Geometry of a single camera

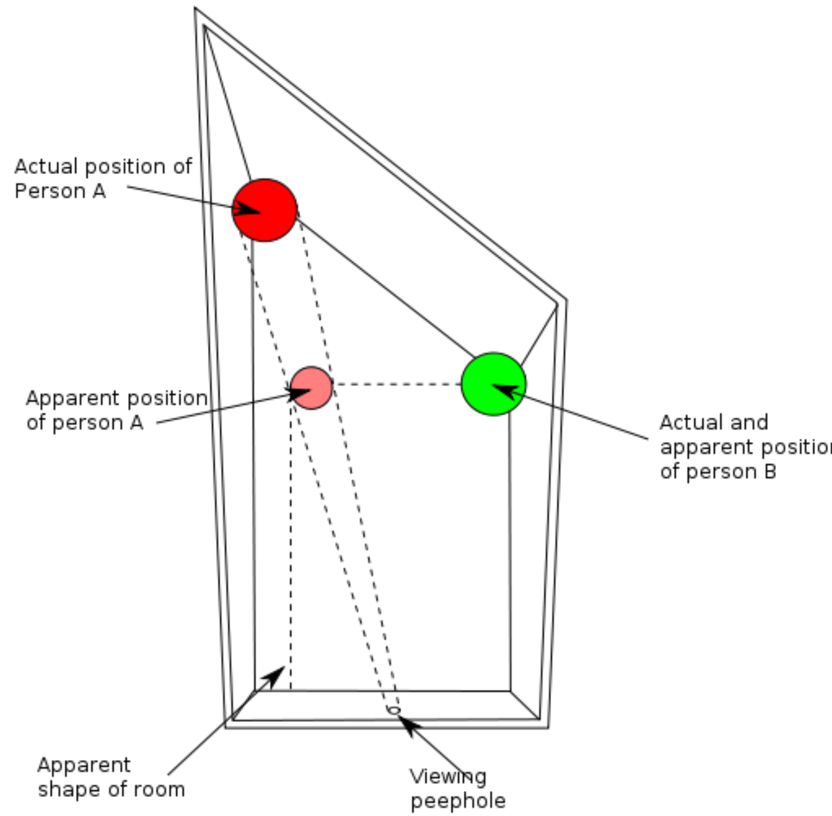
Slides from Derek Hoiem, Svetlana Lazebnik

Our goal: Recovery of 3D structure

J. Vermeer, *Music Lesson*, 1662

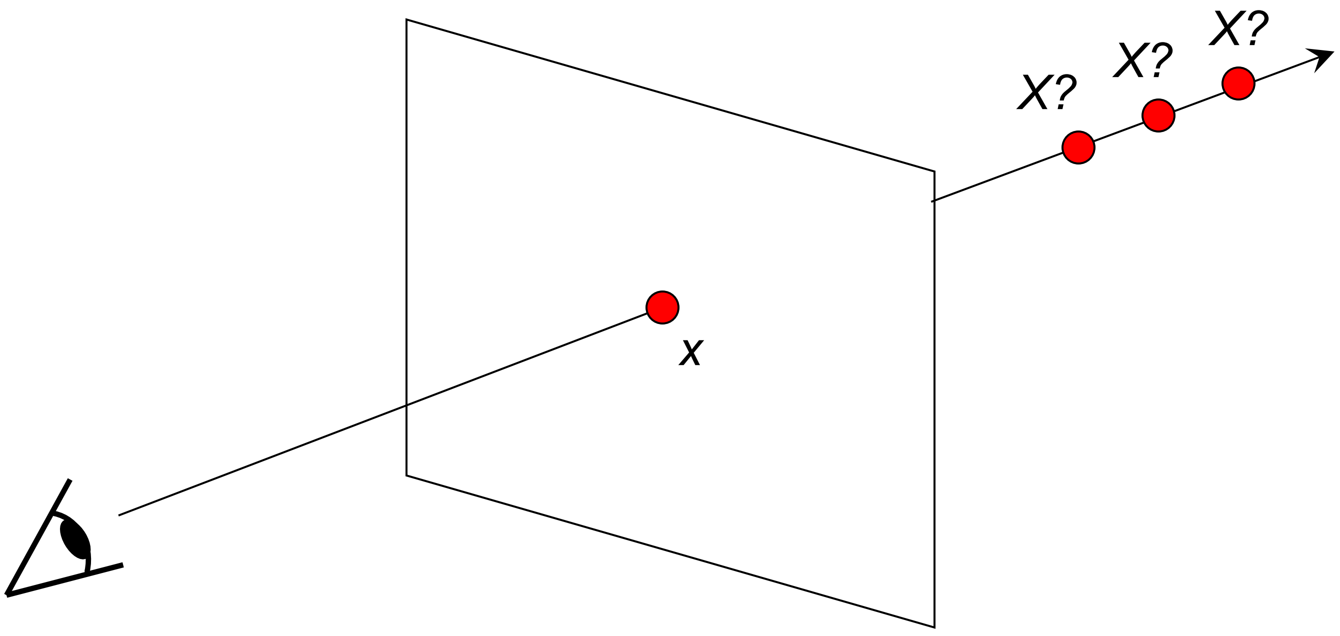
A. Criminisi, M. Kemp, and A. Zisserman, Bringing Pictorial Space to Life: computer techniques for the analysis of paintings, *Proc. Computers and the History of Art*, 2002

Things aren't always as they appear…



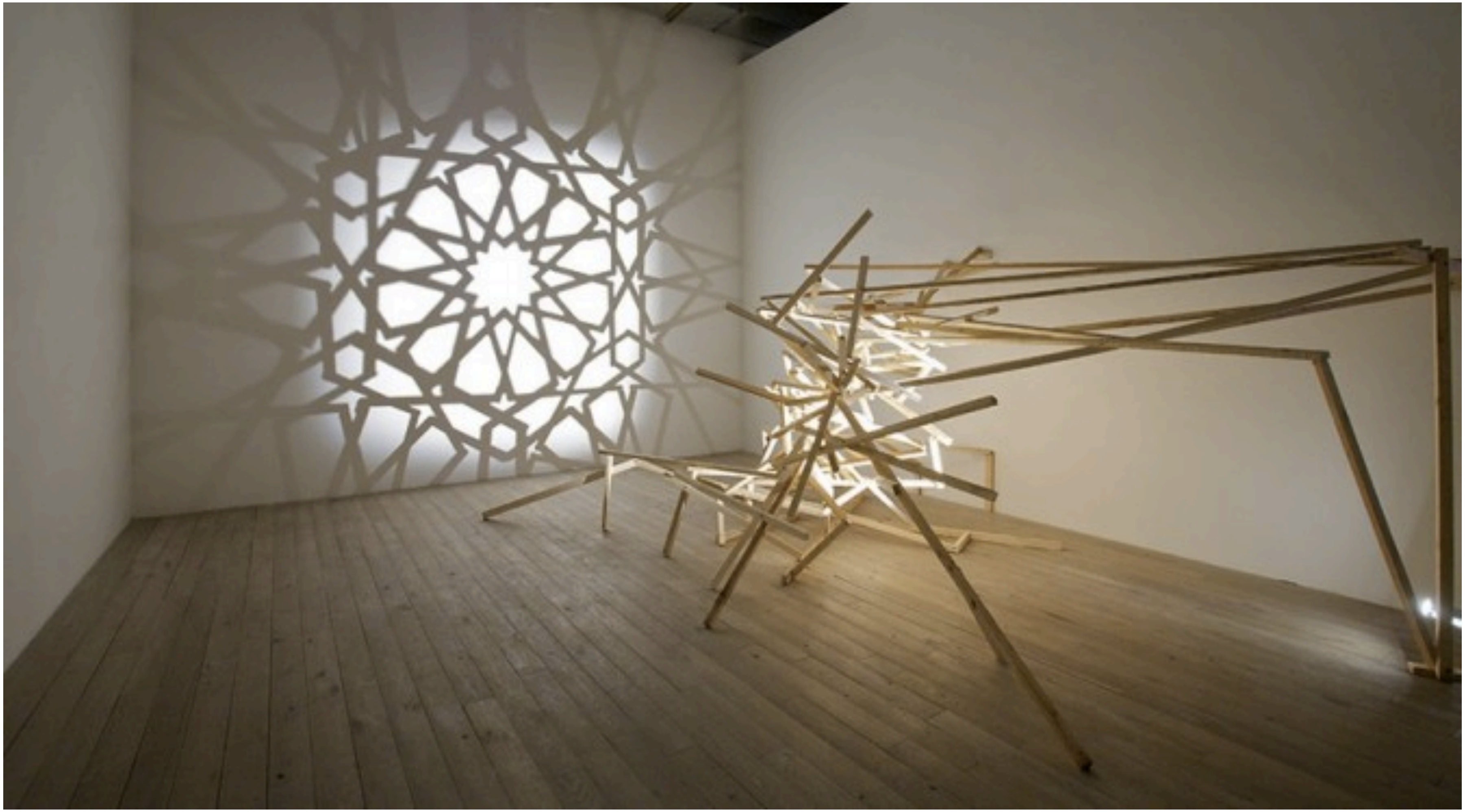
http://en.wikipedia.org/wiki/Ames_room

Single-view ambiguity



Single-view ambiguity

Single-view ambiguity



[Rashad Alakbarov shadow sculptures]

Anamorphic perspective

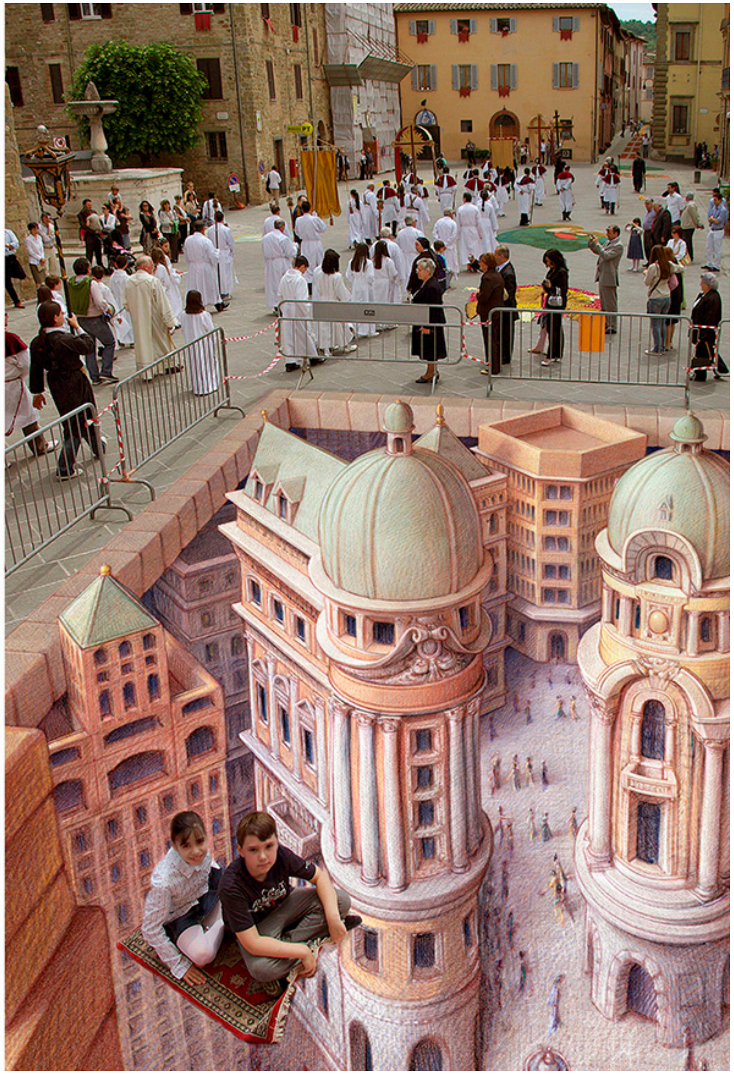


Image source

Anamorphic perspective

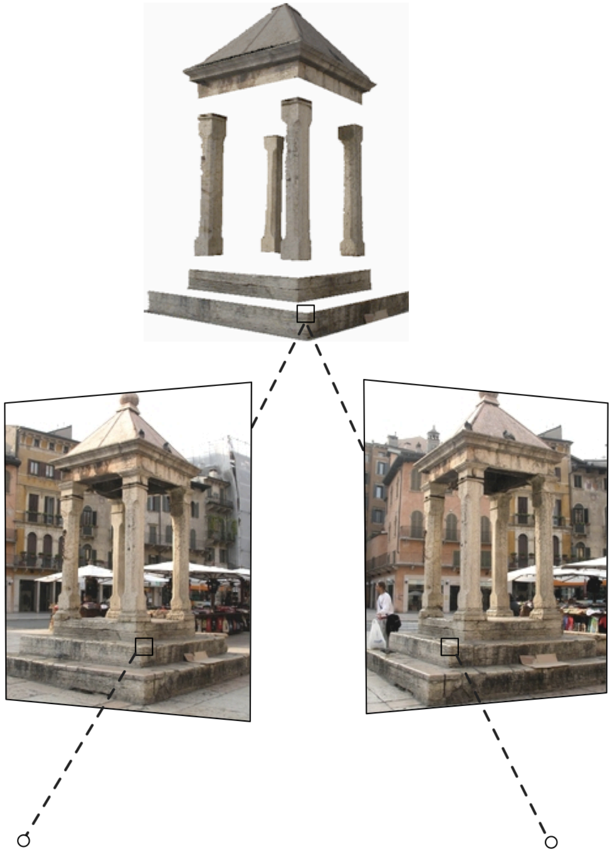
H. Holbein The Younger, *The Ambassadors*, 1533

https://en.wikipedia.org/wiki/Anamorphosis

Our goal: Recovery of 3D structure

- When certain assumptions hold, we can recover structure from a single view

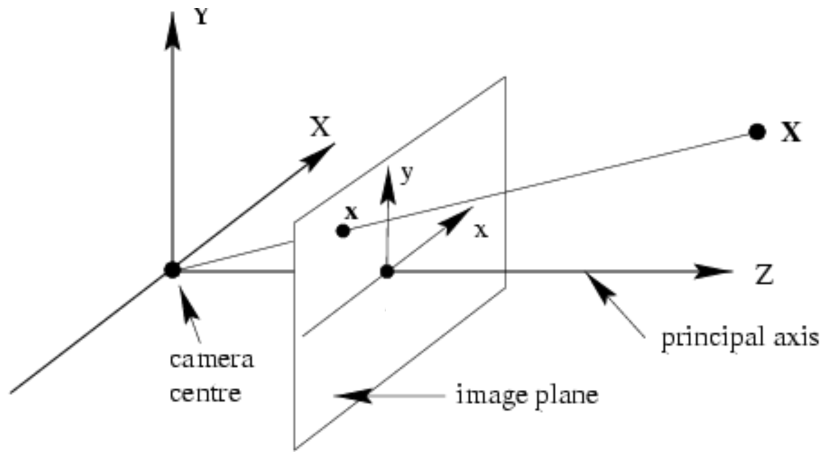
- In general, we need *multi-view geometry*



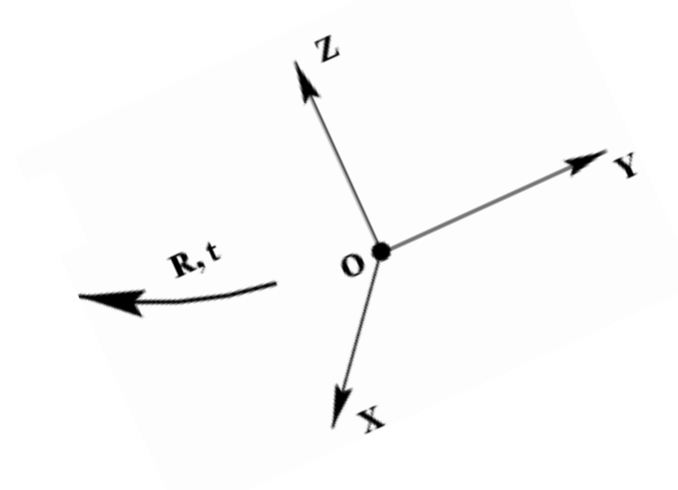
[Image source](#)

- But first, we need to understand the geometry of a single camera…

Camera calibration



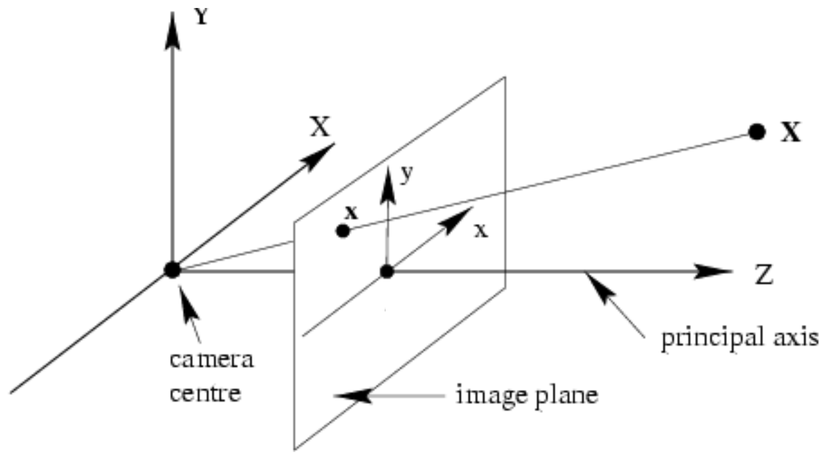
world coordinate system

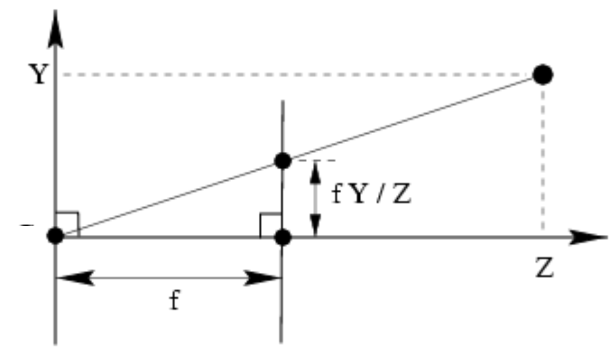


- **Normalized (camera) coordinate system:** camera center is at the origin, the *principal axis* is the z-axis, x and y axes of the image plane are parallel to x and y axes of the world

- **Camera calibration:** figuring out transformation from *world* coordinate system to *image* coordinate system

Review: Pinhole camera model



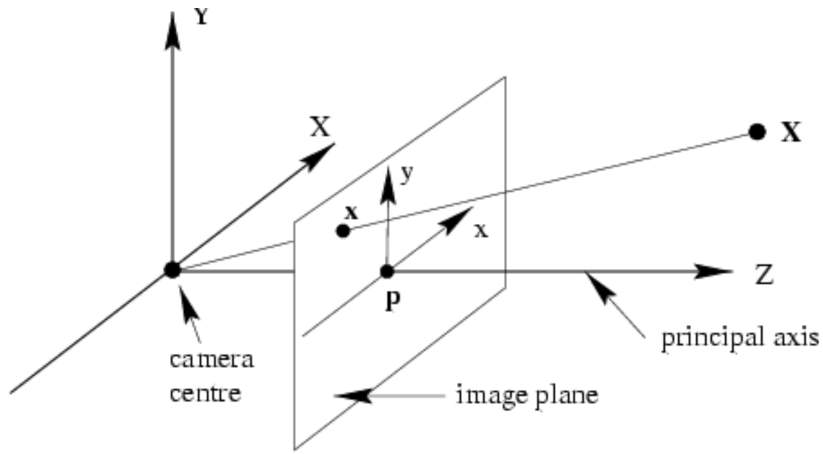


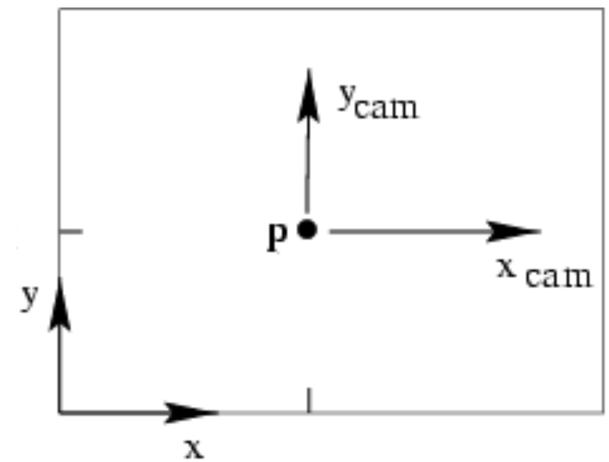
$$(X, Y, Z) \mapsto (f\,X / Z, f\,Y / Z)$$

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} f\,X \\ f\,Y \\ Z \end{pmatrix} = \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

$$\lambda \mathbf{x} = \mathbf{P}\mathbf{X}$$

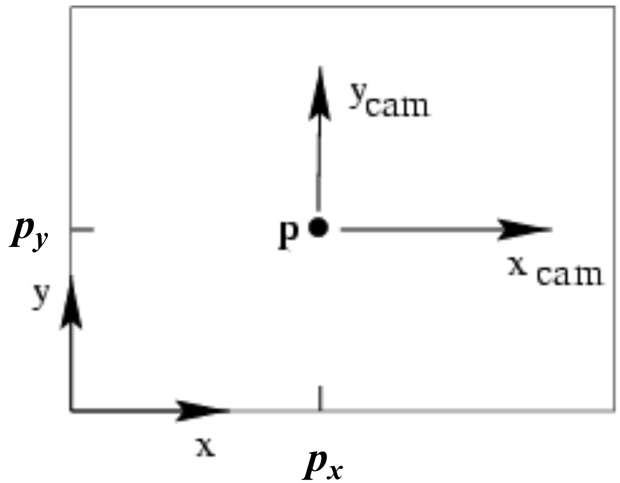
Principal point





- **Principal point (p):** point where principal axis intersects the image plane
- Normalized coordinate system: origin of the image is at the principal point
- Image coordinate system: origin is in the corner

Principal point offset

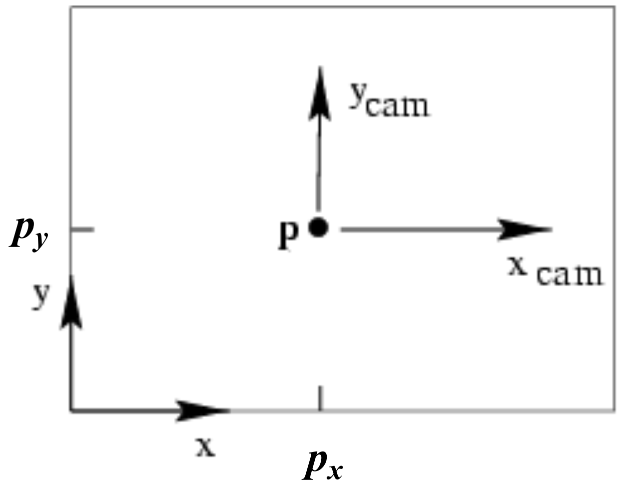


We want the principal point to map to (p_x, p_y) instead of (0,0)

$$(X,Y,Z) \mapsto (f\,X/Z + p_x,\ f\,Y/Z + p_y)$$

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} f\,X + Z\,p_x \\ f\,Y + Z\,p_y \\ Z \end{pmatrix} = \begin{bmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

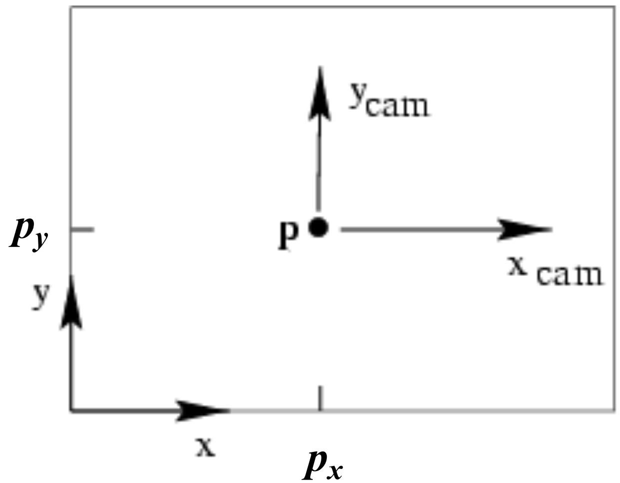
Principal point offset



principal point: (p_x, p_y)

$$\begin{bmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Principal point offset



principal point: (p_x, p_y)

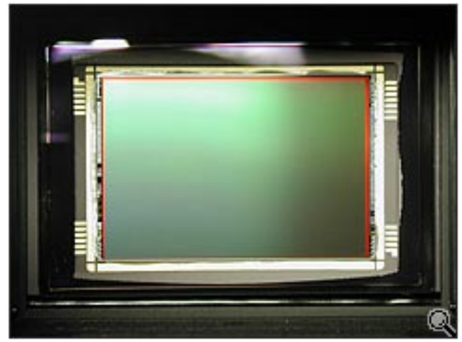
$$\begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & 0 \\ & 1 & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{bmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

calibration matrix K

projection matrix [I | 0]

$$\mathrm{P} = \mathrm{K}\left[\mathrm{I} \mid 0\right]$$

Pixel coordinates



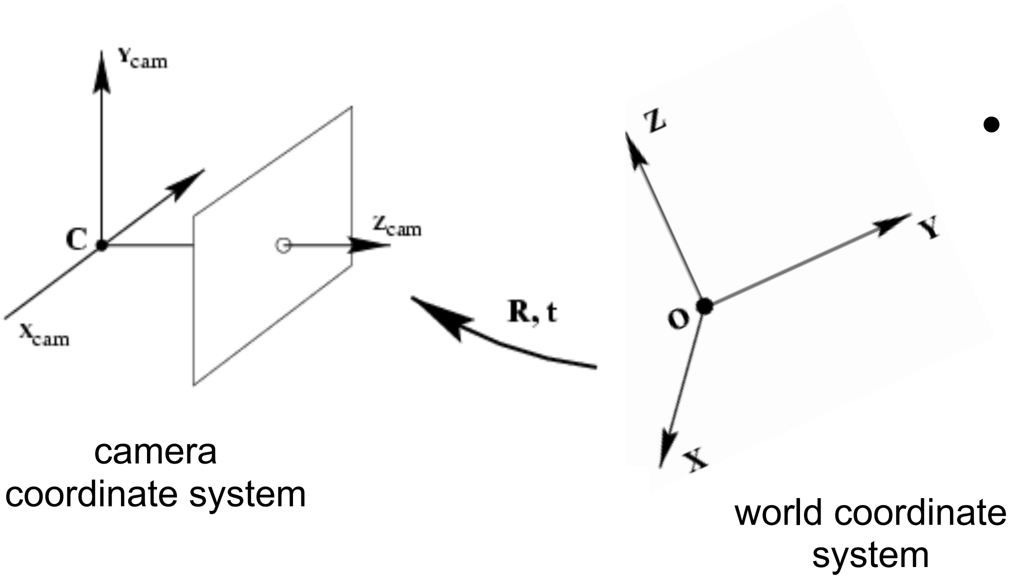
Pixel size: $\frac{1}{m_x} \times \frac{1}{m_y}$

m_x pixels per meter in horizontal direction,
m_y pixels per meter in vertical direction

$$K = \begin{bmatrix} m_x & & \\ & m_y & \\ & & 1 \end{bmatrix} \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & & \beta_x \\ & \alpha_y & \beta_y \\ & & 1 \end{bmatrix}$$

pixels/m m pixels

Camera rotation and translation



- In general, the *camera* coordinate frame will be related to the *world* coordinate frame by a rotation and a translation

- Conversion from world to camera coordinate system (in non-homogeneous coordinates):

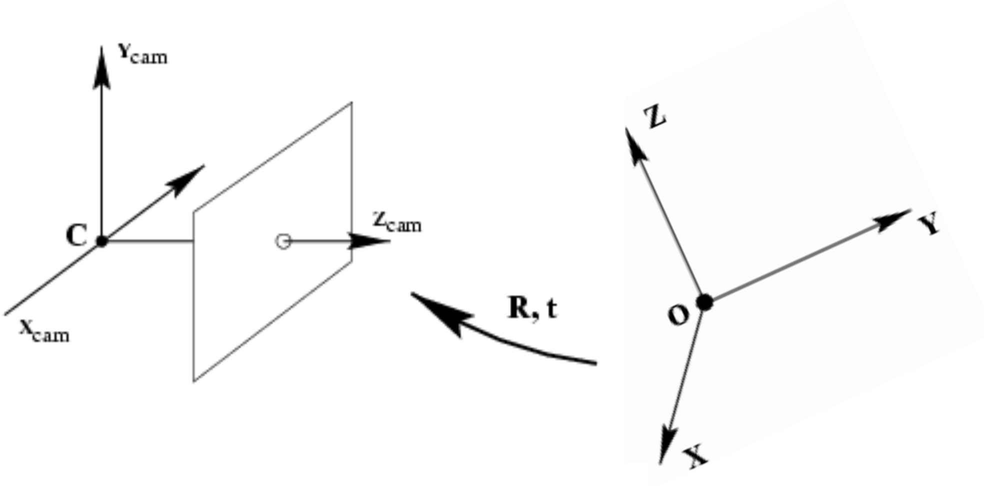
$$\widetilde{X}_{cam} = R\left(\widetilde{X} - \widetilde{C}\right)$$

coords. of point in camera frame

coords. of a point in world frame

coords. of camera center in world frame

Camera rotation and translation

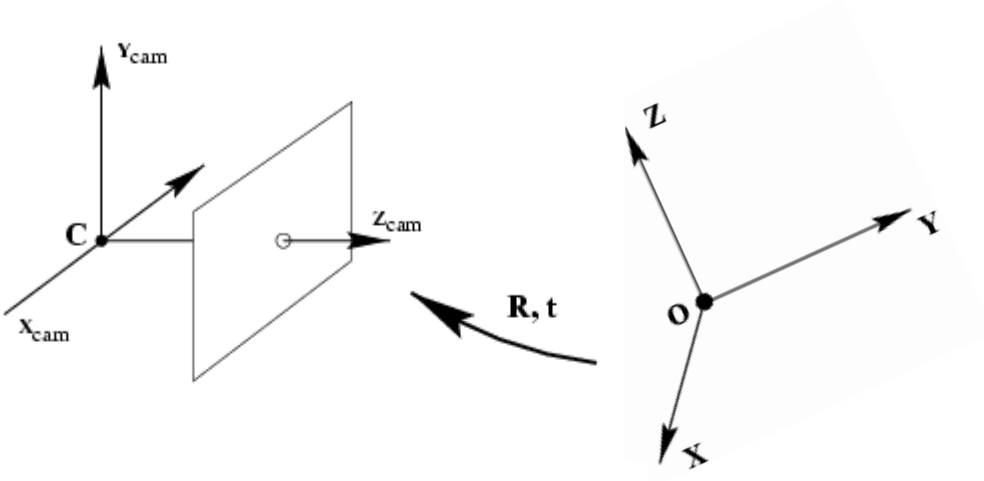


$$\widetilde{X}_{cam} = R\left(\widetilde{X} - \widetilde{C}\right)$$

$$\begin{pmatrix} \widetilde{X}_{cam} \\ 1 \end{pmatrix} = \begin{bmatrix} R & -R\widetilde{C} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} \widetilde{X} \\ 1 \end{pmatrix}$$

3D transformation
matrix (4 x 4)

Camera rotation and translation

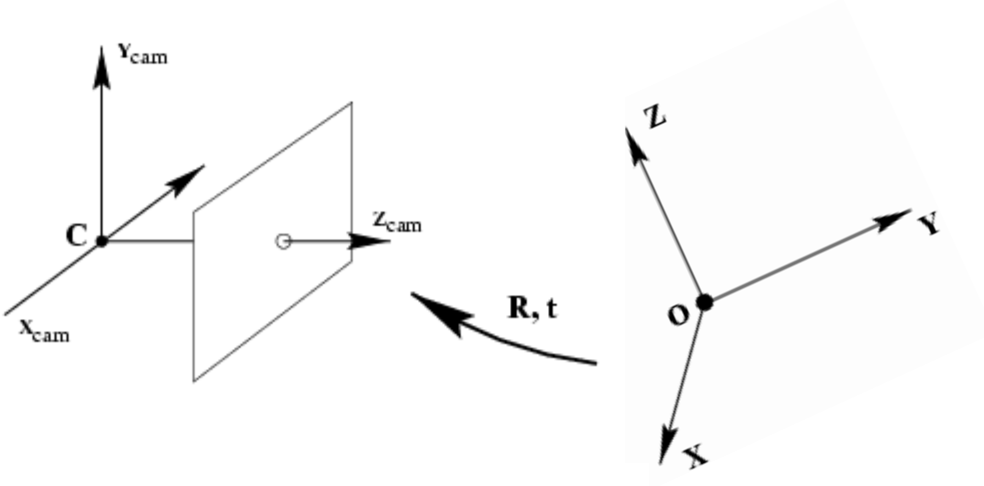


$$\tilde{X}_{cam} = R\left(\tilde{X} - \tilde{C}\right)$$

$$X_{cam} = \begin{bmatrix} R & -R\tilde{C} \\ 0 & 1 \end{bmatrix} X$$

3D transformation
matrix (4 x 4)

Camera rotation and translation



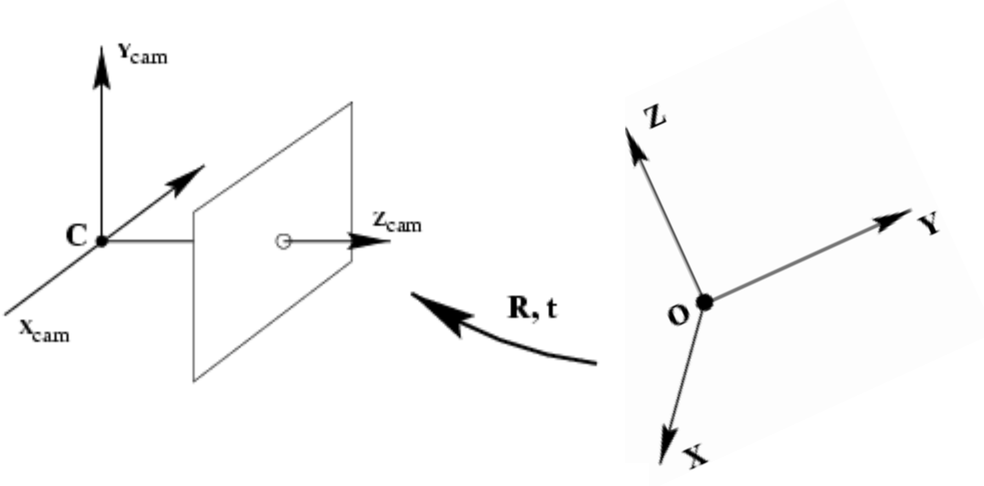
$$\mathbf{x} = \mathbf{K}\left[\mathbf{I} \mid \mathbf{0}\right]\begin{bmatrix} \mathbf{R} & -\mathbf{R}\tilde{\mathbf{C}} \\ \mathbf{0} & 1 \end{bmatrix}\mathbf{X}$$

2D transformation matrix (3 x 3)

perspective projection matrix (3 x 4)

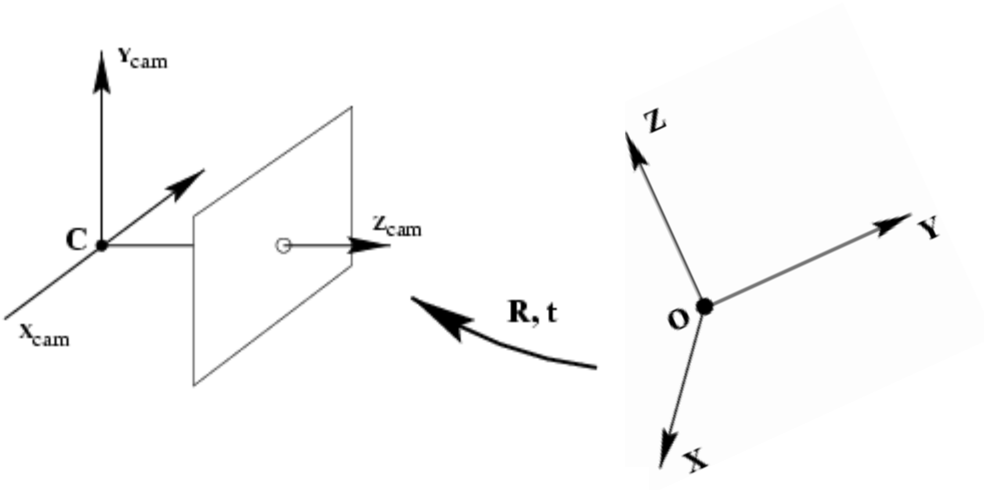
3D transformation matrix (4 x 4)

Camera rotation and translation



$$x = K\left[R \mid -R\tilde{C}\right]X$$

Camera rotation and translation

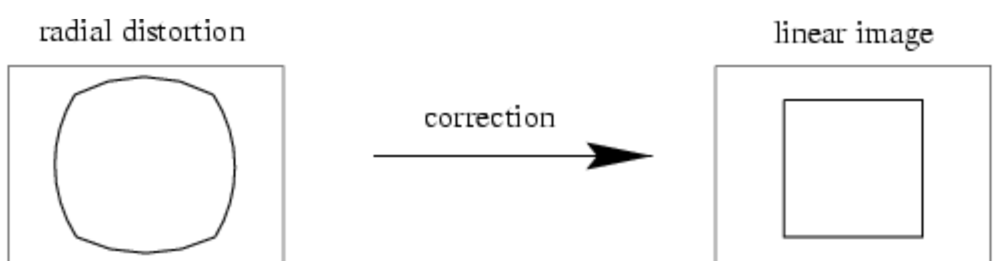


$$x = K[R \mid t]X \qquad t = -R\tilde{C}$$

Camera parameters $\mathbf{P} = \mathbf{K}[\mathbf{R} \quad \mathbf{t}]$

- Intrinsic parameters
  - Principal point coordinates
  - Focal length
  - Pixel magnification factors
  - *Skew (non-rectangular pixels)*
  - *Radial distortion*

$$\mathbf{K} = \begin{bmatrix} m_x & & \\ & m_y & \\ & & 1 \end{bmatrix} \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & & \beta_x \\ & \alpha_y & \beta_y \\ & & 1 \end{bmatrix}$$



Camera parameters $\mathbf{P} = \mathbf{K}[\mathbf{R} \quad \mathbf{t}]$

- Intrinsic parameters
  - Principal point coordinates
  - Focal length
  - Pixel magnification factors
  - *Skew (non-rectangular pixels)*
  - *Radial distortion*
- Extrinsic parameters
  - Rotation and translation relative to world coordinate system

$$\mathbf{P} = \mathbf{K}[\mathbf{R} \quad -\mathbf{R}\tilde{\mathbf{C}}]$$

coords. of camera center in world frame

- What is the projection of the camera center?

$$\mathbf{PC} = \mathbf{K}[\mathbf{R} \quad -\mathbf{R}\tilde{\mathbf{C}}]\begin{bmatrix}\tilde{\mathbf{C}} \\ 1\end{bmatrix} = 0$$

The camera center is the *null space* of the projection matrix!

Camera calibration

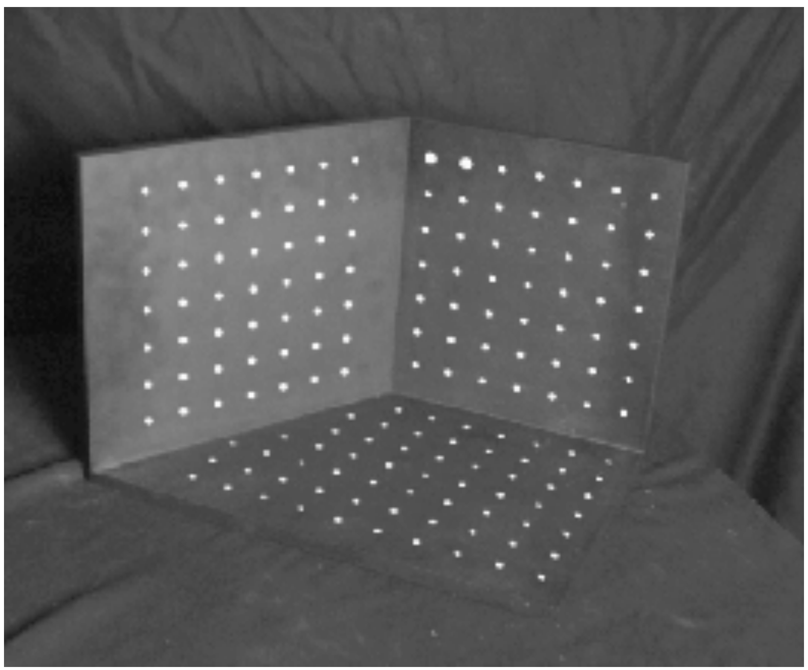
$$\lambda \mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

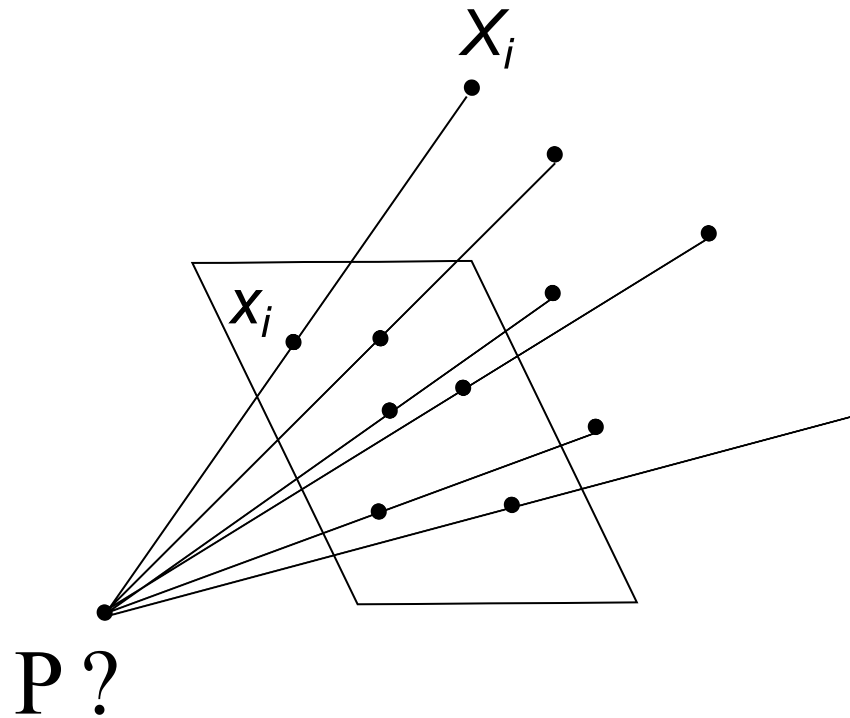
$$\begin{bmatrix} \lambda x \\ \lambda y \\ \lambda \end{bmatrix} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Source: D. Hoiem

Camera calibration

- Given *n* points with known 3D coordinates $\boldsymbol{X}_i$ and known image projections $\boldsymbol{x}_i$, estimate the camera parameters





Camera calibration: Linear method

$$\lambda \mathbf{x}_i = \mathbf{P}\mathbf{X}_i \qquad \mathbf{x}_i \times \mathbf{P}\mathbf{X}_i = 0 \qquad \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \times \begin{bmatrix} \mathbf{P}_1^T \mathbf{X}_i \\ \mathbf{P}_2^T \mathbf{X}_i \\ \mathbf{P}_3^T \mathbf{X}_i \end{bmatrix} = 0$$

$$\begin{bmatrix} 0 & -\mathbf{X}_i^T & y_i \mathbf{X}_i^T \\ \mathbf{X}_i^T & 0 & -x_i \mathbf{X}_i^T \\ -y_i \mathbf{X}_i^T & x_i \mathbf{X}_i^T & 0 \end{bmatrix} \begin{pmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{pmatrix} = 0$$

Two linearly independent equations

Camera calibration: Linear method

$$\begin{bmatrix} 0^T & \mathbf{X}_1^T & -y_1\mathbf{X}_1^T \\ \mathbf{X}_1^T & 0^T & -x_1\mathbf{X}_1^T \\ \cdots & \cdots & \cdots \\ 0^T & \mathbf{X}_n^T & -y_n\mathbf{X}_n^T \\ \mathbf{X}_n^T & 0^T & -x_n\mathbf{X}_n^T \end{bmatrix} \begin{pmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{pmatrix} = 0 \qquad \mathbf{Ap} = \mathbf{0}$$

- P has 11 degrees of freedom
- One 2D/3D correspondence gives us two linearly independent equations
  - 6 correspondences needed for a minimal solution
- Homogeneous least squares: find p minimizing $\|\mathrm{Ap}\|^2$
  - Solution given by eigenvector of $\mathrm{A}^\mathrm{T}\mathrm{A}$ with smallest eigenvalue

Camera calibration: Linear method

$$\begin{bmatrix} 0^T & \mathbf{X}_1^T & -y_1\mathbf{X}_1^T \\ \mathbf{X}_1^T & 0^T & -x_1\mathbf{X}_1^T \\ \cdots & \cdots & \cdots \\ 0^T & \mathbf{X}_n^T & -y_n\mathbf{X}_n^T \\ \mathbf{X}_n^T & 0^T & -x_n\mathbf{X}_n^T \end{bmatrix} \begin{pmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{pmatrix} = 0 \qquad \mathbf{Ap} = \mathbf{0}$$

- Note: for coplanar points that satisfy $\mathbf{\Pi}^T\mathbf{X}=0$, we will get degenerate solutions $(\mathbf{\Pi},\mathbf{0},\mathbf{0})$, $(\mathbf{0},\mathbf{\Pi},\mathbf{0})$, or $(\mathbf{0},\mathbf{0},\mathbf{\Pi})$

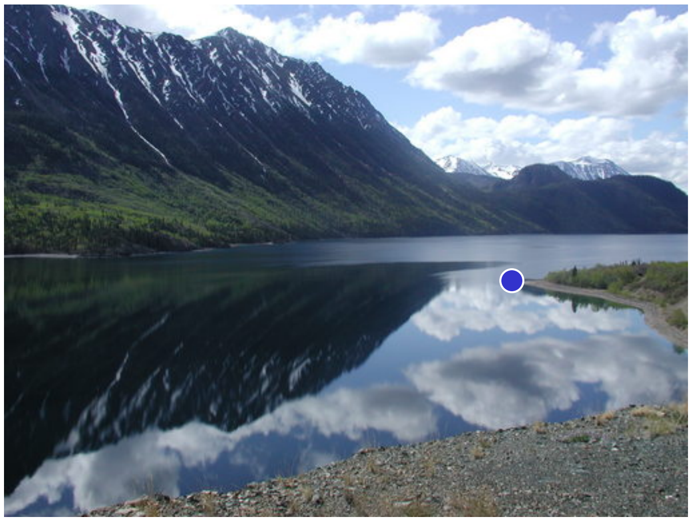
Camera calibration: Linear vs. nonlinear

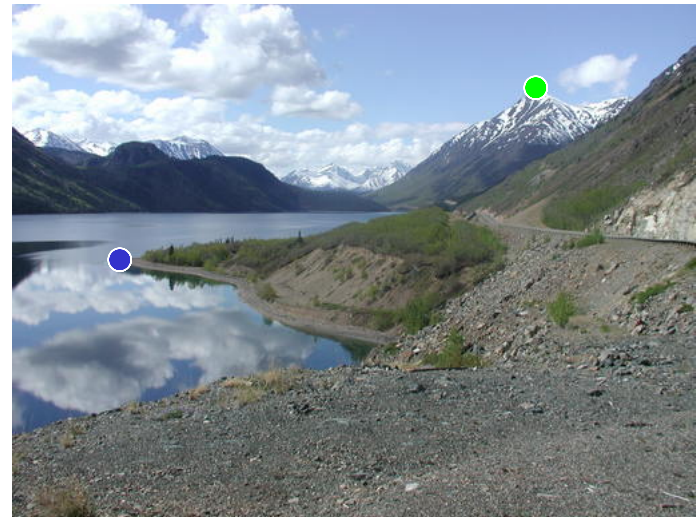
- Linear calibration is easy to formulate and solve, but it doesn't directly tell us the camera parameters

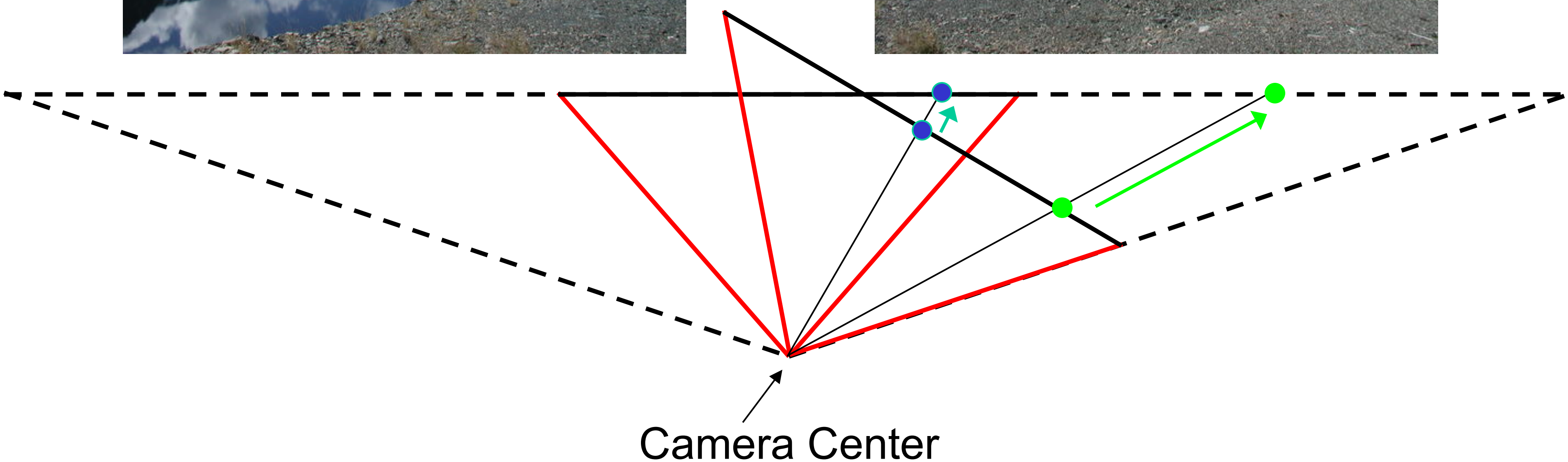
$$\begin{bmatrix} \lambda x \\ \lambda y \\ \lambda \end{bmatrix} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad \text{vs.} \quad \mathbf{x} = \mathbf{K}\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}\mathbf{X}$$

- In practice, non-linear methods are preferred
  - Write down objective function in terms of intrinsic and extrinsic parameters
  - Define error as sum of squared distances between measured 2D points and estimated projections of 3D points
  - Minimize error using Newton's method or other non-linear optimization
  - Can model radial distortion and impose constraints such as known focal length and orthogonality

Homography Example







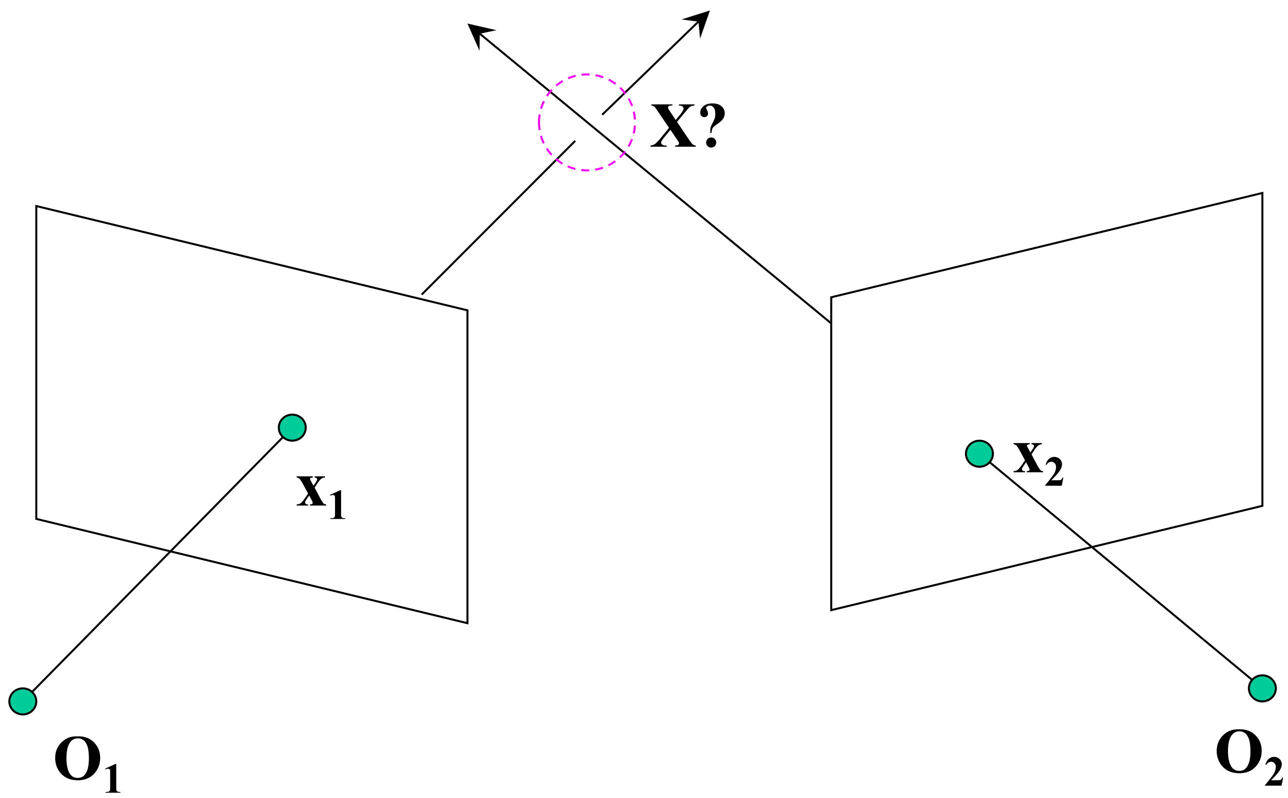
Slide from A. Efros, S. Seitz, D. Hoiem

A taste of multi-view geometry: Triangulation

- Given projections of a 3D point in two or more images (with known camera matrices), find the coordinates of the point

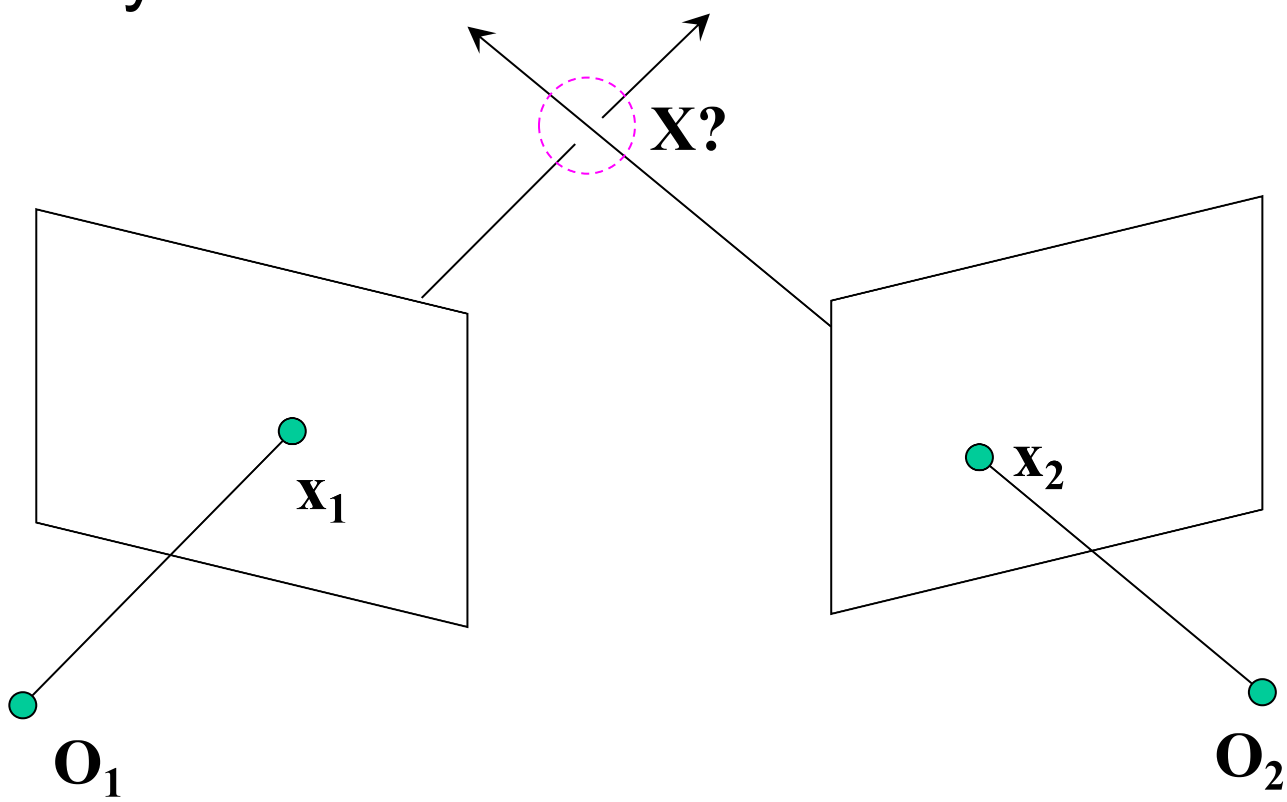
Triangulation

- Given projections of a 3D point in two or more images (with known camera matrices), find the coordinates of the point



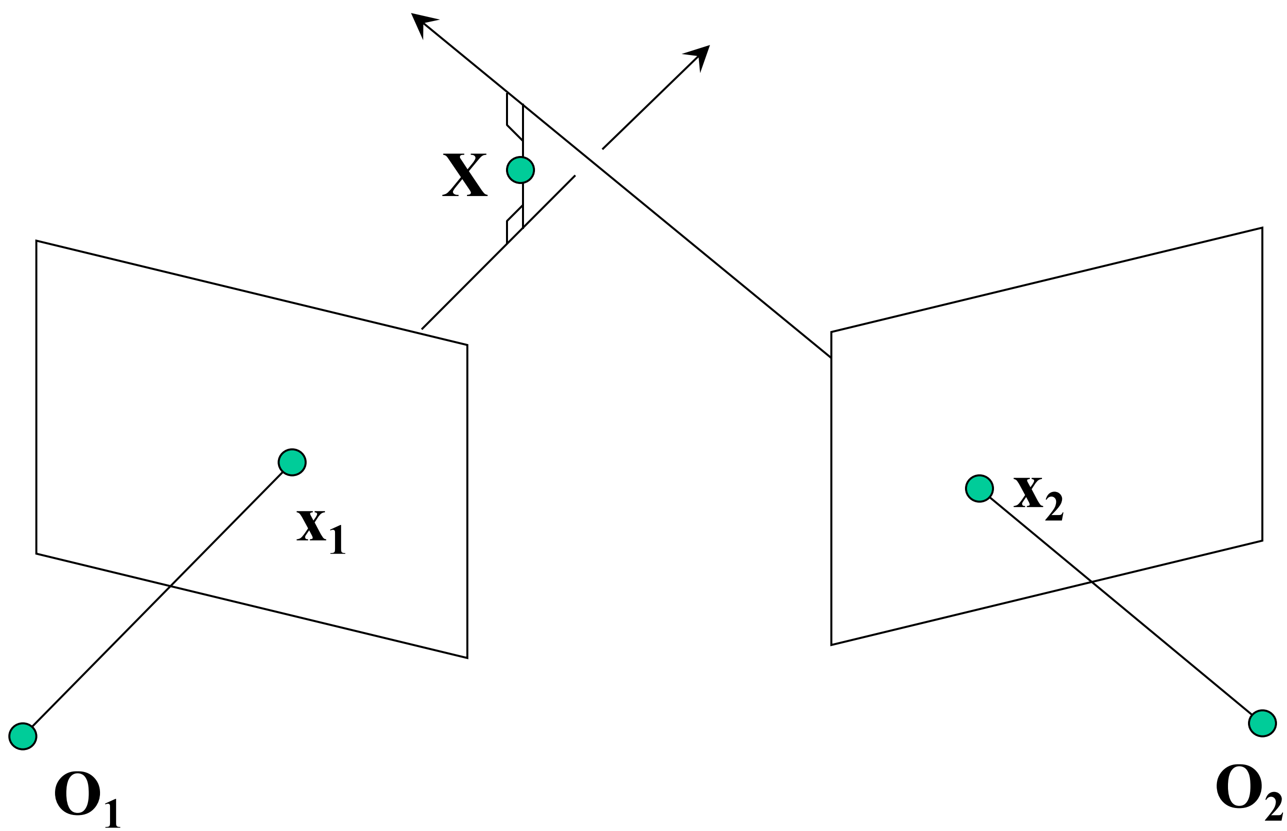
Triangulation

- We want to intersect the two visual rays corresponding to $\mathbf{x_1}$ and $\mathbf{x_2}$, but because of noise and numerical errors, they don't meet exactly



Triangulation: Geometric approach

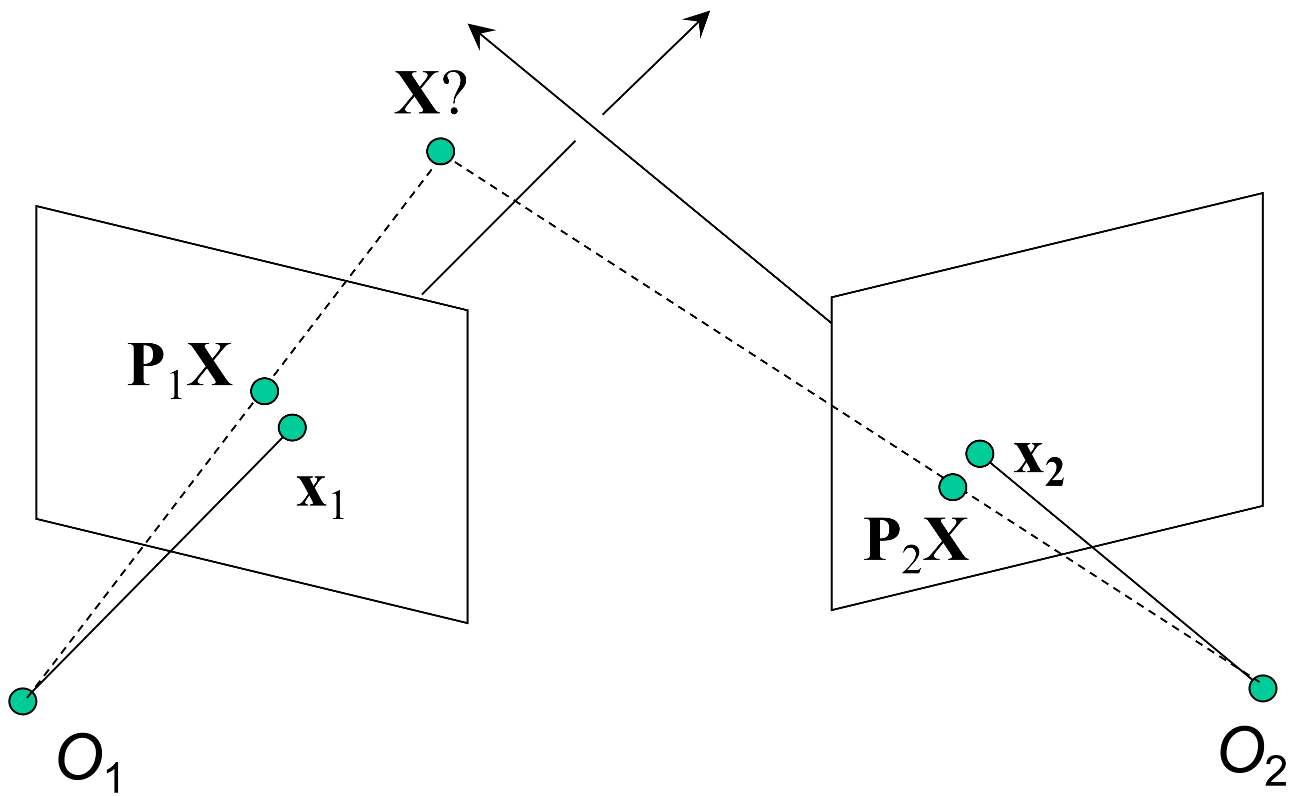
- Find shortest segment connecting the two viewing rays and let **X** be the midpoint of that segment



Triangulation: Nonlinear approach

Find X that minimizes

$$d^2(\mathbf{x_1}, \mathbf{P_1 X}) + d^2(\mathbf{x_2}, \mathbf{P_2 X})$$



Triangulation: Linear approach

$$\lambda_1 \mathbf{x_1} = \mathbf{P_1 X} \qquad \mathbf{x_1} \times \mathbf{P_1 X} = \mathbf{0} \qquad [\mathbf{x_{1\times}}]\mathbf{P_1 X} = \mathbf{0}$$

$$\lambda_2 \mathbf{x_2} = \mathbf{P_2 X} \qquad \mathbf{x_2} \times \mathbf{P_2 X} = \mathbf{0} \qquad [\mathbf{x_{2\times}}]\mathbf{P_2 X} = \mathbf{0}$$

Cross product as matrix multiplication:

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

Triangulation: Linear approach

$$\lambda_1 \mathbf{x_1} = \mathbf{P_1 X} \qquad \mathbf{x_1} \times \mathbf{P_1 X} = \mathbf{0} \qquad [\mathbf{x}_{\mathbf{1}\times}]\mathbf{P_1 X} = \mathbf{0}$$

$$\lambda_2 \mathbf{x_2} = \mathbf{P_2 X} \qquad \mathbf{x_2} \times \mathbf{P_2 X} = \mathbf{0} \qquad [\mathbf{x}_{\mathbf{2}\times}]\mathbf{P_2 X} = \mathbf{0}$$

Two independent equations each in terms of three unknown entries of **X**

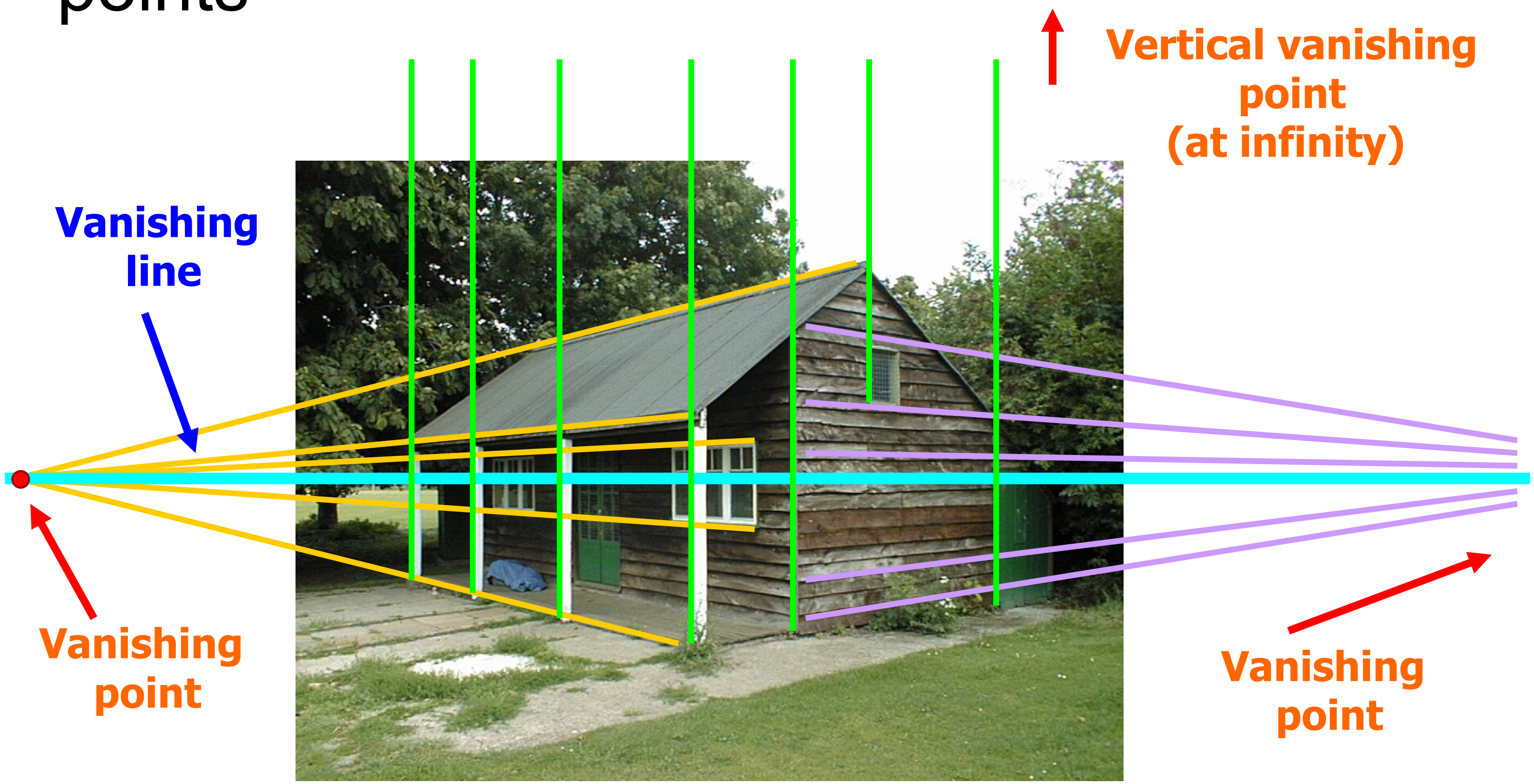
Camera calibration revisited

- What if world coordinates of reference 3D points are not known?
- We can use scene features such as vanishing points

Slide from Efros, Photo from Criminisi

Camera calibration revisited

- What if world coordinates of reference 3D points are not known?
- We can use scene features such as vanishing points



Slide from Efros, Photo from Criminisi

Recall: Homogenous Coordinates

Points

Points at infinity

Lines

Lines passing through 2 points

Intersection of 2 lines

Intersection of 2 parallel lines?

Recall: Homogenous Coordinates

Points

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

Points at infinity

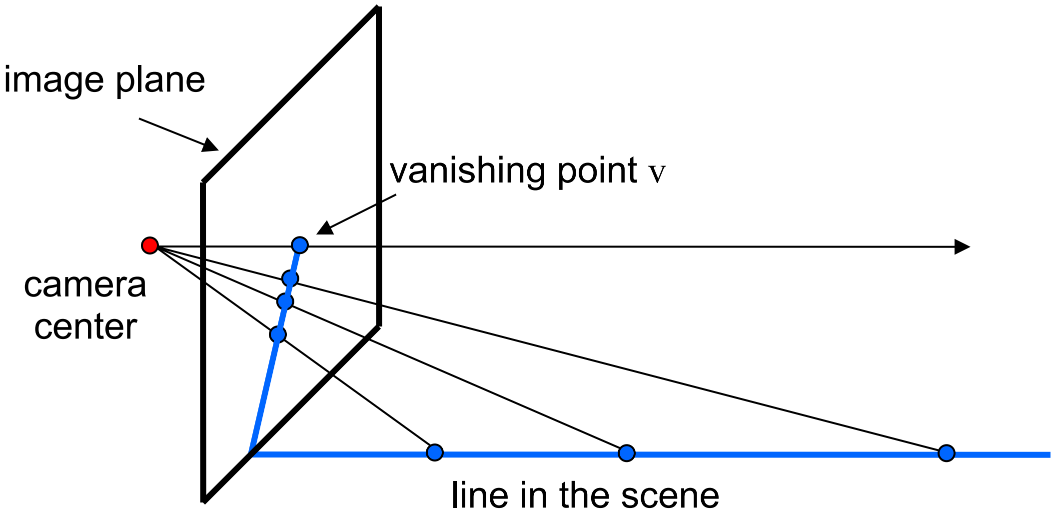
Lines

Lines passing through 2 points

Intersection of 2 lines

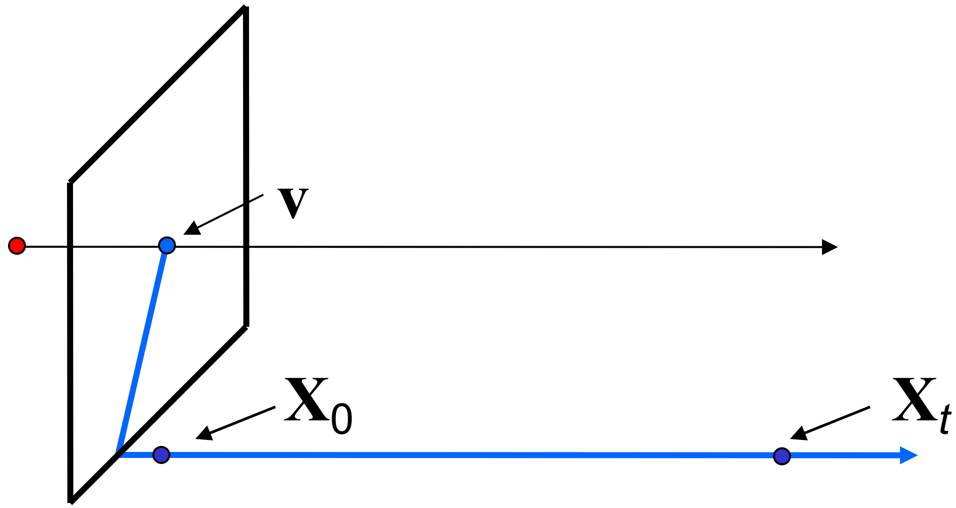
Intersection of 2 parallel lines?

Recall: Vanishing points



- All lines having the same direction share the same vanishing point

Computing vanishing points

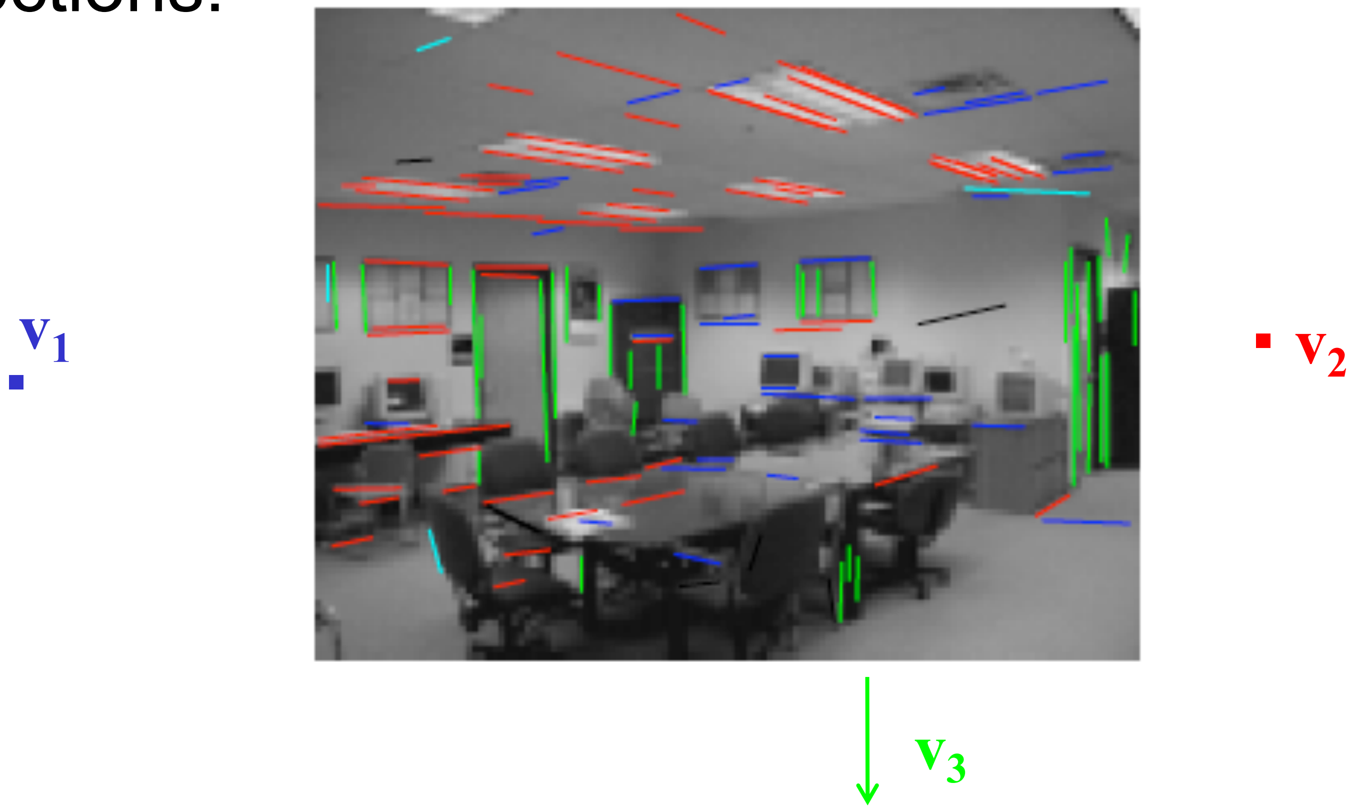


$$\mathbf{X}_t = \begin{bmatrix} x_0 + td_1 \\ y_0 + td_2 \\ z_0 + td_3 \\ 1 \end{bmatrix} = \begin{bmatrix} x_0 / t + d_1 \\ y_0 / t + d_2 \\ z_0 / t + d_3 \\ 1/t \end{bmatrix} \qquad \mathbf{X}_\infty = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ 0 \end{bmatrix}$$

- $\mathbf{X}_\infty$ is a *point at infinity,* $\mathbf{v}$ is its projection: $\mathbf{v} = \mathbf{P}\mathbf{X}_\infty$
- The vanishing point depends only on *line direction*
- All lines having direction $\mathbf{d}$ intersect at $\mathbf{X}_\infty$

Calibration from vanishing points

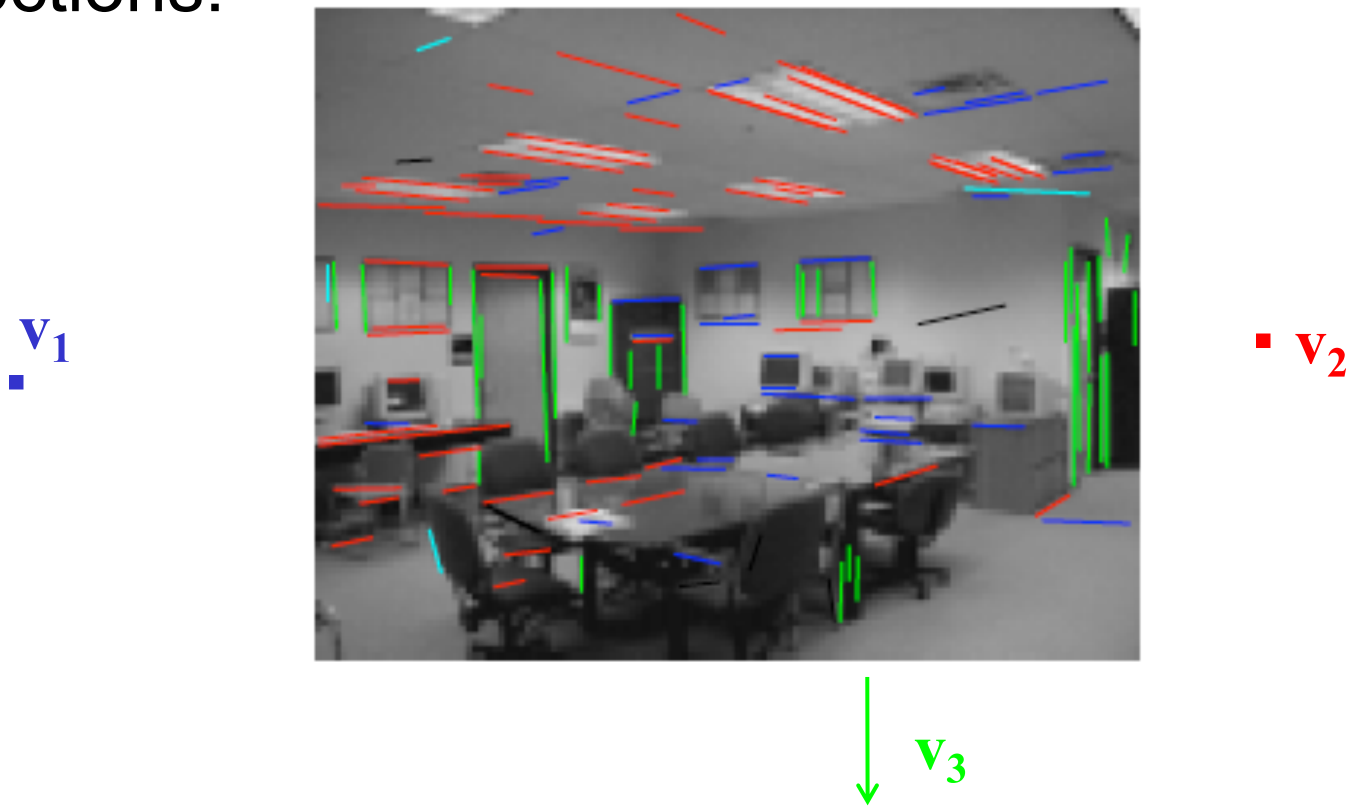
- Consider a scene with three orthogonal vanishing directions:



- Note: $\mathbf{v}_1$, $\mathbf{v}_2$ are *finite* vanishing points and $\mathbf{v}_3$ is an *infinite* vanishing point

Calibration from vanishing points

- Consider a scene with three orthogonal vanishing directions:



- We can align the world coordinate system with these directions

Calibration from vanishing points

$$\mathbf{P} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} = [\mathbf{p}_1 \quad \mathbf{p}_2 \quad \mathbf{p}_3 \quad \mathbf{p}_4]$$

- $\mathbf{p_1} = \mathbf{P}(1,0,0,0)^T$ – the vanishing point in the x direction
- Similarly, $\mathbf{p_2}$ and $\mathbf{p_3}$ are the vanishing points in the y and z directions
- $\mathbf{p_4} = \mathbf{P}(0,0,0,1)^T$ – projection of the origin of the world coordinate system
- Problem: we can only know the four columns up to independent scale factors, additional constraints needed to solve for them

Calibration from vanishing points

- Let us align the world coordinate system with three orthogonal vanishing directions in the scene:

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\lambda_i \mathbf{v}_i = \mathbf{K}\left[\mathbf{R} \mid \mathbf{t}\right] \begin{bmatrix} \mathbf{e}_i \\ 0 \end{bmatrix}$$

Calibration from vanishing points

- Let us align the world coordinate system with three orthogonal vanishing directions in the scene:

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\lambda_i \mathbf{v}_i = \mathbf{K}\mathbf{R}\mathbf{e}_i$$

$$\mathbf{e}_i = \lambda_i \mathbf{R}^T \mathbf{K}^{-1} \mathbf{v}_i$$

- Orthogonality constraint: $\mathbf{e}_i^T \mathbf{e}_j = 0$

$$\underbrace{\mathbf{v}_i^T \mathbf{K}^{-T} \mathbf{R}}_{\mathbf{e}_i^T} \underbrace{\mathbf{R}^T \mathbf{K}^{-1} \mathbf{v}_j}_{\mathbf{e}_j} = 0$$

Calibration from vanishing points

- Let us align the world coordinate system with three orthogonal vanishing directions in the scene:

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\lambda_i \mathbf{v}_i = \mathbf{K}\mathbf{R}\mathbf{e}_i$$

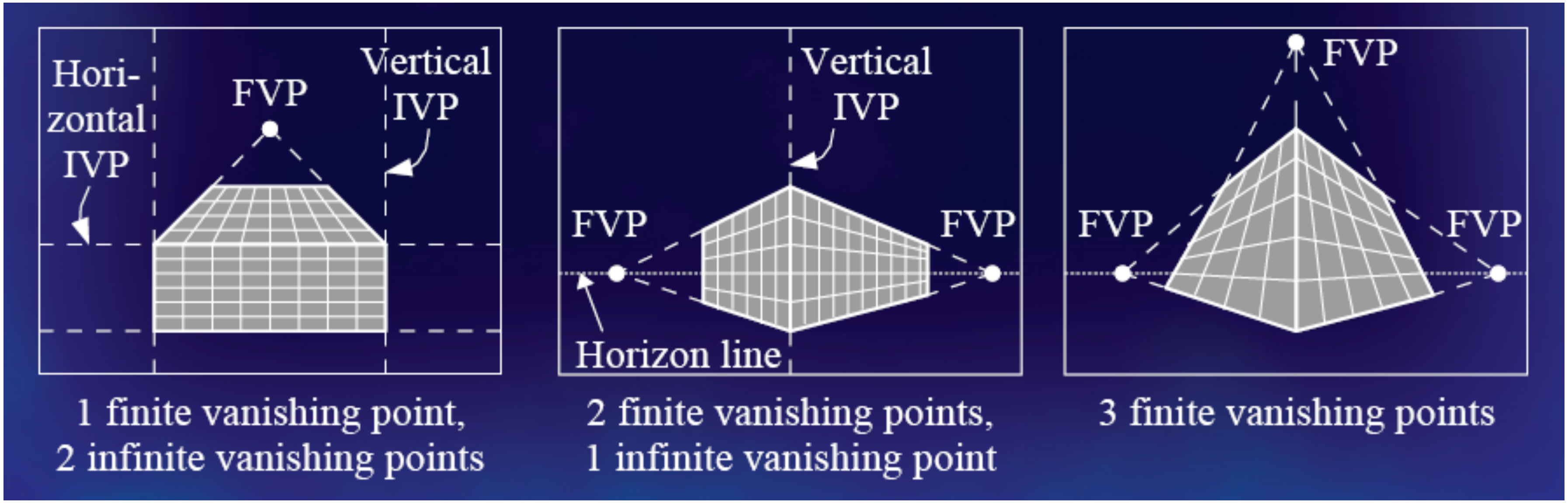
$$\mathbf{e}_i = \lambda_i \mathbf{R}^T \mathbf{K}^{-1} \mathbf{v}_i$$

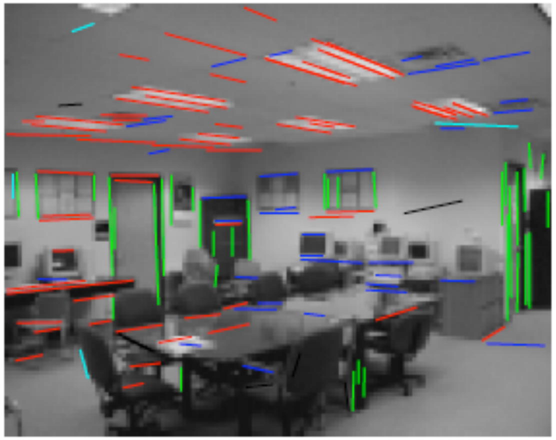
- Orthogonality constraint: $\mathbf{e}_i^T \mathbf{e}_j = 0$

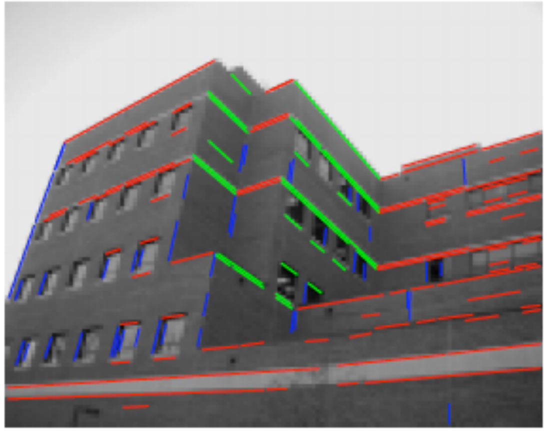
$$\mathbf{v}_i^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{v}_j = 0$$

- Rotation disappears, each pair of vanishing points gives constraint on focal length and principal point

Calibration from vanishing points







Cannot recover focal length, principal point is the third vanishing point

Can solve for focal length, principal point

Rotation from vanishing points

- Constraints on vanishing points: $\lambda_i \mathbf{v}_i = \mathbf{K}\mathbf{R}\mathbf{e}_i$
- After solving for the calibration matrix:

$$\lambda_i \mathbf{K}^{-1}\mathbf{v}_i = \mathbf{R}\mathbf{e}_i$$

- Notice: $\mathbf{R}\mathbf{e}_1 = [\; \mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{r}_3 \;] \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \mathbf{r}_1$

- Thus, $\mathbf{r}_i = \lambda_i \mathbf{K}^{-1}\mathbf{v}_i$

- Get λ_i by using the constraint $\|\mathbf{r}_i\|^2 = 1$.

Calibration from vanishing points: Summary

- Solve for K (focal length, principal point) using three orthogonal vanishing points
- Get rotation directly from vanishing points once calibration matrix is known

- Advantages
  - No need for calibration chart, 2D-3D correspondences
  - Could be completely automatic
- Disadvantages
  - Only applies to certain kinds of scenes
  - Inaccuracies in computation of vanishing points
  - Problems due to infinite vanishing points

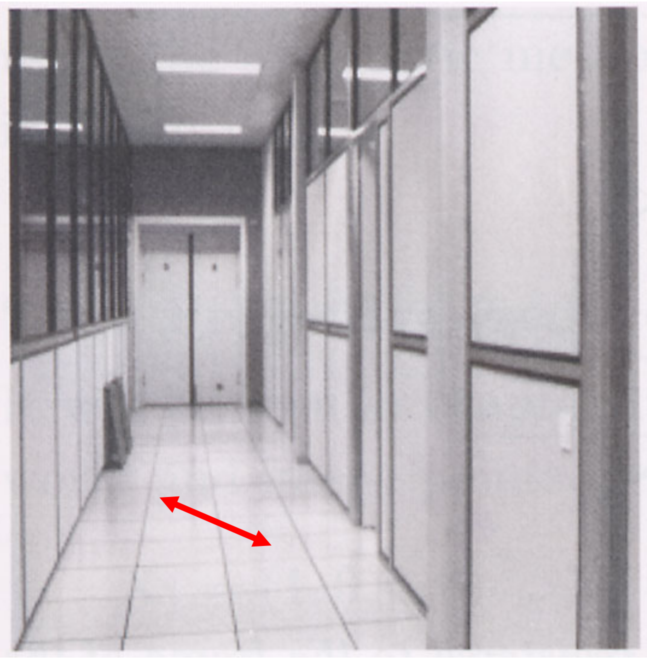
Example

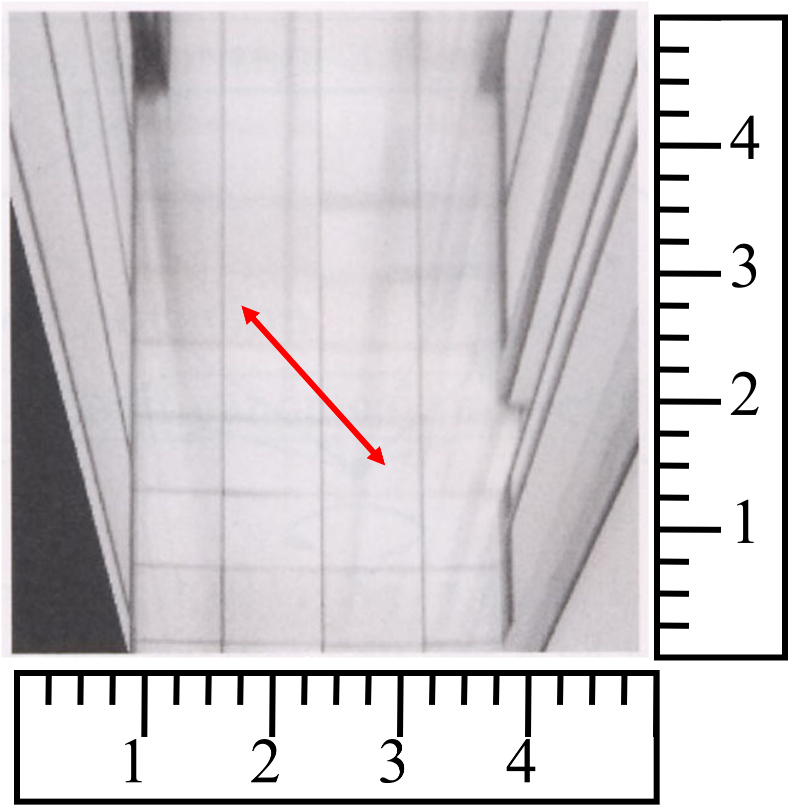
- Are the heights of the two groups of people consistent with one another?
  - Measure heights using Christ as reference

Piero della Francesca*, Flagellation,* ca. 1455

A. Criminisi, M. Kemp, and A. Zisserman,Bringing Pictorial Space to Life: computer techniques for the analysis of paintings,
Proc. Computers and the History of Art, 2002

Measurements on planes

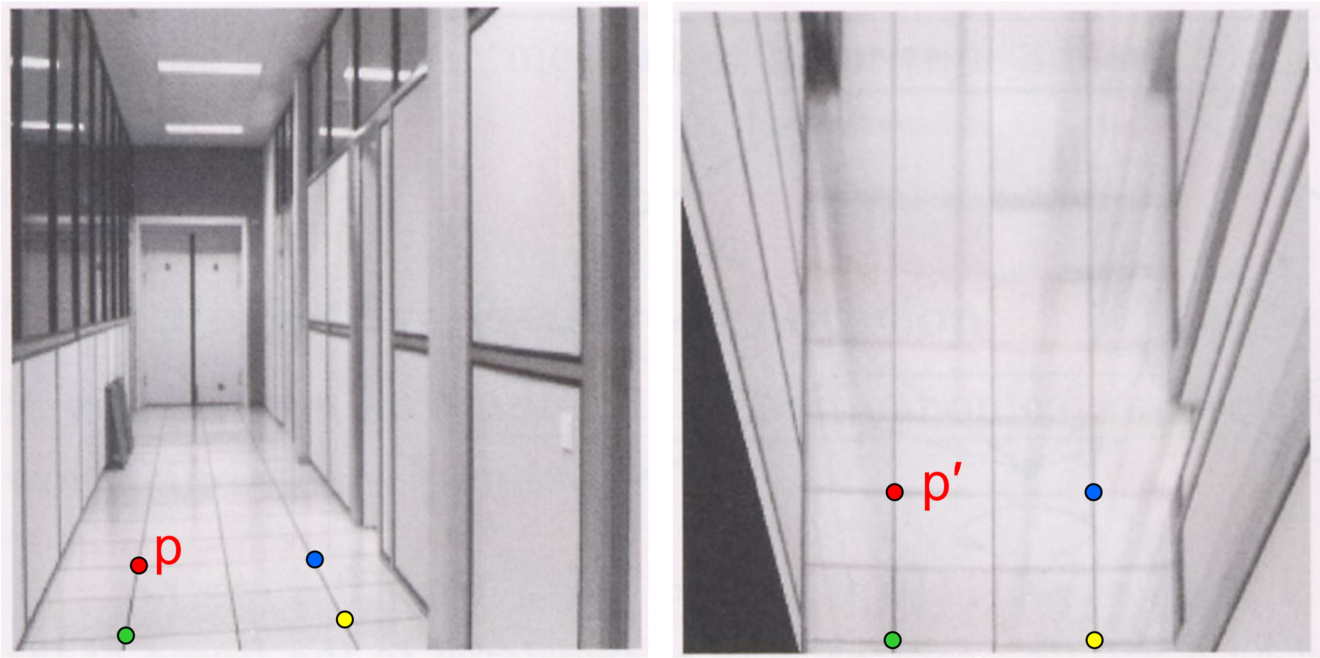




Approach: unwarp then measure

What kind of warp is this?

Image rectification



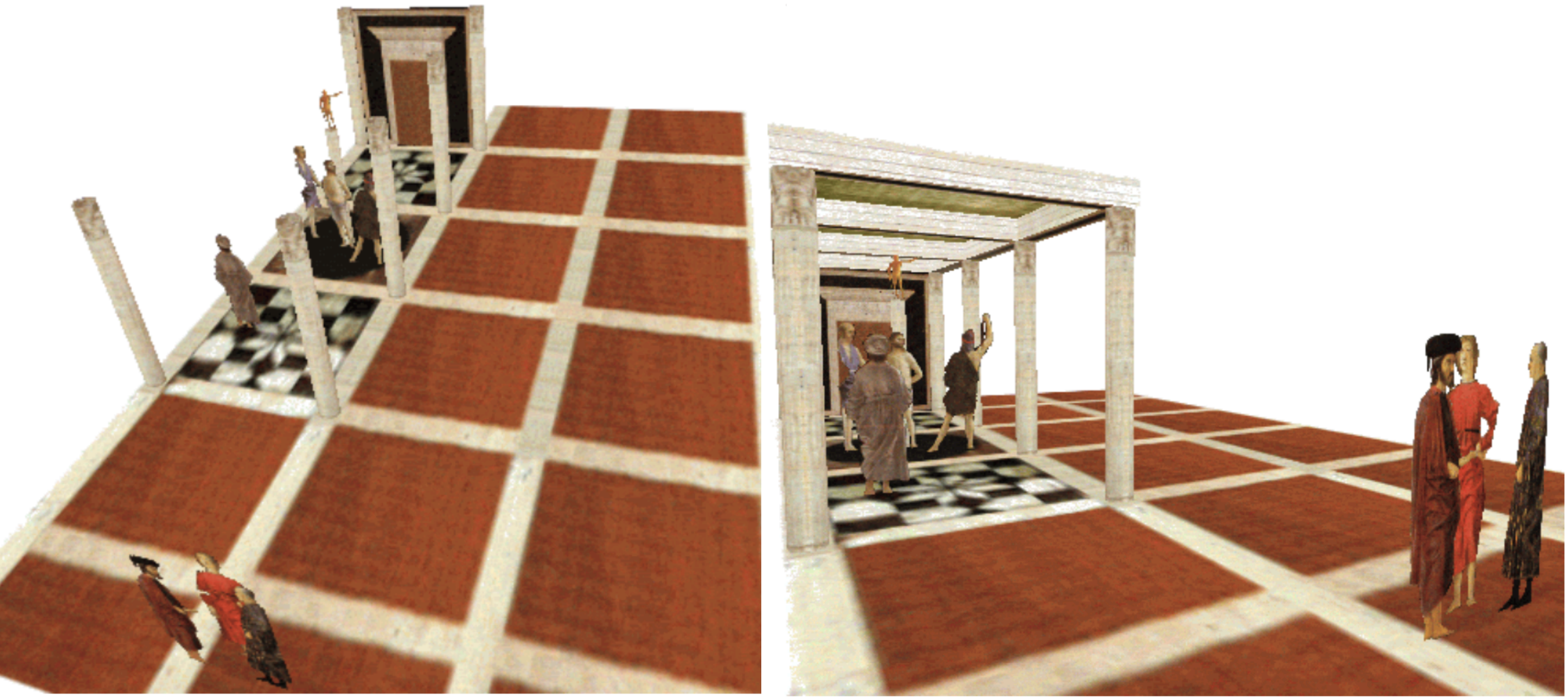
To unwarp (rectify) an image

- solve for homography H given p and p′
  - how many points are necessary to solve for H?

Image rectification: example

Piero della Francesca*, Flagellation,* ca. 1455

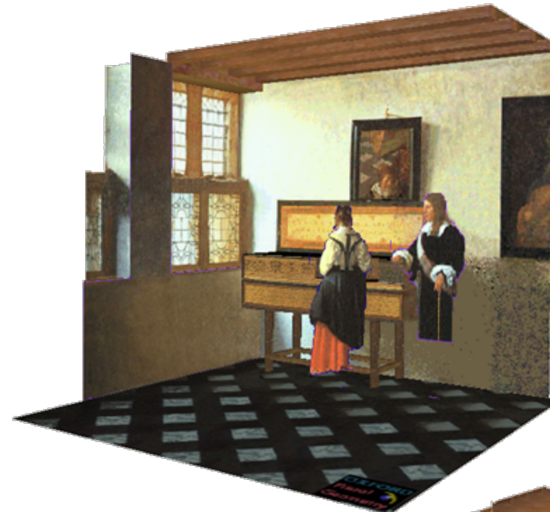
Application: 3D modeling from a single image



A. Criminisi, M. Kemp, and A. Zisserman, [Bringing Pictorial Space to Life: computer techniques for the analysis of paintings](),
Proc. Computers and the History of Art, 2002

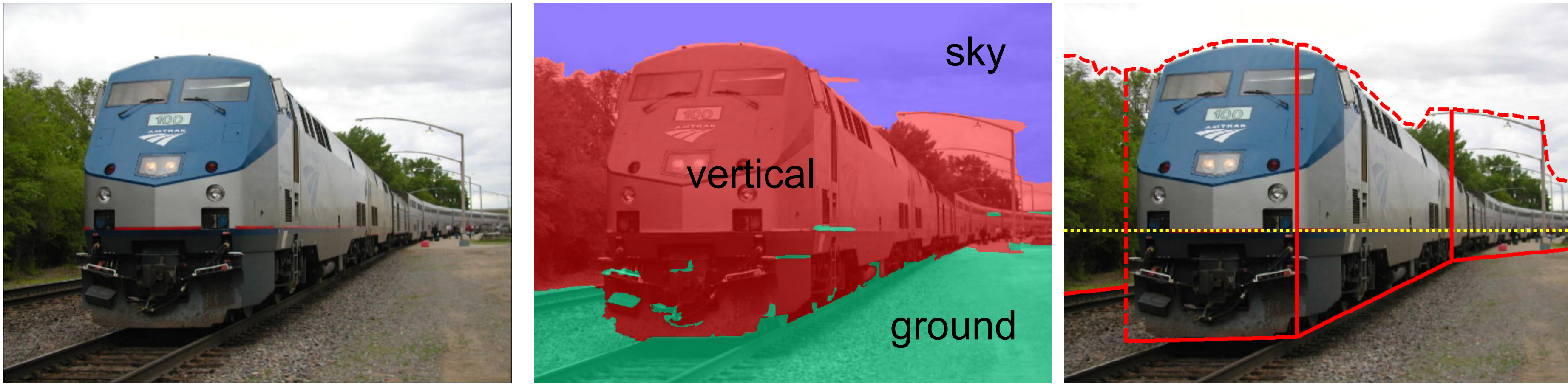
Application: 3D modeling from a single image

J. Vermeer, Music Lesson, 1662



A. Criminisi, M. Kemp, and A. Zisserman,[Bringing Pictorial Space to Life: computer techniques for the analysis of paintings](),
Proc. Computers and the History of Art, 2002

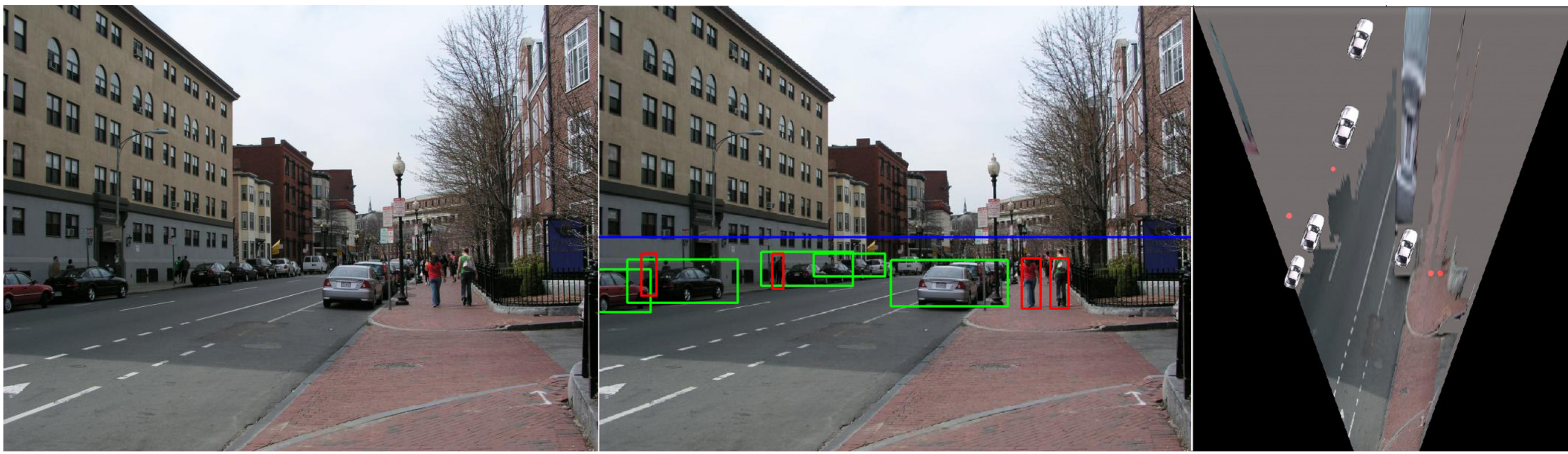
Application: Fully automatic modeling



D. Hoiem, A.A. Efros, and M. Hebert, Automatic Photo Pop-up, SIGGRAPH 2005.

http://dhoiem.cs.illinois.edu/projects/popup/popup_movie_450_250.mp4

Application: Object detection



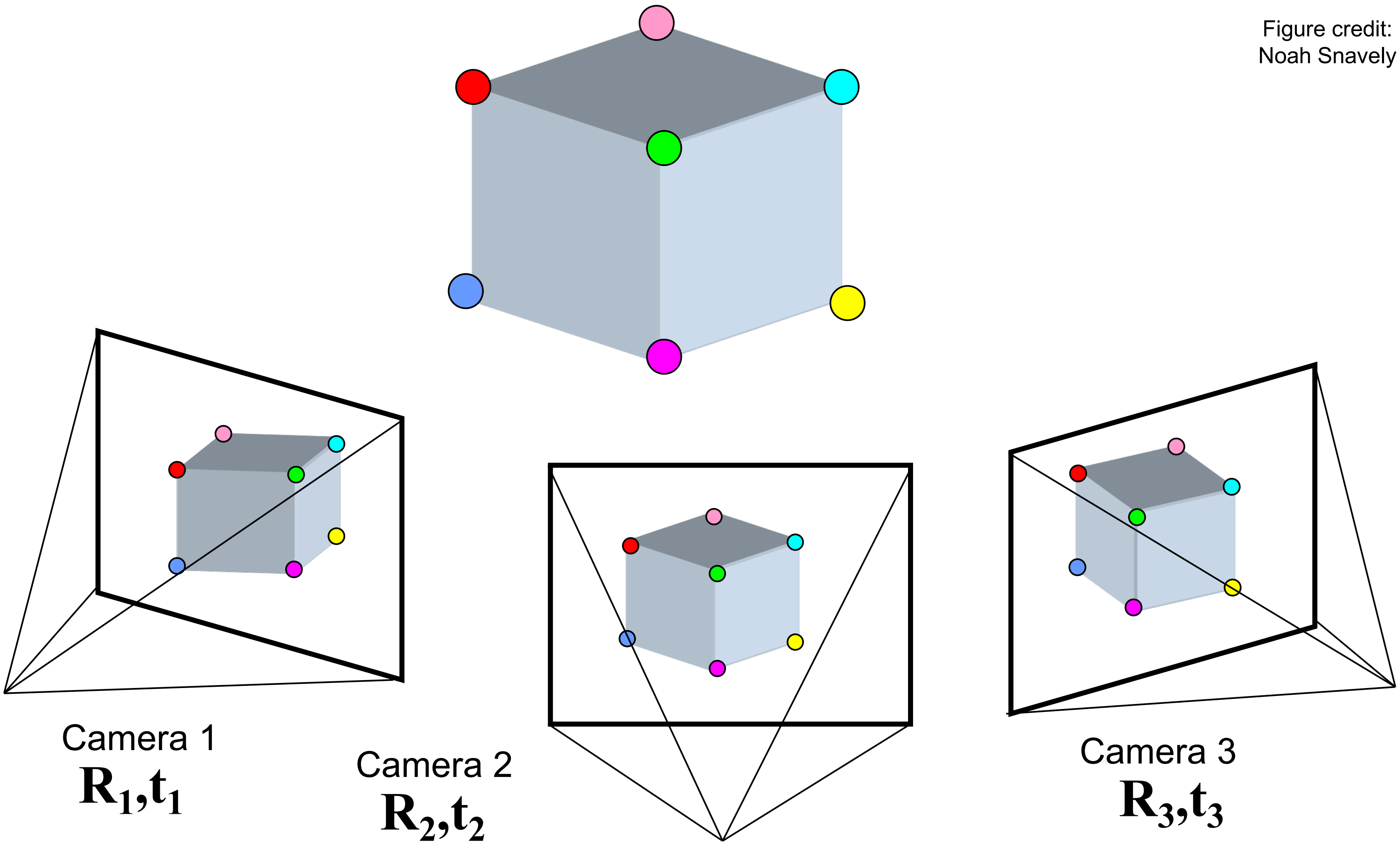
D. Hoiem, A.A. Efros, and M. Hebert, Putting Objects in Perspective, CVPR 2006

Application: Image editing

Inserting synthetic objects into images:
http://vimeo.com/28962540

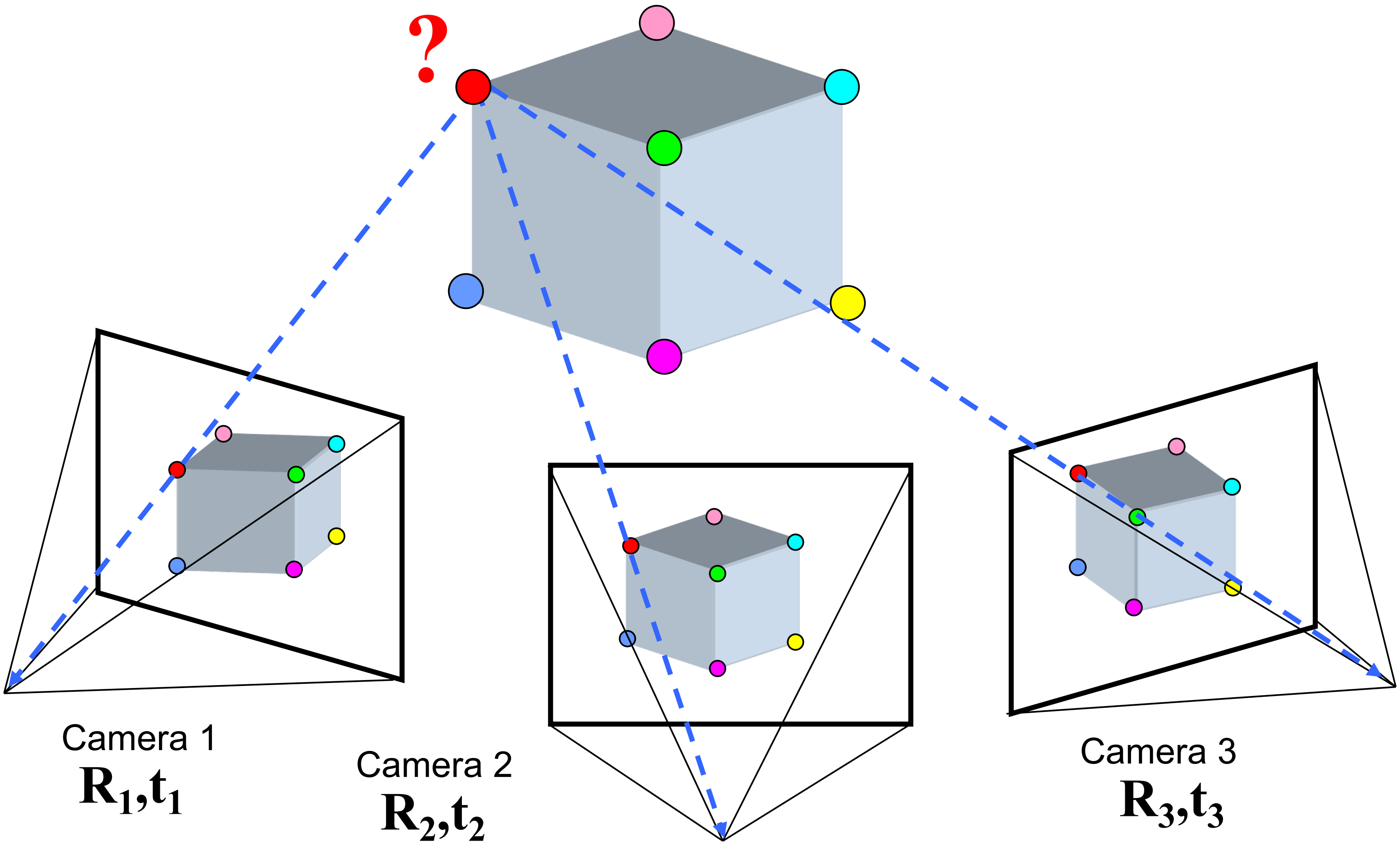
K. Karsch and V. Hedau and D. Forsyth and D. Hoiem, Rendering Synthetic Objects into Legacy Photographs, *SIGGRAPH Asia* 2011

Preview: Structure from motion



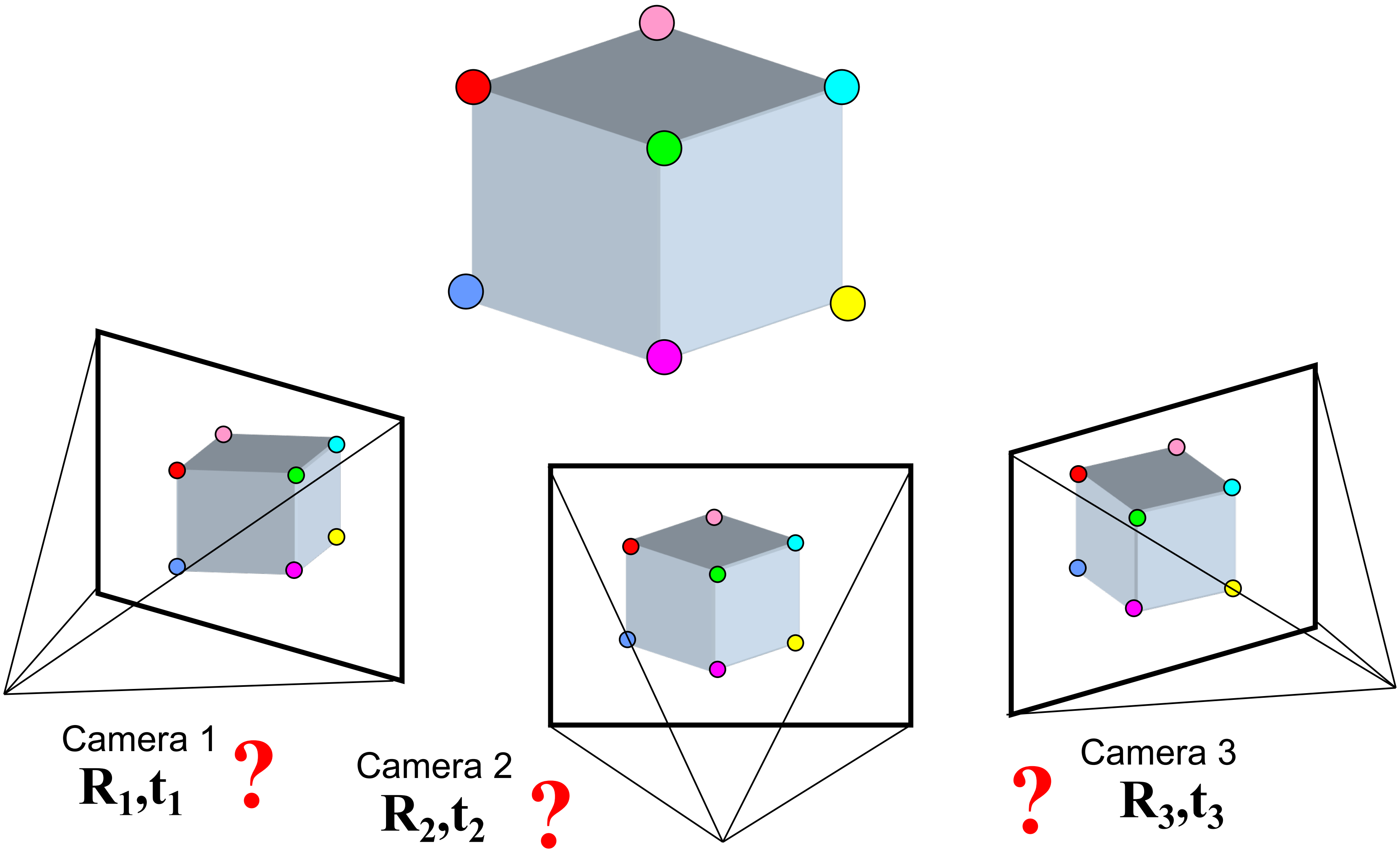
- Given 2D point correspondences between multiple images, compute the camera parameters and the 3D points

Preview: Structure from motion



- **Structure:** Given *known cameras* and projections of the same 3D point in two or more images, compute the 3D coordinates of that point
  - Triangulation!

Preview: Structure from motion



- **Motion:** Given a set of *known* 3D points seen by a camera, compute the camera parameters
  - Calibration!

Useful reference