Projection of the 3D world in 2D images

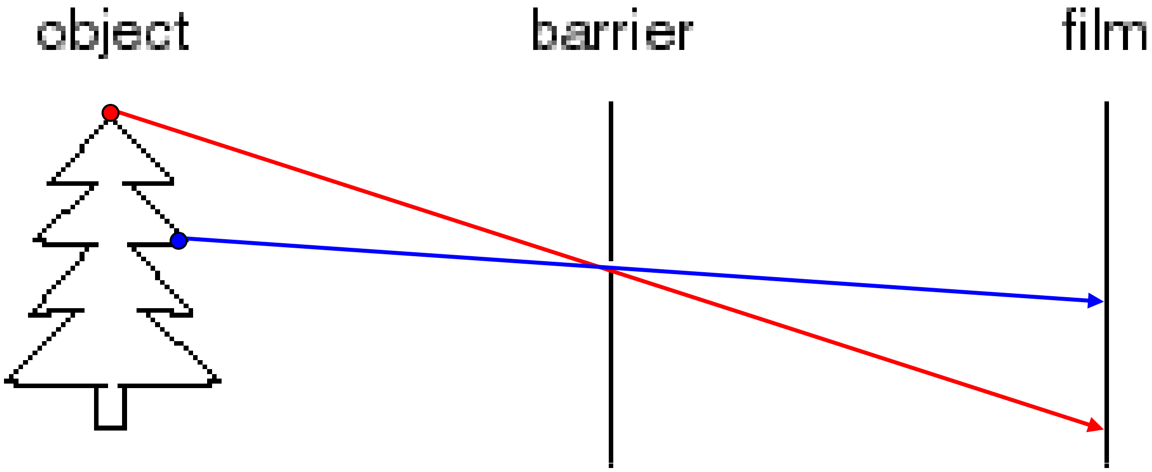
Video source: Camera Obscura & World of Illusions

Many slides adapted from S. Seitz, L. Lazebnik, B. Hariharan, N. Snavley

Outline

- Pinhole camera model
- Modeling pinhole cameras
- Camera calibration
- Triangulation
- *Epipolar geometry*
- Structure from Motion (SfM)
- SfM pipeline
- Depth Sensors

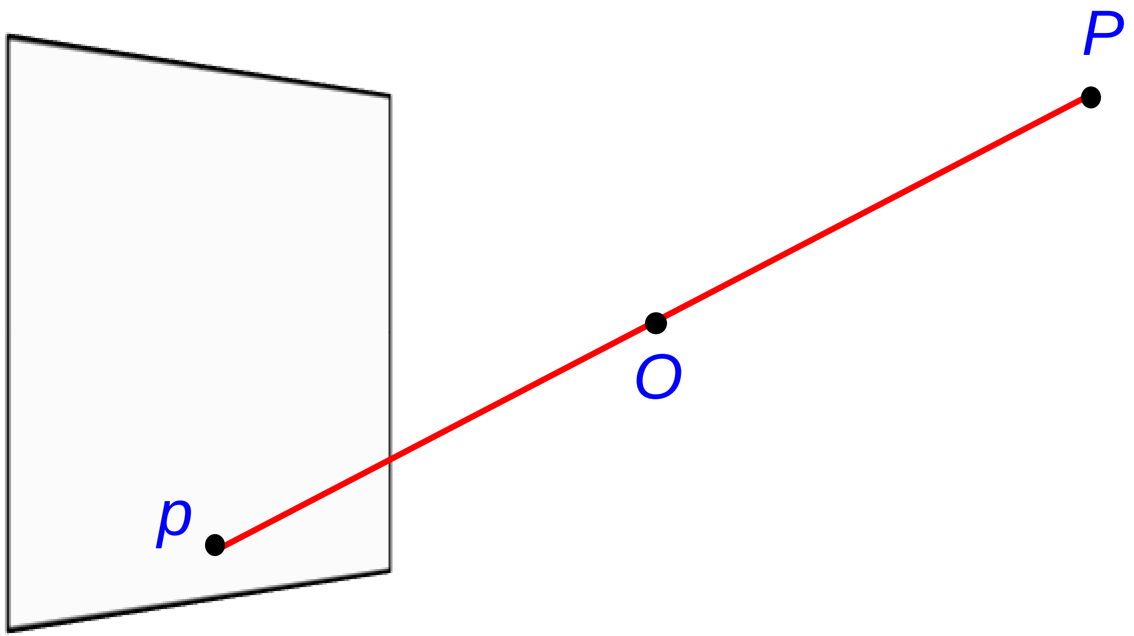
Pinhole camera



- Captures **pencil of rays** – all rays through a single point: **aperture, center of projection, optical center, focal point, camera center**
- The image is formed on the **image plane**

Slide by Steve Seitz

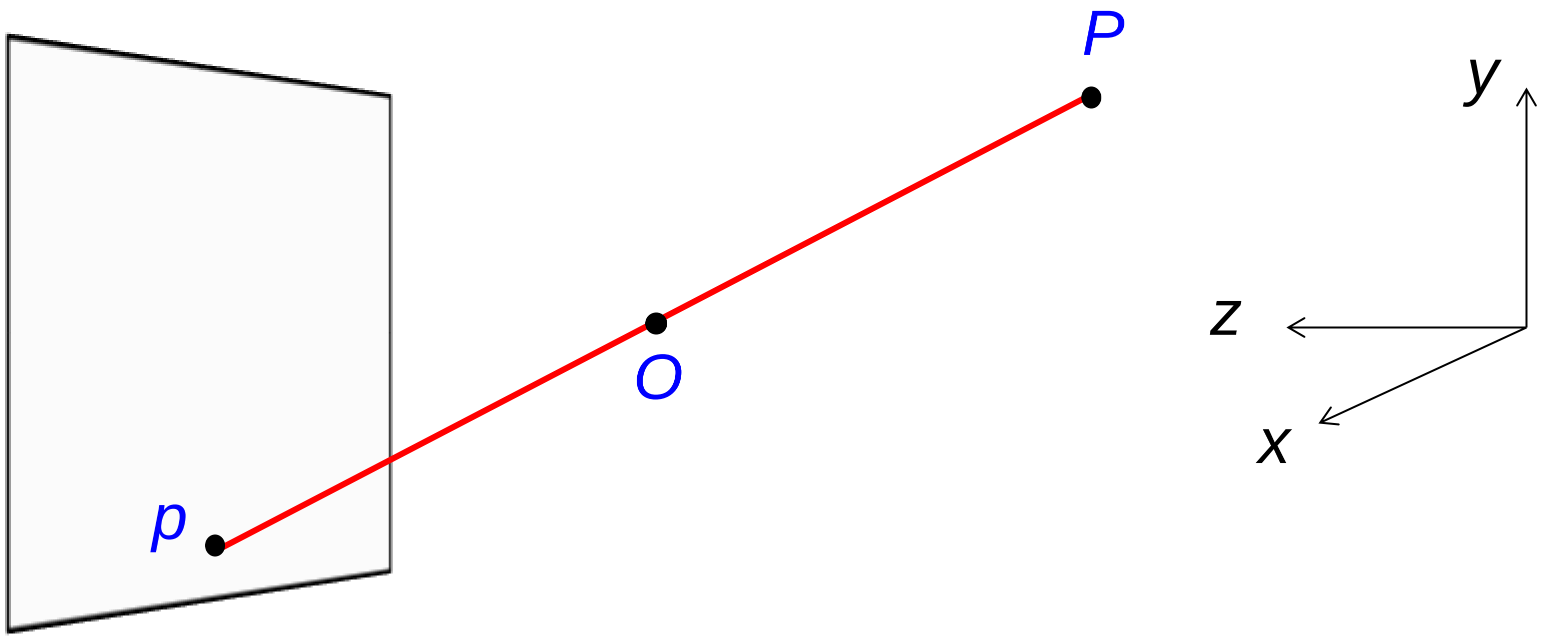
Pinhole projection model



- To compute the projection *p* of a scene point *P*, form the **visual ray** connecting *P* to the **camera center** *O* and find where it intersects the **image plane**

Slide by L. Lazebnik.

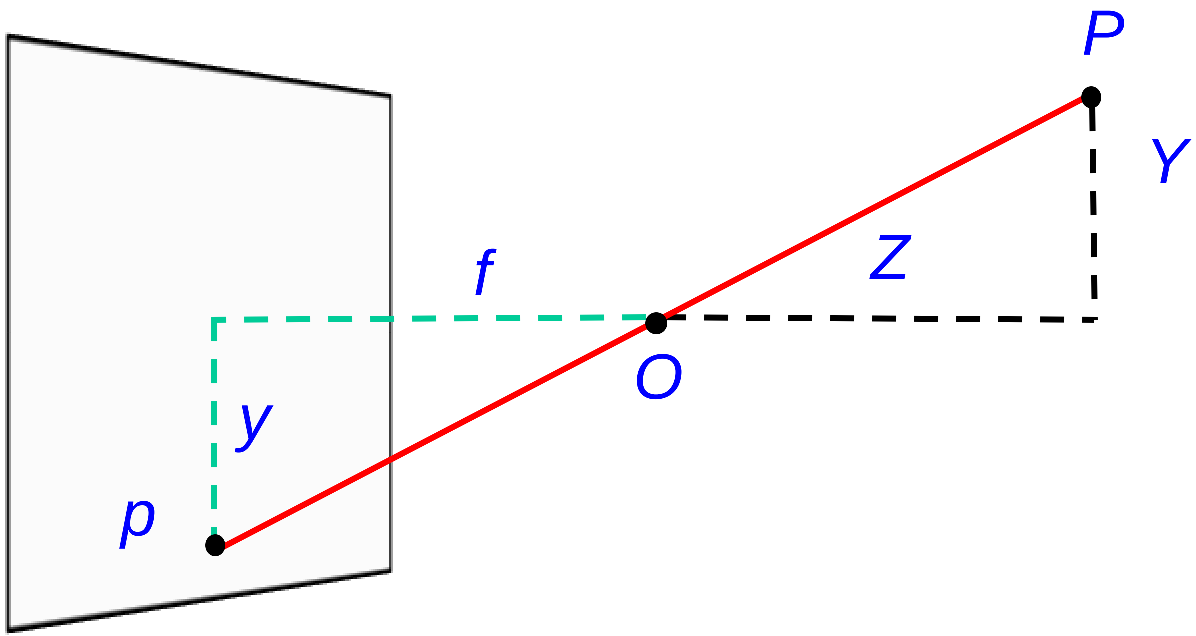
Pinhole projection model



- The coordinate system
  - The optical center (***O***) is at the origin
  - The image plane is parallel to xy-plane or perpendicular to the z-axis, which is the *optical axis*

Slide by L. Lazebnik.

Pinhole projection model



Projection equations

- Derived using similar triangles

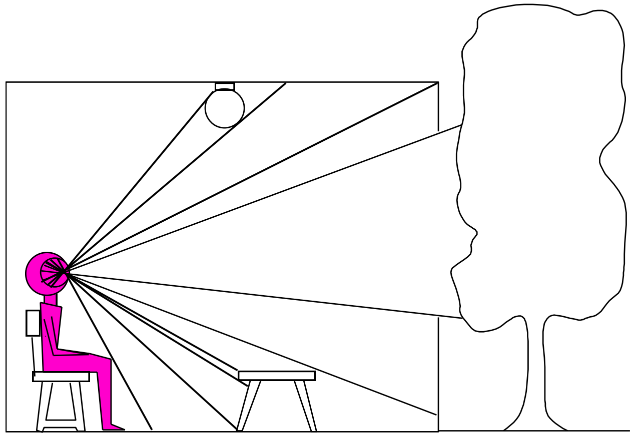
$$(X, Y, Z) \rightarrow \left(\frac{fX}{Z}, \frac{fY}{Z}\right) = (x, y)$$

Note: instead of dealing with an image that is upside down, most of the time we will pretend that the image plane is in front of the camera center.

Slide by L. Lazebnik.

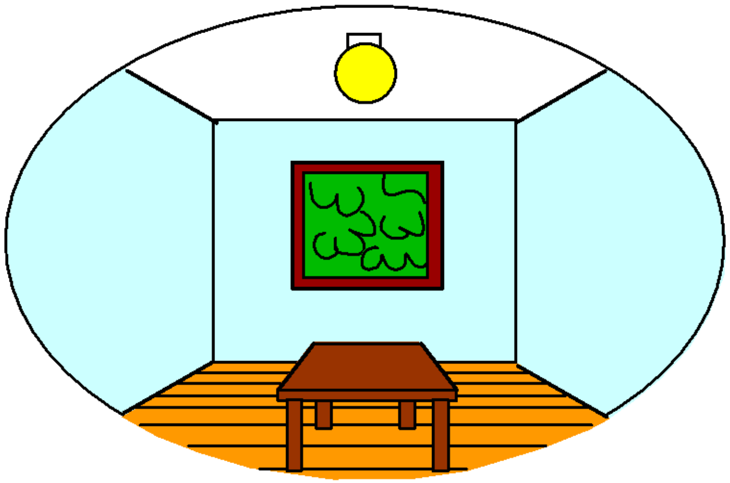
Dimensionality reduction: from 3D to 2D

3D world



Point of observation

2D image

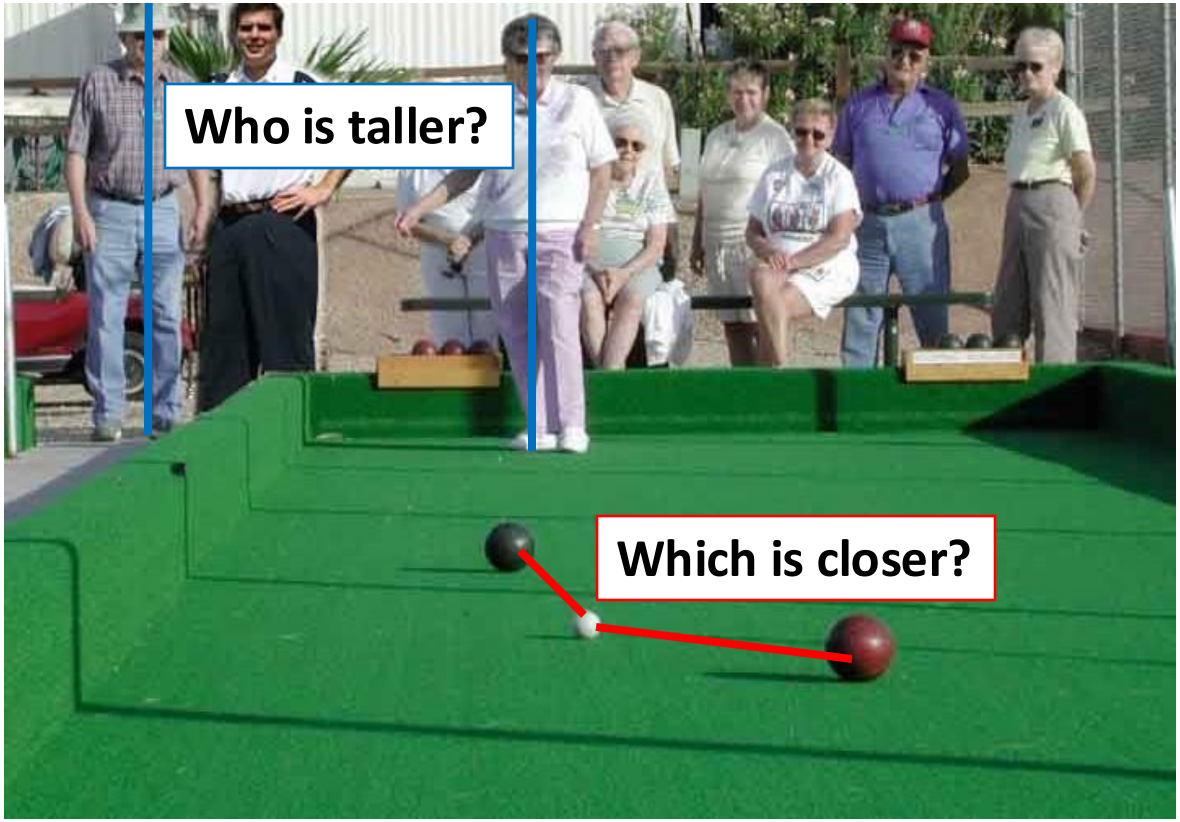


What properties of the world are preserved?

A. Straight Lines
B. Angles between lines
C. Incidence
D. Lengths
E. Parallel Lines
F. Ratio of lengths

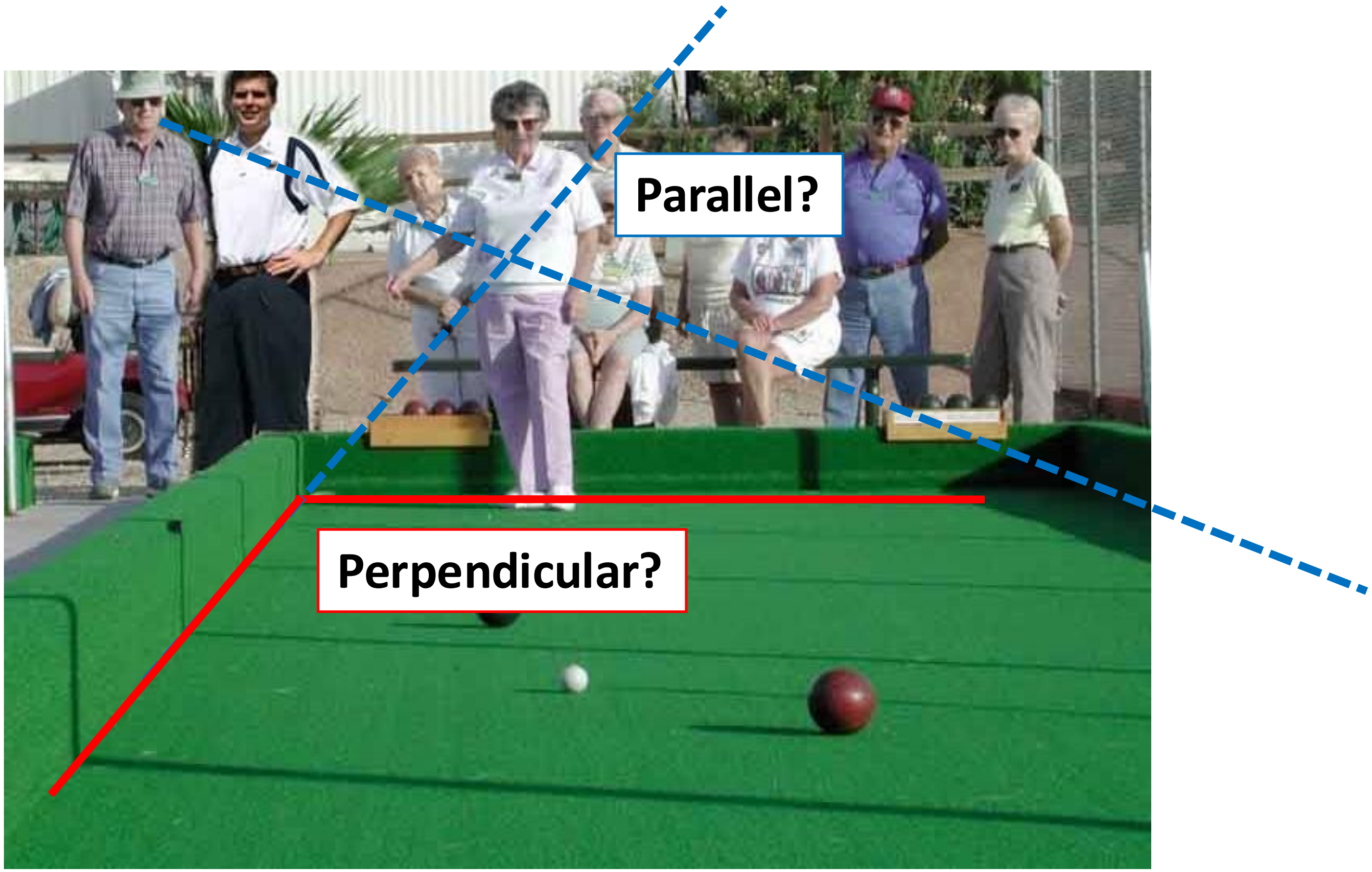
Slide by A. Efros

Properties of projection



Slide by Derek Hoiem

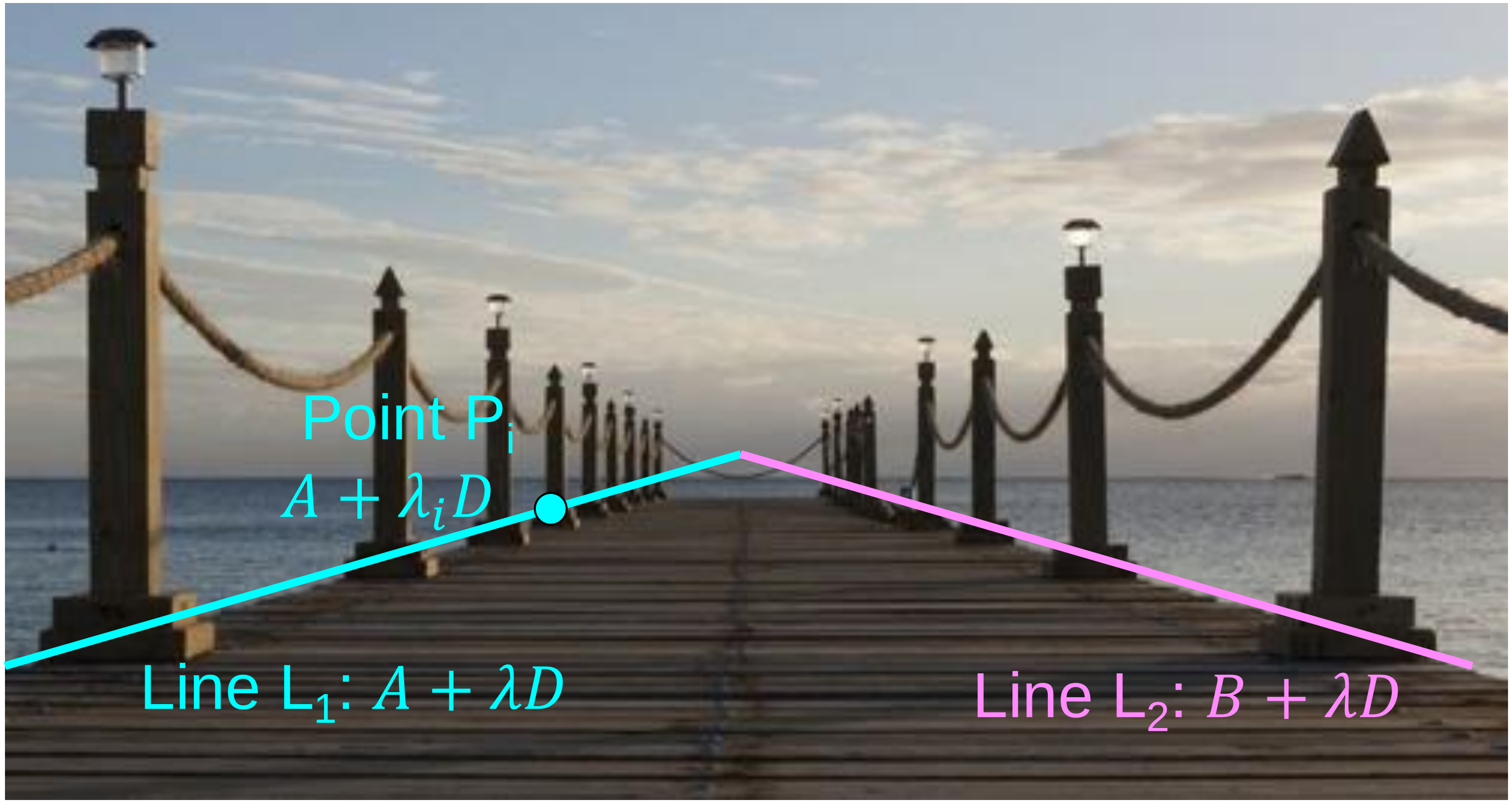
Properties of projection



Slide by Derek Hoiem


Vanishing
Point

Parallel lines converge at a point



$$A = (A_X, A_Y, A_Z)$$
$$B = (B_X, B_Y, B_Z)$$
$$D = (D_X, D_Y, D_Z)$$

$$\begin{aligned} L_1(\lambda) &= A + \lambda D \\ &= (A_X + \lambda D_X, A_Y + \lambda D_Y, A_Z + \lambda D_Z). \end{aligned}$$

$$l_1(\lambda) = \left(f \frac{A_X + \lambda D_X}{A_Z + \lambda D_Z}, f \frac{A_Y + \lambda D_Y}{A_Z + \lambda D_Z} \right)$$

Adapted from B. Hariharan.

Parallel lines converge at a point

$$L_1(\lambda) = A + \lambda D$$
$$= (A_X + \lambda D_X, A_Y + \lambda D_Y, A_Z + \lambda D_Z)$$

$$l_1(\lambda) = \left(f\frac{A_X + \lambda D_X}{A_Z + \lambda D_Z}, f\frac{A_Y + \lambda D_Y}{A_Z + \lambda D_Z} \right).$$

Study, behavior of $l_1(\lambda)$ as $A_Z + \lambda D_Z \rightarrow \infty$,

which is same as $\lambda \rightarrow \infty$.

$$\lim_{\lambda \to \infty} f\frac{A_X + \lambda D_X}{A_Z + \lambda D_Z} = \lim_{\lambda \to \infty} f\frac{A_X/\lambda + D_X}{A_Z/\lambda + D_Z} = \frac{fD_X}{D_Z}.$$

$$\lim_{\lambda \to \infty} l_1(\lambda)$$
$$= \left(f\frac{D_X}{D_Z}, f\frac{D_Y}{D_Z} \right).$$

But, what happens if $D_Z = 0$?

Adapted from B. Hariharan.

Parallel lines converge at a point

$D_Z = 0$ lines?

Adapted from B. Hariharan.

Farther away objects are smaller

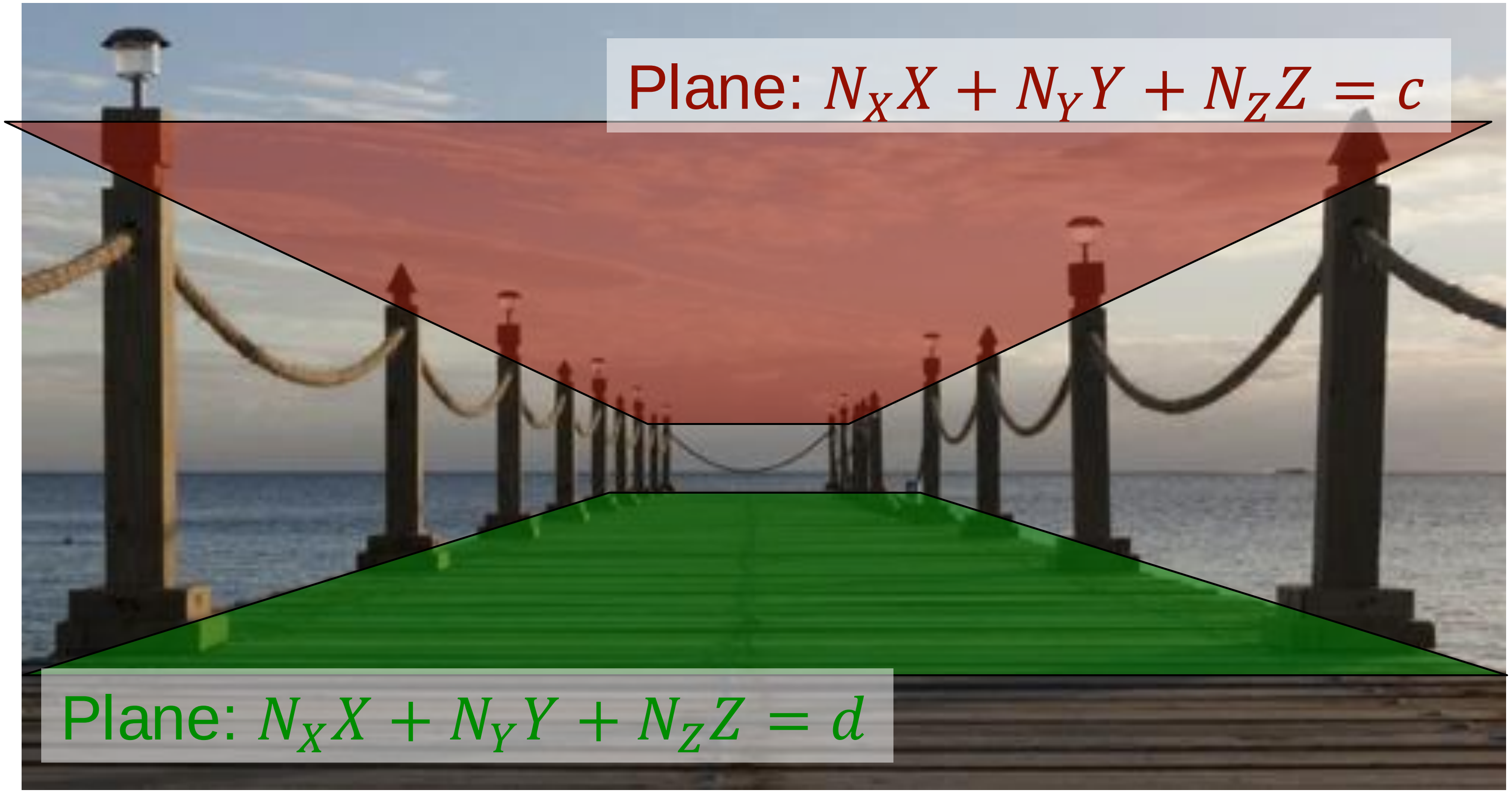
Image of foot: $\left(\frac{fX}{Z}, \frac{fY}{Z}\right)$

Image of head: $\left(\frac{fX}{Z}, \frac{f(Y+h)}{Z}\right)$

Height: $\frac{f(Y+h)}{Z} - \frac{fY}{Z} = \frac{fh}{Z}$

Adapted from B. Hariharan.

What about planes?



$$N_X X + N_y Y + N_Z Z = d$$

$$\frac{N_X f X}{Z} + \frac{N_y f Y}{Z} + f N_Z = \frac{f d}{Z}$$

$$N_X x + N_Y y + f N_Z = \frac{f d}{Z}$$

As $Z \to \infty$,

$$N_X x + N_Y y + f N_Z = 0$$

Planes vanish into a line.

Parallel planes vanish into the same line.

Adapted from B. Hariharan.

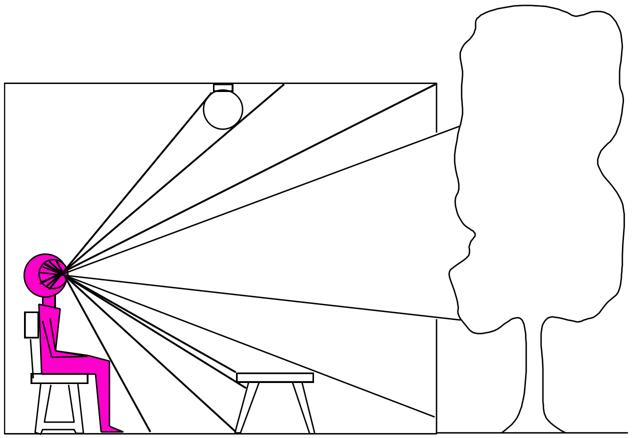
Except?

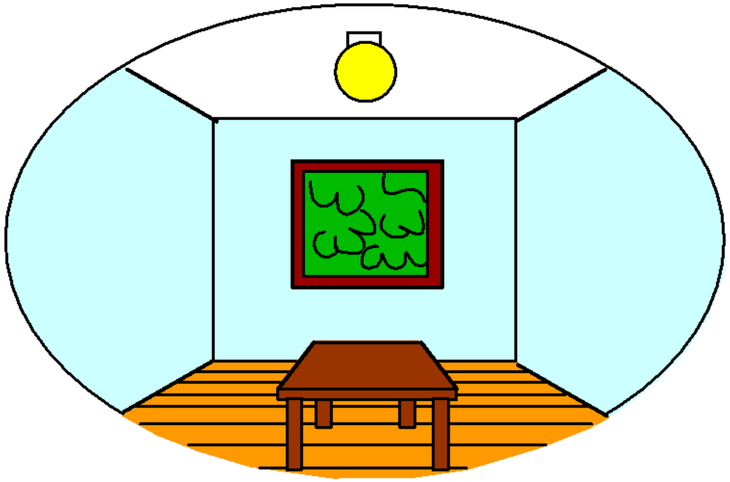
$N_X = 0$ and $N_Y = 0$.

Fronto-parallel plane.

Fronto-parallel planes

- What happens to the projection of a pattern on a plane parallel to the image plane?
  - All points on that plane are at a fixed *depth* z
  - The pattern gets scaled by a factor of f / z, but angles and ratios of lengths/areas are preserved

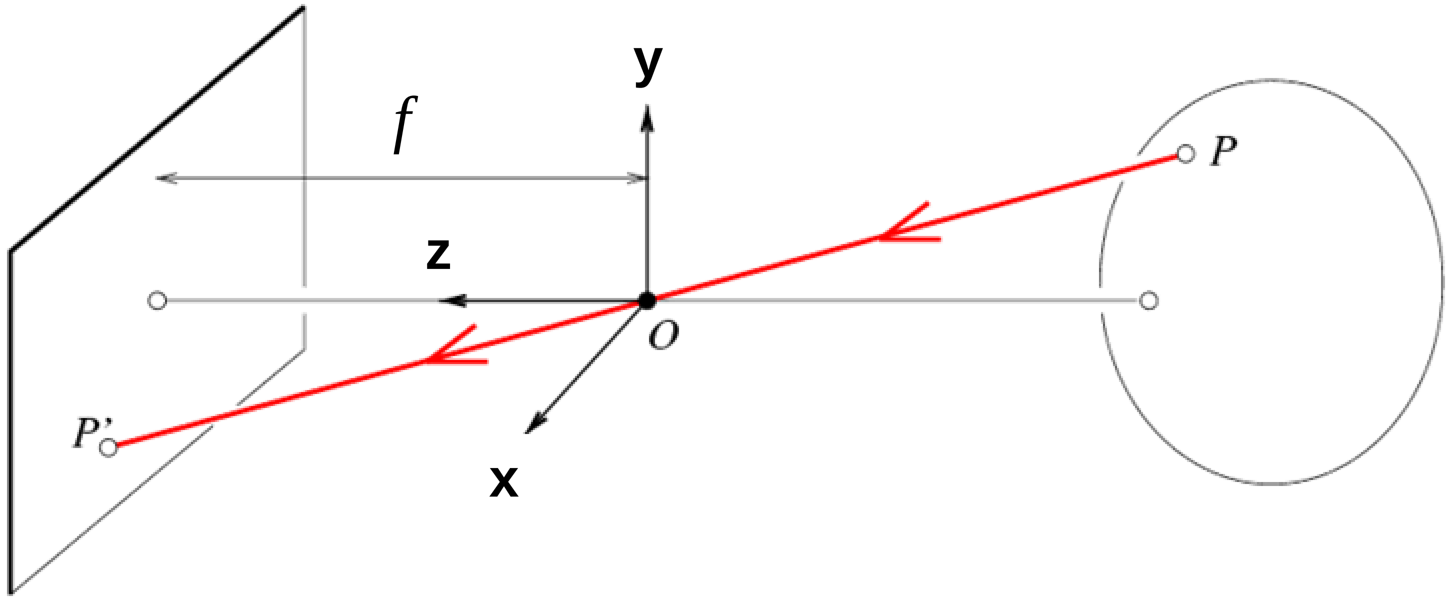




$$(X, Y, Z) \rightarrow \left(\frac{fX}{Z}, \frac{fY}{Z}\right)$$

Slide from L. Lazebnik

Modeling projection



Projection equation: $(X, Y, Z) \rightarrow \left(\frac{fX}{Z}, \frac{fY}{Z}\right) = (x, y)$

Note: instead of dealing with an image that is upside down, most of the time we will pretend that the image plane is in front of the camera center.

Source: J. Ponce, S. Seitz

Homogeneous coordinates

$$(X,Y,Z) \rightarrow \left(\frac{fX}{Z}, \frac{fY}{Z}\right)$$

Is this a linear transformation?

- no—division by z is nonlinear

Trick: add one more coordinate:

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

homogeneous image coordinates

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

homogeneous scene coordinates

Converting *from* homogeneous coordinates

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

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

Slide by Steve Seitz

Perspective Projection Matrix

Projection is a matrix multiplication using homogeneous coordinates

$$\begin{bmatrix} \quad & \quad & \quad & \quad \\ & & & \\ & & & \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} \quad \\ \quad \\ \quad \end{bmatrix}$$

$$\left(f\,\frac{X}{Z},\, f\,\frac{Y}{Z}\right)$$

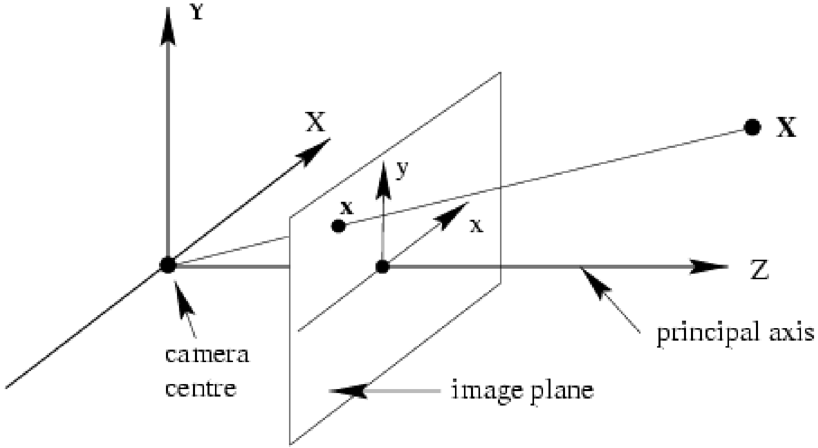
divide by the third coordinate

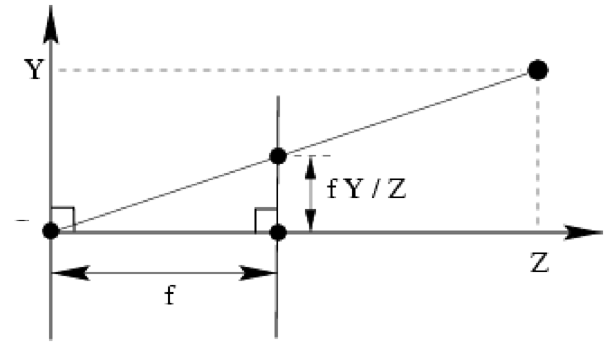
In practice: lots of coordinate transformations…

(2D point (3x1)) = (Camera to pixel coord. trans. matrix (3x3)) (Perspective projection matrix (3x4)) (World to camera coord. trans. matrix (4x4)) (3D point (4x1))

Slide from L. Lazebnik

Review: Pinhole camera model



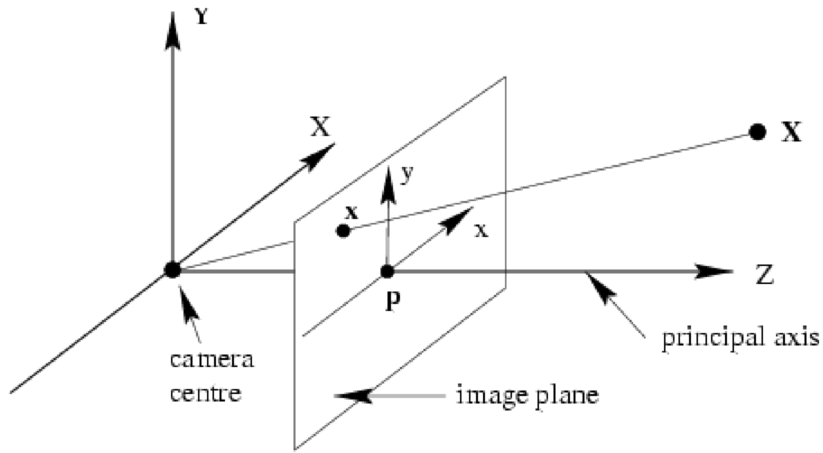


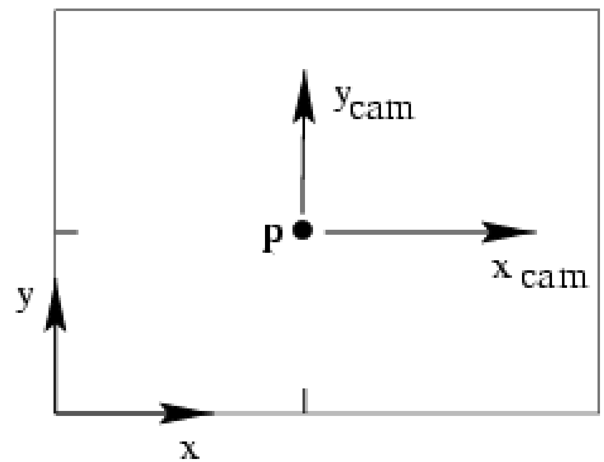
$$(X, Y, Z) \mapsto (f\,X/Z,\ f\,Y/Z)$$

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

$$\lambda \mathrm{x} = \mathrm{PX}$$

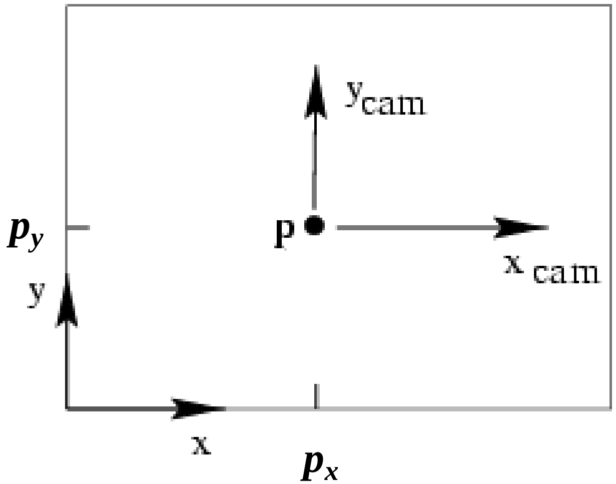
Principal point





- **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

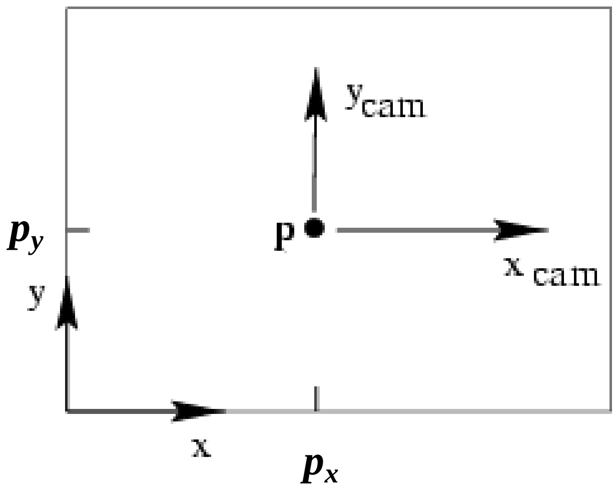


We want the principal point to map to (p_x, p_y) instead of (0,0)

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

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

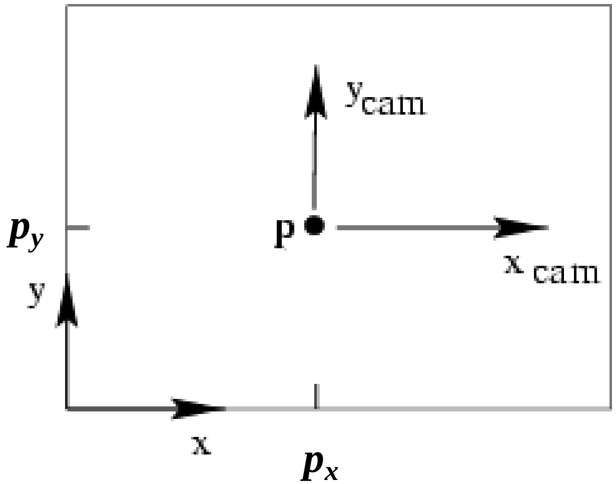
Principal point offset



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



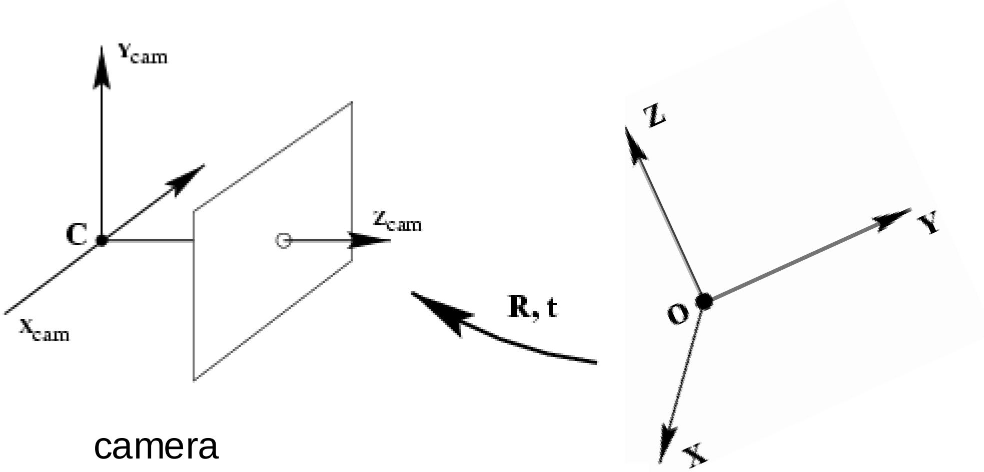
principal point: (p_x, p_y)

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

calibration matrix K projection matrix $[\mathrm{I} \mid 0]$

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

Camera rotation and translation



camera coordinate system

world coordinate system

- 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):

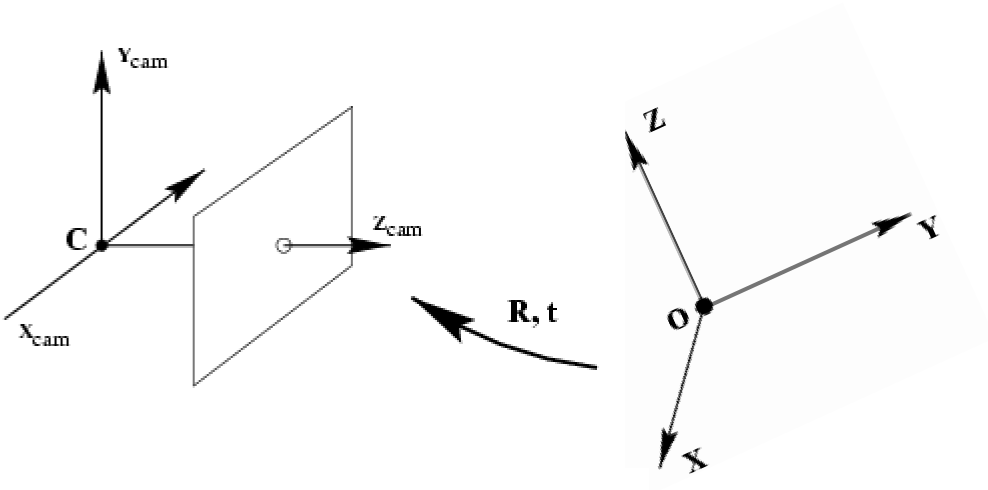
$$\tilde{X}_{cam} = R\left(\tilde{X} - \tilde{C}\right)$$

coords. of point in camera frame

coords. of a point in world frame

coords. of camera center in world frame

Camera rotation and translation

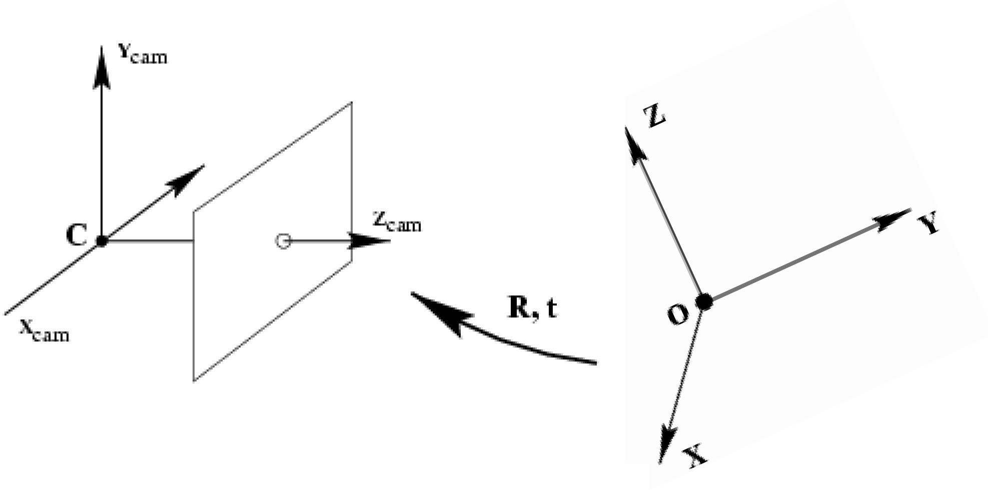


$$\tilde{X}_{cam} = R\left(\tilde{X} - \tilde{C}\right)$$

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

3D transformation
matrix (4 x 4)

Camera rotation and translation

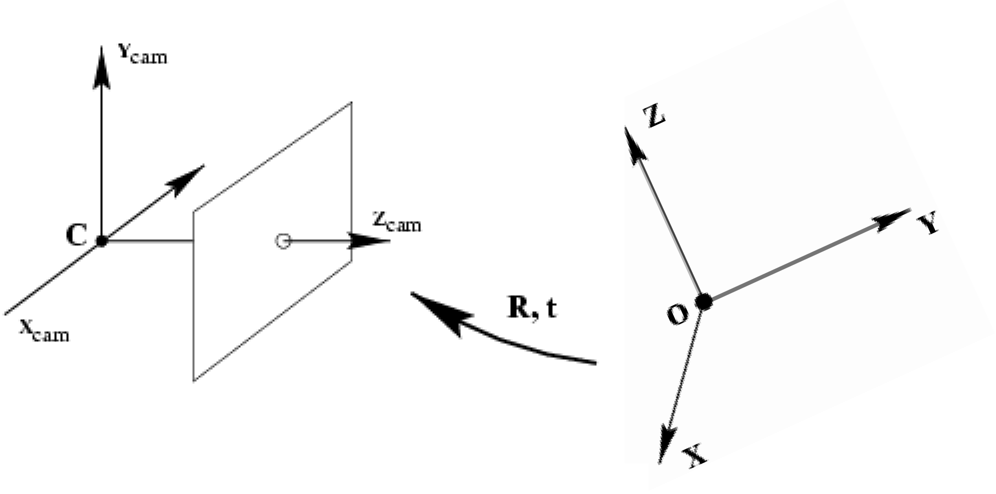


$$\tilde{X}_{cam} = R\left(\tilde{X} - \tilde{C}\right)$$

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

3D transformation
matrix (4 x 4)

Camera rotation and translation



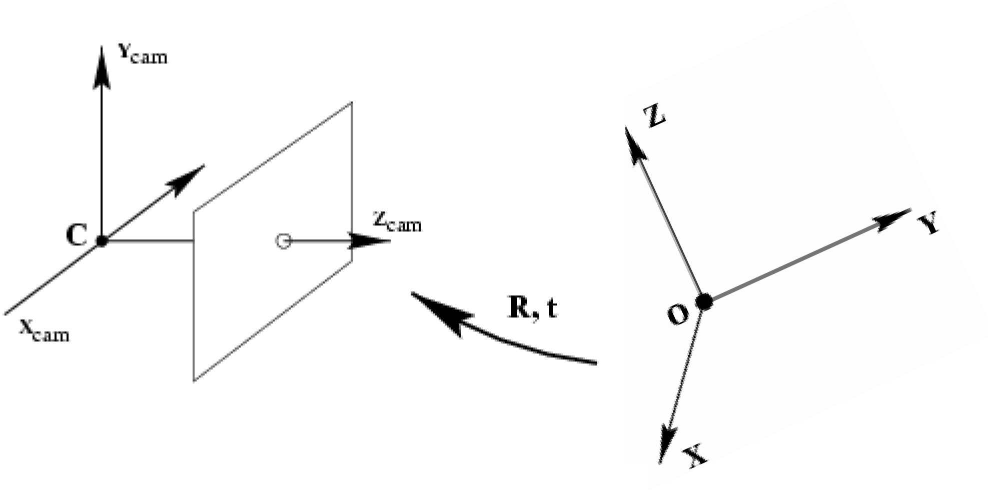
$$x = K[I \mid 0]\begin{bmatrix} R & -R\tilde{C} \\ 0 & 1 \end{bmatrix} X$$

2D transformation matrix (3 x 3)

perspective projection matrix (3 x 4)

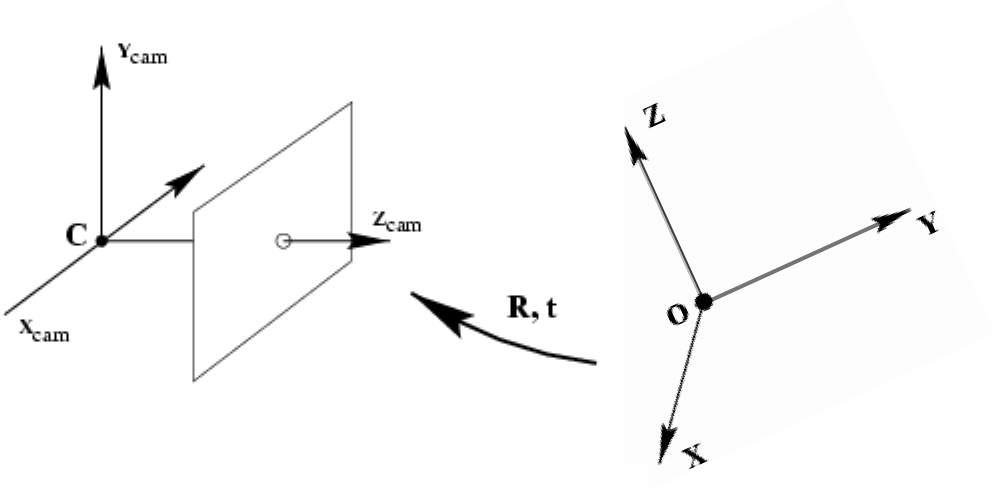
3D transformation matrix (4 x 4)

Camera rotation and translation



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

Camera rotation and translation

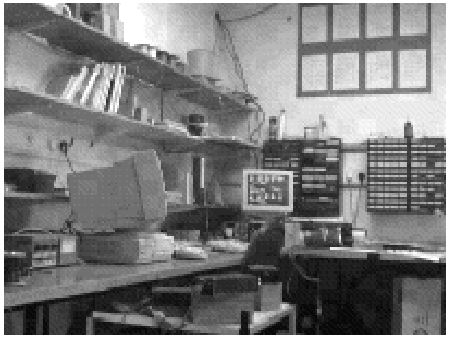


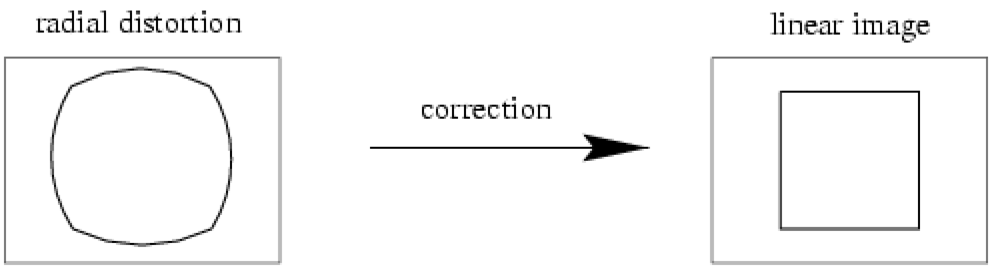
$$\mathrm{x} = \mathrm{K}\left[\mathrm{R} \mid \mathrm{t}\right]\mathrm{X} \qquad \mathrm{t} = -\mathrm{R}\tilde{\mathrm{C}}$$

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?

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

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

Camera calibration

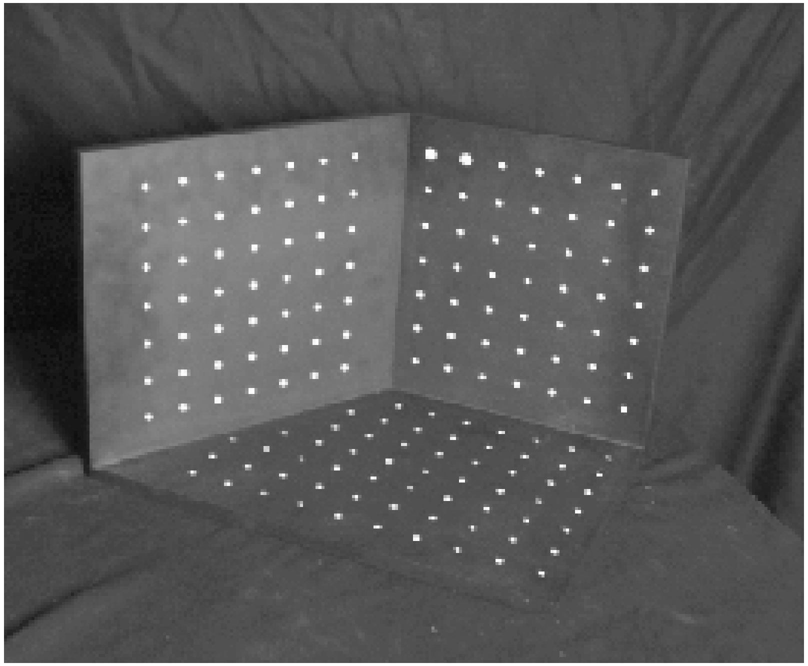
$$\lambda \mathbf{x} = \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \mathbf{X}$$

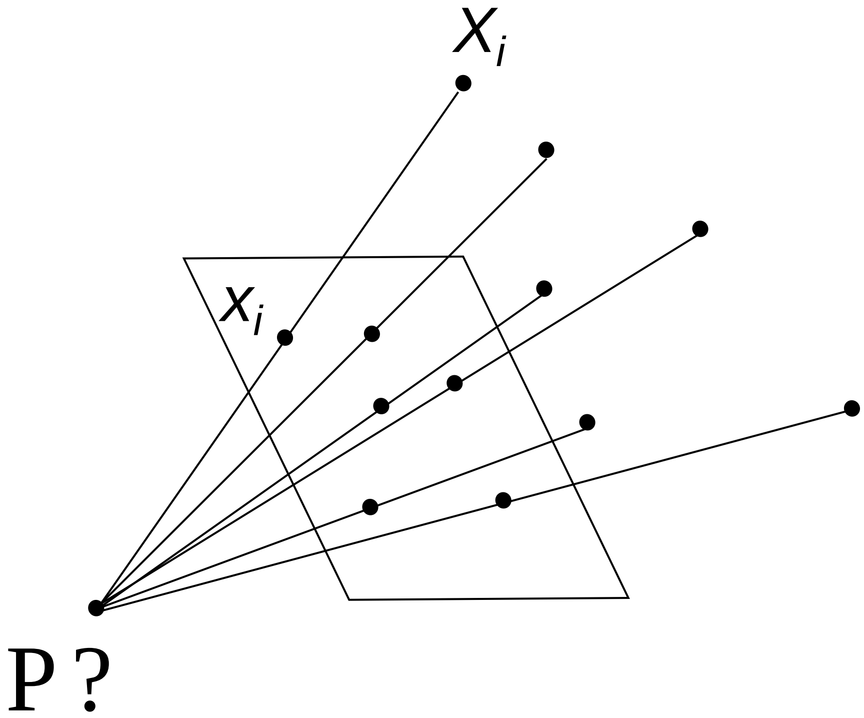
$$\begin{bmatrix} \lambda x \\ \lambda y \\ \lambda \end{bmatrix} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Source: D. Hoiem

Camera calibration

- Given *n* points with known 3D coordinates $\mathbf{X}_i$ and known image projections $\mathbf{x}_i$, 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 $\mathbf{p}$ minimizing $||\mathbf{Ap}||^2$
  - Solution given by eigenvector of $\mathbf{A^T A}$ 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

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

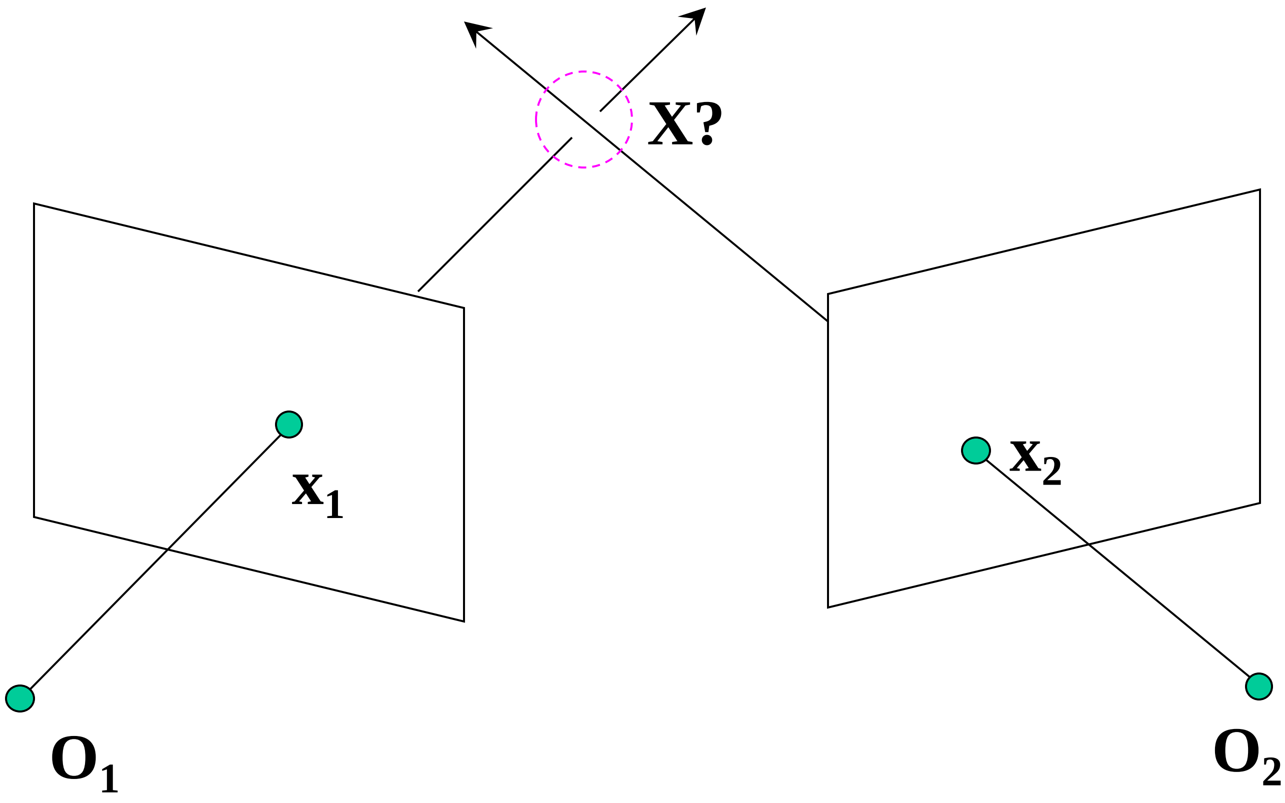
- In practice, non-linear methods are preferred
  - Write down objective function in terms of intrinsic and extrinsic parameters
  - Define error as sum of squared distances between measured 2D points and estimated projections of 3D points
  - Minimize error using Newton's method or other non-linear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

Triangulation

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

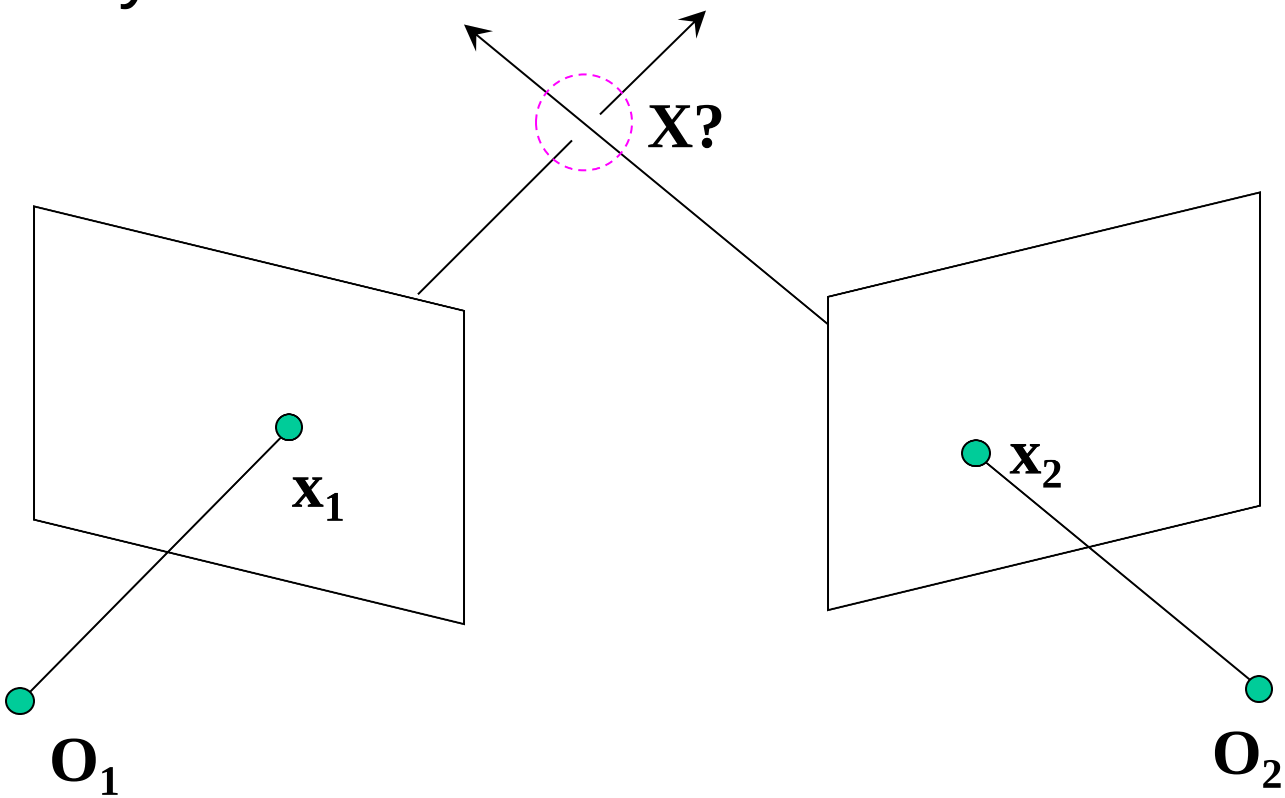
Triangulation

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



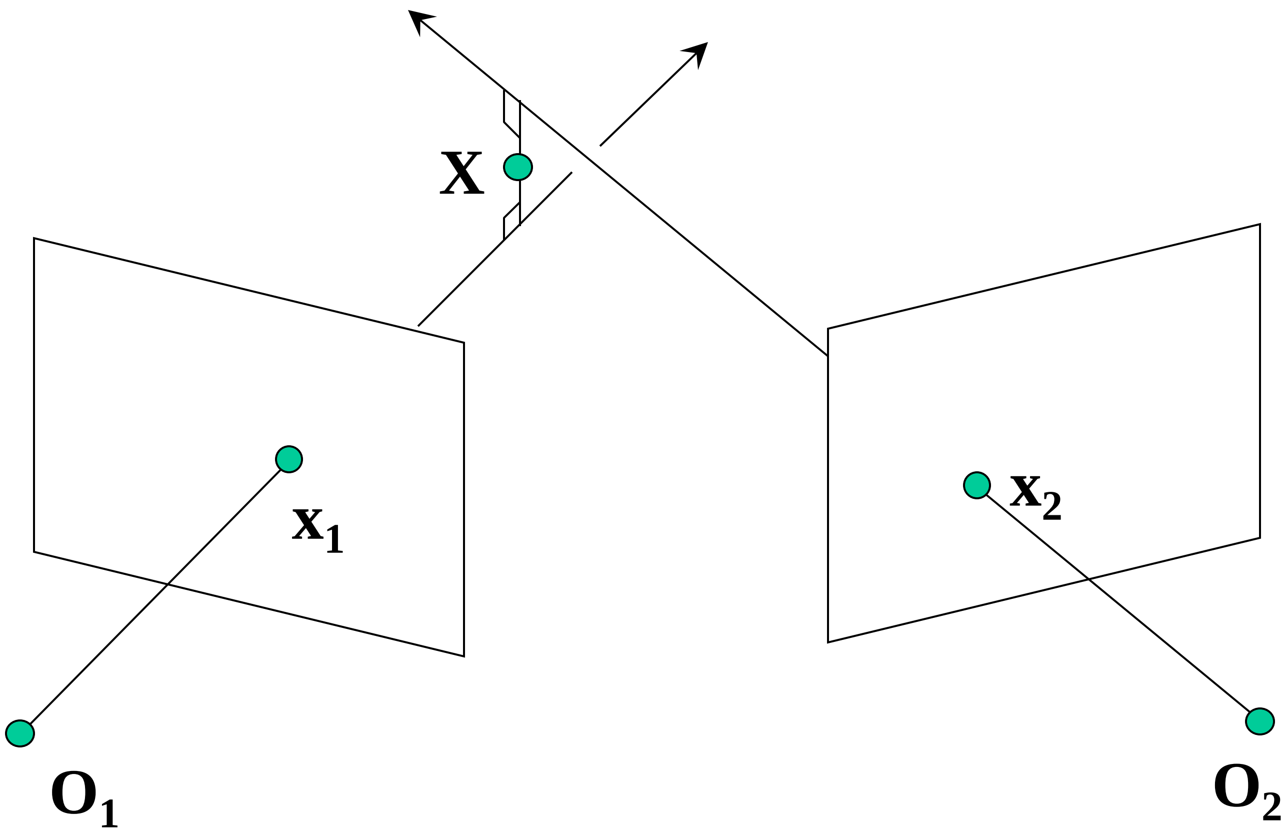
Triangulation

- 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

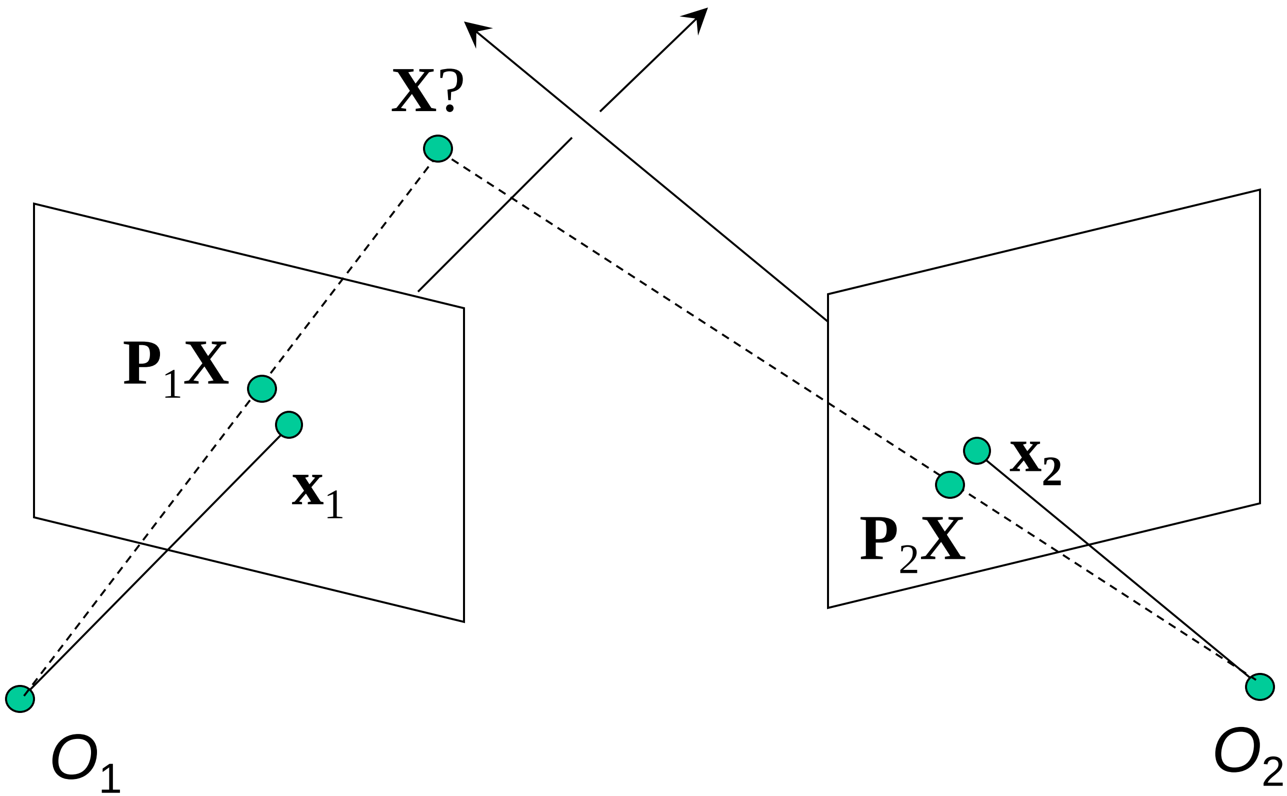
- Find shortest segment connecting the two viewing rays and let **X** be the midpoint of that segment



Triangulation: Nonlinear approach

Find X that minimizes

$$d^2(\mathbf{x_1}, \mathbf{P_1X}) + d^2(\mathbf{x_2}, \mathbf{P_2X})$$



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

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

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

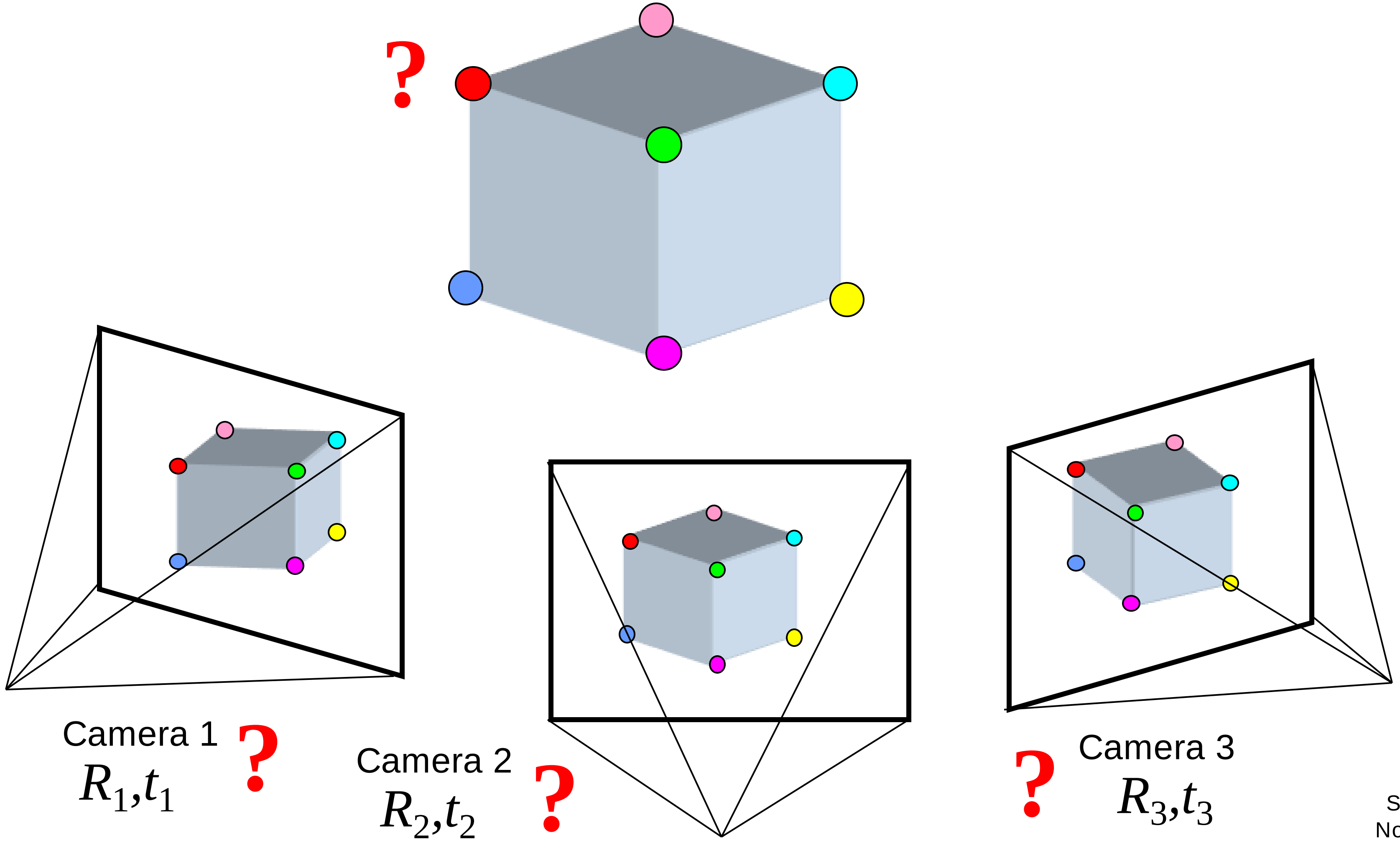
Two independent equations each in terms of three unknown entries of **X**

Outline

- Pinhole camera model
- Modeling pinhole cameras
- Camera Matrices
- Camera calibration
- Triangulation
- *Epipolar geometry*
- Structure from Motion (SfM)
- SfM pipeline
- Depth Sensors

Structure from motion

- Given a set of corresponding points in two or more images, compute the camera parameters and the 3D point coordinates



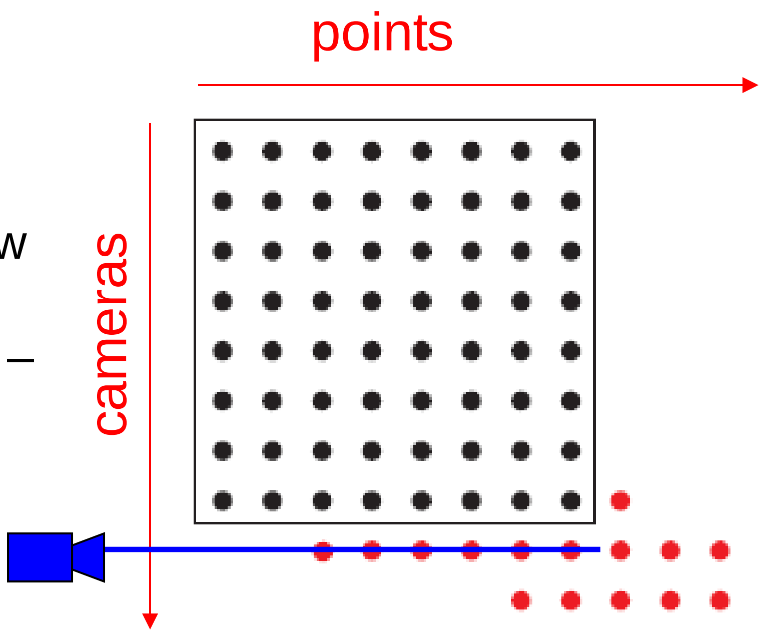
Slide credit: Noah Snavely

Incremental structure from motion

- Initialize motion from two images using fundamental matrix

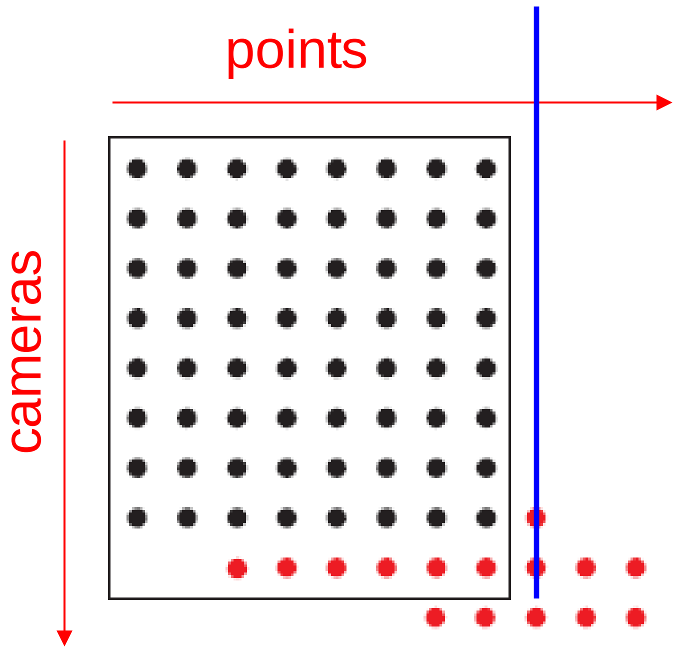
- Initialize structure by triangulation

- For each additional view:
  - Determine projection matrix of new camera using all the known 3D points that are visible in its image – *calibration*



Incremental structure from motion

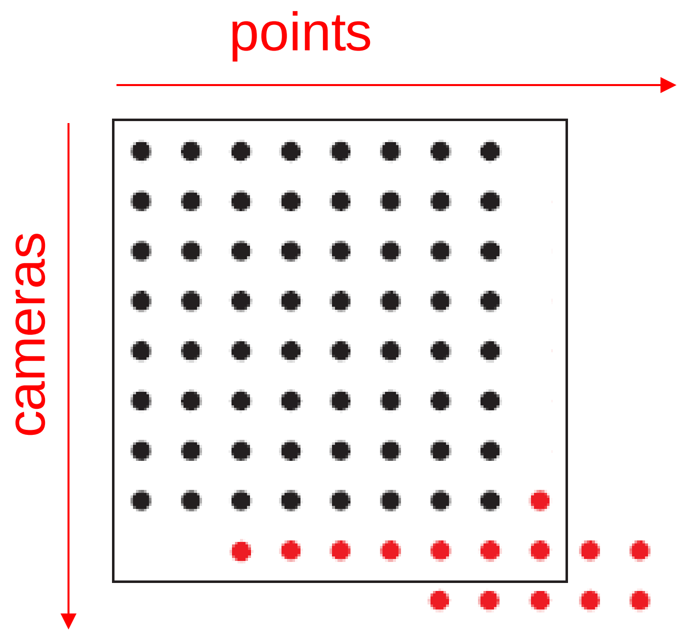
- Initialize motion from two images using fundamental matrix
- Initialize structure by triangulation
- For each additional view:
  - Determine projection matrix of new camera using all the known 3D points that are visible in its image – *calibration*
  - Refine and extend structure: compute new 3D points, re-optimize existing points that are also seen by this camera – *triangulation*



Slide from L. Lazebnik.

Incremental structure from motion

- Initialize motion from two images using fundamental matrix
- Initialize structure by triangulation
- For each additional view:
  - Determine projection matrix of new camera using all the known 3D points that are visible in its image – *calibration*
  - Refine and extend structure: compute new 3D points, re-optimize existing points that are also seen by this camera – *triangulation*
- Refine structure and motion: bundle adjustment



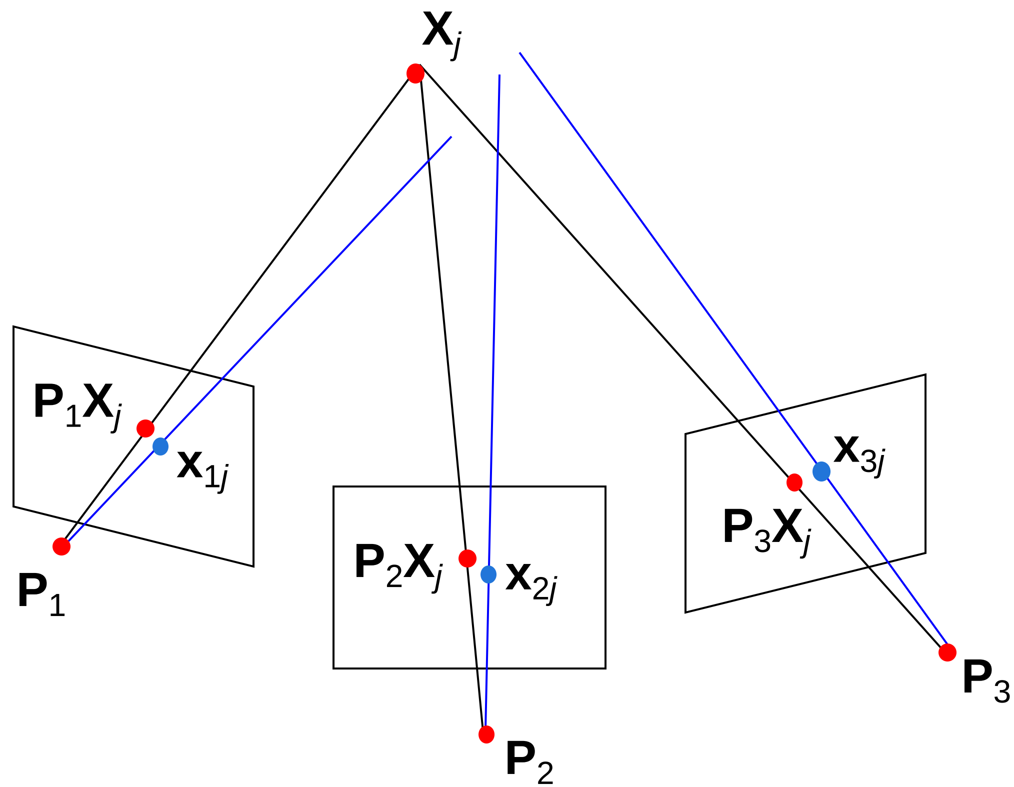
Slide from L. Lazebnik.

Bundle adjustment

- Non-linear method for refining structure and motion
- Minimize reprojection error

$$\sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij} \left\| \mathbf{x}_{ij} - \frac{1}{\lambda_{ij}} \mathbf{P}_i \mathbf{X}_j \right\|^2$$

Visibility Flag: Is point j visible in view i?



Slide from L. Lazebnik.

Initialization

- Compute fundamental matrix $\mathbf{F}$ between the two views
- First camera matrix: $[I|0]$
- Second camera matrix: $[A\,|b]$
- where, b is the epipole (i.e. $F^T b = 0$) and $A = [b_\times]F$

9.5.3 Canonical cameras given F

We have shown that F determines the camera pair up to a projective transformation of 3-space. We will now derive a specific formula for a pair of cameras with canonical form given F. We will make use of the following characterization of the fundamental matrix F corresponding to a pair of camera matrices:

Result 9.12. *A non-zero matrix* F *is the fundamental matrix corresponding to a pair of camera matrices* P *and* P′ *if and only if* $P'^{\mathsf{T}}FP$ *is skew-symmetric.*

Proof. The condition that $P'^{\mathsf{T}}FP$ is skew-symmetric is equivalent to $\mathbf{X}^{\mathsf{T}}P'^{\mathsf{T}}FP\mathbf{X} = 0$ for all $\mathbf{X}$. Setting $\mathbf{x}' = P'\mathbf{X}$ and $\mathbf{x} = P\mathbf{X}$, this is equivalent to $\mathbf{x}'^{\mathsf{T}}F\mathbf{x} = 0$, which is the defining equation for the fundamental matrix. □

Slide from L. Lazebnik, Hartley and Zisserman

F&P sec. 8.3.2

Initialization

- Compute fundamental matrix $\mathbf{F}$ between the two views
- First camera matrix: $[I|0]$
- Second camera matrix: $[A\,|b]$
- where, b is the epipole (i.e. $F^T b = 0$) and $A = [b_\times]F$

Result 9.13. *Let* F *be a fundamental matrix and* S *any skew-symmetric matrix. Define the pair of camera matrices*

$$\mathrm{P} = [\mathrm{I} \mid \mathbf{0}] \quad \textit{and} \quad \mathrm{P}' = [\mathrm{SF} \mid \mathbf{e}'],$$

where $\mathbf{e}'$ *is the epipole such that* $\mathbf{e}'^{\mathsf{T}}\mathrm{F} = \mathbf{0}$*, and assume that* P' *so defined is a valid camera matrix (has rank 3). Then* F *is the fundamental matrix corresponding to the pair* $(\mathrm{P}, \mathrm{P}')$.

Result 9.14. *The camera matrices corresponding to a fundamental matrix* F *may be chosen as* $\mathrm{P} = [\mathrm{I} \mid \mathbf{0}]$ *and* $\mathrm{P}' = [[\mathbf{e}']_\times \mathrm{F} \mid \mathbf{e}']$.

Incremental SFM

- Pick a pair of images with lots of inliers (and preferably, good EXIF data)
  - Initialize intrinsic parameters (focal length, principal point) from EXIF
  - Estimate extrinsic parameters (**R** and **t**) using five-point algorithm
  - Use triangulation to initialize model points
- While remaining images exist
  - Find an image with many feature matches with images in the model
  - Run RANSAC on feature matches to register new image to model
  - Triangulate new points
  - Perform bundle adjustment to re-optimize everything

Slide from L. Lazebnik.

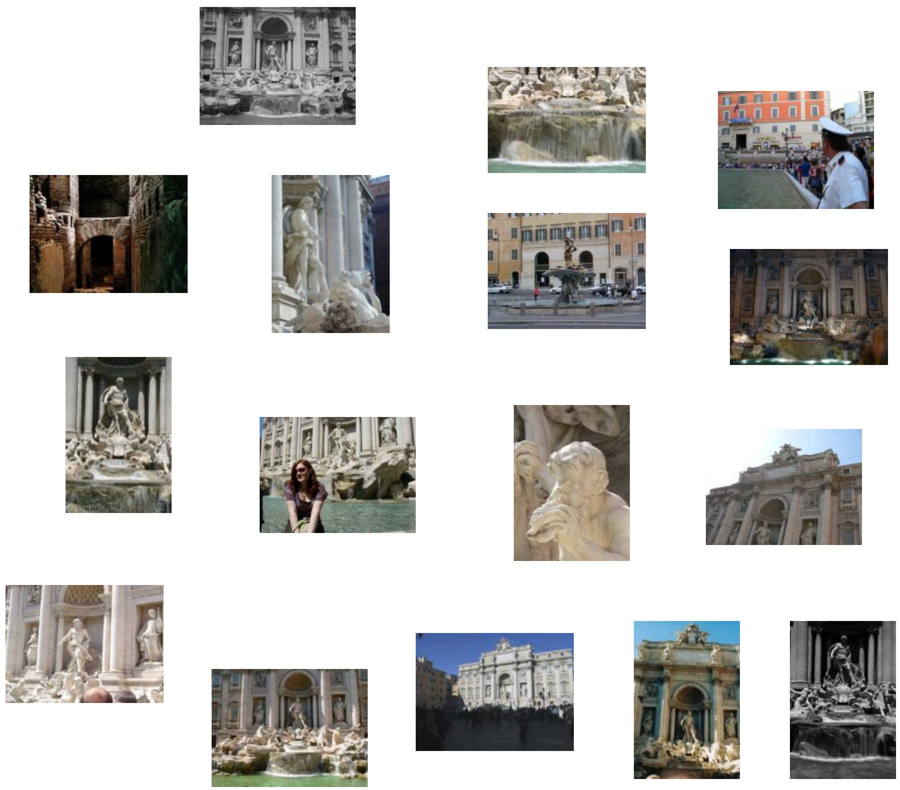
Representative SFM pipeline

N. Snavely, S. Seitz, and R. Szeliski, [Photo tourism: Exploring photo collections in 3D](http://phototour.cs.washington.edu/), SIGGRAPH 2006.
http://phototour.cs.washington.edu/

Slide from L. Lazebnik.

Feature detection

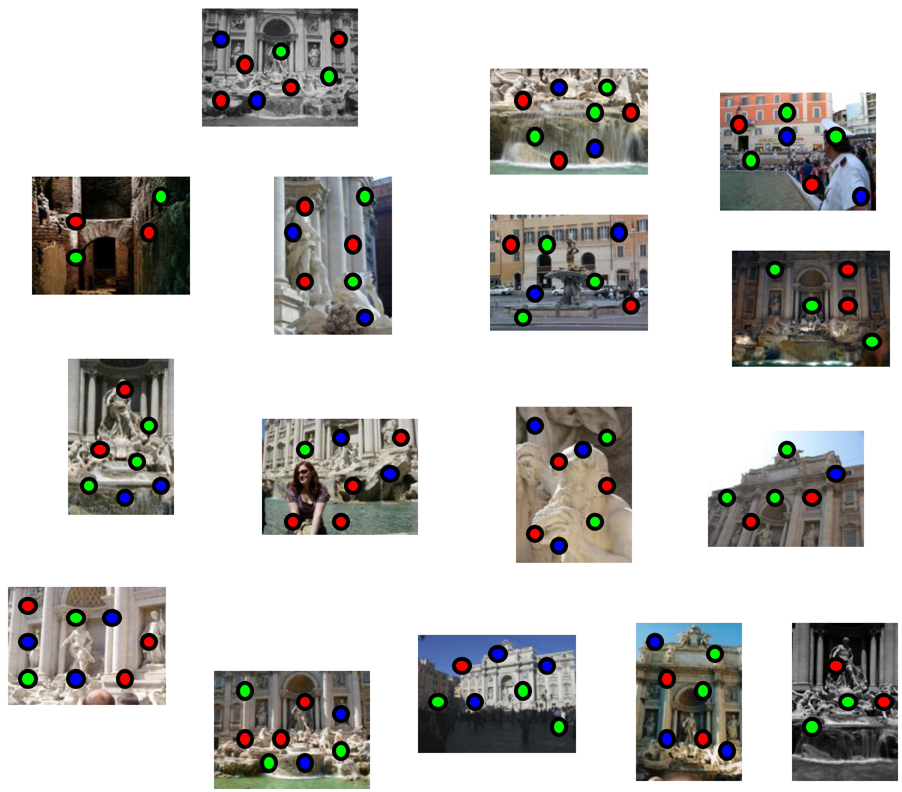
Detect SIFT features



Source: N. Snavely

Feature detection

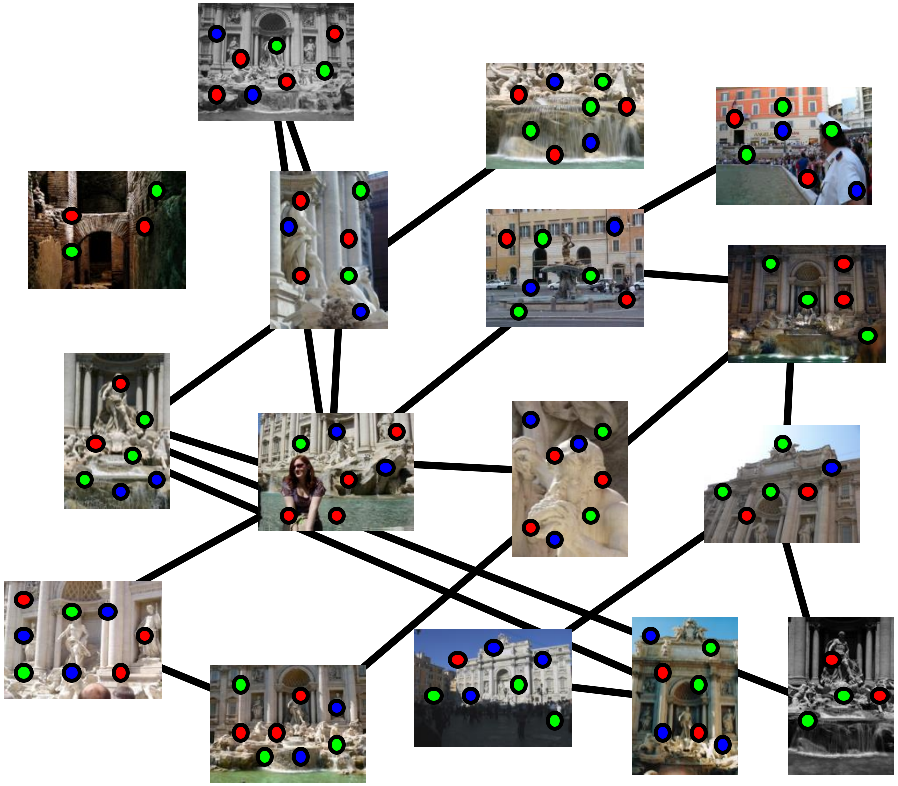
Detect SIFT features



Source: N. Snavely

Feature matching

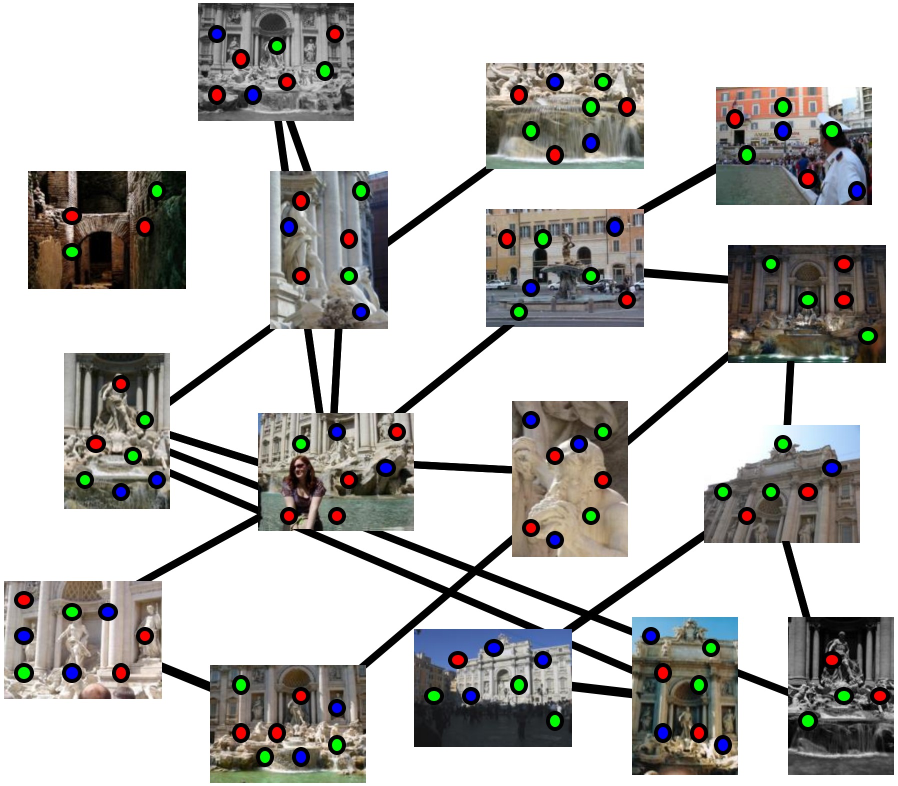
Match features between each pair of images



Source: N. Snavely

Feature matching

Use RANSAC to estimate fundamental matrix between each pair



Source: N. Snavely

Feature matching

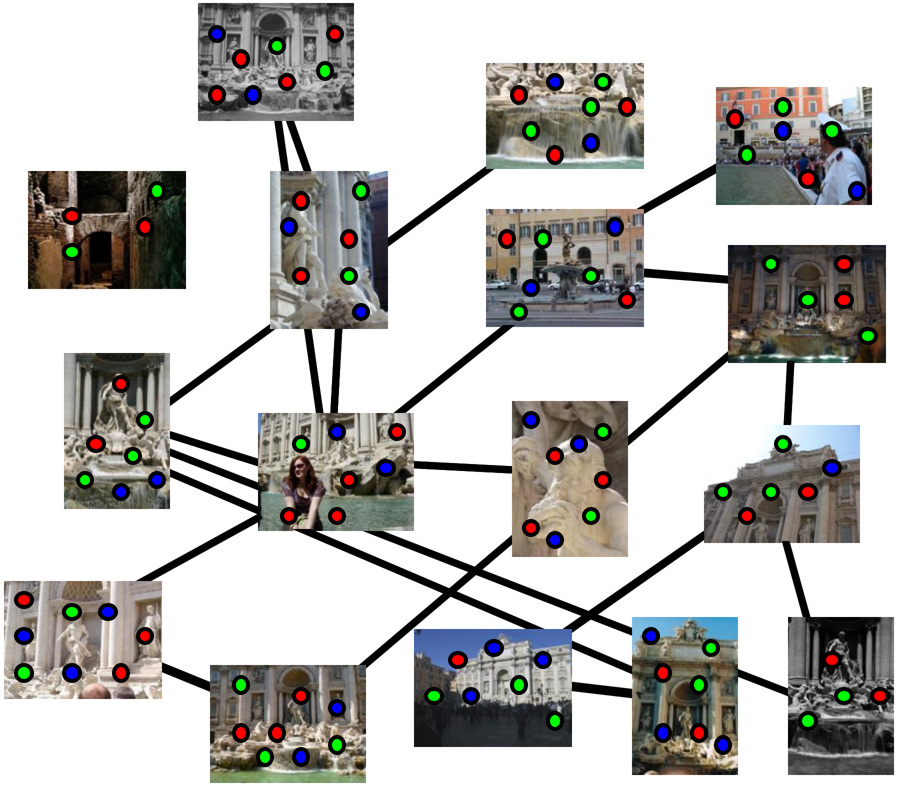
Use RANSAC to estimate fundamental matrix between each pair

Slide from L. Lazebnik.

Image source

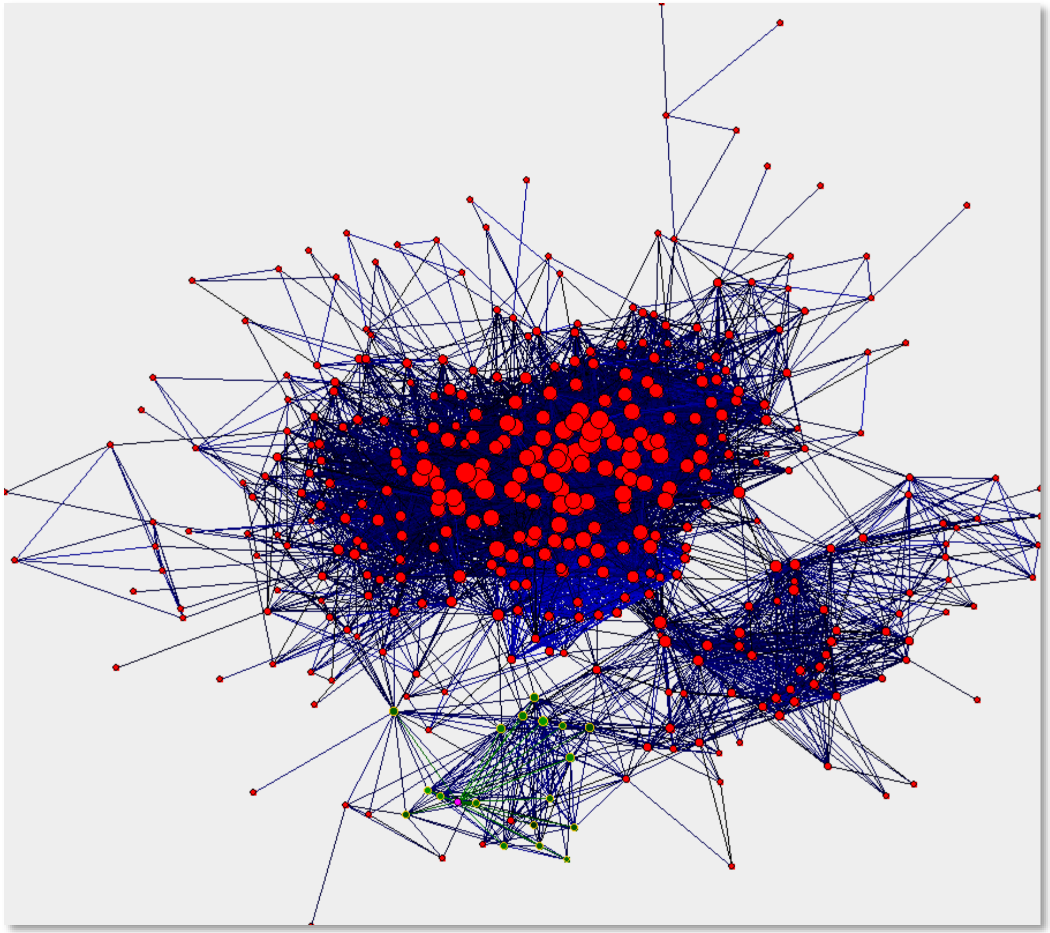
Feature matching

Use RANSAC to estimate fundamental matrix between each pair



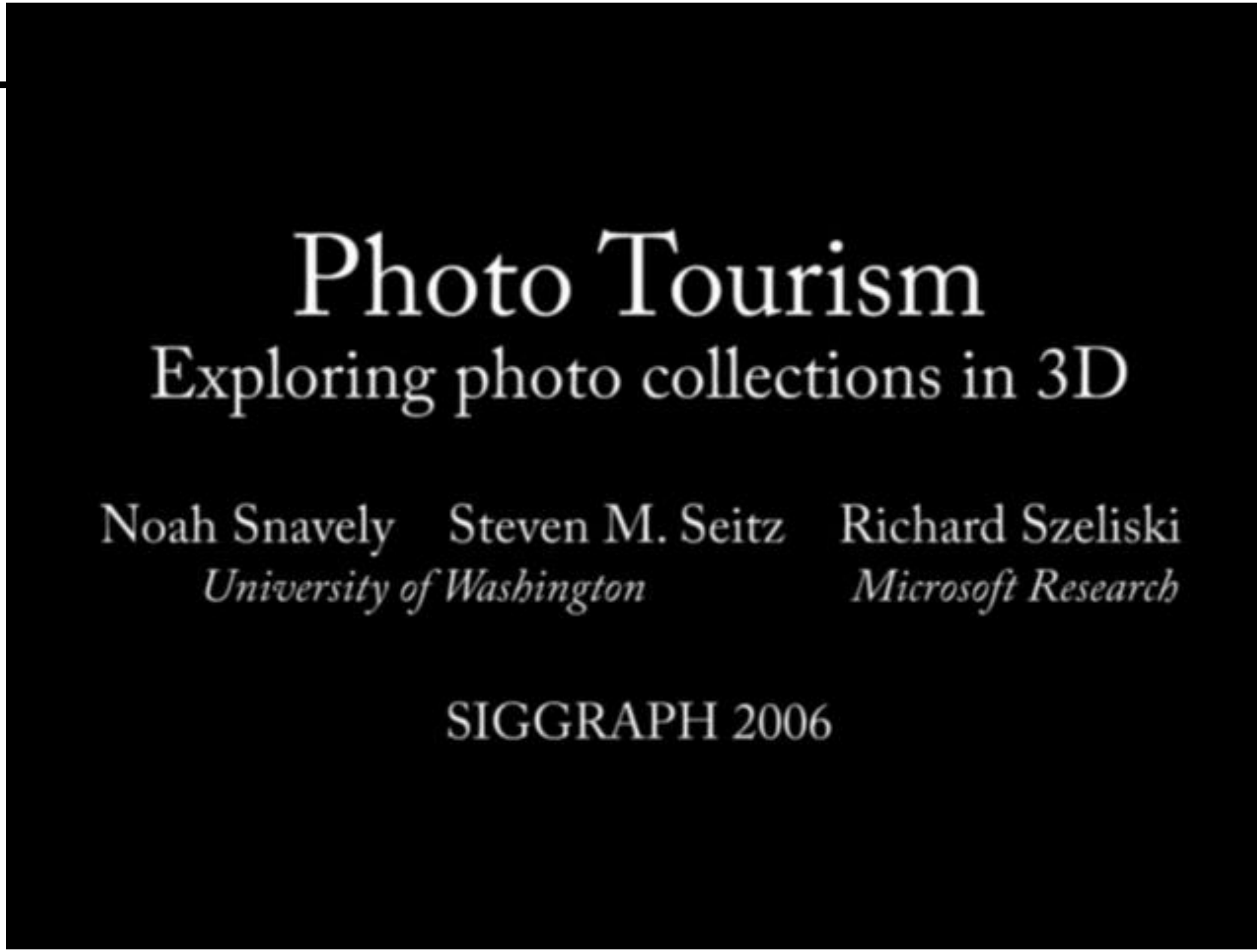
Source: N. Snavely

Image connectivity graph



(graph layout produced using the Graphviz toolkit: http://www.graphviz.org/)

Source: N. Snavely



N. Snavely, S. Seitz, and R. Szeliski, Photo tourism: Exploring photo collections in 3D, SIGGRAPH 2006. http://phototour.cs.washington.edu/
See also: http://grail.cs.washington.edu/projects/rome/

The devil is in the details

- Handling degenerate configurations (e.g., homographies)
- Eliminating outliers
- Dealing with repetitions and symmetries

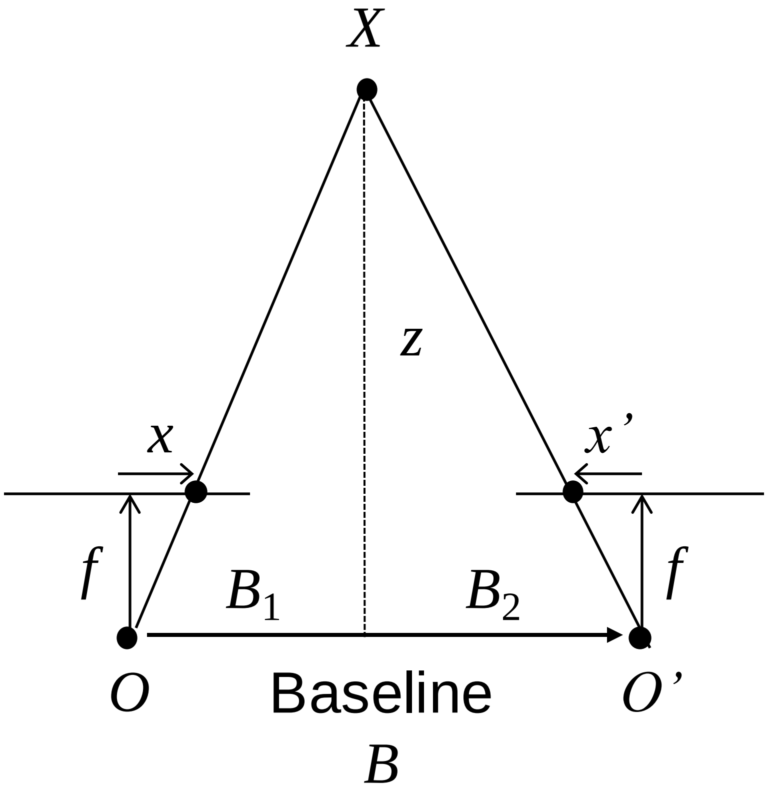
Slide from L. Lazebnik.

SFM software

- Bundler
- OpenSfM
- OpenMVG
- VisualSFM
- COLMAP
- ORB SLAM2
- See also Wikipedia's list of toolboxes

Slide from L. Lazebnik.

Triangulation for Stereo Cameras



$$\frac{x}{f} = \frac{B_1}{z} \qquad \frac{-x'}{f} = \frac{B_2}{z}$$

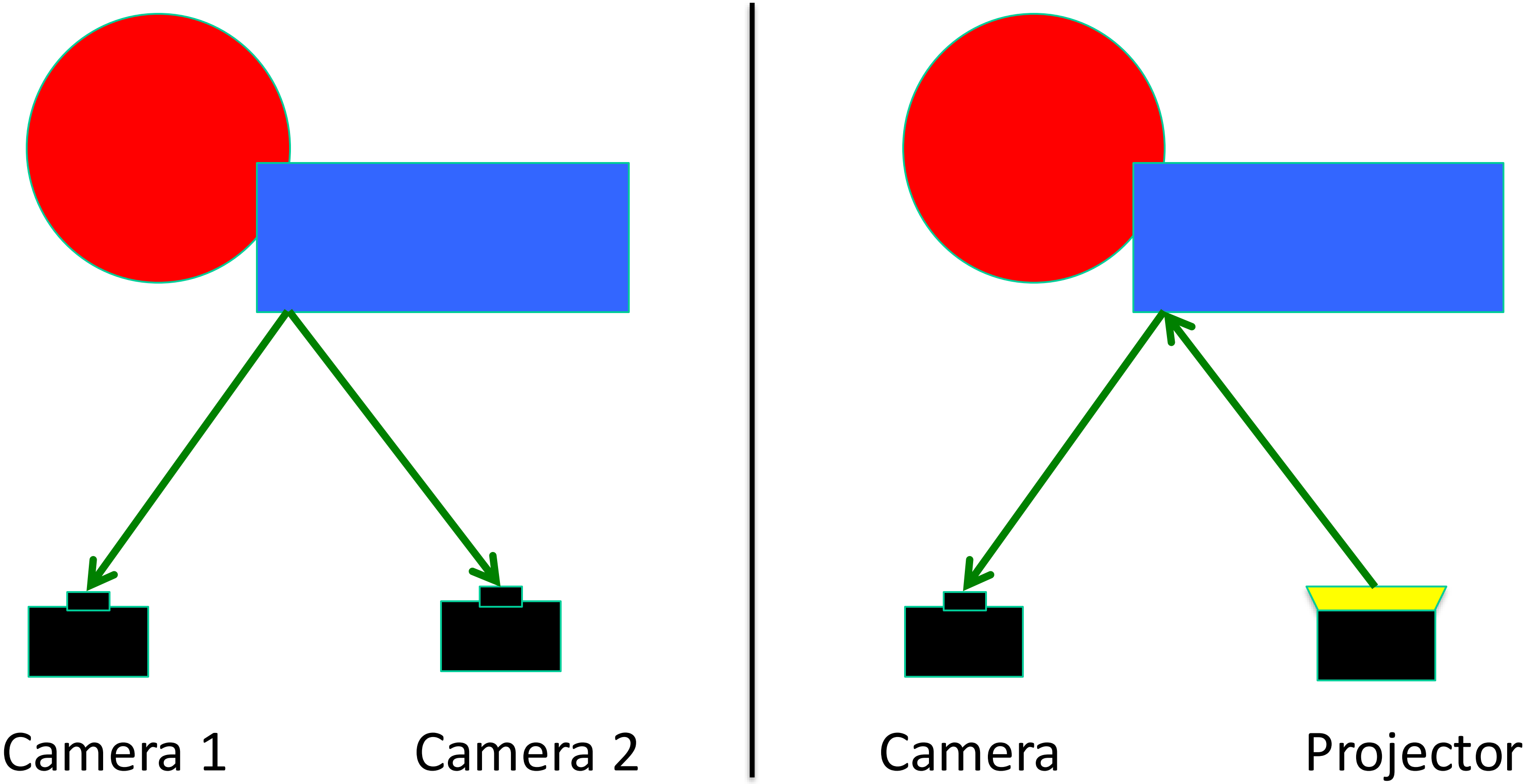
$$\frac{x - x'}{f} = \frac{B_1 + B_2}{z}$$

$$disparity = x - x' = \frac{B \cdot f}{z}$$

Disparity is inversely proportional to depth!

Slide from L. Lazebnik.

Depth from Triangulation



Passive Stereopsis

Active Stereopsis

Active sensing simplifies the problem of estimating point correspondences

Kinect: Structured infrared light

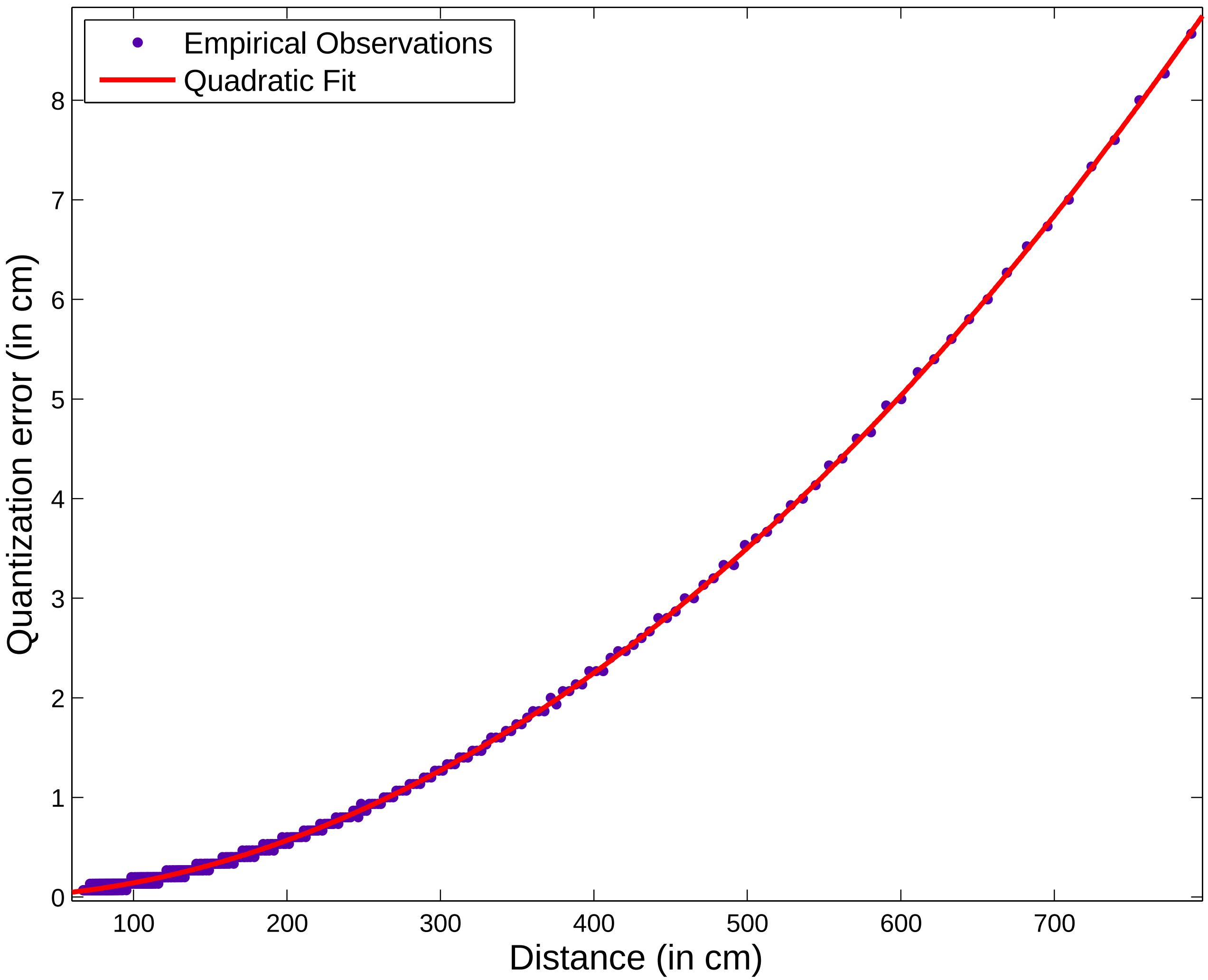
http://bbzippo.wordpress.com/2010/11/28/kinect-in-infrared/

Slide from L. Lazebnik.

Stereo error(distance)

Error in distance estimate increases quadratically with the distance

$$
\begin{aligned}
Z &= \text{distance} \\
d &= \text{disparity} \\
Z &= \frac{C}{d} \\
\delta Z &= \frac{-Z^2}{C}\delta d \\
|\delta Z| &= \frac{Z^2}{C}|\delta d| \\
\text{error} &\propto \text{distance}^2
\end{aligned}
$$



Useful reference