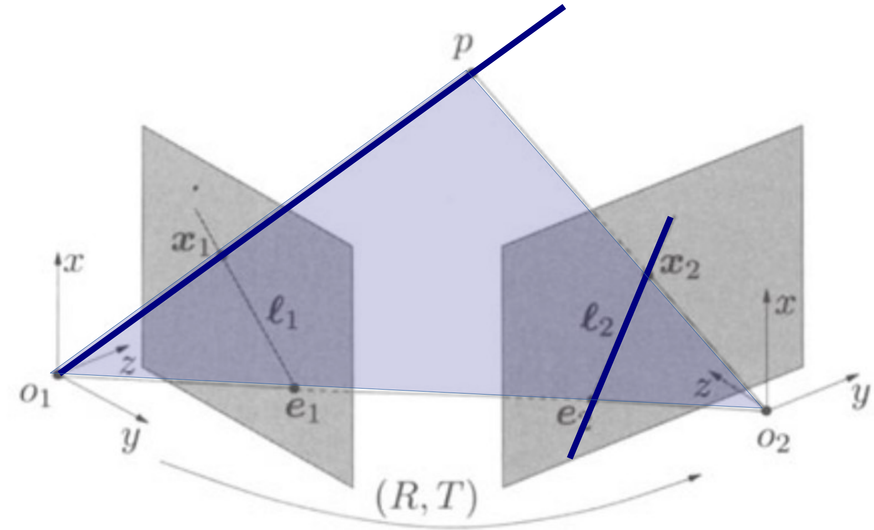


- **Baseline:** Line joining both camera centers
- **Epipoles:** Projection e_1 (e_2) of one camera center onto the image plane of the other camera frame (intersections of the baseline with each image plane)
- **Epipolar-plane:** Plane (o_1, o_2, p) that goes through camera-centers and point p
- Image planes I_1 and I_2
- **Epipolar lines:** Intersection of the epipolar plane of p with one image plane is a line; lines $l_1 \text{ and } l_2$ are the intersection of the plane

Note: Cameras approximated: “Pinhole camera model”

Epipolar Geometry

- All points p are projected to x_1 in the first camera C_1
- In the second camera C_2 the points are projected onto a line l_2



Main implication:

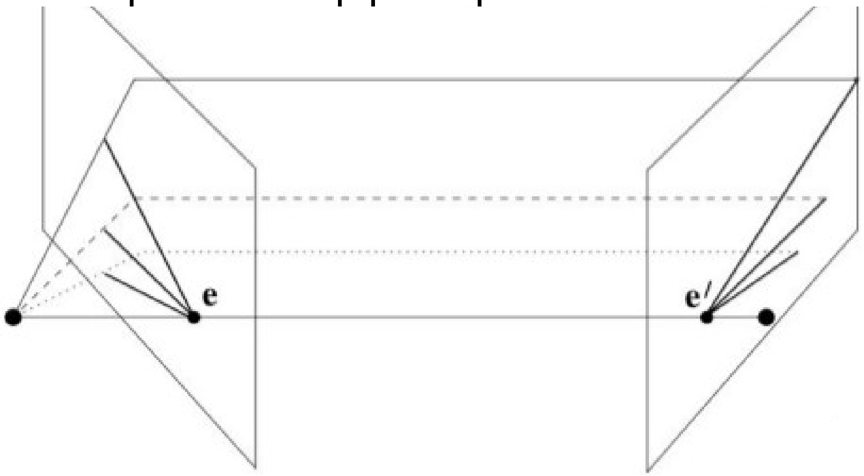
A point x in the first image plane has a matching point on the corresponding **epipolar line** in the second image → no need to look “everywhere”!

Extent idea to both cameras:

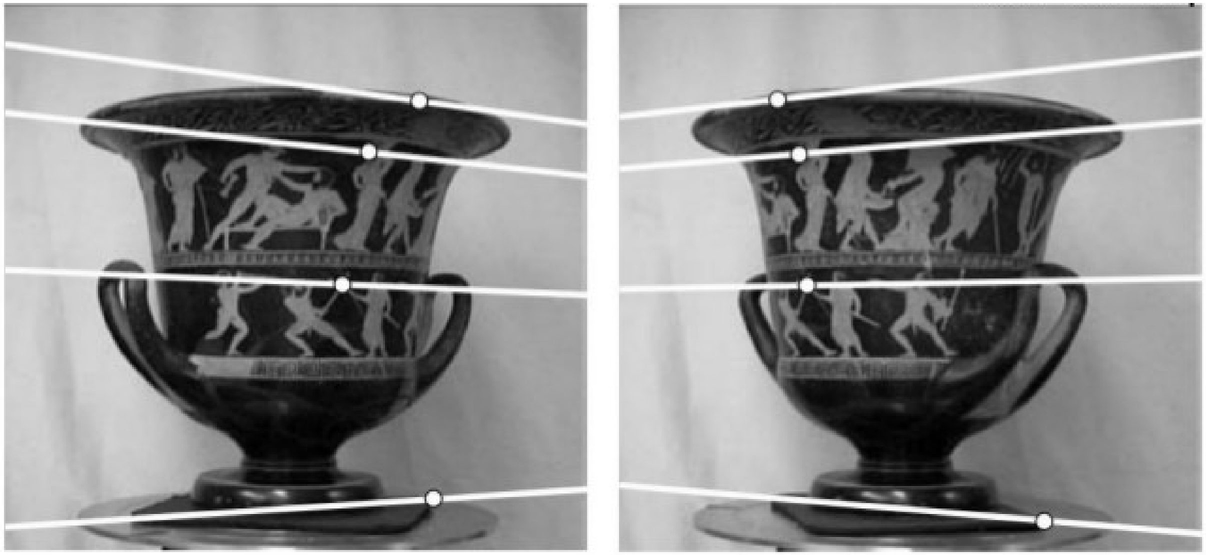
Corresponding points are found only on corresponding epipolar lines that belong to their epipolar plane. → no need to look “everywhere”!

Epipolar Geometry

The Baseline gives rise to a “pencil” of epipolar planes.

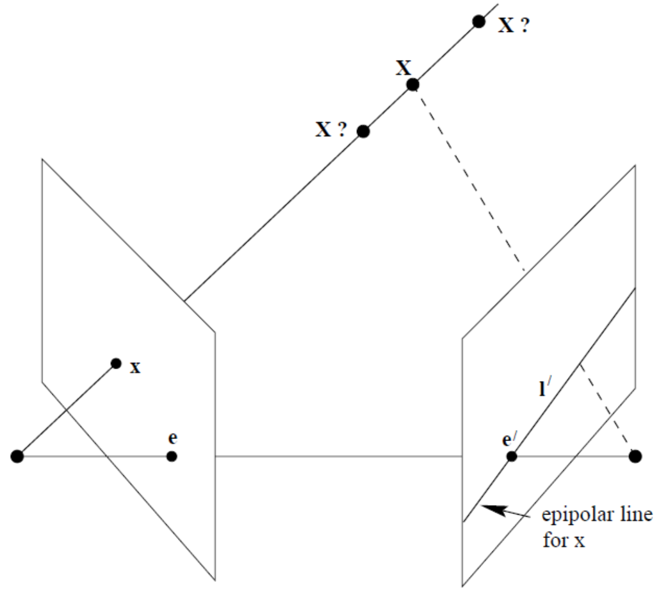


Corresponding points and their epipolar lines.



Epipolar Line

A point in one image “generates” a line in another image (epipolar line)

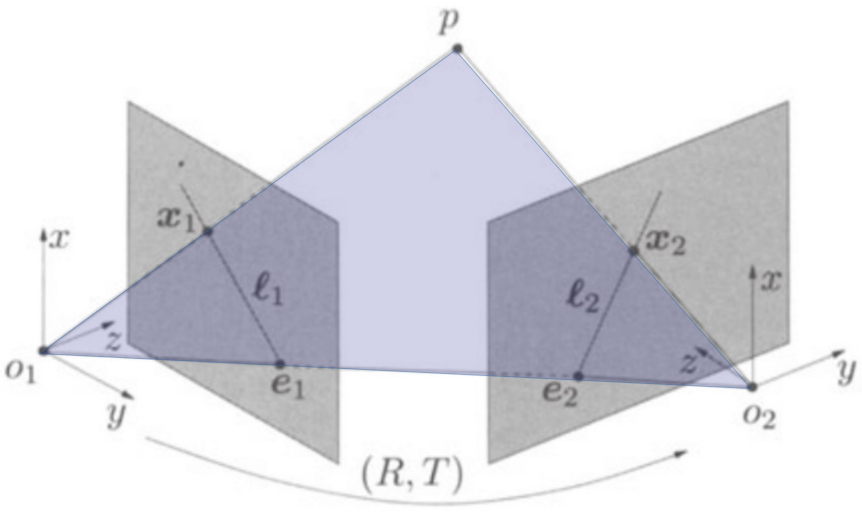


Left line in left camera frame $p_1 = d_1 \hat{x}_1$

Right line in right camera frame $p_2 = d_2 \hat{x}_2$ where $\hat{x}_j = K^{-1} \bar{x}_j$ are the (local) ray directions

Epipolar Constraint

$$X_r = RX_l + T$$
$$\lambda_r \mathbf{x}_r = R\lambda_l \mathbf{x}_l + T$$
$$\lambda_r [T]_\times \mathbf{x}_r = [T]_\times R\lambda_l \mathbf{x}_l + [T]_\times T$$
$$\mathbf{x}_r^\top \lambda_r [T]_\times \mathbf{x}_r = \mathbf{x}_r^\top [T]_\times R\lambda_l \mathbf{x}_l$$
$$0 = \mathbf{x}_r^\top [T]_\times R\mathbf{x}_l$$
$$0 = \mathbf{x}_r^\top E\mathbf{x}_l$$



- The **Epipolar constraint** holds for every pair of corresponding points
- Two images of the same point $\mathbf{p} = \mathbf{x}$ from two camera positions with relative pose $(R,\ T)$, ($R \in SO(3)$ relative orientation, $T \in \mathbb{R}^3$ relative position) satisfy the epipolar constraint equation.
- Where $E = [T]_\times R \in \mathbb{R}^{3\times3}$ is the **essential matrix**
- The epipolar constraint gives the relative pose between two cameras
- The Essential Matrix can be decomposed into R and T recalling that four distinct solutions for R and T are possible. [LonguetHiggins1981]

Compute Essential Matrix

- Kruppa showed in 1913 that 5 image correspondences is the minimal case and that there can be at up to 11 solutions.
- However, in 1988, Demazure showed that there are actually at most 10 distinct solutions.
- In 1996, Philipp proposed an iterative algorithm to find these solutions.
- Only in 2004, the first efficient and non iterative solution was proposed. It uses Groebner basis decomposition [Nister, CVPR'2004].
- The first popular solution uses 8 points and is called the 8-point algorithm or Longuet-Higgins algorithm (1981). Because of its ease of implementation, it is still used today (e.g., NASA rovers).
- 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.

8-Point Algorithm – Essential Matrix

The essential matrix E is defined by

$$\mathbf{x}_r^\top E \mathbf{x}_l = 0 \qquad\qquad E = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix}$$

Each pair of point correspondences $\mathbf{x}_r,\ \mathbf{x}_l$ provides a linear equation, leading for n

Correspondences to the following system of equations $Q\bar{E} = 0$:

$$\begin{bmatrix} x_{1r}x_{1l} & x_{1r}y_{1l} & x_{1r} & y_{1r}x_{1l} & y_{1r}y_{1l} & y_{1r} & x_{1l} & y_{1l} & 1 \\ x_{2r}x_{2l} & x_{2r}y_{2l} & x_{2r} & y_{2r}x_{2l} & y_{2r}y_{2l} & y_{2r} & x_{2l} & y_{2l} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{nr}x_{nl} & x_{nr}y_{nl} & x_{nr} & y_{nr}x_{nl} & y_{nr}y_{nl} & y_{nr} & x_{nl} & y_{nl} & 1 \end{bmatrix} \begin{bmatrix} e_{11} \\ e_{12} \\ e_{13} \\ e_{21} \\ e_{22} \\ e_{23} \\ e_{31} \\ e_{32} \\ e_{33} \end{bmatrix} = 0$$

Where the entries of the matrix Q are known and the entries of the vector $\bar{E}$ are unknown.

Compute Essential Matrix

Minimal solution

- The nx9 matrix Q has rank 8 in order to have a unique (up to a scale) non-trivial solution $\bar{E}$
- Each point correspondence provides 1 independent equation

→ 8 point correspondences are needed

Over-determined solution

- If n > 8
- A possible solution is to minimize $\|Q\bar{E}\|^2$ subject to the constraint $\|\bar{E}\|^2 = 1$

→ The solution is the eigenvector corresponding to the smallest eigenvalue of the matrix $Q^\top Q$ (because it is the unit vector u that minimizes $\|Qu\|^2 = u^\top Q^\top Qu$

→ It can be solved through Singular Value Decomposition (SVD)

Degenerate Configurations

- The solution of the eight-point algorithm is degenerate when the 3D points are coplanar → the five-point algorithm works also for coplanar points

Interpretation

The 8-point algorithm tries to minimize the algebraic error

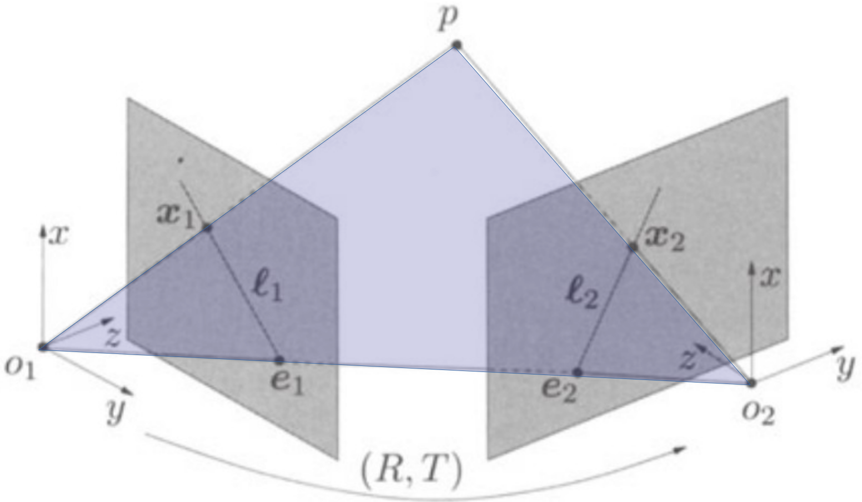
$$\sum_{i=1}^{n} (\mathbf{x}_r^\top E \mathbf{x}_l)^2$$

Using the dot product, it can be observed that

$$\mathbf{x}_r^\top \cdot E\mathbf{x}_l = \|\mathbf{x}_r\| \|E\mathbf{x}_l\| \cos\theta$$

→ product depends on the angle θ between $\mathbf{x}_r$ and the normal $\overrightarrow{n} = E\mathbf{x}_l$ to the epipolar plane.

→ Non-zero if $\mathbf{x}_l, \ \mathbf{x}_r$ and T are not coplanar.



Decompose E

Singular Value Decomposition:

$$E = USV^\top$$

Enforce rank-2: set smallest singular value of S to 0:

$$S = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\hat{T} = U \begin{bmatrix} 0 & \pm 1 & 0 \\ \pm 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} SV^\top$$

$$\hat{T} = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & t_x \\ -t_y & t_x & 0 \end{bmatrix} \Rightarrow \hat{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

$$\hat{R} = U \begin{bmatrix} 0 & \pm 1 & 0 \\ \pm 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} V^\top$$

$$\begin{aligned} \Rightarrow T &= K_r \hat{t} \\ R &= K_r \hat{R} K_l^{-1} \end{aligned}$$

Four possible solutions (calibrated case)

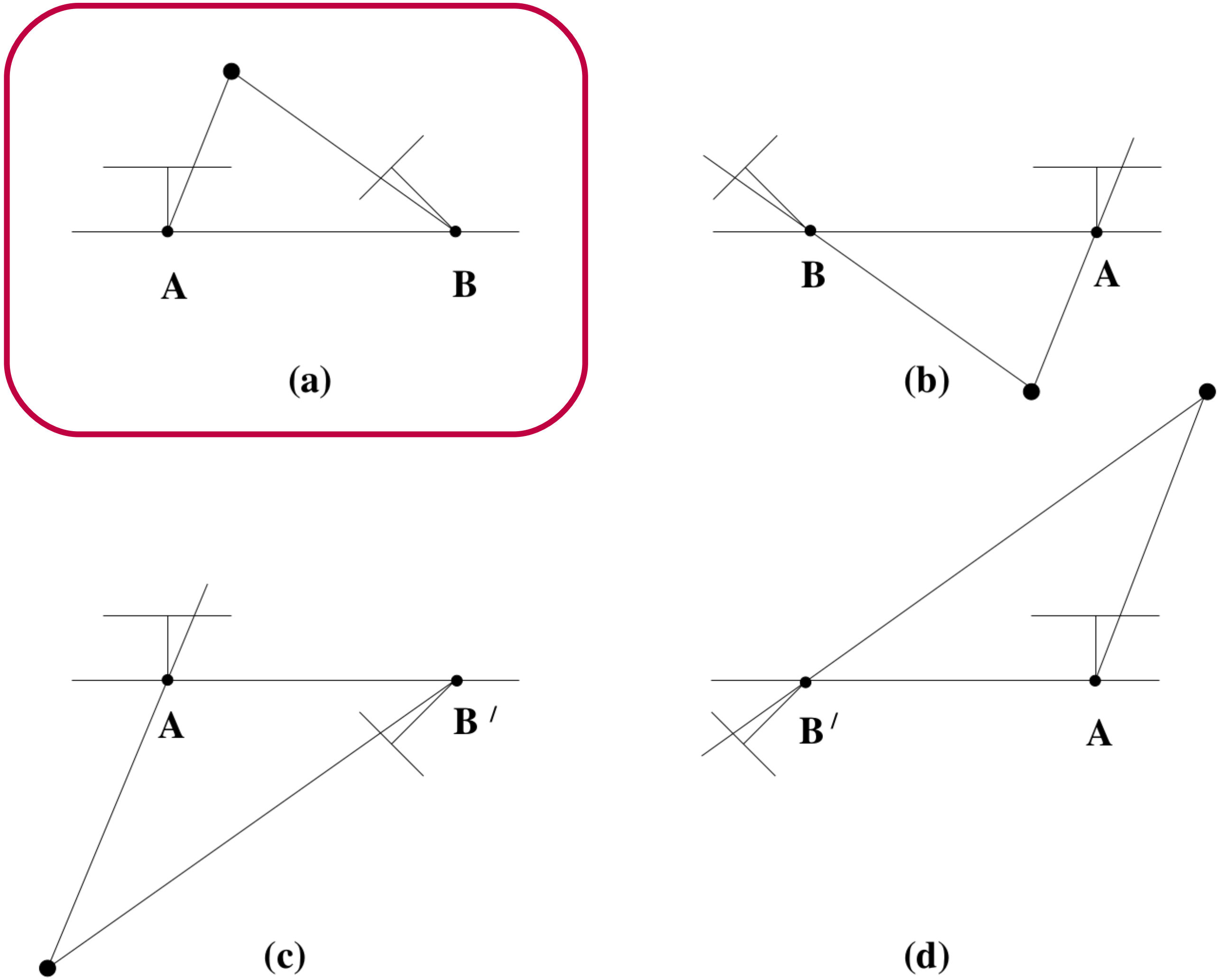


Fig. 9.12. **The four possible solutions for calibrated reconstruction from E.** *Between the left and right sides there is a baseline reversal. Between the top and bottom rows camera B rotates 180° about the baseline. Note, only in (a) is the reconstructed point in front of both cameras.*

(Courtesy: Hartly&Zissermann)

Structure from Motion

Two cases:

Calibrated case: $K_l,\ K_r$ are known.

→ Utilize Essential matrix

$\boxed{\text{Uncalibrated case: } K_l,\ K_r \text{ are unknown.}}$

→ Utilize Fundamental matrix

Fundamental Matrix

We use normalized image coordinates to state the epipolar constraint:

$$\mathbf{x}_r^\top E \mathbf{x}_l = 0$$

$$\mathbf{x}_r^\top [T]_\times R \mathbf{x}_l = 0$$

$$\mathbf{x'}_r^\top K_r^{-T} [T]_\times R K_l^{-1} \mathbf{x'}_l = 0$$

$$\mathbf{x'}_r^\top F \mathbf{x'}_l = 0$$

$$\begin{bmatrix} x_l \\ y_l \\ 1 \end{bmatrix} = K^{-1} \begin{bmatrix} x'_l \\ y'_l \\ 1 \end{bmatrix}$$

Where

$$F = K_r^{-T} [T]_\times R K_l^{-1}$$
$$F = K_r^{-T} E K_l^{-1}$$

Is the so-called Fundamental Matrix

In case K is represented as the unit matrix then $F = E$

8-point Algorithm – Fundamental Matrix

The same 8-point algorithm to compute the essential matrix from a set of normalized image coordinates can also be used to determine the Fundamental matrix:

$$\mathbf{x'}_r^\top F \mathbf{x'}_l = 0$$

→ representation in pixel coordinates!

$$\begin{bmatrix} x'_{1r}x'_{1l} & x'_{1r}y'_{1l} & x'_{1r} & y'_{1r}x'_{1l} & y'_{1r}y'_{1l} & y'_{1r} & x'_{1l} & y'_{1l} & 1 \\ x'_{2r}x'_{2l} & x'_{2r}y'_{2l} & x'_{2r} & y'_{2r}x'_{2l} & y'_{2r}y'_{2l} & y'_{2r} & x'_{2l} & y'_{2l} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_{nr}x'_{nl} & x'_{nr}y'_{nl} & x'_{nr} & y'_{nr}x'_{nl} & y'_{nr}y'_{nl} & y'_{nr} & x'_{nl} & y'_{nl} & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

8-point Algorithm – Fundamental Matrix

Problem: Orders of magnitude difference between column of data matrix

→ least-squares yields poor results

250906.36	183269.57	921.81	200931.10	146766.13	738.21	272.19	198.81	1.00
2692.28	131633.03	176.27	6196.73	302975.59	405.71	15.27	746.79	1.00
416374.23	871684.30	935.47	408110.89	854384.92	916.90	445.10	931.81	1.00
191183.60	171759.40	410.27	416435.62	374125.90	893.65	465.99	418.65	1.00
48988.86	30401.76	57.89	298604.57	185309.58	352.87	846.22	525.15	1.00
164786.04	546559.67	813.17	1998.37	6628.15	9.86	202.65	672.14	1.00
116407.01	2727.75	138.89	169941.27	3982.21	202.77	838.12	19.64	1.00
135384.58	75411.13	198.72	411350.03	229127.78	603.79	681.28	379.48	1.00

$$\begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

- Poor numerical conditioning → results are very sensitive to noise
- Solution: Normalized 8-point algorithm [Hartley, PAMI'97]

Normalized 8-point Algorithm

Objective

Given $n \geq 8$ image point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$, determine the fundamental matrix $\mathtt{F}$ such that $\mathbf{x}'^{\mathsf{T}}_i \mathtt{F} \mathbf{x}_i = 0$.

Algorithm

(i) **Normalization:** Transform the image coordinates according to $\hat{\mathbf{x}}_i = \mathtt{T}\mathbf{x}_i$ and $\hat{\mathbf{x}}'_i = \mathtt{T}'\mathbf{x}'_i$, where $\mathtt{T}$ and $\mathtt{T}'$ are normalizing transformations consisting of a translation and scaling.

(ii) Find the fundamental matrix $\hat{\mathtt{F}}'$ corresponding to the matches $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$ by

(a) **Linear solution:** Determine $\hat{\mathtt{F}}$ from the singular vector corresponding to the smallest singular value of $\hat{\mathtt{A}}$, where $\hat{\mathtt{A}}$ is composed from the matches $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$ as defined in (11.3).

(b) **Constraint enforcement:** Replace $\hat{\mathtt{F}}$ by $\hat{\mathtt{F}}'$ such that $\det \hat{\mathtt{F}}' = 0$ using the SVD (see section 11.1.1).

(iii) **Denormalization:** Set $\mathtt{F} = \mathtt{T}'^{\mathsf{T}} \hat{\mathtt{F}}' \mathtt{T}$. Matrix $\mathtt{F}$ is the fundamental matrix corresponding to the original data $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$.

Algorithm 11.1. *The normalized 8-point algorithm for* $\mathtt{F}$.

Normalized 8-point Algorithm

Estimate the Fundamental matrix on a set of Normalized correspondences (with better numerical properties) and then unnormalizes the result to obtain the fundamental matrix for the given (unnormalized) correspondences

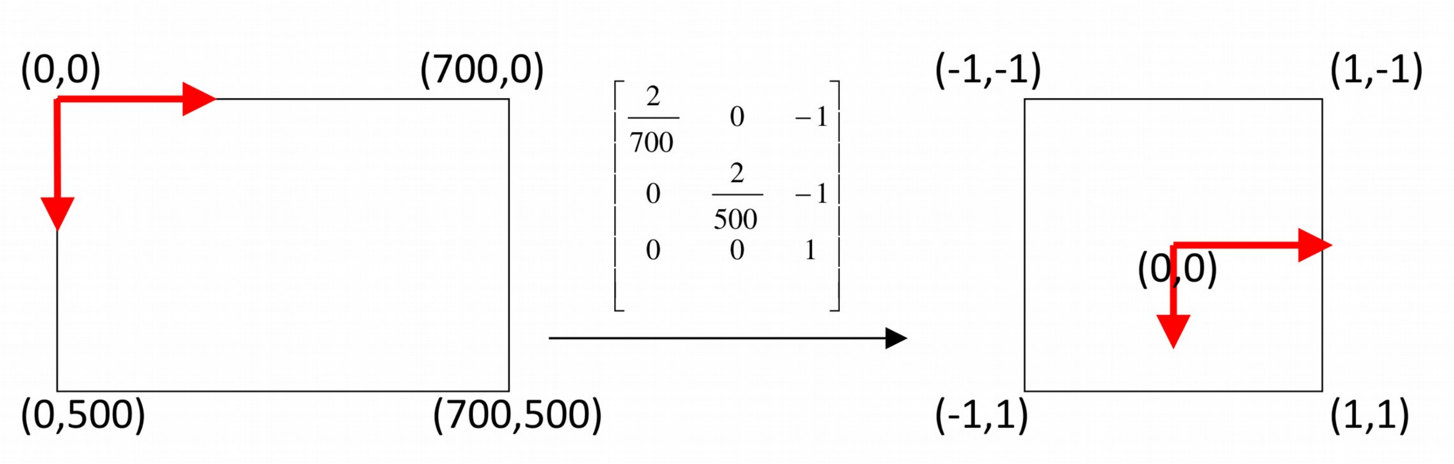
[Hartley1997] In defense of the eight-point algorithm, PAMI

Idea:

Transform image coordinates so that they are in the range $[-1,1] \times [-1,1]$

Example:

Apply the rescaling and shift



Normalized 8-point Algorithm

- Rescale the two point sets such that the centroid of each set is 0 and the mean standard deviation 2, so that the “average” point is equal to [0, 0, 1]T (in homogeneous coordinates).
- For every point compute

$$\hat{\mathbf{x}}_i = \frac{\sqrt{2}}{\sigma}(\mathbf{x_i} - \mu)$$

Where $\mu = (\mu_x, \mu_y) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$ is the centroid and $\sigma = \frac{1}{n}\sum_{1=1}^{n} \|\mathbf{x}_i - \mu\|^2$

is the mean standard deviation of the point set.

This transformation can be expressed in matrix form using homogeneous coordinates:

$$\hat{\mathbf{x}}_i = \begin{bmatrix} \frac{\sqrt{2}}{\sigma} & 0 & -\frac{\sqrt{2}}{\sigma}\mu_x \\ 0 & \frac{\sqrt{2}}{\sigma} & -\frac{\sqrt{2}}{\sigma}\mu_y \\ 0 & 0 & 1 \end{bmatrix} \mathbf{x}_i$$

[Hartley1997] In defense of the eight-point algorithm, PAMI

Normalized 8-point Algorithm

The Normalized 8-point algorithm can be summarized in three steps:

1. Normalize point correspondences $\hat{\mathbf{x}}_l = B_l \mathbf{x}_l, \ \hat{\mathbf{x}}_r = B_r \mathbf{x}_r$
2. Estimate normalized $\hat{F}$ with 8-point algorithm using normalized coordinates $\hat{\mathbf{x}}_l, \ \hat{\mathbf{x}}_r$
3. Compute unnormalized F from $\hat{F}$: $B_r^\top \hat{F} B_l$

$$\hat{\mathbf{x}}_r^\top \hat{F} \hat{\mathbf{x}}_l = 0$$

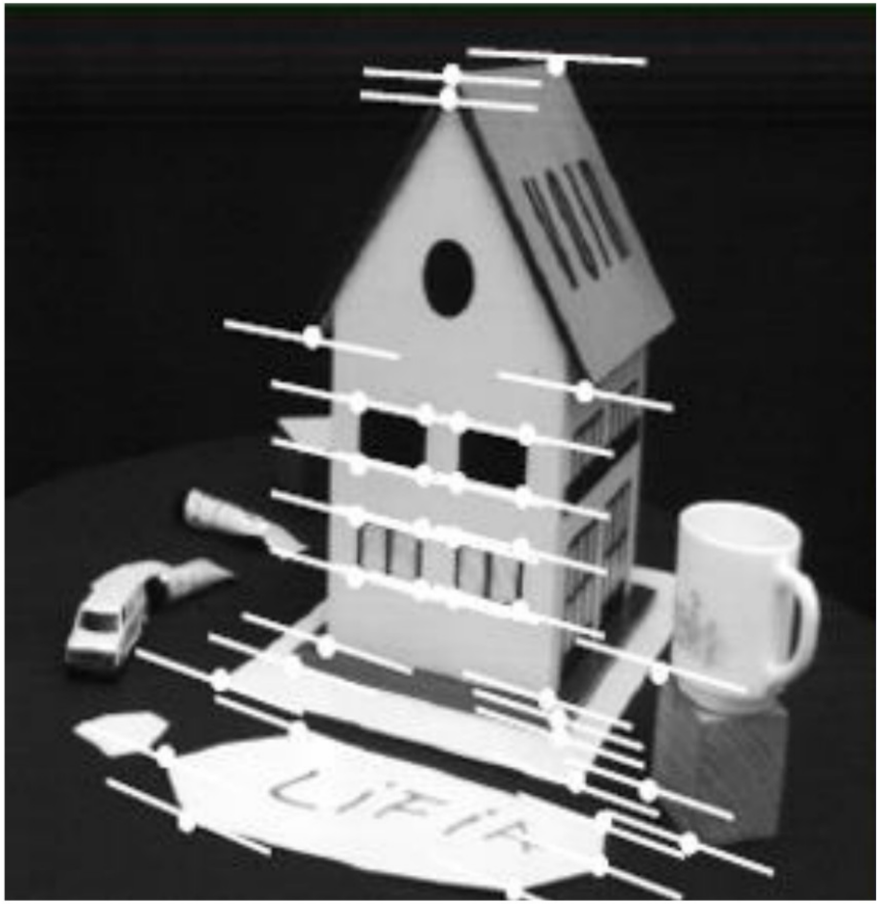
$$\mathbf{x}_r^\top B_r^\top \hat{F} B_l \mathbf{x}_l = 0$$

$$\Rightarrow F = B_r^\top \hat{F} B_l$$

Can $R, \ T, \ K_l$ and K_r extracted from F ?

- In general no: infinite solutions exist
- However, if the coordinates of the principal points of each camera are known and the two cameras have the same focal length f in pixels, then $R, \ T, \ f$ can determined uniquely.

Normalized 8-point Algorithm



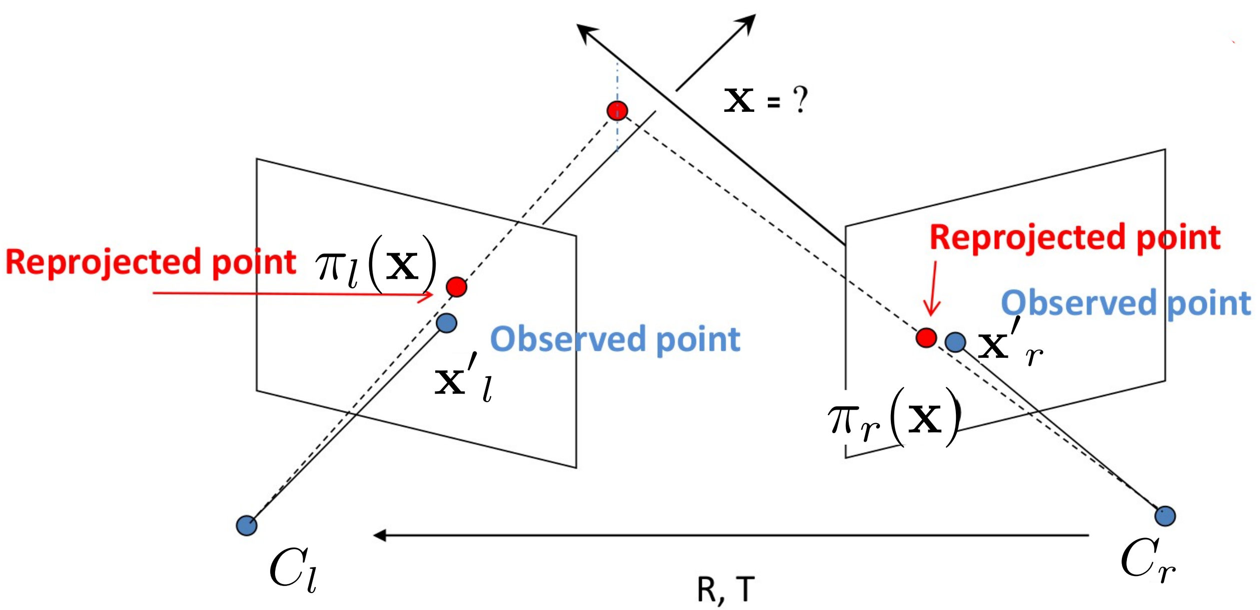
	8-point	Normalized 8-point	Nonlinear refinement
Avg. Ep. Line Distance l	2.33 px	0.92 px	0.86 px

Error Measures for the Essential Matrix

The quality of the estimated Essential matrix can be measured using different error metrics.

- Algebraic error
- Directional Error
- Epipolar Line Distance
- Reprojection Error $\varepsilon = \sum_{i=1}^{n} \|\mathbf{x}'_{il} - \pi_l(\mathbf{x}_i)\|^2 + \|\mathbf{x}'_{ir} - \pi_r(\mathbf{x}_i, R, T)\|^2$

  (most accurate one)



Robust Structure from Motion

In order to implement RANSAC for Structure From Motion (SFM), we need three key ingredients:

1. model in SFM

2. minimum number of points to estimate the model

3. compute the distance of a point from the model- define a distance metric that measures how well a point fits the model

What's the model in SFM?

ad1.: Essential Matrix (for calibrated cameras), Fundamental Matrix (for uncalibrated cameras), (alternatively R and T)

ad2.: 5 points (theoretical minimum number of points); 8-point algorithm: 8 is the minimum

ad3.: distance of a point from the model: algebraic error; directional error, Epipolar line distance or reprojection error

RANSAC (→ L04) for SfM:

1. Randomly select 8 point correspondences
2. Fit the model to all other points and count the inliers
3. Repeat from 1 for k times

Typical Stereo Vision Systems

Stereo vision system can be composed into four main processing steps:

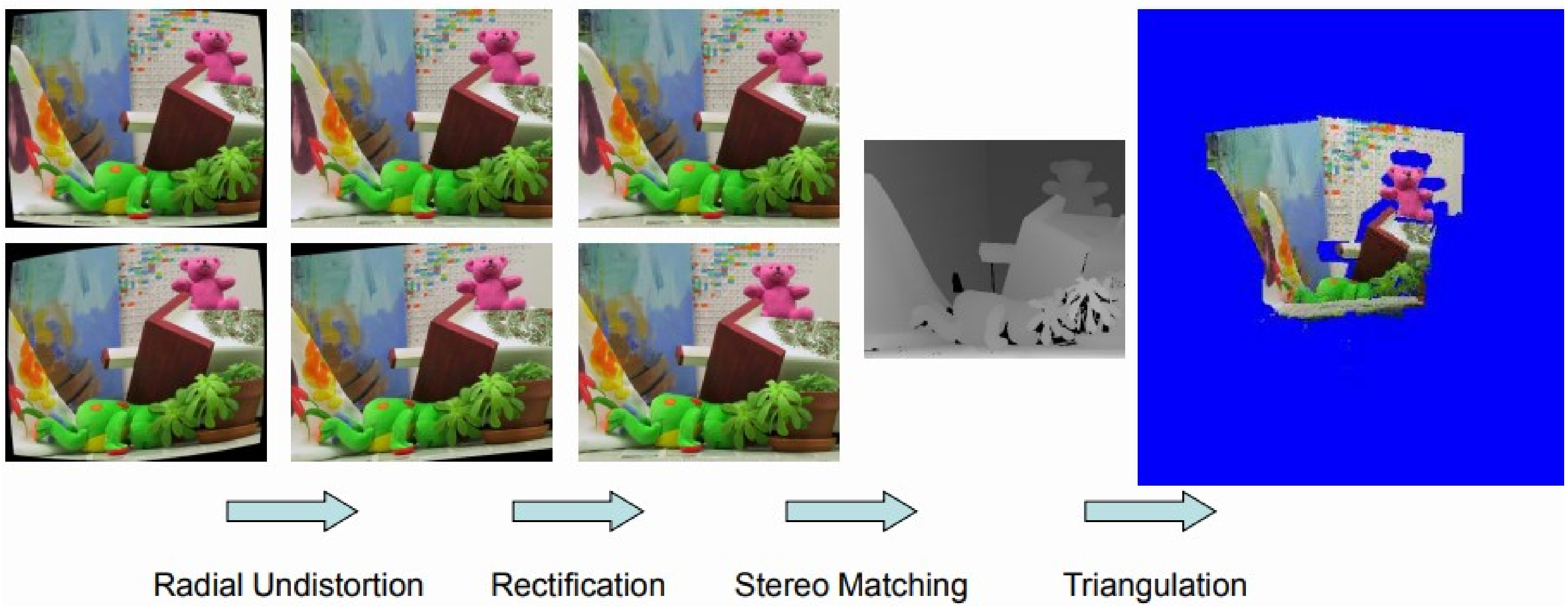


Image courtesy: Christian Unger, TUM

1. Single Camera Calibration (left and right camera): Radial Undistortion
2. Image Rectification: Minimize Image Distortion
3. Stereo Matching: Compute Disparity
4. Triangulation: 3D reconstruction

Literature

HZ Ch. 4.4.4: Normalizing transformations

HZ Ch. 9.1: Epipolar geometry

HZ Ch. 9.2: The fundamental matrix

HZ Ch. 9.6: The essential matrix

HZ Ch. 10.2: Reconstruction ambiguity

HZ Ch. 11.1: Basic equations

HZ Ch. 11.2: The normalized 8-point algorithm

HZ Ch. 12.2: Linear triangulation methods

Additional Reading

[Kruppa 1918] E. Kruppa. Zur Ermittlung eines Objektes aus zwei Perspektiven mit Innerer Orientierung, Sitz.-Ber. Akad. Wiss., Wien, Math. Naturw. Kl., Abt. IIa., 1913. – English Translation plus original paper by Guillermo Gallego, Arxiv, 2017

[LonguetHiggis1981] H. Christopher Longuet-Higgins, A computer algorithm for reconstructing a scene from two projections, Nature, 1981, PDF.

[Niester2004] D. Nister, An Efficient Solution to the Five-Point Relative Pose Problem, PAMI, 2004, PDF

[Hartley1997] Hartley, In defense of the eight-point algorithm, PAMI'97