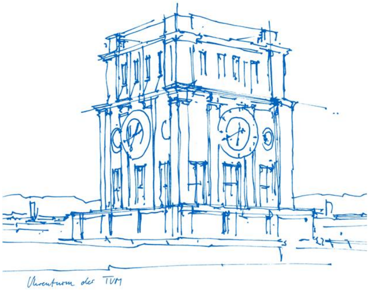
Computer Vision II: Multiple View Geometry (IN2228)

Chapter 06 2D-2D Geometry
(Part 2 Camera Pose Estimation)

Dr. Haoang Li

07 June 2023 12:00-13:30



Announcements before Class

➢ Supplementary Exercise Codes

✓ For some knowledge (e.g., computing normalized image coordinates and computing the relative camera pose from the absolute camera pose), our exercise sessions do not involve them due to time and space limit.

✓ As compensation, we are preparing some MATLAB codes to help you review some important knowledge that is not covered by our exercise sessions but is still highlighted in our review document (for Chapters 01--05).

✓ **Please note that it is optional for you to check these codes.**

✓ For details, please refer to https://www.moodle.tum.de/mod/forum/discuss.php?d=431822

Announcements before Class

➢ Update about Exercise Session Schedule

- ✓ For 2D-2D Geometry, we originally intend to introduce basic configuration (two views) and a more complex configuration (multiple views).
- ✓ Due to time limit, we skip the case of **multiple views** (last year, Prof. Cremers also skipped this part).
- ✓ Our lecture schedule remains unchanged. However, **we cancel the Exercise 7**.

Wed 31.05.2023 Chapter 05: Correspondence Estimation (Part 3)
Thu 01.06.2023 Chapter 06: 2D-2D Geometry (Part 1)

Wed 07.06.2023 Chapter 06: 2D-2D Geometry (Part 2)
Thu 08.06.2023 No lecture (Public Holiday)

Wed 14.06.2023 Chapter 06: 2D-2D Geometry (Part 3)
Thu 15.06.2023 Chapter 07: 3D-2D Geometry (Part 1)

Wed 21.06.2023 Chapter 07: 3D-2D Geometry (Part 2)
Thu 22.06.2023 Chapter 08: 3D-3D Geometry

Wed 28.06.2023 Chapter 09: Single-view Geometry (Part 1)
Thu 29.06.2023 Chapter 09: Single-view Geometry (Part 2)

Core part

Wed 14.06.2023 Exercise 6: Reconstruction from two views

Wed 21.06.2023 Exercise 7: Reconstruction from multiple views

Wed 05.07.2023 Exercise 8: Direct Image Alignment

Wed 12.07.2023 Exercise 9: Direct Image Alignment

Announcements before Class

- Notations and Formulas

✓ Problems

There are some inconsistent symbols in slides (e.g., intrinsic parameters).
Some equations and formulas are in the image format.

✓ Reasons

Some equations are copied from different academic papers.

✓ Solutions

We are adjusting notations and formulas. Due to limited time, the progress is relatively slow. If some of them affect your understanding, please let me know (via email or Moodle) and I will prioritize their update.

Announcements before Class

➢ Correction for Rank Analysis (Tsai’s Method)

- ✓ Q $(2n \times 12)$ should have rank 11 to have **a unique (up-to-scale)** non-zero solution of vector M. **If rank equals 12, we only have zero solution.**
- ✓ Dimension of null space is 1. Vector M can be expressed by **a single basis vector** multiplied by an arbitrary scalar.

$$\mathrm{Q} \cdot \mathrm{M} = 0$$

2n*12 matrix
(known)

12-dimensional vector
(unknown)

- ✓ Please refer to the updated page 14/48 of Chapter 04.

Explanation for Tsai's Method

- QR Decomposition

✓ Assume that we have computed M matrix based on DLT. We aim to recover intrinsic matrix K, rotation matrix R, and translation vector t.

known $\boxed{\mathrm{M}} = \boxed{\mathrm{K}(\mathrm{R} \mid t)}$

✓ First step: re-write the right-hand side based on distributive law

$$M = K\,[R|t] = [\,KR|Kt\,]$$

✓ Second step: use RQ decomposition to decompose the known KR to obtain K and R. Note: QR decomposition is a generic term that may refer to QR, QL, RQ, and LQ decompositions, with L being a lower triangular matrix.
(https://en.wikipedia.org/wiki/QR_decomposition#Relation_to_RQ_decomposition)

✓ Third step: compute *t* based on the known Kt and the estimated *K*.

Today’s Outline

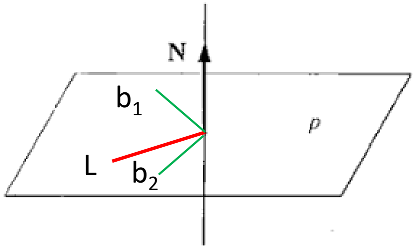
- Five-point Method
- Pose Recovery from Essential matrix
- 1D Correspondence Search

Five-point Method

➢ Review on Rank and Null Space

✓ An example

Computing a 3D direction L orthogonal to normal N



N: 1*3

Num(unknowns) = Rank(N) + Dim(null space)

i.e., dimension of L

3 1 2

$\langle N, L \rangle = 0$ | $N^T L = 0$ | $\{b_1, b_2\}$
Dot product | A linear system | Basis of null space

Five-point Method

➢ Two Properties of Essential Matrix

✓ Recap on definition of essential matrix

$$E = [\boldsymbol{t}]^{\wedge} R$$

✓ Properties of essential matrix

$$det([t]^{\wedge}) = 0 \quad Rank([t]^{\wedge}) = 2$$

$$det(E) = 0$$

Scalar

$$\det\left(\mathbf{A}_1 \mathbf{A}_2 \cdots \mathbf{A}_n\right) = \det\left(\mathbf{A}_1\right) \det\left(\mathbf{A}_2\right) \cdots \det\left(\mathbf{A}_n\right)$$

$$EE^T E - \frac{1}{2} trace(EE^T)E = 0$$

3*3 zero matrix

Properties will not be asked in the exam

Five-point Method

➢ Revisiting Eight-point Method

✓ Linear System

$$\begin{bmatrix} u_2^1u_1^1 & u_2^1v_1^1 & u_2^1 & v_2^1u_1^1 & v_2^1v_1^1 & v_2^1 & u_1^1 & v_1^1 & 1 \\ u_2^2u_1^2 & u_2^2v_1^2 & u_2^2 & v_2^2u_1^2 & v_2^2v_1^2 & v_2^2 & u_1^2 & v_1^2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_2^nu_1^n & u_2^nv_1^n & u_2^n & v_2^nu_1^n & v_2^nv_1^n & v_2^n & u_1^n & v_1^n & 1 \end{bmatrix} \begin{bmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \\ e_8 \end{bmatrix} = 0$$

$$\underset{N*9}{Q}\overset{9*1}{e} = 0$$

- Each correspondence can provide one constraint.
- The minimal case of Q is **8***9 if we neglect the inherent constraints of elements of e.
- The minimal case is **5** correspondences if we consider these constraints. (we no longer solve a linear system)

✓ Definition of E matrix

$$\boxed{E = [T_\times]R \quad \textit{essential matrix}}$$

- R has three degrees of freedom
- T has two degrees of freedom
- **E has five degrees of freedom**
- Rank of Q equals five

$$\underset{9}{\text{Dim(e)}} = \underset{5}{\overset{5*9}{\text{Rank(Q)}}} + \underset{4}{\text{Dim(null space)}}$$

Five-point Method

$$\boxed{Qe = 0}$$

➤ Polynomial Generation

✓ Basis of null space

X, Y, Z, W Dim(null space) = 4

Known 9D basis vectors computed based on the known 5*9 coefficient matrix

$$E = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix}$$

✓ Linear expression of vector **e**

$$e = xX + yY + zZ + wW$$

x, y, z are unknown coefficients w= 1

✓ Constraints of E matrix

$$det(E) = 0 \qquad EE^T E - \frac{1}{2} trace(EE^T)E = 0$$

1 constraint 9 constraints (only two of them are linear independent)

⇨ A polynomial system with respect to unknown x, y, and z

Five-point Method

- ➢ Polynomial Generation

✓ Polynomial System

Unknown vector with respect to x, y, z

$$[x^3, y^3, x^2y, xy^2, x^2z, x^2, y^2z, y^2, xyz, xy, \\ xz^2, xz, x, yz^2, yz, y, z^3, z^2, z, 1].$$

A p = 0

Known coefficient matrix

A	x^3	y^3	x^2y	xy^2	x^2z	x^2	y^2z	y^2	xyz	xy	x	y	1
$\langle a \rangle$	1	.	.	.	.	.	.	.	.	.	[2]	[2]	[3]
$\langle b \rangle$		1	.	.	.	.	.	.	.	.	[2]	[2]	[3]
$\langle c \rangle$			1	.	.	.	.	.	.	.	[2]	[2]	[3]
$\langle d \rangle$				1	.	.	.	.	.	.	[2]	[2]	[3]
$\langle e \rangle$					1						[2]	[2]	[3]
$\langle f \rangle$						1					[2]	[2]	[3]
$\langle g \rangle$							1				[2]	[2]	[3]
$\langle h \rangle$								1			[2]	[2]	[3]
$\langle i \rangle$									1		[2]	[2]	[3]
$\langle j \rangle$										1	[2]	[2]	[3]

10 rows (10 equations)

A 10-th degree univariate polynomial with respect to z

$$e = xX + yY + \boxed{z}Z + wW$$

W= 1

Computed coefficients

Solving this non-linear system will not be asked in the exam

Pose Recovery from Essential matrix

- Essential Matrix Decomposition

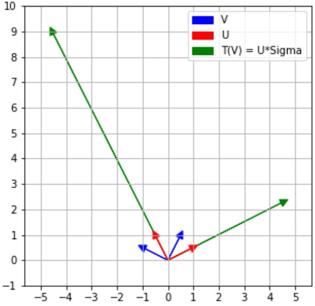
✓ Assume that we have obtained an essential matrix E. Recall that essential matrix E encodes the camera pose information. We aim to recover rotation and translation from the matrix E.

$$\boxed{\mathrm{E} = [T_\times]R \quad \textit{essential matrix}}$$

✓ Recap on singular value decomposition

$$\mathbf{T} = \underset{\uparrow}{\mathbf{U}}\, \boldsymbol{\Sigma}\, \mathbf{V}^{-1}$$

Composed of orthogonal basis vectors



Pose Recovery from Essential matrix

- Essential Matrix Decomposition

✓ Lemma: singular value of 3*3 Essential matrix satisfies the form $\boxed{[\sigma, \sigma, 0]^T}$
We do not provide proof here. If you have interest, you can check the reference [1]

✓ Decomposition of essential matrix E
Mathematical and geometric forms

$$E = U\Sigma V^T = U \boxed{\begin{bmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & 0 \end{bmatrix}} V^T = [t]^\wedge R.$$

Mathematical form

Geometric form

[1] "Multiple View Geometry in Computer Vision": R. Hartley and A. Zisserman
Link: https://www.robots.ox.ac.uk/~vgg/hzbook/

Pose Recovery from Essential matrix

➢ Essential Matrix Decomposition

✓ Can we directly extract R and t from SVD result?

$$E = U\Sigma V^T = U \boxed{\begin{bmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & 0 \end{bmatrix}} V^T = [\boldsymbol{t}]^\wedge R.$$

We cannot directly extract t and R (no skew-symmetric matrix)

We have to further transform it

Skew-symmetric matrix

✓ Rewrite $\sum$ by matrix multiplication (not unique)

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

Pose Recovery from Essential matrix

➢ Essential Matrix Decomposition

✓ Decomposition of essential matrix

SVD of E

Decomposition of $\sum$

$$E = U \begin{bmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T = U \begin{bmatrix} 0 & a & 0 \\ -a & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} V^T$$

$$= U \begin{bmatrix} 0 & a & 0 \\ -a & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} U^T U \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} V^T = [\boldsymbol{t}]^\wedge R$$

Skew-symmetric matrix

Associative law

Introducing an **identity** matrix for derivation

Pose Recovery from Essential matrix

- Essential Matrix Decomposition

$$\boxed{U\begin{bmatrix}0 & a & 0\\ -a & 0 & 0\\ 0 & 0 & 0\end{bmatrix}U^T}\boxed{U\begin{bmatrix}0 & -1 & 0\\ 1 & 0 & 0\\ 0 & 0 & 1\end{bmatrix}V^T} = \boxed{[\boldsymbol{t}]^\wedge}\boxed{R}$$

✓ Rotation and translation results

$$\boxed{[\boldsymbol{t}_1]^\wedge = U\begin{bmatrix}0 & a & 0\\ -a & 0 & 0\\ 0 & 0 & 0\end{bmatrix}U^T} = [U\underbrace{\begin{bmatrix}0\\ 0\\ -a\end{bmatrix}}_{\text{3*1 vector}}]^\wedge$$

Matrix

$$\boxed{\boldsymbol{t}_1} = U\begin{bmatrix}0\\ 0\\ -a\end{bmatrix} = [U_0 \quad U_1 \quad U_2]\begin{bmatrix}0\\ 0\\ -a\end{bmatrix} = \boxed{-U_2 a}$$

Columns

Translation

$$\boxed{R_1 = U\begin{bmatrix}0 & -1 & 0\\ 1 & 0 & 0\\ 0 & 0 & 1\end{bmatrix}V^T}$$

Rotation

Pose Recovery from Essential matrix

- Essential Matrix Decomposition

$$\Sigma = \begin{bmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & 0 \end{bmatrix} = \boxed{\begin{bmatrix} 0 & a & 0 \\ -a & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}$$

Previous decomposition

✓ Another decomposition of $\sum$ leads to another result

$$\Sigma = \begin{bmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & 0 \end{bmatrix} = \boxed{\begin{bmatrix} 0 & -a & 0 \\ a & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}$$

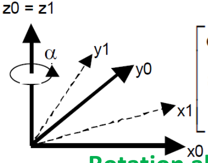
$$\Rightarrow \begin{cases} \boxed{t_2} = U \begin{bmatrix} 0 \\ 0 \\ a \end{bmatrix} = U_2 a = \boxed{-t_1} \\ R_2 = U \boxed{\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}} V^T \triangleq U W^T V^T \end{cases}$$

Another rotation and translation results

Pose Recovery from Essential matrix

➢ Essential Matrix Decomposition

✓ A more concise expression of rotation and translation



$$\begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Rotation along the z-axis (introduced before)

$$R_z(+\frac{\pi}{2}) = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Rotation derived before

$$R_1 = U \boxed{\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}} V^T \qquad R_2 = U \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} V^T$$

$$\Sigma = \begin{bmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Defined before (eigen values)

$$R_1 = U \boxed{R_z(+\frac{\pi}{2})} V^T, [t_1]^\wedge = U \boxed{R_z(-\frac{\pi}{2})} \Sigma U^T$$

$$R_2 = U R_z(-\frac{\pi}{2}) V^T, [t_2]^\wedge = U R_z(+\frac{\pi}{2}) \Sigma U^T$$

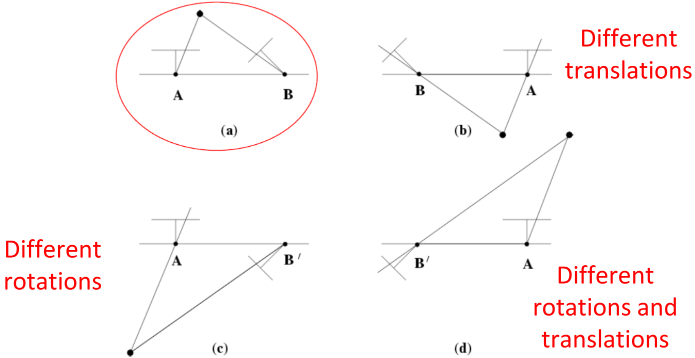
$$[t_1]^\wedge = U \boxed{\begin{bmatrix} 0 & a & 0 \\ -a & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}} U^T$$

Translation derived before

Pose Recovery from Essential matrix

- Essential Matrix Decomposition

✓ **Four possible solutions**: There exists only one solution where points are in front of both cameras



3D reconstruction will be introduced in the next class

Pose Recovery from Essential matrix

➢ Application of Fundamental Matrix

✓ Can R, T, K_1, K_2 be extracted from *F*?

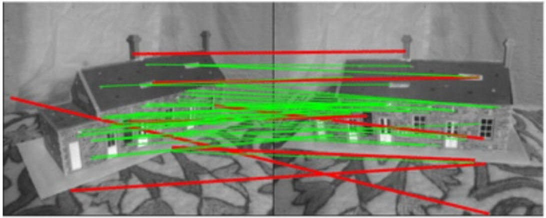
- In general we cannot achieve this since infinite solutions exist
- However, if the coordinates of the **principal points** of each camera are known and the two cameras have the same **focal length** f in pixels, then R,T,f can determined uniquely.
- This is an advanced knowledge. We will not introduce it in our class.

Pose Recovery from Essential matrix

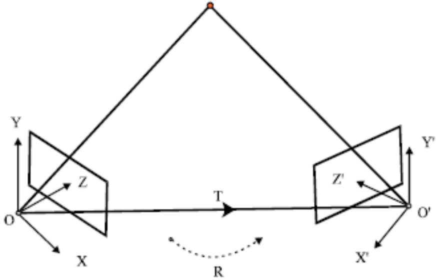
➢ Application of Fundamental Matrix

✓ If we do not use Fundamental matrix to recover camera pose, what is its application?

- We do not need to normalize image points.
- We can use consensus constraint w.r.t. fundamental matrix to remove outliers.



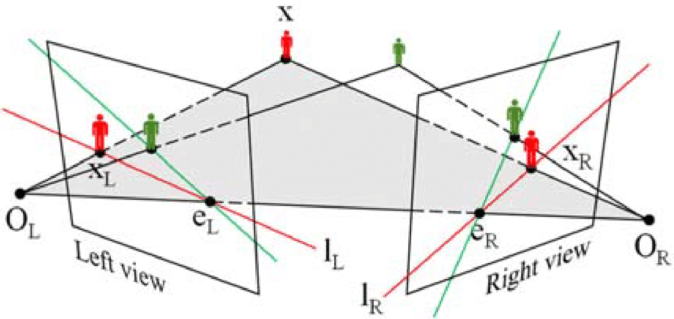
(a) 55 inliers (green) and 8 outliers (red) among 63 pairs



1D Correspondence Search

➢ Overview

✓ Reviewing drawback of brute-force matching
✓ 1D search based on epipolar constraint (**application of epipolar lines**)



1D Correspondence Search

➢ Review and Motivation

Given a point, p_L, in the left image, how do we find its correspondence, p_R, in the right image? A straightforward strategy is brute-force matching (search) strategy.

Left image

Right image

1D Correspondence Search

- Review and Motivation

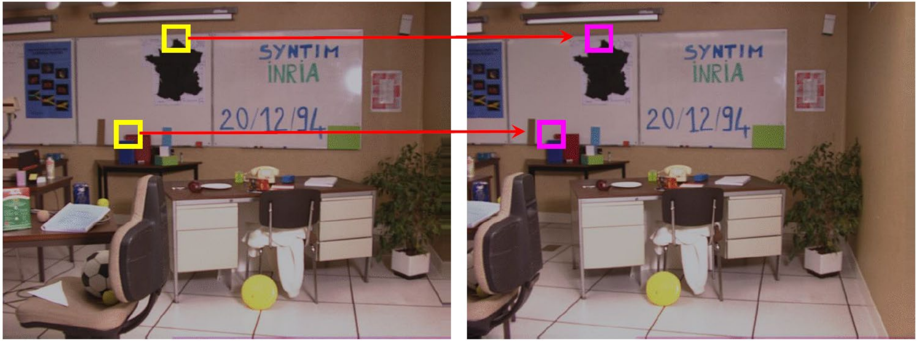
Brute-force Matching: compare each candidate patch from the image with all possible candidate patches from the right image.

Left image Right image

1D Correspondence Search

➢ Review and Motivation

This 2D exhaustive search is computationally very expensive!

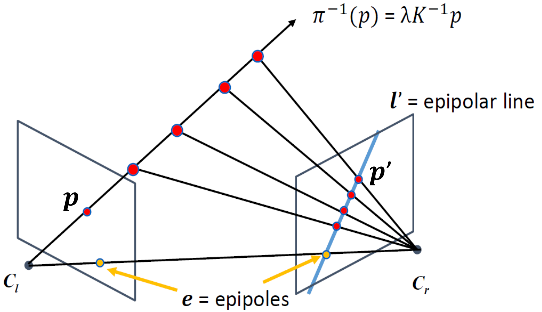


1D Correspondence Search

➢ Problem Formulation

✓ Can we make the correspondence search 1D?

- The epipolar line is the projection of a back projected ray $\pi^{-1}(p)$ onto the other camera image
- Potential matches for $\boldsymbol{p}$ have to lie on the corresponding epipolar line $\boldsymbol{l}'$

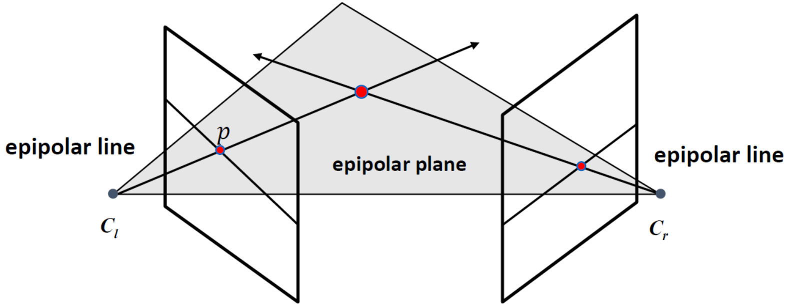


1D Correspondence Search

➢ Problem Formulation

✓ Corresponding points must lie along the epipolar lines: this constraint is called **epipolar constraint**.

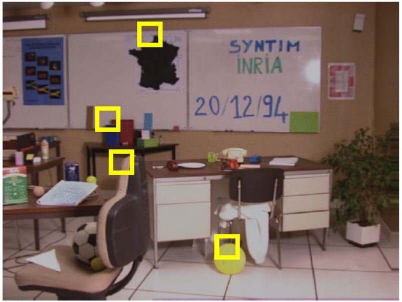
✓ The epipolar constraint reduces correspondence problem to 1D search along the epipolar line.



1D Correspondence Search

- Problem Formulation

Thanks to the epipolar constraint, corresponding points can be searched for along epipolar lines. Accordingly, the computational cost reduced to 1 dimension.



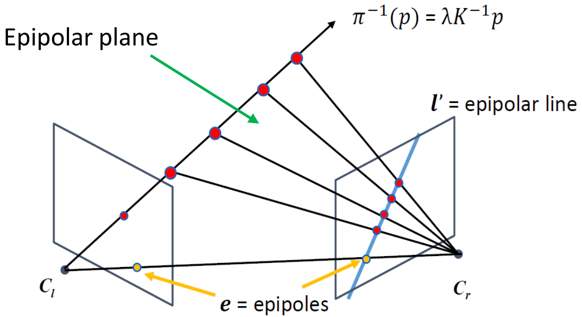
Left image

Right image

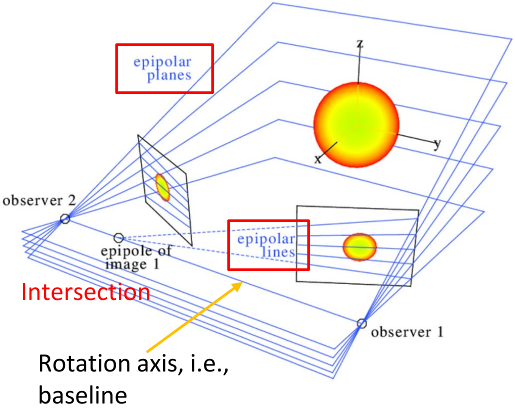
1D Correspondence Search

➢ Example Configurations of Epipolar Lines

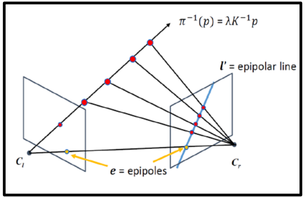
Fundamental: All the epipolar lines intersect at the epipole



Basis of epipolar geometry



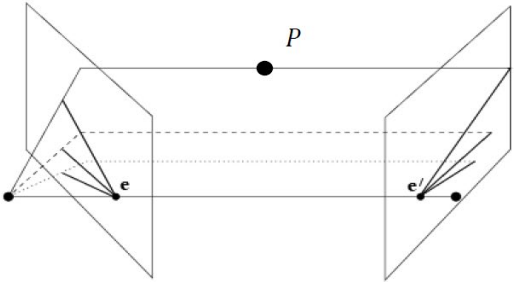
1D Correspondence Search

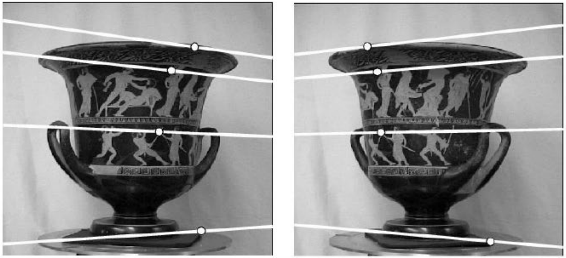


➢ Example 1: Converging Cameras

✓ A classical case (non-parallel principal axes)

As the position of the 3D point ***P*** changes, the epipolar lines **rotate about the baseline**





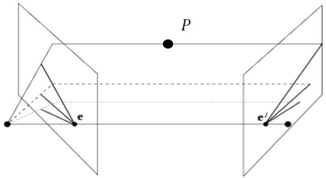
Left image

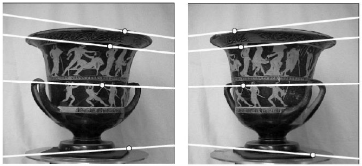
Right image

1D Correspondence Search

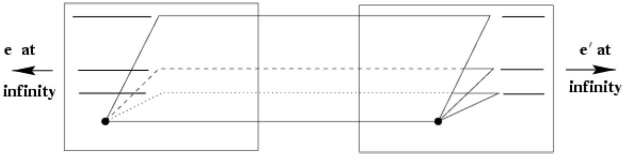
➢ Example 2: Identical and Horizontally-Aligned Cameras

✓ A special case (principal axes are parallel, and orthogonal to moving direction/baseline) Parallel epipolar lines do not intersect (or, the intersection lies at the infinity).



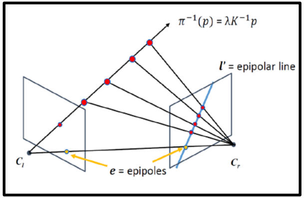


Left image Right image



Left image Right image

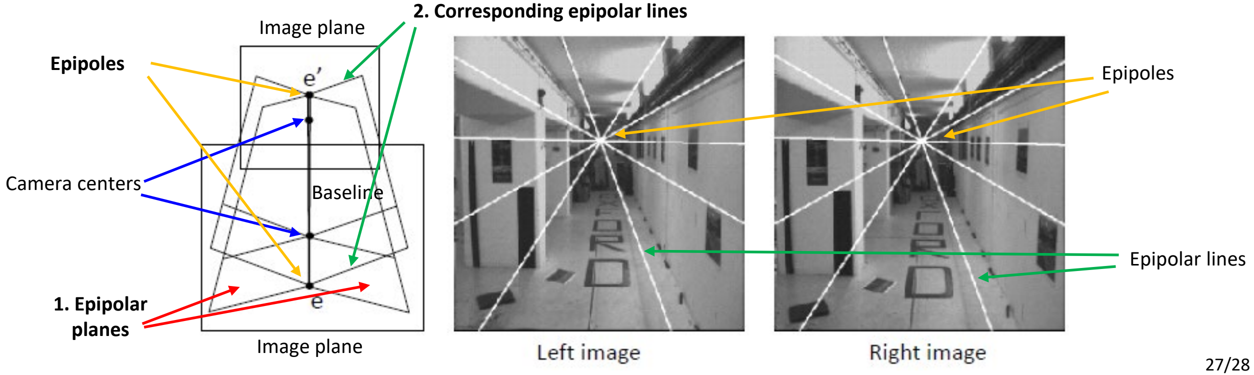
1D Correspondence Search



➢ Example 3: Forward Motion

✓ Another special case (principal axes are aligned to moving direction)

Epipolar lines radiating from the epipole (coordinates remain unchanged)



Summary

- Five-point Method
- Pose Recovery from Essential Matrix
- 1D Correspondence Search

Thank you for your listening!
If you have any questions, please come to me :-)