Computer Vision I - Geometry Estimation from two Images

Carsten Rother

25/11/2015

Roadmap for next four lectures

- Appearance-based Matching (sec. 4.1)
- Projective Geometry - Basics (sec. 2.1.1-2.1.4)
- Geometry of a Single Camera (sec 2.1.5, 2.1.6)
  - Camera versus Human Perception
  - The Pinhole Camera
  - Lens effects
- Geometry of two Views (sec. 7.2)
  - The Homography (e.g. rotating camera)
  - Camera Calibration (3D to 2D Mapping)
  - The Fundamental and Essential Matrix (two arbitrary images)
- Robust Geometry estimation for two cameras (sec. 6.1.4)
- Multi-View 3D reconstruction (sec. 7.3-7.4)
  - General scenario
  - From Projective to Metric Space
  - Special Cases

In the following we always ask same questions…

- Two-view transformations we look at:
  - Homography H: between two views
  - Camera matrix P (mapping from 3D to 2D)
  - Fundamental matrix F between two un-calibrated views
  - Essential matrix E between two calibrated views

- Derive geometrically: H, P, F, E, i.e. what do they mean?

- Calibration: Take primitives (points, lines, planes, cones,…) to compute H, P, F, E :
  - What is the minimal number of points to compute them (this topic is justified when we look at robust methods)
  - If we have many points with noise: what is the best way to computer them: algebraic error versus geometric error?

- Can we derive the intrinsic (K) an extrinsic (R, C) parameters from H, P, F, E?

- What can we do with H, P, F, E? (e.g. Panoramic Stitching)

Topic 1: Homography H

- Derive geometrically H
- Calibration: Take measurements (points) to compute H
  - How do we do that with a minimal number of points?
  - How do we do that with many points?
- Can we derive the intrinsic (K) an extrinsic (R, C) parameters from H?
- What can we do with H ?

Definition Homography

- **Definition**: A projectivity (or homography) h is an invertible mapping h from P^2 to P^2 such that three points x_1, x_2, x_3 lie on the same line if an only if $h(x_1), h(x_2), h(x_3)$ do.

- **Theorem**: A mapping h from P^2 to P^2 is a homography if and only if there exists a non-singular 3×3 matrix H with $h(x) = Hx$

- In equations: $\mathrm{x}' = Hx$

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Transformation matrix H

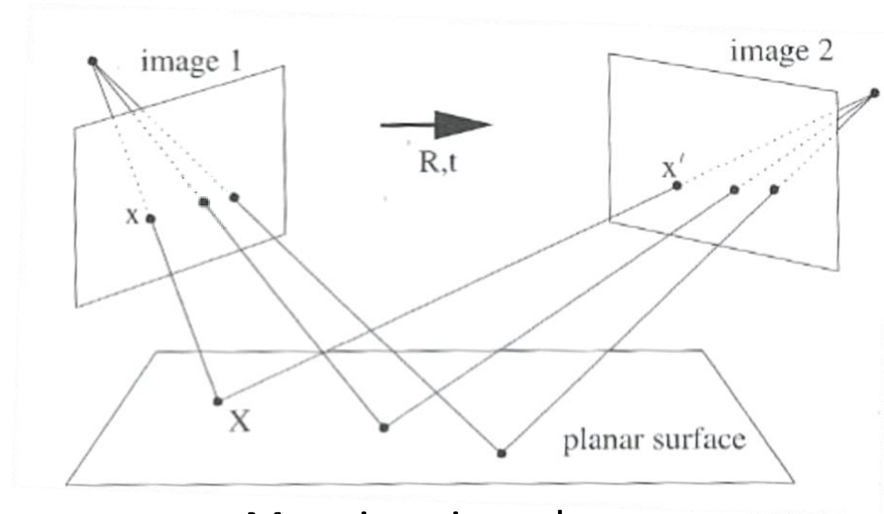
- H has 8 DoF

Homographies in the real world

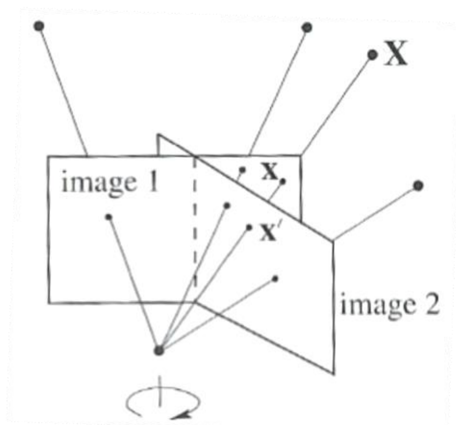
Image 2 Image 1

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

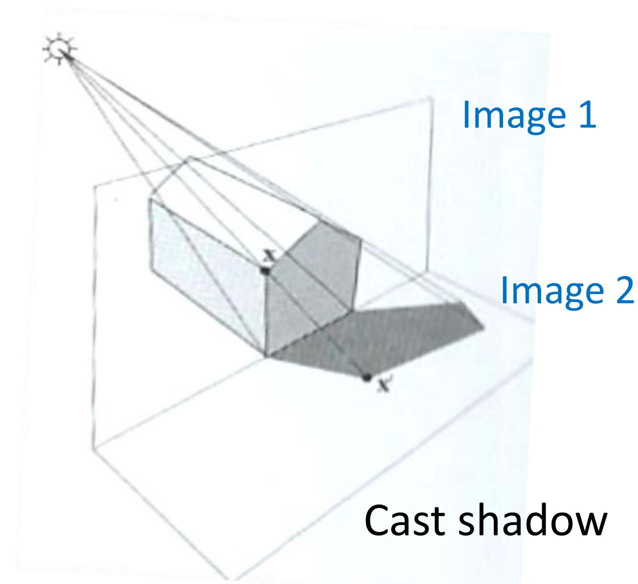
Transformation matrix H



Mapping via a plane



Rotating camera



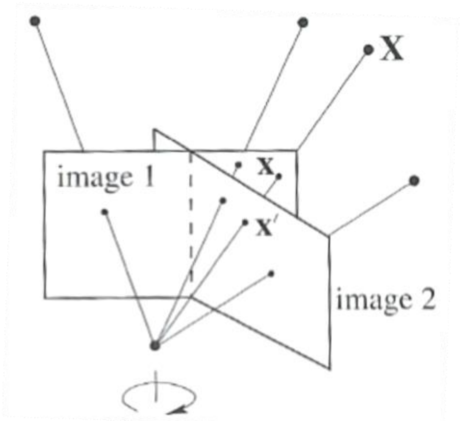
Cast shadow

Homography from a rotating camera - Derivation

Notation: $\mathbf{x}$ (homogenous 2D), $\tilde{\mathbf{x}}$ (inhomogenous 2D),
$\boldsymbol{X}$ (homogenous 3D), $\tilde{\boldsymbol{X}}$(inhomogenous 3D)

$$\boldsymbol{K} = \begin{bmatrix} f & s & p_x \\ 0 & mf & p_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{x} = \boldsymbol{K}\,\boldsymbol{R}\,(\boldsymbol{I}_{3\times 3} \mid -\tilde{\boldsymbol{C}})\,\boldsymbol{X}$$



Camera 0: $\boldsymbol{x_0} = \boldsymbol{K_0}\,\tilde{\boldsymbol{X}}$ $(\text{in 3D}: \boldsymbol{K_0^{-1}}\boldsymbol{x_0} = \tilde{\boldsymbol{X}})$
Camera 1: $\boldsymbol{x_1} = \boldsymbol{K_1}\,\boldsymbol{R}\,\tilde{\boldsymbol{X}}$

Put it toghter: $\boldsymbol{x_1} = \boldsymbol{K_1}\boldsymbol{R}\boldsymbol{K_0^{-1}}\boldsymbol{x_0}$
Hence $\boldsymbol{H} = \boldsymbol{K_1}\boldsymbol{R}\boldsymbol{K_0^{-1}}$ is a homography (general 3x3 matrix) with 8 DoF

How to compute (i.e. calibrate) H

- We have $\lambda x' = Hx$

- H has 8 DoF

- We get for each pair of matching points (x', x) the 3 equations:

  1) $h_{11}x_1 + h_{12}x_2 + h_{13}x_3 = \lambda x_1'$

  2) $h_{21}x_1 + h_{22}x_2 + h_{23}x_3 = \lambda x_2'$

  3) $h_{31}x_1 + h_{32}x_2 + h_{33}x_3 = \lambda {x_3}'$

- Eliminate λ (by taking ratios). This gives 2 linear independent equations:
  Here 1) divide by 2) gives:

$$(x_1x_2', x_2x_2', x_3x_2', -x_1x_1', -x_2x_1', -x_3x_1')(h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23})^T = 0$$

How to compute/calibrate H

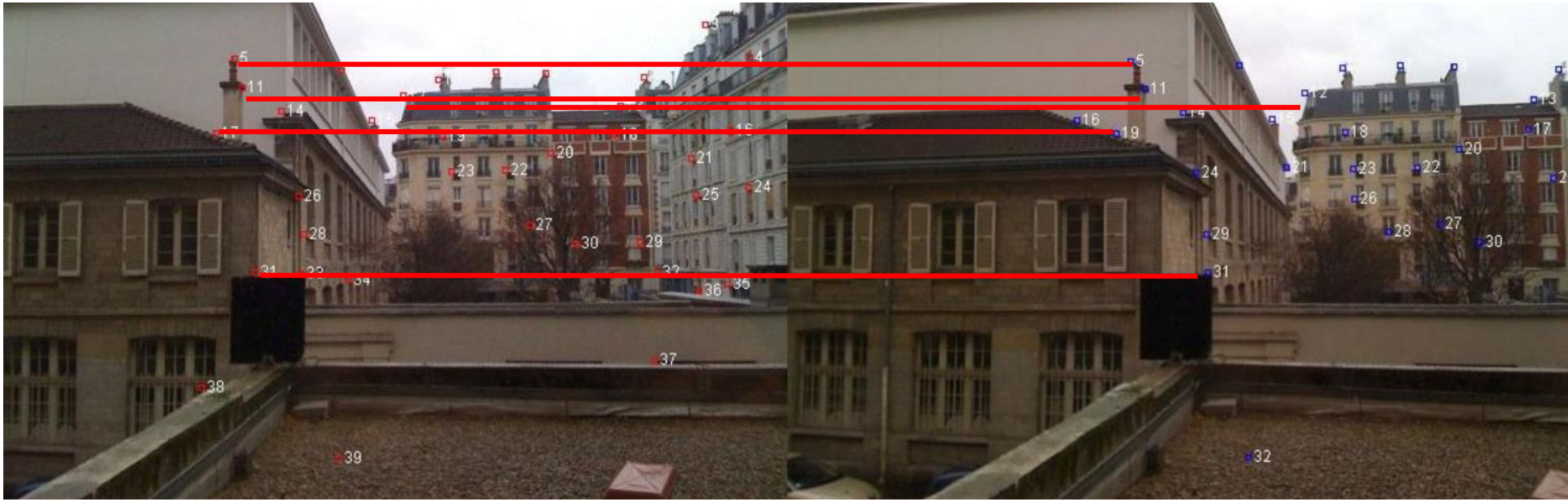
- Put it together:

$$\begin{bmatrix} x_1x_2' & x_2x_2' & x_3x_2' & -x_1x_1' & -x_2x_1' & -x_3x_1' & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1x_3' & x_2x_3' & x_3x_3' & -x_1x_2' & -x_2x_2' & -x_3x_2' \\ & & & & \vdots & & & & \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = 0$$

- We need a minimum of 4 points to get $Ah = 0$ with $A\ is\ 8 \times 9$ matrix, and $h\ is\ 9 \times 1$ vector

- Solution for h is the right null space of A

Often we have many, slightly wrong point-matches



We know how to do: $\boldsymbol{x}^* = argmin_{\boldsymbol{x}} ||A\boldsymbol{x}||$ subject to $||\boldsymbol{x}|| = \mathbf{1}$

<u>Algorithm:</u>

1) Take $m \geq 4$ point matches (x, x')
2) Assemble A with $A\boldsymbol{h} = \mathbf{0}$
3) compute $\boldsymbol{h}^* = argmin_h ||A\boldsymbol{h}||$ subject to $||\boldsymbol{h}|| = \mathbf{1}$, use SVD to do this.

A numerically more stable solution

- Coefficients of an equation system should be in the same order of magnitude, in order to not lose significant digits
- In pixels: $x_a x_b' \sim 1e6$
- Conditioning: scale and shift points to be in [-1..1] (or +/- $\sqrt{2}$)
- A general rule, not only for homography computation
- How to do it:

$$s = \max_i(\|x_i\|)$$
$$t = \mathrm{mean}(x_i)$$

$$T = \begin{bmatrix} \frac{1}{s} & 0 & -\frac{t_x}{s} \\ 0 & \frac{1}{s} & -\frac{t_y}{s} \\ 0 & 0 & 1 \end{bmatrix} \qquad u = Tx$$

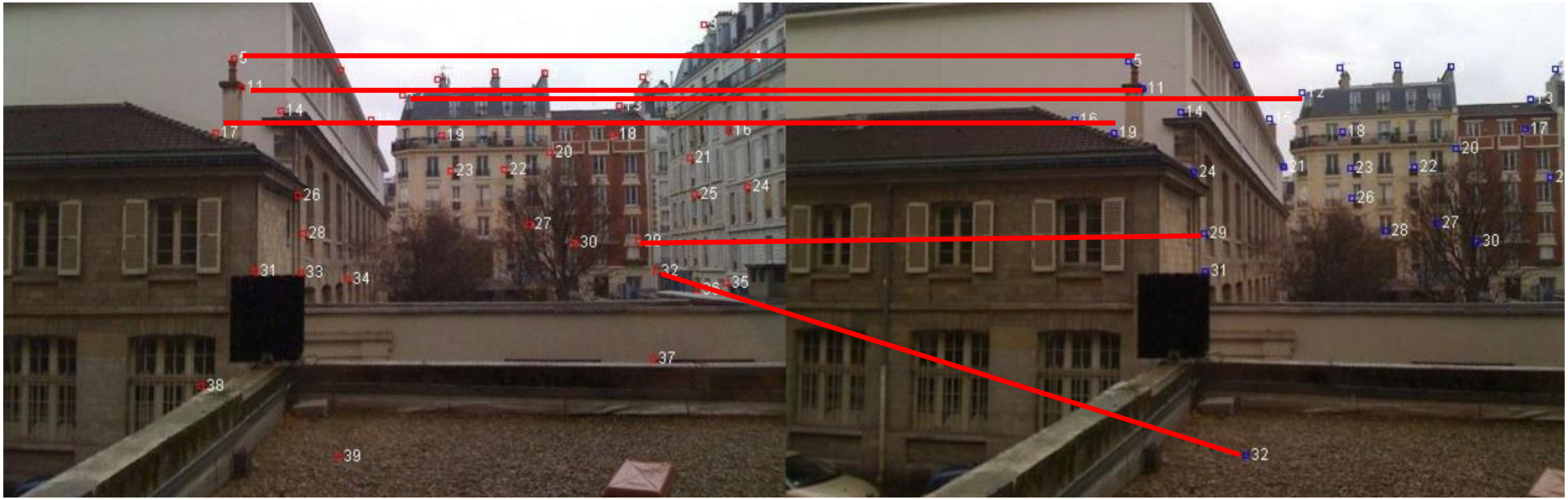
A more numerically stable solution

Algorithm:

1) Take $m \geq 4$ point matches (x, x')
2) Compute T, and condition points: $x_n = Tx$; $x_n' = T'x'$
3) Assemble A with $A\boldsymbol{h} = 0$
4) compute $\boldsymbol{h}^* = argmin_h ||A\boldsymbol{h}||$ subject to $||\boldsymbol{h}|| = \mathbf{1}$,
   use SVD to do this.
4) Get H of unconditioned points: $T'^{-1}HT$ (Note: $T'x' = HTx$)

[See HZ page 109]

Motivation for next lecture



Question 1: If a match is completely wrong then $argmin_h ||A\boldsymbol{h}||$ is a bad idea

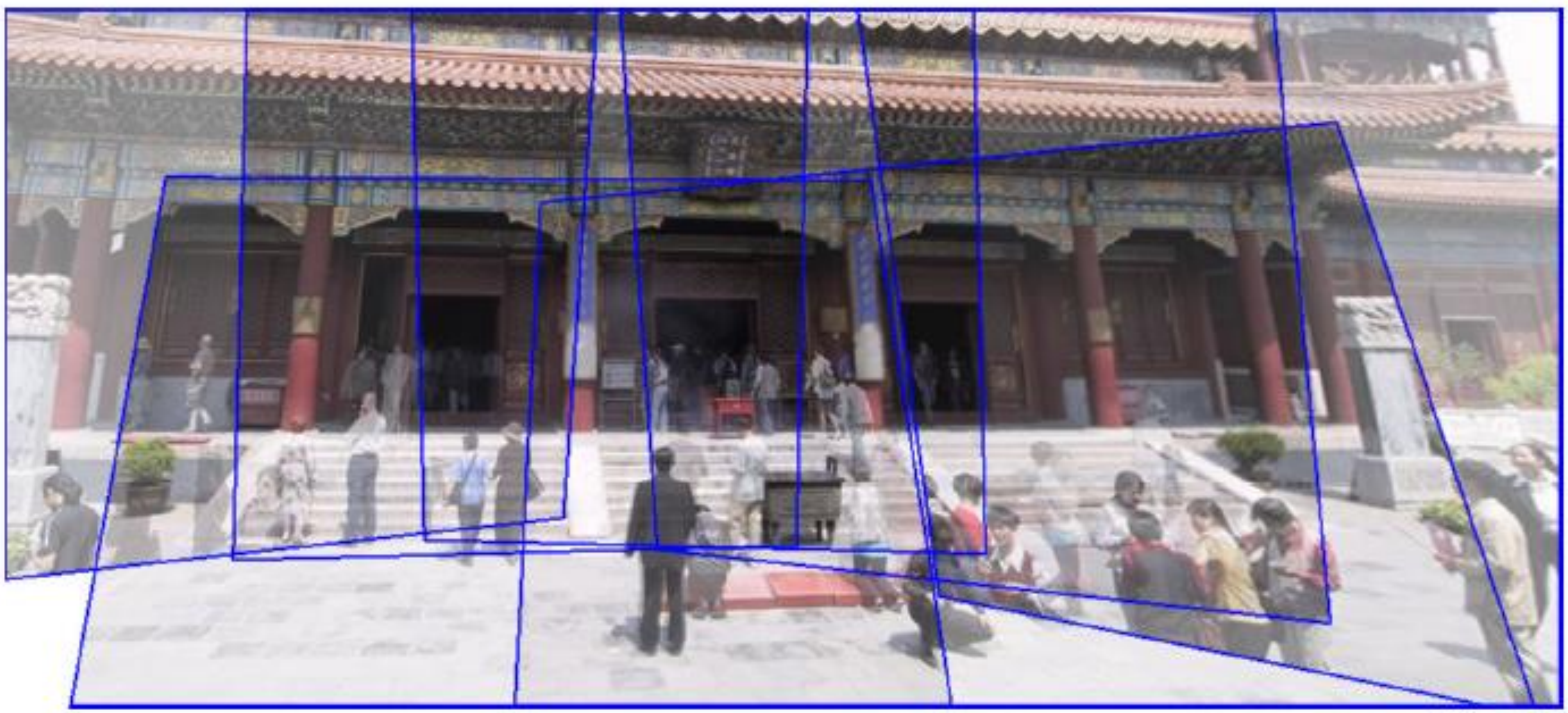
Question 2: If a match is slightly wrong then $argmin_h ||A\boldsymbol{h}||$ might not be perfect.
Better might be a geometric error: $argmin_h ||\boldsymbol{H}x - x'||$

Can we get K's and R from H?

- Assume we have $H = K_1 R K_0^{-1}$ of a rotating camera, can we get out K_1, R, K_0 ?
- H has 8 DoF
- K_1, R, K_0 have together 13 DoF
- Not directly possible, only with assumptions on K.
(No application needs such a decomposition)

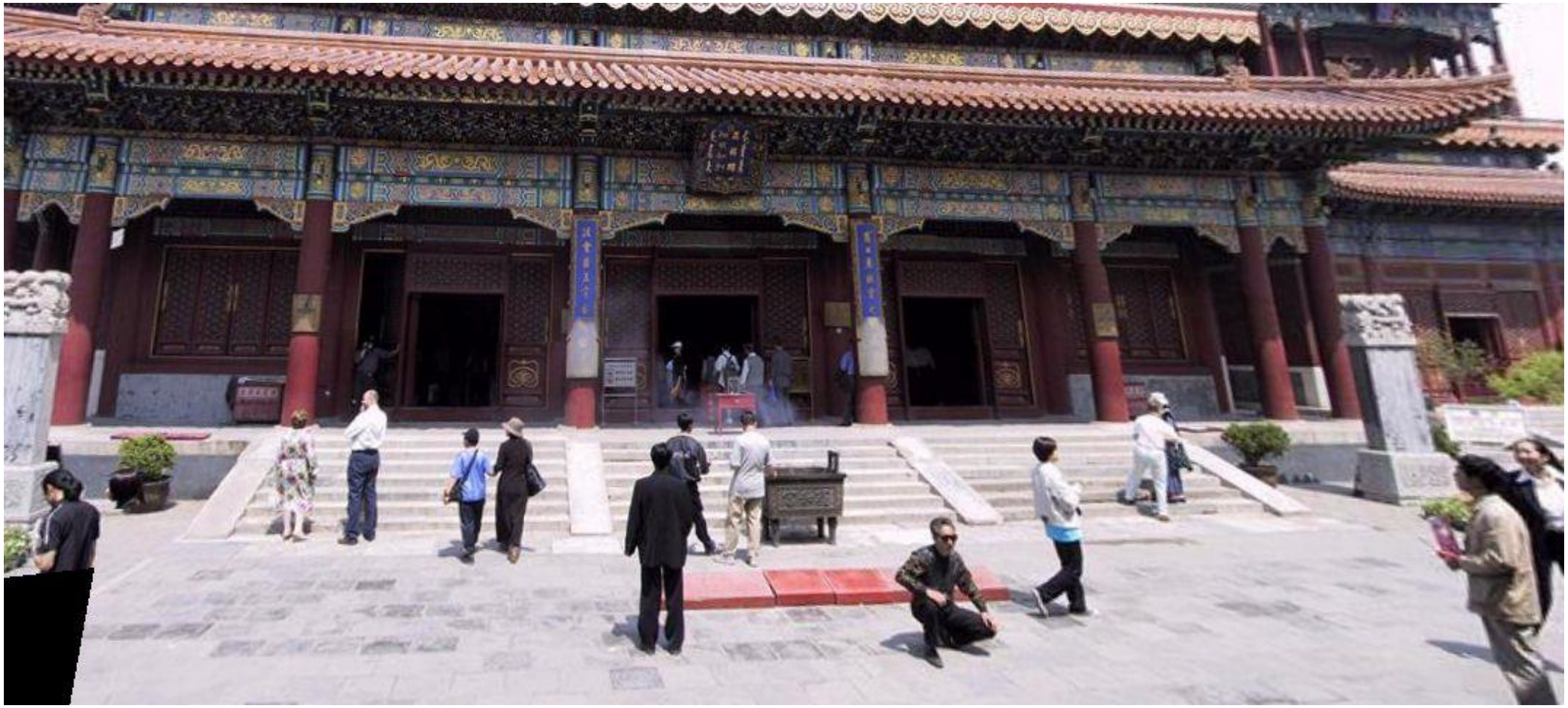
What can we do with H?

- Panoramic stitching with rotating camera (exercise later)



Warp images into a canonical view: $x' = Hx$

What can we do with H?



What can we do with H?

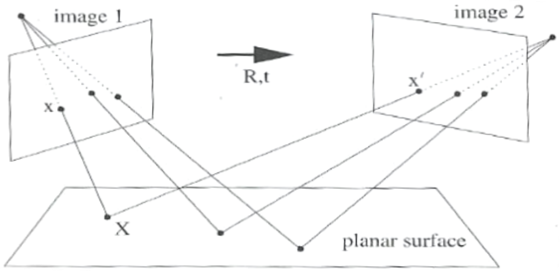
- Plane-based augmented realty

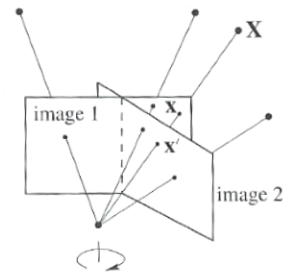
The figure appears to stand on the board. For this the mapping between the board and the image plane is needed.

[Graz University]

Homography H: Summary

- Derive geometrically H





- Calibration: Take measurements (points) to compute H
  - Minimum of 4 points. Solution: right null space of $Ah = 0$
  - Many points. Use SVD to solve $\boldsymbol{h}^* = argmin_h||A\boldsymbol{h}||$

- Can we derive the intrinsic (K) an extrinsic (R, C) parameters from H?
  -> hard. Not discussed much

- What can we do with H ?

  -> augmented reality on planes, panoramic stitching

Roadmap for next four lectures

- Appearance-based Matching (sec. 4.1)
- Projective Geometry - Basics (sec. 2.1.1-2.1.4)
- Geometry of a Single Camera (sec 2.1.5, 2.1.6)
  - Camera versus Human Perception
  - The Pinhole Camera
  - Lens effects
- Geometry of two Views (sec. 7.2)
  - The Homography (e.g. rotating camera)
  - Camera Calibration (3D to 2D Mapping)
  - The Fundamental and Essential Matrix (two arbitrary images)
- Robust Geometry estimation for two cameras (sec. 6.1.4)
- Multi-View 3D reconstruction (sec. 7.3-7.4)
  - General scenario
  - From Projective to Metric Space
  - Special Cases

Topic 2: Camera Matrix P

- Derive geometrically P

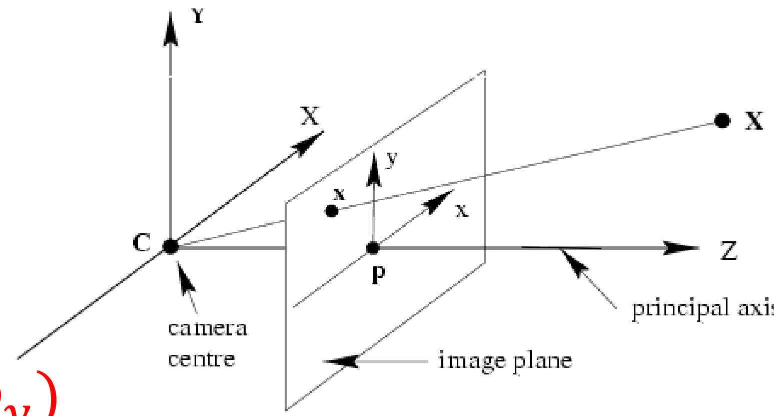
- Calibration: Take measurements (points) to compute P
  - How do we do that with a minimal number of points?
  - How do we do that with many points?

- Can we derive the intrinsic (K) an extrinsic (R, C) parameters from P?

- What can we do with P?

Geometric Derivation: Camera Matrix (Reminder)

- Camera matrix P has 11 DoF



- Intrinsic parameters
  - Principal point coordinates (p_x, p_y)
  - Focal length f
  - Pixel magnification factors m
  - Skew (non-rectangular pixels) s

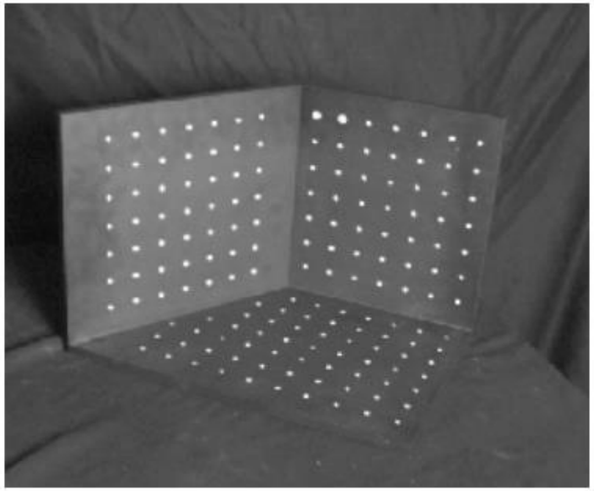
$$\boldsymbol{x} = \boldsymbol{P}\,\boldsymbol{X}$$

$$\boldsymbol{x} = \boldsymbol{K}\,\boldsymbol{R}\,(\boldsymbol{I}_{3\times3} \mid -\tilde{\boldsymbol{C}})\,\boldsymbol{X}$$

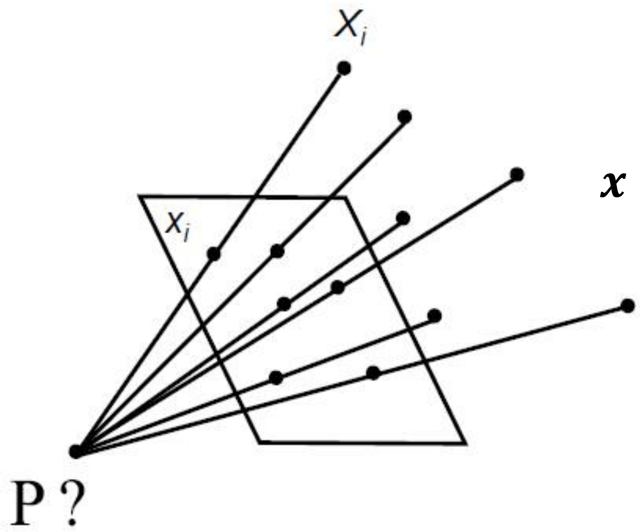
$$\boldsymbol{K} = \begin{bmatrix} f & s & p_x \\ 0 & mf & p_y \\ 0 & 0 & 1 \end{bmatrix}$$

- Extrinsic parameters
  - Rotation $\boldsymbol{R}$ (3DoF) and translation $\tilde{\mathbf{C}}$ (3DoF) relative to world coordinate system

How can we compute/calibrate P?



Calibration pattern



$$x = P\,X$$

$$x = K\,R\,(I_{3\times3} \mid -\tilde{C})\,X$$

Important move in all directions: x, y, z

How can we compute/calibrate P?

- We have $\lambda x' = PX$
- P has 11 DoF
- We get for each point pair (x', X) 3 equations, but only 2 linear independent once, by taking ration (to get rid of λ)
- We need a minimum of 6 Points to get 12 equations

<u>Algorithm (DLT - Direct Linear Transform):</u>

1) Take $m \geq 6$ points.
2) Condition points X, x' using T, T'
3) Assemble A with $Ap = 0$ ($A\ is\ m \times 12$ and p is vectorized P)
4) compute $p^* = argmin_p \|Ap\|$ subject to $\|p\| = 1$
   use SVD to do this.
5) Get out unconditioned $P = T'^{-1}PT$ (note $T'x' = P\,TX$)

Note: a version with minimal number of points (6) is same as with many points

[See extended version: HZ page 181]

How can we get K,R,$\tilde{C}$ from P

- Assume P is known, can we get out $K, R, \tilde{C}$?
- P has 11 DoF
- $K, R, \tilde{C}$ have together 5+3+3=11 DoF (so it is possible)

$$\boldsymbol{x} = \boldsymbol{P}\,\boldsymbol{X}$$
$$\boldsymbol{x} = \boldsymbol{K}\,\boldsymbol{R}\,(\boldsymbol{I}_{3\times 3} \mid -\tilde{\boldsymbol{C}})\,\boldsymbol{X}$$

- <u>How to do it:</u>

1) The camera center $\tilde{C}$ is the right nullspace of P

$$PC = K\,R\,(\tilde{C} - \tilde{C}) = 0$$

2) $P = [KR| - KR\tilde{C}]$;

$$A = KR$$

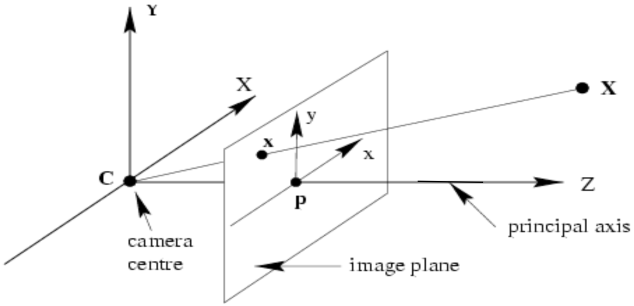
can be done with unique RQ decomposition, where R is upper-triangular matrix and Q a rotation matrix (see HZ page 579)

What can we do with P?

- Many things can be done with an externally and internally calibrated camera
- Robot navigation, augmented reality, photogrammetry …

Camera Matrix P: Summary

- Derive geometrically P



$$\boldsymbol{x} = \boldsymbol{P}\,\boldsymbol{X}$$
$$\boldsymbol{x} = \boldsymbol{K}\,\boldsymbol{R}\,(\boldsymbol{I}_{3\times 3} \mid -\tilde{\boldsymbol{C}})\,\boldsymbol{X}$$

- Calibration: Take measurements (points) to compute P
  - 6 or more points. Use SVD to solve $\boldsymbol{p}^* = argmin_h \|A\boldsymbol{p}\|$

- Can we derive the intrinsic (K) an extrinsic (R, C) parameters from H?
  -> yes, use SVD and RQ decomposition

- What can we do with P ?

  -> very many things (robotic, photogrammetry, augmented reality, …)

Roadmap for next four lectures

- Appearance-based Matching (sec. 4.1)

- Projective Geometry - Basics (sec. 2.1.1-2.1.4)

- Geometry of a Single Camera (sec 2.1.5, 2.1.6)
  - Camera versus Human Perception
  - The Pinhole Camera
  - Lens effects

- Geometry of two Views (sec. 7.2)
  - The Homography (e.g. rotating camera)
  - Camera Calibration (3D to 2D Mapping)
  - The Fundamental and Essential Matrix (two arbitrary images)

- Robust Geometry estimation for two cameras (sec. 6.1.4)

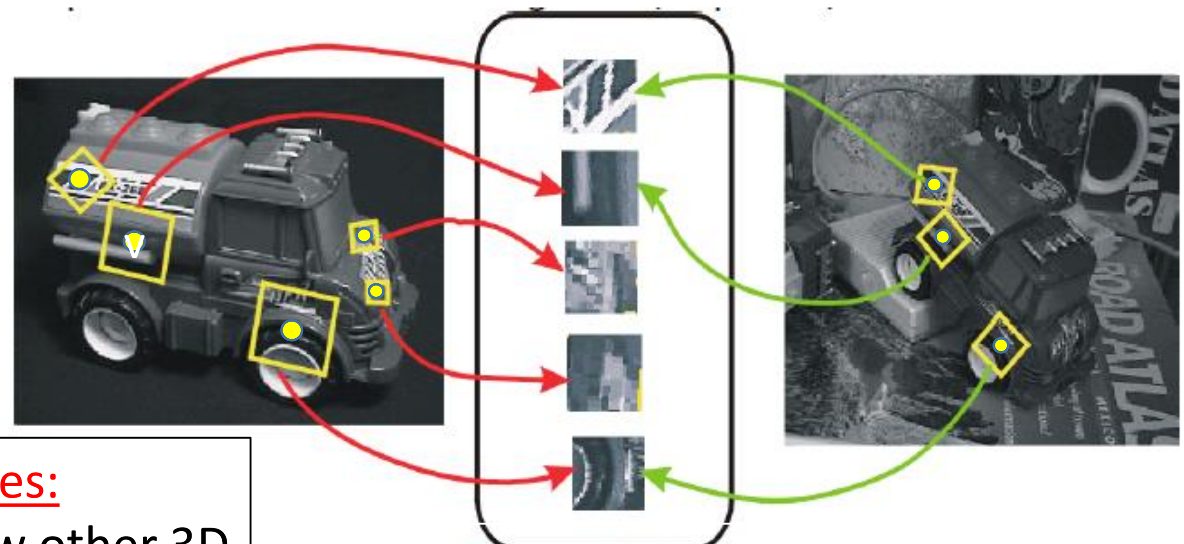
- Multi-View 3D reconstruction (sec. 7.3-7.4)
  - General scenario
  - From Projective to Metric Space
  - Special Cases

Topic 3: Fundamental/Essential Matrix F/E

- Derive geometrically F/E
- Calibration: Take measurements (points) to compute F/E
  - How do we do that with a minimal number of points?
  - How do we do that with many points?
- Can we derive the intrinsic (K) an extrinsic (R, C) parameters from F/E?
- What can we do with F/E?

Reminder: Matching two Images

- Find interest points
- Find orientated patches around interest points to capture appearance
- Encode patch appearance in a descriptor
- Find matching patches according to appearance (similar descriptors)
- Verify matching patches according to geometry (later lecture)

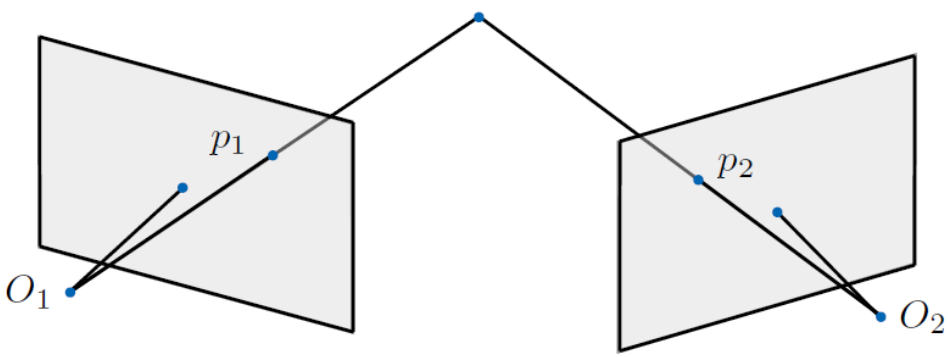


Feature descriptors

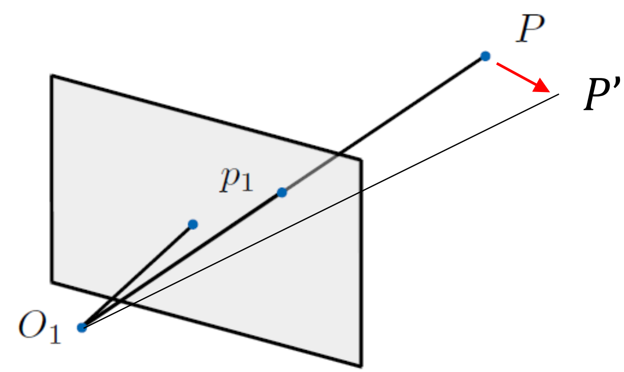
We will discover in next slides:
Seven 3D points defines how other 3D points match between 2 views!

3D Geometry

Non-moving scene

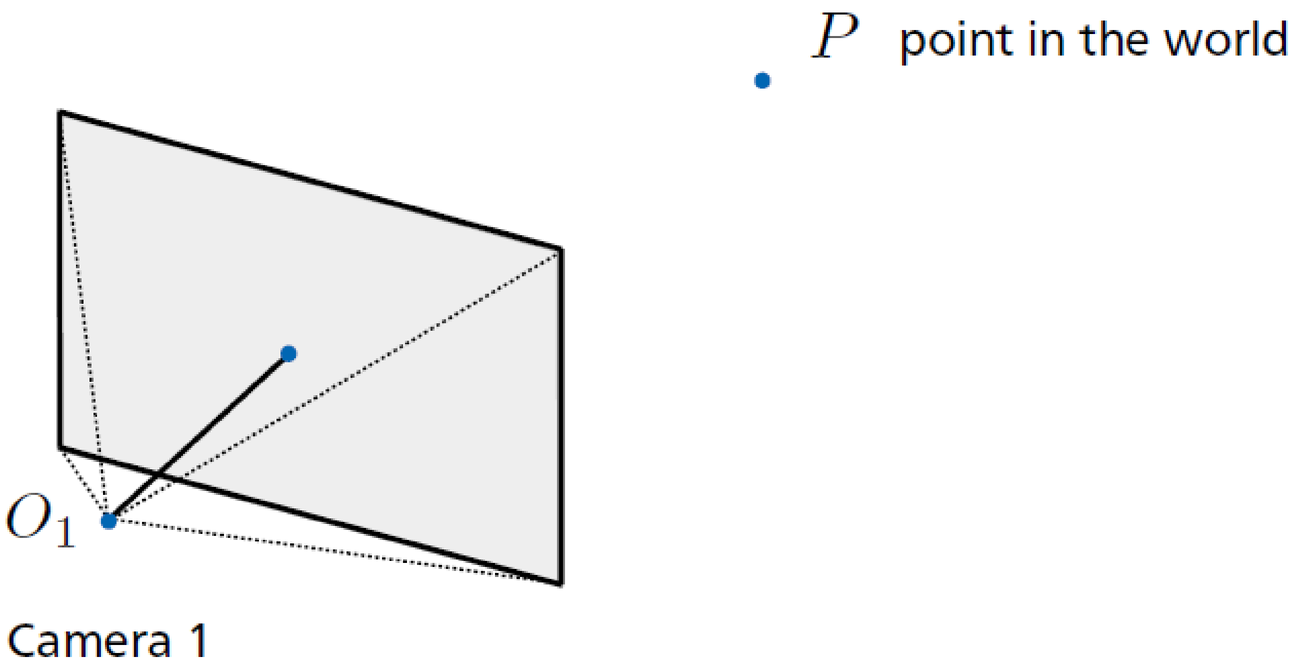


Rigidly (6D) moving scene

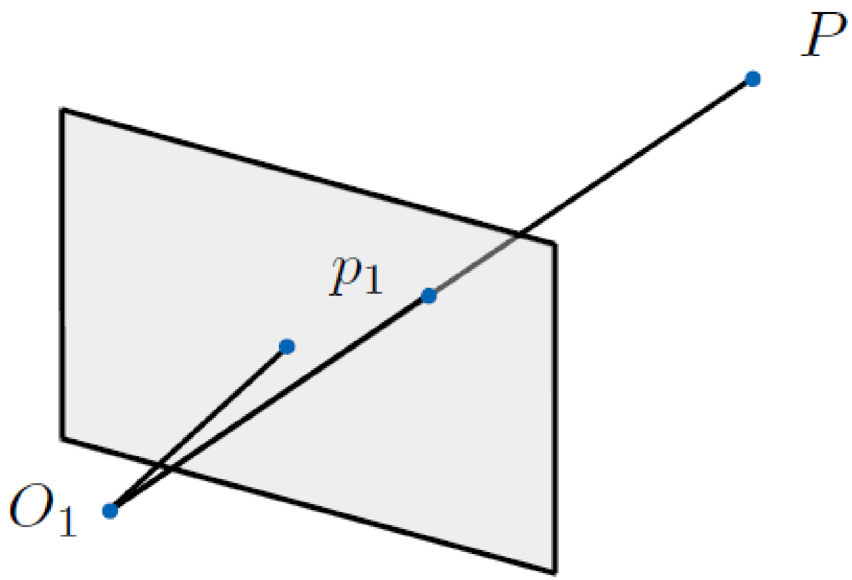


Both cases are equivalent for the following derivations

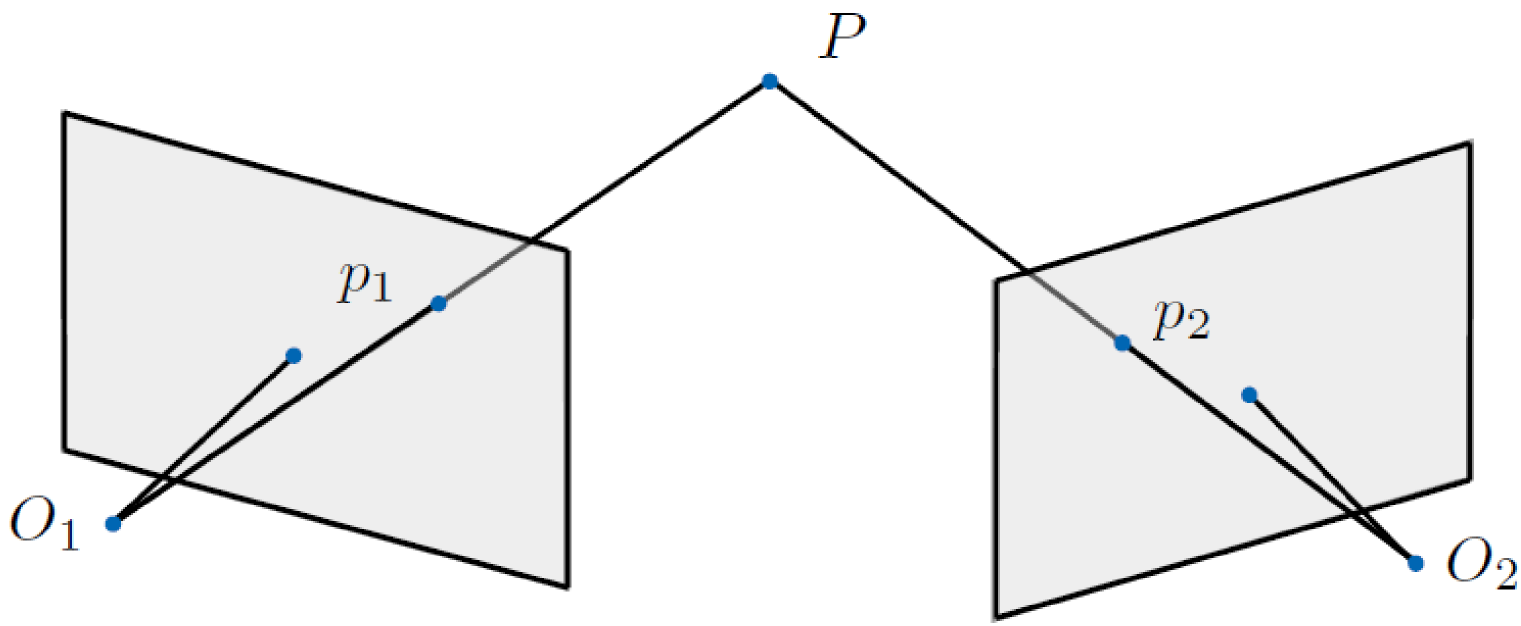
Epipolar Geometry



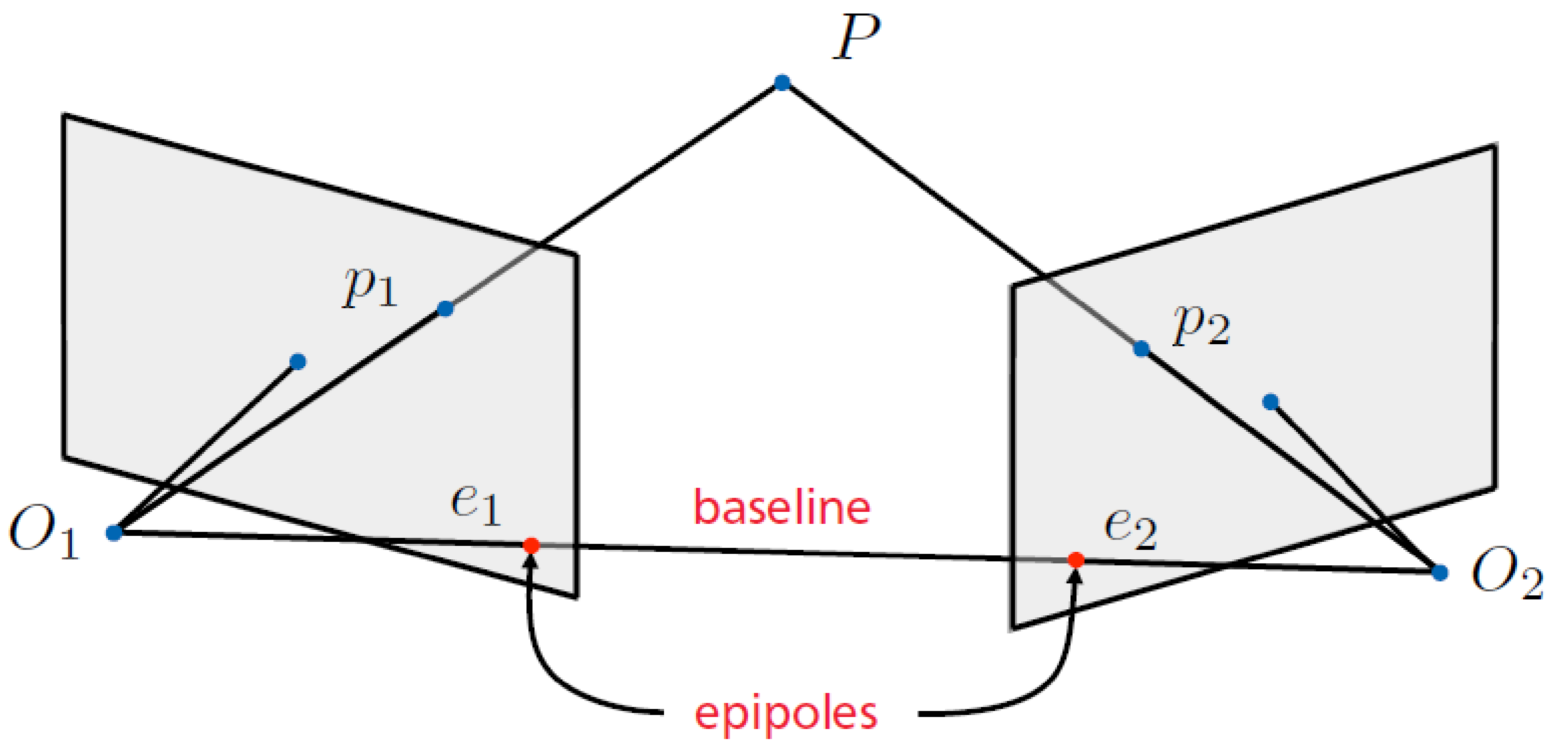
Epipolar Geometry



Epipolar Geometry

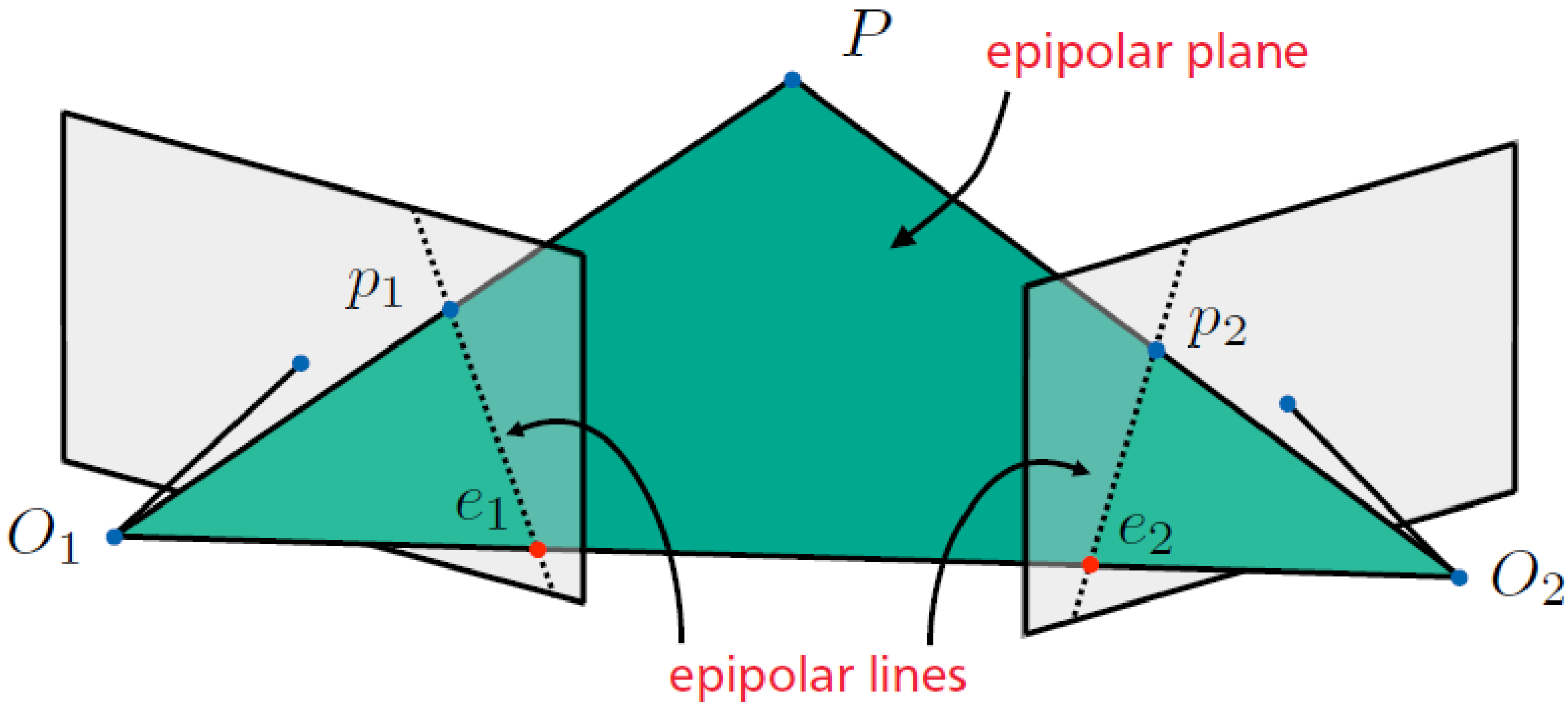


Epipolar Geometry



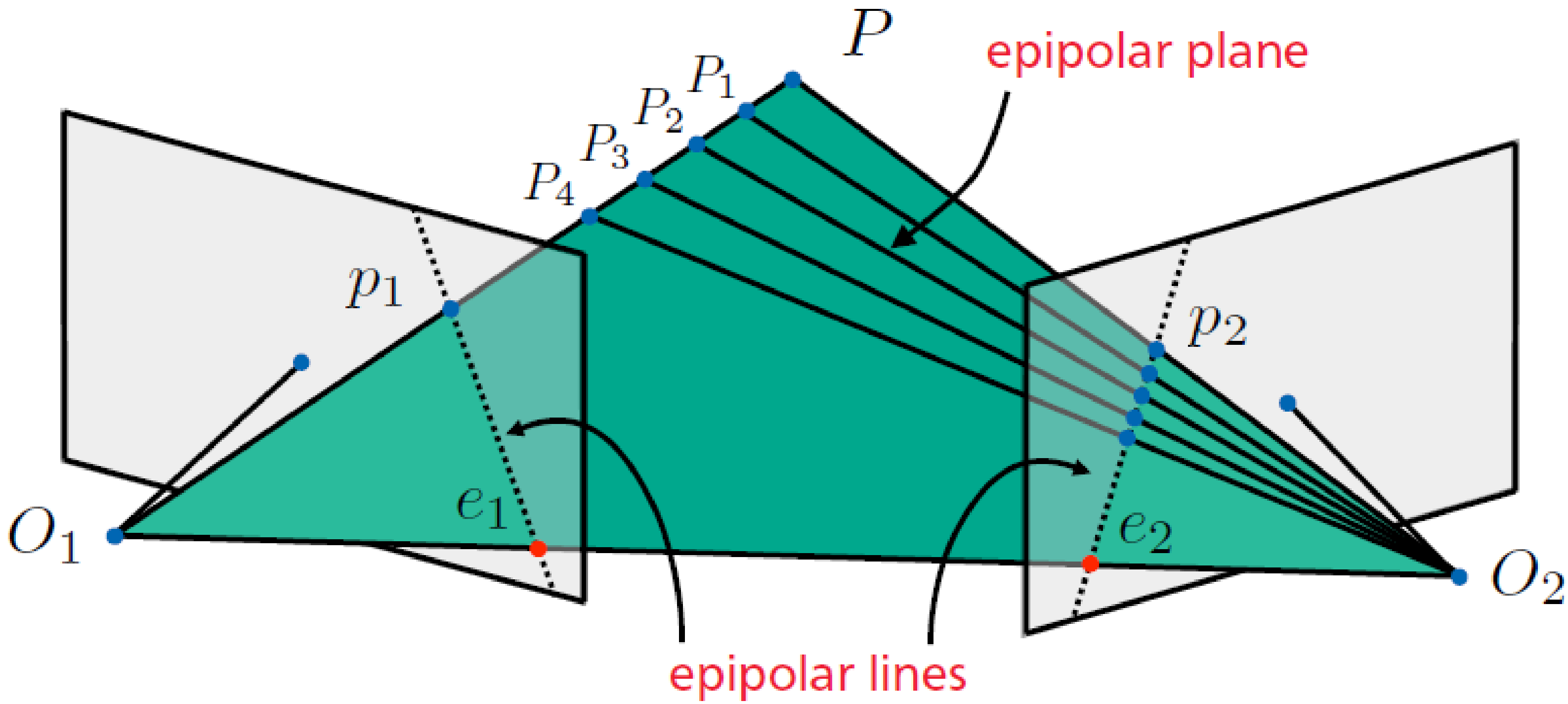
- Epipole: Image location of the optical center of the other camera.
  (Can be outside of the visible area)

Epipolar Geometry



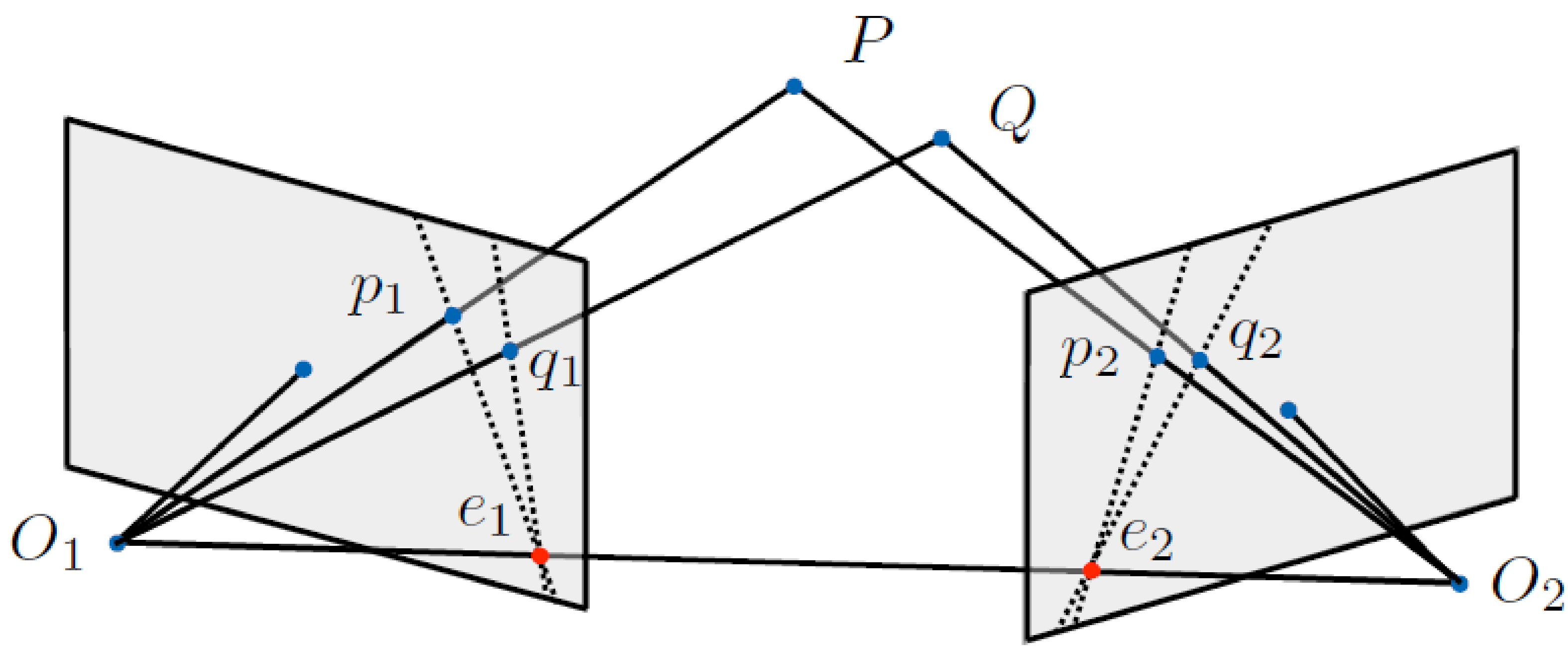
Epipolar plane: Plane through both camera centers and world point.

Epipolar Geometry



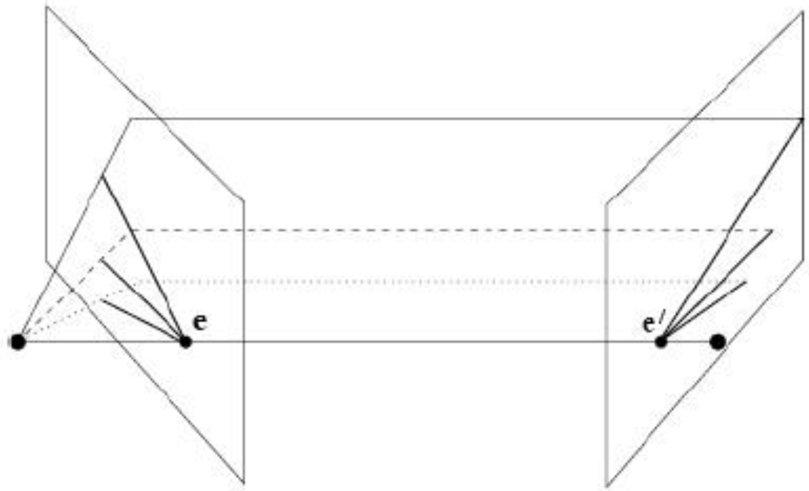
- Epipolar line: Constrains the location where a particular point (here p_1) from one view can be found in the other.

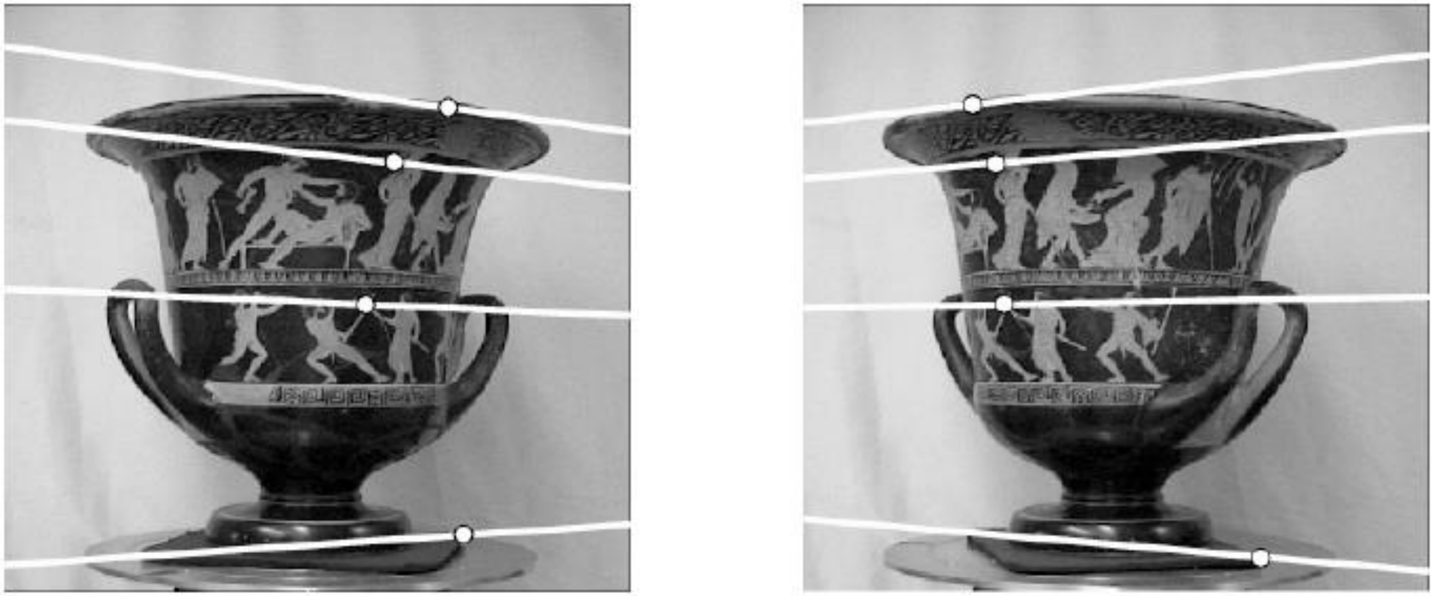
Epipolar Geometry



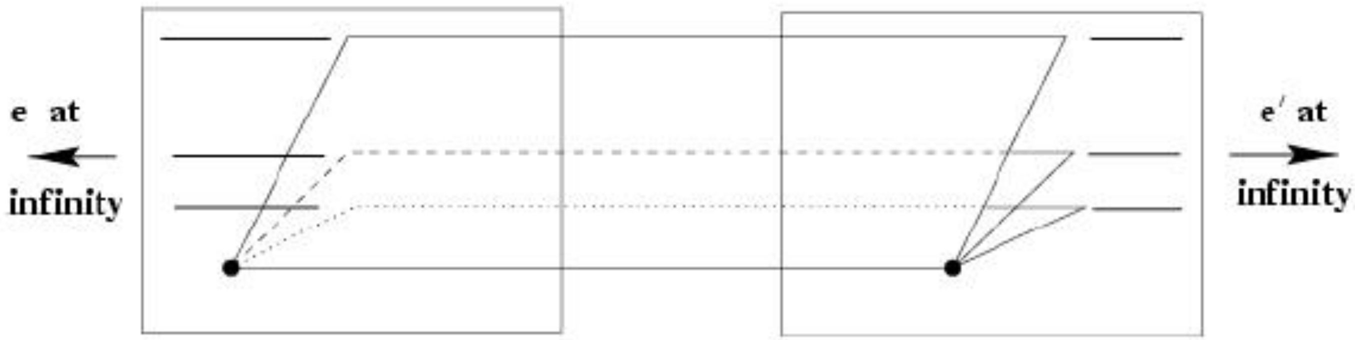
- Epipolar lines:
  - Intersect at the epipoles
  - In general not parallel

Example: Converging Cameras



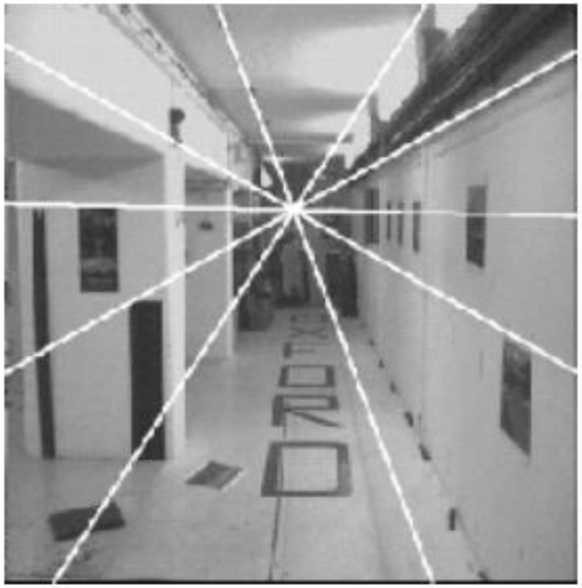


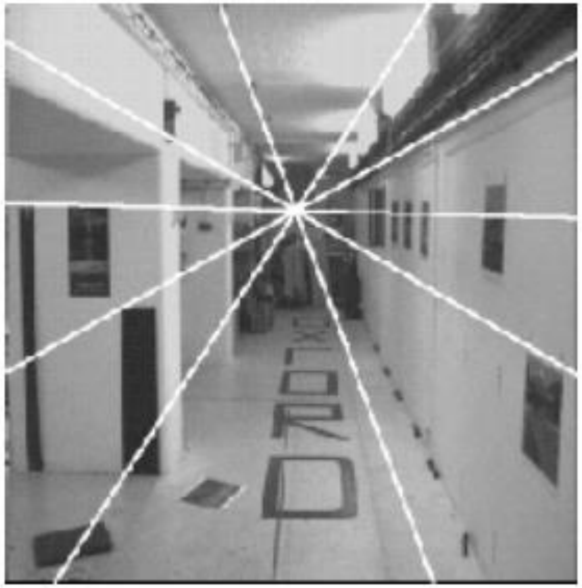
Example: Motion Parallel to Camera

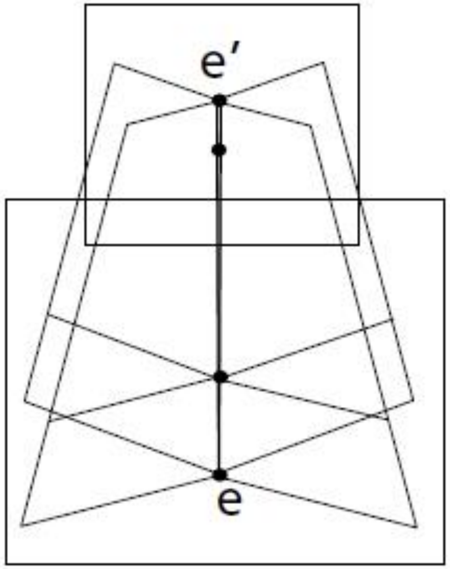


- We will use this idea when it comes to stereo matching

Example: Forward Motion



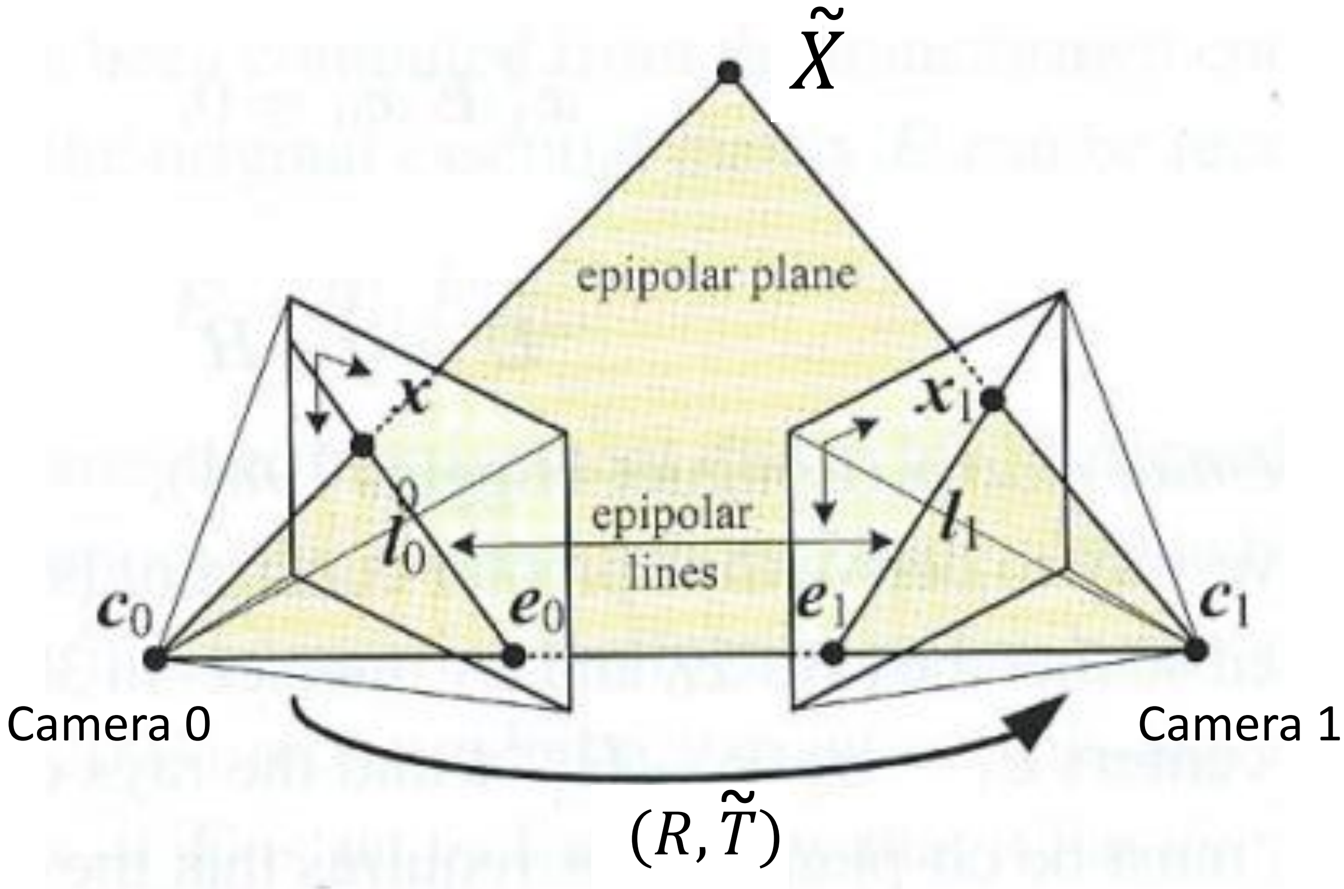




- Epipoles have same coordinate in both images
- Points move along lines radiating from epipole
  “focus of expansion”

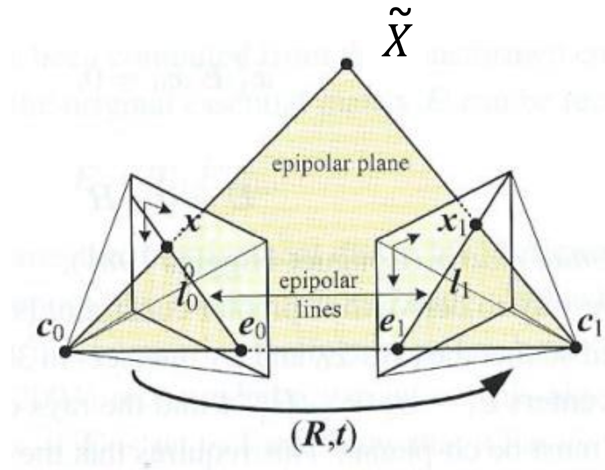
The maths behind it: Fundamental/Essential Matrix

derivation on black board



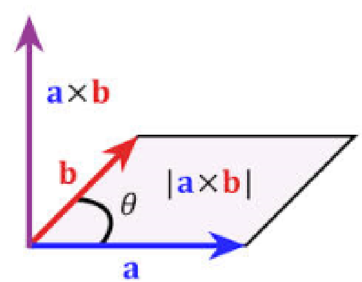
The maths behind it: Fundamental/Essential Matrix

The 3 vectors are in same plane (co-planar):
1) $\tilde{T}$ $(= \tilde{C}_1 - \tilde{C}_0)$
2) $\tilde{X} - \tilde{C}_0$
3) $\tilde{X} - \tilde{C}_1$



Set camera matrix: $x_0 = K_0[I|0]\,X$ and $x_1 = K_1 R^{-1}[I|-\tilde{C}_1]\,X$
Hence, $\tilde{C}_0 = 0$; $K_0^{-1}x_0 = \tilde{X}$; $RK_1^{-1}x_1 + \tilde{C}_1 = \tilde{X}$ (note $X = (\tilde{X}, 1)^T$)

The three vectors can be re-writting using x_0, x_1:
1) $\tilde{T}$
2) $\tilde{X} - \tilde{C}_0 = \tilde{X} = K_0^{-1}x_0$
3) $\tilde{X} - \tilde{C}_1 = RK_1^{-1}x_1 + \tilde{C}_1 - \tilde{C}_1 = RK_1^{-1}x_1$



We know that:
$(K_0^{-1}x_0)^T[\tilde{T}]_\times RK_1^{-1}x_1 = 0$ which gives: $x_0^T K_0^{-T}[\tilde{T}]_\times RK_1^{-1}x_1 = 0$

The Maths behind it: Fundamental/Essential Matrix

- In an un-calibrated setting (K's not known):

$$x_0^T K_0^{-T} [\tilde{T}]_\times R K_1^{-1} x_1 = 0$$

- In short: $x_0^T F x_1 = 0$ where F is called the Fundamental Matrix (discovered by Faugeras and Luong 1992, Hartley 1992)

- In an calibrated setting (K's are known):

we use rays: $x_i = K_i^{-1} x_i$

then we get: $x_0^T [\tilde{T}]_\times R x_1 = 0$

In short: $x_0^T E x_1 = 0$ where E is called the Essential Matrix

(discovered by Longuet-Higgins 1981)

Halfway Slide

1 Min Break

Fundamental Matrix: Properties

- We have $x_0^T F x_1 = 0$ where F is called the Fundamental Matrix

- It is det F = 0. Hence F has 7 DoF

Proof: $F = K_0^{-T} [\tilde{T}]_\times R K_1^{-1}$

F has Rank 2 since $[\tilde{T}]_\times$ has Rank 2 (see also last lecture)

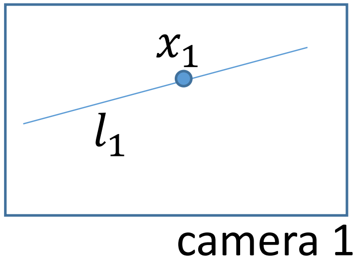
$$[\boldsymbol{x}]_\times = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}$$

Check: det$([x]_\times) = x_3(x_3 0 - x_1 x_2) + x_2(x_1 x_3 + x_2 0) = 0$

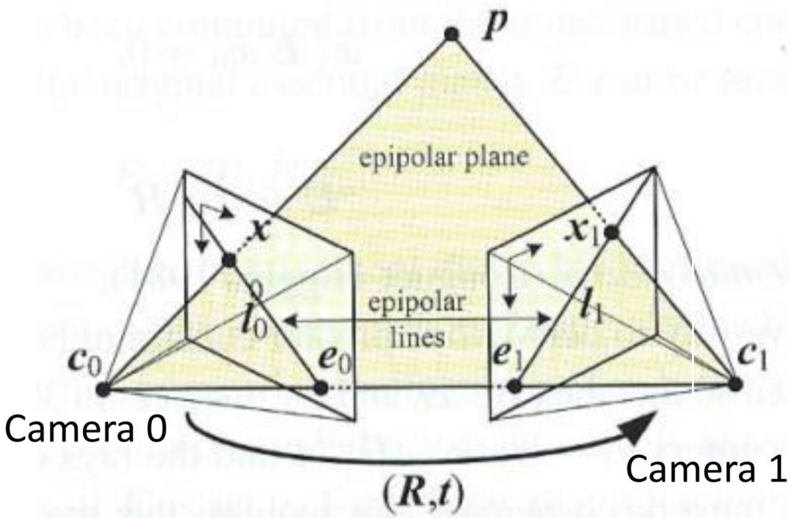
Fundamental Matrix: Properties

- For any two matching points (i.e. they have the same 3D point) we have: $x_0^T F x_1 = 0$

- Compute epipolar line in camera 1 of a point x_0:
  $l_1^T = x_0^T F$ (since $l_1^T x_1 = x_0^T F x_1 = 0$)



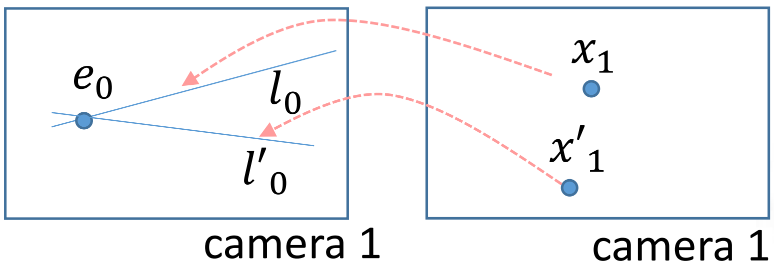
- Compute epipolar line in camera 0 of a point x_1:
  $l_0 = F x_1$ (since $x_0^T l_0 = x_0^T F x_1 = 0$)



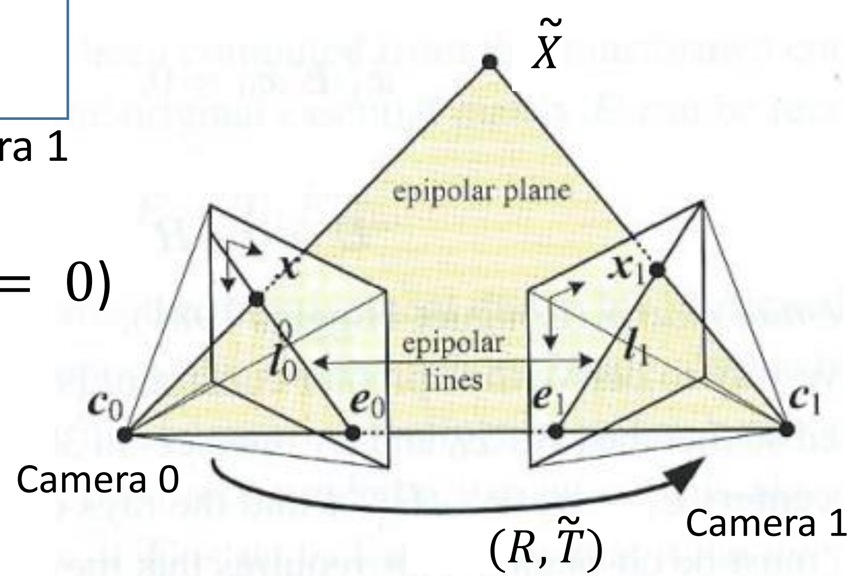
Fundamental Matrix: Properties

- For any two matching points (i.e. have the same 3D point) we have: $x_0^T F x_1 = 0$

- Compute e_0 with $e_0^T F = \mathbf{0}^T$ (i.e. left nullspace of F; can be computed with SVD)
  This is the Epipole e_0. It is: $e_0^T F x_1 = \mathbf{0}^T x_1 = 0$ for all points x_1. Hence all lines l_0 for any x_1: $l_0 = F x_1$ go through e_0^T.



- Epipole e_1 is right null space of F ($F e_1 = 0$)



How can we compute F (2-view calibration) ?

- Each pair of matching points gives one linear constraint $x^T F x' = 0$ in F. For x, x' we get:

$$\begin{bmatrix} x_1x_1' & x_1x_2' & x_1x_3' & x_2x_1' & x_2x_2' & x_2x_3' & x_3x_1' & x_3x_2' & x_3x_3' \\ & & & & \vdots & & & & \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

(here $x = (x_1, x_2, x_3)^T$)

- Given $m \geq 8$ matching points (x', x) we can compute the F in a simple way.

How can we compute F (2-view calibration) ?

Method (normalized 8-point algorithm):

1) Take $m \geq 8$ points
2) Compute T, and condition points: $x = Tx$; $x' = T'x'$
3) Assemble A with $Af = 0$, here A is of size $m \times 9$, and f vectorized F
4) Compute $f^* = argmin_f \|Af\|$ subject to $\|f\| = 1$.

   Use SVD to do this.
5) Get F of unconditioned points: $T^T F T'$ (note: $(Tx)^T F\, T'x' = 0$)
4) Make $rank(F) = 2$

$$s = \max_i(\|x_i\|)$$
$$t = \text{mean}(x_i)$$

$$T = \begin{bmatrix} \frac{1}{s} & 0 & -\frac{t_x}{s} \\ 0 & \frac{1}{s} & -\frac{t_y}{s} \\ 0 & 0 & 1 \end{bmatrix}$$

$$u = Tx$$

[See HZ page 282]

How to make F Rank 2

- (Again) Use SVD:

$$\boldsymbol{A} = \left[\begin{array}{c|c|c} u_0 & \cdots & u_{p-1} \end{array} \right] \begin{bmatrix} \sigma_0 & & \\ & \ddots & \\ & & \sigma_{p-1} \end{bmatrix} \left[\begin{array}{c} v_0^T \\ \hline \cdots \\ \hline v_{p-1}^T \end{array} \right]$$

Set last singular value σ_{p-1} to 0 then A has Rank $p-1$ and not p (assuming A has originally full Rank)

Proof: diagonal matrix has Rank $p-1$ hence A has Rank $p-1$

Can we compute F with just 7 points?

Method (7-point algorithm):

1) Take $m = 7$ points

2) Assemble A with $Af = 0$, here A is of size 7×9, and f vectorized F

3) Compute 2D right null space: F_1 and F_2 from last two rows in V^T
(use the SVD decomposition: $A = UDV^T$)

4) Choose: $F = \alpha F_1 + (1 - \alpha) F_2$ (see comments next slide)

5) Determine $\alpha's$ (either 1 or 3 solutions for α) by using the constraint:
det$(\alpha F_1 + (1 - \alpha) F_2)$ = 0.(see comments next slide)

- Note an 8th point would determine which of these 3 solutions is the correct one.
- We will see later that the 7-point algorithm is the best choice for the robust case.

Comments to previous slide

Step 4) Choose: $F = \alpha F_1 + (1 - \alpha) F_2$

- The full null-space is given by: $F = \alpha F_1 + \beta F_2$. We can say that some norm of F has a fixed value.
- We are free to say that we want: $\|\alpha F_1 + \beta F_2\| \geq 1$ (here F_1, F_2 are in vectorised form. Note that this is the same as having F_1, F_2 in matrix form and using the Frobenius norm for matrices)
- It is: $\alpha\|F_1\| + \beta\|F_2\| = \|\alpha F_1\| + \|\beta F_2\| \geq \|\alpha F_1 + \beta F_2\|$ (triangulation inequality)
- Hence we want: $\alpha\|F_1\| + \beta\|F_2\| \geq 1$
- Hence we want: $\alpha + \beta \geq 1$ (since F_1, F_2 are rows in V^T)
- Hence we can choose: $\beta = 1 - \alpha$

Comments to previous slide

Step 5) Compute $\det(\alpha F_1 + (1-\alpha)F_2) = 0$

$$\left| \alpha \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} + (1-\alpha) \begin{pmatrix} a' & b' & c' \\ d' & e' & f' \\ g' & h' & i' \end{pmatrix} \right| = \begin{vmatrix} \alpha a + (1-\alpha)a' & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{vmatrix}$$

$$= (\alpha a + (1-\alpha)a') \begin{vmatrix} \alpha e + (1-\alpha)e' & \alpha f + (1-\alpha)f' \\ \alpha h + (1-\alpha)h' & \alpha i + (1-\alpha)i' \end{vmatrix} + \text{"similar terms"}$$

$$= (\alpha a + (1-\alpha)a')(\alpha e + (1-\alpha)e')(\alpha i + (1-\alpha)i) + \text{"similar terms"}$$

$$= k\alpha^3 + l\alpha^2 + m\alpha + n$$

(This is a cubic polynomial equation for α which has one or three real-value solutions for α)

Can we get K's, $R, \tilde{T}$ from F?

- Assume we have $F = x_0^T K_0^{-T} [\tilde{T}]_\times R K_1^{-1}$

  Can we get out $K_1, R, K_0, \tilde{T}$?

- F has 7 DoF

- K_1, R, K_0, T have together 16 DoF

- Not directly possible. Only with assumptions such as:
  - External constraints
  - Camera does not change over several frames

  (This is an challanging topic (more than 10 years of research!) called auto-calibration or self-calibration. We look at it in detail in next lecture.)

Coming back to Essential Matrix

- In a calibrated setting (K's are known):

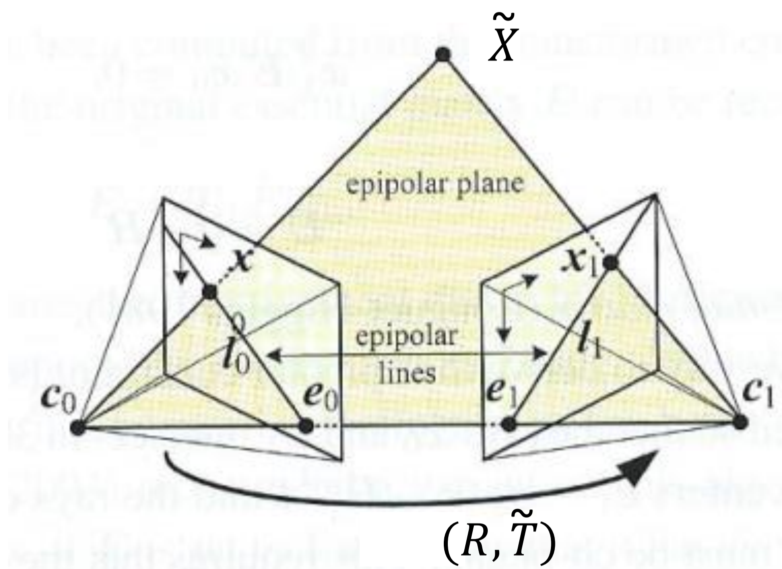
  we use rays: $x_i = K_i^{-1} x_i$

  then we get: $x_0^T [\tilde{T}]_\times R x_1 = 0$

  In short: $x_0^T E x_1 = 0$ where E is called the Essential Matrix

- E has 5 DoF, since $\tilde{T}$ has 3DoF, R 3DoF (note overall scale of $\tilde{T}$ is unknown)

- E has also Rank 2



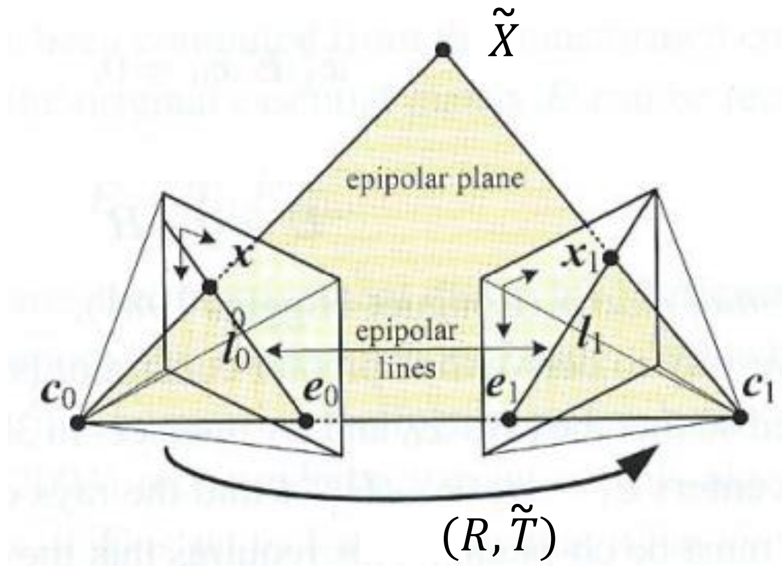
How to compute E

- We have: $x_0^T E x_1 = 0$
- Given $m \geq 8$ matching run 8-point algorithm (as for F)
- Given $m = 7$ run 7-point algorithm and get 1 or 3 solutions
- Given $m = 5$ run 5-point algorithm to get up to 10 solutions. This is the minimal case since E has 5 DoF.

- <u>5-point algorithm history:</u>
  - Kruppa, “Zur Ermittlung eines Objektes aus zwei Perspektiven mit innere Orientierung,” Sitz.-Ber. Akad. Wiss., Wien, Math.-Naturw. Kl., Abt. IIa, (122):1939-1948, 1913.
    found 11 solutions
  - M. Demazure, “Sur deux problemes de reconstruction,” Technical Report Rep. 882, INRIA, Les Chesnay, France, 1988
    showed that only 10 valid solutions exist
  - D. Nister, “An Efficient Solution to the Five-Point Relative Pose Problem,” IEEE Conference on Computer Vision and Pattern Recognition, Volume 2, pp. 195-202, 2003
    fast method which gives out 10 solutions of a 10 degree polynomial

Can we get $R, \tilde{T}$ from E?

- Assume we have $E = [\tilde{T}]_\times R$, can we get out $R, \tilde{T}$?
- E has 5 DoF
- $R, \tilde{T}$ have together 6 DoF
- Yes: We can get $\tilde{T}$ up to scale, and a unique R



How to get a unique $\tilde{T}, R$?

1) Compute $\tilde{T}$
   Note: E has rank 2, and $\tilde{T}$ is in the left nullspace of E since $\tilde{T}^t[\tilde{T}]_\times = (0,0,0)$
   This means that an SVD of E must look like:

$$E = UDV^T = [\boldsymbol{u}_0 \quad \boldsymbol{u}_1 \quad \tilde{T}] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_0^T \\ \boldsymbol{v}_1^T \\ \boldsymbol{v}_2^T \end{bmatrix}$$

   This fixes the norm of $\tilde{T}$ to 1, and correct sign $(+/-\tilde{T})$ is done in step 3

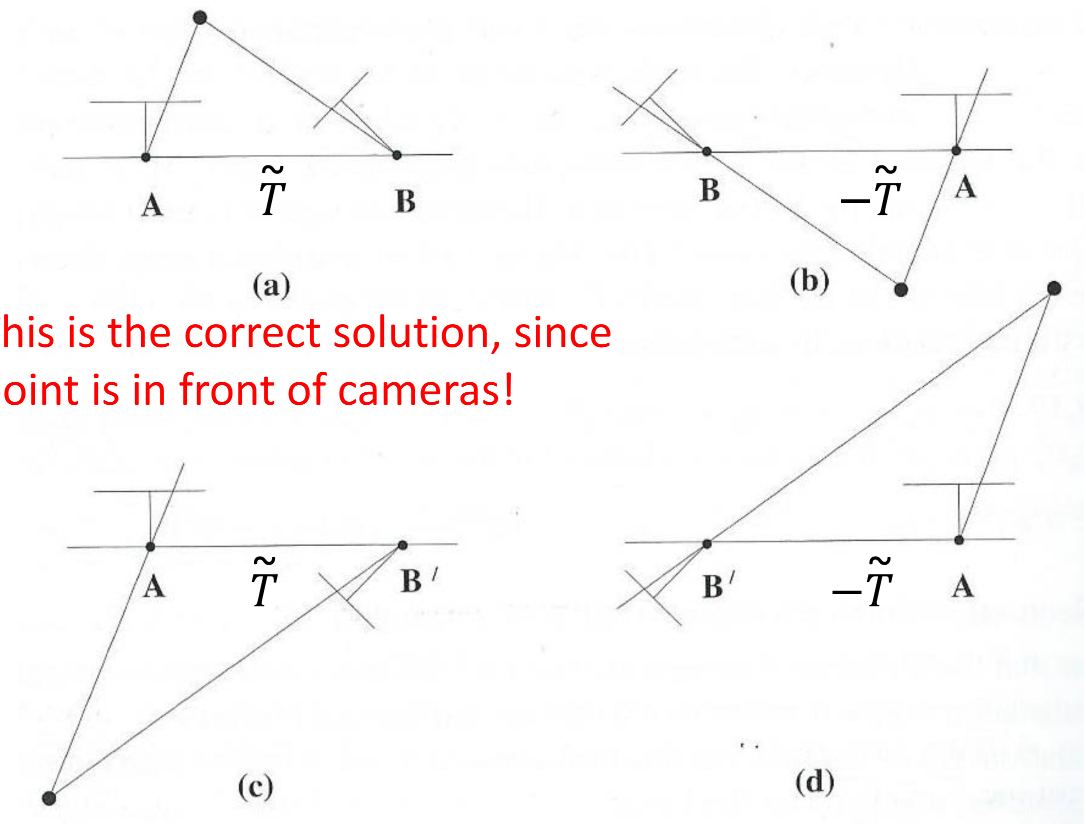
2) Compute 4 possible solutions for R

$R_{1,2} = +/- U R_{90^o}^T V^T$; $R_{3,4} = +/- U R_{-90^o}^T V^T$ (see derivation HZ page 259; Szeliski page 310)

where $E = UDV^T$, $R_{90} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$, $R_{-90} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

3) Derive the unique solution for R and sign for $\tilde{T}$:
   1) $det(R) = 1$
   2) Reconstruct a 3D point and choose the solution where it lies in front of the two cameras. (In robust case: Take solution where most (≥ 5) points lie in front of the cameras)

Visualization of the 4 solutions for $R, \tilde{T}$

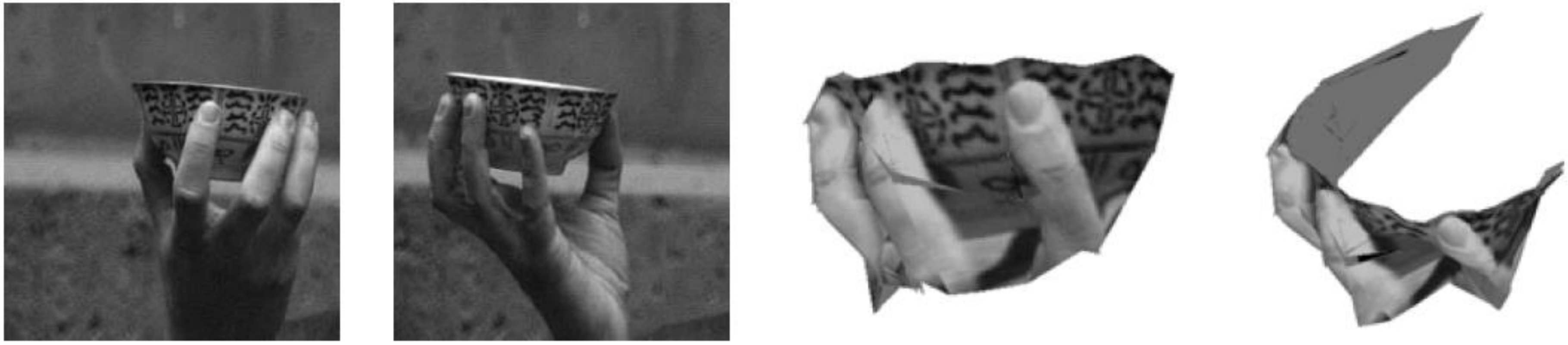


This is the correct solution, since point is in front of cameras!

The property that points must lie in front of the camera is known as Chirality (Hartley 1998)

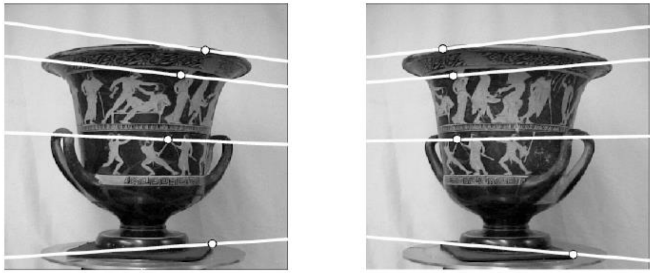
What can we do with F, E?

- F/E encode the geometry of 2 cameras
- Can be used to find matching points (dense or sparse) between two views (we use this a lot in later lecture on stereo matching!)
- F/E encodes the essential information to do 3D reconstruction



Fundamental and Essential Matrix: Summary

- Derive geometrically F, E :
  - F for un-calibrated cameras
  - E for calibrated cameras



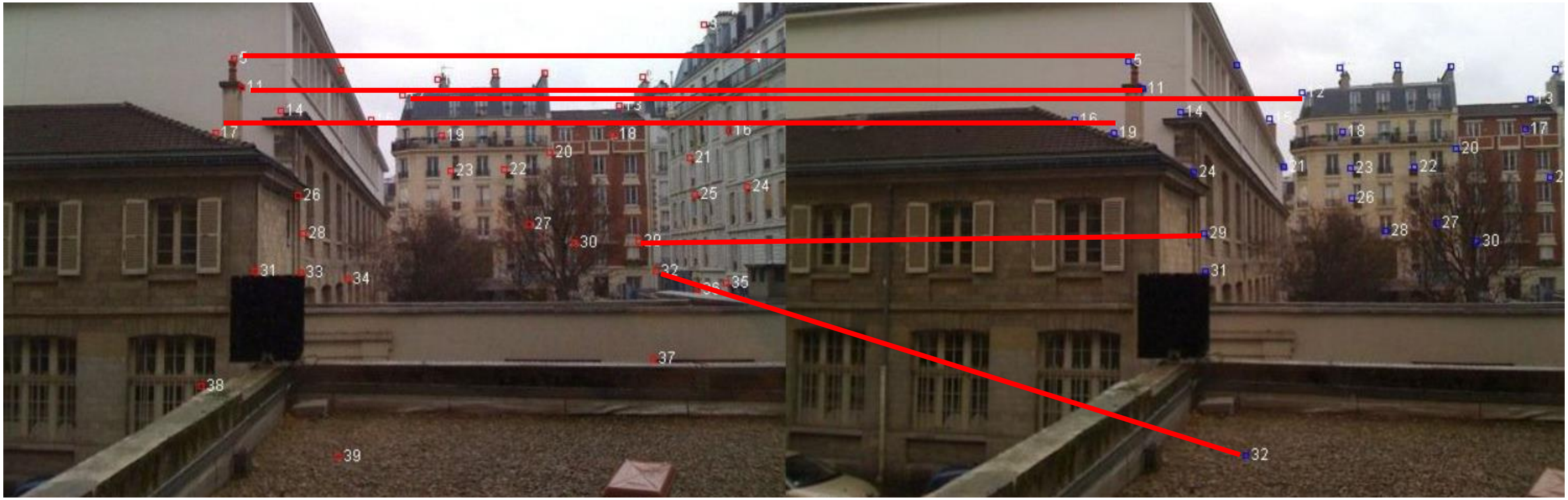
- Calibration: Take measurements (points) to compute F, E
  - F minimum of 7 points -> 1 or 3 real solutions.
  - F many points -> least square solution with SVD
  - E minimum of 5 points -> 10 solutions
  - E many points -> least square solution with SVD

- Can we derive the intrinsic (K) an extrinsic (R, T) parameters from F, E?
  -> F next lecture
  -> E yes can be done (translation up to scale)

- What can we do with F, E ?

  -> essential tool for 3D reconstruction

Motivation for next lecture



Question 1: If a match is completely wrong then $argmin_h \|A\boldsymbol{h}\|$ is a bad idea

Question 2: If a match is slightly wrong then $argmin_h \|A\boldsymbol{h}\|$ might not be perfect.
Better might be a geometric error: $argmin_h \|\boldsymbol{H}x - x'\|$