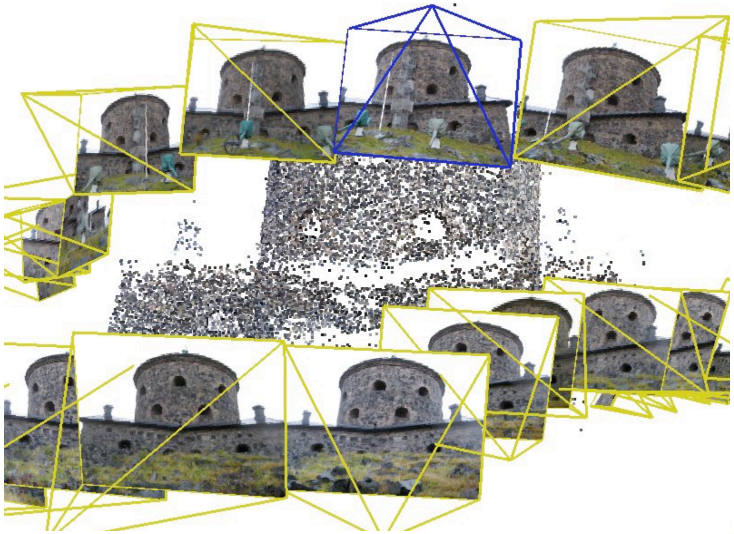
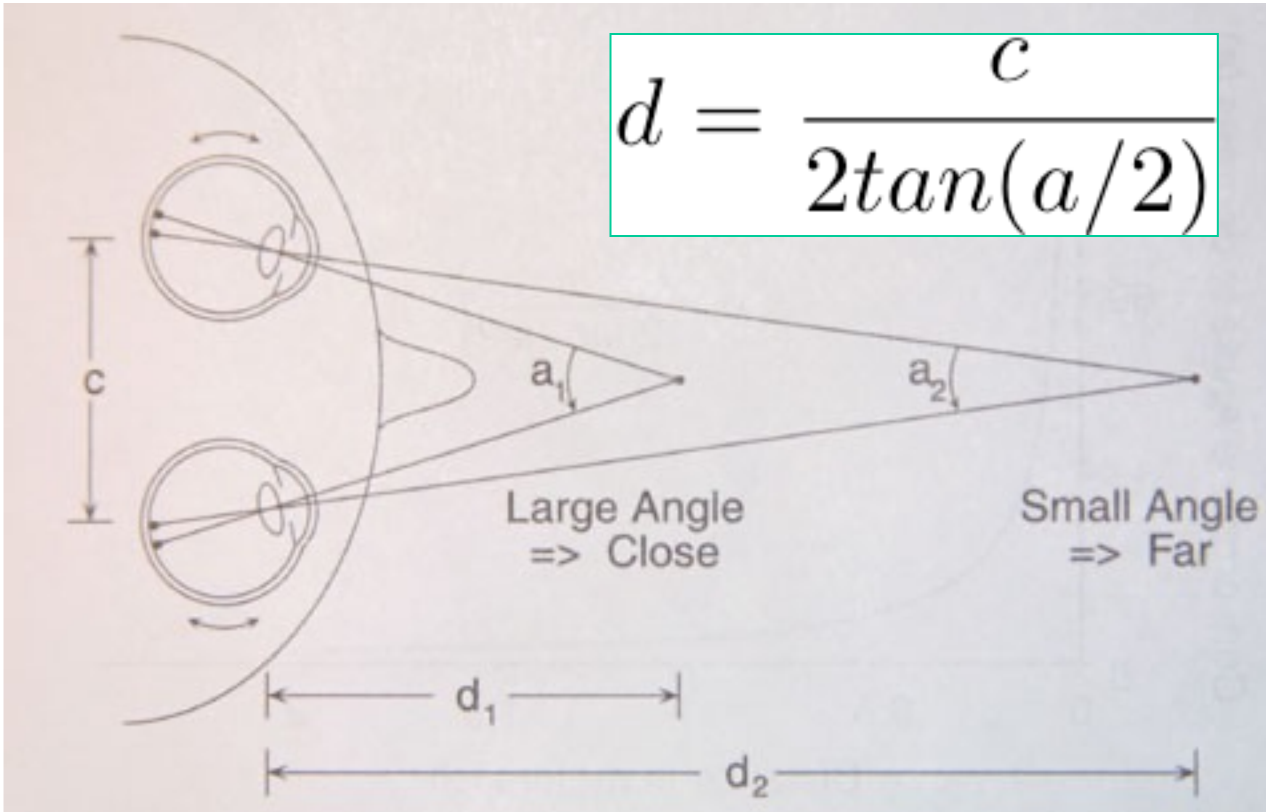
Multi-View Geometry



Slides from Yuri Boykov…with materials from H&Z and Carl Olsson

Motivation: **triangulation** gives depth

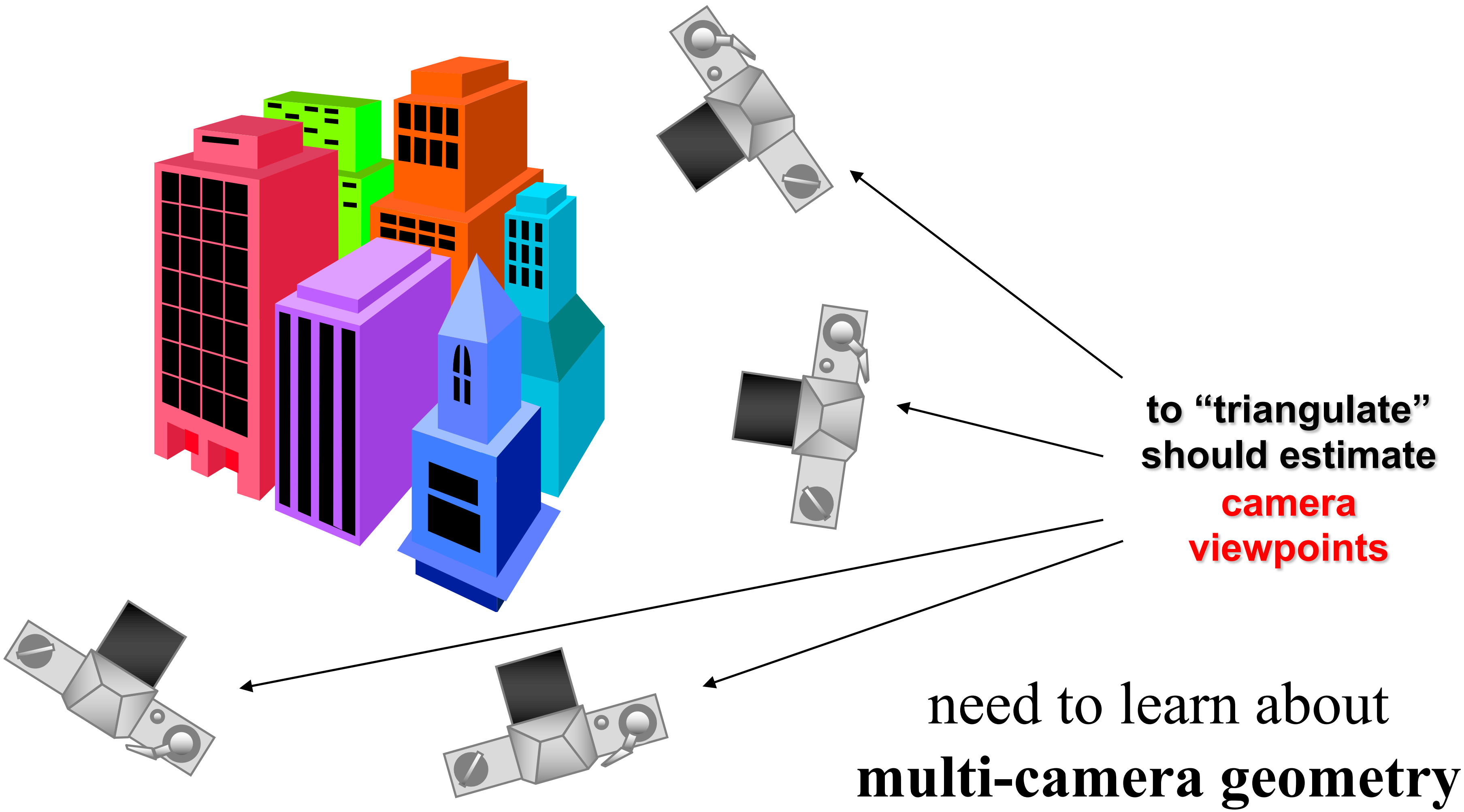


$$d = \frac{c}{2tan(a/2)}$$

Human performance: up to 6-8 feet

Motivation: reconstruction problems

Multi-view reconstruction: **shape from two or more images**



need to learn about

multi-camera geometry

Summary:

- Projective Camera Model
  - intrinsic and extrinsic parameters
  - projection matrix (a.k.a. camera matrix)
  - camera calibration (from known 3D points)
    - resection problem
    - estimating intrinsic/extrinsic parameters

- Two cameras (*epipolar* geometry)
  - essential and fundamental matrices: E and F
  - estimating E (from matched features)
  - computing projection matrices from E

- *Structure-from-Motion (SfM)* problem - quick overview
  - estimating "**motion**": camera positions (projection matrices)
  - estimating "**structure**": scene points in 3D space

at the same time (both are unknown)

Additional readings:

- Hartley and Zisserman *"Multiple View Geometry"*
  Cambridge University Press, Ed.2

- Heyden and Pollefeys *"Multiple View Geometry"*
  short course at CVPR 2001
  https://inf.ethz.ch/personal/marc.pollefeys/pubs/HeydenPollefeysCVPR01.pdf

Towards projective camera model

First, if there is only one camera, can use a **camera-centered 3D coordinate system** (x,y,z):

camera-centered coordinate system

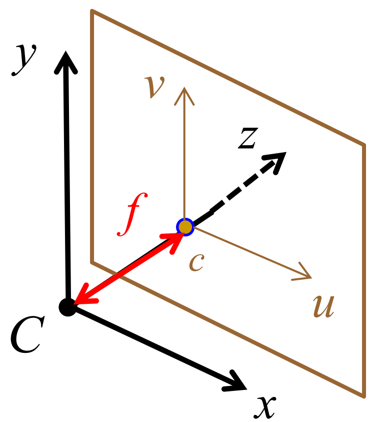


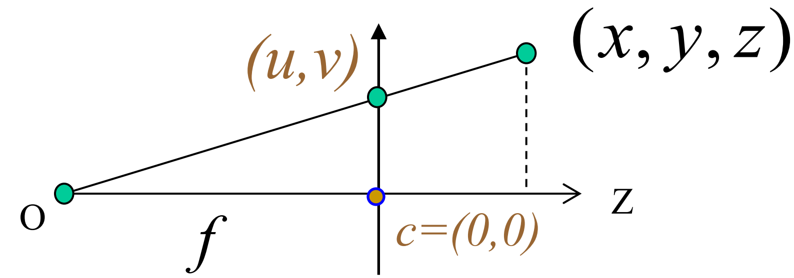
image coordinate system

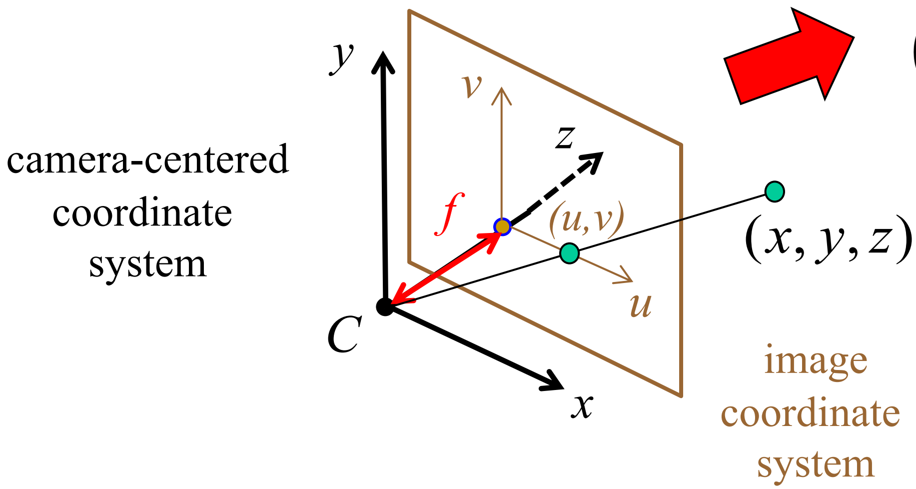
as seen in lecture 2

- optical center is point (0,0,0)
- x and y axis are parallel to the image plane
- x and y axis parallel to u and v axis of the image coordinate system
- optical axis (z) intersects image plane at image point $c = (0,0)$

Camera-centered coordinate system

For simplicity,
illustration below assumes
world point (x,y,z)
is inside x-z plane





$$(x, y, z) \rightarrow (\underbrace{f\frac{x}{z}}_{u}, \underbrace{f\frac{y}{z}}_{v})$$

image-based coordinates
of the **projection point**

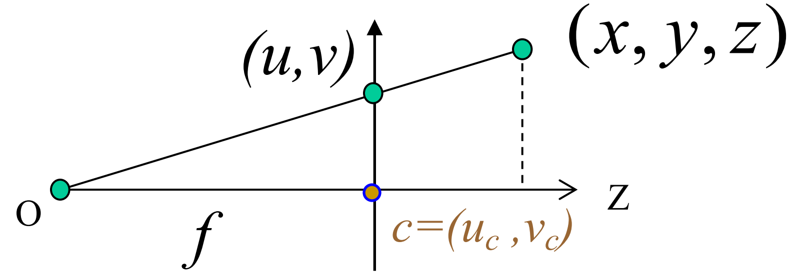
as seen in lecture 2

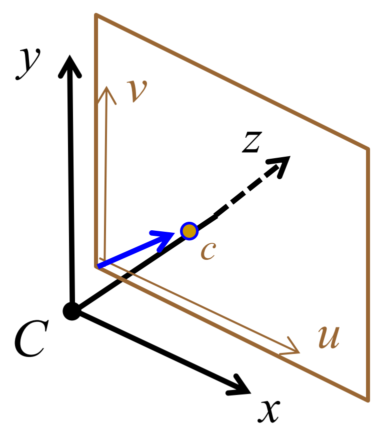
- optical center is point (0,0,0)
- x and y axis are parallel to the image plane
- x and y axis parallel to u and v axis of the image coordinate system
- optical axis (z) intersects image plane at image point $c = (0,0)$

Camera-centered coordinate system

In general, image coordinate center can be anywhere (often in image corner).

Thus, optical axis may intersect image plane at a point with image coordinates $c=(u_c, v_c)$ contributing **additional shift**





$$(x, y, z) \rightarrow \underbrace{\left(f\frac{x}{z} + u_c\right.}_{u}, \underbrace{\left.f\frac{y}{z} + v_c\right)}_{v}$$

image-based coordinates of the **projection point**

Camera-centered coordinate system

camera projection
can be represented as
matrix multiplication

$$(x, y, z) \rightarrow (\underbrace{f\frac{x}{z} + u_c}_{u}, \underbrace{f\frac{y}{z} + v_c}_{v})$$

u *v*
image-based coordinates of the **projection point**

using **homogeneous representation** for image points

$$\begin{bmatrix} wu \\ wv \\ w \end{bmatrix} = \underbrace{\begin{bmatrix} f & 0 & u_c \\ 0 & f & v_c \\ 0 & 0 & 1 \end{bmatrix}}_{K} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

K
matrix of intrinsic camera parameters

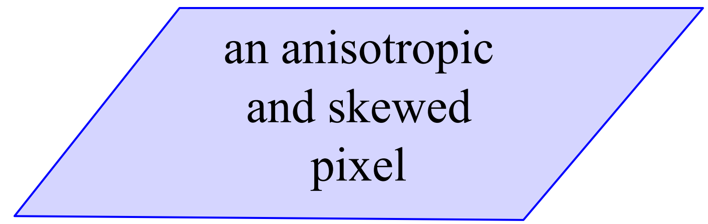
camera centered coordinates for 3D world points

NOTE: $w = z$ **(depth)**

Camera-centered coordinate system

Generally, **anisotropic** or **skewed** pixels result in

- different f_x and f_y
- skew coefficient s



s - skew/tilt

$\frac{f_x}{f_y}$ - aspect ratio

using **homogeneous representation** for image points

$$\begin{bmatrix} wu \\ wv \\ w \end{bmatrix} = \begin{bmatrix} f_x & s & u_c \\ 0 & f_y & v_c \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

K **matrix of intrinsic camera parameters**

camera centered coordinates for 3D world points

Camera-centered coordinate system

In general, matrix K of intrinsic camera parameters is 3x3 **upper triangular**. It has 5 degrees of freedom. For square pixels, K has 3 d.o.f.

using **homogeneous representation** for image points

$$\begin{bmatrix} wu \\ wv \\ w \end{bmatrix} = \underbrace{\begin{bmatrix} f_x & s & u_c \\ 0 & f_y & v_c \\ 0 & 0 & 1 \end{bmatrix}}_{K} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

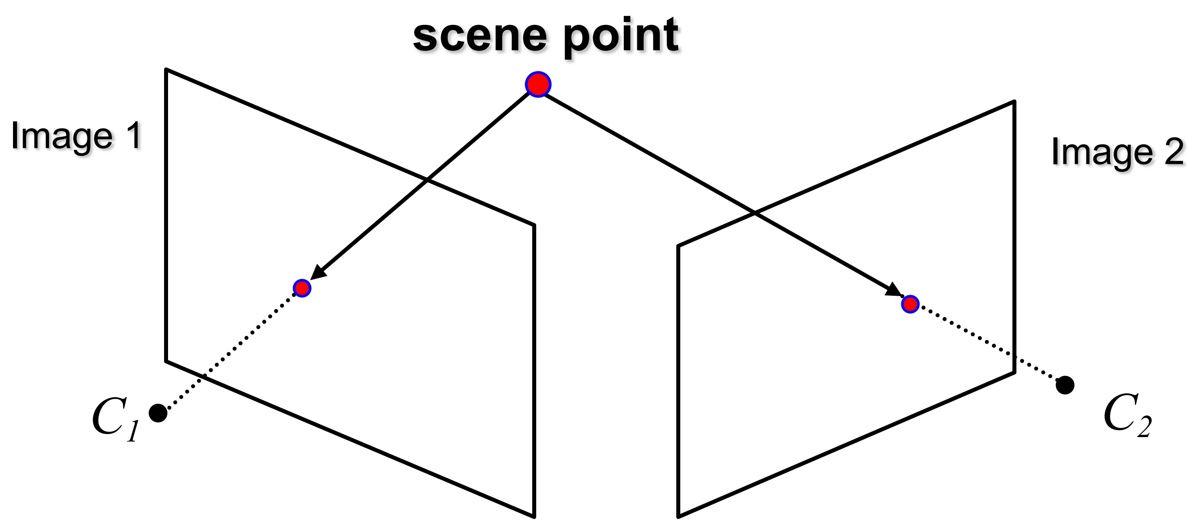
NOTE: here matrix K maps $\mathbb{R}^3$ to $\mathbb{R}^2$ ($\mathbb{P}^2$) (**not a homography** $\mathbb{P}^2 \rightarrow \mathbb{P}^2$)

K **matrix of intrinsic camera parameters**

camera centered coordinates for 3D world points

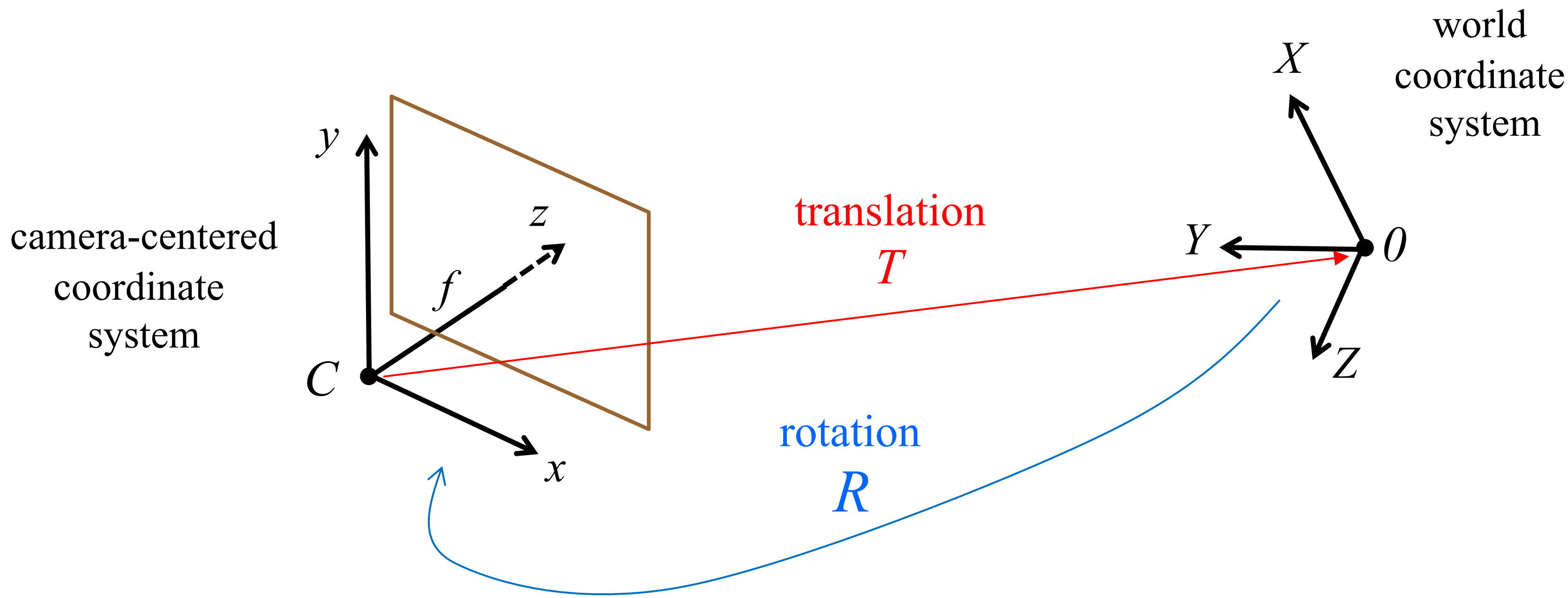
What if there are more than one camera?

Projecting 3D scene onto images with different view-points



Only one camera can serve for world coordinate system.
Other cameras will have their **camera-centered 3D coordinates different from the world coordinate system**.

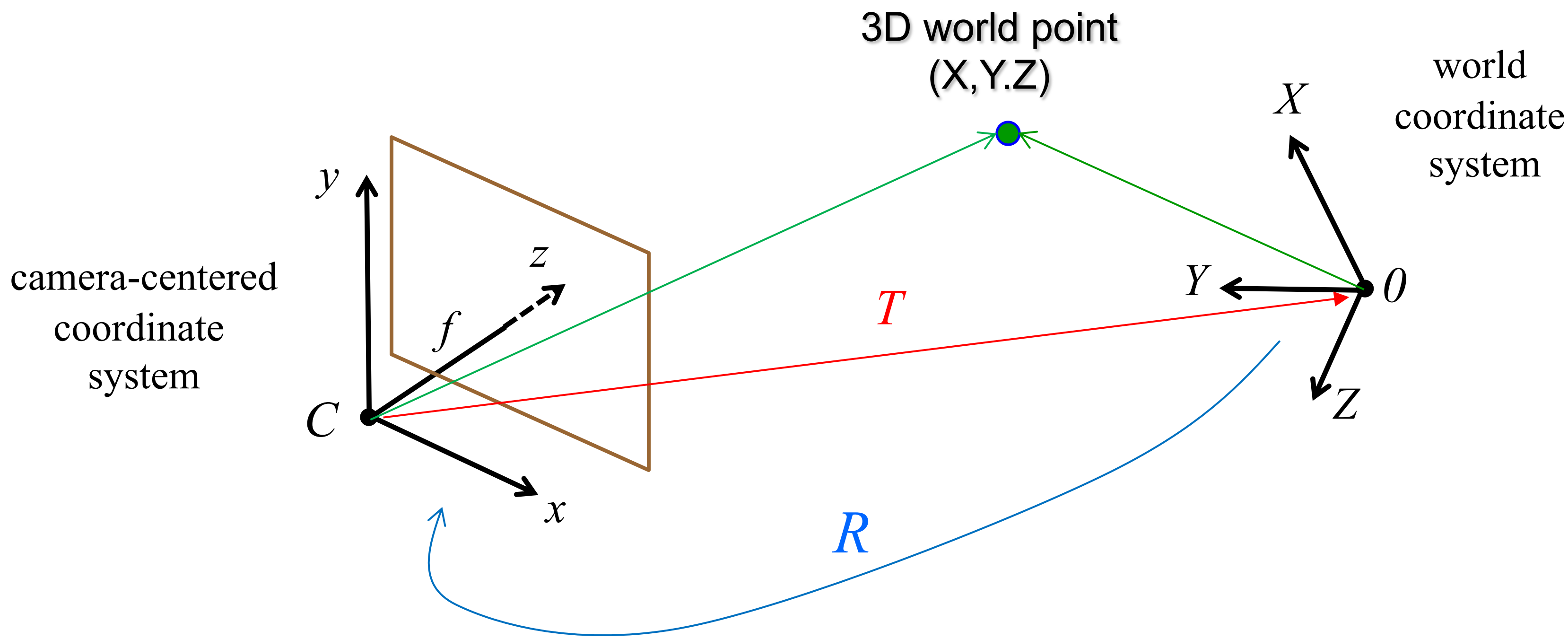
Camera projection matrix



<u>In case of two or more cameras, 3D world coordinate system maybe different from a camera-based coordinate system</u>:

- T is a (translation) vector defining relative position of camera's center
- orientation of x,y,z-axis of the camera-based coordinate system can be related to the axis of the world coordinate system via rotation matrix R

Camera projection matrix



Converting world coordinates of a point into camera-based 3D coordinate system

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + T$$

camera-based 3D coordinates — world 3D coordinates

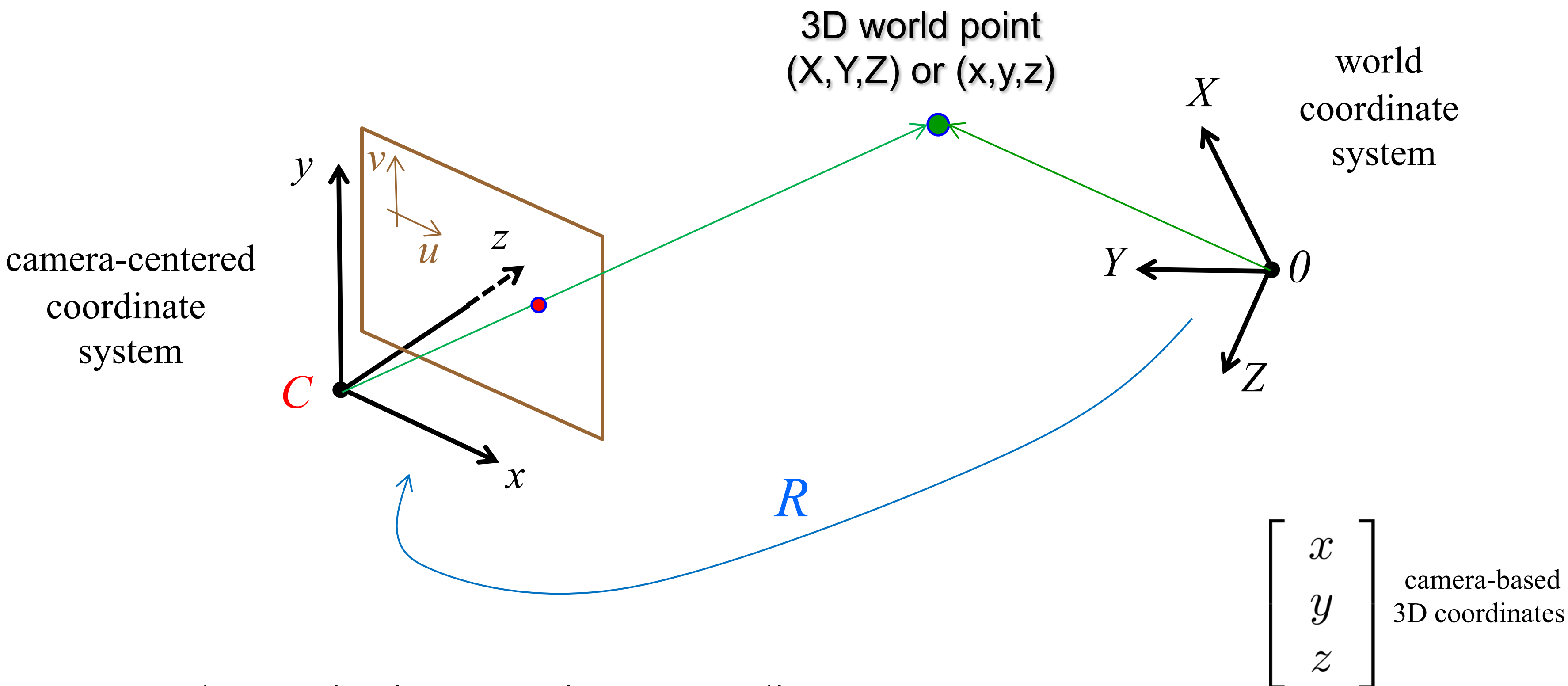
(here vector T is world's center in camera's coordinates)

using **homogeneous representation** for 3D points in world coordinate system

$$\underset{3\text{x}1}{\begin{bmatrix} x \\ y \\ z \end{bmatrix}} = \underbrace{\begin{bmatrix} R & \vert & T \end{bmatrix}}_{3\text{x}4} \cdot \underset{4\text{x}1}{\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}}$$

we get a **linear transformation (matrix multiplication)**

Camera projection matrix



Remember, projecting to 2D image coordinates…

$$\begin{bmatrix} wu \\ wv \\ w \end{bmatrix} = K \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad \Rightarrow \quad \begin{bmatrix} wu \\ wv \\ w \end{bmatrix} = K \cdot \begin{bmatrix} R & T \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

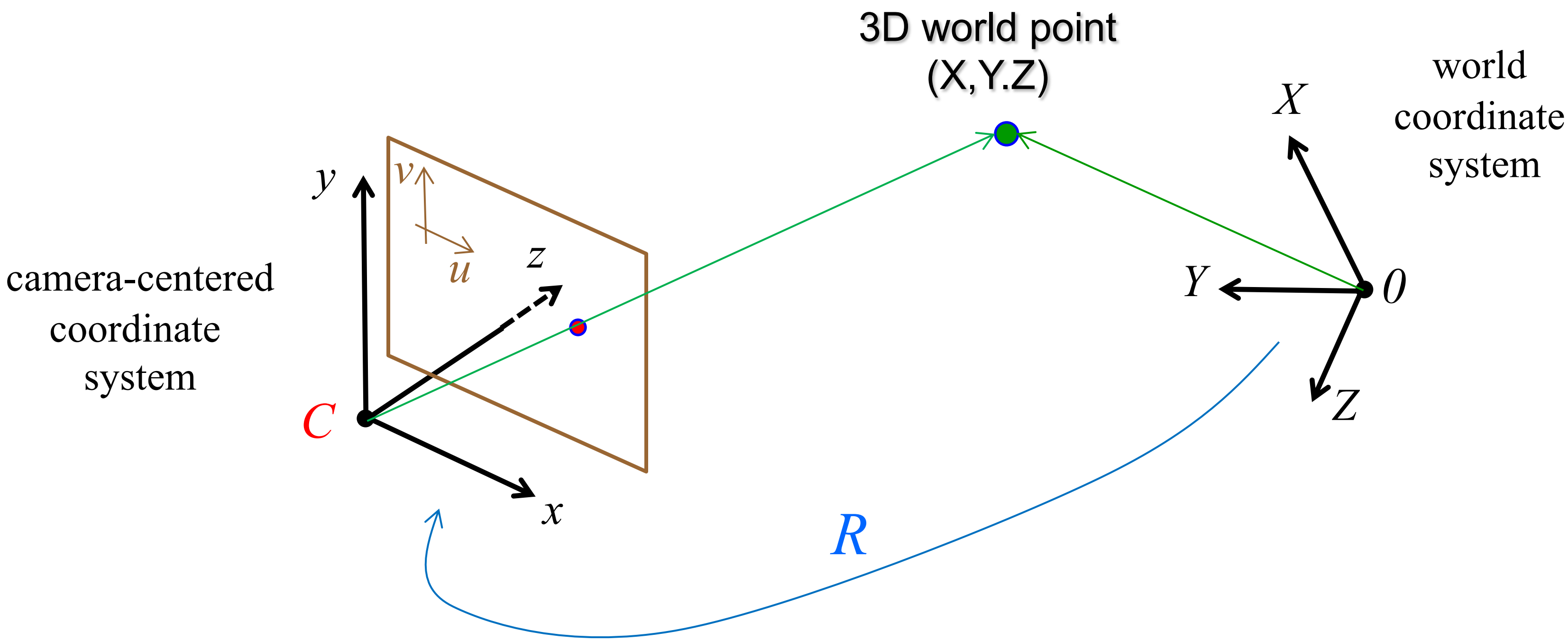
homogeneous image coordinates; camera-based 3D coordinates

5 d.o.f; 3 d.o.f; 3 d.o.f

3x3; 3x3; 3x4; 4x1

project **rotate** **translate**

Camera projection matrix



projection matrix P

$$\begin{bmatrix} wu \\ wv \\ w \end{bmatrix} = K \cdot \left[\begin{array}{c|c} R & T \end{array} \right] \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \iff \tilde{p} = P \cdot \tilde{X}$$

3x1 3x4 4x1

homogeneous 2D image coordinates $\tilde{p}$

intrinsic camera parameters

extrinsic camera parameters

homogeneous 3D world coordinates $\tilde{X}$

Homogeneous coordinates in 2D and 3D

Trick of adding one more coordinate

- translation becomes matrix multiplication
- 2D points become 3D rays

$$\text{in } \mathbb{R}^2 \quad (u,v) \Rightarrow \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \sim \begin{bmatrix} wu \\ wv \\ w \end{bmatrix} \text{ in } \mathbb{P}^2$$

homogeneous 2D image coordinates

$$\text{in } \mathbb{R}^3 \quad (X,Y,Z) \Rightarrow \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \sim \begin{bmatrix} wX \\ wY \\ wZ \\ w \end{bmatrix} \text{ in } \mathbb{P}^3$$

homogeneous 3D scene coordinates

Converting *from* homogeneous coordinates

$$\underset{\text{in } \mathbb{P}^2}{\begin{bmatrix} x \\ y \\ w \end{bmatrix}} \Rightarrow \underset{\text{in } \mathbb{R}^2}{\left(x/w,\ y/w\right)}$$

$$\underset{\text{in } \mathbb{P}^3}{\begin{bmatrix} X \\ Y \\ Z \\ w \end{bmatrix}} \Rightarrow \underset{\text{in } \mathbb{R}^3}{\left(X/w,\ Y/w,\ Z/w\right)}$$

Camera calibration

Goal: estimate intrinsic camera parameters

- focal length f, image center (u_c, v_c), other elements of **matrix *K***
- if needed, corrections for lens distortions (*radial distortion* in fish eye lenses)
not represented by K

Motivation:

- if K is known, only 6 *d.o.f* remains in projection matrix $P = K \cdot (R|T)$
(3 *d.o.f.* for each rotation R and translation T)

=> it becomes **easier to estimate projection matrices** corresponding to different viewpoints as camera(s) move around

- using *calibrated* camera(s) is a way to **remove projective ambiguity** in *structure from motion* 3D reconstruction (*more later*)

Camera calibration

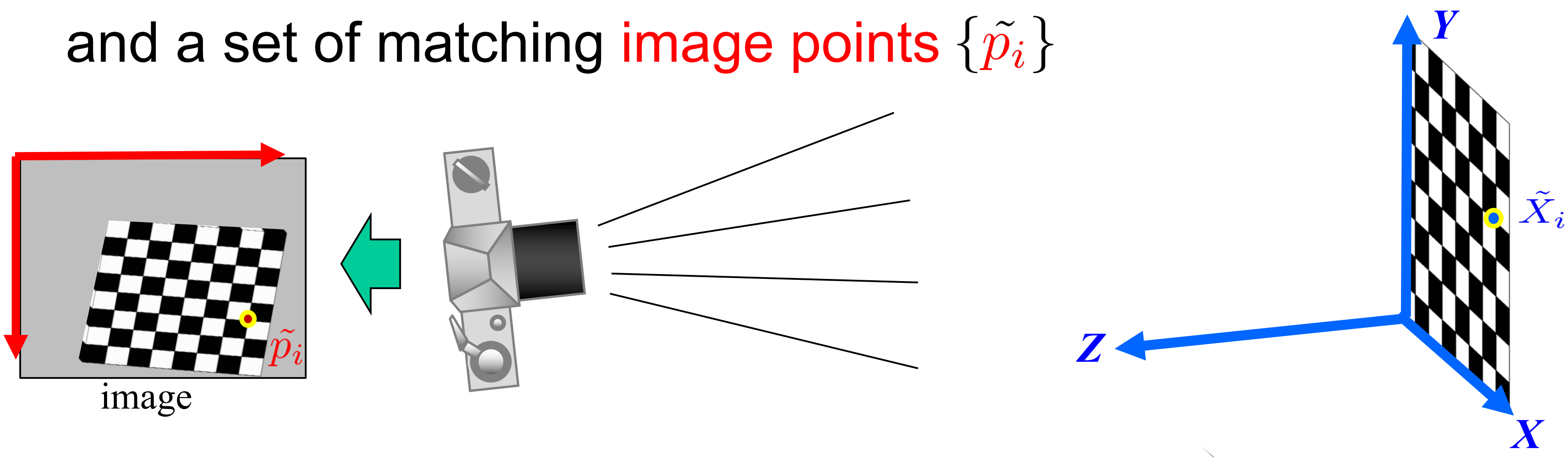
Basic calibration technique:

assume a set of 3D points $\{\tilde{X}_i\}$

with known world coordinates

and a set of matching image points $\{\tilde{p}_i\}$

calibration pattern
and tied 3D coordinates



$\tilde{X}_i \leftrightarrow \tilde{p}_i$

- find camera matrix P from known matches (**resection problem**)
- then, find intrinsic and extrinsic parameters (use **matrix factorization**)

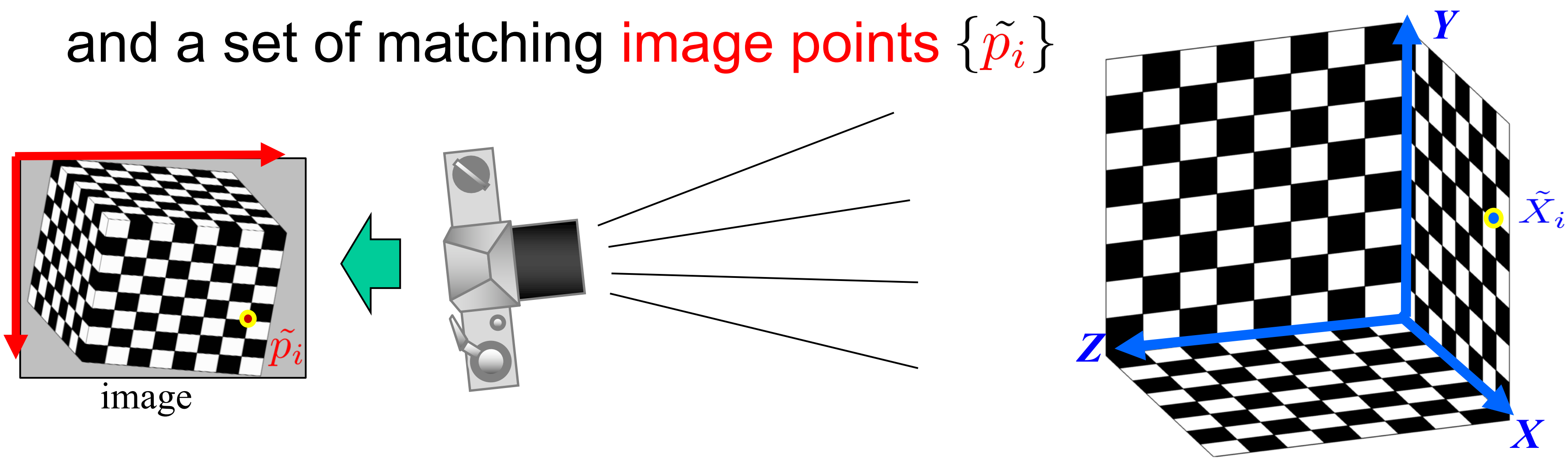
Camera calibration

NOTE: should not use 3D points $\{\tilde{X}_i\}$ on a single plane
("degenerate configurations", see H&Z Sec 7.1)

Basic calibration technique:

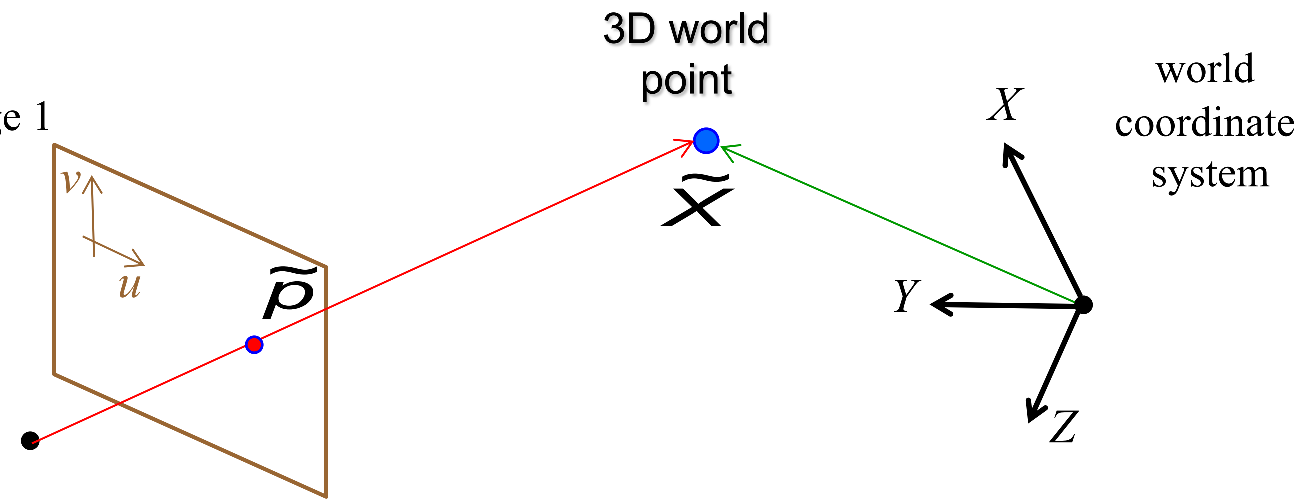
assume a set of 3D points $\{\tilde{X}_i\}$
with known world coordinates
and a set of matching image points $\{\tilde{p}_i\}$

calibration rig (*Tsai grid*)



- find camera matrix P from known matches $\tilde{X}_i \leftrightarrow \tilde{p}_i$
  (**resection problem**)
- then, find intrinsic and extrinsic parameters
  (use **matrix factorization**)

Camera projection matrix (estimating from $\color{blue}{\tilde{X}_i} \leftrightarrow \color{red}{\tilde{p}_i}$)



P has 12 entries, 11 d.o.f.

Q: How many matched pairs $\color{blue}{\tilde{X}_i} \leftrightarrow \color{red}{\tilde{p}_i}$ are needed ? **A:** 5.5 ☺

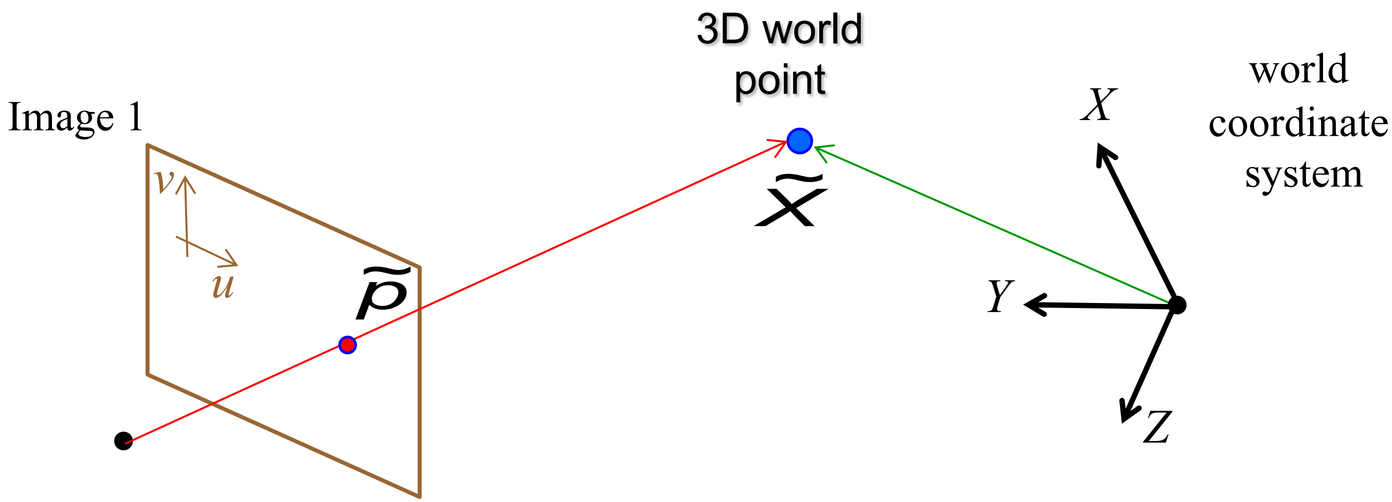
$$\begin{bmatrix} wu \\ wv \\ w \end{bmatrix} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & g & k & l \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

estimate unknown projection matrix P

Q: Solving for $a, b, \ldots, k, l$?
A: similar to estimating homographies
(see Topic 3, or H&Z p.179)

(resection problem)

Camera projection matrix (estimating from $\tilde{X}_i \leftrightarrow \tilde{p}_i$)



$$\begin{bmatrix} wu \\ wv \\ w \end{bmatrix} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & g & k & l \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

estimate unknown projection matrix P

(resection problem)

- Use more than 6 matched pairs

$$\tilde{X}_i \leftrightarrow \tilde{p}_i$$

to compensate for errors (*homogeneous least squares*)

Extracting intrinsic parameters from P

Now, assume that 3x4 projection matrix P is already estimated

$$P = \underbrace{\begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & g & k & l \end{bmatrix}}_{\text{known}} = \underbrace{\overset{3\times3}{K} \cdot \overbrace{\left[\begin{array}{c|c} R & T \end{array}\right]}^{3\times4}}_{\text{unknown}}$$

How can we get K (as well as R,T) from P ?

Extracting intrinsic parameters from P

$$P = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & g & k & l \end{bmatrix} \overset{?}{=} K \cdot \left[\begin{array}{c|c} \quad R \quad & T \end{array}\right]$$

matrix factorization: H&Z Sec 6.2.4 (p. 163)

Theorem [$\mathbb{QR}$ or $\mathbb{RQ}$ factorization]: for any $n \times n$ matrix A there is an orthogonal matrix $\mathbb{Q}$ and an upper (or ***r***ight) triangular matrix $\mathbb{R}$ such that $A = \mathbb{RQ}$.

$$P = \underbrace{\begin{bmatrix} a & b & c \\ e & f & g \\ i & g & k \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} d \\ h \\ l \end{bmatrix}}_{a} \overset{A = \mathcal{R}\mathcal{Q}}{=} \underbrace{\mathcal{R}}_{K} \cdot \left[\underbrace{\mathcal{Q}}_{R} \;\middle|\; \underbrace{\mathcal{R}^{-1} a}_{T} \right]$$

scale $\mathbb{R}$ to make bottom right element equal 1

Calibrated Camera (*camera normalization*)

Once intrinsic parameters K are known

- can "**normalize**" the camera:

  switch to a new image coordinate system $(\tilde{u}, \tilde{v})$ defined as

$$\begin{bmatrix} w\tilde{u} \\ w\tilde{v} \\ w \end{bmatrix} = K^{-1} \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

Q: what kind of transform is this for camera's image?

- then, camera's **new projection matrix** $\tilde{P}$ becomes

$$\tilde{P} = K^{-1} P = \cancel{K^{-1} \cdot K} \cdot \left[\begin{array}{c|c} R & T \end{array} \right] = \boxed{\left[\begin{array}{c|c} R & T \end{array} \right]}$$

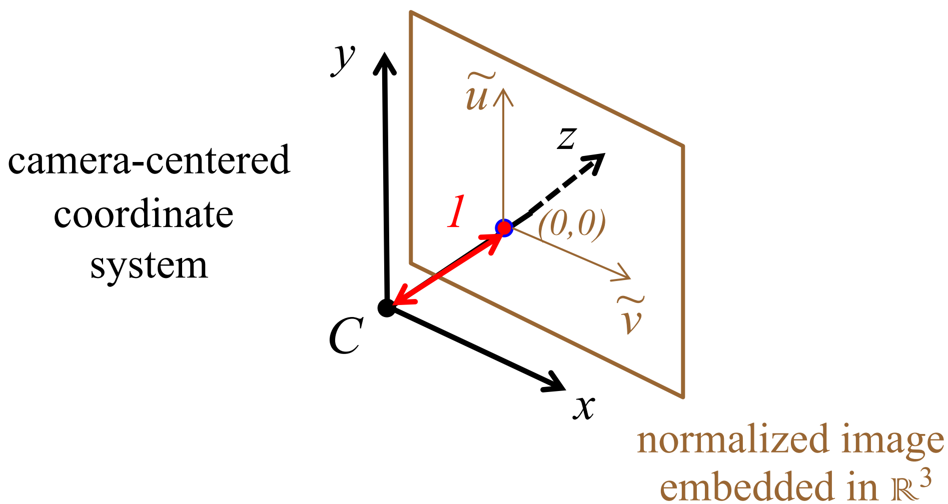
rotation and **translation** only

Calibrated (Normalized) Camera

After normalization, "effective" intrinsic parameters form an **identity matrix**

$$\begin{bmatrix} R & | & T \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\tilde{K} = I} \cdot \begin{bmatrix} R & | & T \end{bmatrix}$$

extrinsic parameters



Geometric interpretation:

focal length $f = 1$

point (0,0) = intersection of image plane with optical axis

Calibrated (Normalized) Camera

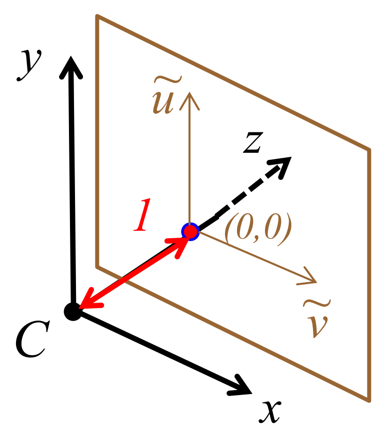
To project onto a calibrated camera (a.k.a. *normalized camera*) one needs only its position (**translation+rotation**) in world coordinates

calibrated/normalized camera's projection matrix

$$P = \left[\begin{array}{c|c} R & T \end{array} \right]$$

still 3x4 matrix
but only 6 d.o.f

camera-centered coordinate system



normalized image embedded in $\mathbb{R}^3$

Calibrated (Normalized) Camera

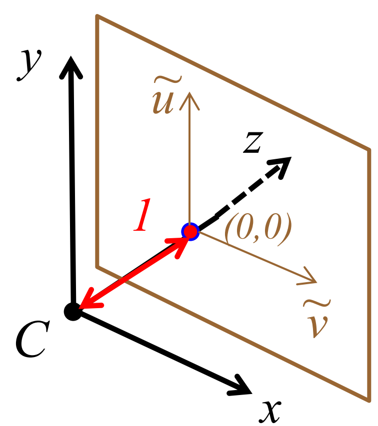
To project onto a calibrated camera (a.k.a. *normalized camera*) one needs only its position (**translation+rotation**) in world coordinates

calibrated/normalized camera's projection matrix

$$P = \left[\begin{array}{c|c} R & T \end{array} \right]$$

still 3x4 matrix
but only 6 d.o.f

camera-centered coordinate system



normalized image embedded in $\mathbb{R}^3$

The main point of calibration/normalization: converts any camera to a "standardized" pin hole camera model shown on the left. After calibration, images are independent of how the camera is made and depend only on camera's location/orientation.

NOTE: in general, "calibration" process also correct for lens distortions (barrel, etc.)

Calibrated (Normalized) Camera

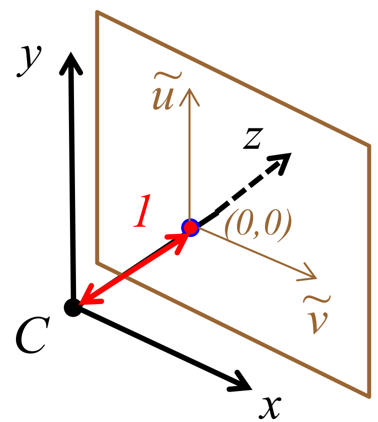
To project onto a calibrated camera (a.k.a. *normalized camera*) one needs only its position (**translation+rotation**) in world coordinates

calibrated/normalized camera's projection matrix

$$P = \left[\begin{array}{c|c} R & T \end{array} \right]$$

still 3x4 matrix
but only 6 d.o.f

camera-centered coordinate system



normalized image embedded in $\mathbb{R}^3$

Estimating multiple viewpoints P_n is the "motion" part of the *structure-from-motion* problem

NOTE: *camera calibration* uses known 3D points $\{\tilde{X}_i\}$.

The "structure" part of *SfM* problem estimates unknown 3D scene points $\{\tilde{X}_i\}$.

(later in this topic)

Calibrated (Normalized) Camera

For simplicity, the rest of this topic assumes that all images are normalized (calibrated cameras)

unless explicitly stated otherwise

Two cameras geometry

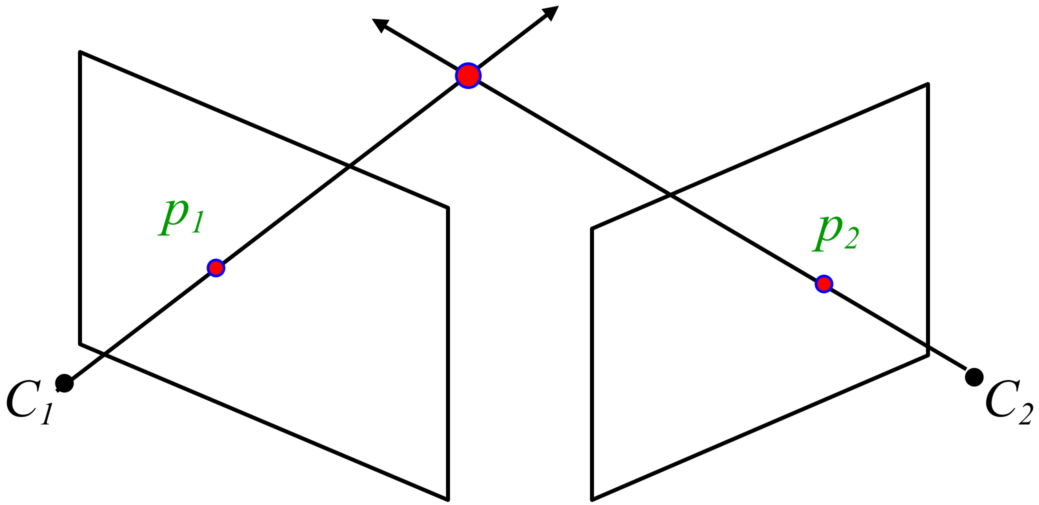
Epipolar geometry

essential & fundamental matrices

Motivation: helps reconstruction

Stereo reconstruction

From 2D images back to 3D scene

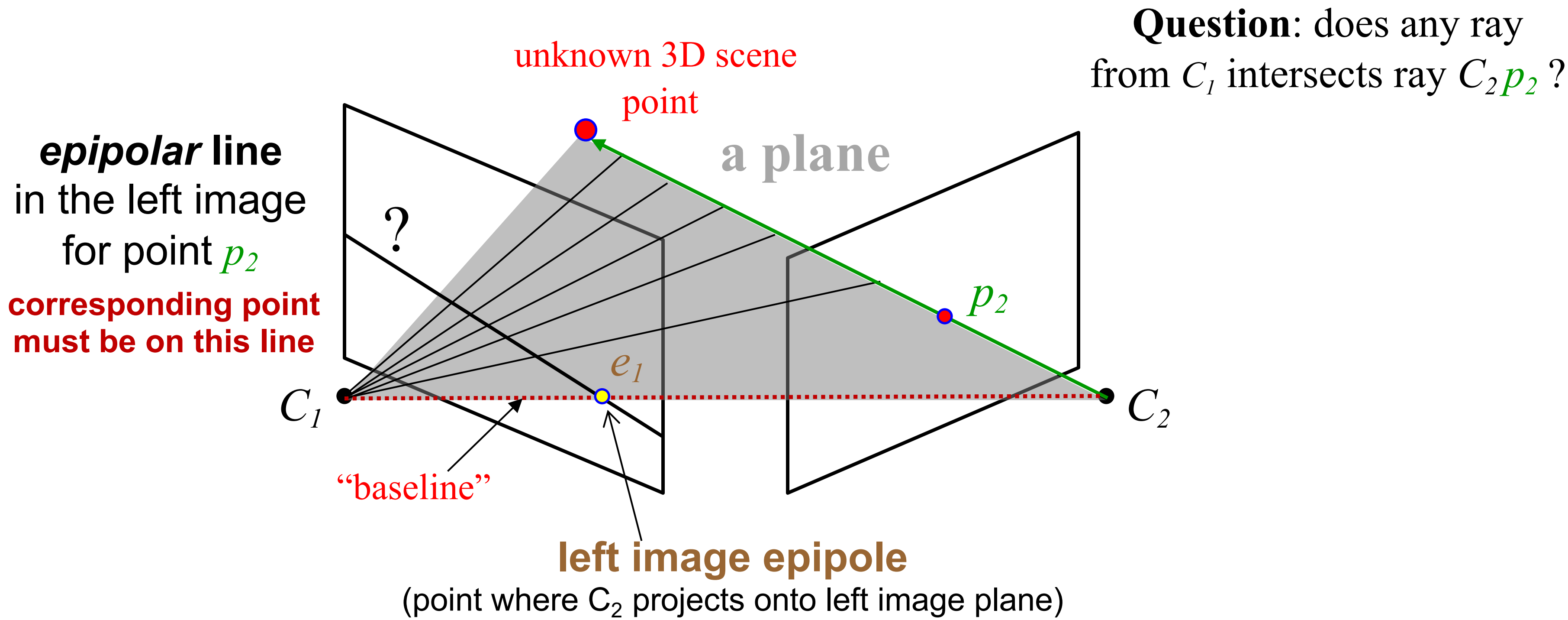


Triangulation: can reconstruct a point as an intersection of two rays, **<u>assuming</u>**…

- known projection matrix (camera position)
- known ***point correspondence***

Epipolar lines

- Find pairs of corresponding pixels (that come from the same 3D scene point)
  - not trivial (remember mosaicing)



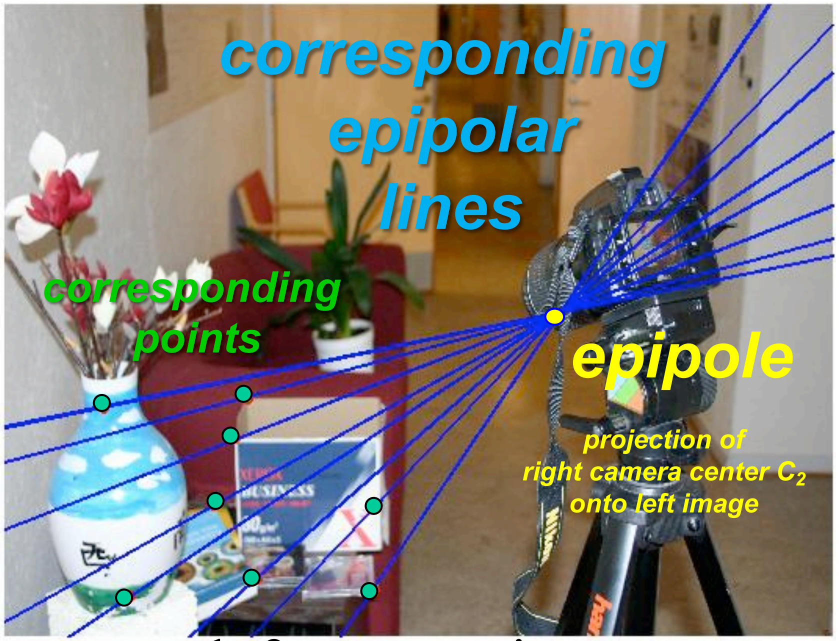
Any right image point p_2 corresponds to some left image **epipolar line**.

It is a projection of ray $C_2 \rightarrow p_2$ (ray $C_2 \rightarrow$ unknown 3D scene point).

Epipolar lines

Example [from Carl Olsson]
(two stationary cameras)

consider some features
in the right image
(projections of some 3D points)



left camera image
(contains the right camera)

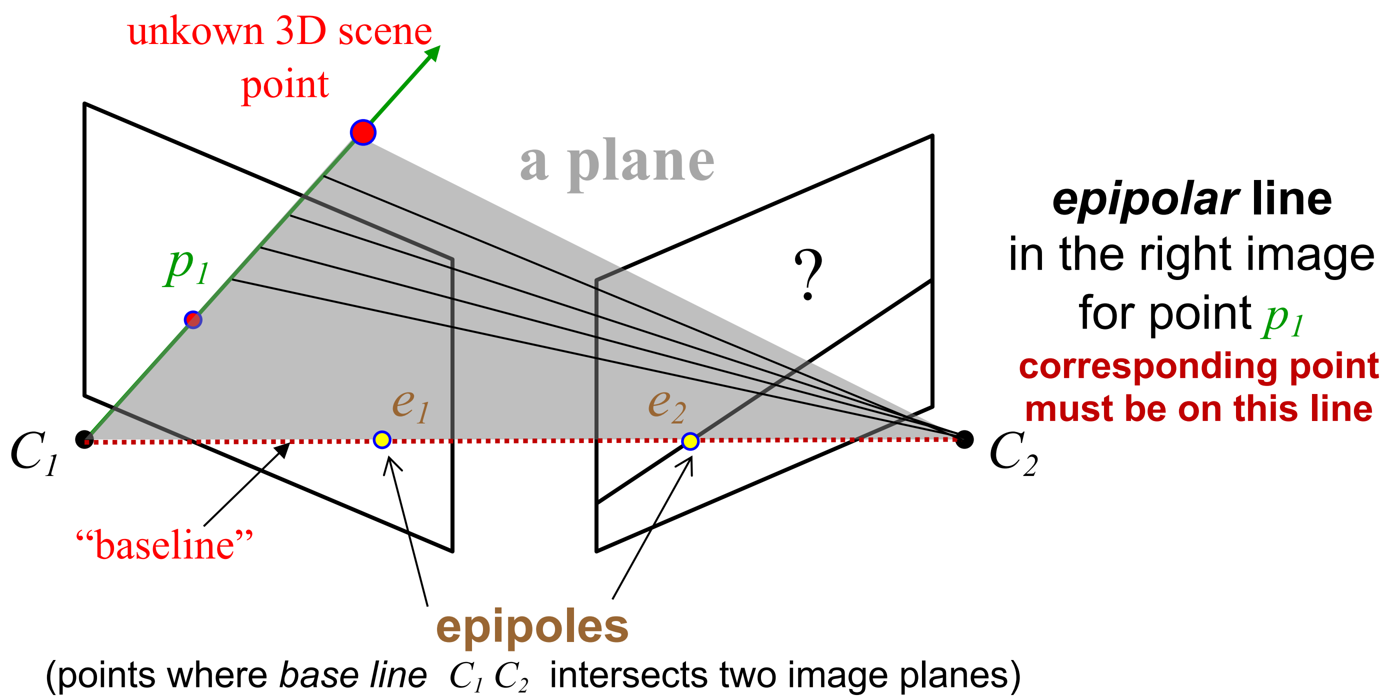
right camera image

Any right image point p_2 corresponds to some left image **epipolar line**.

It is a projection of ray $C_2 \rightarrow p_2$ (ray $C_2 \rightarrow$ unknown 3D scene point).

Epipolar lines

Similarly, for any given point p_1 in the left image…

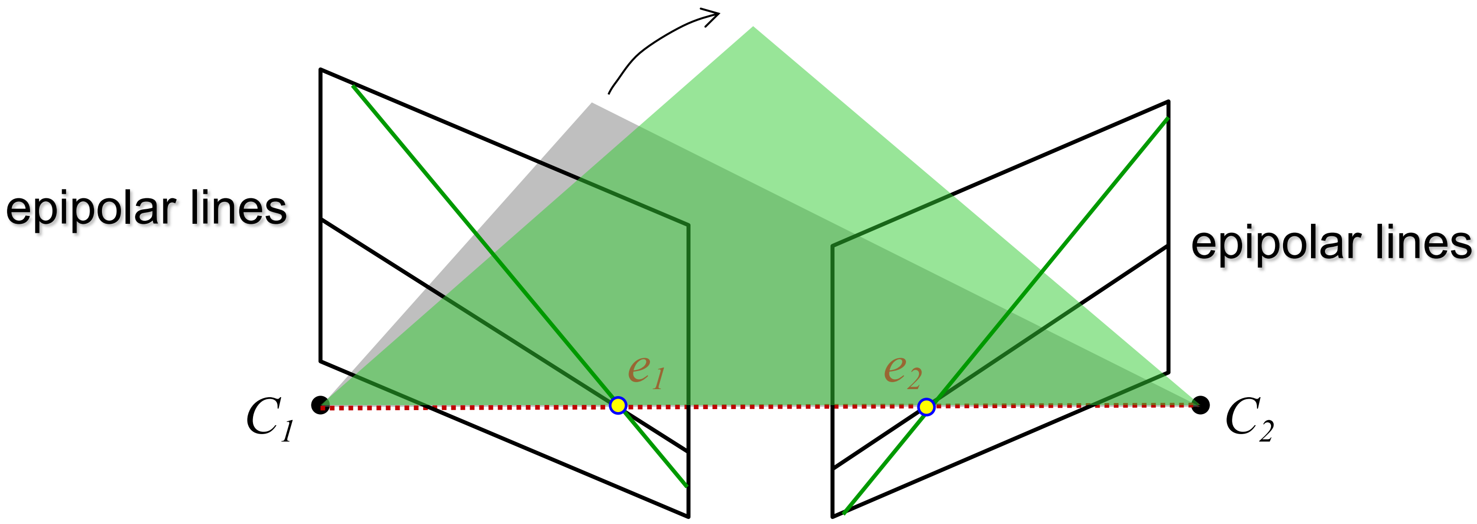


epipolar constraint for the right image: for any point p_1 in the left image, the corresponding point in the right image must be on the line where plane $p_1\, C_1\, C_2$ intersects the right image (right image *epipolar line*)

- reduces correspondence problem to 1D search along conjugate *epipolar lines*

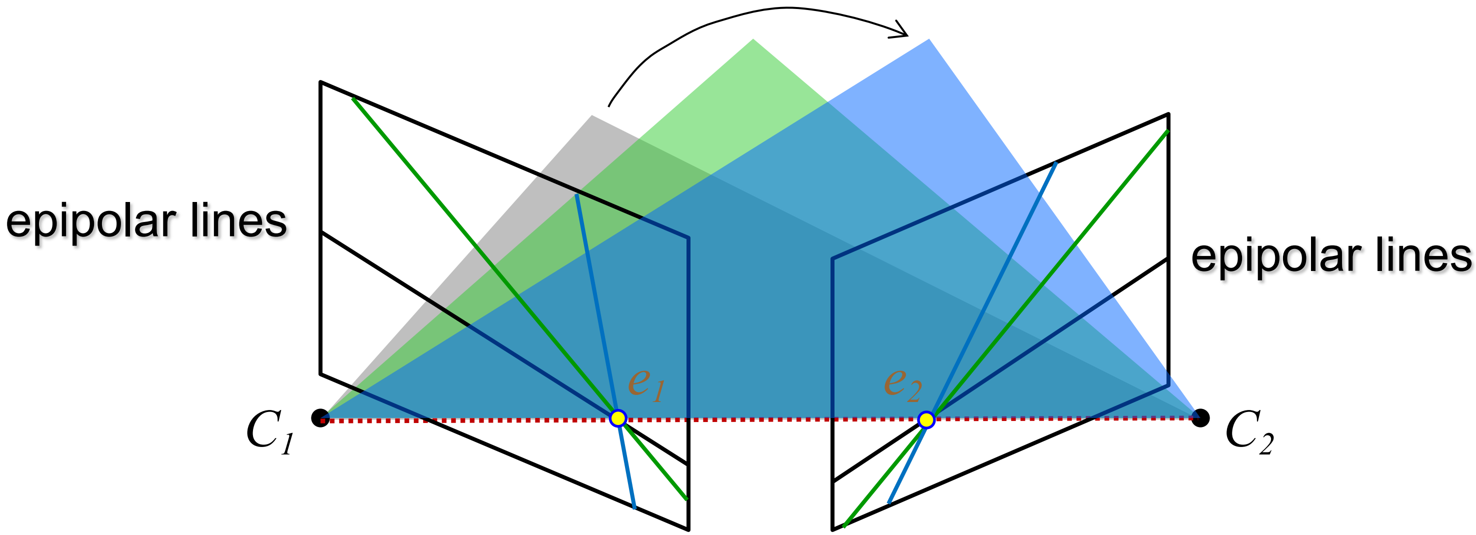
Epipolar lines

System of corresponding epipolar lines depends only on camera set up and it does not depend on 3D scene.



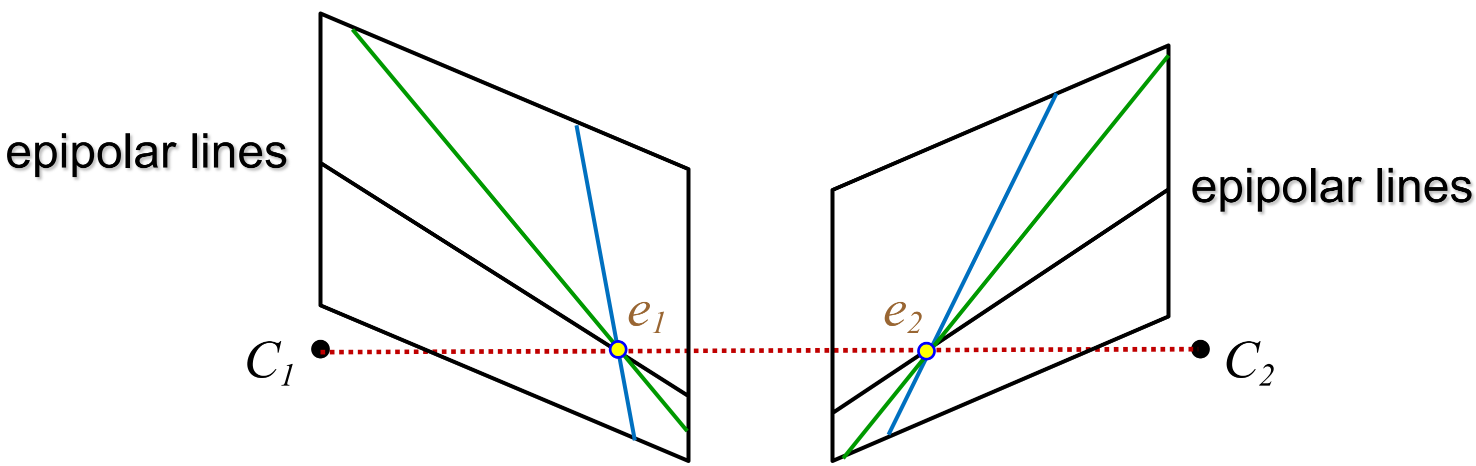
Epipolar lines

System of corresponding epipolar lines depends only on camera set up and it does not depend on 3D scene.



- Intersection of **epipolar planes** (planes containing base line C_1C_2) with image planes define a system of corresponding *epipolar lines*
- <u>Corresponding points can be only on corresponding epipolar lines</u>
  - important to know such lines when searching for corresponding pairs of points

Epipolar lines



- **How can we compute epipolar lines for a given pair of images?**

 - if known, camera projection matrices P_1 and P_2 contain all information

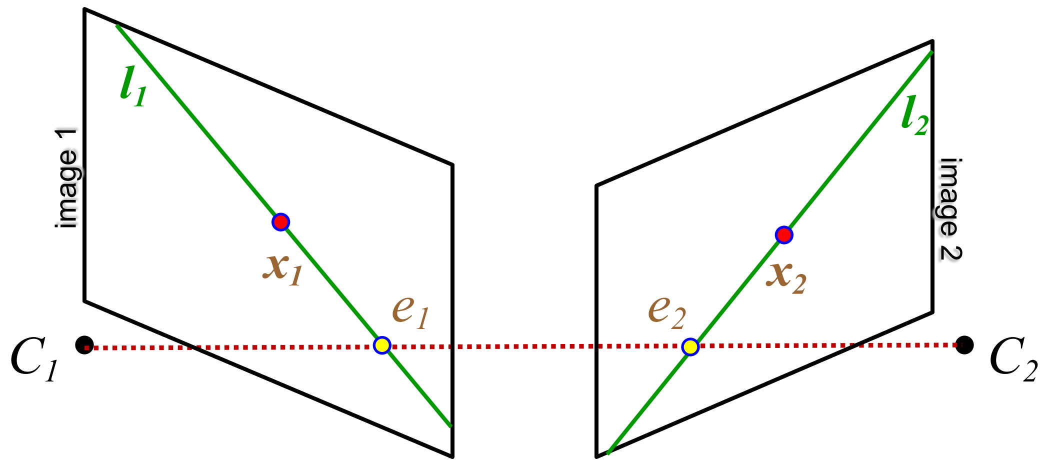
$$e_1 = P_1\, C_2 \qquad e_2 = P_2\, C_1 \qquad x_1 = P_1\, X \qquad x_2 = P_2\, X \qquad (X - \text{any 3D point})$$

- <u>but only relative position of two cameras really matters</u>:

  can estimate a single 3x3 ***essential matrix*** rather than two 3x4 matrices $P = (R|T)$ …

Essential matrix E (definition)

The system of corresponding epipolar lines
is fully described by a 3x3 matrix E in equation below



3x3 matrix

$$x_2^T E x_1 = 0$$

$(l_1)^T$ l_2

for any pair of pixels/points x_1 and x_2 on the corresponding epipolar lines (assuming calibrated cameras)

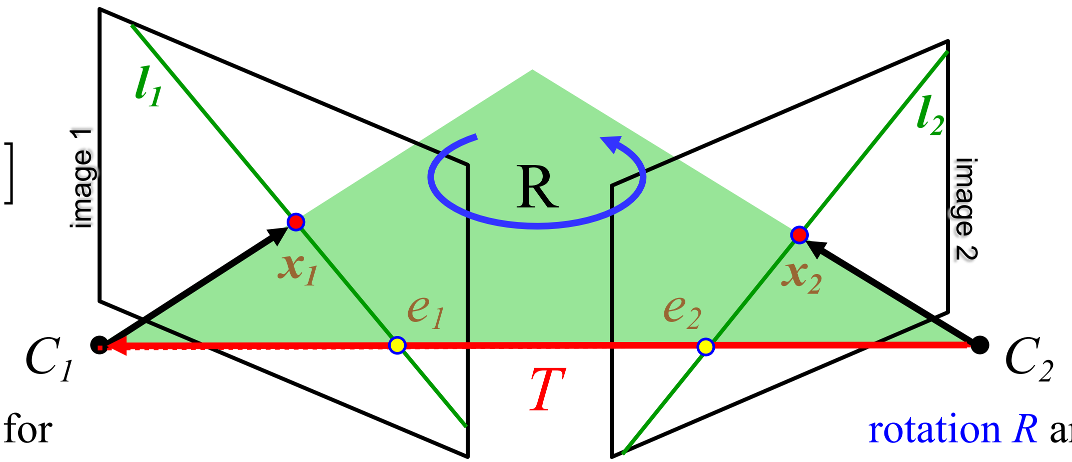
NOTE: given x_1 in image 1 vector $l_2 = E x_1$ gives equation $x_2 \cdot l_2 = 0$ (a line in image 2)

given x_2 in image 2 vector $l_1 = E^T x_2$ gives equation $x_1 \cdot l_1 = 0$ (a line in image 1)

Essential matrix E (proof of existence)

Recall: assuming calibrated cameras, pixels x_1 and x_2 in (homogeneous) image coordinates can be treated as **3D points (vectors)** in the corresponding camera-centered coordinates of 3D space

$$P_1 = [\ I \mid 0\ ]$$



$$P_2 = [\ R \mid T\ ]$$

use camera 1 for "**world coordinates**"

rotation R and translation T convert camera 1 (world) coordinates to camera 2

dot product cross product

$$x_2 \cdot [T \times (Rx_1)] = 0$$

for any pair of pixels/points x_1 and x_2 on the corresponding epipolar lines (assuming calibrated cameras)

co-planarity constraint for x_1 and x_2

treating x_1 and x_2 as vectors in $\mathbb{R}^3$

NOTE: Rx_1 is vector x_1 in camera 2 coordinates and $T \times Rx_1$ is the green plane's normal (camera 2 coordinates)

Essential matrix E (proof of existence)

NOTE: cross product $a \times b$ can be represented as matrix multiplication

$$a = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \quad b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad \Rightarrow \quad a \times b = \underbrace{\begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}}_{\text{notation: } [a]_\times} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$a \times b \equiv [a]_\times b$$

3x3 *skew-symmetric* matrix, **rank 2**
(a.k.a. *antisymmetric* matrix $M = -M^T$)

Q: null space of $[a]_x$ dimensions? A: 0 B: 1 C: 2 D: 3

dot product cross product

$$x_2 \cdot [T \times (R x_1)] = 0 \quad \Leftrightarrow \quad x_2^T [T]_\times R x_1 = 0$$

co-planarity constraint for x_1 and x_2
treating x_1 and x_2 as vectors in $\mathbb{R}^3$

matrix expression

Essential matrix E (proof of existence)

NOTE: due to homogeneous coordinates, scale of E is arbitrary

$$x_2^T E x_1 = 0$$

essential matrix

E

$$x_2^T [T]_\times R x_1 = 0$$

matrix expression

Essential matrix E

Theorem [***existence*** and ***uniqueness*** of essential matrix]:
Assume two calibrated cameras with non-zero baseline.
There exists (unique up to scale) 3x3 matrix E such that
for any $X \in \mathcal{P}^3$

$$x_1^T E x_2 = 0$$

where $x_1, x_2 \in \mathcal{P}^2$ are projections of X on two cameras,
i.e. $x_i = P_i X$ for cameras' projection matrices P_1 and P_2.

NOTE: due to homogeneous coordinates, scale of E is arbitrary

$$x_2^T E x_1 = 0$$

essential matrix

E

$$x_2^T [T]_\times R x_1 = 0$$

matrix expression

Essential matrix E

Theorem [***existence*** and ***uniqueness*** of essential matrix]:
Assume two calibrated cameras with non-zero baseline.
There exists (unique up to scale) 3x3 matrix E such that
for any $X \in \mathcal{P}^3$

$$x_1^T E x_2 = 0$$

where $x_1, x_2 \in \mathcal{P}^2$ are projections of X on two cameras,
i.e. $x_i = P_i X$ for cameras' projection matrices P_1 and P_2.

nontrivial exercise: prove up-to-scale uniqueness of E

NOTE: due to homogeneous coordinates, scale of E is arbitrary

$$x_2^T E x_1 = 0$$

essential matrix

E

$$x_2^T [T]_\times R x_1 = 0$$

matrix expression

E is defined by a relative position of two cameras (R and T), as expected

$$E = [T]_\times R$$

Q: How many *d.o.f* in E ?

A: 5 = 3 (rotation) + 3-1 (**scale of *T* is arbitrary**)

Essential matrix E

Theorem [***existence*** and ***uniqueness*** of essential matrix]: Assume two calibrated cameras with non-zero baseline. There exists (unique up to scale) 3x3 matrix E such that for any $X \in \mathcal{P}^3$

$$x_1^T E x_2 = 0$$

where $x_1, x_2 \in \mathcal{P}^2$ are projections of X on two cameras, *i.e.* $x_i = P_i X$ for cameras' projection matrices P_1 and P_2.

nontrivial exercise: prove up-to-scale uniqueness of E

NOTE: due to homogeneous coordinates, scale of E is arbitrary

$$x_2^T E x_1 = 0$$

essential matrix

E

$$x_2^T [T]_\times R x_1 = 0$$

matrix expression

E is defined by a relative position of two cameras (R and T), as expected

$$E = [T]_\times R$$

Q: What is the rank of E ?

Fundamental matrix *F*

$$x_2^T E x_1 = 0$$

This assumes calibrated camera coordinates

Remember: $\tilde{x} = K^{-1} x$

calibrated (normalized) coordinates

original image coordinates

$$\Rightarrow \quad \underbrace{\overbrace{x_2^T K^{-T}}^{\tilde{x}_2^T} E \overbrace{K^{-1} x_1}^{\tilde{x}_1}}_{} = 0$$

F - fundamental matrix ($F = K^{-T} E K^{-1}$)

$$x_2^T F x_1 = 0$$

defines epipolar lines for uncalibrated cameras

Essential and Fundamental matrices

essential matrix E	fundamental matrix F
• epipolar lines $x_2^T E x_1 = 0$ (for two calibrated cameras)	• epipolar lines $x_2^T F x_1 = 0$ (for two arbitrary cameras)
• rank 2 $E = [T]_\times R$	• rank 2 $F = K^{-T} E K^{-1}$
• epipoles e_1 and e_2 are right and left null vectors for E $Ee_1 = \mathbf{0}$ $e_2^T E = \mathbf{0}^T$	• epipoles e_1 and e_2 are right and left null vectors for F $Fe_1 = \mathbf{0}$ $e_2^T F = \mathbf{0}^T$
• 5 d.o.f (6 from *R&T,* - scale of T)	• 7 d.o.f (9 par., - scale & det F=0)
• two equal non-zero singular values	• two non-zero singular values

What's left to cover

- Estimation of E and F
  - simpler **8-point method** (no explicit enforcement of rank or other constraints for E or F)
  - more advanced **5-point method** (see H&Z book, we do not cover this in class)
  - similarly to homography estimation in previous topics, we cover only least squares for *algebraic* errors (*reprojection* errors use more advanced optimization)
- Extraction of cameras (projection matrices) from E
- Structure from Motion
  - match, find E, find cameras (estimate pose), **triangulate** (estimate structure)
  - bundle adjustment
  - reconstruction ambiguities

Estimating F or E from $N \geq 8$ matches

8-point method

Assume corresponding points $\mathbf{x}_i \leftrightarrow \bar{\mathbf{x}}_i$ in two images

(matched pair corresponding to a projection of unknown 3D point X_i)

They must lie on the corresponding epipolar lines, thus

$$\bar{\mathbf{x}}_i^T F \mathbf{x}_i = 0$$

(use E for calibrated images)

If $\mathbf{x}_i = (x_i, y_i, z_i)$ and $\bar{\mathbf{x}}_i = (\bar{x}_i, \bar{y}_i, \bar{z}_i)$ then

$$\begin{aligned} \bar{\mathbf{x}}_i^T F \mathbf{x}_i = \; & F_{11}\bar{x}_i x_i + F_{12}\bar{x}_i y_i + F_{13}\bar{x}_i z_i \\ + \; & F_{21}\bar{y}_i x_i + F_{22}\bar{y}_i y_i + F_{23}\bar{y}_i z_i \\ + \; & F_{31}\bar{z}_i x_i + F_{32}\bar{z}_i y_i + F_{33}\bar{z}_i z_i = 0 \end{aligned}$$

One matching pair $\mathbf{x}_i \leftrightarrow \bar{\mathbf{x}}_i$ gives **only one linear equation**.

Eight is enough to determine elements of 3x3 matrix F (as scale is arbitrary)

Note: enforcing known properties (e.g. rank=2) allows to use fewer points.

Estimating F or E from $N \geq 8$ matches

In matrix form: one row for each of $N \geq 8$ correspondences

$$\underbrace{\begin{bmatrix} \bar{x}_1 x_1 & \bar{x}_1 y_1 & \bar{x}_1 z_1 & \cdots & \bar{z}_1 z_1 \\ \bar{x}_2 x_2 & \bar{x}_2 y_2 & \bar{x}_2 z_2 & \cdots & \bar{z}_2 z_2 \\ \bar{x}_3 x_3 & \bar{x}_3 y_3 & \bar{x}_3 z_3 & \cdots & \bar{z}_3 z_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \bar{x}_N x_N & \bar{x}_N y_N & \bar{x}_N z_N & \cdots & \bar{z}_N z_N \end{bmatrix}}_{\mathbf{A}\ (\text{Nx9})} \underbrace{\begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ \vdots \\ F_{33} \end{bmatrix}}_{\mathbf{f}\ (\text{9x1})} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{\mathbf{0}\ (\text{Nx1})}$$

$$\boxed{\mathbf{A}\,\mathbf{f} = \mathbf{0}}$$

If matched points have some errors (not exact locations) ?

Estimating F or E from $N \geq 8$ matches

solve *homogeneous least squares*

$$\min_{\|\mathbf{f}\|=1} \|\mathbf{A}\mathbf{f}\|$$

as in homography estimation,
constraint $\|\mathbf{f}\|=1$ fixes the scale of $\mathbf{f}$ (i.e. F)

$$\begin{bmatrix} E_{11} \\ E_{12} \\ E_{13} \\ \vdots \\ E_{33} \end{bmatrix}$$

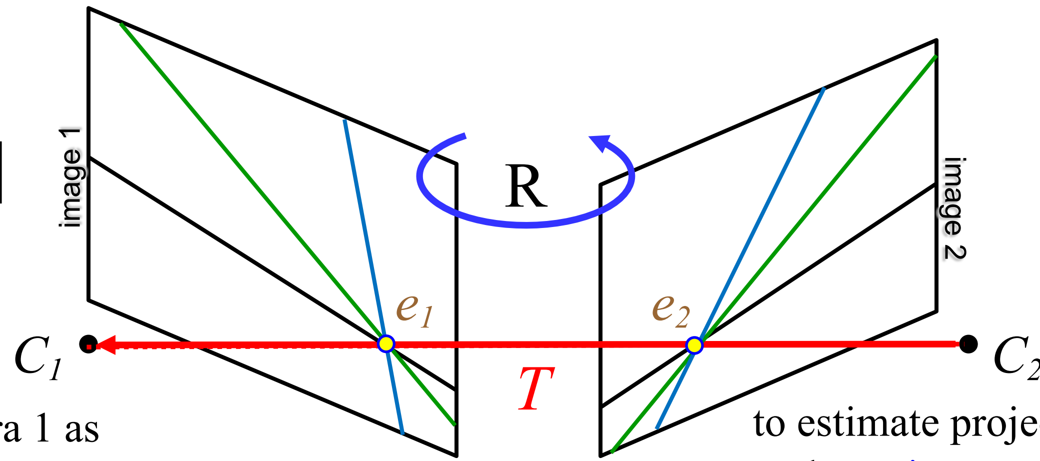
for E use $\mathbf{e}$ instead of $\mathbf{f}$

Use eigen vector for the smallest eigen value of 9x9 matrix $\mathbf{A}^T\mathbf{A}$

Extracting cameras from essential matrix E

Now assume essential matrix E is given, need to find P_1 and P_2

$$P_1 = [\ I \mid 0\ ]$$



$$P_2 = [\ R \mid T\ ]$$

can choose camera 1 as "**world coordinates**"

to estimate projection matrix for camera 2 need rotation R and translation T converting camera 1 (world) coordinates to camera 2

Given essential matrix $E = U \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T$

find rotation R and translation T such that $E = [T]_\times R$

mathematical formulation of the problem

Extracting cameras from essential matrix E

Four distinct R,T solutions

(up to scale)

Assume SVD decomposition $$E = U \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T$$

such that $det(UV^T) = 1$ (if $det(UV^T)$ = -1 switch the sign of the last column in V).

Then, using special matrix $$W := \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$ we have

$E = [T]_\times R$ for any combination of $R = UWV^T$ or UW^TV^T
and $T = \pm U_3$ (scale is arbitrary)

see [H&Z:sec 9.6.2, p.258] for proof

the last column of U

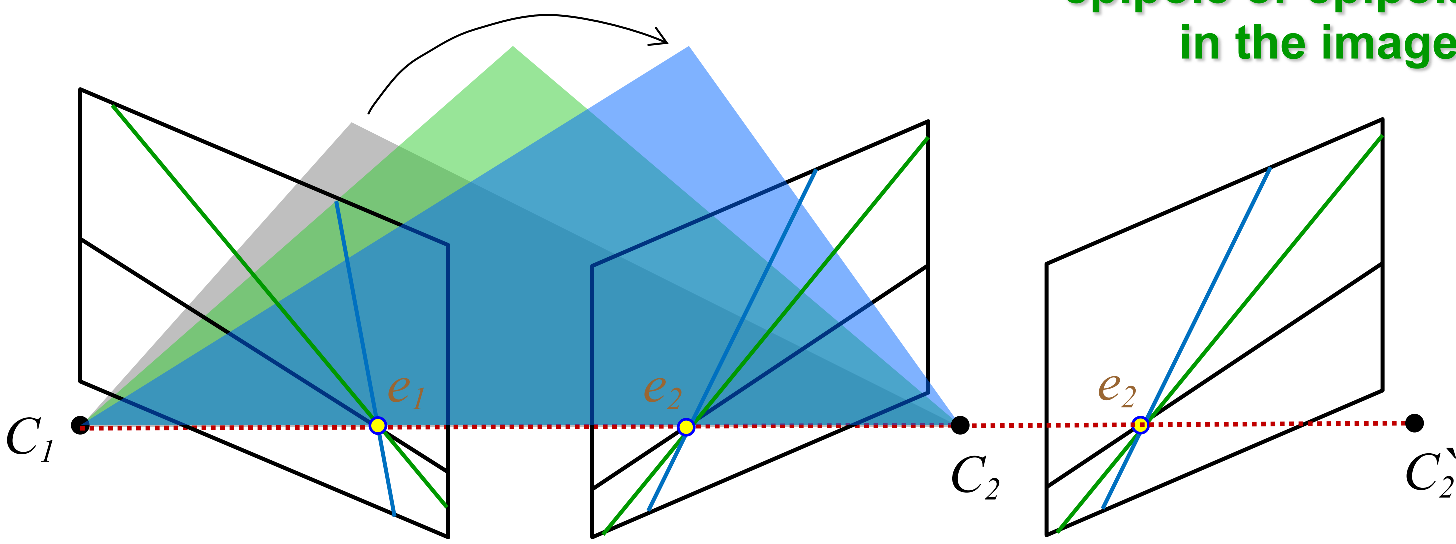
Q: Why?

Extracting cameras from essential matrix E

Four distinct R, T solutions

(up to scale of T)

baseline length |T| does not change epipole or epipolar lines in the images



$$E = [T]_\times R \text{ for any combination of } \begin{array}{rcl} R & = & UWV^T \text{ or } UW^TV^T \\ \text{and } T & = & \pm U_3 \end{array}$$ (scale is arbitrary)

see [H&Z:sec 9.6.2, p.258] for proof

the last column of U

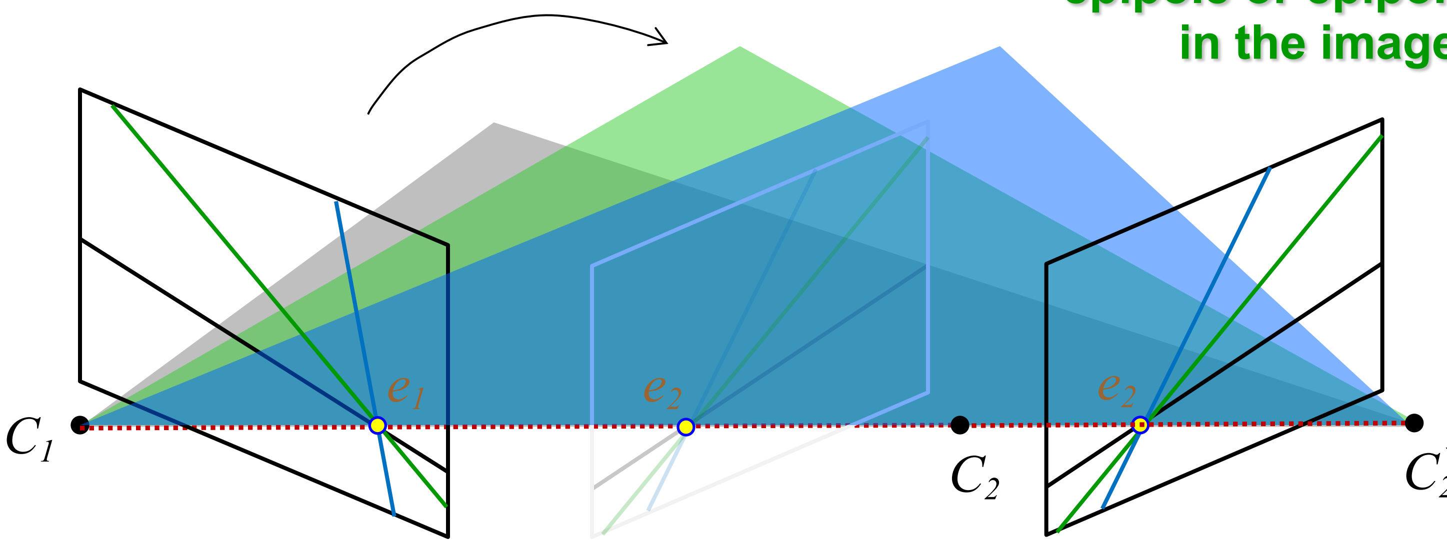
Q: Why?

Extracting cameras from essential matrix E

Four distinct R, T solutions

(up to scale of T)

baseline length |T| does not change epipole or epipolar lines in the images



same epipolar planes same epipolar lines

$$E = [T]_\times R \text{ for any combination of } \quad R = UWV^T \text{ or } UW^TV^T$$
$$\text{and} \quad T = \pm U_3 \quad \text{(scale is arbitrary)}$$

see [H&Z:sec 9.6.2, p.258] for proof

the last column of U

Q: Why?

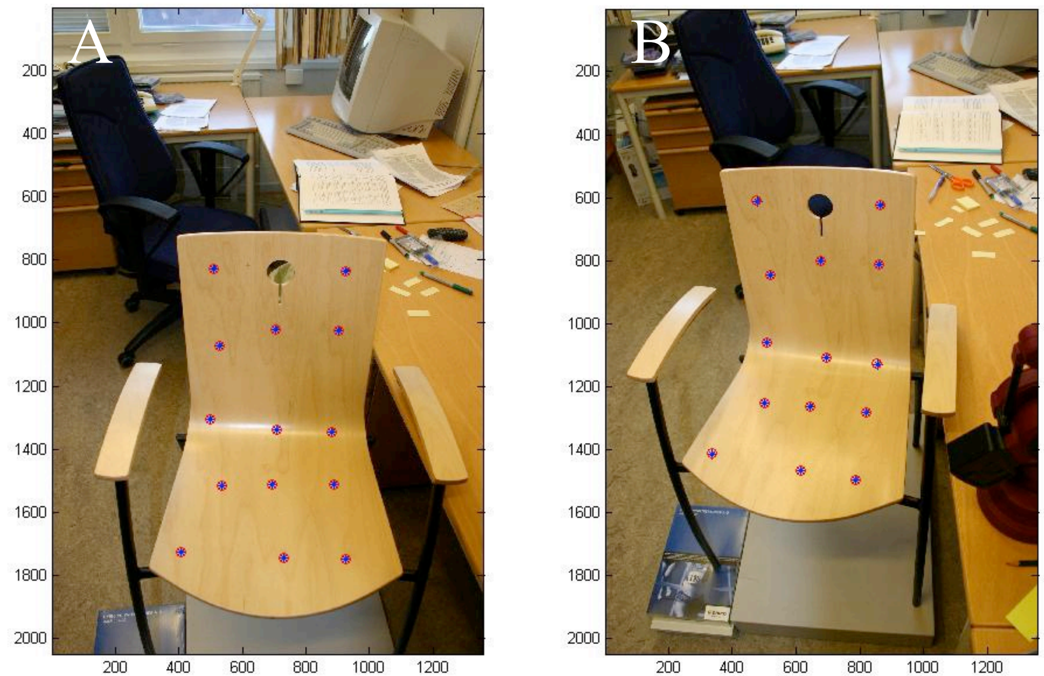
Extracting cameras from essential matrix E

Four distinct R,T solutions

(up to scale of T)

Example:
[from Carl Olsson]

Two given views of a chair



14 known correspondences (for 14 **non-coplanar** 3D points)
allow to estimate essential matrix $\boldsymbol{E}$
assuming K is known
(e.g. 8 point method)

Extracting cameras from essential matrix E

Four distinct R,T solutions

(up to scale of T)

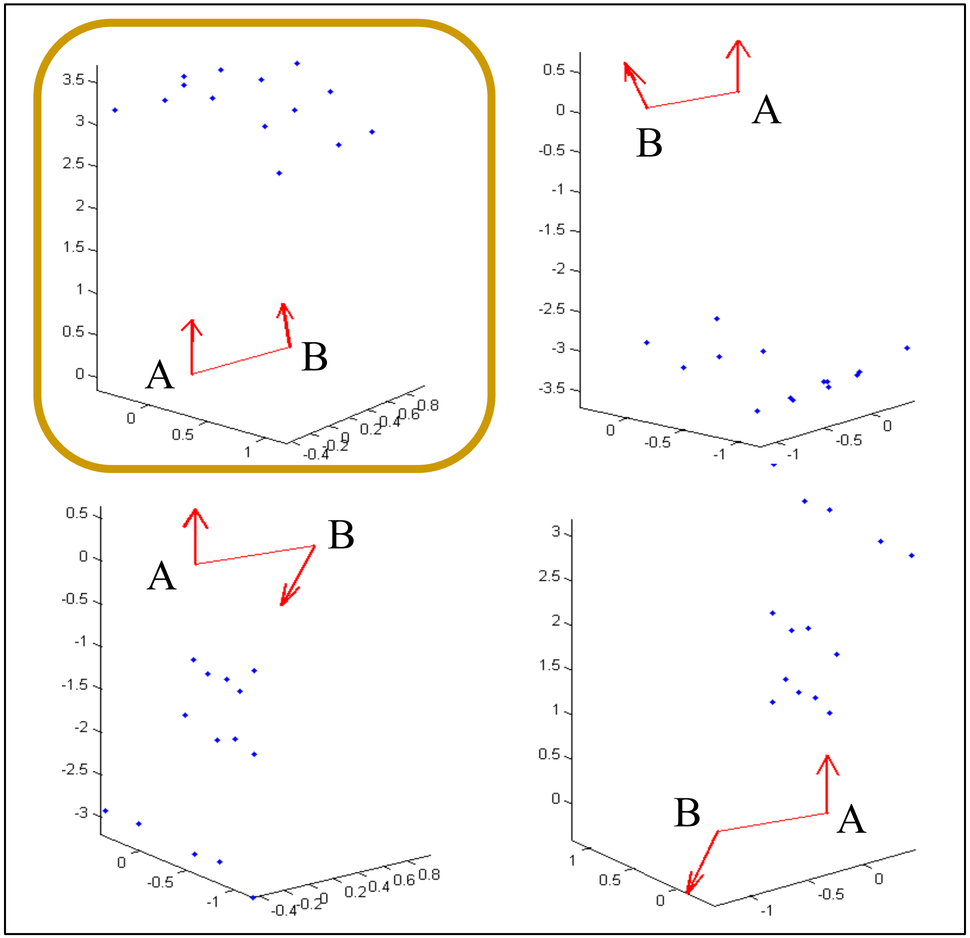
Example:

[from Carl Olsson]

- four distinct **relative camera positions** (motion R, T) computed from E (up to scale)

- 3D structure $\{X_i\}$ computed from correspondences $\mathbf{x}_i \leftrightarrow \bar{\mathbf{x}}_i$ by *triangulation* (more soon...) up to a *similarity transformation* (i.e. scale+position+orientation)

baseline reversal ($T=\pm U_3$)



camera B orientation flips ($R = UWV^T$ or UW^TV^T)

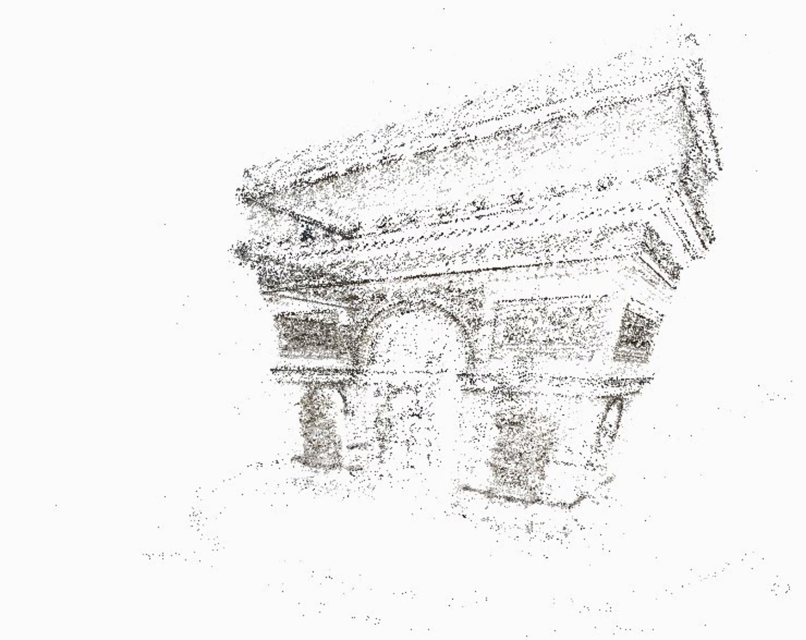
Note: only one solution has positive "depths" for both cameras

Extracting cameras from fundamental matrix F

One can also estimate camera projection matrices from **fundamental matrix**, but there are more ambiguities [see H&Z]

Examples

[from Carl Olsson]

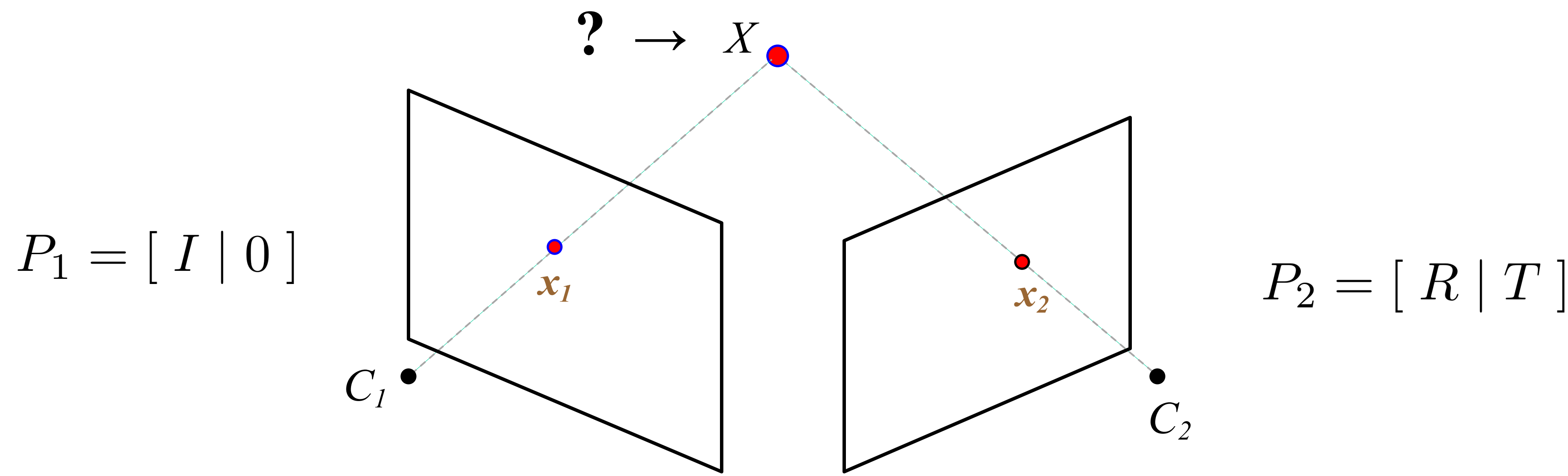


"projective" ambiguity
(cameras estimated from F)

3D reconstruction with
similarity transform ambiguity
(cameras estimated from E)

Triangulation

Now, assume known projection matrices P_1 , P_2 and a match $\mathbf{x}_1 \leftrightarrow \mathbf{x}_2$



projection constraints

$$\begin{bmatrix} w_1 u_1 \\ w_1 v_1 \\ w_1 \end{bmatrix} = P_1 \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \qquad \begin{bmatrix} w_2 u_2 \\ w_2 v_2 \\ w_2 \end{bmatrix} = P_2 \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

6 equations with 5 unknown (X, Y, Z, w_1, w_2)

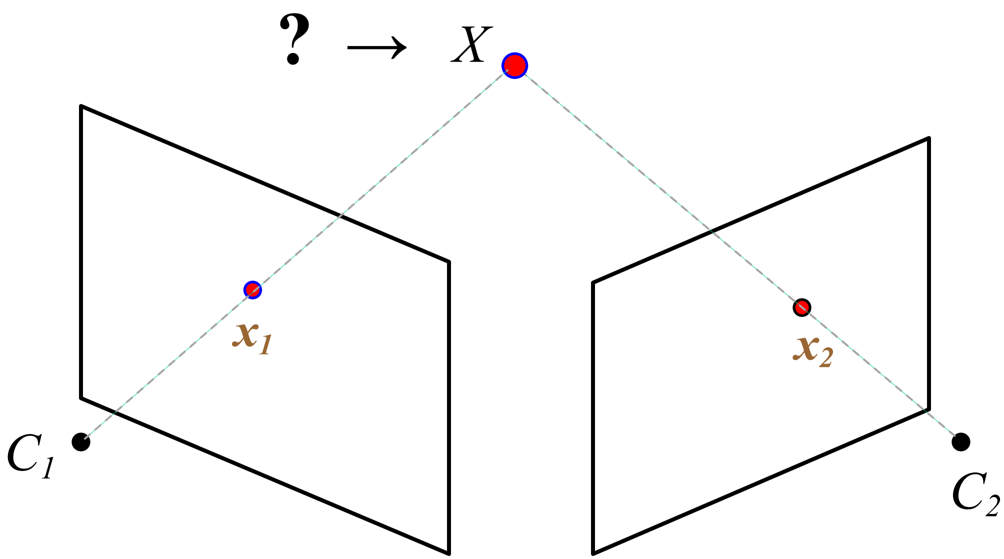
But, we do not care about w_1 & w_2 – **eliminate** them (*à la* slide 15 topic 6)

=> 4 equations with 3 unknown (X, Y, Z)

Triangulation

Now, assume known projection matrices P_1 , P_2 and a match $\mathbf{x}_1 \leftrightarrow \mathbf{x}_2$

$$P_1 = [\ I \mid 0\ ]$$



$$P_2 = [\ R \mid T\ ]$$

projection constraints

$$\begin{bmatrix} w_1 u_1 \\ w_1 v_1 \\ w_1 \end{bmatrix} = P_1 \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \qquad \begin{bmatrix} w_2 u_2 \\ w_2 v_2 \\ w_2 \end{bmatrix} = P_2 \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

One equation is redundant only if points x_1, x_2 are exactly on the corresponding epipolar lines (the corresponding rays intersect in 3D).

Due to errors, use least squares.

Structure-from-Motion workflow

Basic sequential reconstruction

- For the first two images, use 8-point algorithm to estimate essential matrix E, cameras, and triangulate some points $\{X_i\}$.
- Each new view should see some previously reconstructed scene points $\{X_i\}$ ("feature matches" with previous cameras). Use such points to estimate new camera position (*resection problem*).
- Add new scene points using triangulation, e.g. for new "matches" with previously non-matched (and non-triangulated) features in earlier views.
- If there are more cameras, iterate previous two steps.

- **Issues**
  - errors can accumulate
  - new views are used only to add new 3D points, but they can help to improve accuracy for previously reconstructed scene

Structure-from-Motion workflow

"Bundle adjustment"

i-th "feature track" $tr_i := \{x_{ik} | k \in V(i)\}$

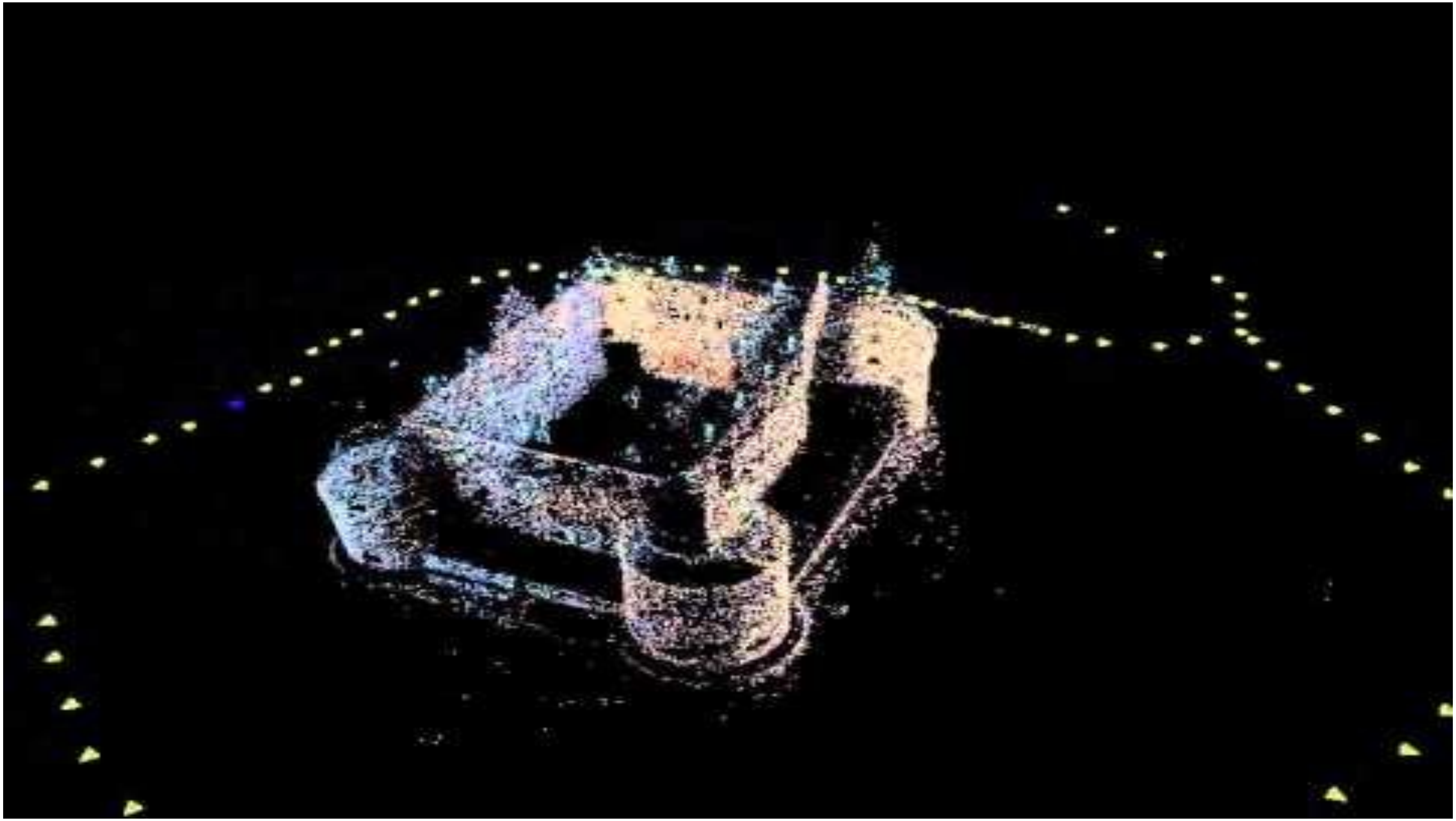
feature *i* location in image k

set of images where feature *i* is visible

$$\min_{\{P_k\},\{X_i\}} \sum_i \sum_{k \in V(i)} \|x_{ik} - P_k X_i\|$$

re-projection error

Structure-from-Motion workflow



https://www.youtube.com/watch?v=i7ierVkXYa8
from Carl Olsson

Applications of multi-view geometry:

Pose estimation

Rigid motion segmentation

Augmented reality

Special effects in video

Volumetric 3D reconstruction

Depth reconstruction (stereo-next topic)