CS231M · Mobile Computer Vision

Announcements

- P2 due 5/8 (this Friday!)
- Project proposals are due on 5/11 (on Monday!)
- Paper presentations will be starting next Wed (5/13)

CS231M · Mobile Computer Vision

Lecture 11

Inferring 3D geometry from images

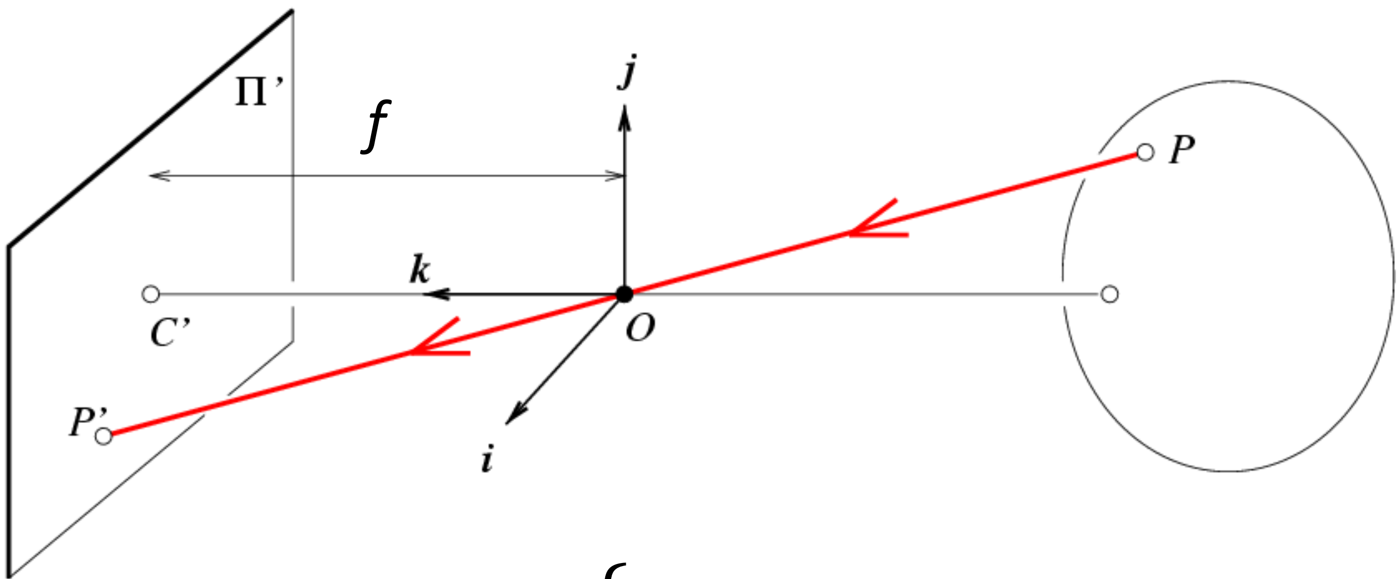
- Cameras
- Single view metrology
- Epipolar geometry
- Structure from motion

Background reading:

[HZ] Chapter 6 “Camera Models”
[HZ] Chapter 7 “Computation of Camera Matrix P”
[HZ] Chapter 2 “Projective Geometry and Transformation in 2D”
[HZ] Chapter 3 “Projective Geometry and Transformation in 3D”
[HZ] Chapter 8 “More Single View Geometry”
[HZ] Chapter: 9 “Epip. Geom. and the Fundam. Matrix Transf.”
[HZ] Chapter: 18 “N view computational methods”
[FP] Chapters: 8 “Structure from Motion”

[PF] = Forsyth, Ponce “Computer vision: a modern approach”, 2011
[HZ] = R. Hartley and A. Zisserman. “Multiple View Geometry in Computer Vision”, 2003.

Pinhole camera



$$P = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \rightarrow P' = \begin{bmatrix} x' \\ y' \end{bmatrix}$$

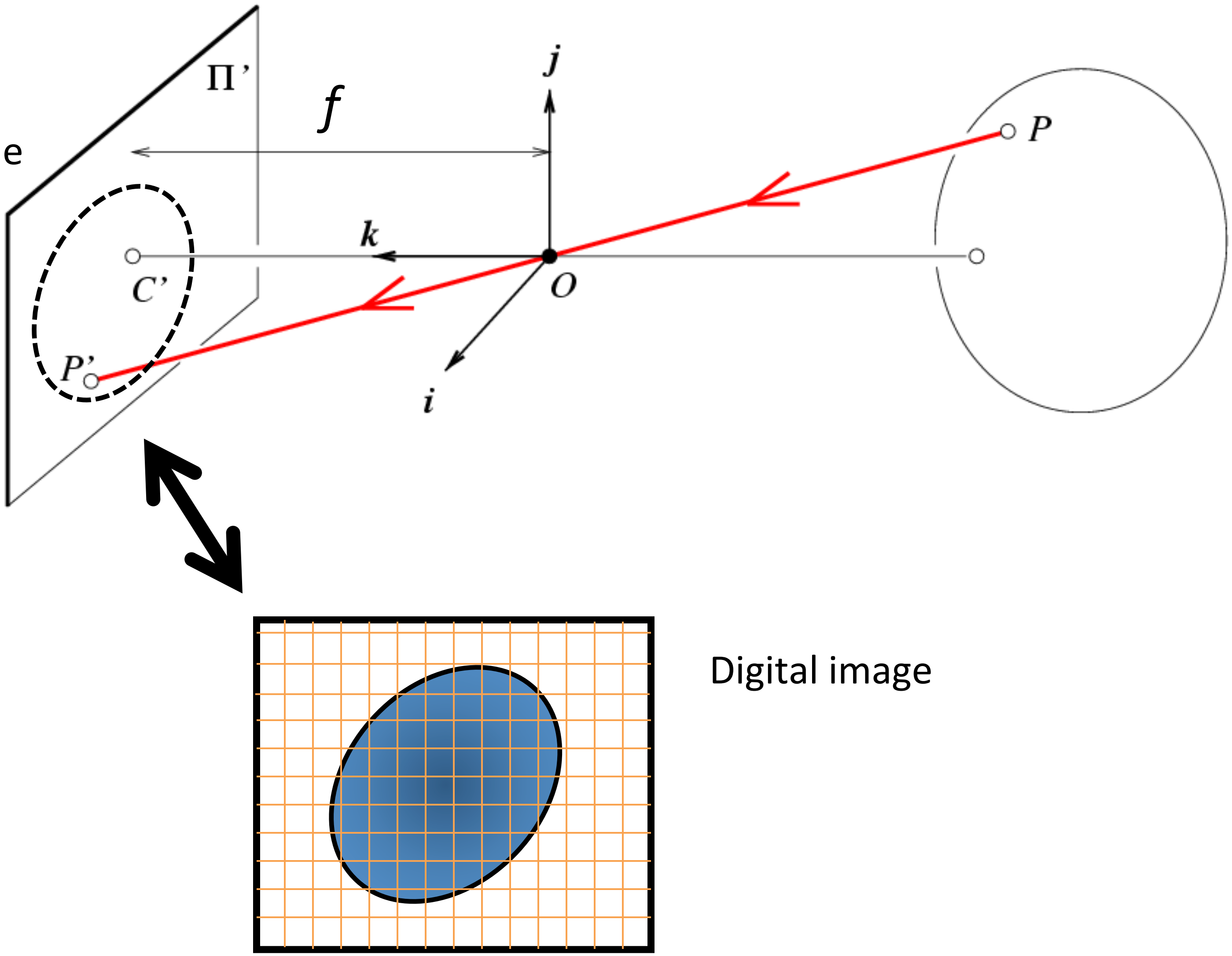
$$\begin{cases} x' = f \dfrac{x}{z} \\ y' = f \dfrac{y}{z} \end{cases}$$

$$\Re^3 \xrightarrow{E} \Re^2$$

f = focal length
o = center of the camera

From retina plane to images

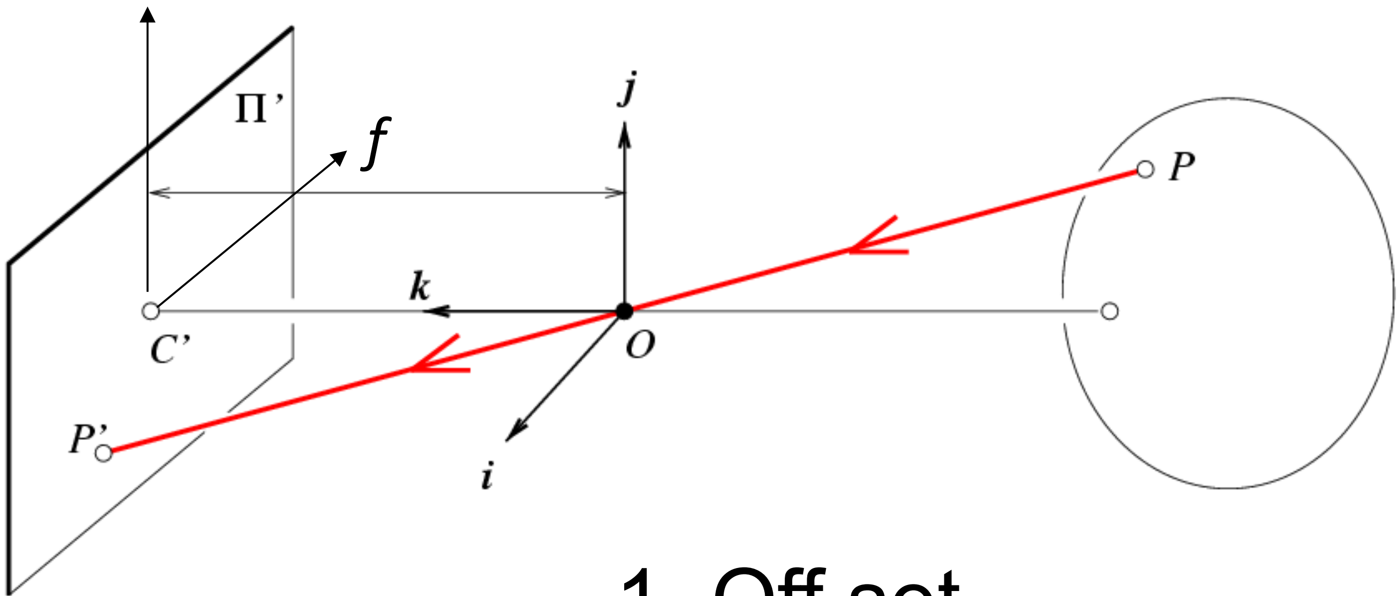
Retina plane



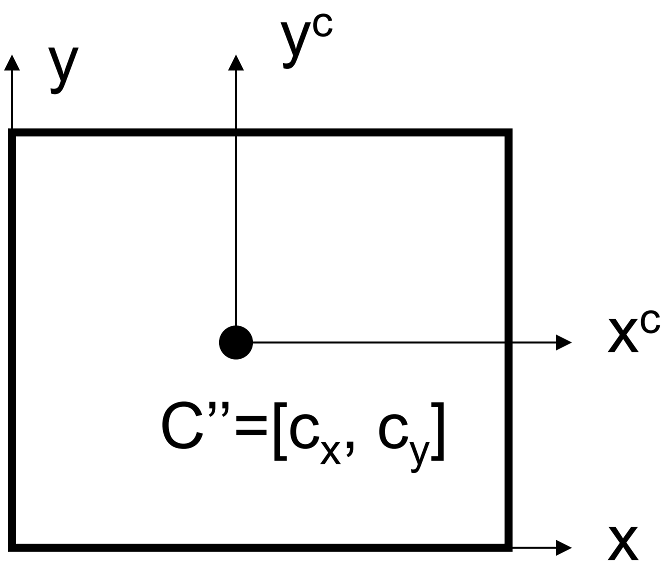
Digital image

Pixels, bottom-left coordinate systems

Coordinate systems

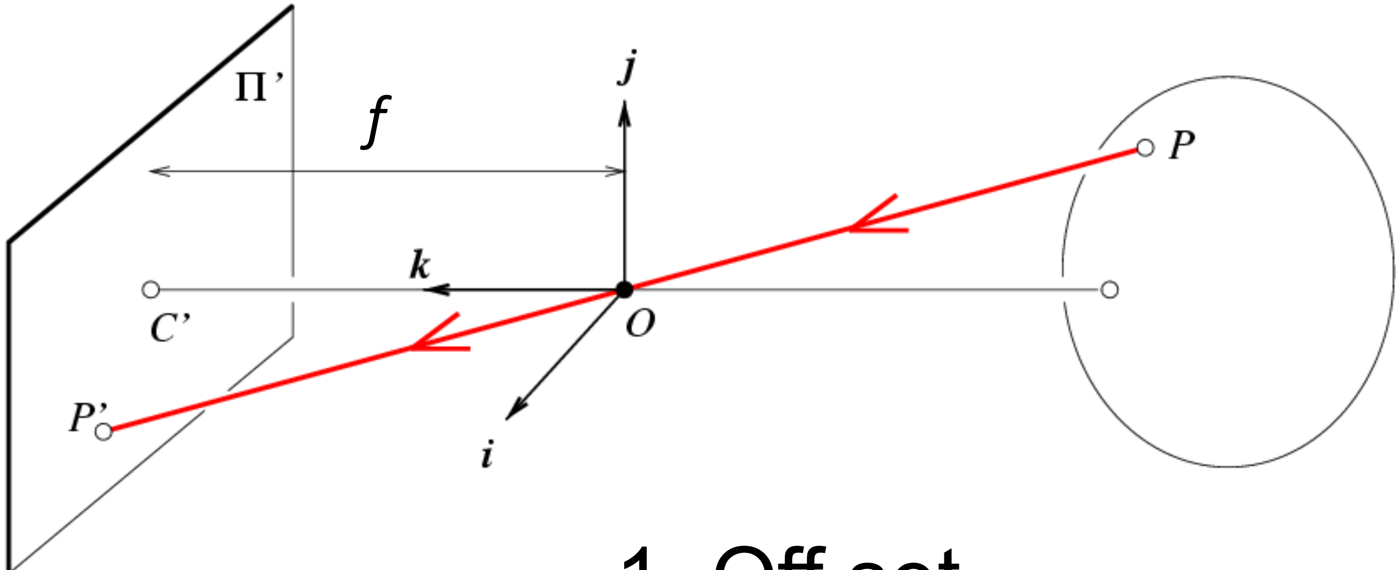


1. Off set

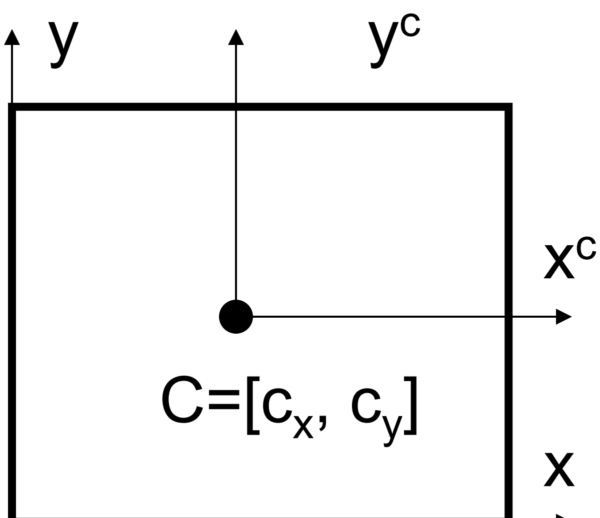


$$(x, y, z) \rightarrow \left(f\frac{x}{z} + c_x, f\frac{y}{z} + c_y\right)$$

Converting to pixels



1. Off set
2. From metric to pixels



$$(x, y, z) \rightarrow \left(\underbrace{\boxed{f\,k}}_{\alpha}\frac{x}{z} + c_x,\ \underbrace{\boxed{f\,l}}_{\beta}\frac{y}{z} + c_y\right)$$

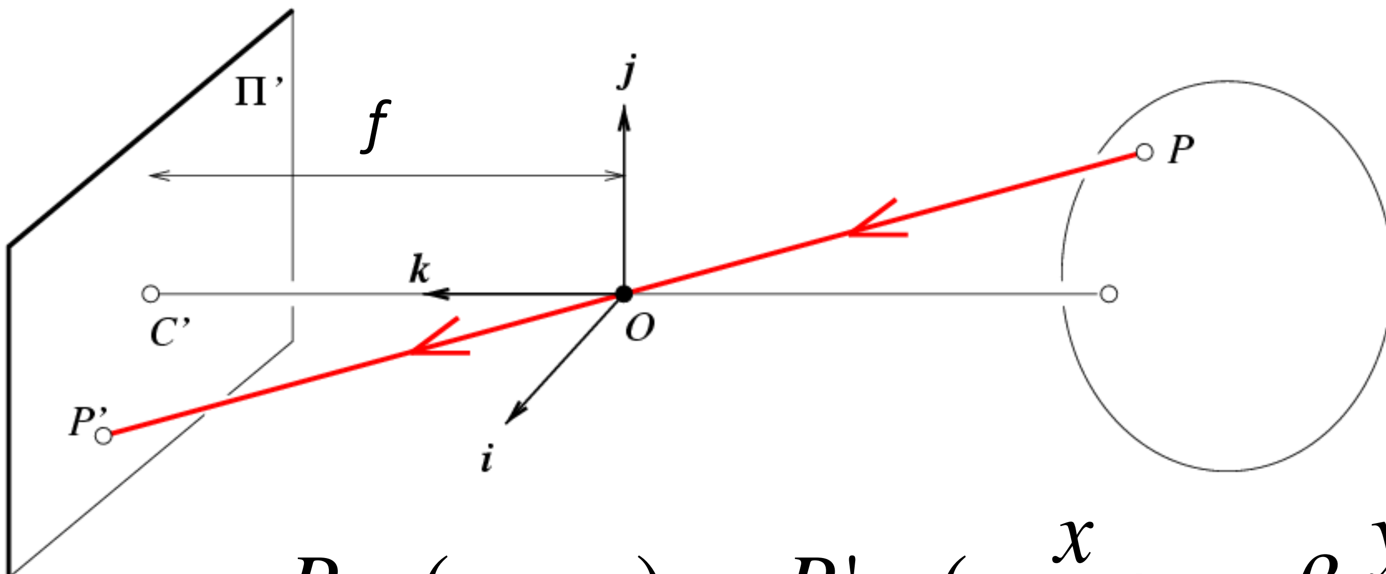
Units: k,l : pixel/m

f : m

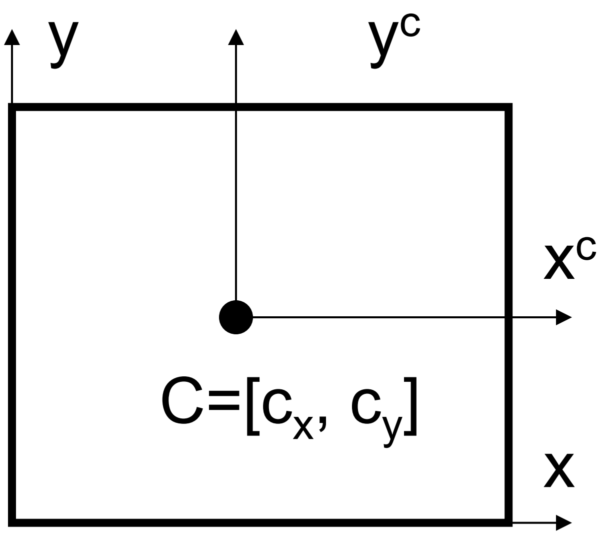
Non-square pixels

α, β : pixel

Converting to pixels



$$P = (x, y, z) \rightarrow P' = (\alpha \frac{x}{z} + c_x, \beta \frac{y}{z} + c_y)$$



- We can expressed it in a matrix form?

Homogeneous coordinates

E→H

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

homogeneous image coordinates

$$(x, y, z) \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

homogeneous scene coordinates

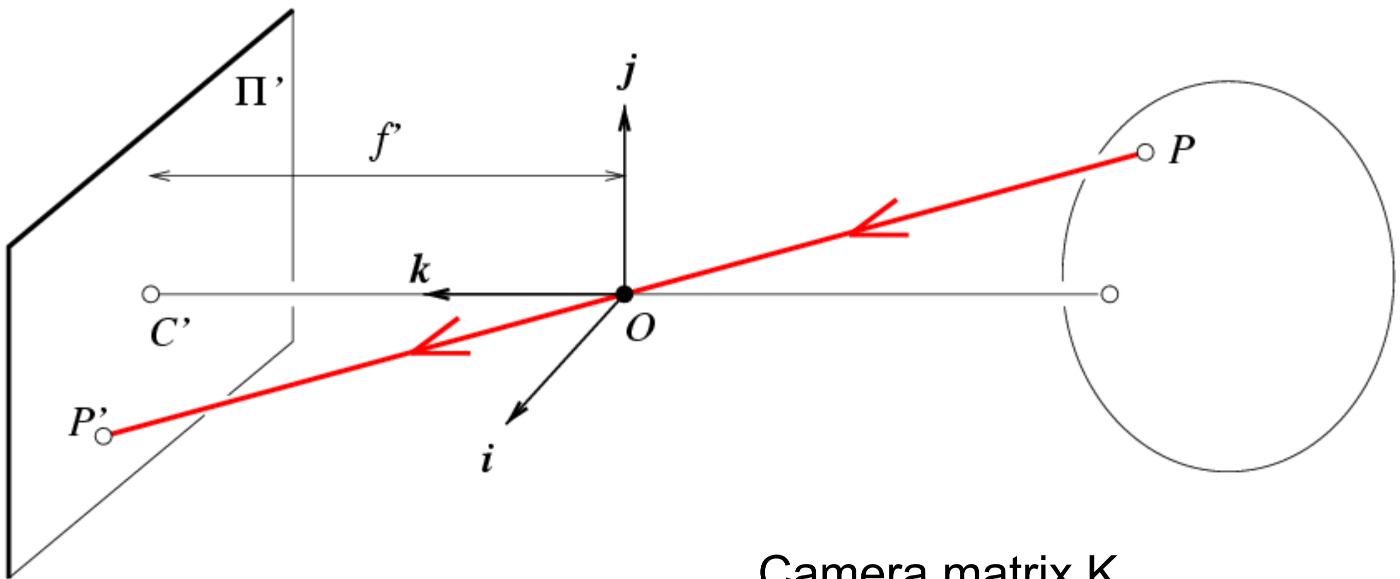
- Converting back *from* homogeneous coordinates

H→E

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow (x/w, y/w, z/w)$$

Camera Matrix



Camera matrix K

$$P' = M\,P$$

$$= K\begin{bmatrix} I & 0 \end{bmatrix} P$$

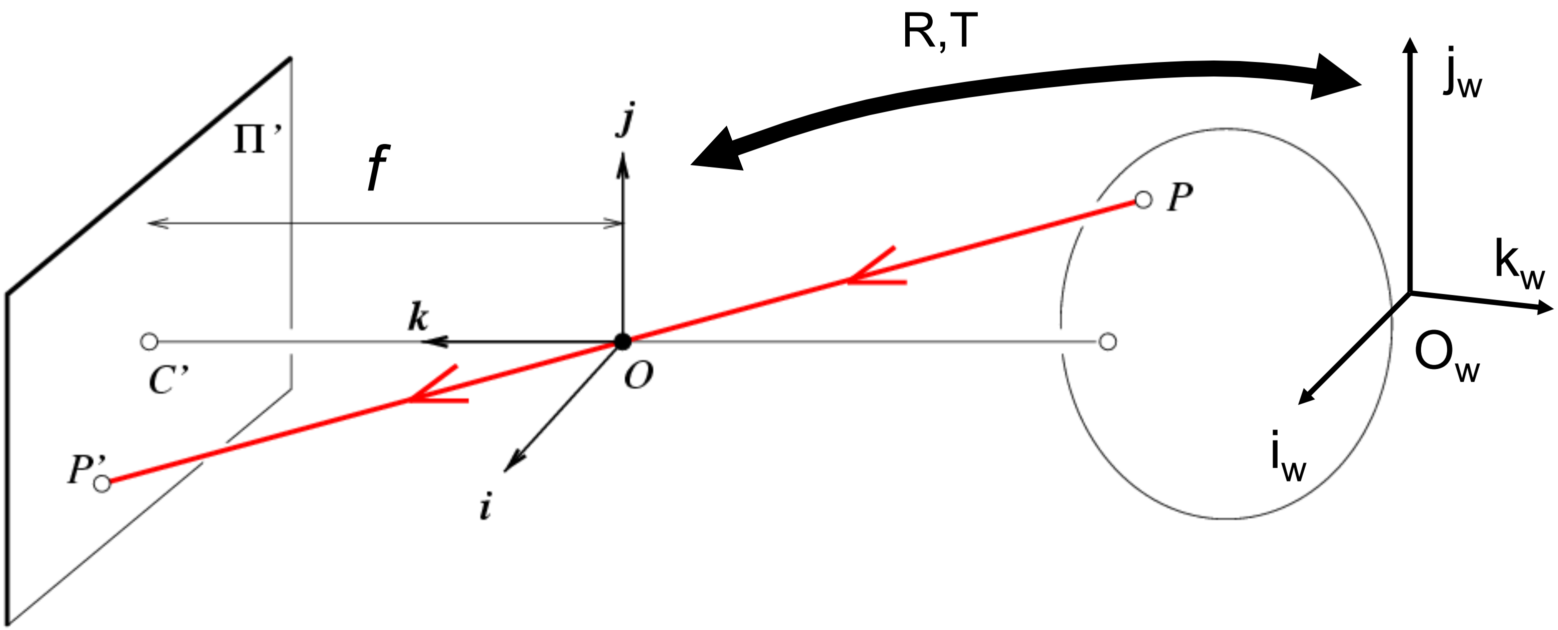
$$P' = \begin{bmatrix} \alpha\, x + c_x z \\ \beta\, y + c_y z \\ z \end{bmatrix} = \begin{bmatrix} \alpha & 0 & c_x & 0 \\ 0 & \beta & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\rightarrow \left(f\frac{x}{z}, f\frac{y}{z}\right)$$

With zero offset and unitary square pixels

For details see lecture 2 CS231A

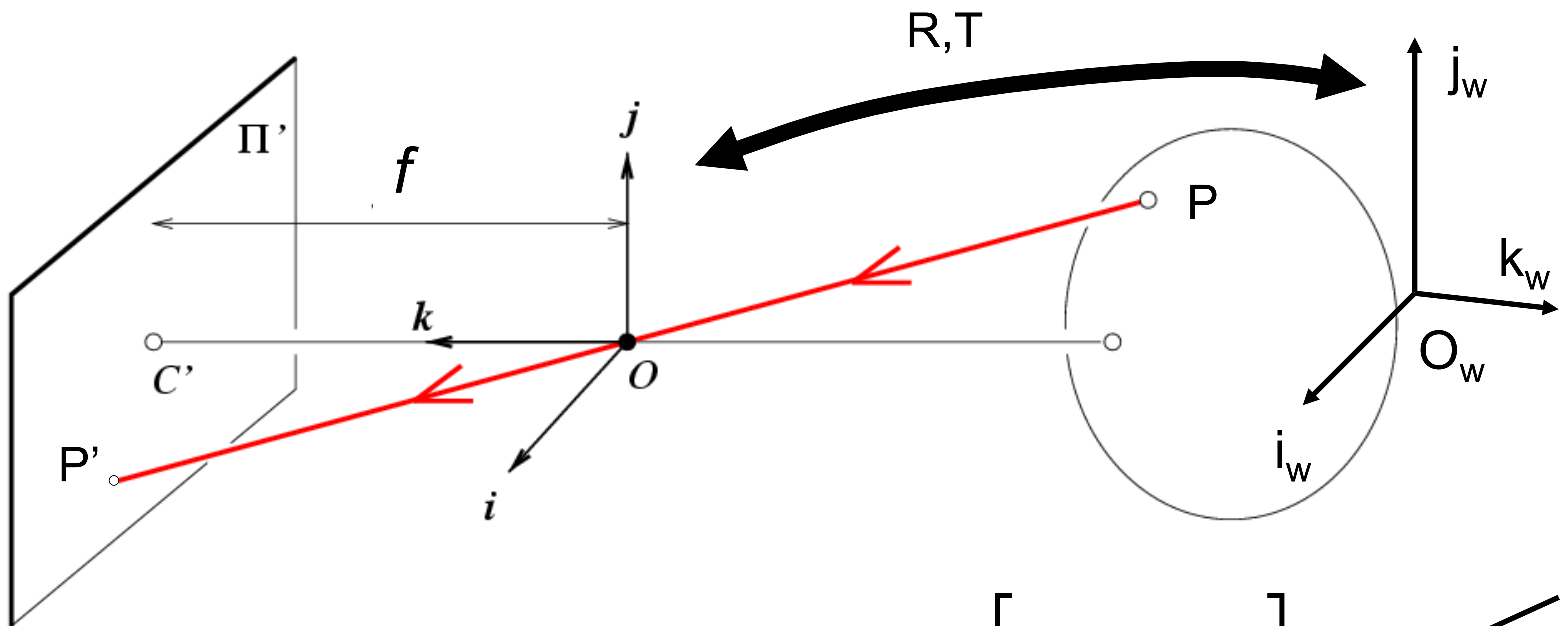
World reference system



•The mapping so far is defined within the camera reference system

• What if an object is represented in the world reference system

World reference system



In 4D homogeneous coordinates: $P = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}_{4\times4} P_w$, $\quad P_w = \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$

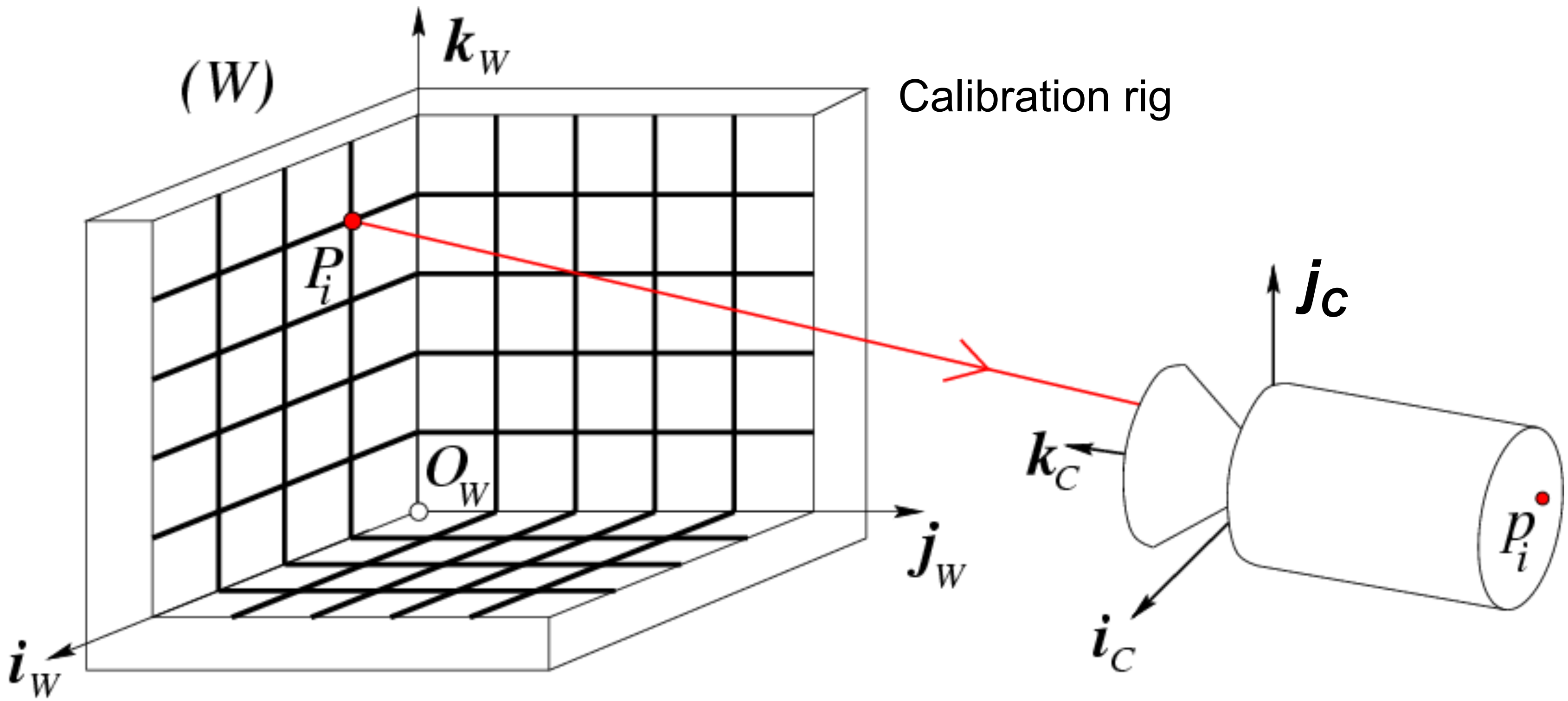
Internal parameters

External parameters

$$P' = K\begin{bmatrix} I & 0 \end{bmatrix} P = K\begin{bmatrix} I & 0 \end{bmatrix}\begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}_{4\times4} P_w = \underbrace{\boxed{K\begin{bmatrix} R & T \end{bmatrix}}}_{M} P_w$$

[Eq.11]

Camera Calibration

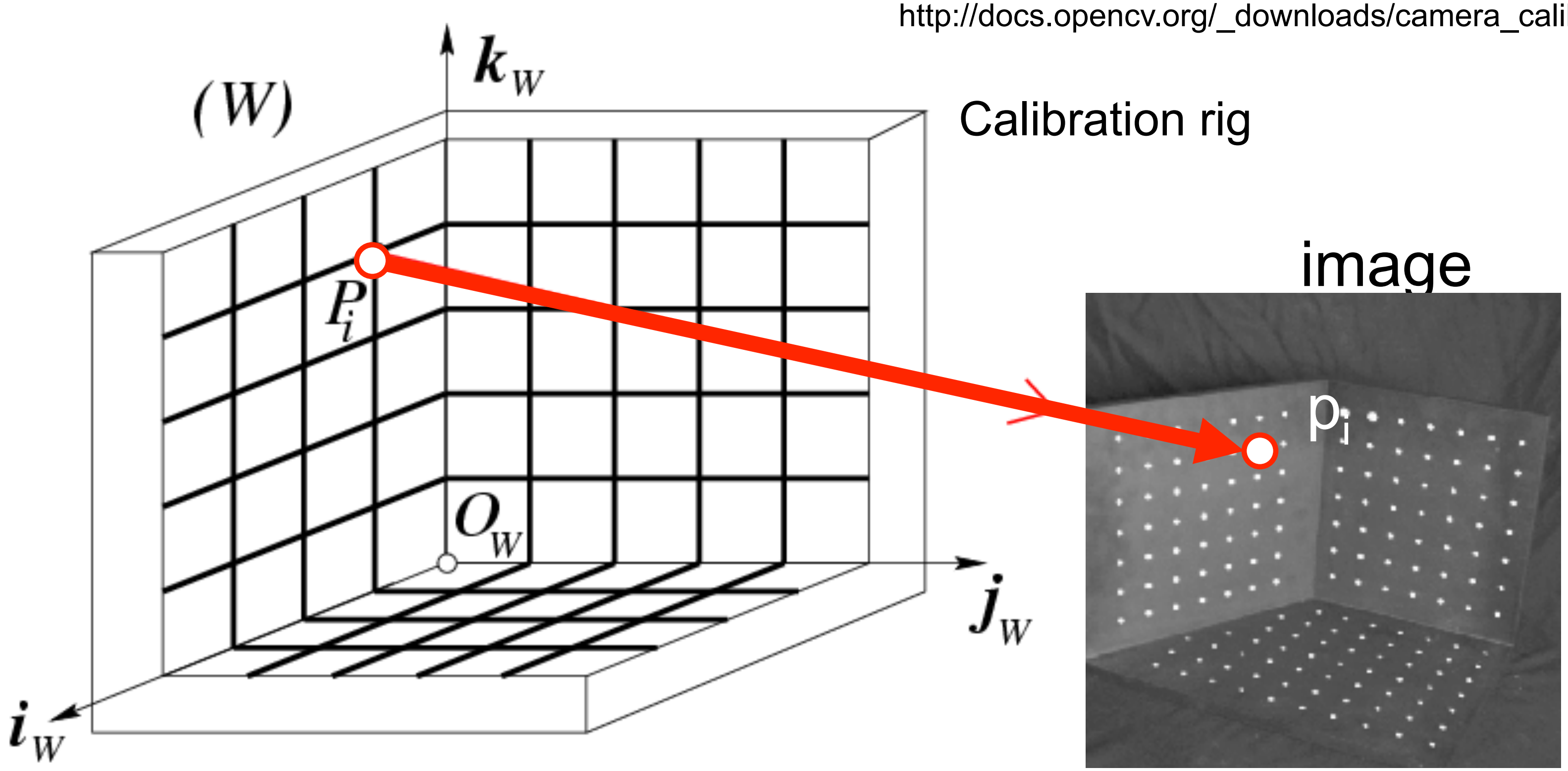


$$P' = K\begin{bmatrix} I & 0 \end{bmatrix} P = K\begin{bmatrix} R & T \end{bmatrix} P_w$$

- $P_1 \ldots P_n$ with known positions in $[O_w, i_w, j_w, k_w]$
- $p_1, \ldots p_n$ known positions in the image

Camera Calibration

http://docs.opencv.org/_downloads/camera_calibration.cpp



$$P' = K\begin{bmatrix} I & 0 \end{bmatrix} P = K\begin{bmatrix} R & T \end{bmatrix} P_w$$

- $P_1 \ldots P_n$ with known positions in $[O_w, i_w, j_w, k_w]$
- $p_1, \ldots p_n$ known positions in the image

Goal: compute intrinsic and extrinsic parameters

For details see lecture 3 CS231A

Properties of Projection

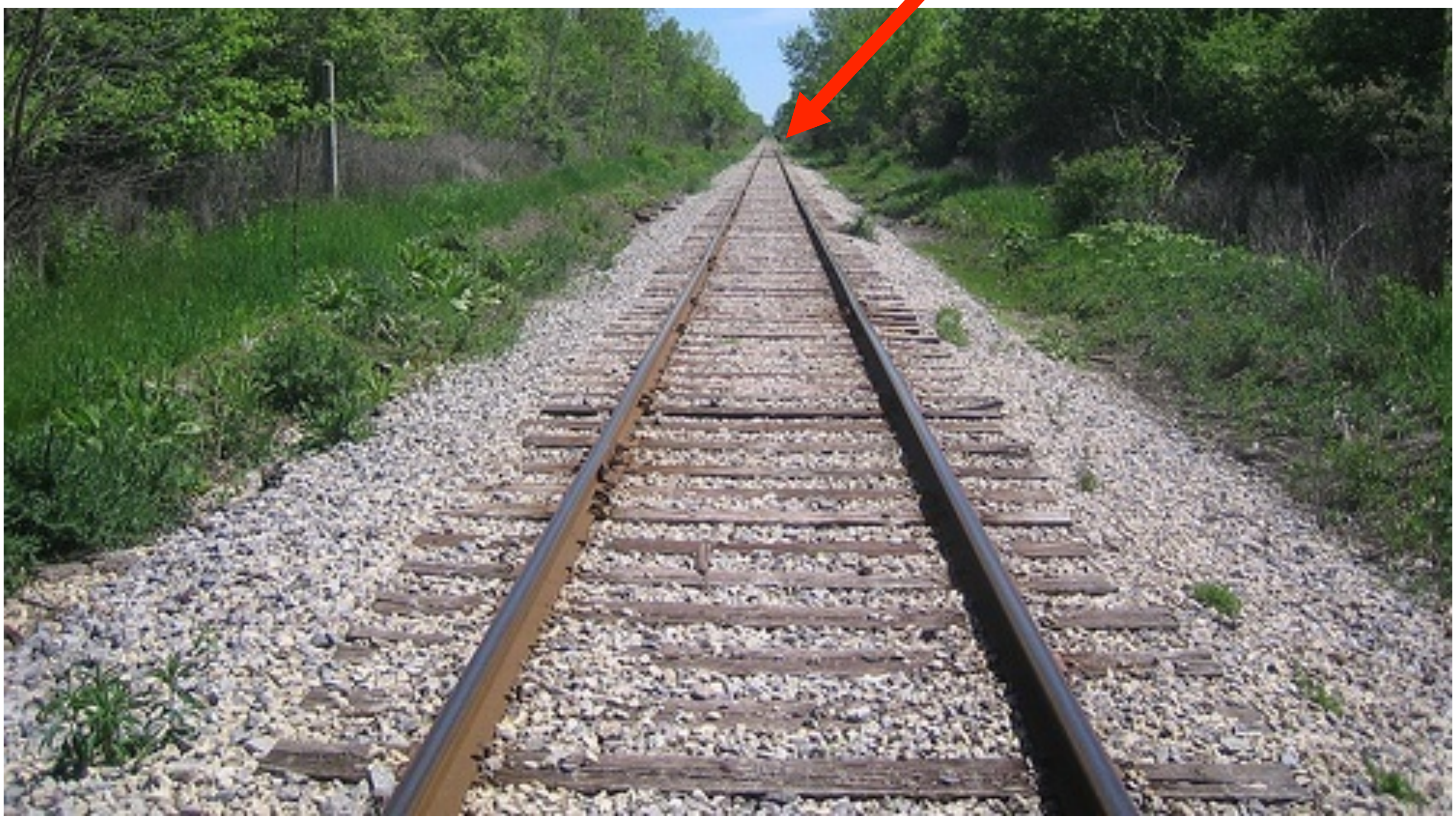
- Points project to points
- Lines project to lines
- Distant objects look smaller

$$P_w \rightarrow MP_w \rightarrow p$$

Properties of Projection

- Angles are not preserved
- Parallel lines meet!

Vanishing point

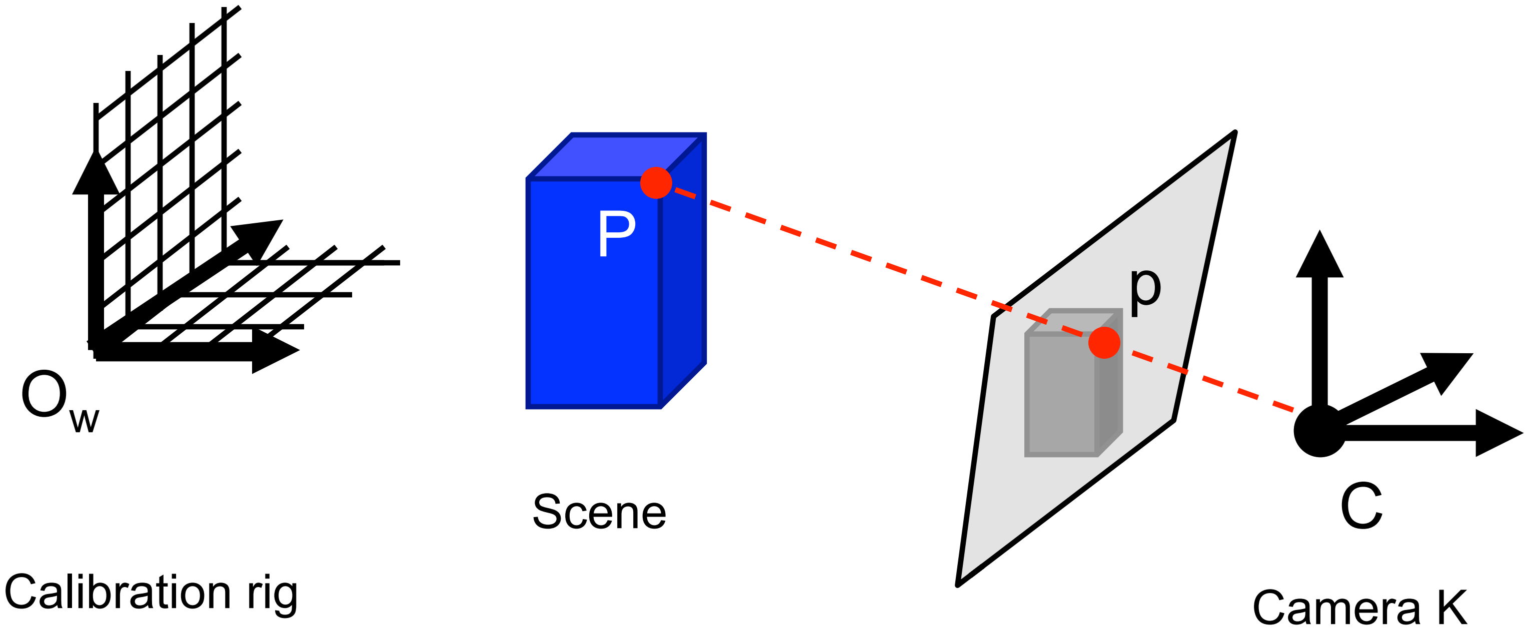


Lecture 11

Inferring 3D geometry from images

- Cameras
- Single view metrology
- Epipolar geometry
- Structure from motion

Can we recover the structure from a single view?



Why is it so difficult?

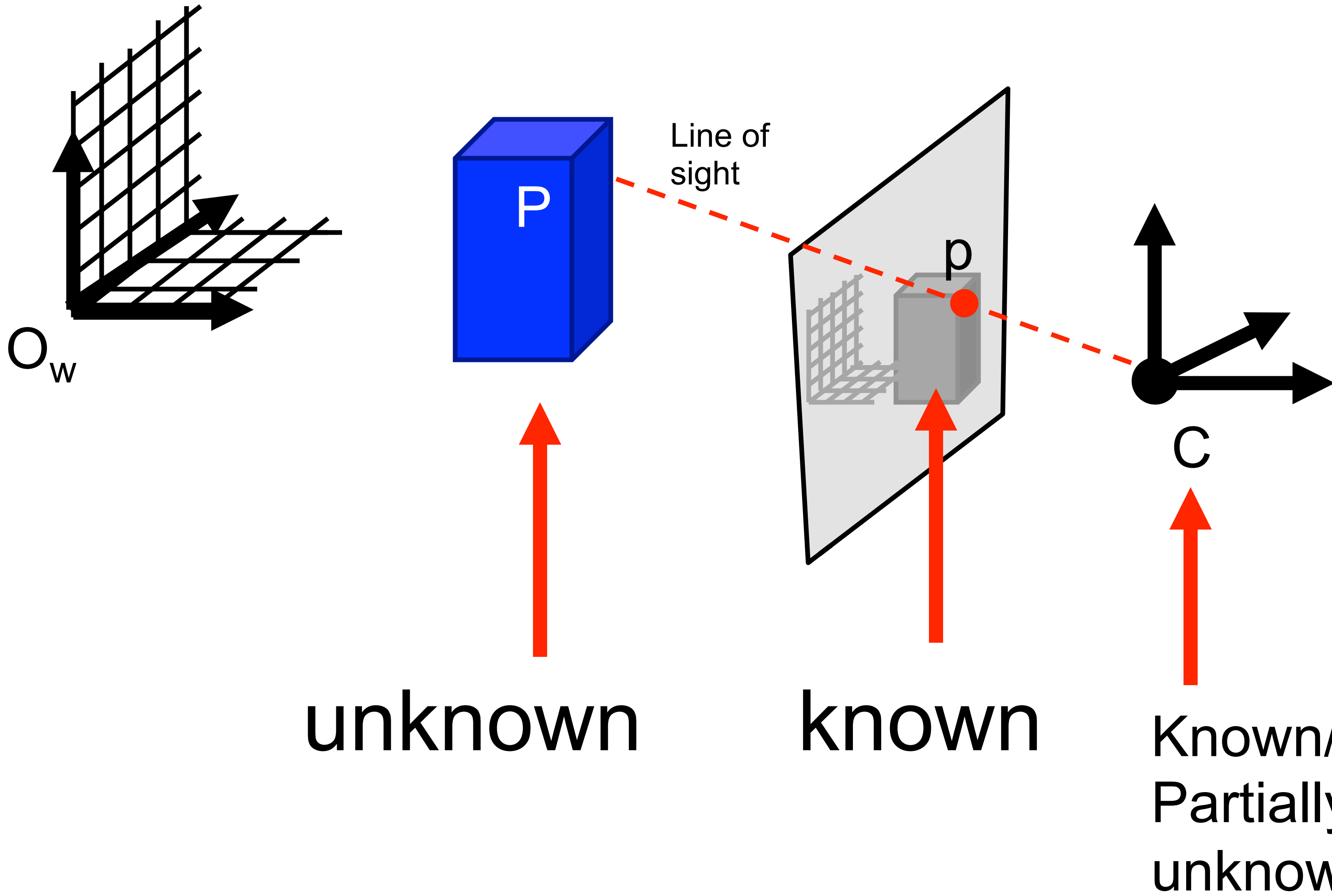
Intrinsic ambiguity of the mapping from 3D to image (2D)

Can we recover the structure from a single view?

Intrinsic ambiguity of the mapping from 3D to image (2D)

Courtesy slide S. Lazebnik

Recovering structure from a single view

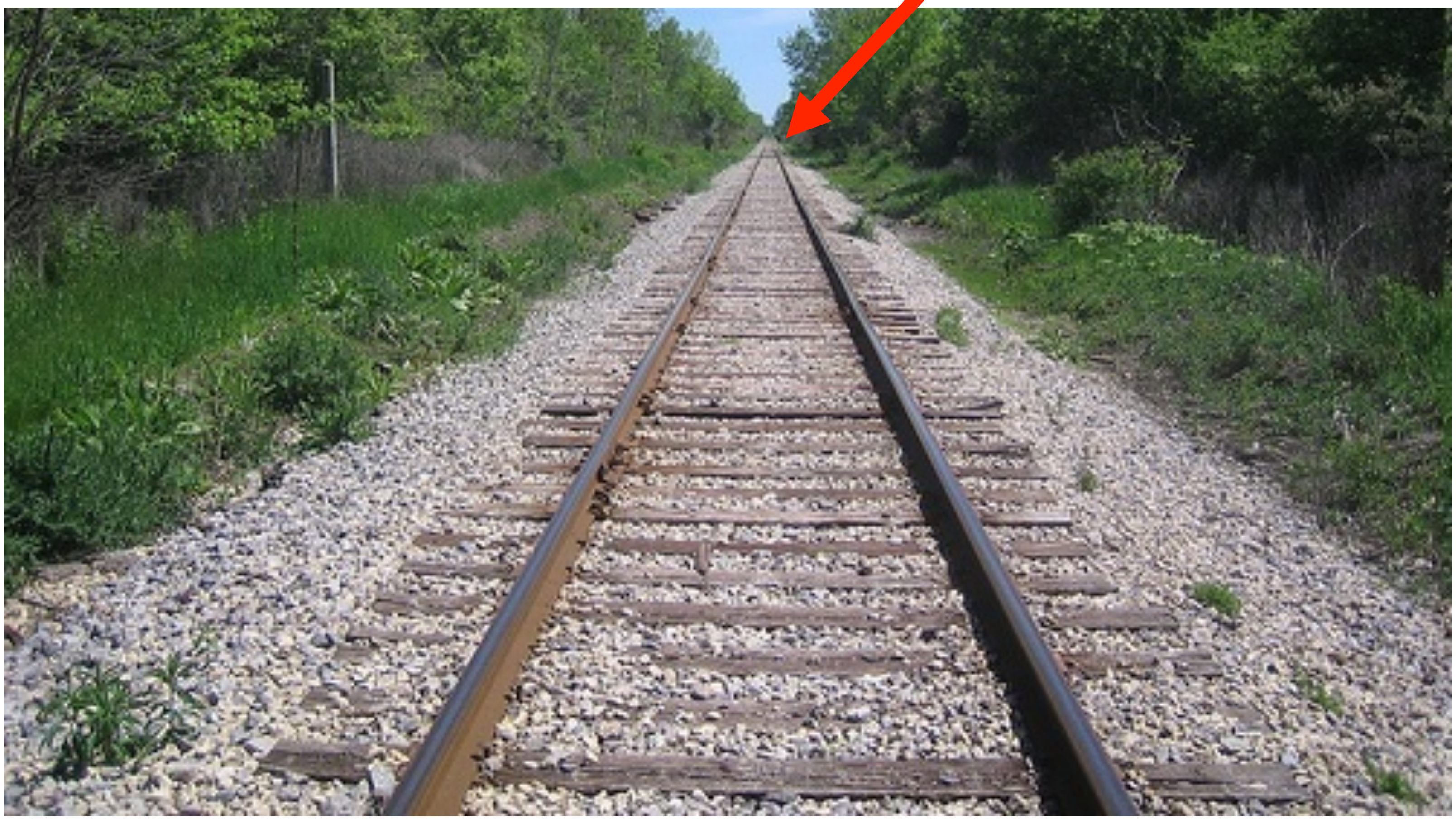


Prior knowledge about the environment helps infer 3D geometry!

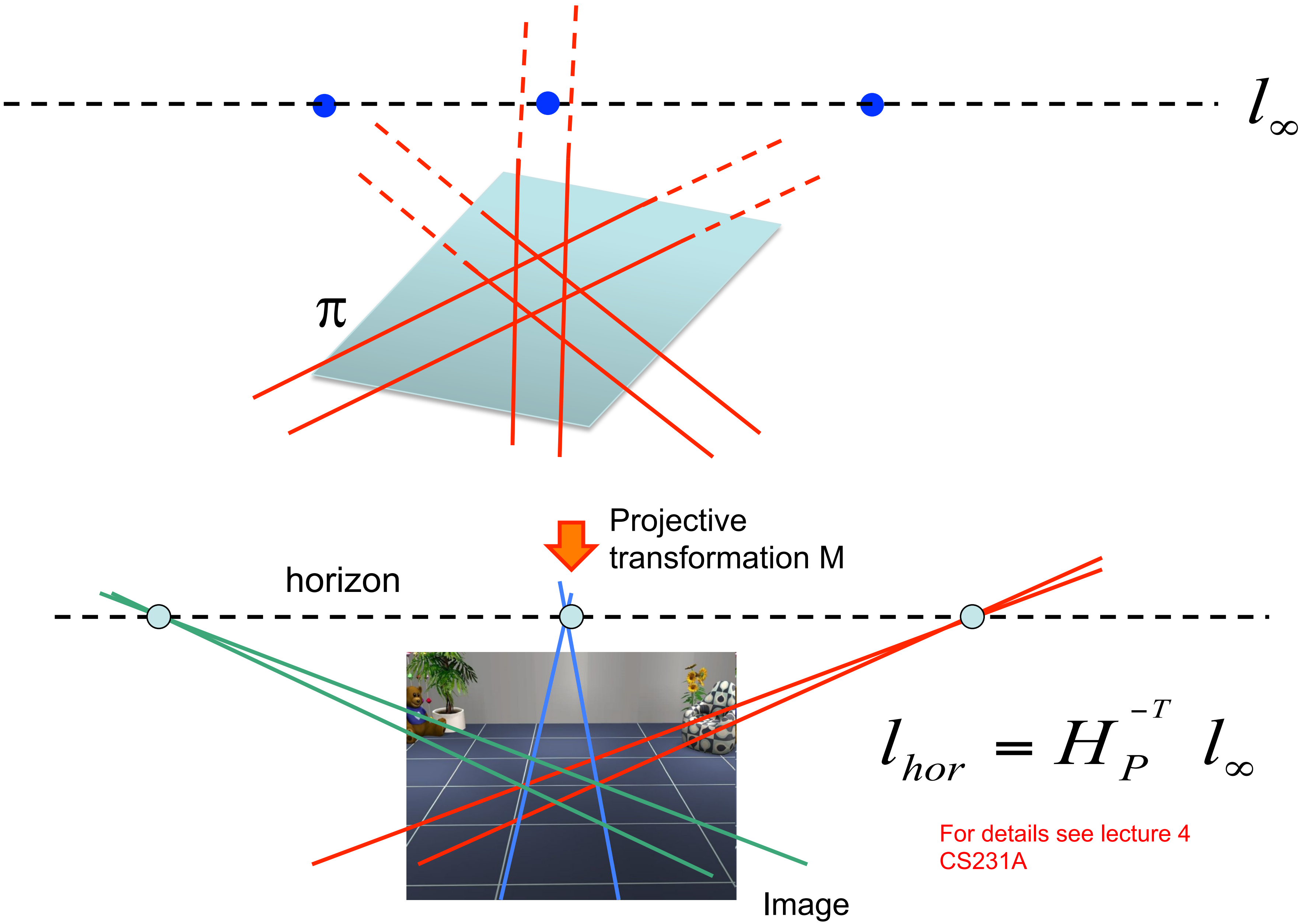
Properties of Projection

- Angles are not preserved
- Parallel lines meet!

Vanishing point



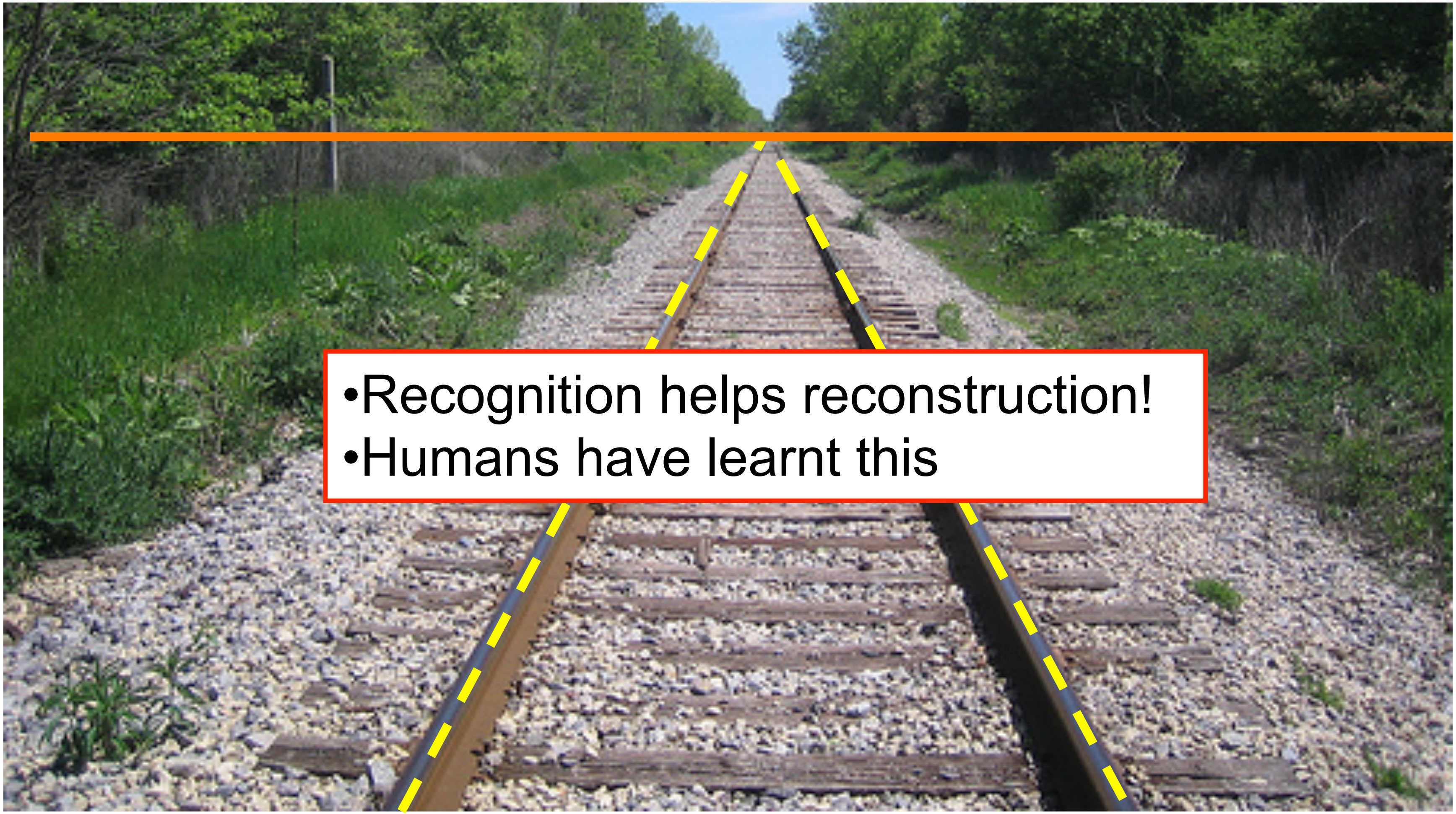
Vanishing (horizon) line



$$l_{hor} = H_P^{-T}\ l_\infty$$

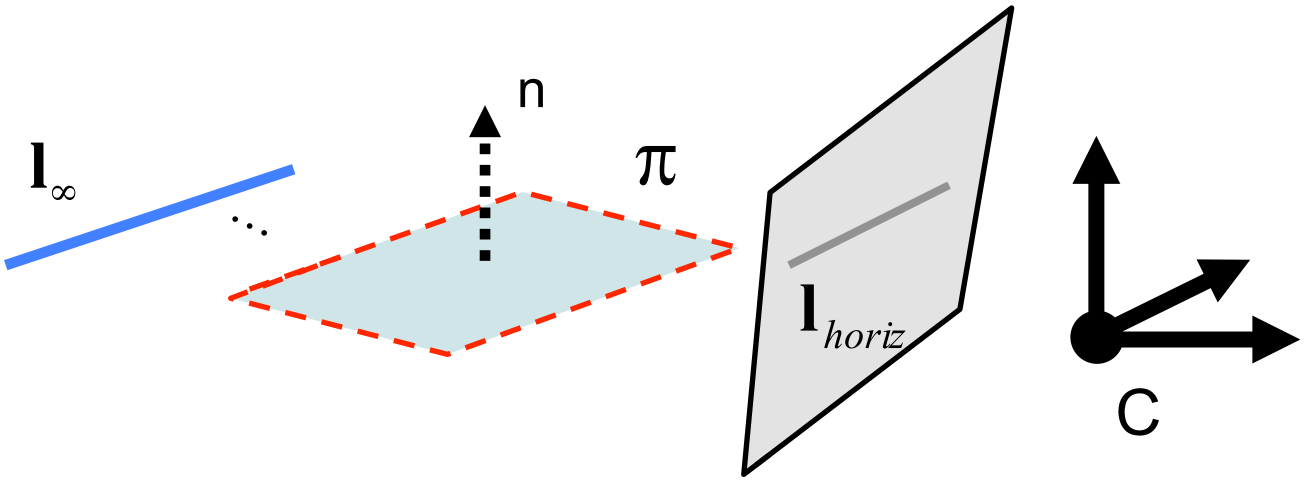
For details see lecture 4
CS231A

Example: Are these two lines parallel or not?

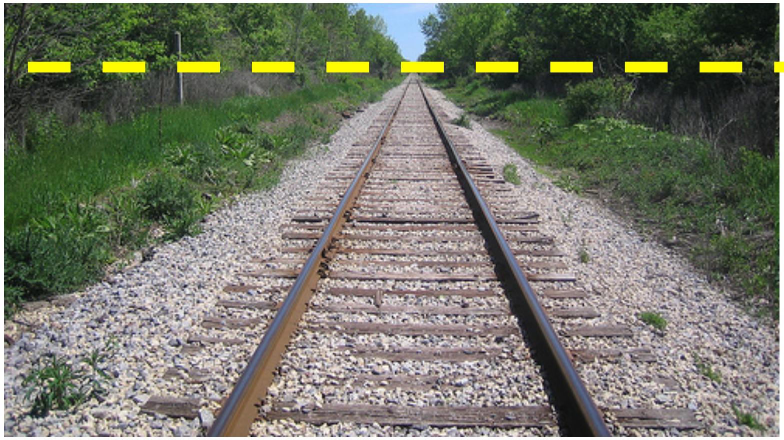


- Recognize the horizon line
- Measure if the 2 lines meet at the horizon
- if yes, these 2 lines are // in 3D

Vanishing points and planes



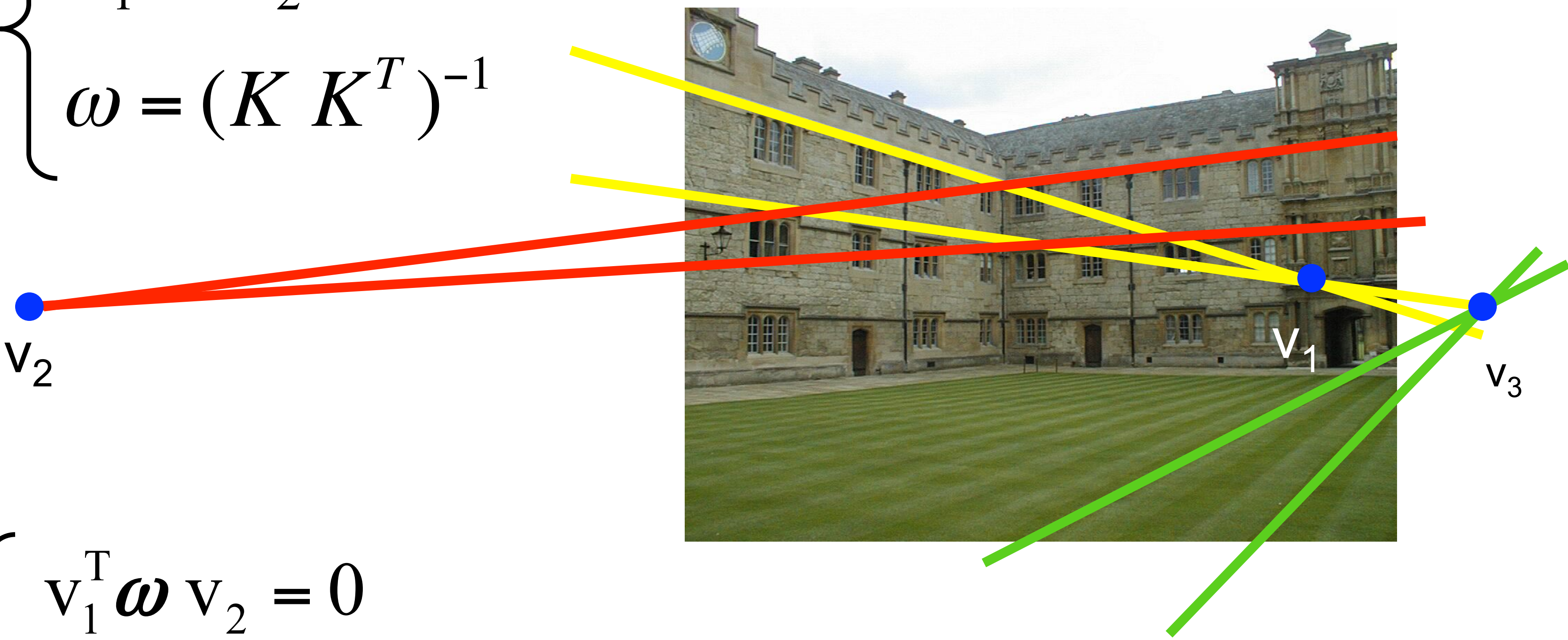
$$\mathbf{n} = K^T \mathbf{l}_{\text{horiz}}$$



Vanishing points and planes

For details see lecture 4
CS231A

$$\begin{cases} v_1^T \boldsymbol{\omega}\, v_2 = 0 \\ \omega = (K\,K^T)^{-1} \end{cases}$$

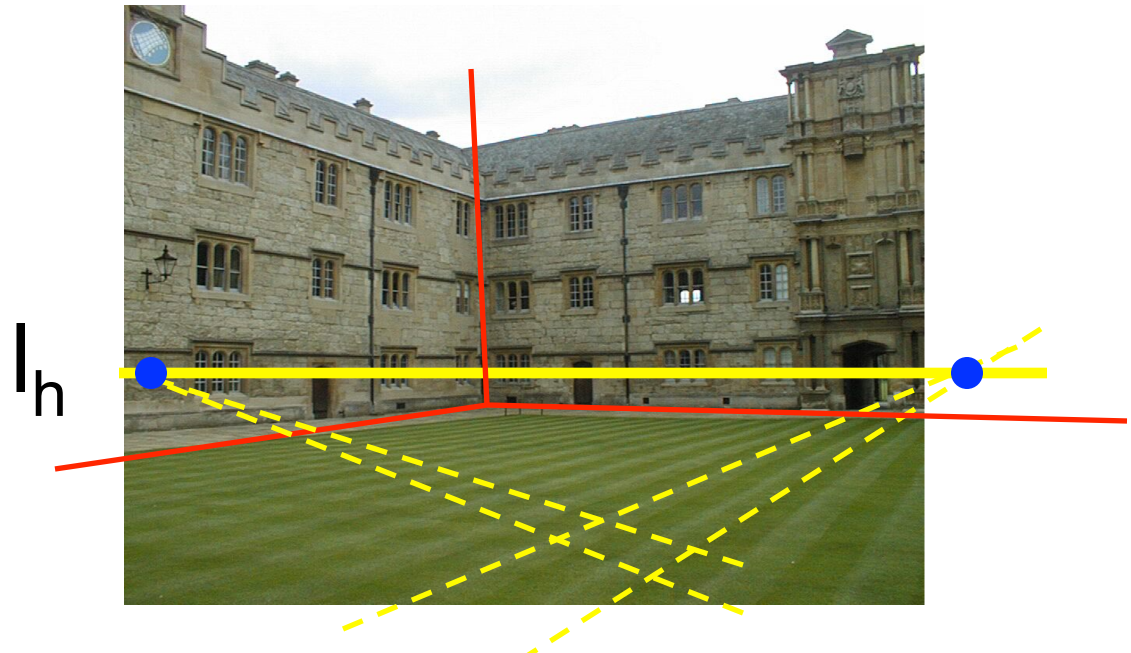


$$\begin{cases} v_1^T \boldsymbol{\omega}\, v_2 = 0 \\ v_1^T \boldsymbol{\omega}\, v_3 = 0 \\ v_2^T \boldsymbol{\omega}\, v_3 = 0 \end{cases}$$

→ Set up a system of equations that allows to compute K

Vanishing points and planes

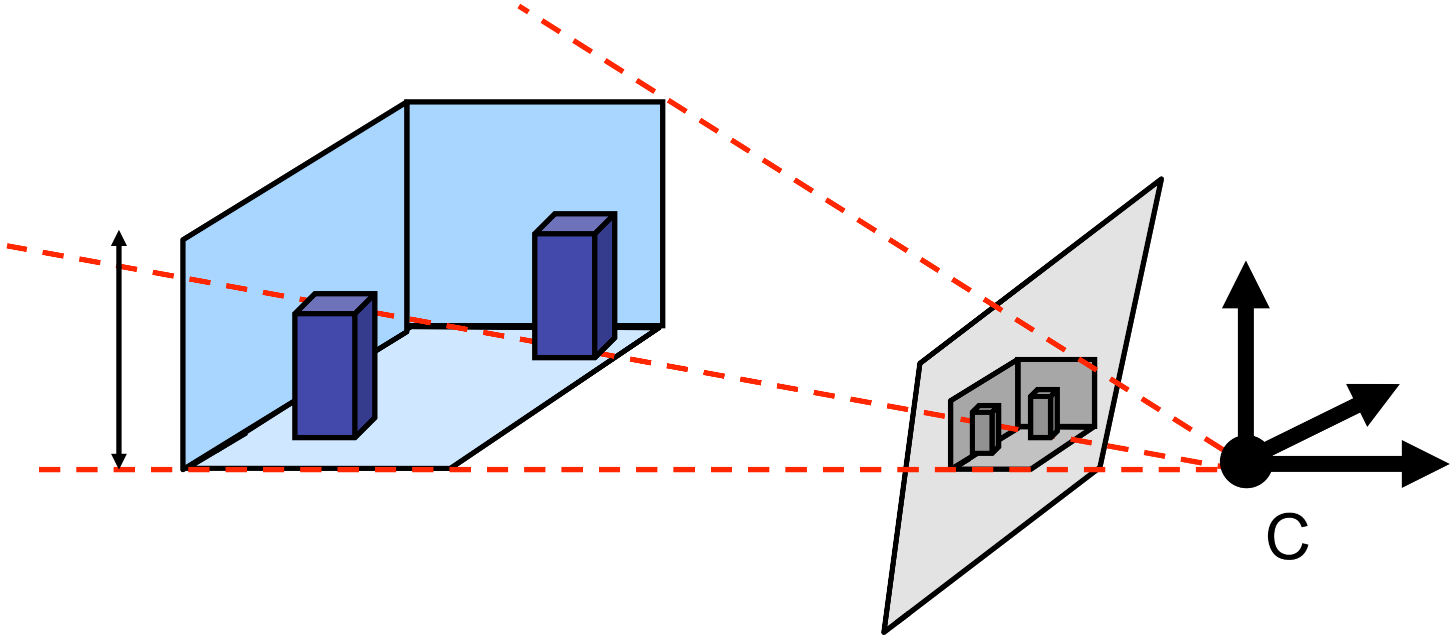
For details see lecture 4
CS231A



K known $\longrightarrow$ $\mathbf{n} = K^T \mathbf{l}_{horiz}$ = Scene plane orientation in the camera reference system

Select orientation discontinuities

Single view reconstruction - example



Recover the structure within the camera reference system

Notice: the actual scale of the scene is NOT recovered

- Recognition helps reconstruction!
- Humans have learnt this

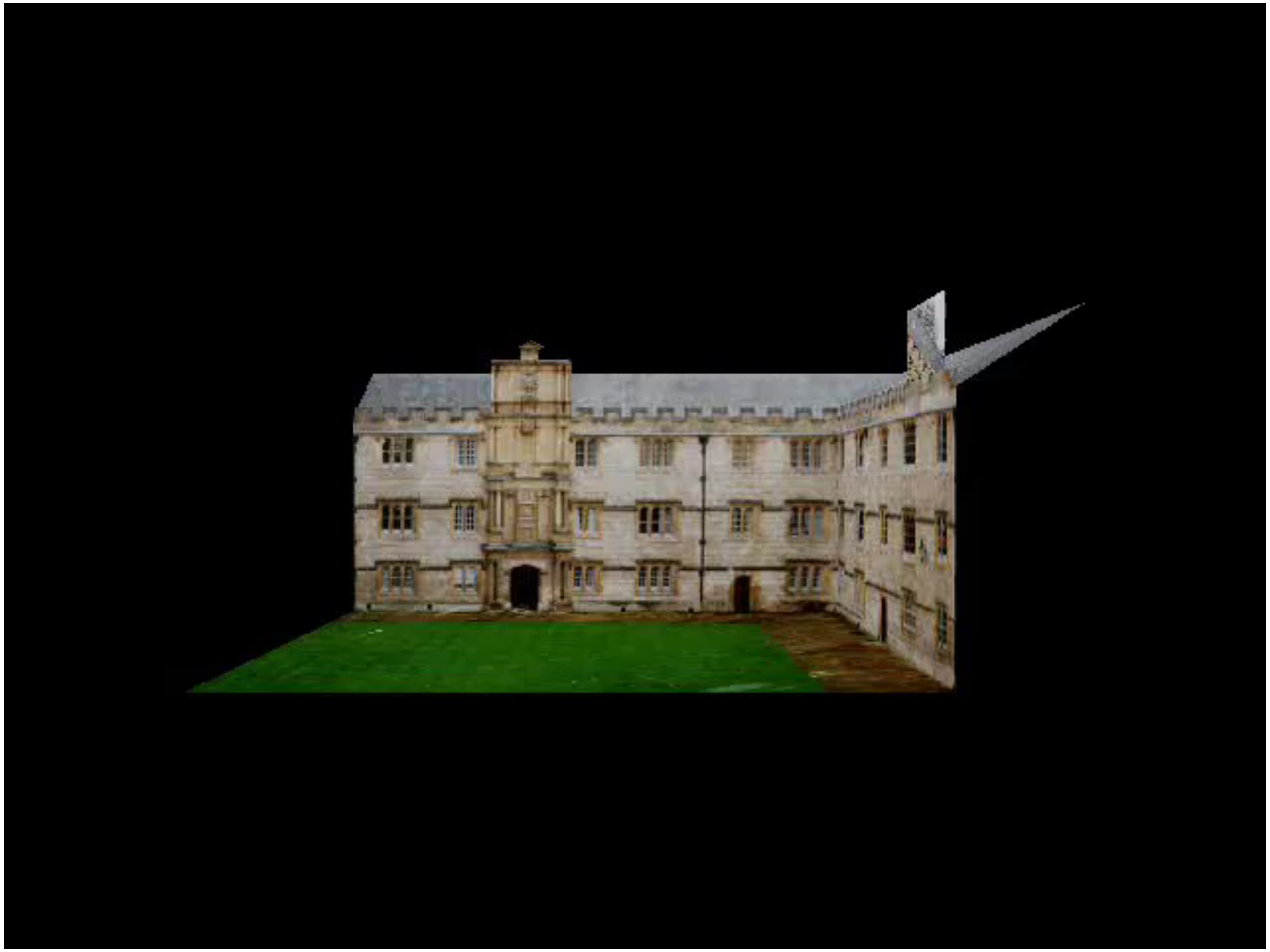
Recovering structure from a single view

http://www.robots.ox.ac.uk/~vgg/projects/SingleView/models/hut/hutme.wrl

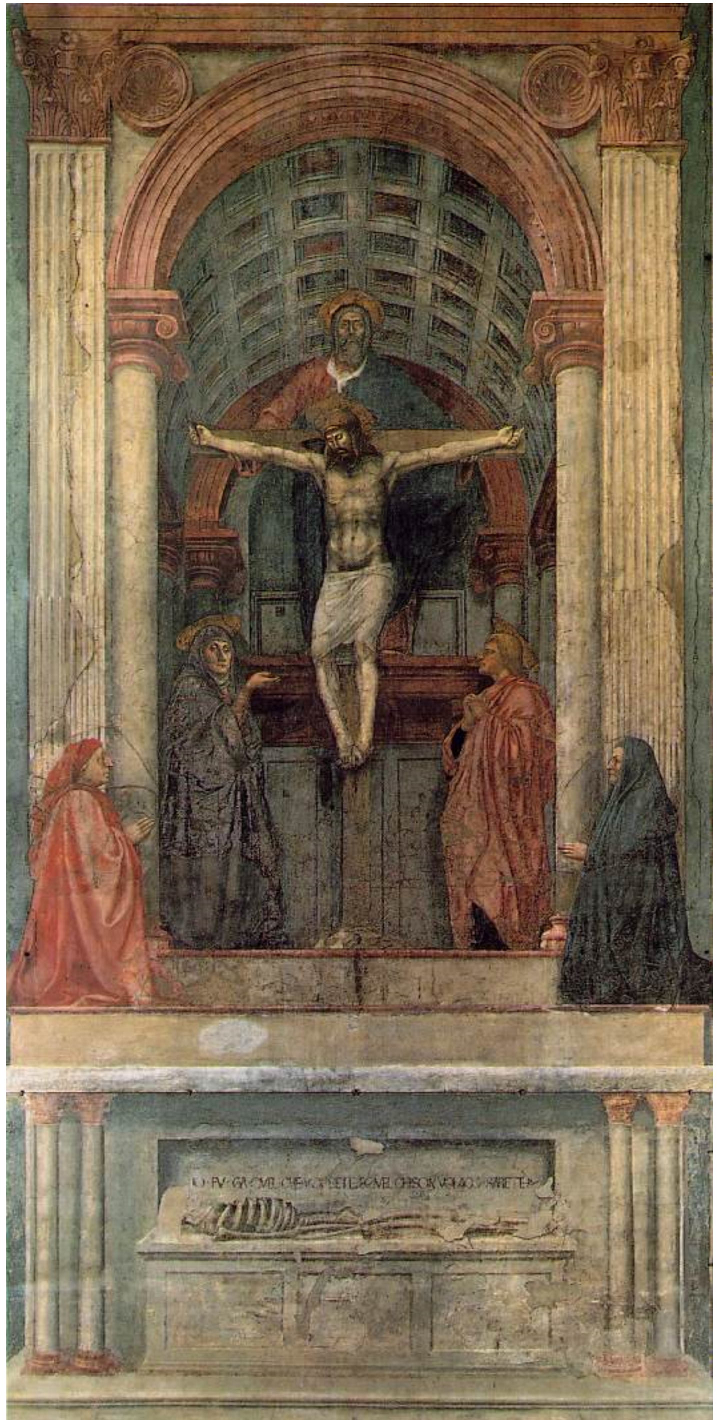
Criminisi & Zisserman, 99

http://www.robots.ox.ac.uk/~vgg/projects/SingleView/models/merton/merton.wrl

Criminisi & Zisserman, 99



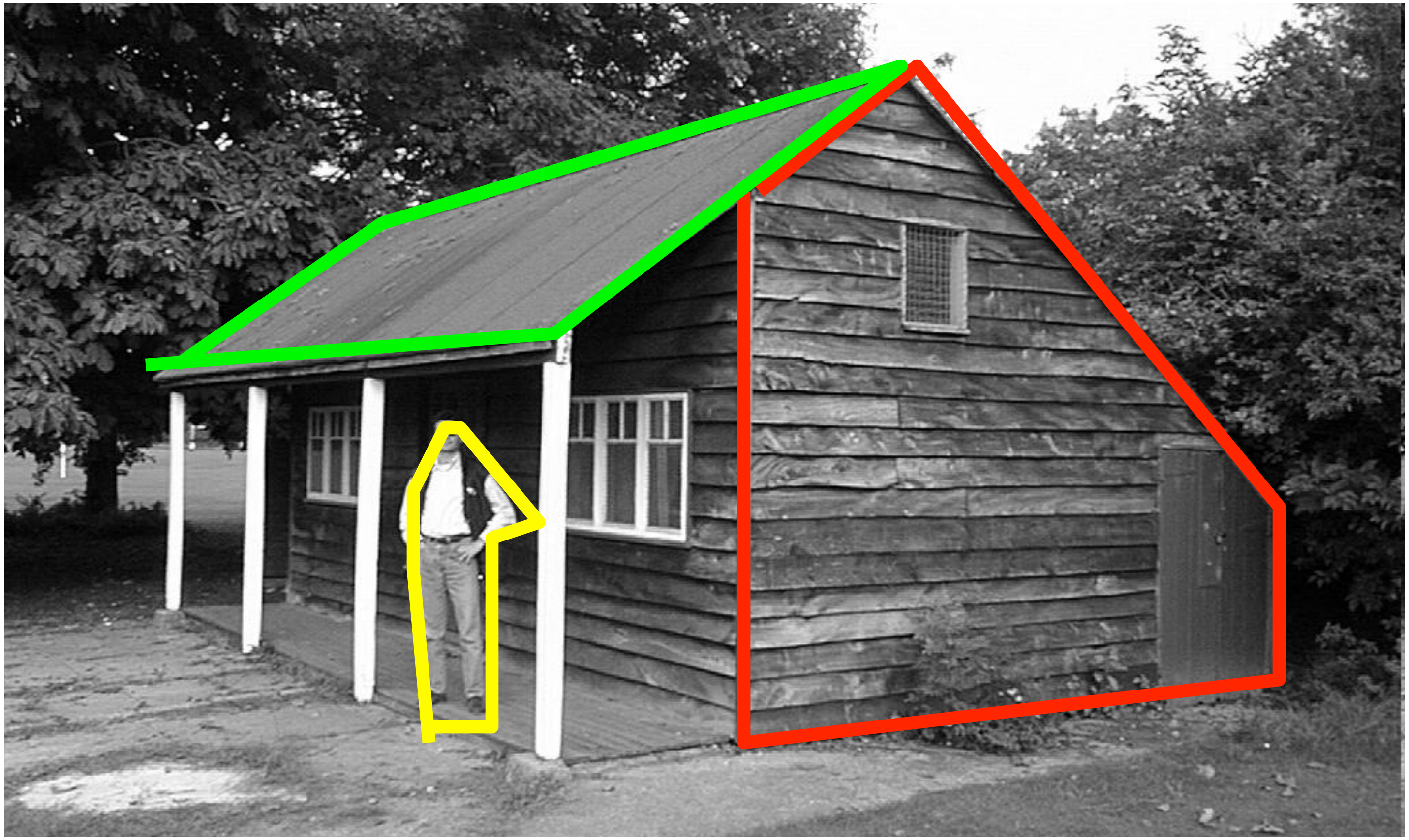
http://www.robots.ox.ac.uk/~vgg/projects/SingleView/models/merton/merton.wrl



La Trinita' (1426)
Firenze, Santa Maria
Novella; by Masaccio
(1401-1428)

La Trinita' (1426)
Firenze, Santa Maria
Novella; by Masaccio
(1401-1428)

Single view reconstruction - drawbacks

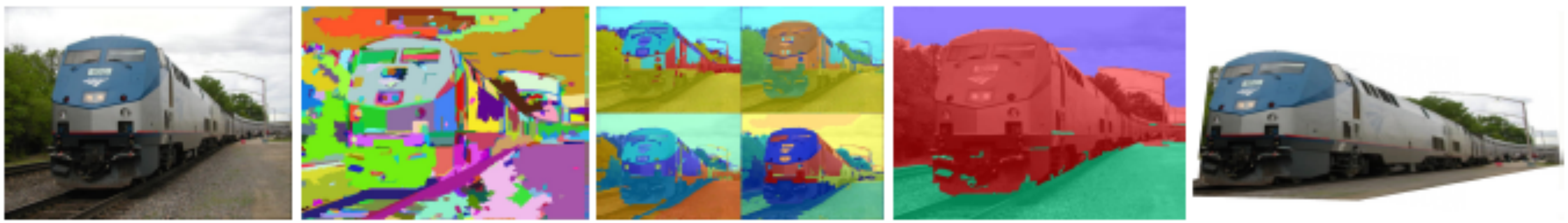


Manually select:

- Vanishing points and lines;
- Planar surfaces;
- Occluding boundaries;
- Etc..

Automatic Photo Pop-up

Hoiem et al, 05



Automatic Photo Pop-up

Hoiem et al, 05…

Automatic Photo Pop-up

Hoiem et al, 05…

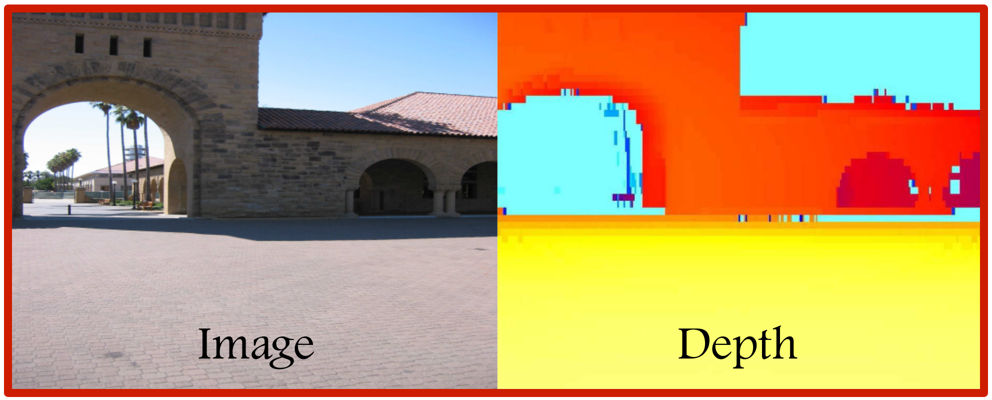
Software:

http://www.cs.uiuc.edu/homes/dhoiem/projects/software.html

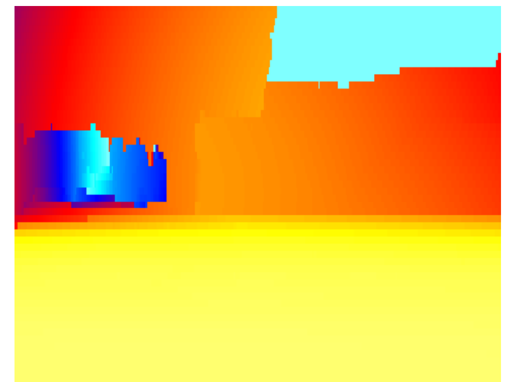
Make3D

Saxena, Sun, Ng, 05…

Training

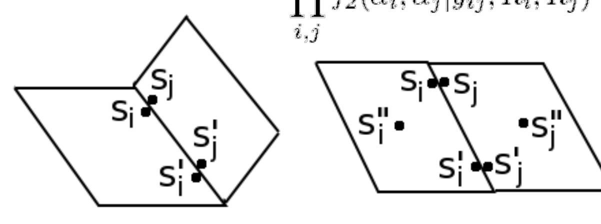


Prediction



Plane Parameter MRF

$$P(\alpha|X,\nu,y,R;\theta) = \frac{1}{Z}\prod_i f_1(\alpha_i|X_i,\nu_i,R_i;\theta) \prod_{i,j} f_2(\alpha_i,\alpha_j|y_{ij},R_i,R_j)$$



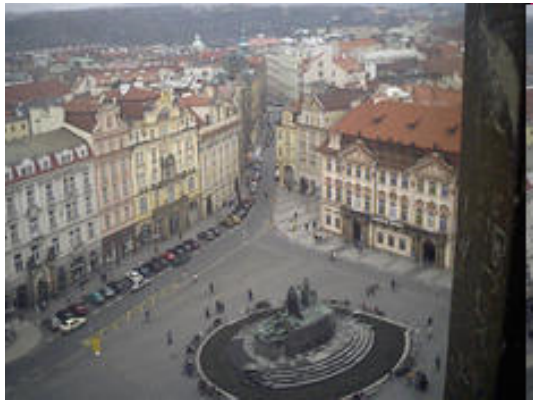
(a) Connectivity

(b) Co-Planarity

youtube

Single Image Depth Reconstruction

Saxena, Sun, Ng, 05…


A software: **Make3D**
"Convert your image into 3d model"

http://make3d.stanford.edu/

http://make3d.cs.cornell.edu/

Room layout estimation

Varsha Hedau, Derek Hoiem, David Forsyth, “Recovering the Spatial Layout of Cluttered Rooms,” in the Twelfth IEEE International Conference on Computer Vision, 2009.

Also: Alexander G. Schwing, Tamir Hazan, Marc Pollefeys, Raquel Urtasun:
Efficient structured prediction for 3D indoor scene understanding. CVPR 2012: 2815-2822

Efficient, suitable for real time implementation!

Lecture 11

Inferring 3D geometry from images

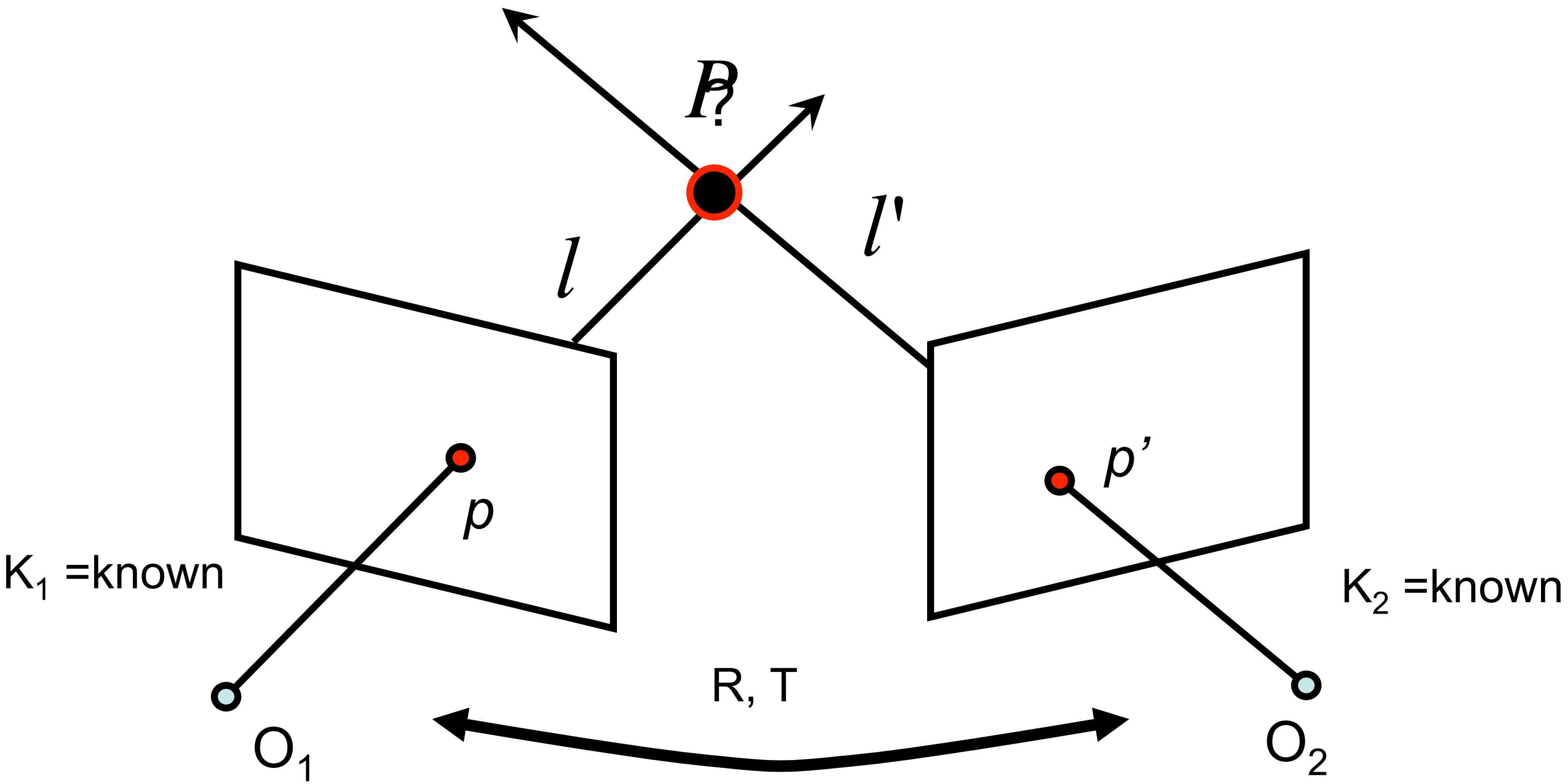
- Cameras
- Single view metrology
- Epipolar geometry
- Structure from motion

Can we recover the structure from a single view?

Intrinsic ambiguity of the mapping from 3D to image (2D)

Courtesy slide S. Lazebnik

Two eyes help!

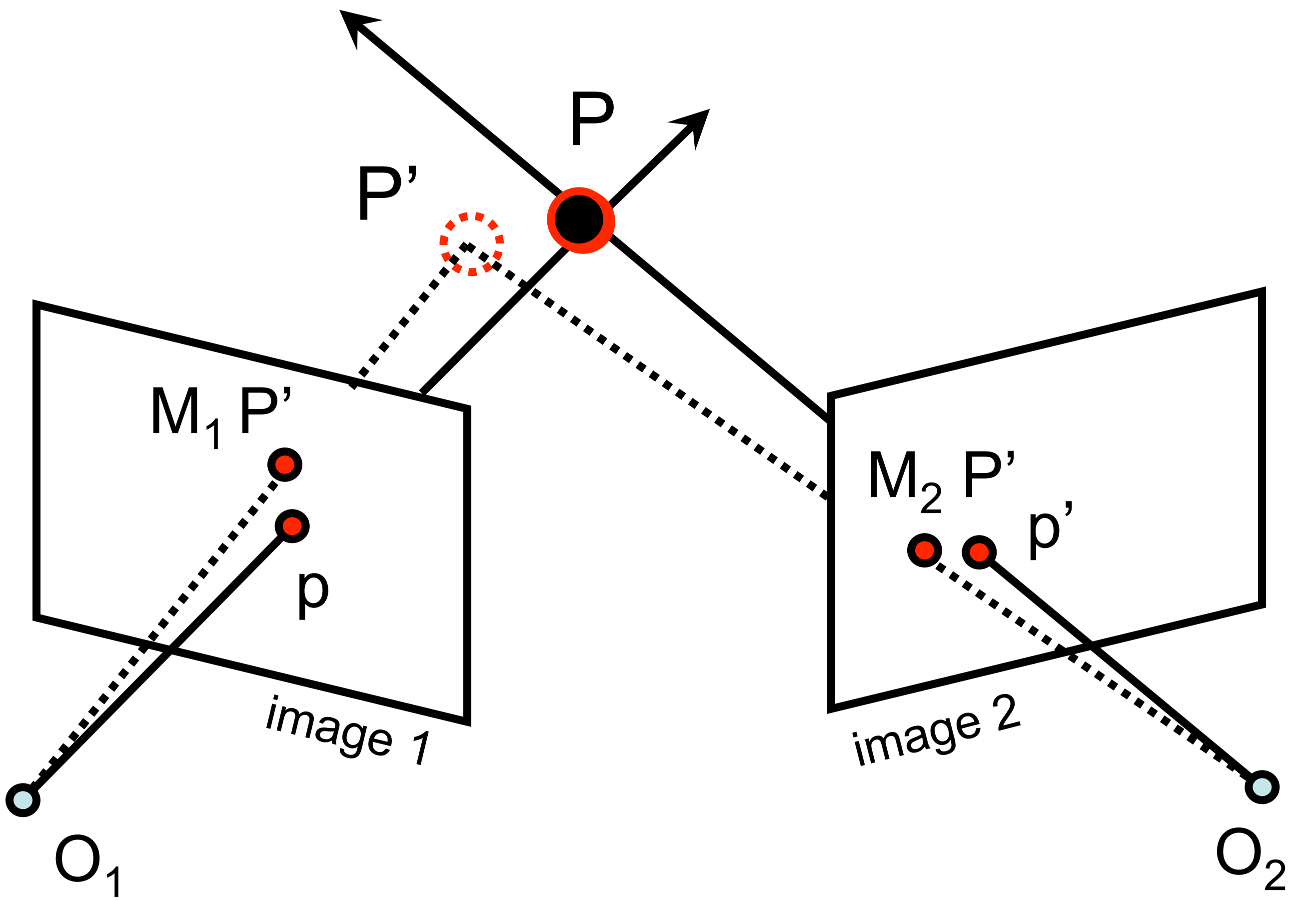


This is called triangulation

Triangulation

- Find P’ that minimizes

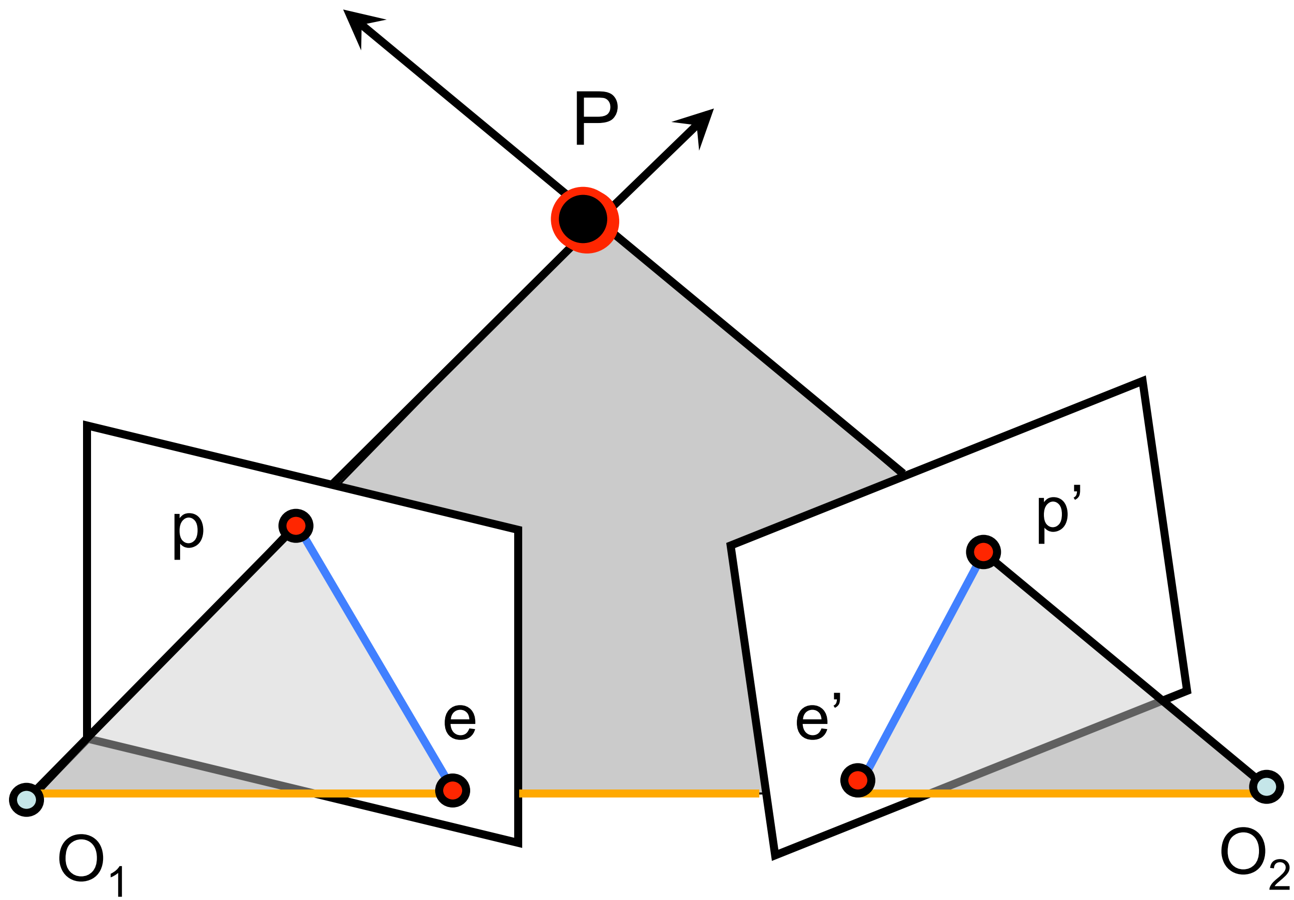
$$d(p, M_1 P') + d(p', M_2 P')$$



Stereo-view geometry

- **Correspondence:** Given a point p in one image, how can I find the corresponding point p' in another one?
- **Camera geometry:** Given corresponding points in two images, find camera matrices, position and pose.
- **Scene geometry:** Find coordinates of 3D point from its projection into 2 or multiple images.

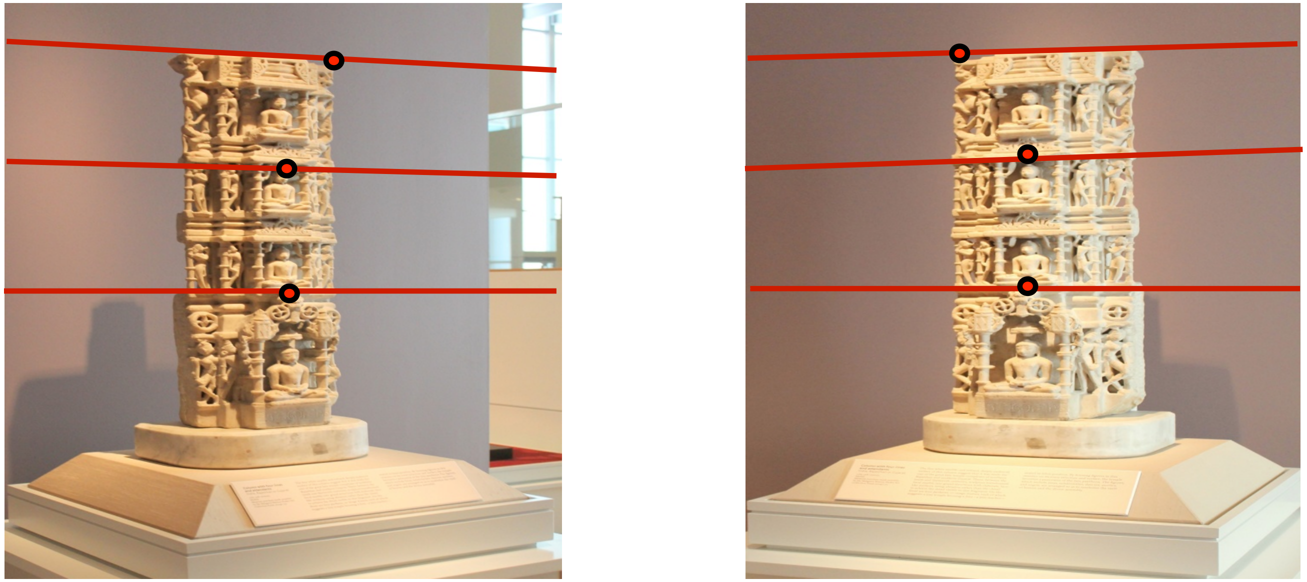
Epipolar geometry



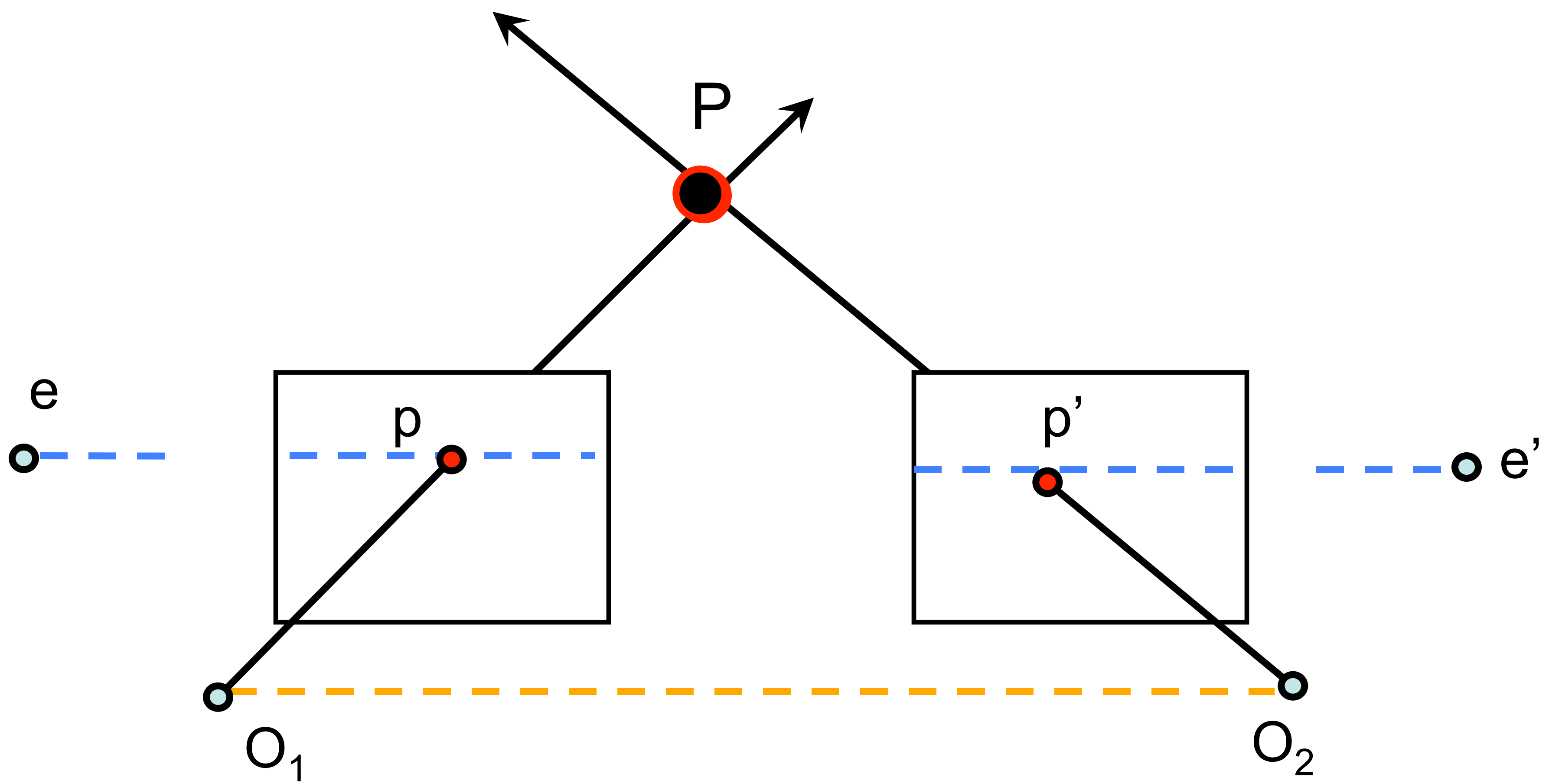
- Epipolar Plane
- Baseline
- Epipolar Lines
- Epipoles e, e’

= intersections of baseline with image planes
= projections of the other camera center

Example of epipolar lines

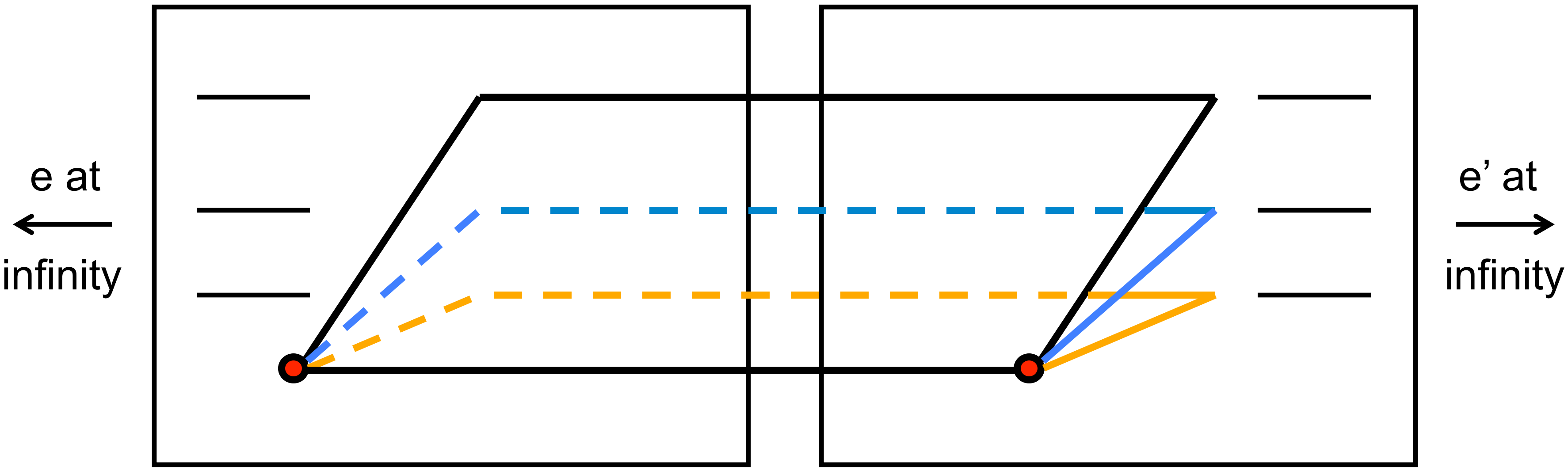


Example: Parallel image planes



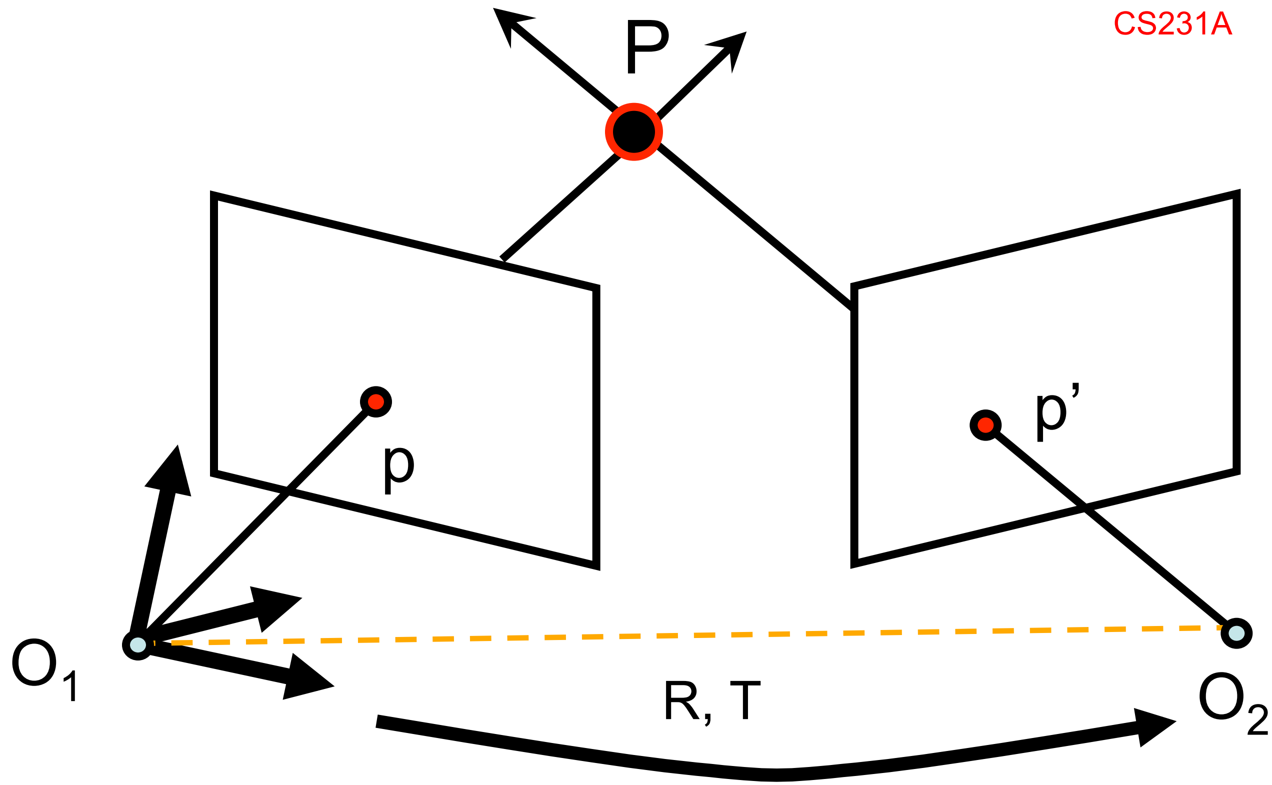
- Baseline intersects the image plane at infinity
- Epipoles are at infinity
- Epipolar lines are parallel to x axis

Example: Parallel Image Planes



Epipolar Constraint

For details see lecture 4
CS231A



$$p^T \cdot \boxed{[T_\times] \cdot R}\, p' = 0$$

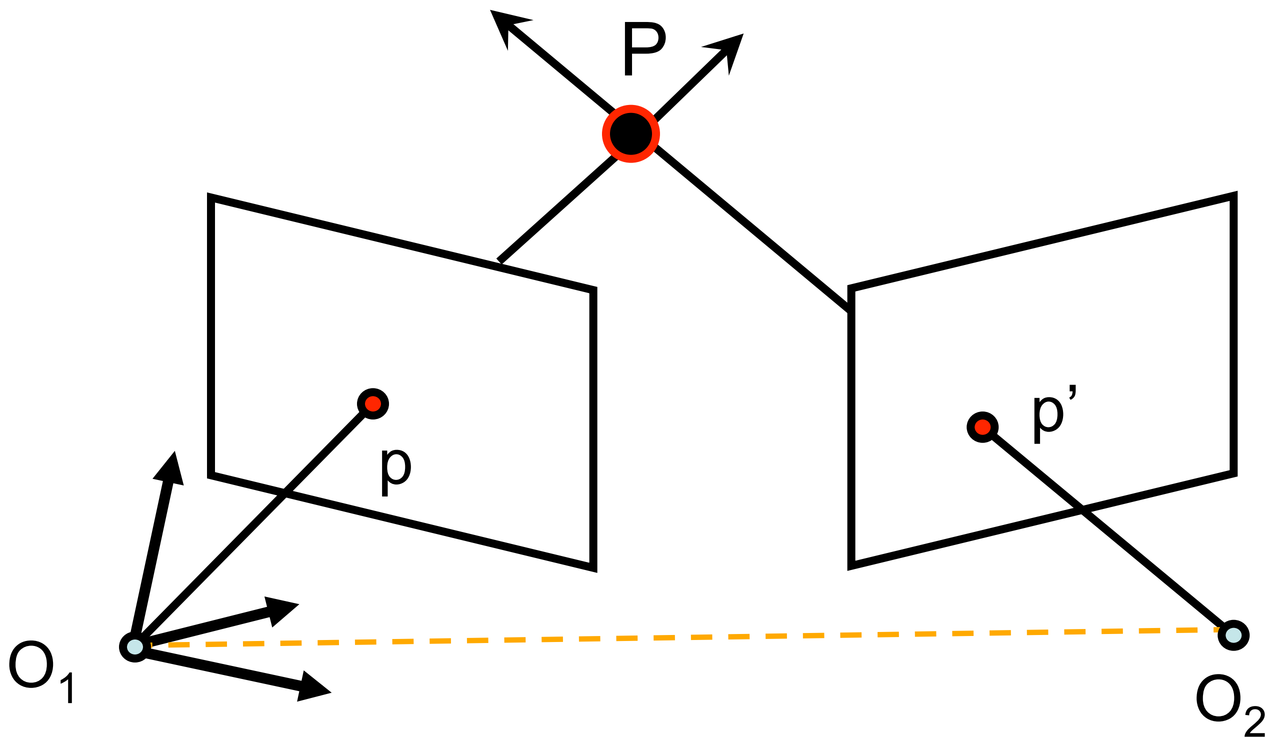
E = Essential matrix

(Longuet-Higgins, 1981)

Cross product as matrix multiplication

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

Epipolar Constraint



$$\boxed{p^T F p' = 0}$$

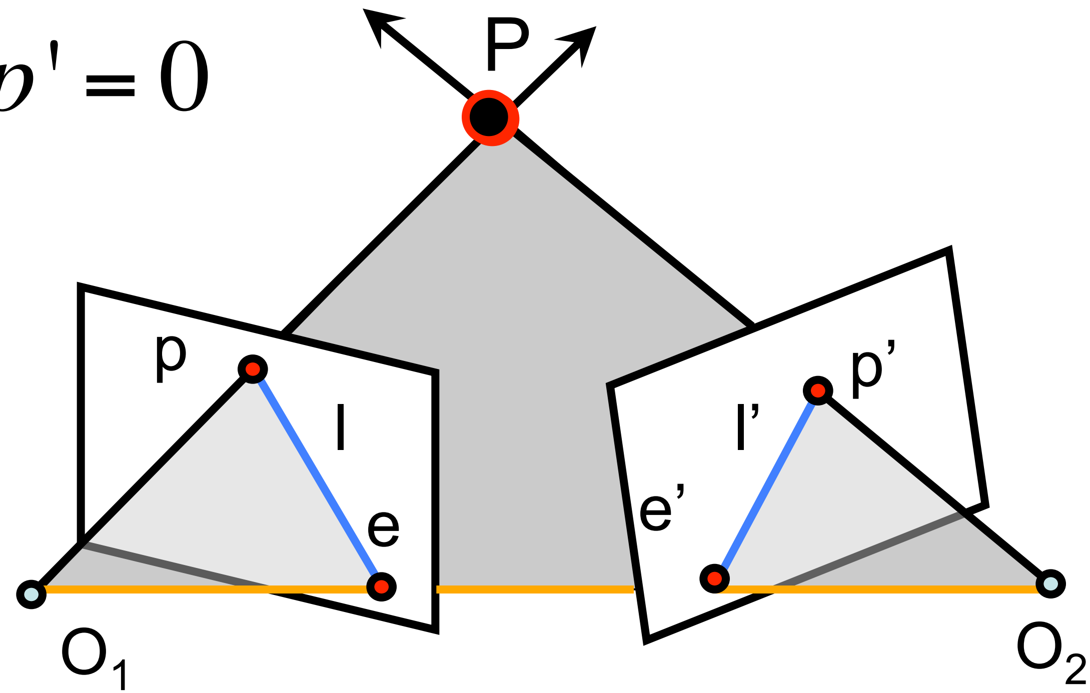
$$F = K^{-T} \cdot [T_\times] \cdot R\, K'^{-1}$$

F = Fundamental Matrix

(Faugeras and Luong, 1992)

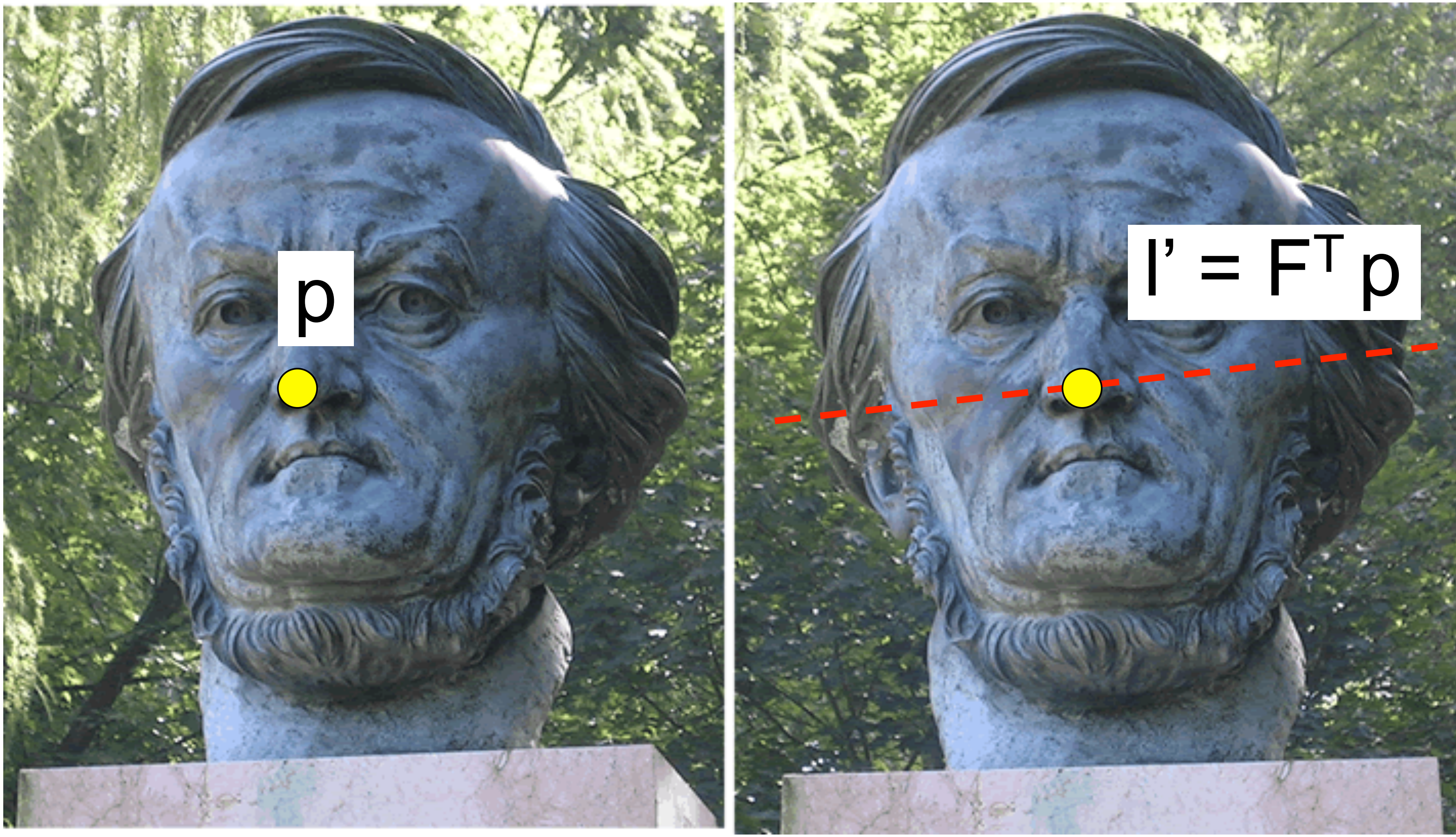
Epipolar Constraint

$$p^T \cdot F\, p' = 0$$



- $l = F\,p'$ is the epipolar line associated with p'
- $l' = F^T p$ is the epipolar line associated with p
- $F\,e' = 0$ and $F^T e = 0$
- F is 3x3 matrix; 7 DOF
- F is singular (rank two)

Why F is useful?



- Suppose F is known
- No additional information about the scene and camera is given
- Given a point on left image, how can I find the corresponding point on right image?

Why F is useful?

- F captures information about the epipolar geometry of 2 views + camera parameters

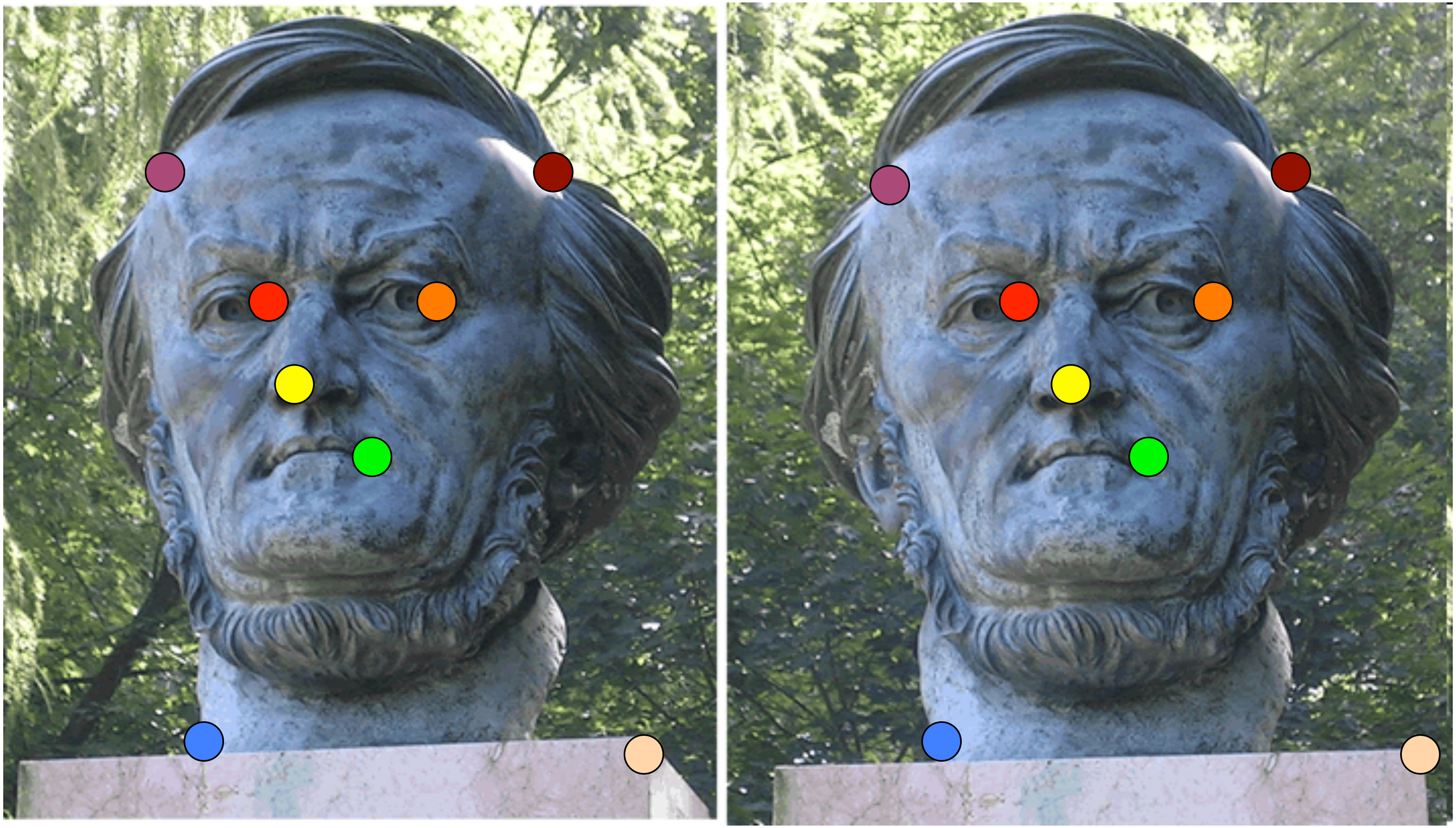
- MORE IMPORTANTLY: F gives constraints on how the scene changes under view point transformation (without reconstructing the scene!)

- Powerful tool in:
    - 3D reconstruction
    - Multi-view object/scene matching

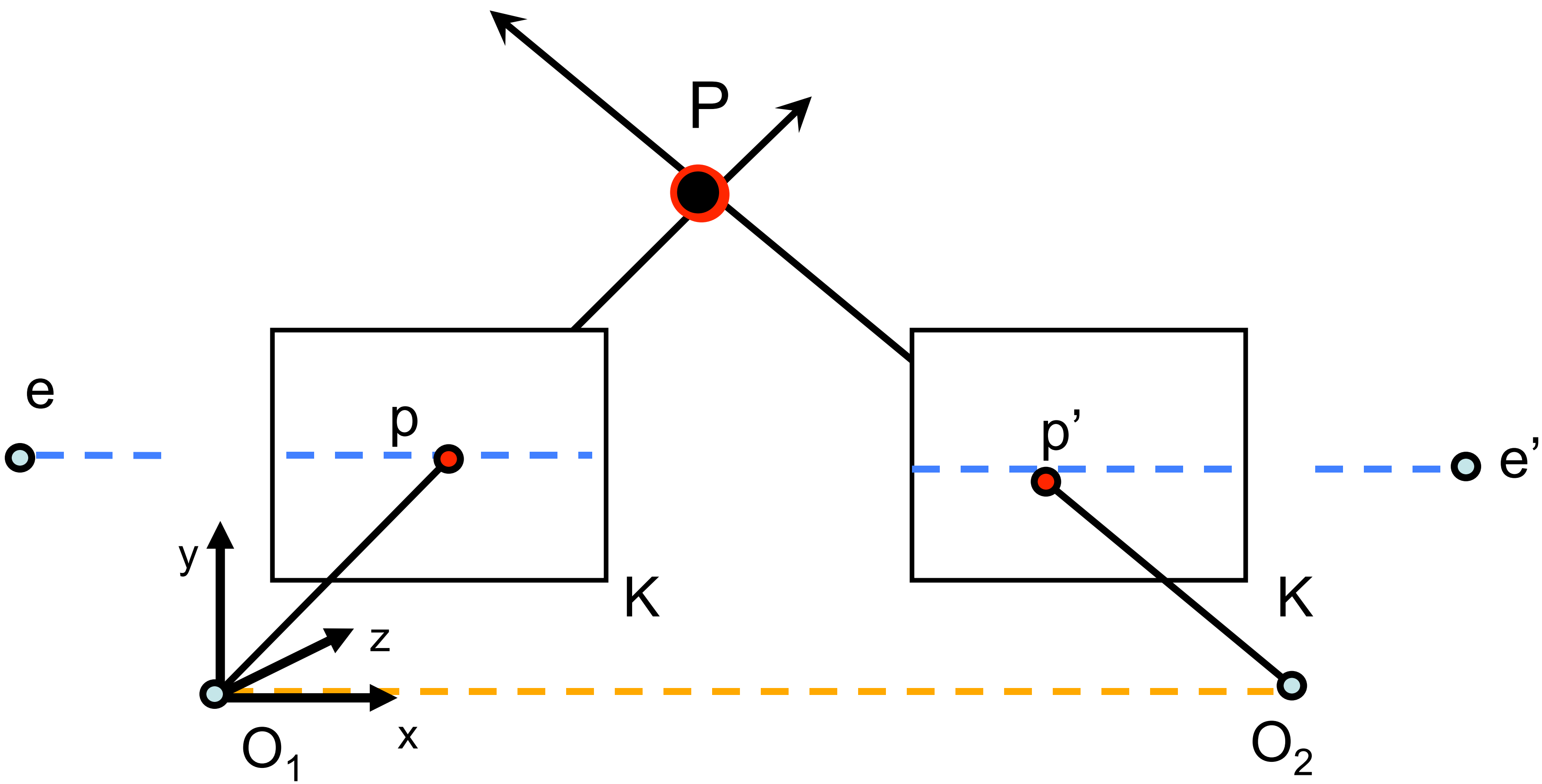
The Eight-Point Algorithm for estimating F

(Longuet-Higgins, 1981)

(Hartley, 1995)



Example: Parallel image planes



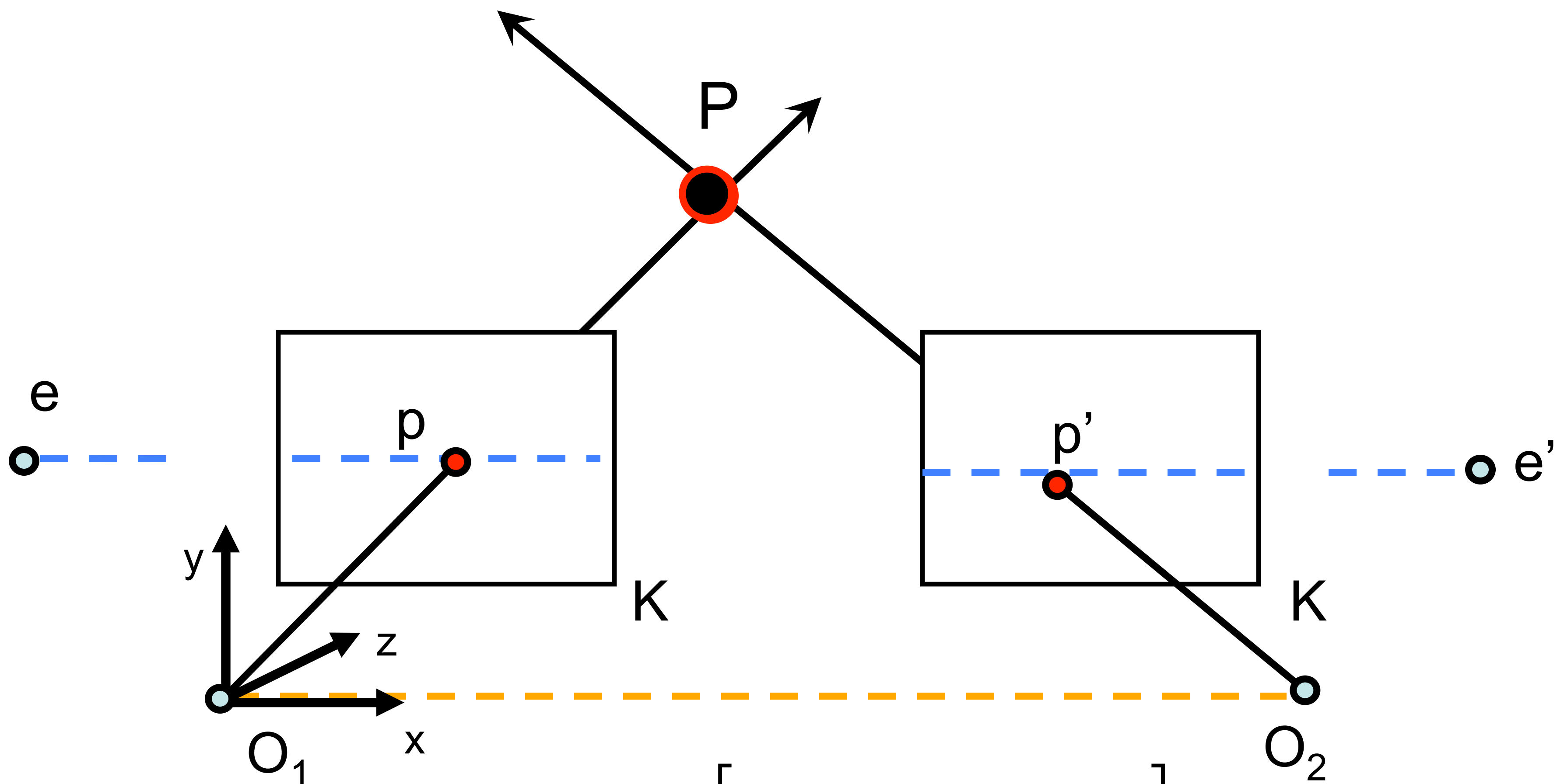
$K_1 = K_2$ = known

x parallel to O_1O_2

$\boxed{\mathbf{E}=?}$

$\boldsymbol{R} = I \qquad \boldsymbol{T} = (T, 0, 0)$

Example: Parallel image planes

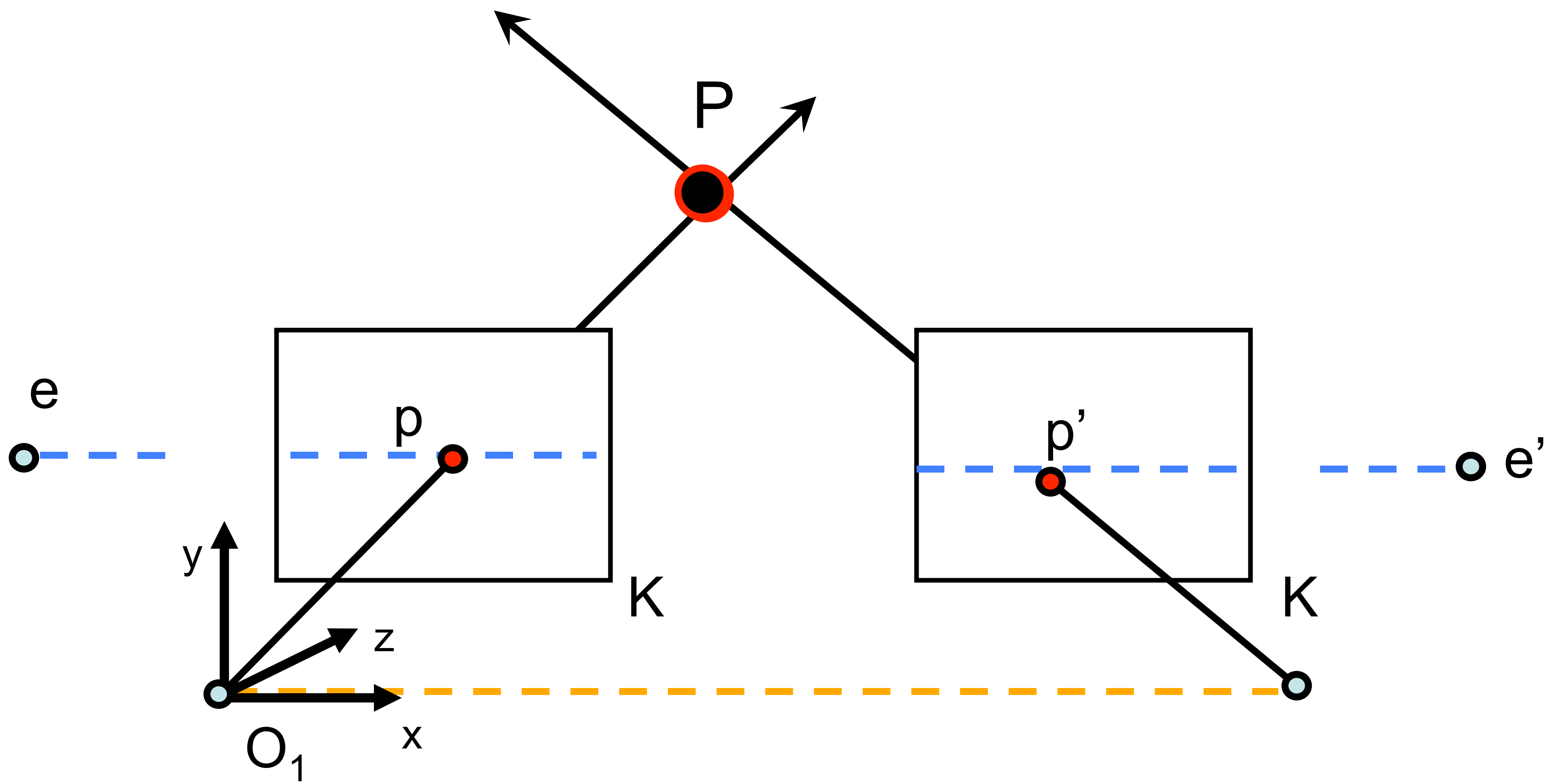


$K_1 = K_2$ = known

x parallel to O_1O_2

$$\mathbf{E} = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix}$$

Example: Parallel image planes

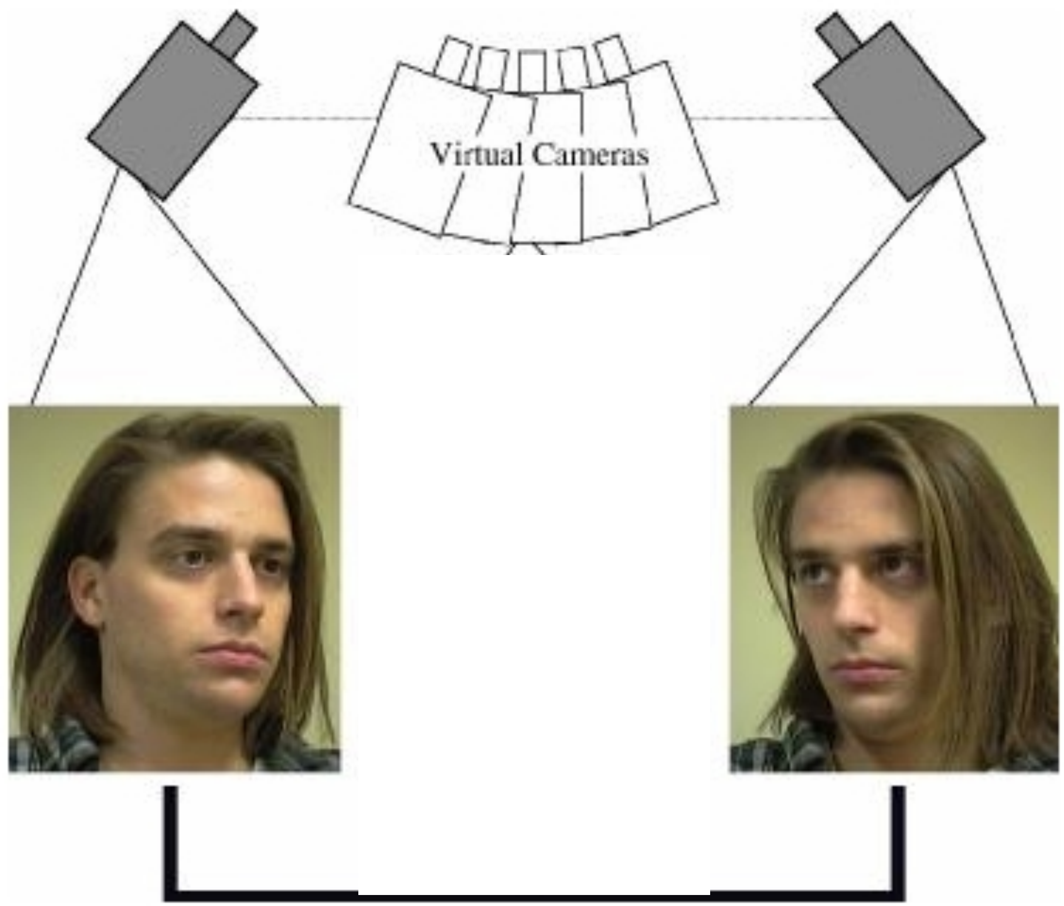


Rectification: making two images “parallel”

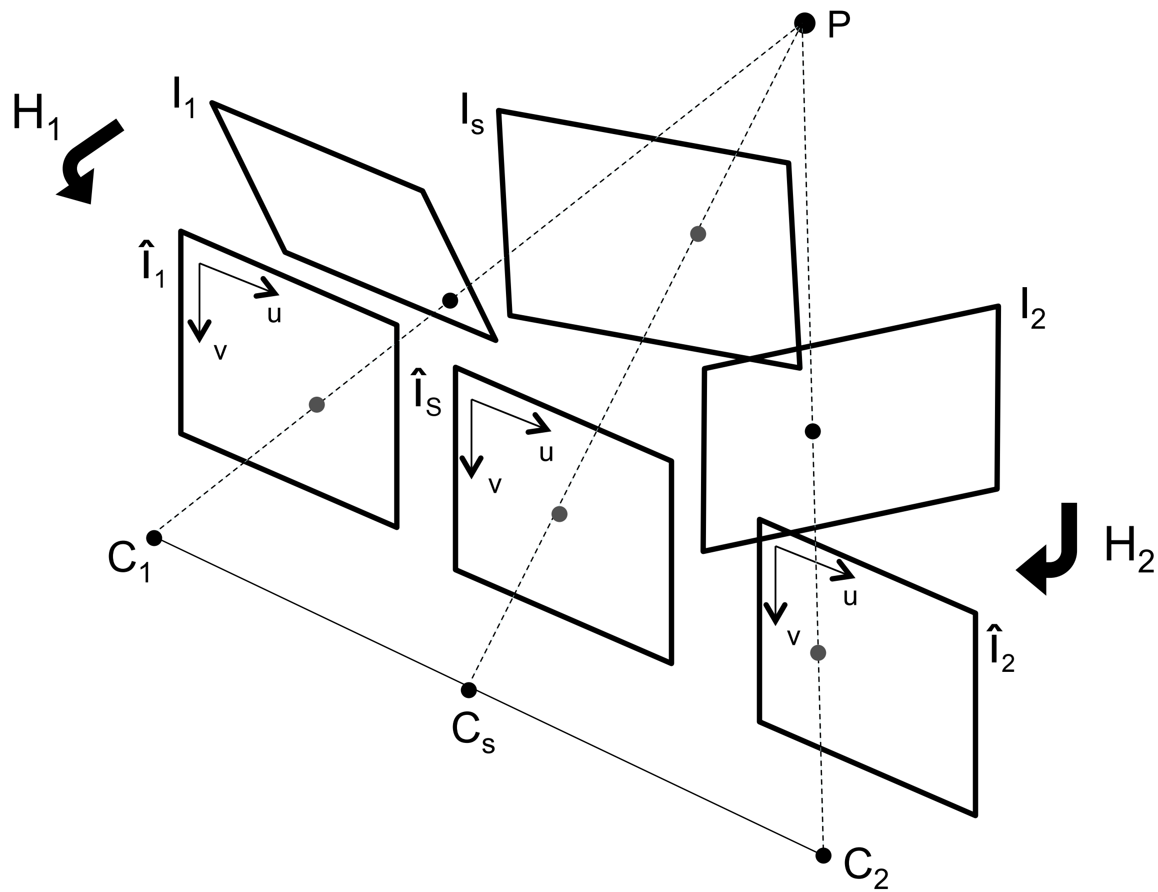
Why it is useful?
- Epipolar constraint → v = v’
- New views can be synthesized by linear interpolation

Application: view morphing

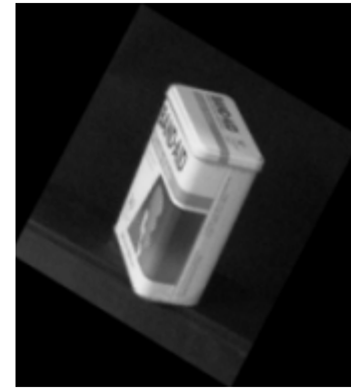
S. M. Seitz and C. R. Dyer, *Proc. SIGGRAPH 96*, 1996, 21-30

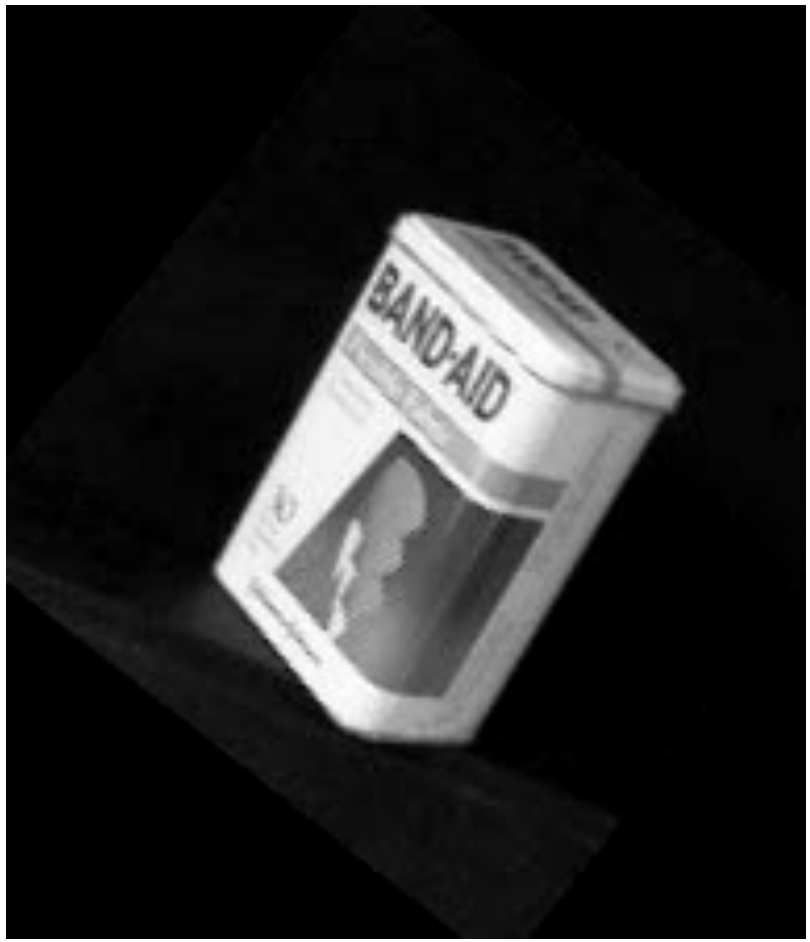


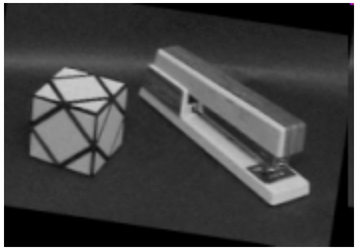
Rectification

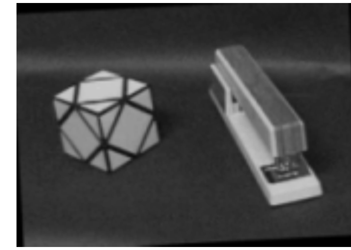


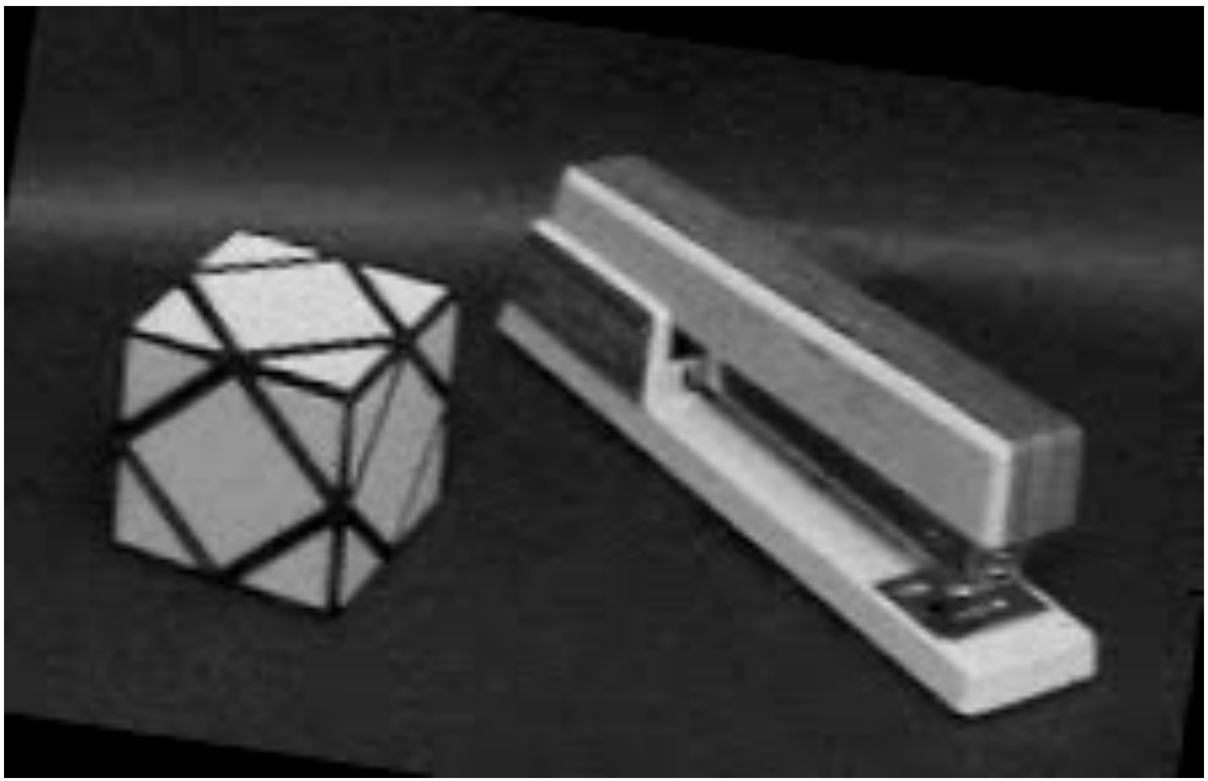

BAND-AID



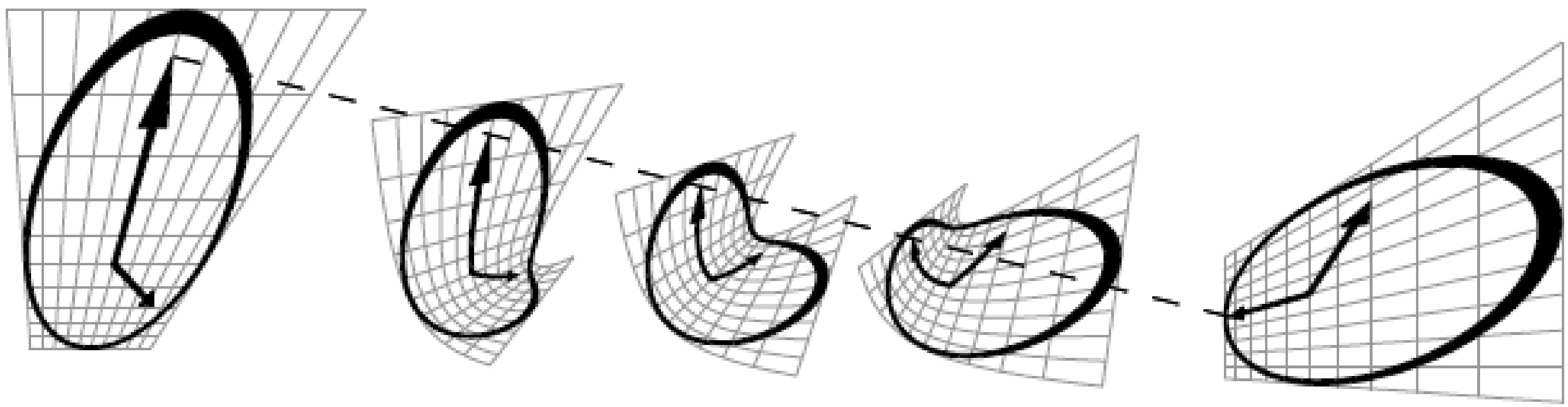

BAND-AID

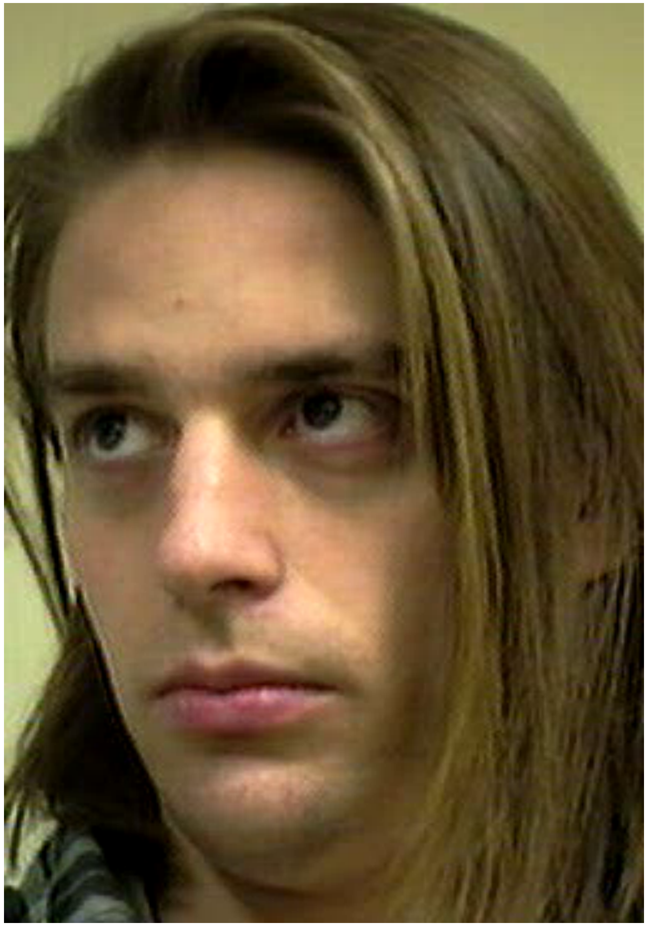


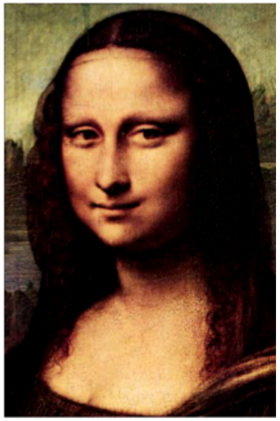


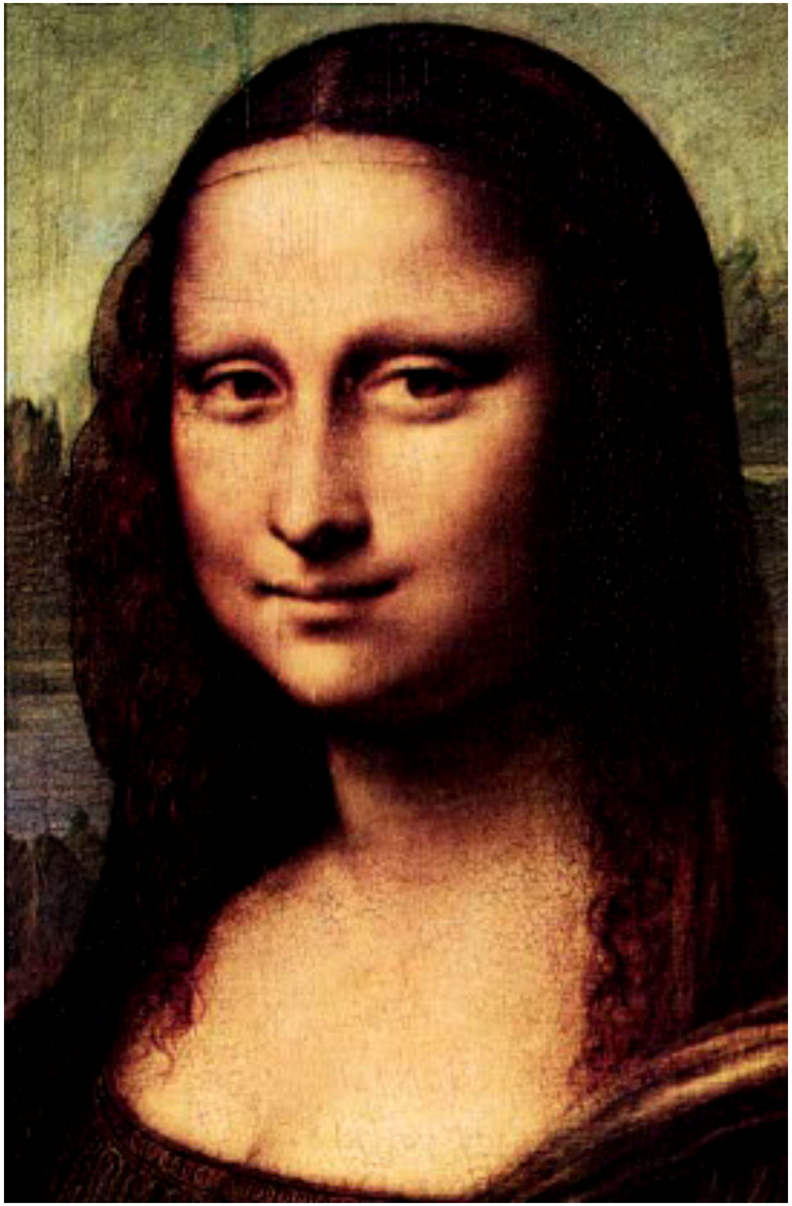


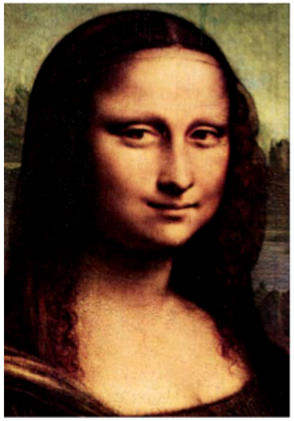
Morphing without rectifying



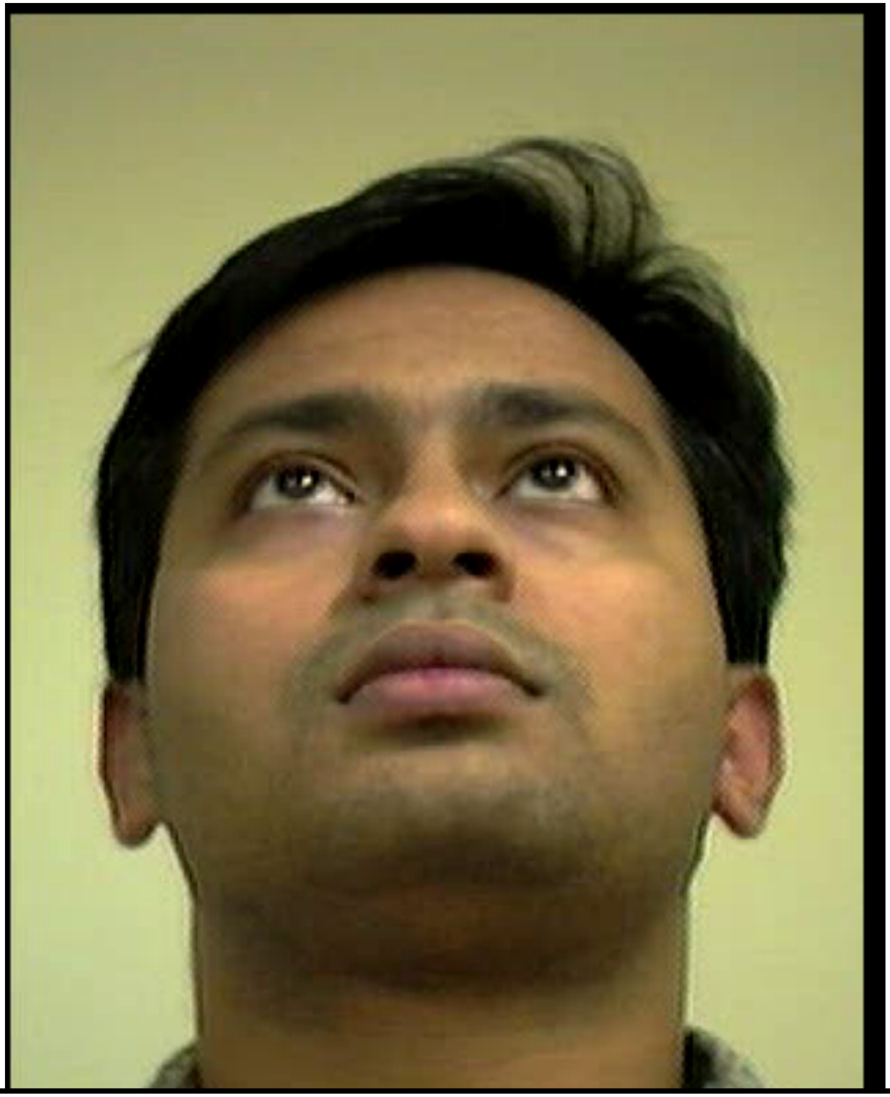








From its reflection!



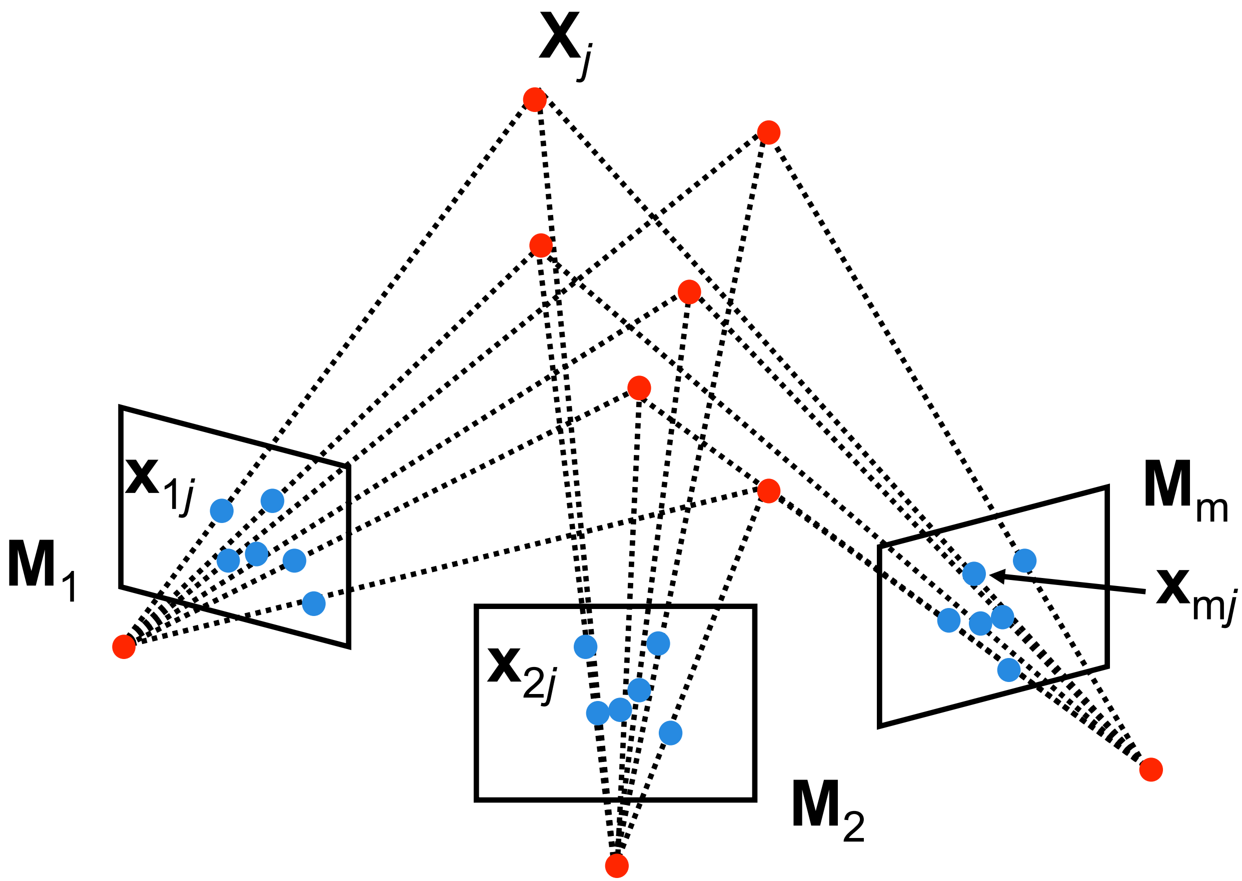
See also: Novel Multi-view Synthesis from a Stereo Image Pair for 3D Display on Mobile Phone, Chen-Hao Wei, Chen-Kuo Chiang, Yu-Wei Sun, Mei-Huei Lin, Shang-Hong Lai, ACCV 2012

Lecture 11

Inferring 3D geometry from images

- Cameras
- Single view metrology
- Epipolar geometry
- Structure from motion

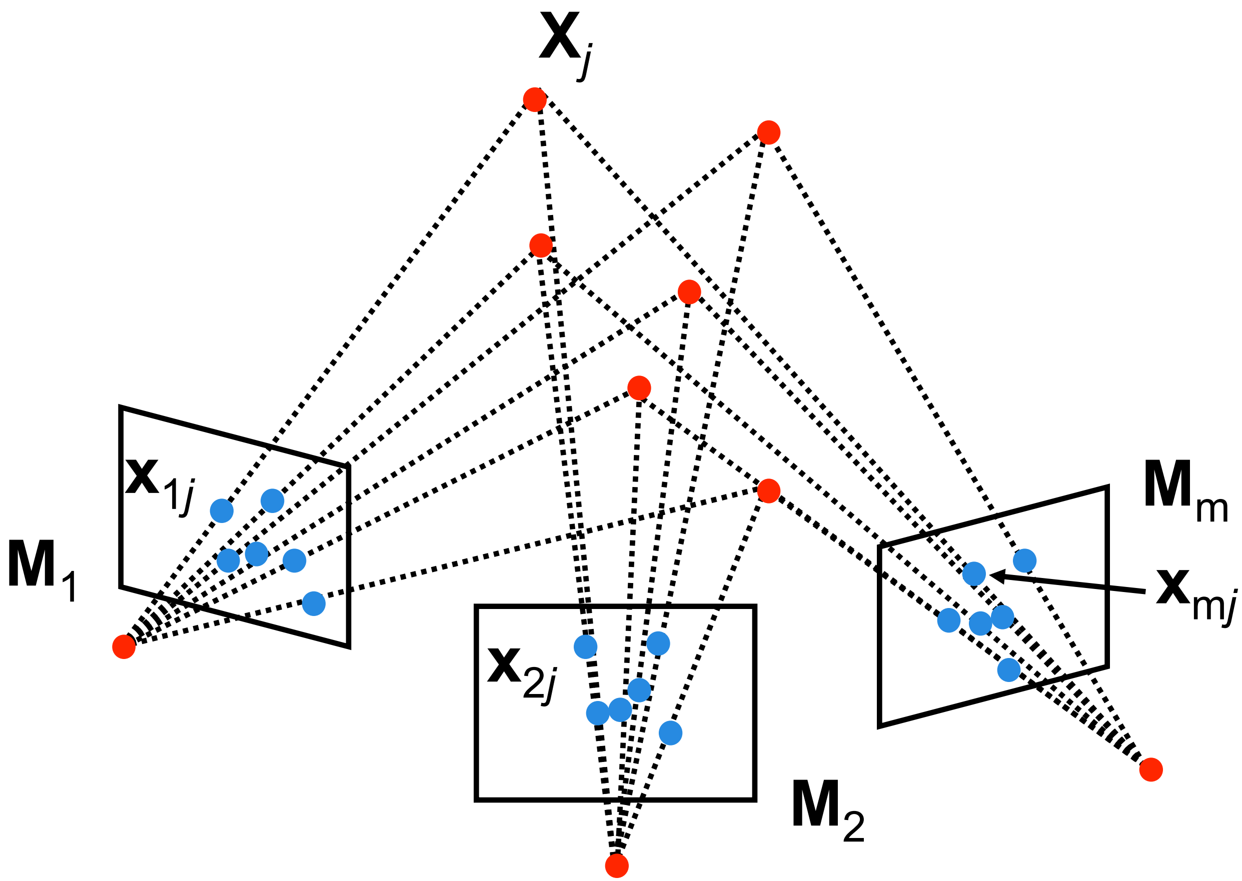
Structure from motion problem



Given *m* images of *n* fixed 3D points

- $\mathbf{x}_{ij} = \mathbf{M}_i \mathbf{X}_j, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n$

Structure from motion problem

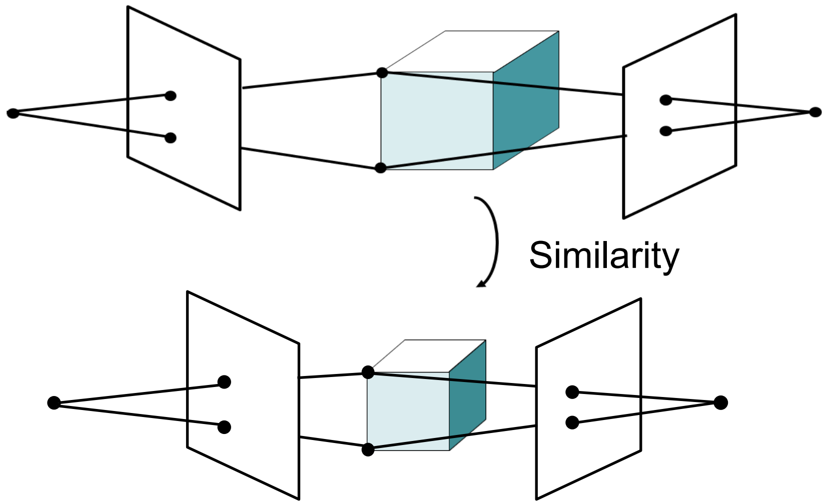


From the mxn correspondences $\mathbf{x}_{ij}$, can we estimate:

- m projection matrices $\mathbf{M}_i$ motion
- n 3D points $\mathbf{X}_j$ structure

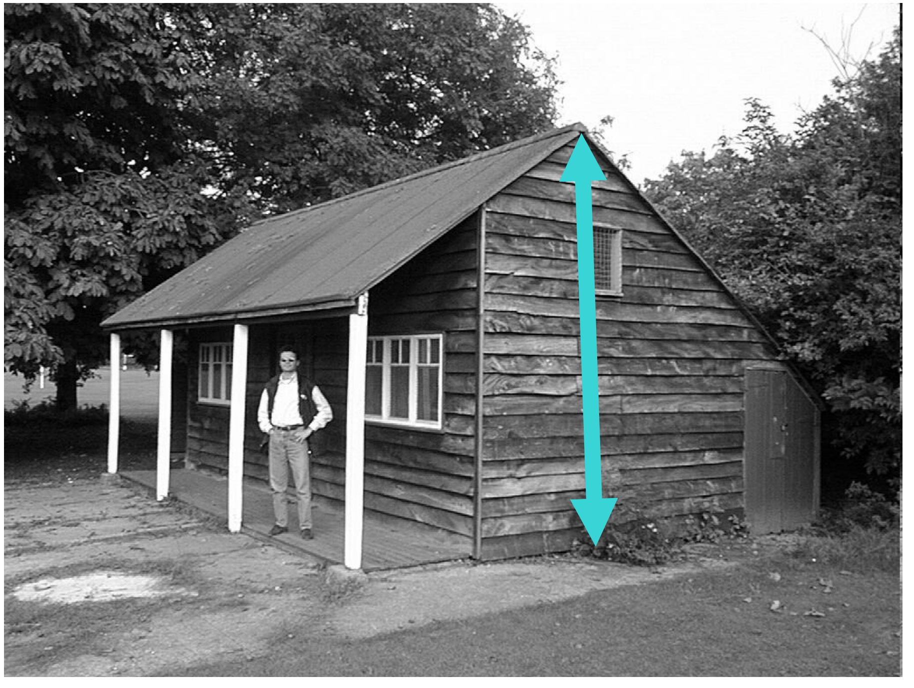
Similarity Ambiguity

- The scene is determined by the images only up a similarity transformation (rotation, translation and scaling)
- This is called **metric reconstruction**



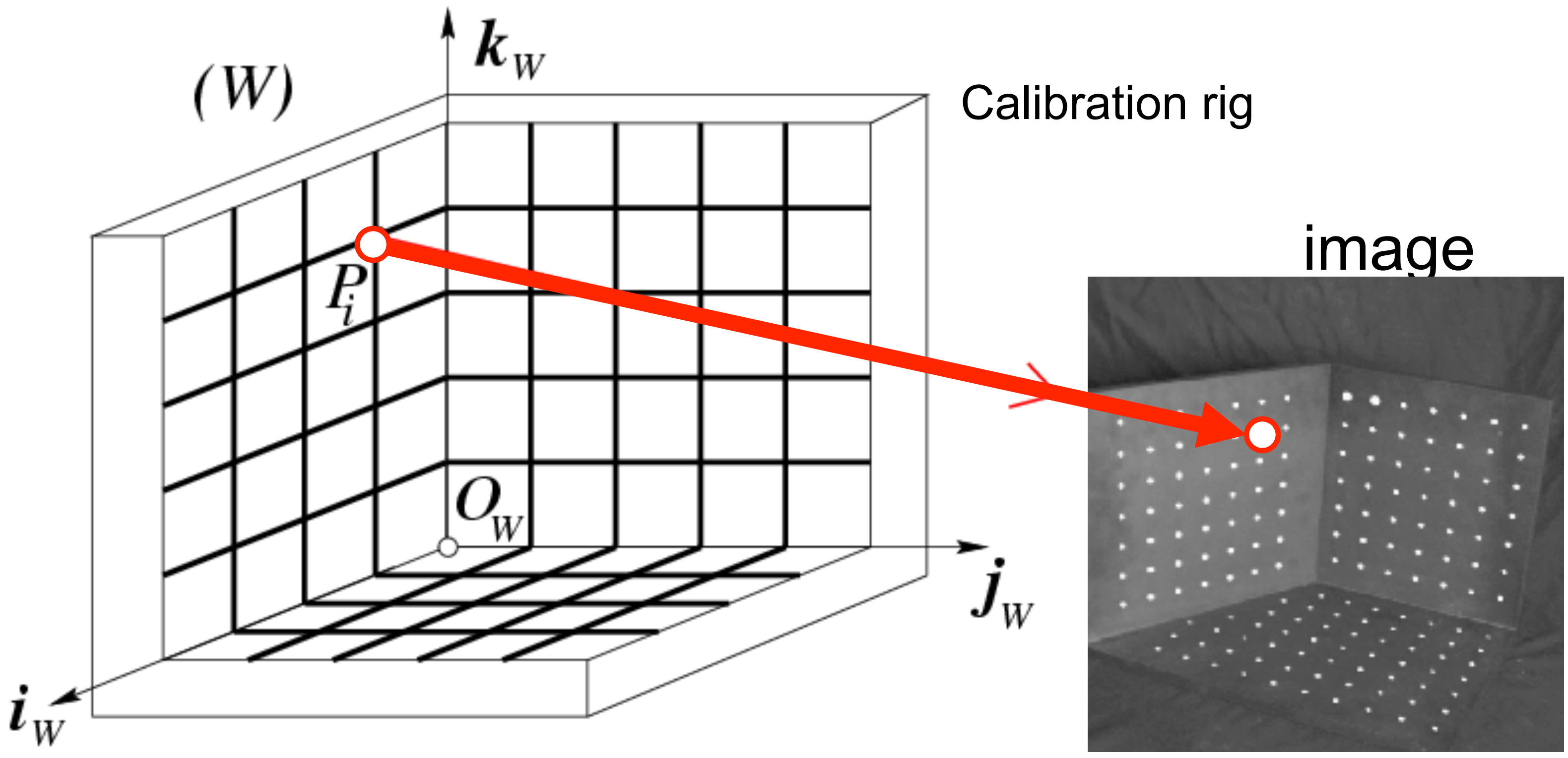
Similarity Ambiguity

- It is impossible based on the images alone to estimate the absolute scale of the scene (i.e. house height)



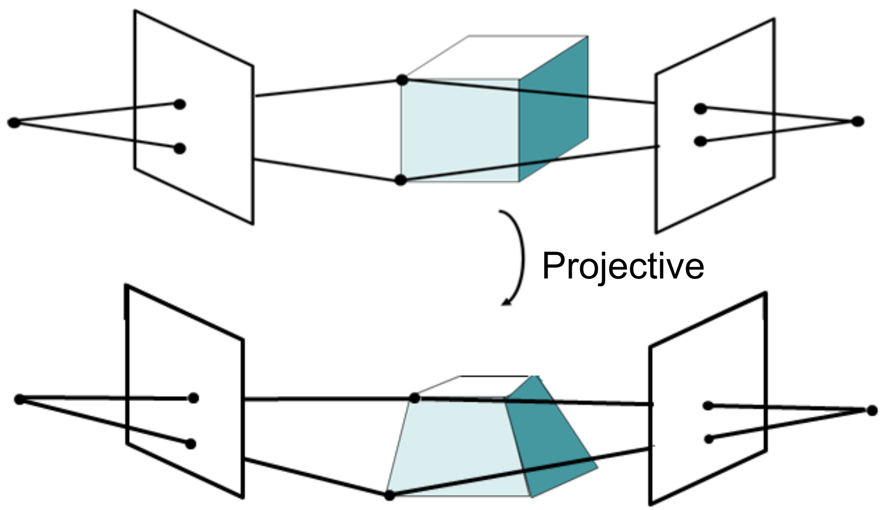
http://www.robots.ox.ac.uk/~vgg/projects/SingleView/models/hut/hutme.wrl

Camera Calibration



This is what we do, when we calibrate the camera

Structure from Motion Ambiguities



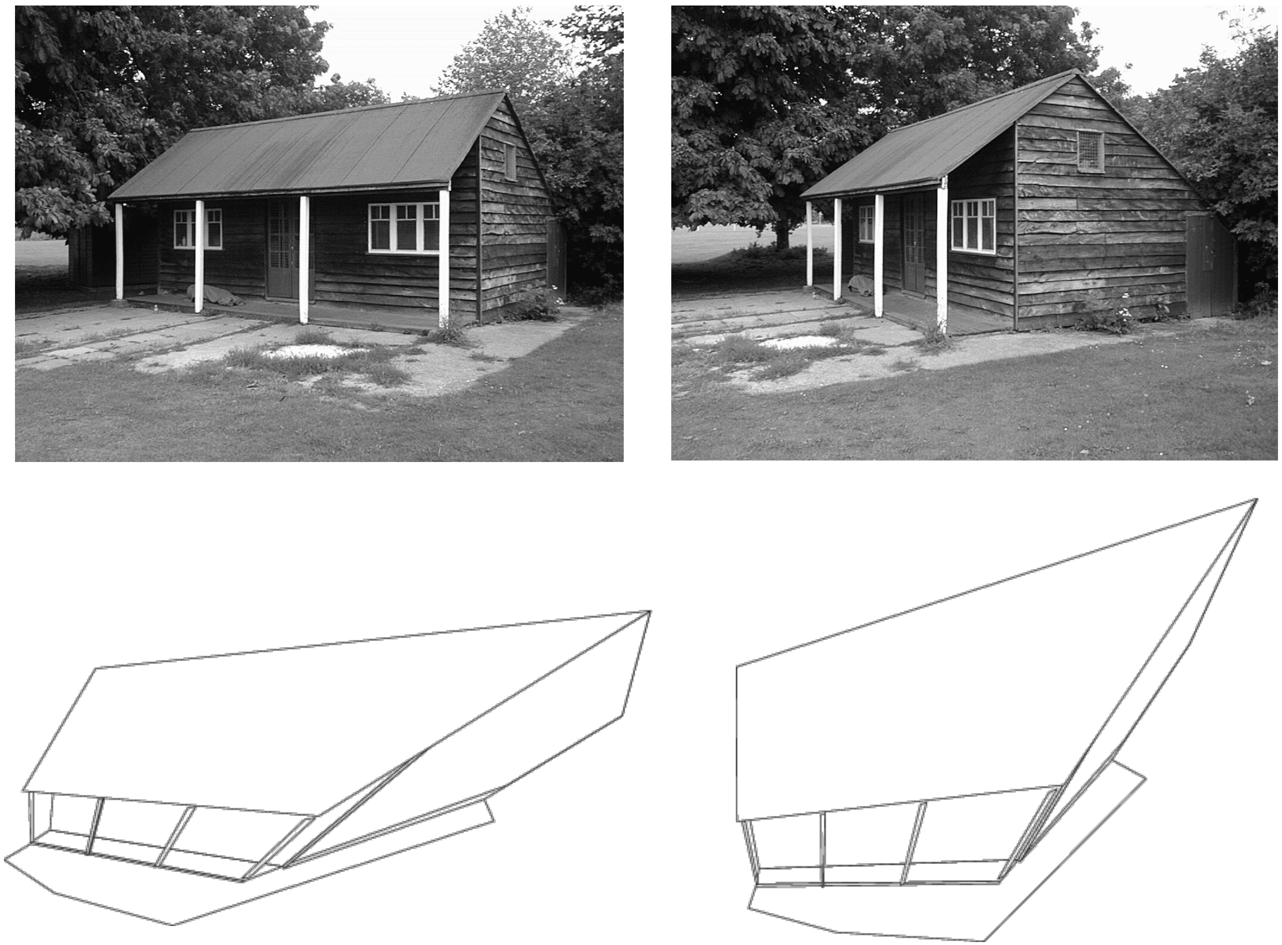
- In the general case (nothing is known) the ambiguity is expressed by an arbitrary affine or projective transformation

$$x_j = M_i \boxed{X_j} \quad \boxed{M_i} = K_i \begin{bmatrix} R_i & T_i \end{bmatrix}$$

$$\boxed{H X_j} \quad \boxed{M_i H^{-1}}$$

$$x_j = M_i X_j = \left(M_i H^{-1}\right)\left(H X_j\right)$$

Projective Ambiguity

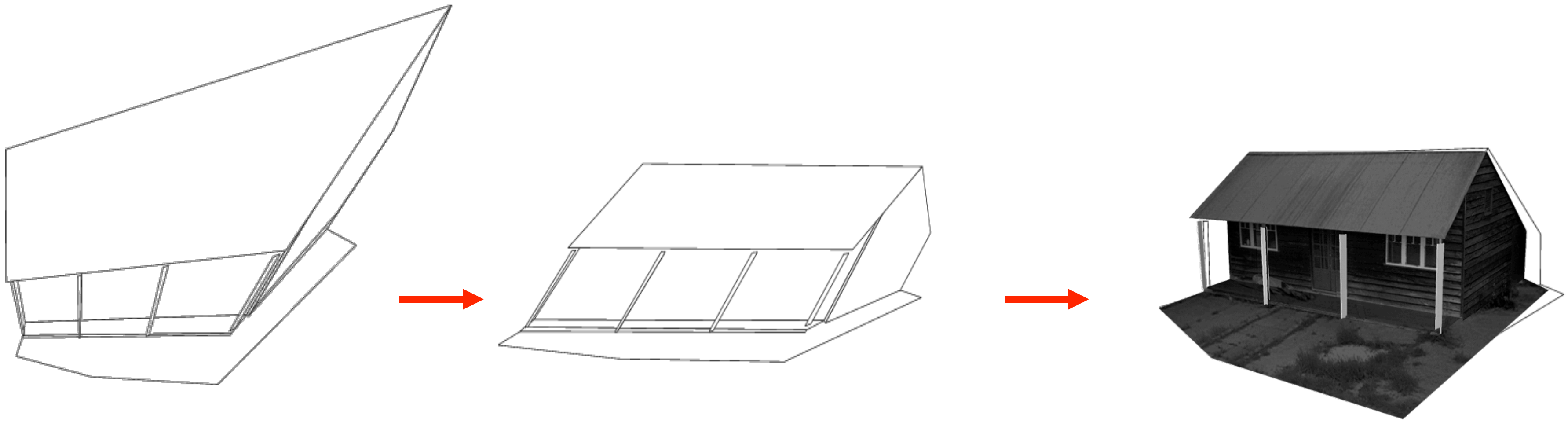


R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, 2nd edition, 2003

Metric reconstruction (upgrade)

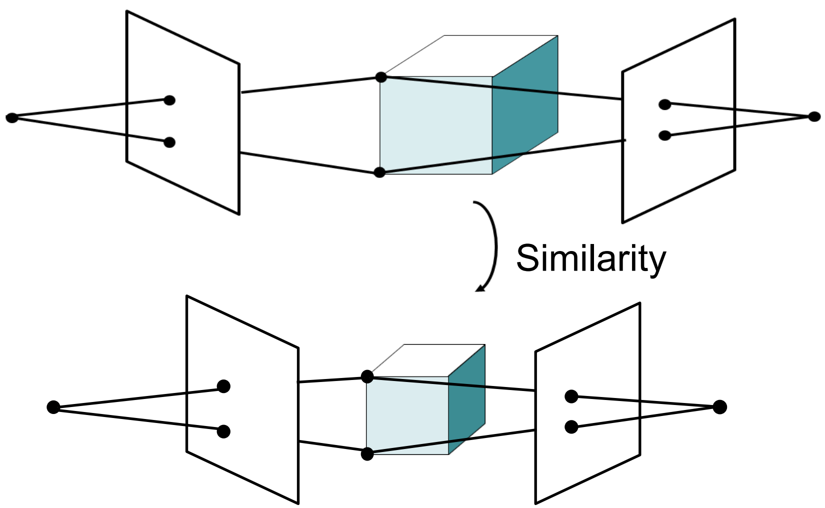
- The problem of recovering the metric reconstruction from the perspective one is called self-calibration

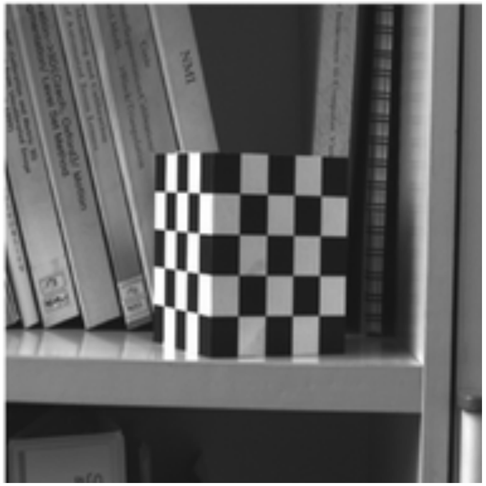
- Stratified reconstruction:
    - from perspective to affine
    - from affine to metric



Mobile SFM

- Intrinsic camera parameters are known or can be calibrated.
- For calibrated cameras, the similarity ambiguity is the only ambiguity [Longuet-Higgins ’81]
- No need for stratified solution or auto-calibration



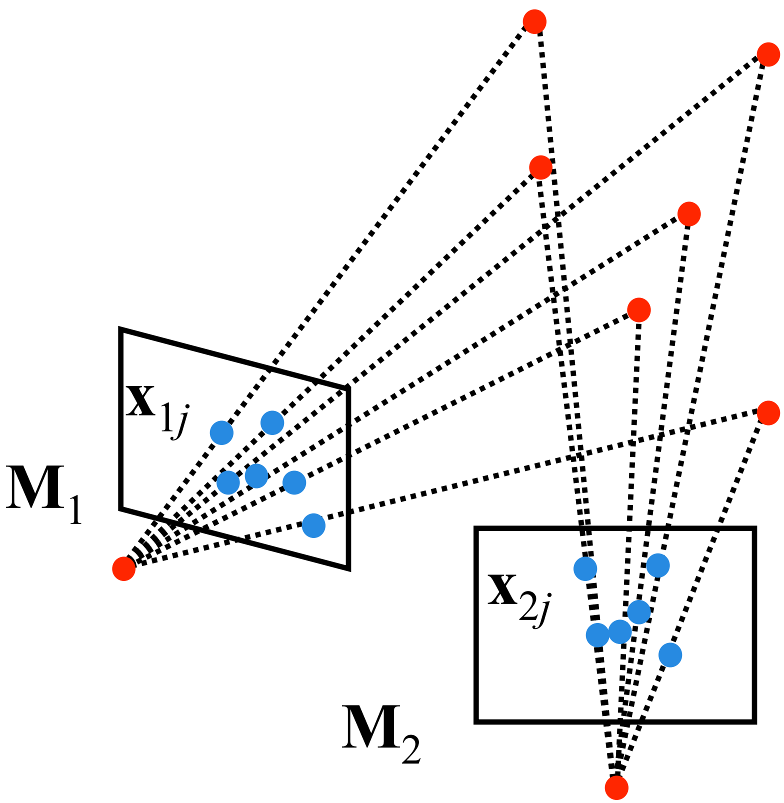


- Metric reconstruction can be determined if a calibration pattern is used or the absolute size of an known object is given.

Structure-from-Motion Algorithms

- Algebraic approach (by fundamental matrix)
- Factorization method (by SVD)
- Bundle adjustment

Algebraic approach (2-view case)



$$x_{ij} = M_i \, X_j$$

Apply a projective transformation H such that:

$$M_1 H^{-1} = \begin{bmatrix} I & 0 \end{bmatrix} \qquad M_2 H^{-1} = \begin{bmatrix} A & b \end{bmatrix}$$

Canonical perspective cameras

Algebraic approach (2-view case)

1. Compute the fundamental matrix **F** from two views (eg. 8 point algorithm)
2. Compute **b** and **A** from **F**

Compute **b** as least sq. solution of **F b** = 0, with |**b**|=1 using SVD; **b** is an epipole

$$\mathbf{A} = -[\mathbf{b}_{\times}]\ \mathbf{F}$$

3. Use **b** and **A** to estimate projective cameras

$$M_1 = \begin{bmatrix} I & 0 \end{bmatrix} \qquad M_2 = \begin{bmatrix} -[\mathbf{b}_x]\mathbf{F} & \mathbf{b} \end{bmatrix}$$

4. Use these cameras to triangulate and estimate points in 3D

For details, see CS231A, lecture 7

Structure-from-Motion Algorithms

- Algebraic approach (by fundamental matrix)
- z • Factorization method (by SVD)
- Bundle adjustment

C. Tomasi and T. Kanade
Shape and motion from image streams under orthography: A factorization method. *IJCV*, 9(2): 137-154, November 1992.

For details, see CS231A, lecture 6

Structure-from-Motion Algorithms

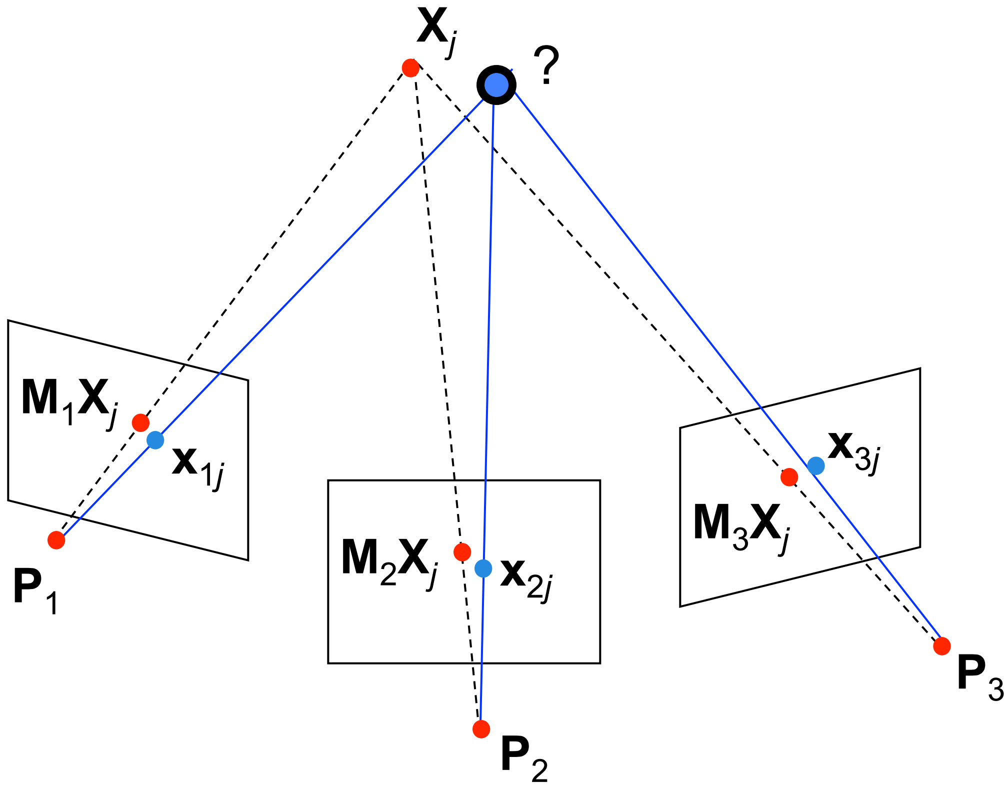
- Algebraic approach (by fundamental matrix)
- Factorization method (by SVD)
- Bundle adjustment

Bundle adjustment

Non-linear method for refining structure and motion

Minimizing re-projection error

$$E(\mathrm{M}, \mathbf{X}) = \sum_{i=1}^{m} \sum_{j=1}^{n} D\left(\mathbf{x}_{ij}, \mathrm{M}_i \mathbf{X}_j\right)^2$$



Bundle adjustment

Non-linear method for refining structure and motion

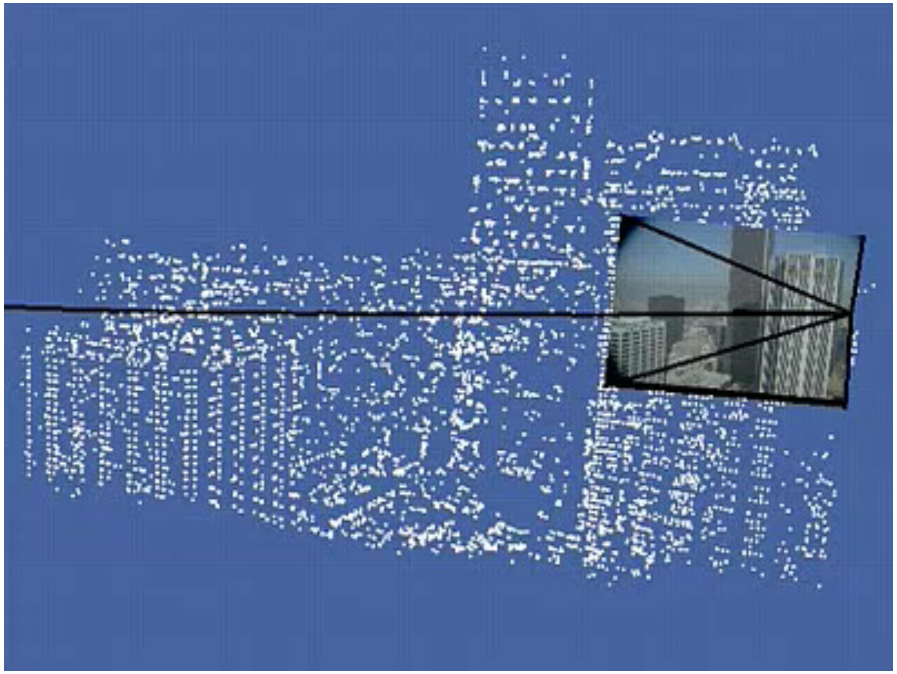
Minimizing re-projection error

$$E(M, \mathbf{X}) = \sum_{i=1}^{m} \sum_{j=1}^{n} D\left(\mathbf{x}_{ij}, M_i \mathbf{X}_j\right)^2$$

- Advantages
  - Handle large number of views
  - Handle missing data
  - Can leverage standard optimization packaged such as Levenberg-Marquardt
- Limitations
  - Large minimization problem (parameters grow with number of views)
  - Requires good initial condition

Used as the final step of SFM; key ingredient for VLSAM

Structure from motion problem



Courtesy of Oxford **Visual Geometry Group**

Lucas & Kanade, 81
Chen & Medioni, 92
Debevec et al., 96
Levoy & Hanrahan, 96
Fitzgibbon & Zisserman, 98
Triggs et al., 99
Pollefeys et al., 99
Kutulakos & Seitz, 99

Levoy et al., 00
Hartley & Zisserman, 00
Dellaert et al., 00
Rusinkiewic et al., 02
Nistér, 04
Brown & Lowe, 04
Schindler et al, 04
Lourakis & Argyros, 04
Colombo et al. 05

Golparvar-Fard, et al. JAEI 10
Pandey et al. IFAC , 2010
Pandey et al. ICRA 2011
Microsoft's PhotoSynth
Snavely et al., 06-08
Schindler et al., 08
Agarwal et al., 09
Frahm et al., 10

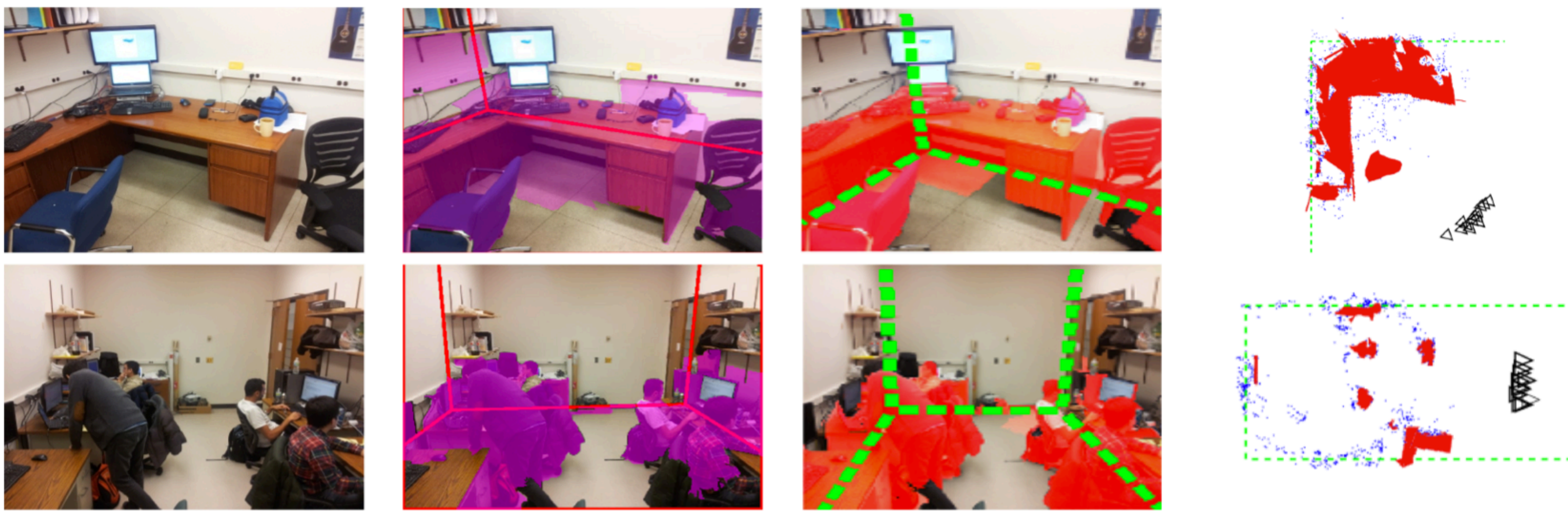
SFM and Photosynth

Noah Snavely, Steven M. Seitz, Richard Szeliski, "[Photo tourism: Exploring photo collections in 3D](#)," ACM Transactions on Graphics (SIGGRAPH Proceedings),2006,

https://photosynth.net/preview

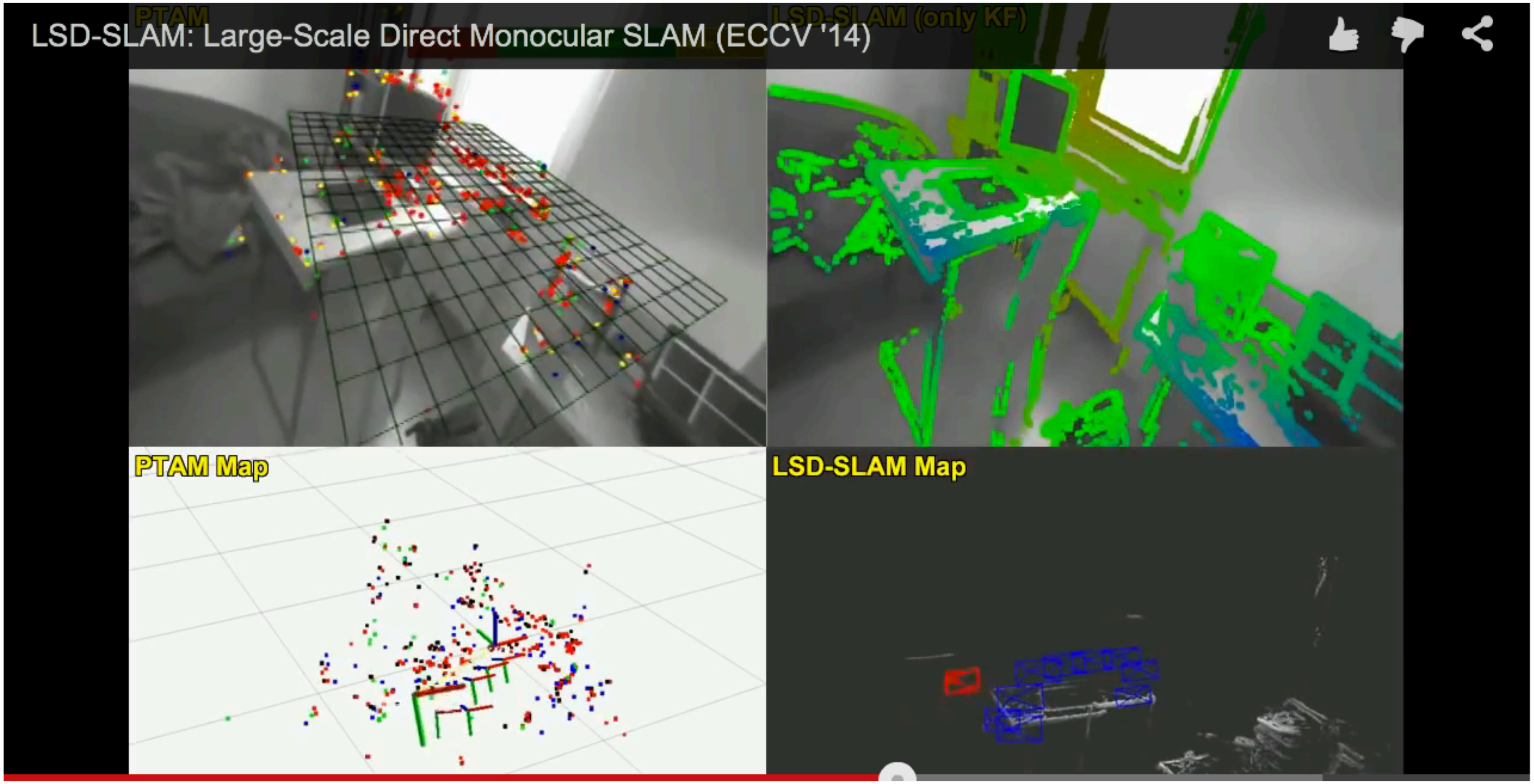
SFM and room layout estimation

Y. Bao, A. Furlan, L. Fei-Fei, S. Savarese, Understanding the 3D Layout of a Cluttered Room From Multiple Images, in IEEE Winter Conference on Applications of Computer Vision (WACV), 2014.



LSD-SLAM: Large-Scale Direct Monocular SLAM

Jakob Engel, Thomas Schöps, Prof. Dr. Daniel Cremers



http://vision.in.tum.de/research/lsdslam

Recent papers for single or multi-view reconstruction on mobiles

Engel, Jakob, Thomas Schöps, and Daniel Cremers. "LSD-SLAM: Large-scale direct monocular SLAM." Computer Vision–ECCV 2014. Springer International Publishing, 2014. 834-849.
http://link.springer.com/chapter/10.1007/978-3-319-10605-2_54#page-1
Includes an optimized mobile implementation

Forster, Christian, Matia Pizzoli, and Davide Scaramuzza. "SVO: Fast semi-direct monocular visual odometry." Robotics and Automation (ICRA), 2014 IEEE International Conference on. IEEE, 2014.
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6906584&tag=1
Includes an optimized mobile implementation

Kolev, Kalin, et al. "Turning mobile phones into 3D scanners." Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on. IEEE, 2014.
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6909899

Yu, Fisher, and David Gallup. "3D Reconstruction from Accidental Motion." Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on. IEEE, 2014.
http://yf.io/p/tiny/
A similar algorithm is implemented for lens blur in Google's Android Camera App.

Gasparini, Simone, and Pascal Bertolino. "Stereo camera tracking for mobile devices." Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on. IEEE, 2013.
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6595845

Hedborg, Johan, Andreas Robinson, and Michael Felsberg. "Robust Three-View Triangulation Done Fast." Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on. IEEE, 2014.
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6909973

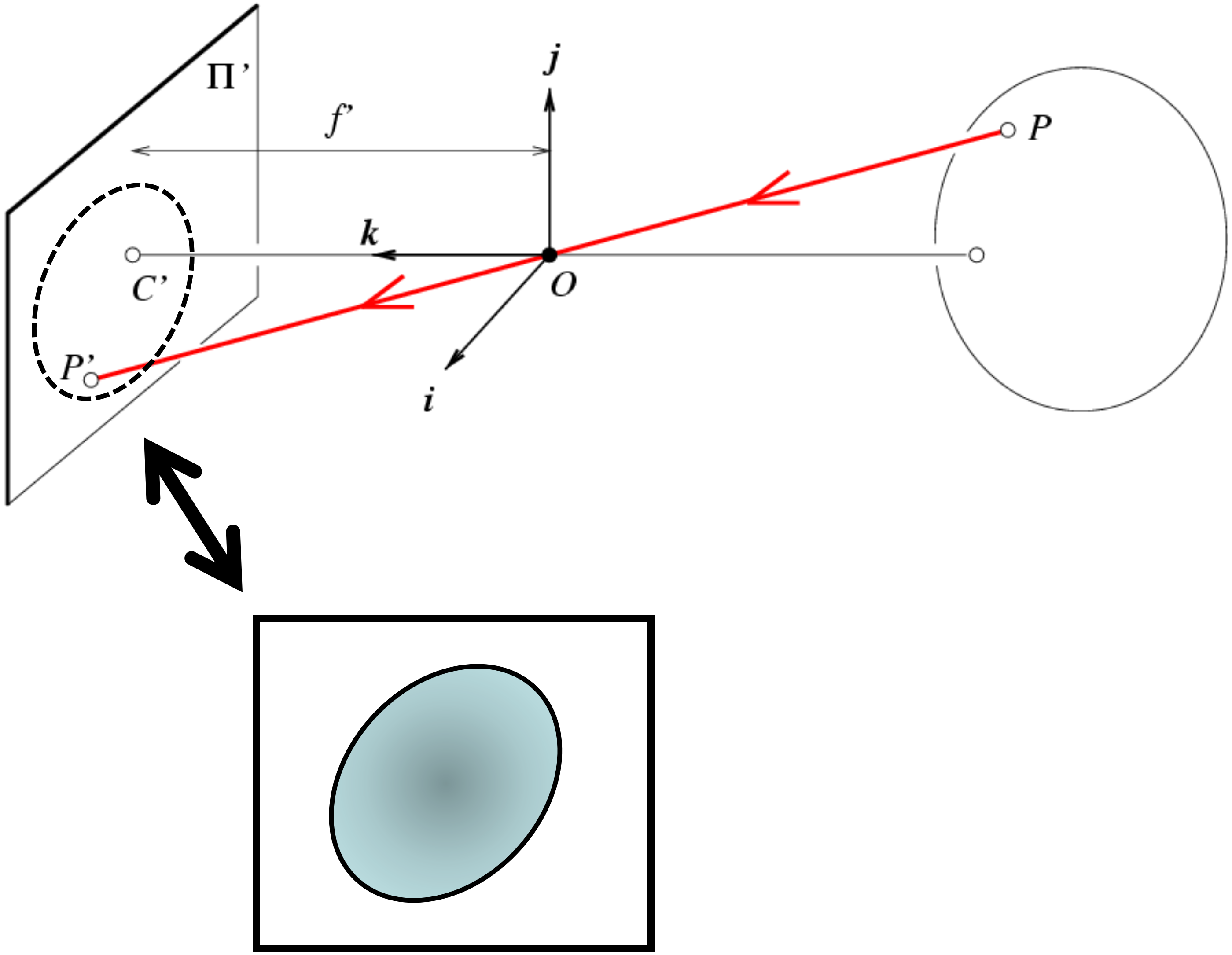
Olmschenk, Greg, and Zhigang Zhu. "3D Hallway Modeling Using a Single Image." Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on. IEEE, 2014.
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6909974

CS231M · Mobile Computer Vision

Next lecture:

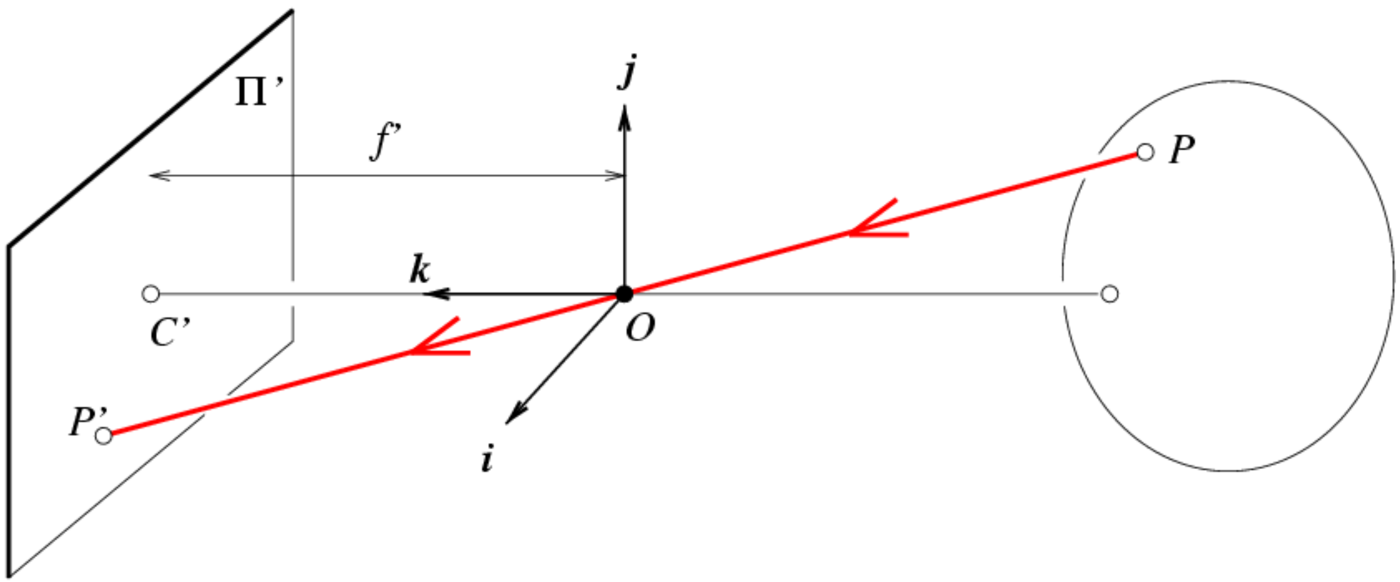
- Geotagging and Geospatial Analysis

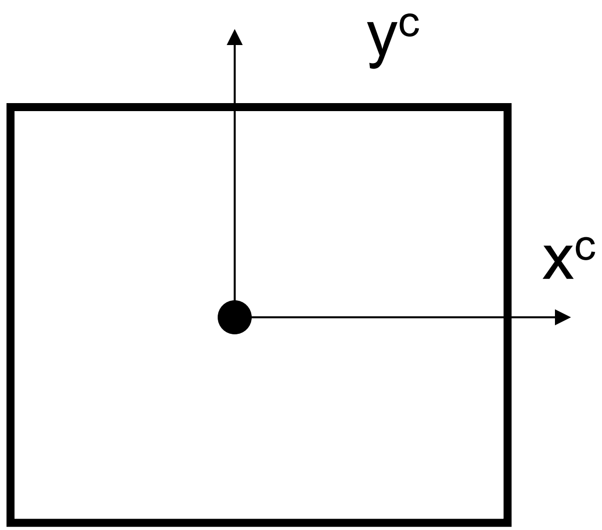
From retina plane to images



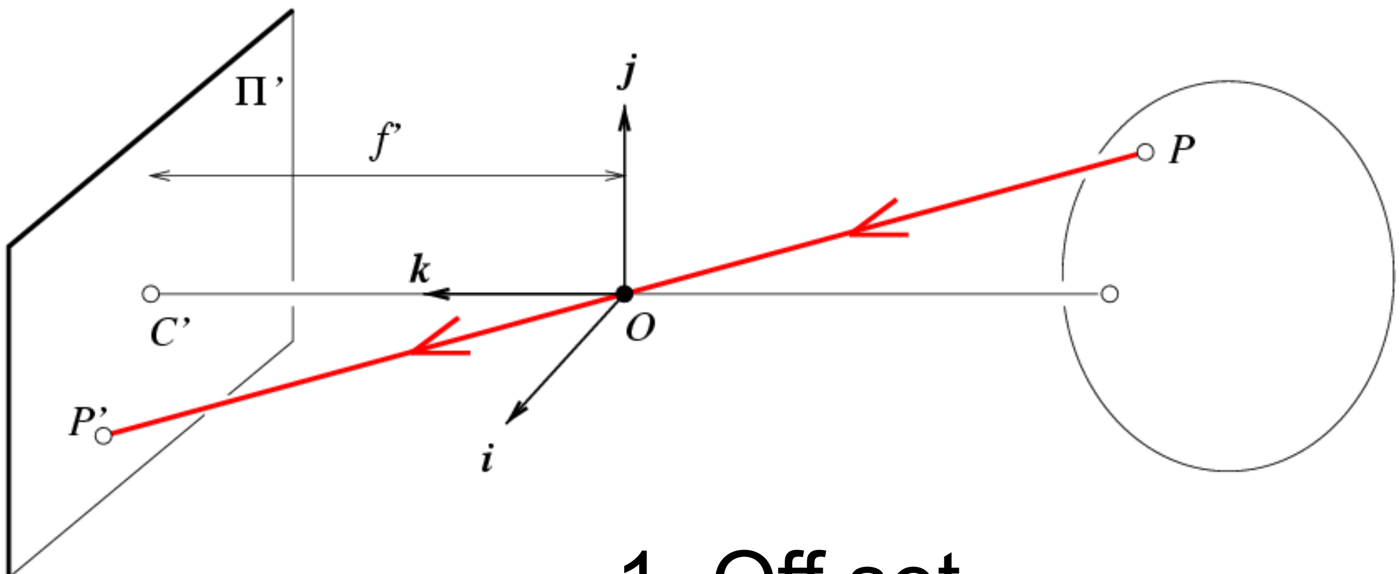
Pixels, bottom-left coordinate systems

From retina plane to images

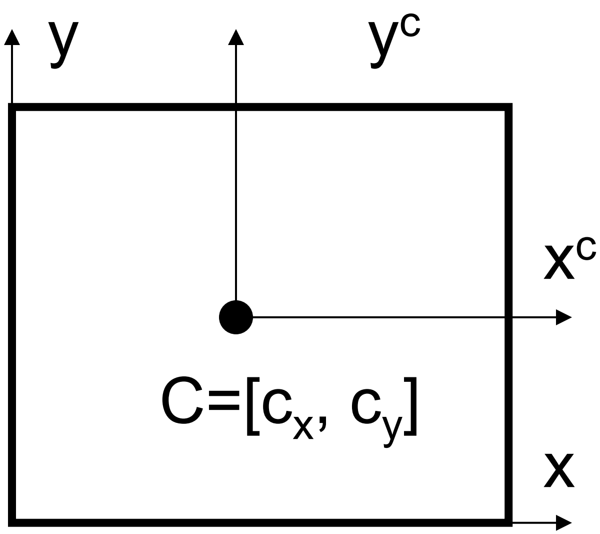




Converting to pixels

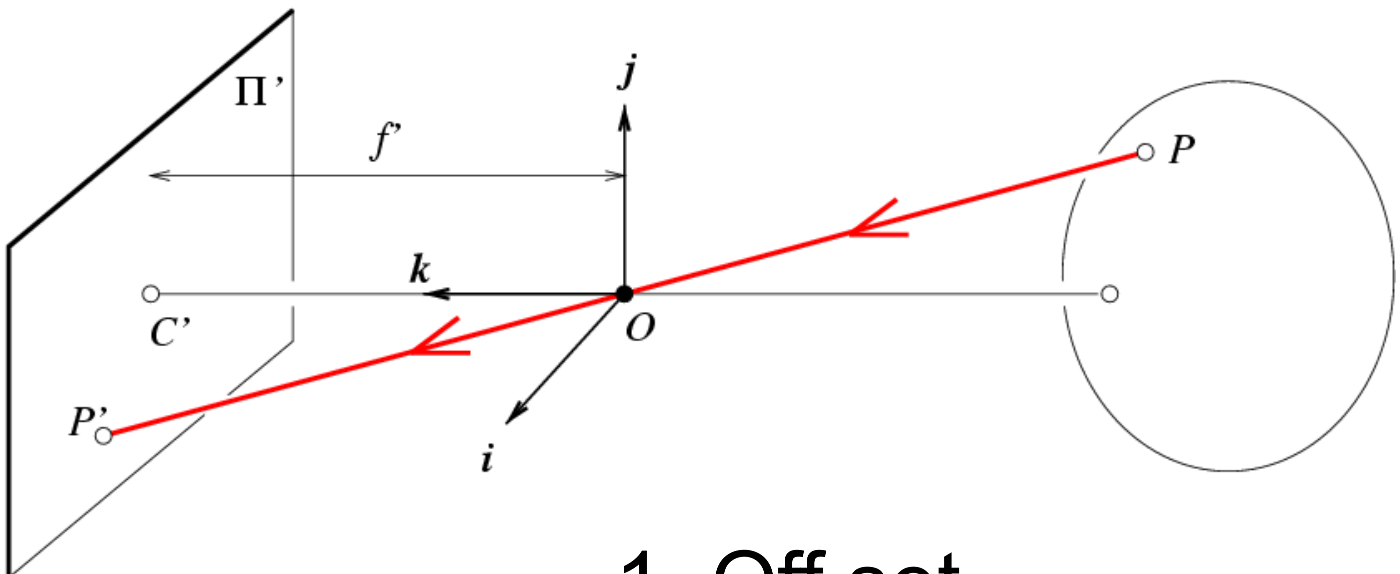


1. Off set

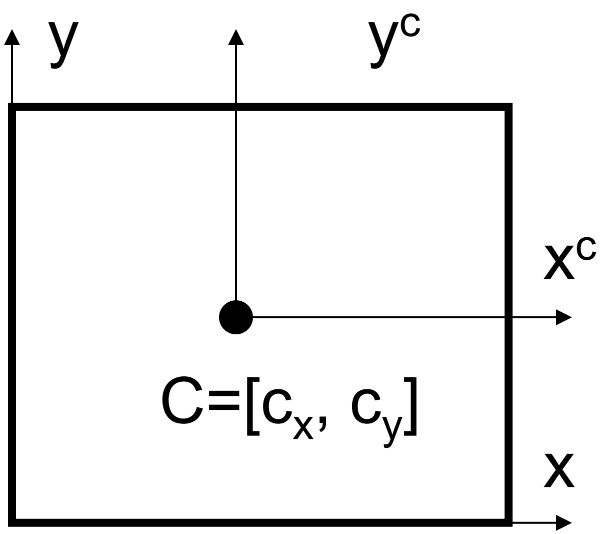


$$(x, y, z) \rightarrow (f\frac{x}{z} + c_x, f\frac{y}{z} + c_y)$$

Converting to pixels



1. Off set
2. From metric to pixels



$$(x, y, z) \rightarrow \left(\underbrace{\boxed{f\,k}}_{\boldsymbol{\alpha}}\frac{x}{z} + c_x,\ \underbrace{\boxed{f\,l}}_{\boldsymbol{\beta}}\frac{y}{z} + c_y\right)$$

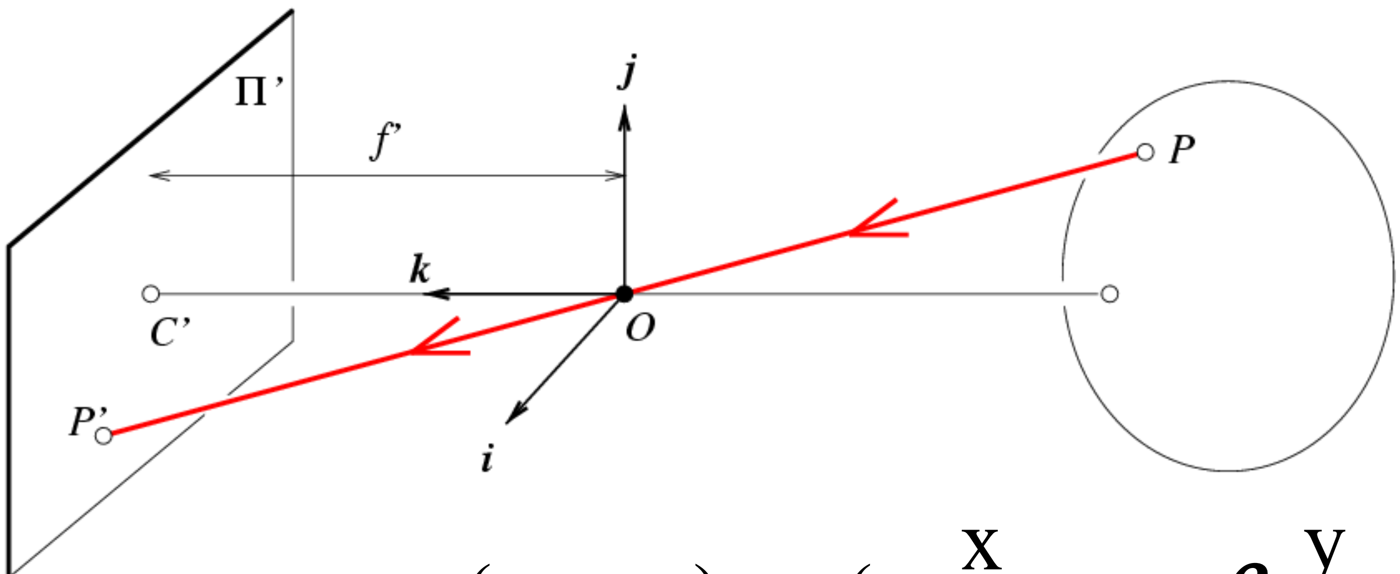
Units: k,l : pixel/m

f : m

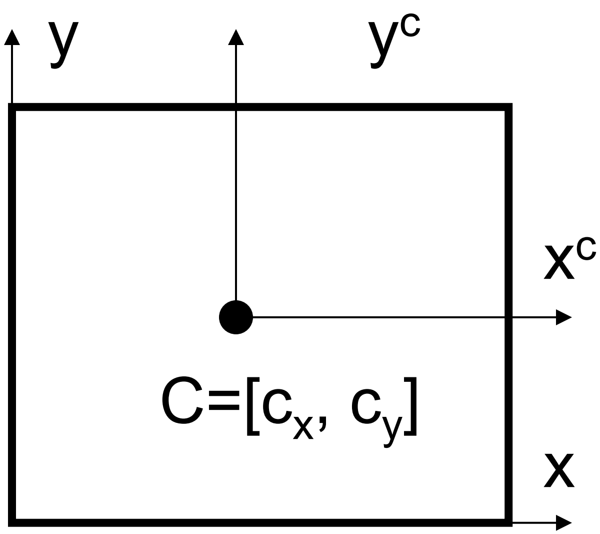
Non-square pixels

$\boldsymbol{\alpha}, \boldsymbol{\beta}$: pixel

Camera Matrix



$$(\mathrm{x}, \mathrm{y}, \mathrm{z}) \rightarrow \left(\boldsymbol{\alpha} \frac{\mathrm{x}}{\mathrm{z}} + \mathrm{c}_{\mathrm{x}}, \boldsymbol{\beta} \frac{\mathrm{y}}{\mathrm{z}} + \mathrm{c}_{\mathrm{y}}\right)$$



- Matrix form?

Homogeneous coordinates

For details see lecture on transformations in CS131A

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

homogeneous image coordinates

$$(x, y, z) \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

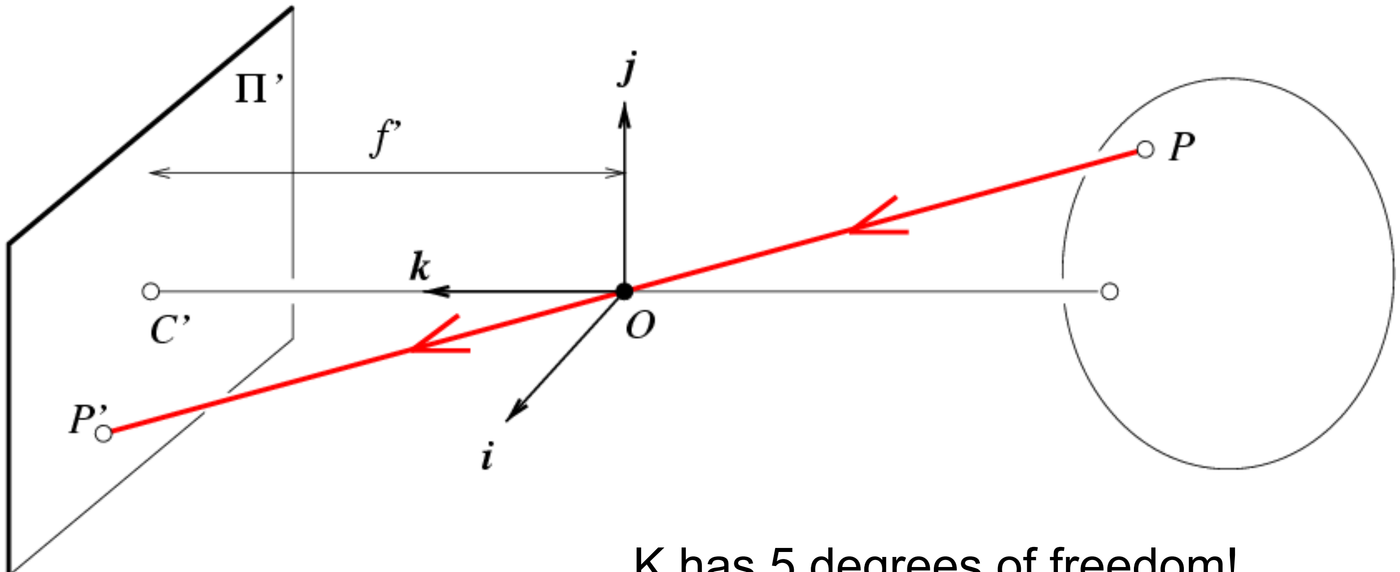
homogeneous scene coordinates

- Converting *from* homogeneous coordinates

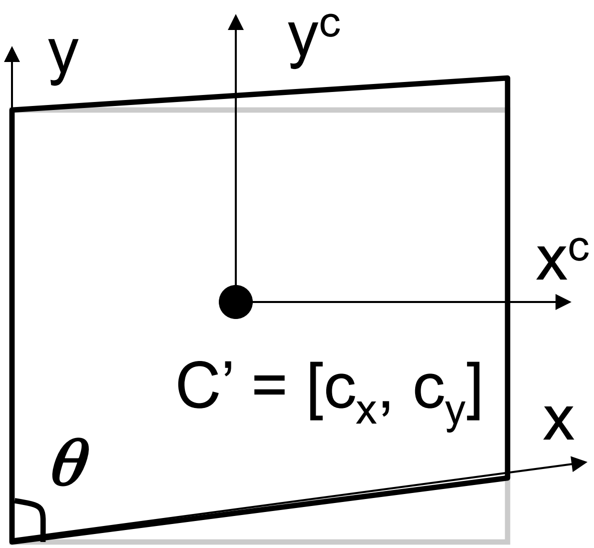
$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow (x/w, y/w, z/w)$$

Camera Skew



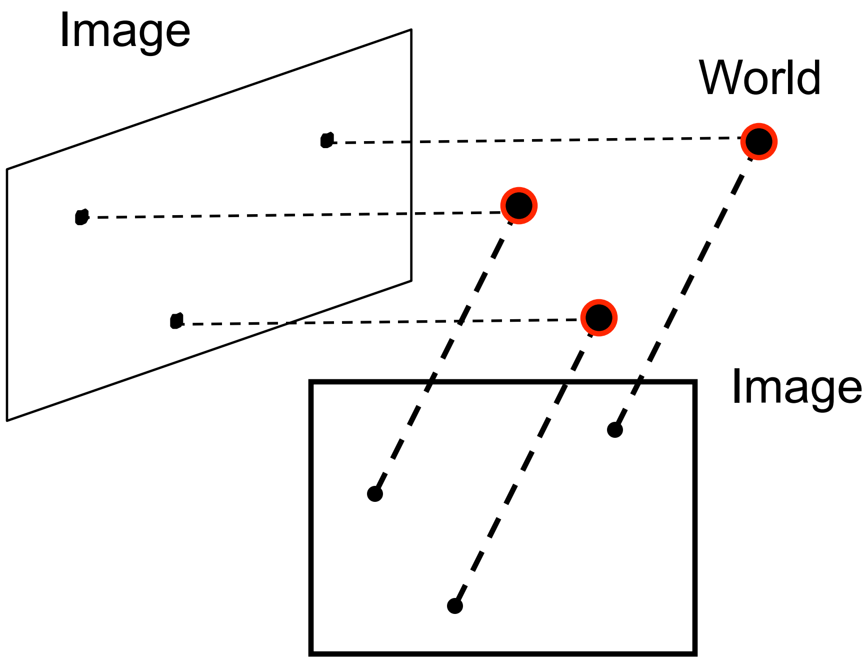
K has 5 degrees of freedom!



$$P' = \begin{bmatrix} \alpha & -\alpha\cot\theta & c_x & 0 \\ 0 & \dfrac{\beta}{\sin\theta} & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$P' = M\,P = K\begin{bmatrix} I & 0 \end{bmatrix} P$$

Affine structure from motion
(simpler problem)

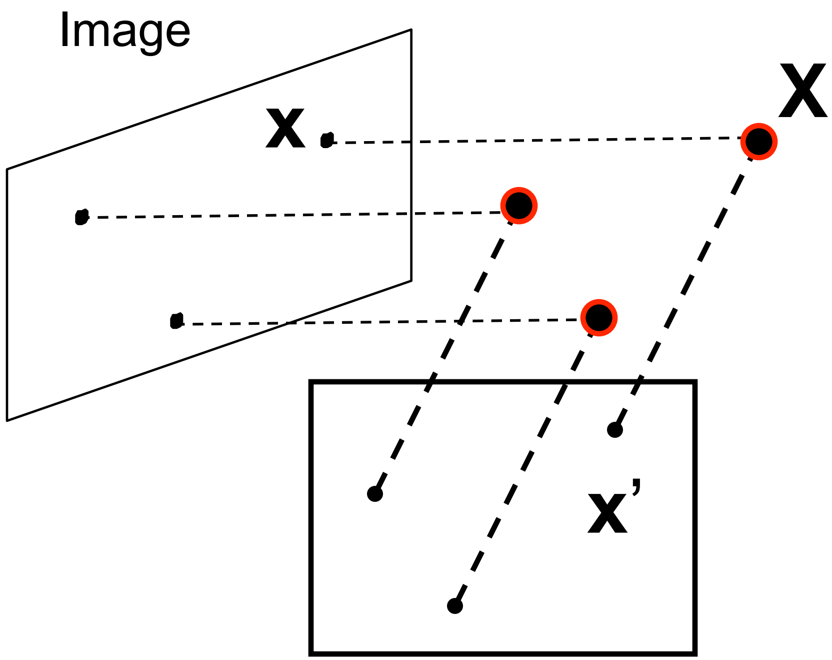


From the m x n correspondences $\mathbf{x}_{ij}$, estimate:

- *m* projection matrices $\mathbf{M}_i$ (affine cameras)
- *n* 3D points $\mathbf{X}_j$

Affine structure from motion

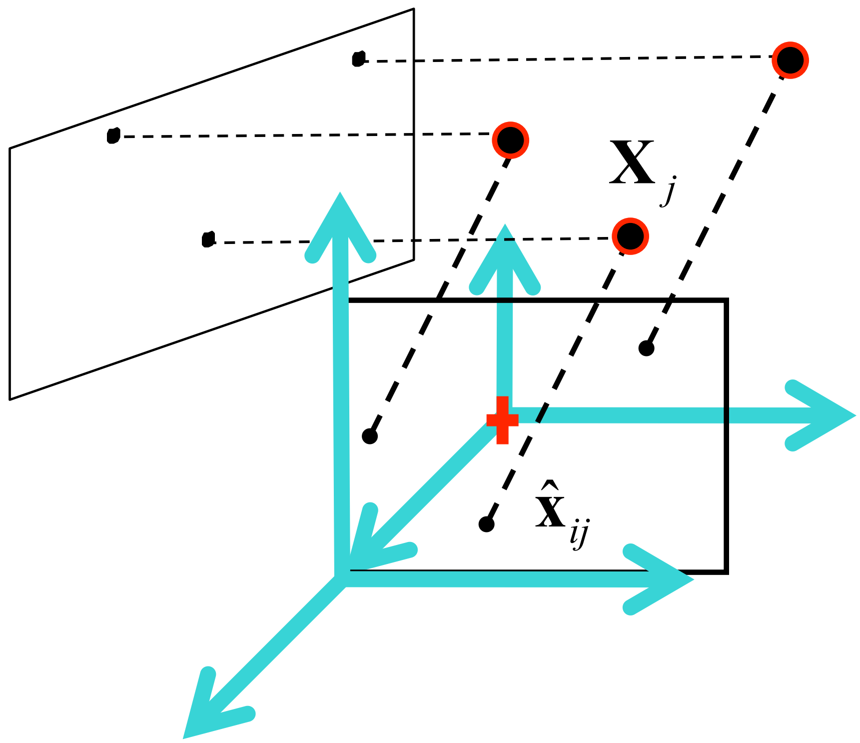
(simpler problem)



Camera matrix M for the affine case

$$\mathbf{x} = \begin{pmatrix} u \\ v \end{pmatrix} = M \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{A}\mathbf{X} + \mathbf{b}; \qquad \mathrm{M} = \begin{bmatrix} \mathrm{A} & \mathrm{b} \end{bmatrix}$$

Centering the data



Normalize points w.r.t. centroids of measurements from each image

$$\mathbf{x}_{ij} = \mathbf{A}\mathbf{X}_j + \mathbf{b} \;\longrightarrow\; \hat{\mathbf{x}}_{ij} = \mathbf{A}_i\mathbf{X}_j$$

For details, see CS231A, lecture 6, 7

A factorization method - factorization

Let's create a $2m \times n$ data (measurement) matrix:

$$\mathbf{D} = \begin{bmatrix} \hat{\mathbf{x}}_{11} & \hat{\mathbf{x}}_{12} & \cdots & \hat{\mathbf{x}}_{1n} \\ \hat{\mathbf{x}}_{21} & \hat{\mathbf{x}}_{22} & \cdots & \hat{\mathbf{x}}_{2n} \\ & & \ddots & \\ \hat{\mathbf{x}}_{m1} & \hat{\mathbf{x}}_{m2} & \cdots & \hat{\mathbf{x}}_{mn} \end{bmatrix} \Bigg\downarrow \text{cameras } (2m)$$

$$\xrightarrow{\hspace{6cm}}$$

points (n)

A factorization method - factorization

Let's create a $2m \times n$ data (measurement) matrix:

$$\mathbf{D} = \begin{bmatrix} \hat{\mathbf{x}}_{11} & \hat{\mathbf{x}}_{12} & \cdots & \hat{\mathbf{x}}_{1n} \\ \hat{\mathbf{x}}_{21} & \hat{\mathbf{x}}_{22} & \cdots & \hat{\mathbf{x}}_{2n} \\ & & \ddots & \\ \hat{\mathbf{x}}_{m1} & \hat{\mathbf{x}}_{m2} & \cdots & \hat{\mathbf{x}}_{mn} \end{bmatrix} = \underbrace{\boxed{\begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_m \end{bmatrix}}}_{\text{M}} \overbrace{\boxed{\begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_n \end{bmatrix}}}^{\text{S}}$$

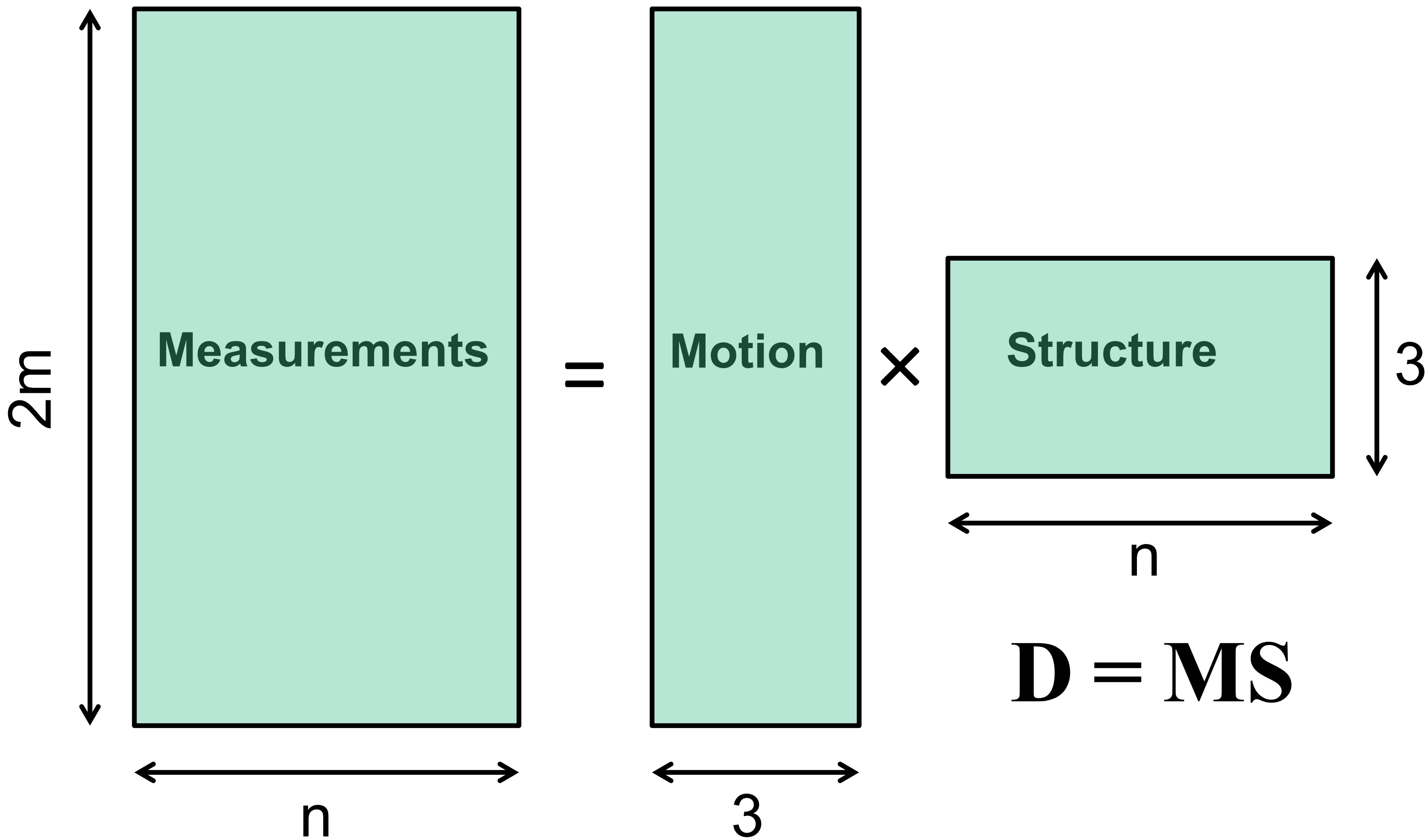
($2m \times$ n)

points ($3 \times n$)

cameras
($2m \times 3$)

The measurement matrix **D = M S** has rank 3
(it's a product of a 2mx3 matrix and 3xn matrix)

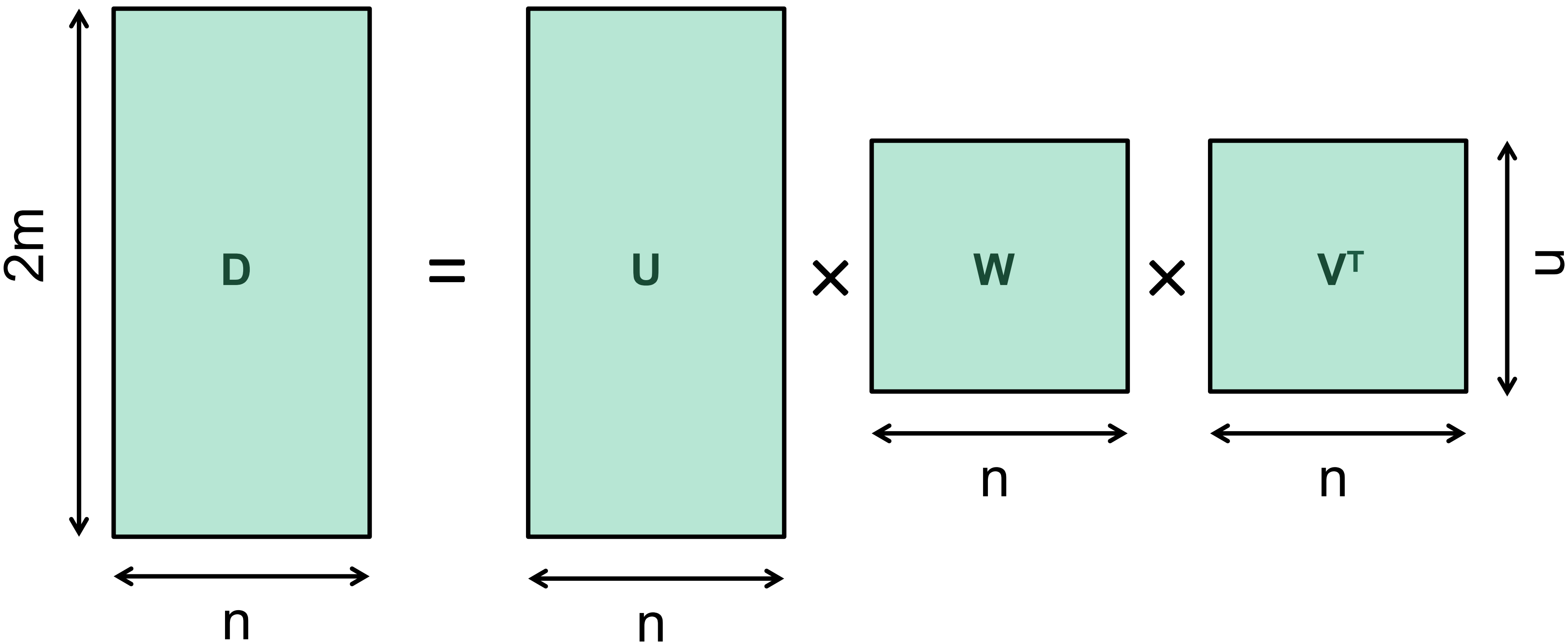
Factorizing the Measurement Matrix



$$\mathbf{D} = \mathbf{MS}$$

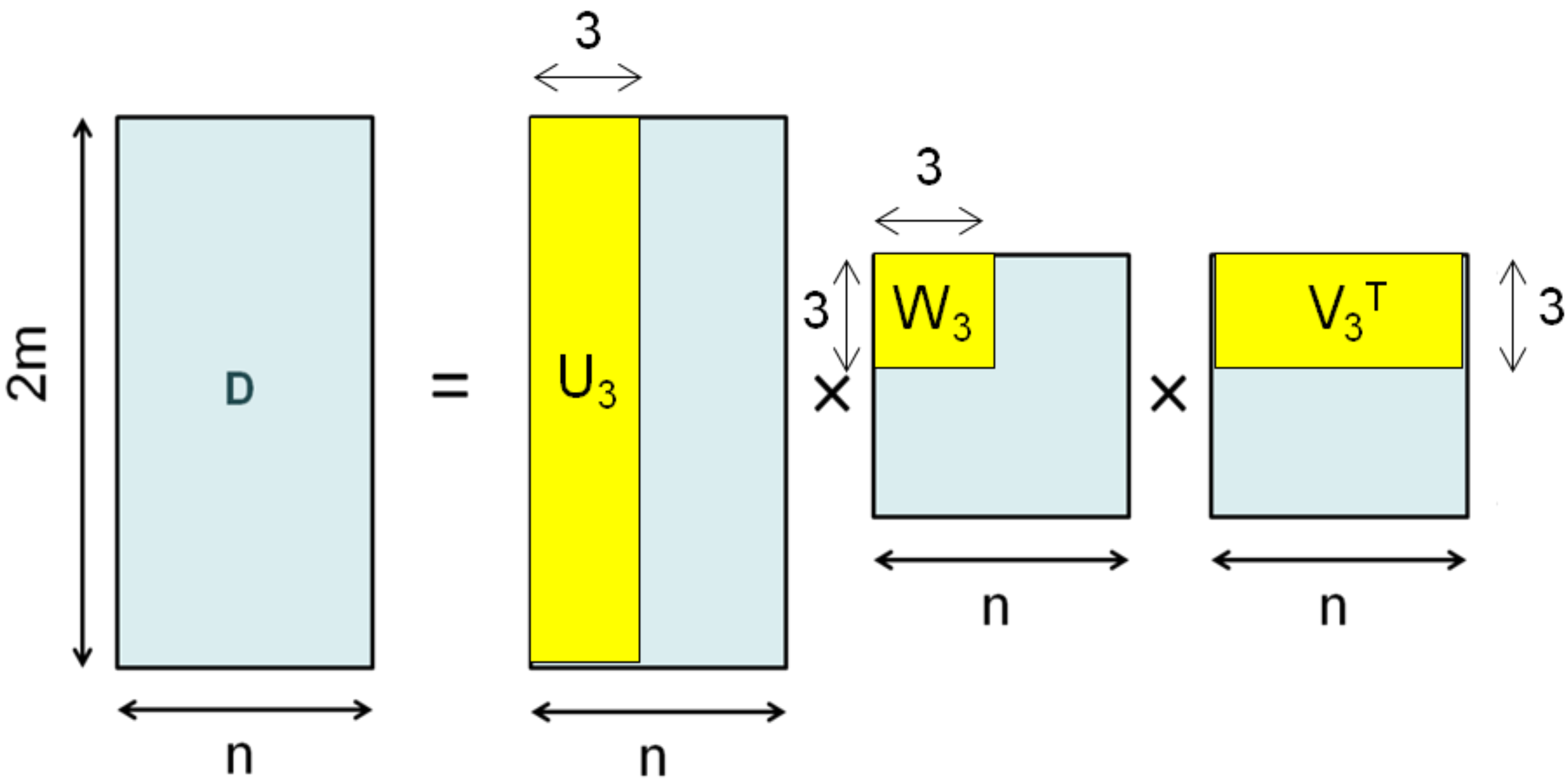
Factorizing the Measurement Matrix

- Singular value decomposition of D:

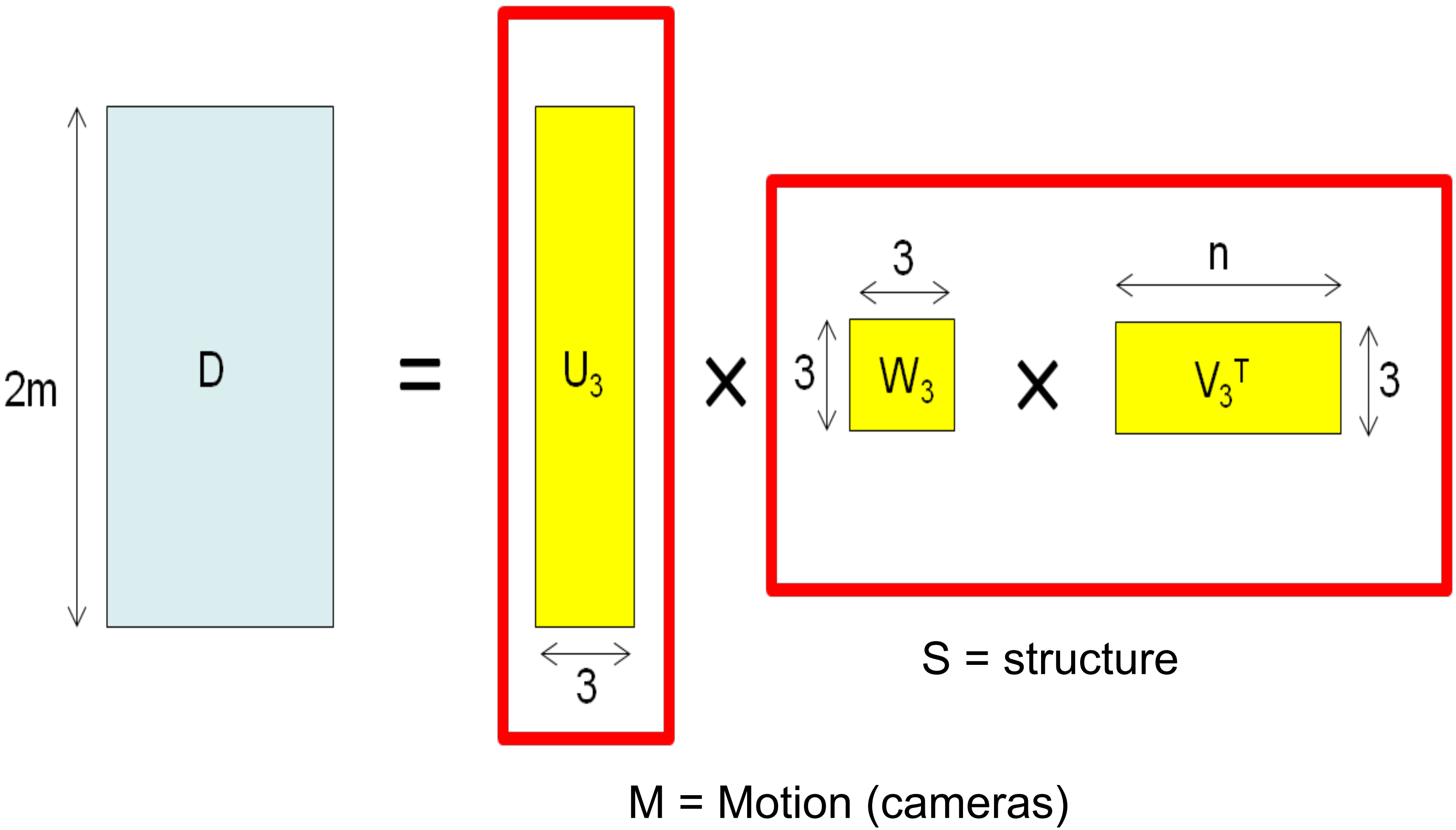


Factorizing the Measurement Matrix

Since rank (D)=3, there are only 3 non-zero singular values



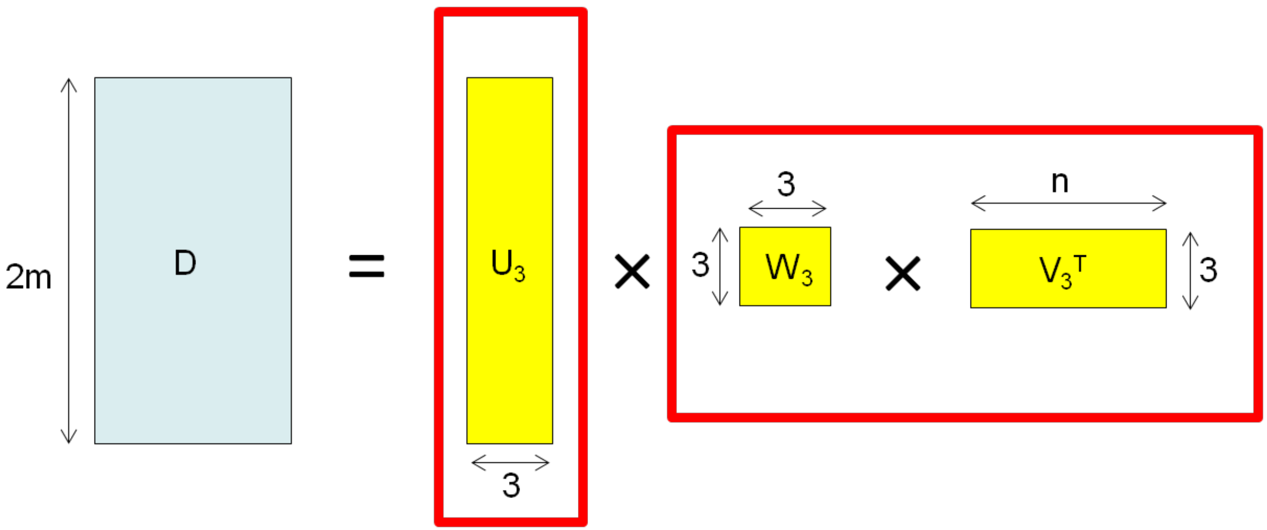
Factorizing the Measurement Matrix



Factorizing the Measurement Matrix

What is the issue here? D has rank>3 because of:

- measurement noise
- affine approximation



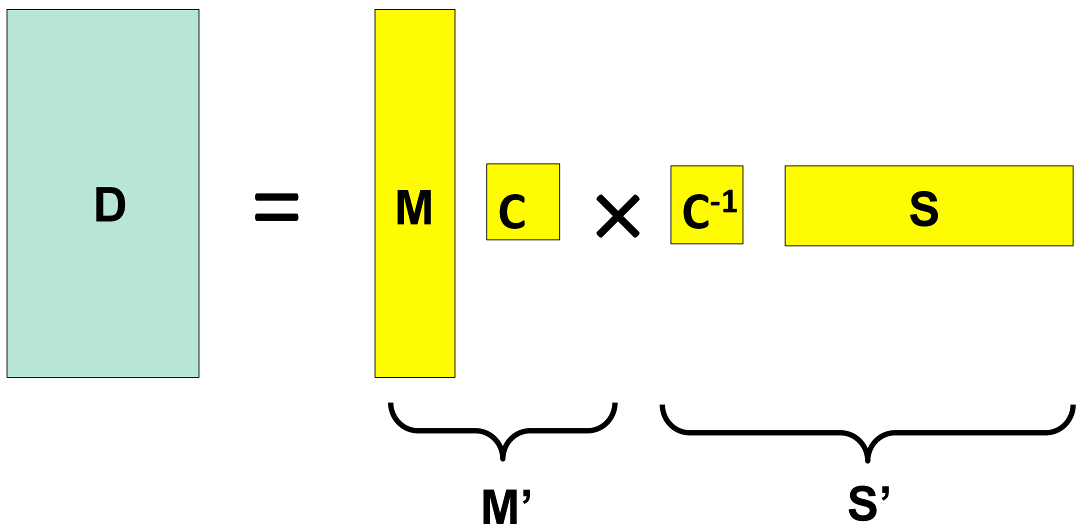
Theorem: When $\mathbf{D}$ has a rank greater than p, $\mathbf{U}_p\mathbf{W}_p\mathbf{V}_p^T$ is the best possible rank-p approximation of $\mathbf{A}$ in the sense of the Frobenius norm.

$$\mathbf{D} = \mathbf{U}_3\mathbf{W}_3\mathbf{V}_3^T$$

$$\begin{cases} \mathbf{A}_0 = \mathbf{U}_3 \\ \mathbf{P}_0 = \mathbf{W}_3\mathbf{V}_3^T \end{cases}$$

$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2} = \sqrt{\sum_{i=1}^{\min\{m,n\}}\sigma_i^2}$$

Affine Ambiguity



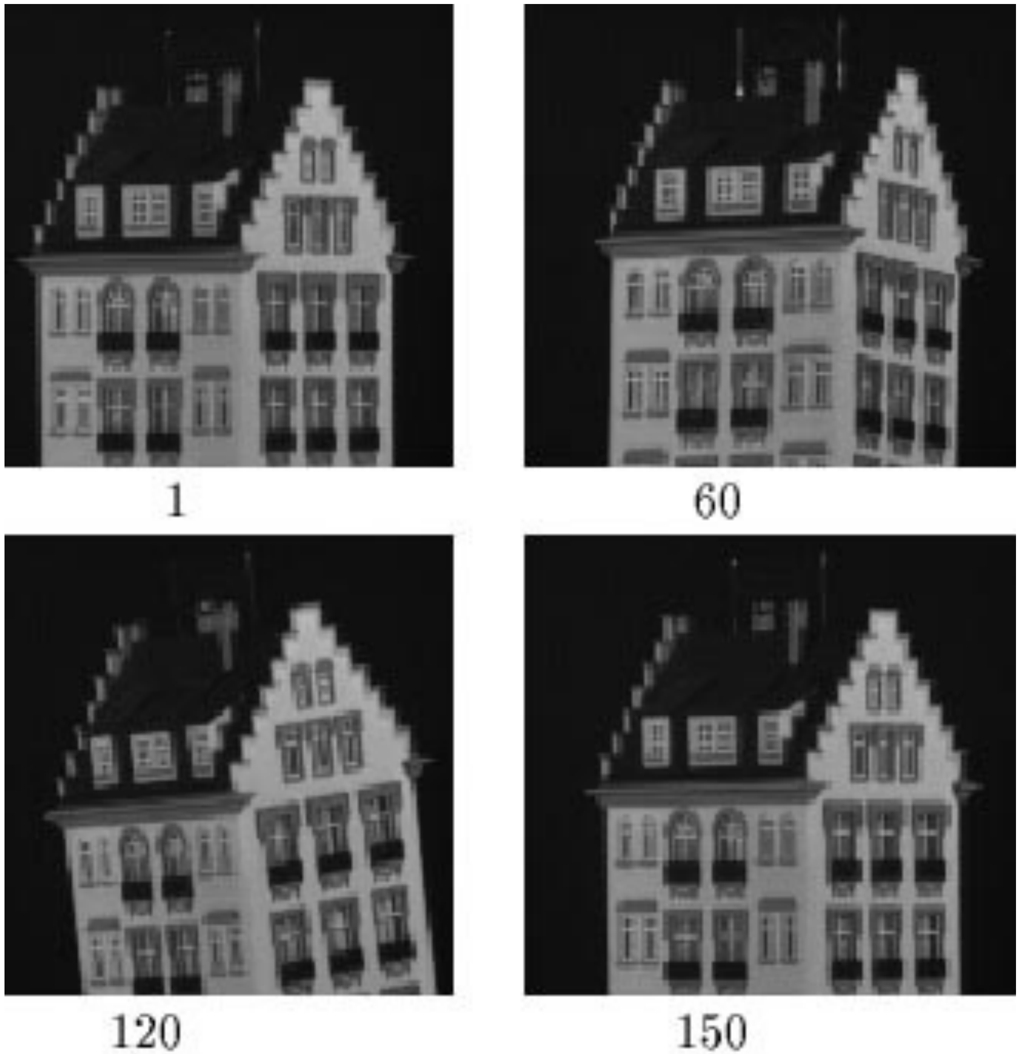
- The decomposition is not unique. We get the same **D** by using any 3×3 matrix **C** and applying the transformations:

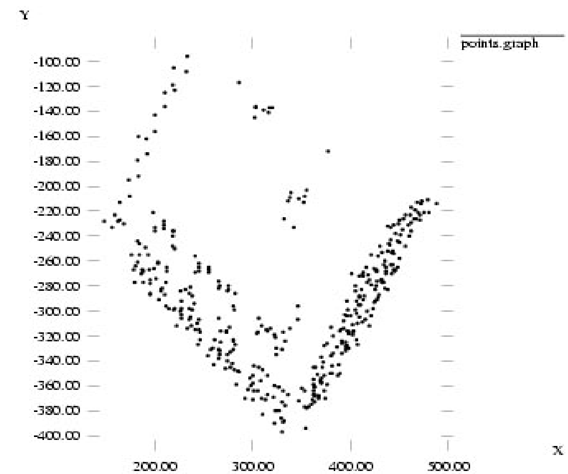
$$\mathbf{M} \rightarrow \mathbf{MC}$$

$$\mathbf{S} \rightarrow \mathbf{C^{-1}S}$$

- Additional constraints must be enforced to resolve this ambiguity

Reconstruction results





C. Tomasi and T. Kanade. [Shape and motion from image streams under orthography: A factorization method.](#) *IJCV*, 9(2):137-154, November 1992.

SfM objective function

- Given point x and rotation and translation R, t

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \mathbf{R}\mathbf{x} + \mathbf{t} \qquad u' = \frac{fx'}{z'} \qquad v' = \frac{fy'}{z'} \qquad \begin{bmatrix} u' \\ v' \end{bmatrix} = \mathbf{P}(\mathbf{x}, \mathbf{R}, \mathbf{t})$$

- Minimize sum of squared reprojection errors:

$$g(\mathbf{X}, \mathbf{R}, \mathbf{T}) = \sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} \cdot \Bigg\| \underbrace{\mathbf{P}(\mathbf{x}_i, \mathbf{R}_j, \mathbf{t}_j)}_{\text{predicted image location}} - \underbrace{\begin{bmatrix} u_{i,j} \\ v_{i,j} \end{bmatrix}}_{\text{observed image location}} \Bigg\|^2$$

Bundle Adjustment

$$\boxed{\begin{aligned}\hat{u}_{ij} &= f(\mathbf{K}, \mathbf{R}_j, \mathbf{t}_j, \mathbf{x}_i) \\ \hat{v}_{ij} &= g(\mathbf{K}, \mathbf{R}_j, \mathbf{t}_j, \mathbf{x}_i)\end{aligned}}$$

- What makes this non-linear minimization hard?
  - many more parameters: potentially slow
  - poorer conditioning (high correlation)
  - potentially lots of outliers
  - gauge (coordinate) freedom

Levenberg-Marquardt

- Iterative non-linear least squares [Press
  - Line ions

$$
\begin{aligned}
\hat{u}_i &= f(\mathbf{m}, \mathbf{x}_i) + \frac{\partial f}{\partial \mathbf{m}} \Delta \mathbf{m} \\
\hat{v}_i &= g(\mathbf{m}, \mathbf{x}_i) + \frac{\partial g}{\partial \mathbf{m}} \Delta \mathbf{m}
\end{aligned}
$$

$$
\sum_i \sigma_i^{-2} \left(\hat{u}_i - u_i + \frac{\partial f}{\partial \mathbf{m}} \Delta \mathbf{m}\right)^2 + \cdots
$$

  - Substitute into log-likelihood equation:

Levenberg-Marquardt

- Iterative non-linear least squares [Press'92]
  – Solve for minimum

$$\frac{\partial C}{\partial \mathbf{m}} = 0$$

Hessian:

error:

$$\begin{aligned}
\mathbf{A}\Delta\mathbf{m} &= \mathbf{b} \\
\mathbf{A} &= \left[\sum_i \sigma_i^{-2} \frac{\partial f}{\partial \mathbf{m}} \left(\frac{\partial f}{\partial \mathbf{m}}\right)^T + \cdots\right] \\
\mathbf{b} &= \left[\sum_i \sigma_i^{-2} \frac{\partial f}{\partial \mathbf{m}} (u_i - \hat{u}_i) + \cdots\right]
\end{aligned}$$

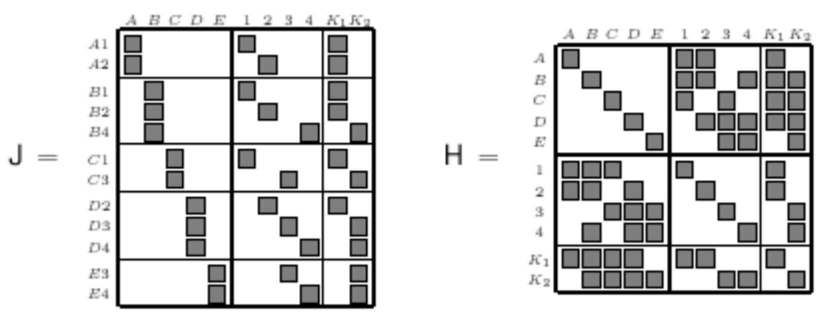
Levenberg-Marquardt

- What if it doesn't converge?
  - Multiply diagonal by $(1 + \lambda)$, increase λ until it does
  - Halve the step size $\Delta\mathbf{m}$
  - Use line search
  - Other ideas?
- Uncertainty analysis: covariance $\Sigma = A^{-1}$
- Is *maximum* likelihood the best idea?
- How to start in vicinity of global minimum?

Lots of parameters: sparsity

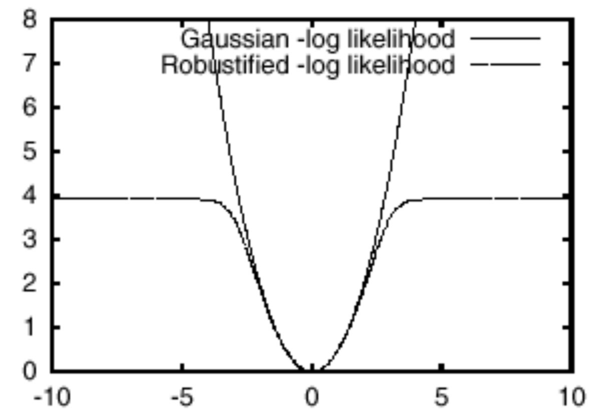
$$
\begin{aligned}
\widehat{u}_{ij} &= f(\mathrm{K}, \mathrm{R}_j, \mathbf{t}_j, \mathbf{x}_i) \\
\widehat{v}_{ij} &= g(\mathrm{K}, \mathrm{R}_j, \mathbf{t}_j, \mathbf{x}_i)
\end{aligned}
$$

- Only a few entries in Jacobian are non-zero $\dfrac{\partial \widehat{u}_{ij}}{\partial \mathrm{K}}, \quad \dfrac{\partial \widehat{u}_{ij}}{\partial \mathrm{R}_j}, \quad \dfrac{\partial \widehat{u}_{ij}}{\partial \mathbf{t}_j}, \quad \dfrac{\partial \widehat{u}_{ij}}{\partial \mathbf{x}_i},$



Robust error models

- Outlier rejection
  - use robust penalty applied to each set of joint measurements



$$\sum_i \sigma_i^{-2} \rho\left(\sqrt{(u_i - \hat{u}_i)^2 + (v_i - \hat{v}_i)^2}\right)$$

  - for extremely bad data, use random sampling [RANSAC, Fischler & Bolles, CACM'81]