UNIVERSITÄT BONN

Juergen Gall

Camera Calibration

MA-INF 2201 - Computer Vision

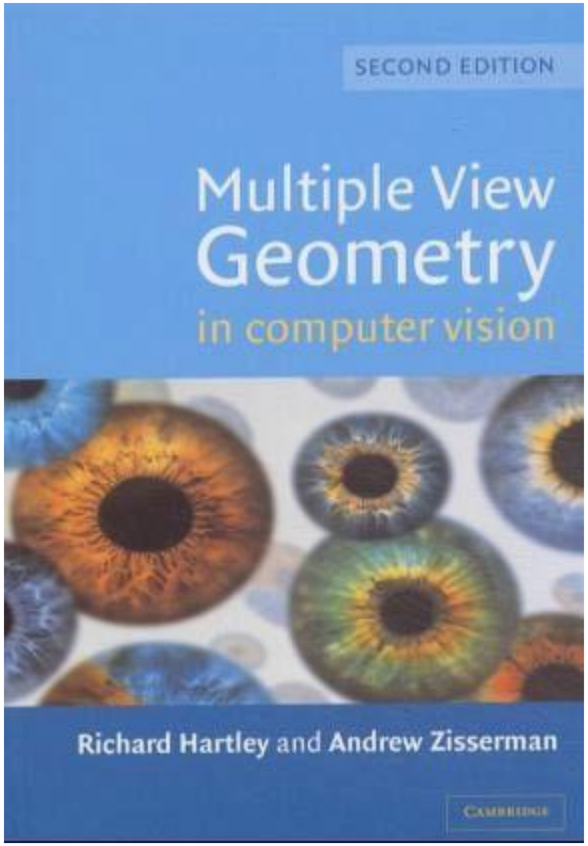
WS24/25

MA-INF 2206 - Seminar Vision

- First meeting: Friday, 24.1., 11:45, Friedrich-Hirzebruch Allee 5, HS 3

- Second meeting: Friday, 31.1., 11:45, Friedrich-Hirzebruch Allee 5, HS 3

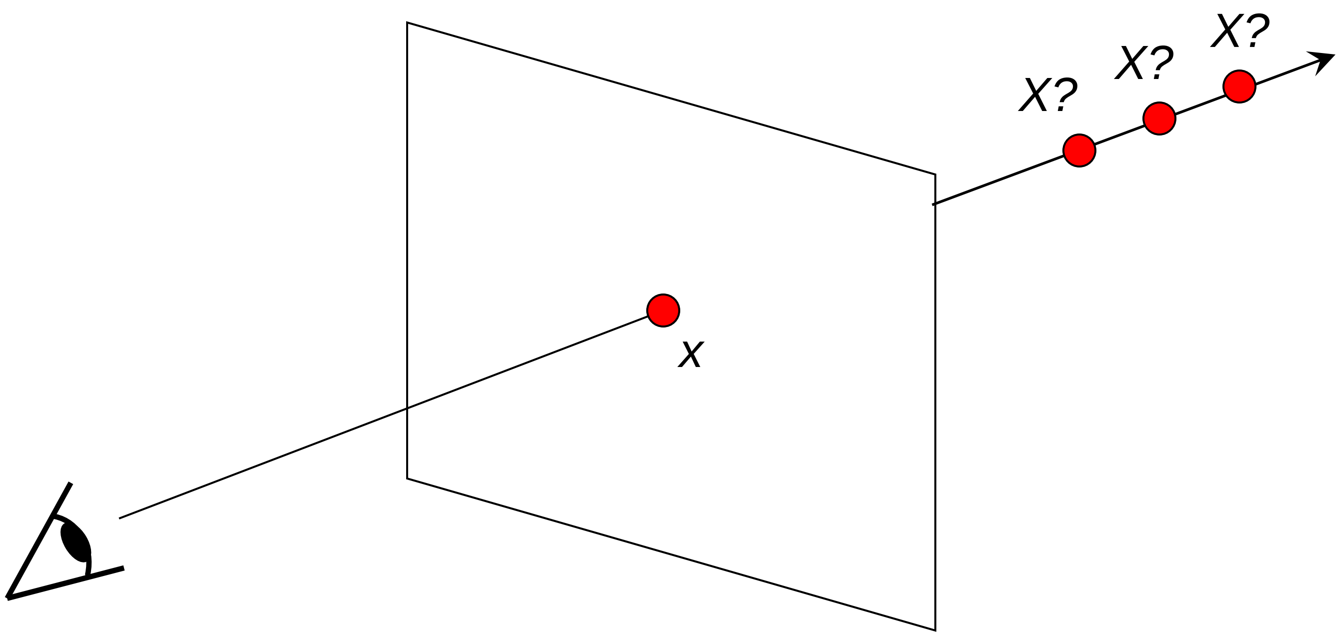
Literature



Multiple View Geometry in Computer Vision, Second Edition, Richard Hartley and Andrew Zisserman, Cambridge University Press, March 2004.

Our goal: Recovery of 3D structure

Recovery of structure from one image is inherently ambiguous



Lazebnik

Our goal: Recovery of 3D structure

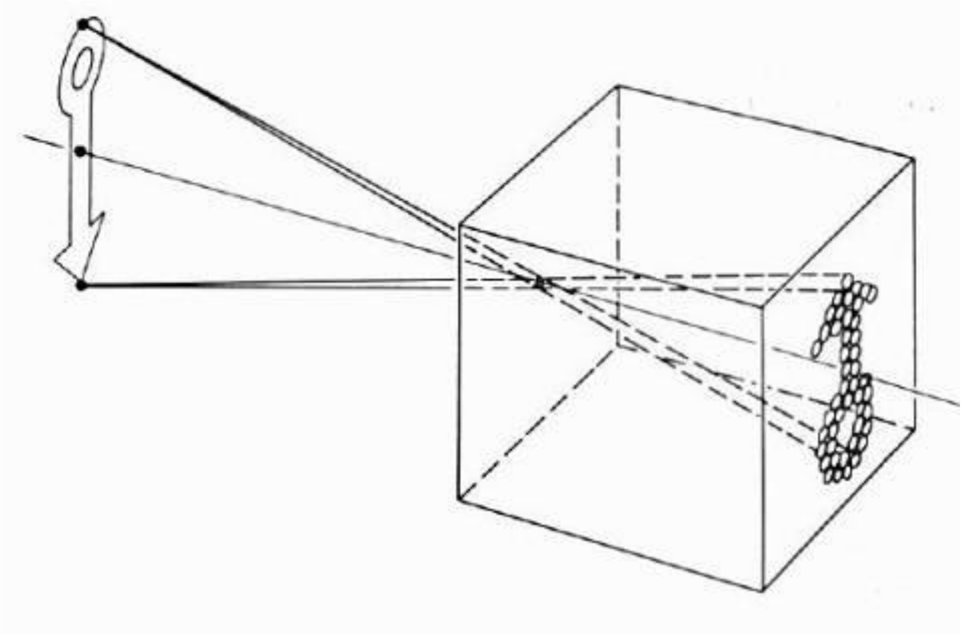
We will need multi-view geometry

Lazebnik

Cameras

Lana Lazebnik

Pinhole camera model



Pinhole model:

- Captures pencil of rays – all rays through a single point
- The point is called Center of Projection (focal point)
- The image is formed on the Image Plane

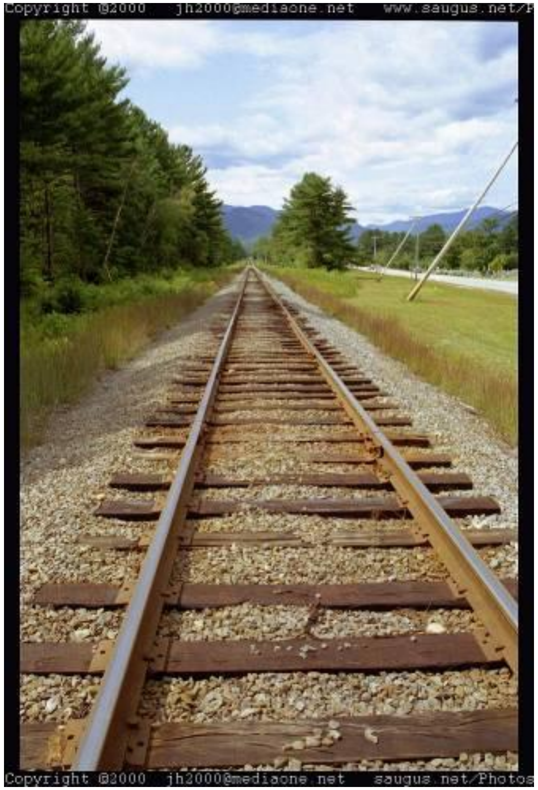
Steve Seitz

Vanishing points

Each direction in space has its own vanishing point

– All lines going in that direction converge at that point

– Exception: directions parallel to the image plane



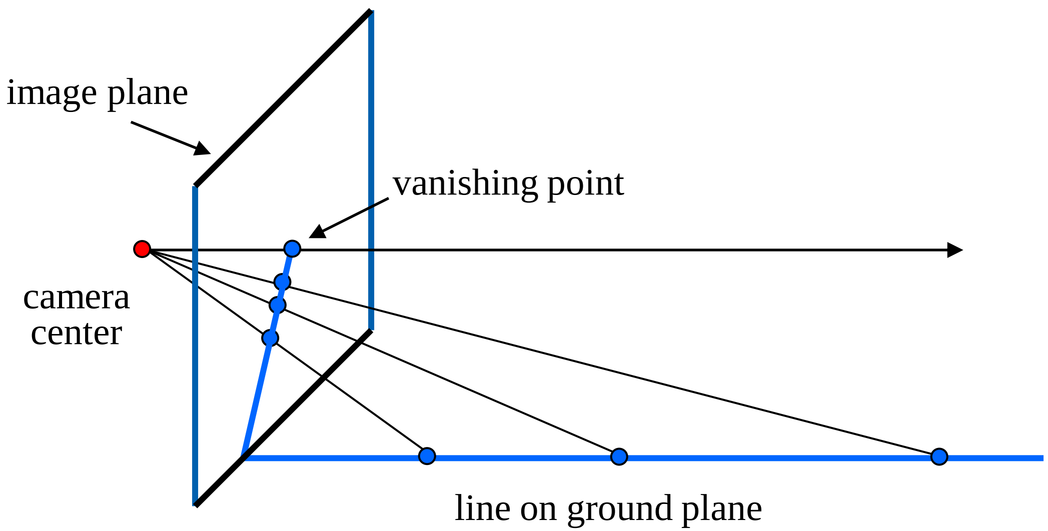
Lana Lazebnik

Vanishing points

Each direction in space has its own vanishing point

- All lines going in that direction converge at that point
- Exception: directions parallel to the image plane

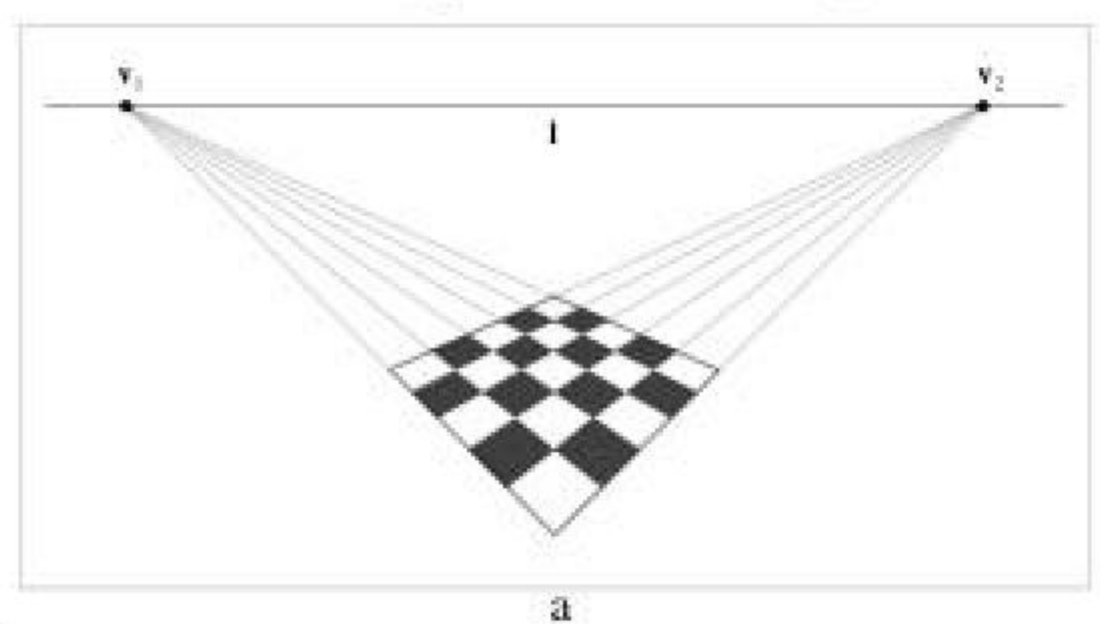
How do we construct the vanishing point of a line?



Lana Lazebnik

Vanishing points

- Each direction in space has its own vanishing point
  - All lines going in that direction converge at that point
  - Exception: directions parallel to the image plane
- How do we construct the vanishing point of a line?
  - What about the vanishing line of a plane?



Lana Lazebnik

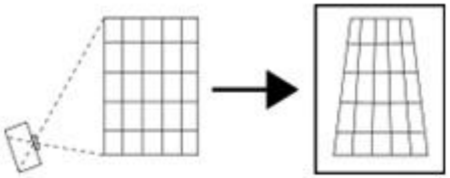
Perspective distortion

Problem for architectural photography: converging verticals

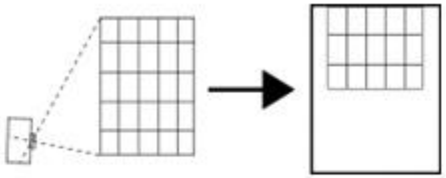
F. Durand

Perspective distortion

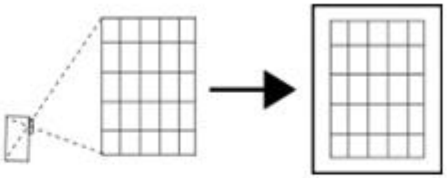
Problem for architectural photography: converging verticals



Tilting the camera upwards results in converging verticals

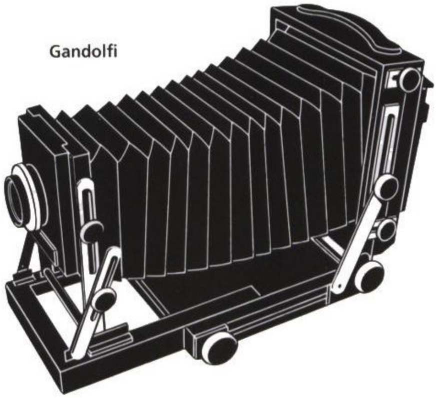


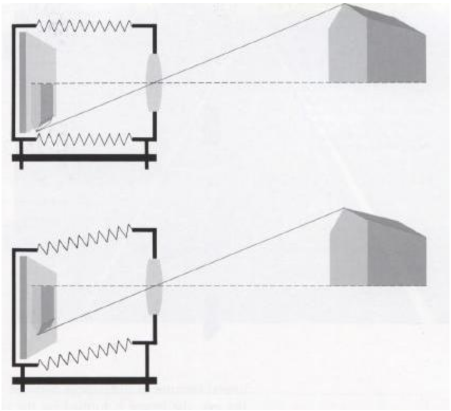
Keeping the camera level, with an ordinary lens, captures only the bottom portion of the building



Shifting the lens upwards results in a picture of the entire subject

Solution: view camera (lens shifted w.r.t. film)





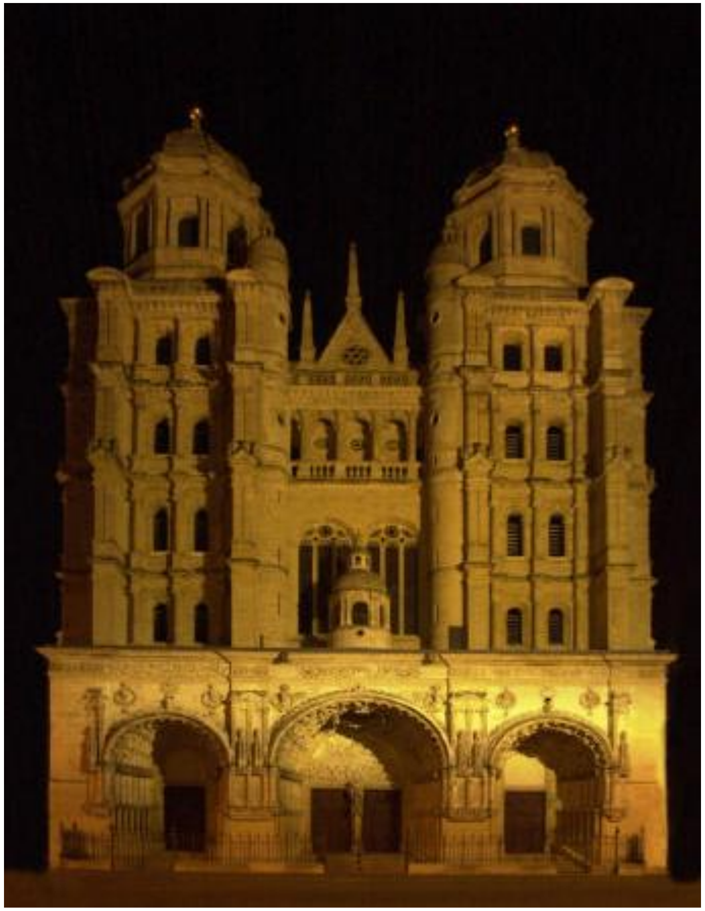
http://en.wikipedia.org/wiki/Perspective_correction_lens

F. Durand

Perspective distortion

Problem for architectural photography: converging verticals

Result:

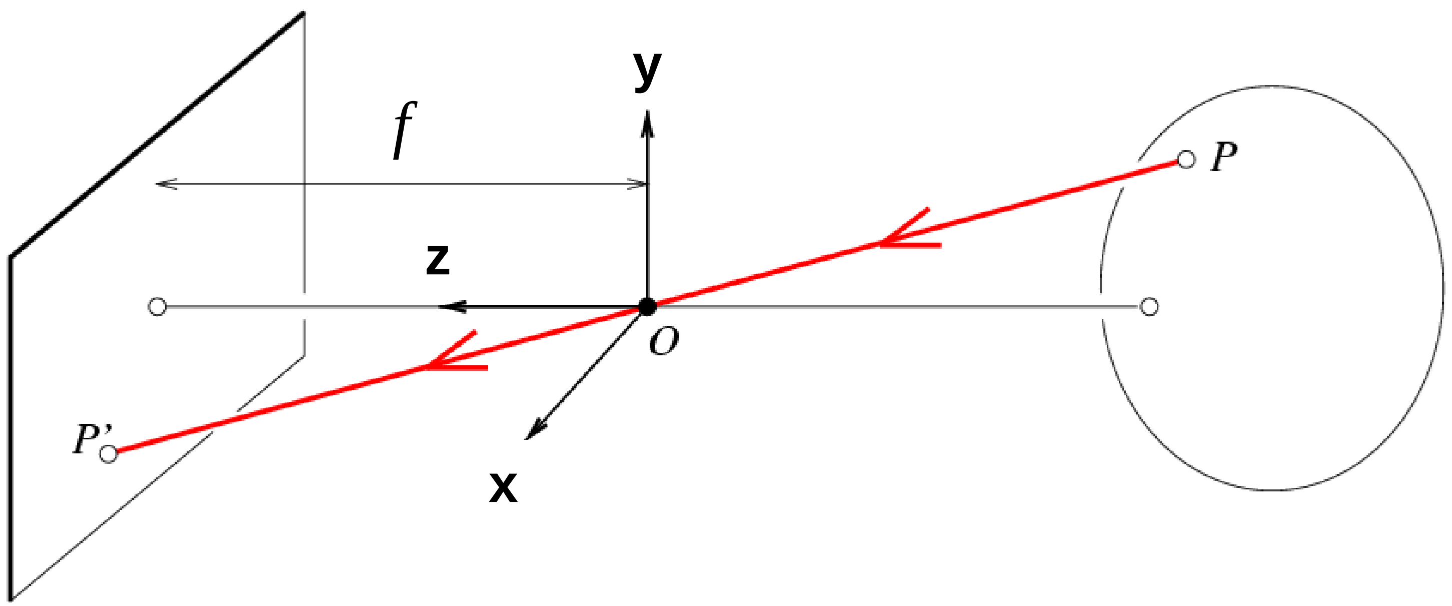


F. Durand

Modeling projection

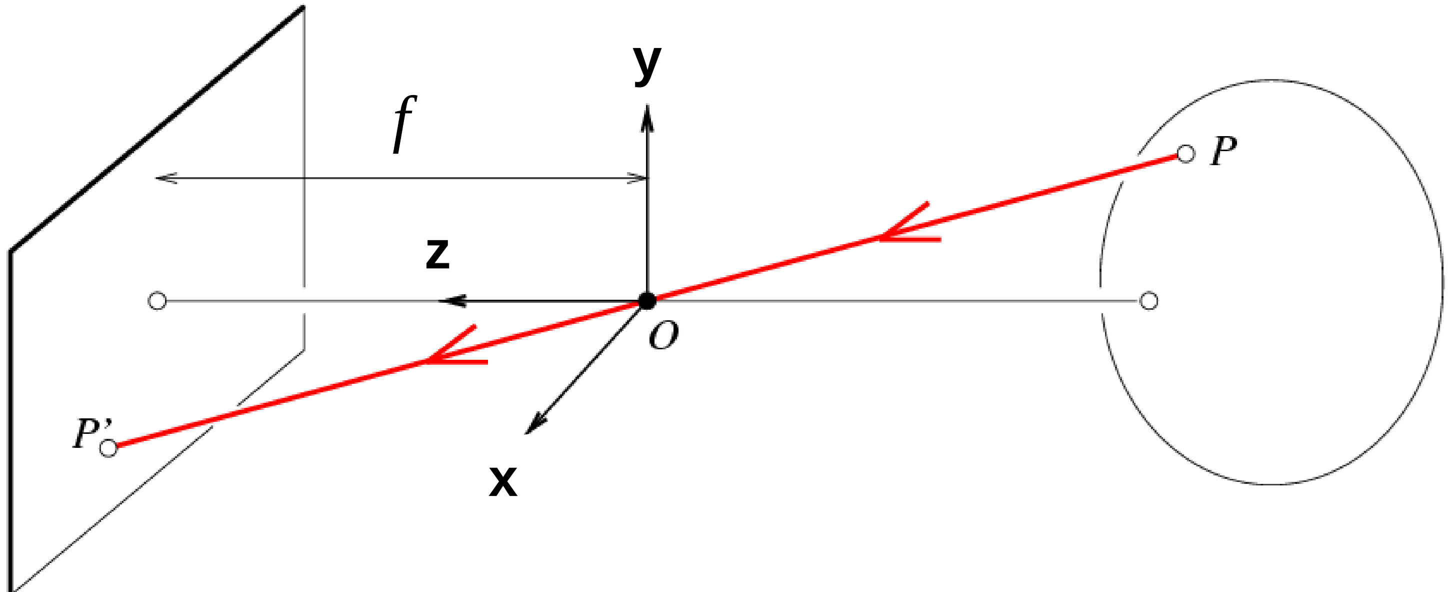
The coordinate system

- The optical center (O) is at the origin
- The image plane is parallel to xy-plane (perpendicular to z axis)



J. Ponce, S. Seitz

Modeling projection



- Projection equations
  - Compute intersection with image plane of ray from ***P*** = (x,y,z) to ***O***
  - Derived using similar triangles

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

  - We get the projection by throwing out the last coordinate:

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

J. Ponce, S. Seitz

Homogeneous coordinates

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

Is this a linear transformation?

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

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

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

Steve Seitz

Perspective Projection Matrix

Projection is a matrix multiplication using homogeneous coordinates

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z/f \end{bmatrix} \Rightarrow \left(f\frac{x}{z}, f\frac{y}{z}\right)$$

divide by the third coordinate

Lana Lazebnik

Perspective Projection Matrix

Projection is a matrix multiplication using homogeneous coordinates

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z/f \end{bmatrix} \Rightarrow \left(f\frac{x}{z}, f\frac{y}{z} \right)$$

divide by the third coordinate

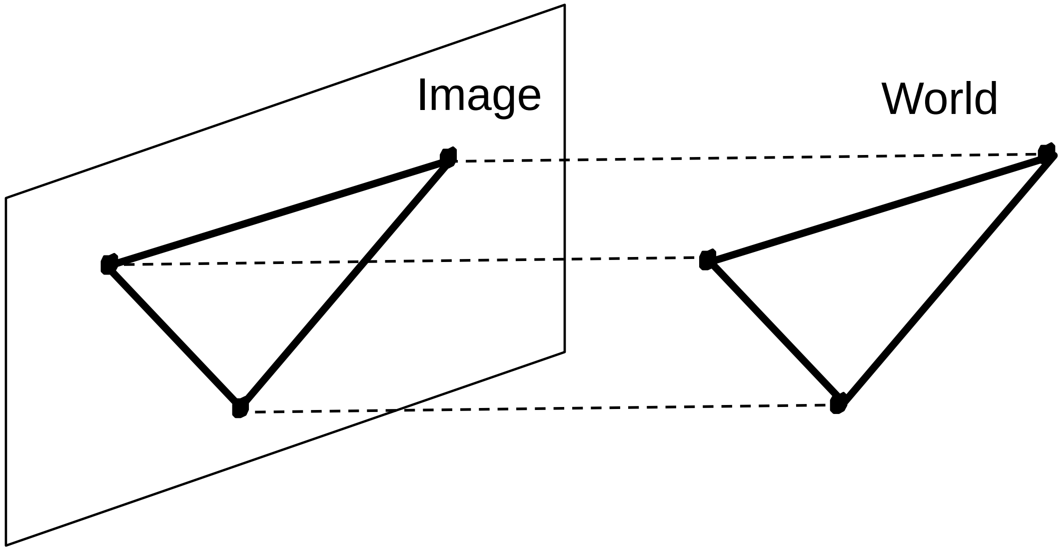
In practice: lots of coordinate transformations…

$$\begin{pmatrix} \text{2D} \\ \text{point} \\ \text{(3x1)} \end{pmatrix} = \begin{pmatrix} \text{Camera to} \\ \text{pixel coord.} \\ \text{trans. matrix} \\ \text{(3x3)} \end{pmatrix} \begin{pmatrix} \text{Perspective} \\ \text{projection matrix} \\ \text{(3x4)} \end{pmatrix} \begin{pmatrix} \text{World to} \\ \text{camera coord.} \\ \text{trans. matrix} \\ \text{(4x4)} \end{pmatrix} \begin{pmatrix} \text{3D} \\ \text{point} \\ \text{(4x1)} \end{pmatrix}$$

Lana Lazebnik

Orthographic Projection

- Distance from center of projection to image plane is infinite



- Also called “parallel projection”
- What’s the projection matrix?

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \Rightarrow (x, y)$$

Steve Seitz

Building a real camera

Lana Lazebnik

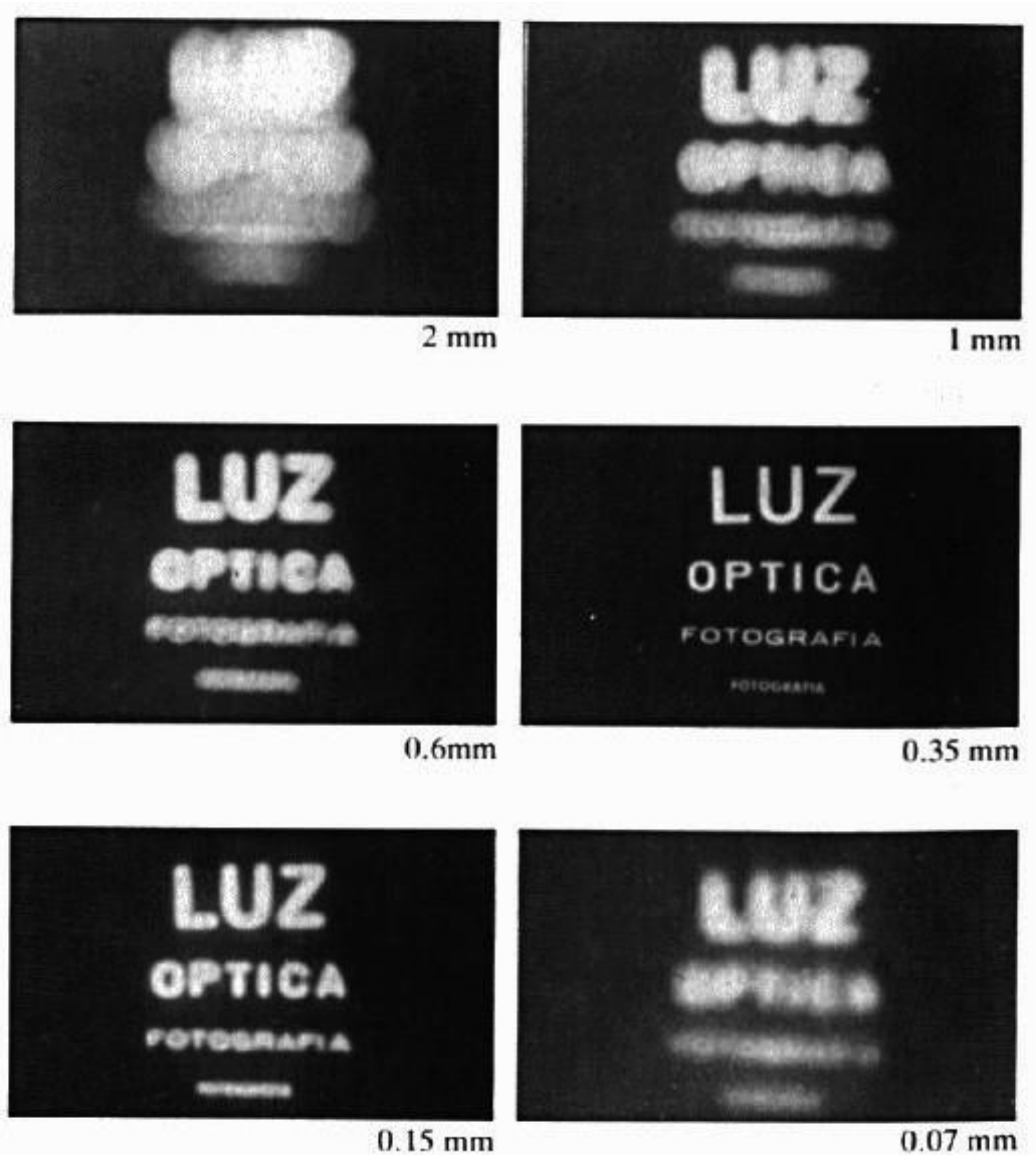
Home-made pinhole camera

Why so blurry?

A. Efros

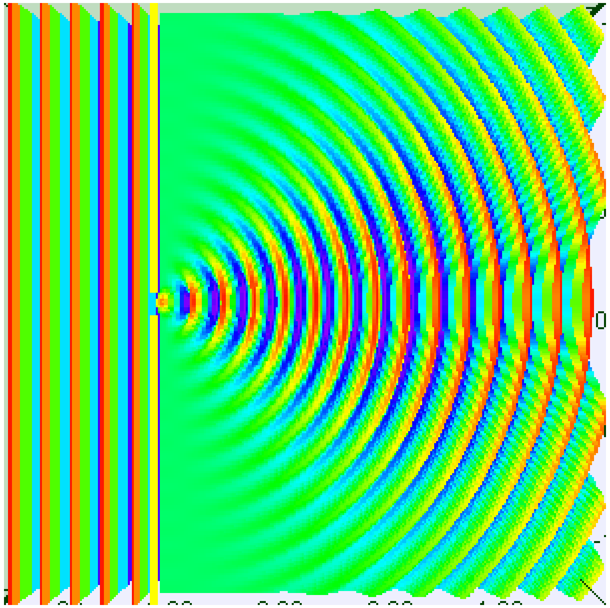
http://www.debevec.org/Pinhole/

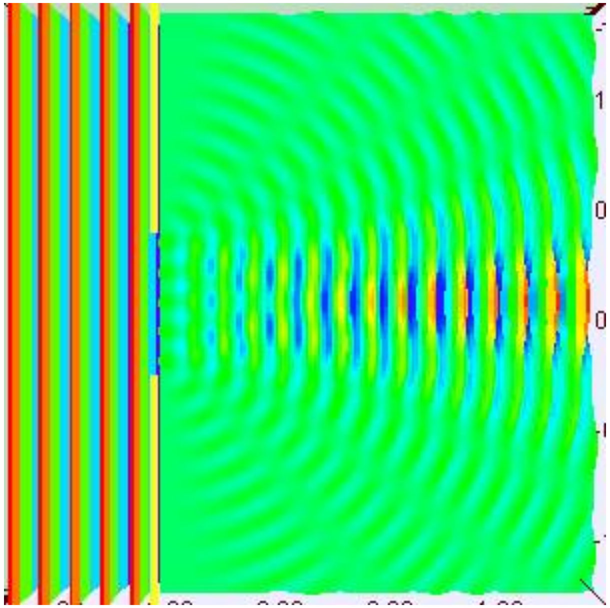
Shrinking the aperture



Lana Lazebnik

Shrinking apperture



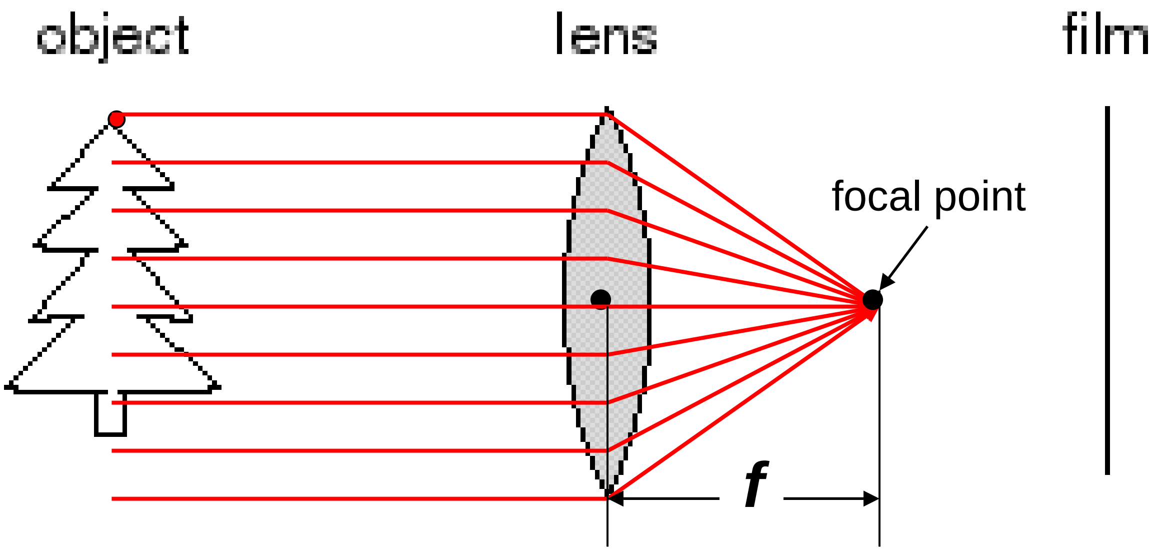


Adding a lens

A lens focuses light onto the film

– Thin lens model:

- Rays passing through the center are not deviated (pinhole projection model still holds)
- All parallel rays converge to one point on a plane located at the focal length f

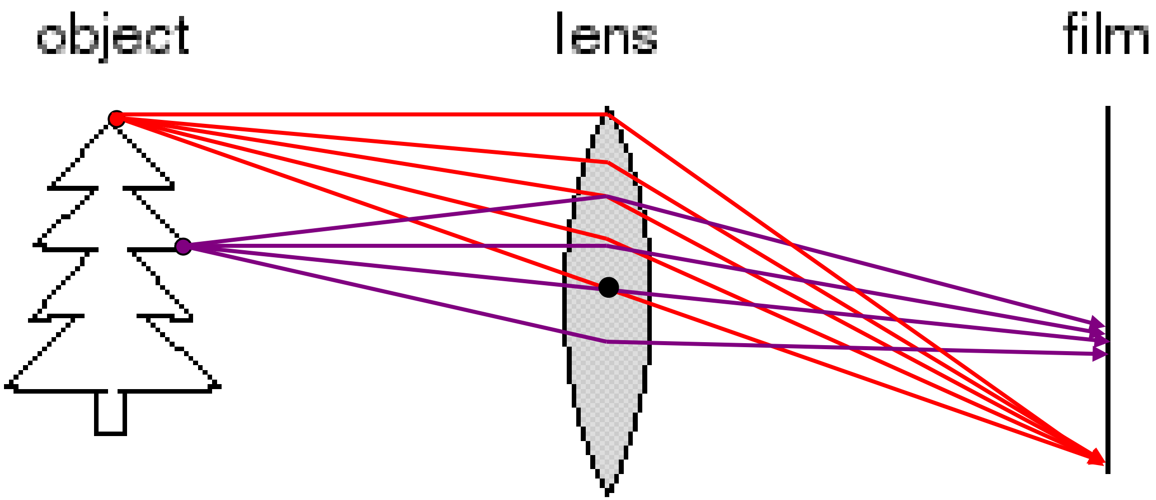


Steve Seitz

Adding a lens

A lens focuses light onto the film

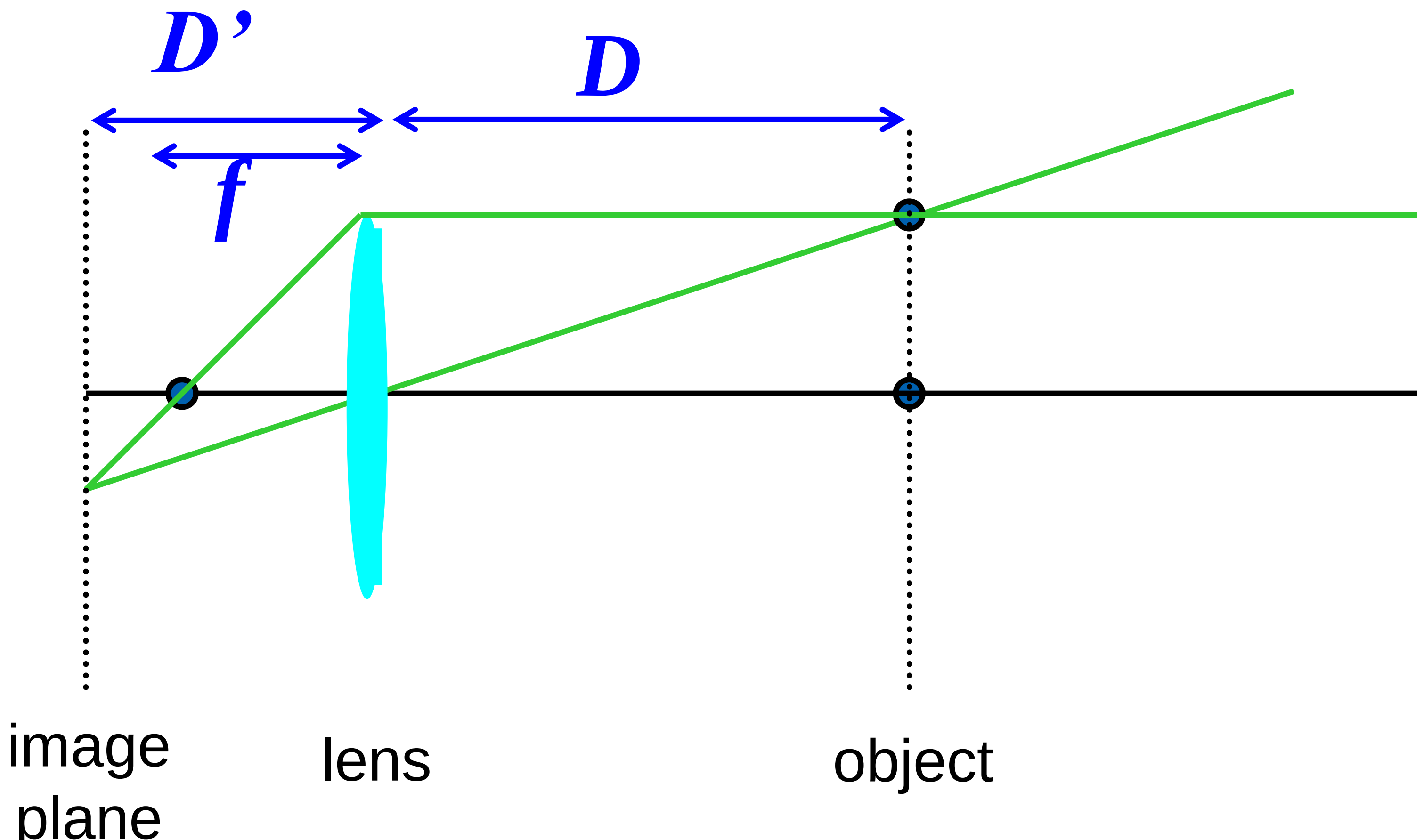
– There is a specific distance at which objects are “in focus”



Steve Seitz

Thin lens formula

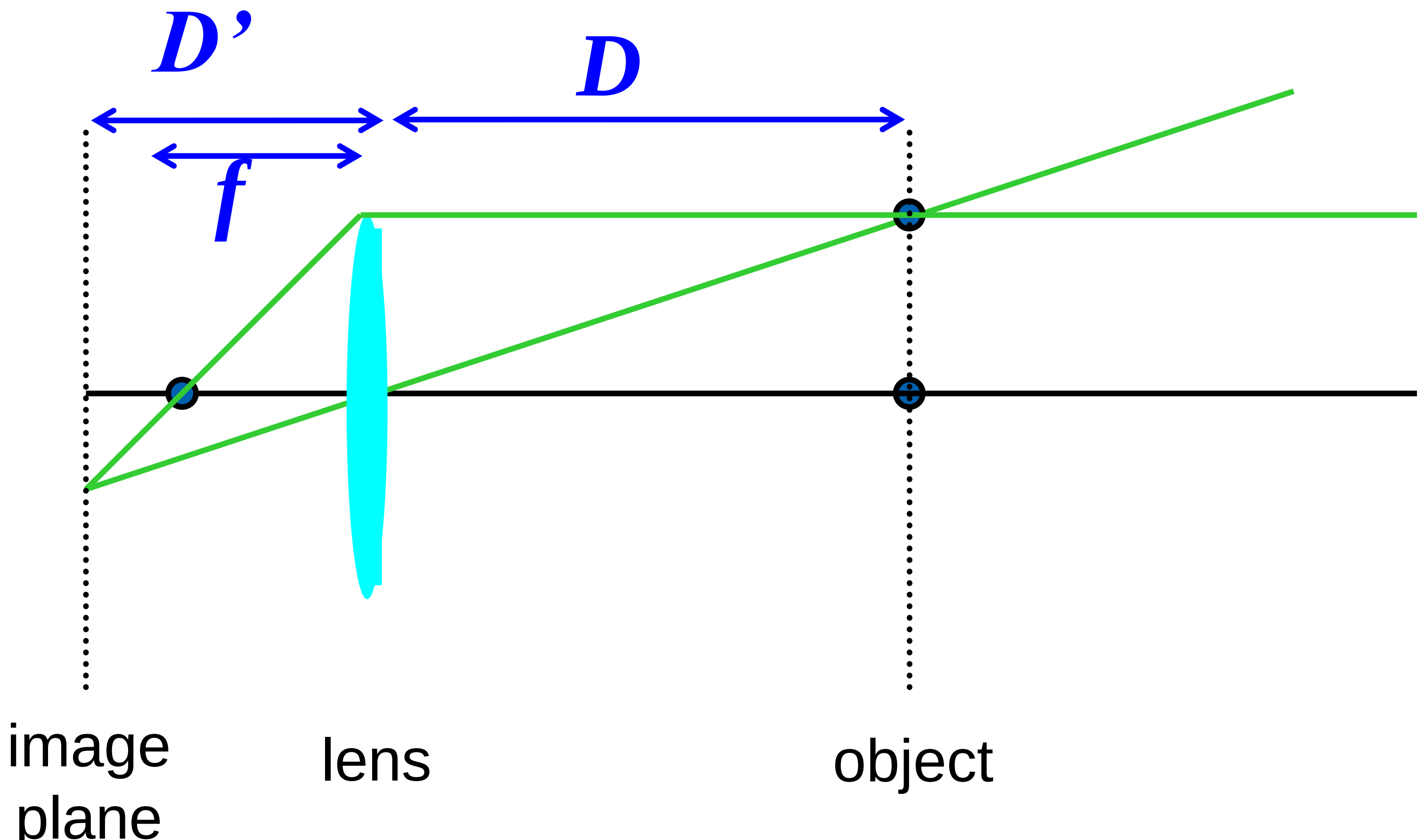
What is the relation between the focal length (f),
the distance of the object from the optical center (D),
and the distance at which the object will be in focus (D')?



Frédo Durand

Thin lens formula

Similar triangles everywhere!

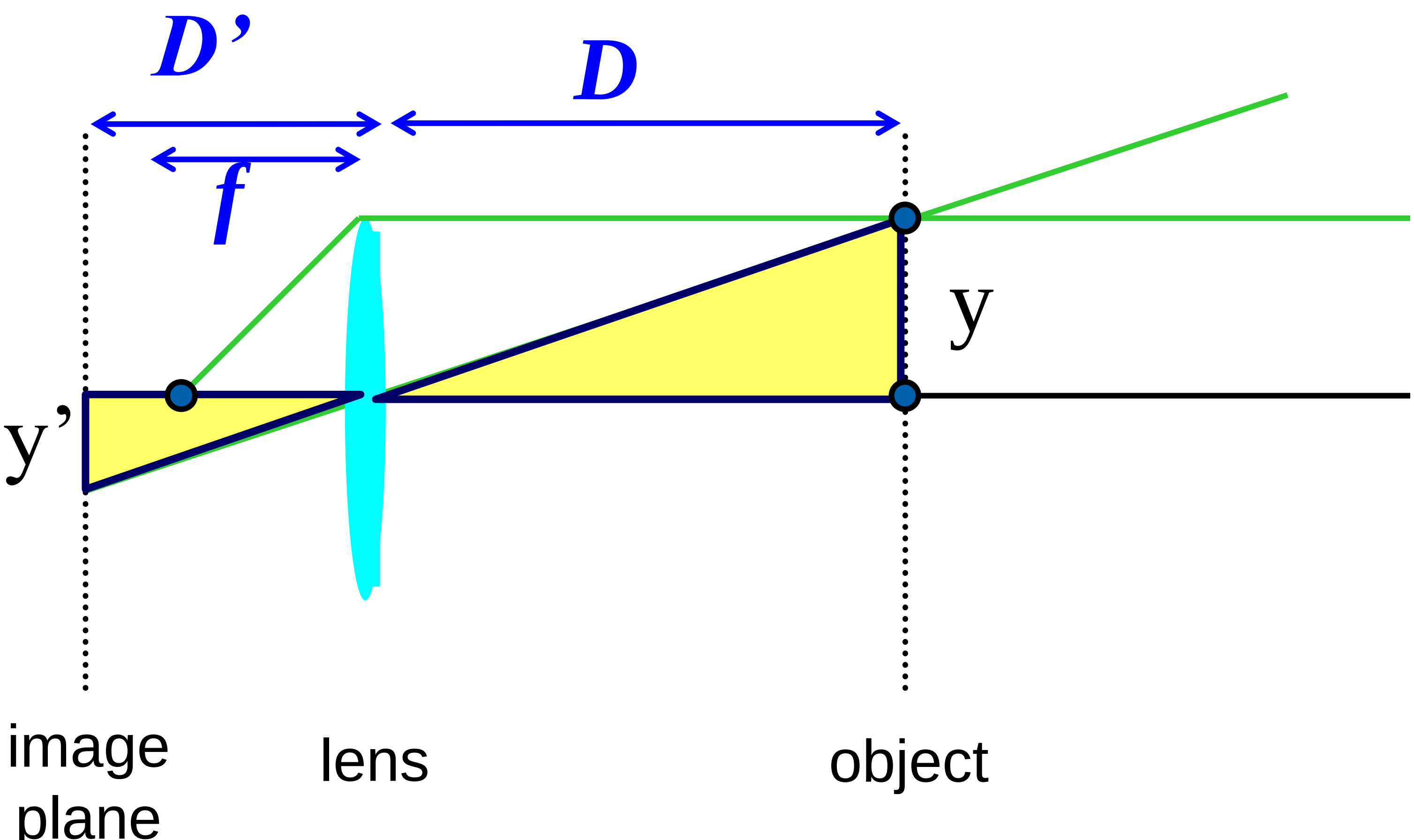


Frédo Durand

Thin lens formula

Similar triangles everywhere!

$$y'/y = D'/D$$



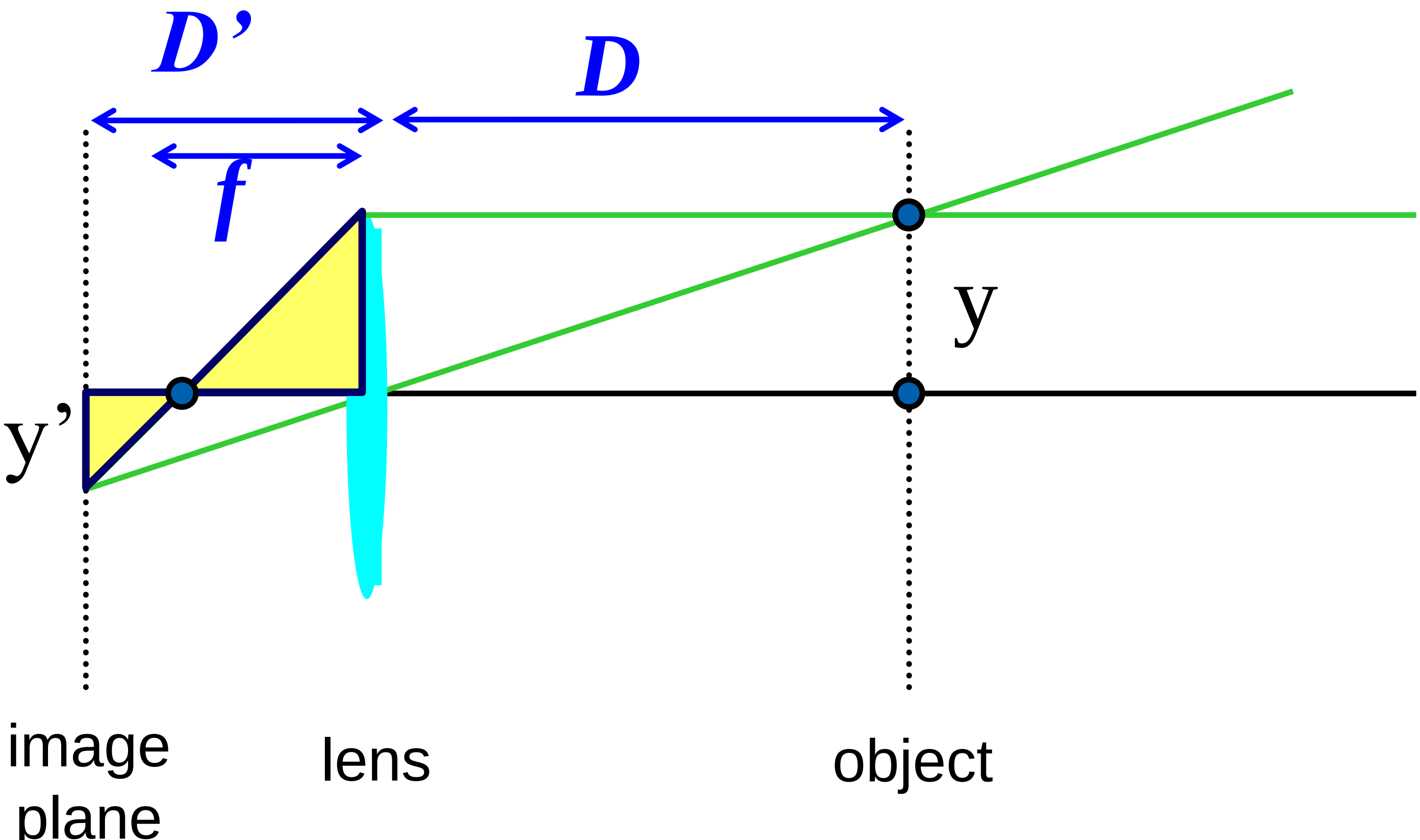
Frédo Durand

Thin lens formula

Similar triangles everywhere!

$$y'/y = D'/D$$

$$y'/y = (D'-f)/f$$



Frédo Durand

Thin lens formula

Similar triangles everywhere!

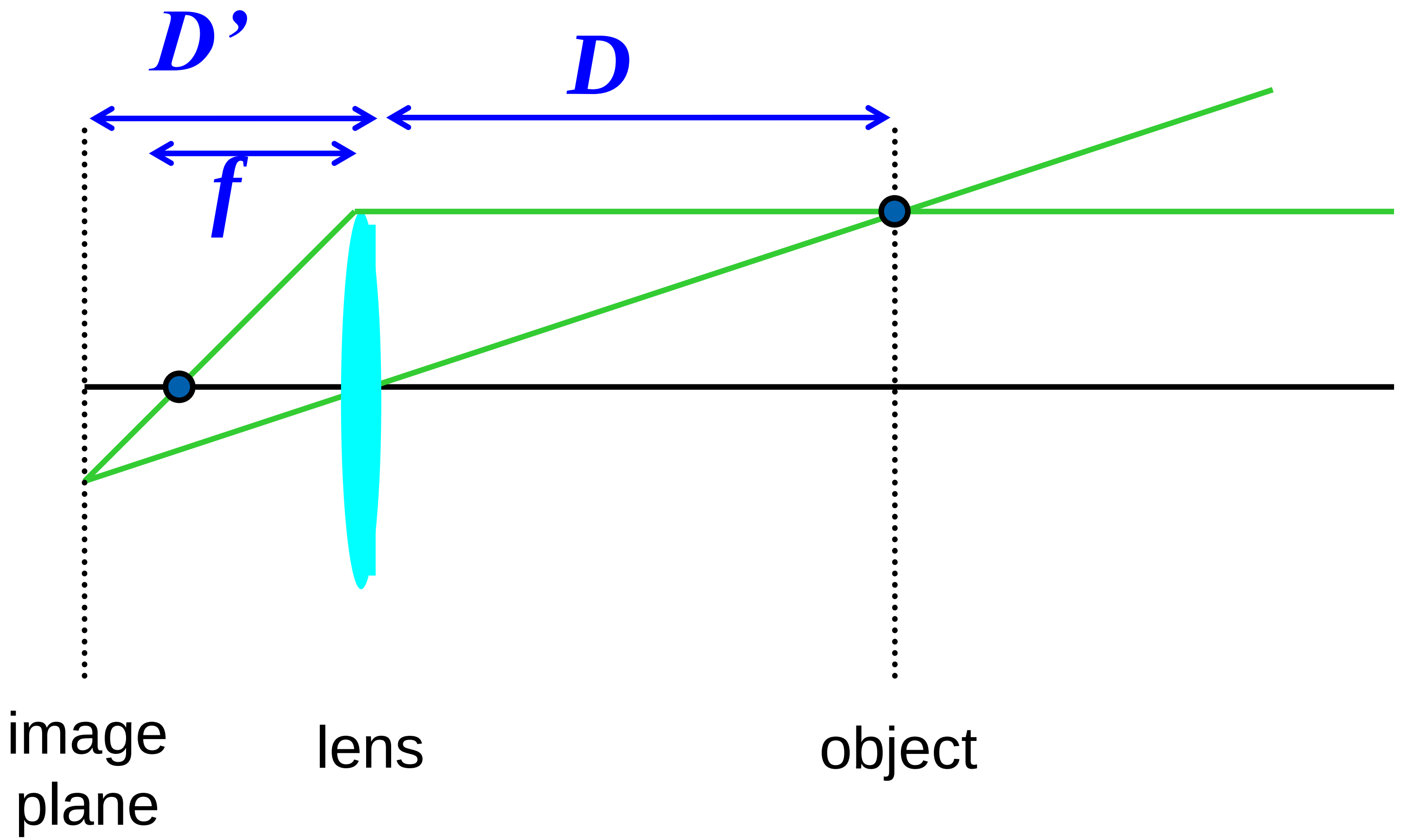
$$y'/y = D'/D$$

$$y'/y = (D'-f)/f$$

Thin lens formula

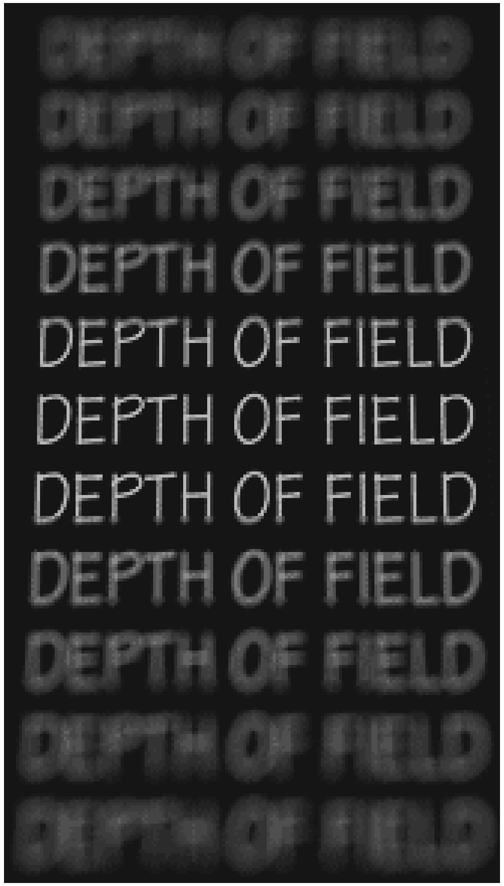
$$\boxed{\frac{1}{D'} + \frac{1}{D} = \frac{1}{f}}$$

Any point satisfying the thin lens equation is in focus.



Frédo Durand

Depth of Field

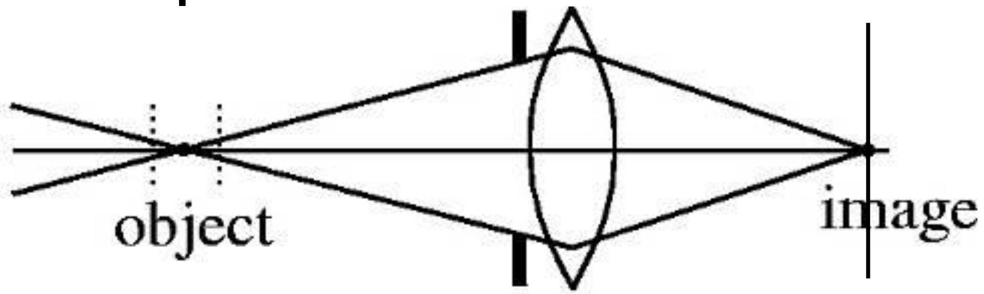


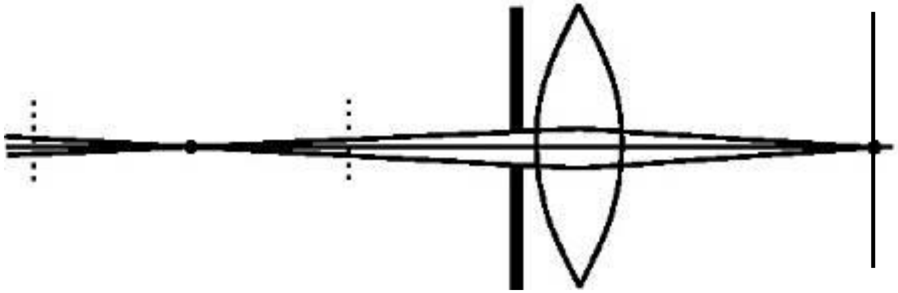
A. Efros

How can we control the depth of field?

Changing the aperture size affects depth of field

- A smaller aperture increases the range in which the object is approximately in focus
- But small aperture reduces amount of light – need to increase exposure





A. Efros

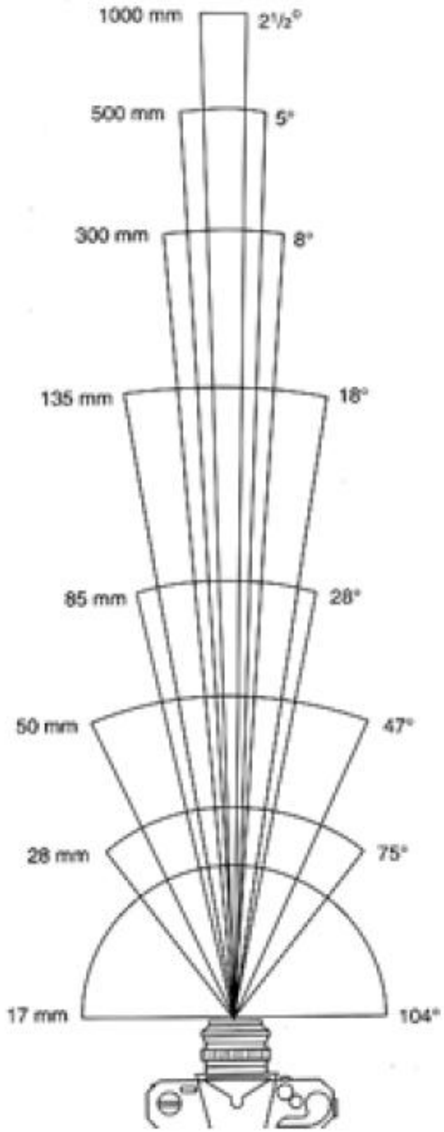
Varying the aperture

Large aperture = small DOF

Small aperture = large DOF

A. Efros

Field of View




17mm

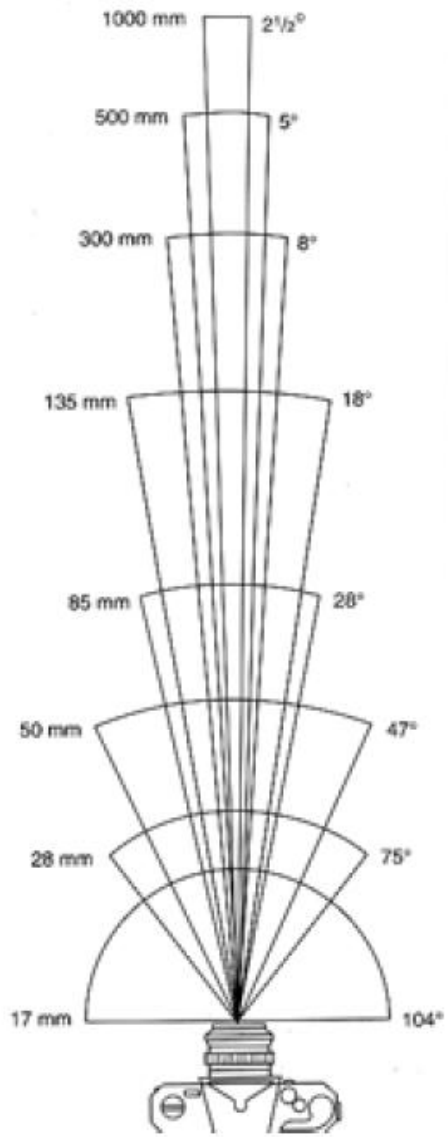

28mm

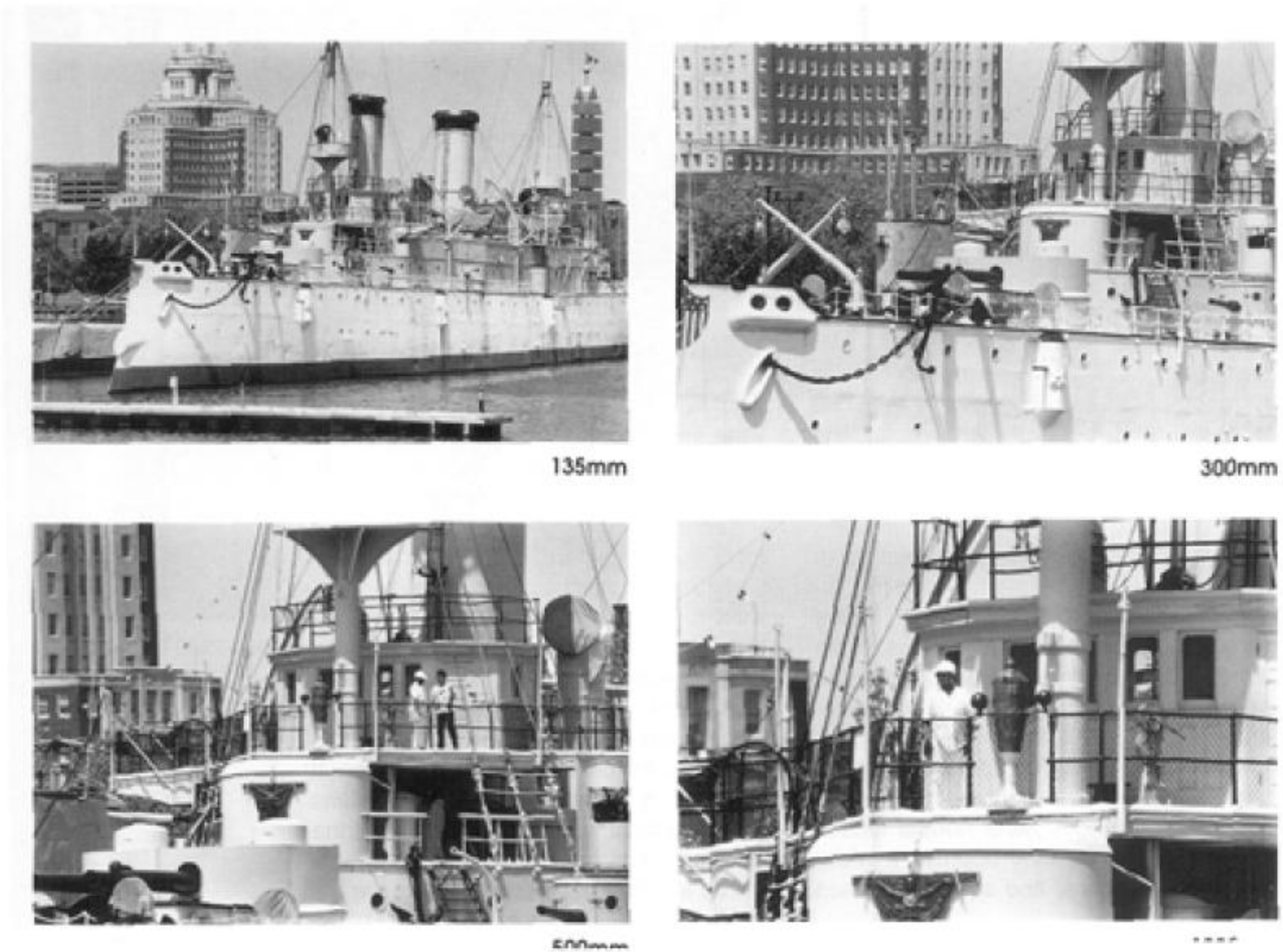

50mm


85mm

A. Efros

Field of View

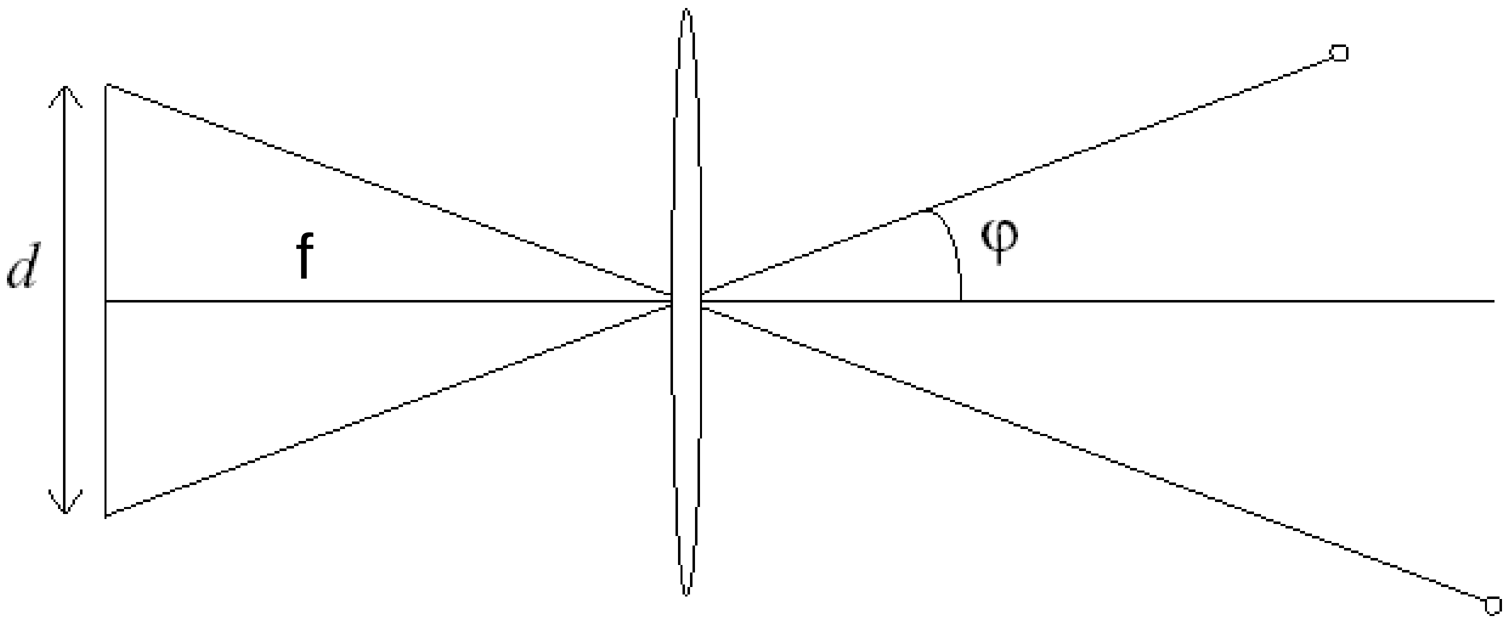




What does FOV depend on?

A. Efros

Field of View



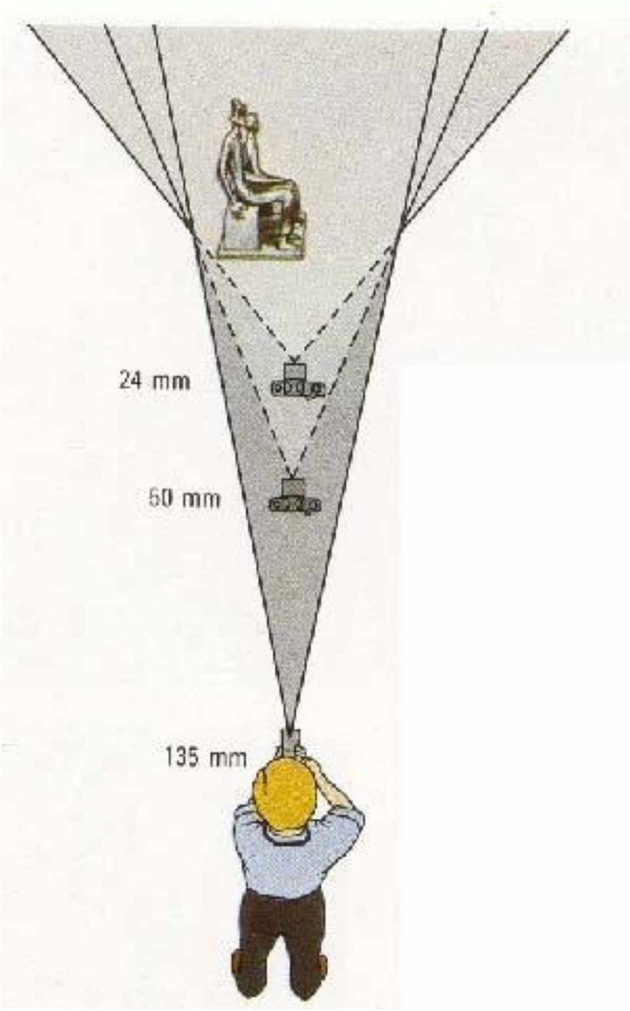
FOV depends on focal length and size of the camera retina

$$\varphi = \tan^{-1}\left(\frac{d}{2f}\right)$$

Smaller FOV = larger Focal Length

A. Efros

Field of View / Focal Length

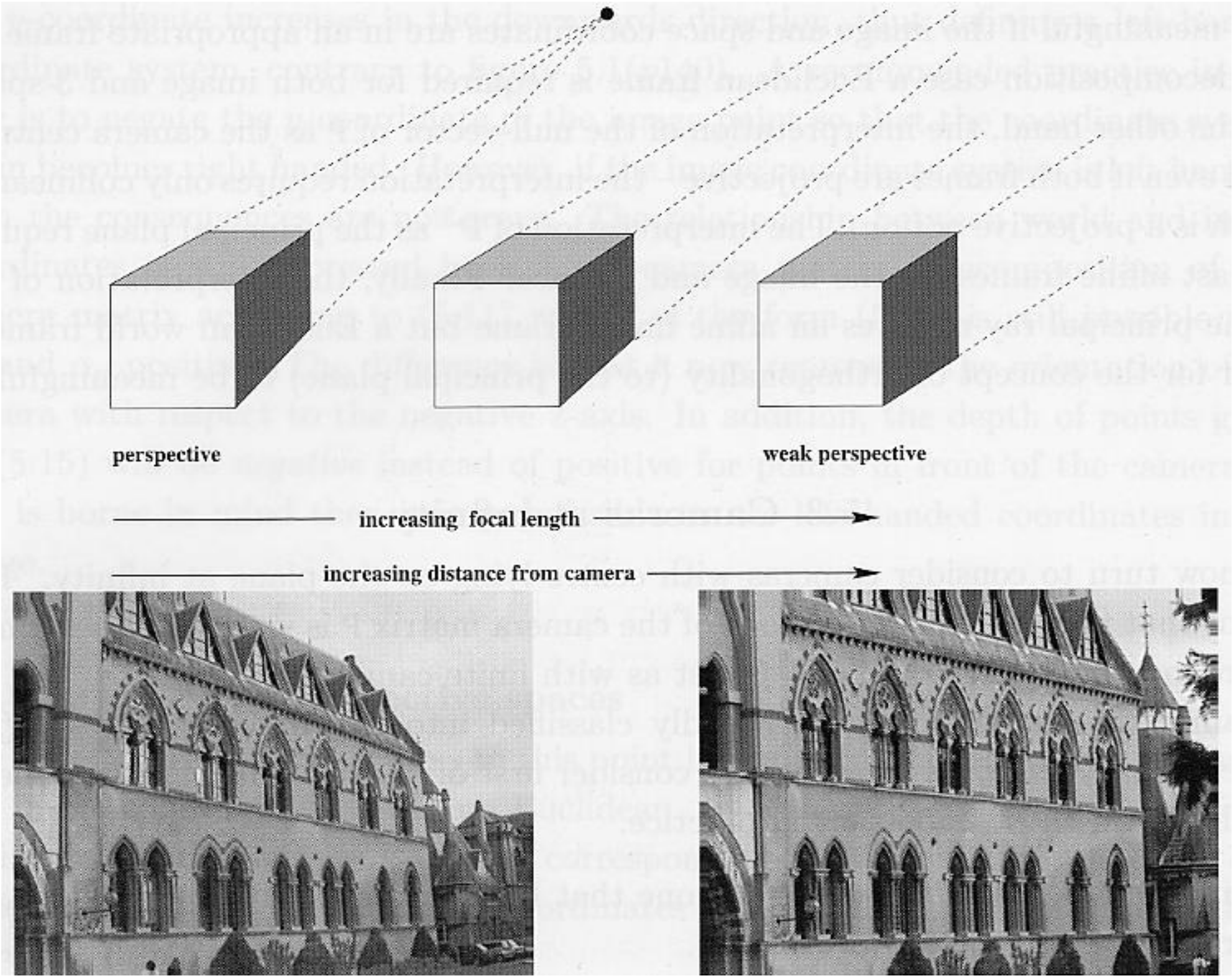


Large FOV, small f
Camera close to car

Small FOV, large f
Camera far from the car

A. Efros, F. Durand

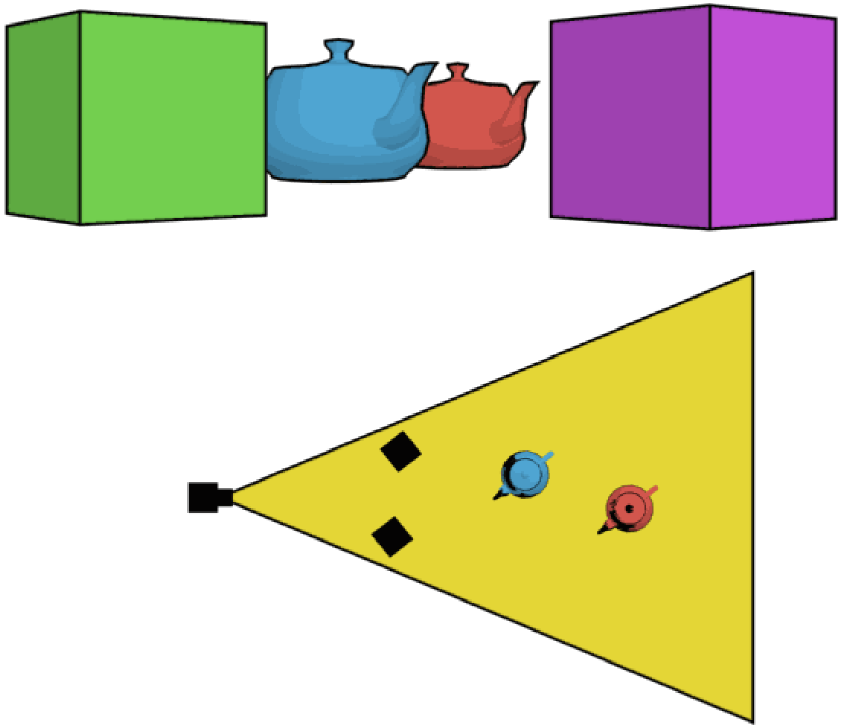
Approximating an affine camera



Hartley & Zisserman

The dolly zoom

Continuously adjusting the focal length while the camera moves away from (or towards) the subject



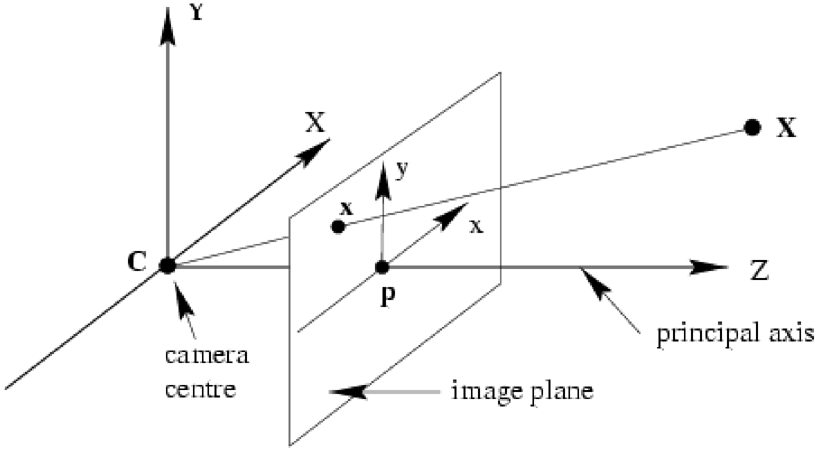
http://en.wikipedia.org/wiki/Dolly_zoom

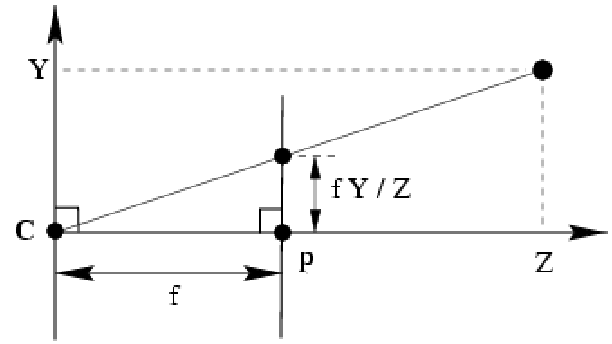
Lana Lazebnik

Dolly zoom

Dolly zoom

Pinhole camera model



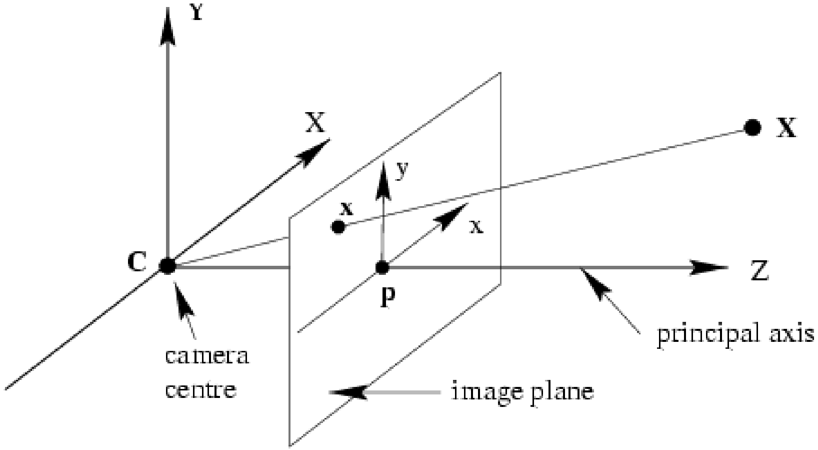


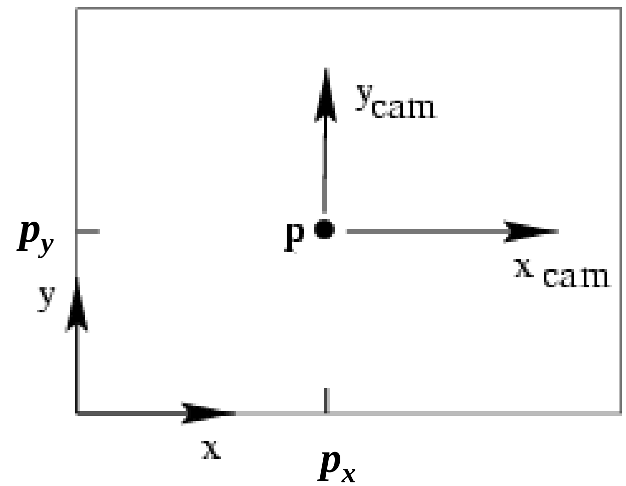
$$(X, Y, Z) \mapsto (f X / Z, f Y / Z)$$

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} f X \\ f Y \\ Z \end{pmatrix} = \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \qquad \mathrm{x} = \mathrm{PX}$$

Lazebnik

Principal point

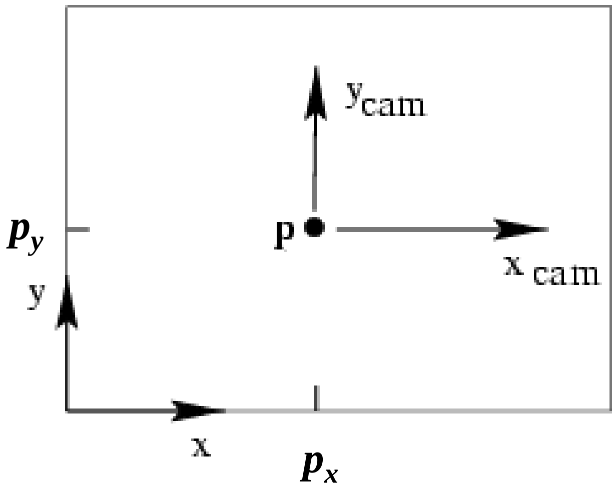




- Principal point (p): point where principal axis intersects the image plane (origin of normalized coordinate system)
- Normalized coordinate system: origin is at the principal point
- Image coordinate system: origin is in the corner
- How to go from normalized coordinate system to image coordinate system?

Lazebnik

Principal point offset



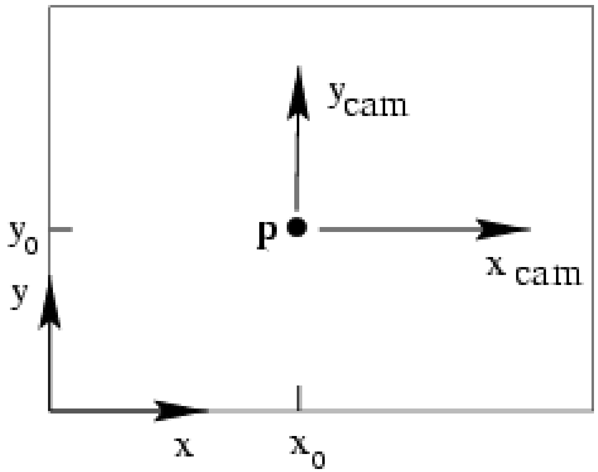
principal point: (p_x, p_y)

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

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

Lazebnik

Principal point offset



principal point: (p_x, p_y)

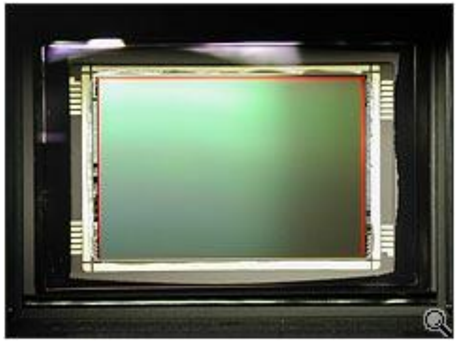
$$\begin{pmatrix} fX + Zp_x \\ fY + Zp_y \\ Z \end{pmatrix} = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & 0 \\ & 1 & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

$$K = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}$$ calibration matrix

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

Lazebnik

Pixel coordinates



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

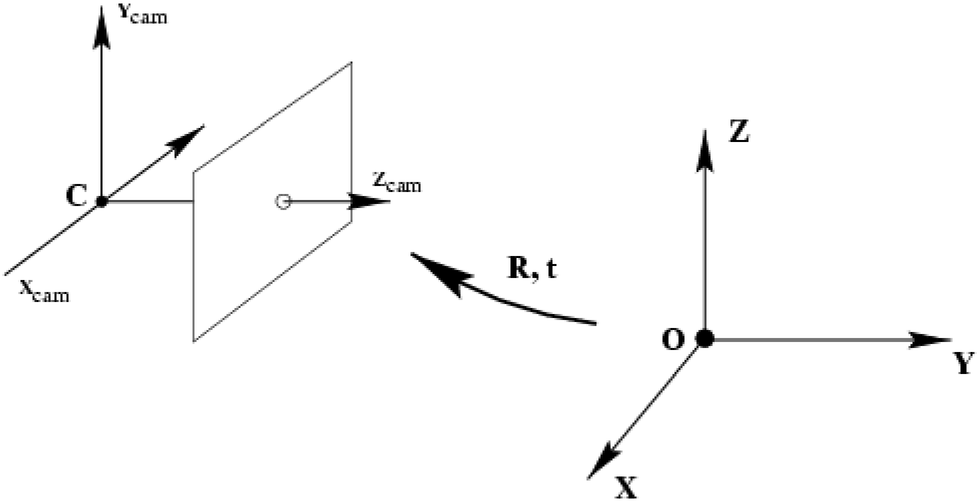
m_x pixels per meter in horizontal direction,
m_y pixels per meter in vertical direction

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

pixels/m　　　m　　　pixels

Lazebnik

Camera rotation and translation



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

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

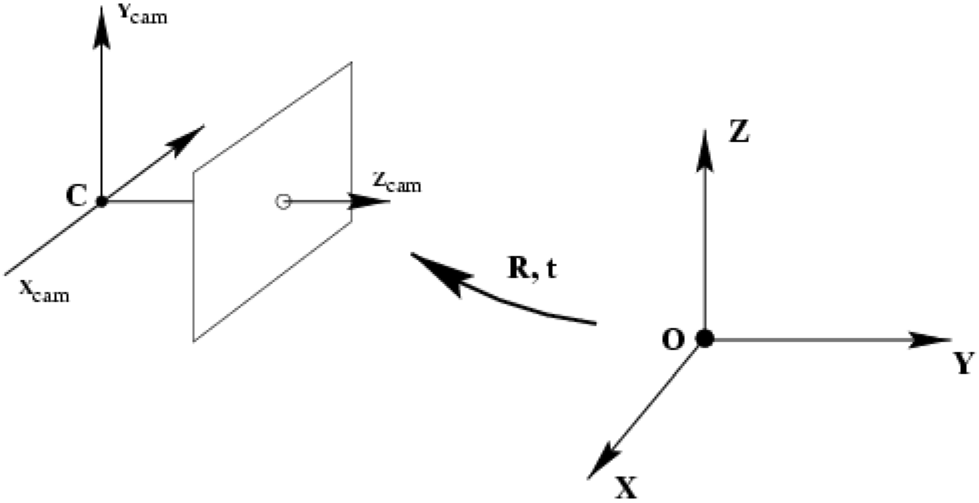
coords. of point in camera frame

coords. of a point in world frame (nonhomogeneous)

coords. of camera center in world frame

Lazebnik

Camera rotation and translation



In non-homogeneous coordinates:

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

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

$$x = K[I \mid 0] X_{cam} = K\left[R \mid -R\tilde{C}\right] X \qquad P = K[R \mid t], \qquad t = -R\tilde{C}$$

Lazebnik

Camera parameters

- Intrinsic parameters
  - Principal point coordinates
  - Focal length
  - Pixel magnification factors

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

Lazebnik

Camera parameters

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

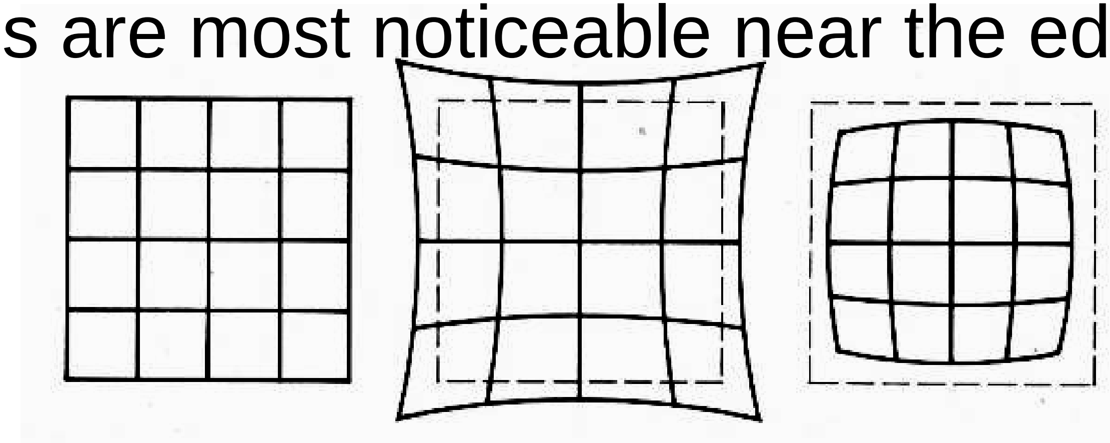
$$K = \begin{pmatrix} \alpha_x & \gamma & \beta_x \\ 0 & \alpha_y & \beta_y \\ 0 & 0 & 1 \end{pmatrix}$$

Lazebnik

Radial Distortion

Caused by imperfect lenses

Deviations are most noticeable near the edge of the lens



No distortion Pin cushion Barrel

Lana Lazebnik

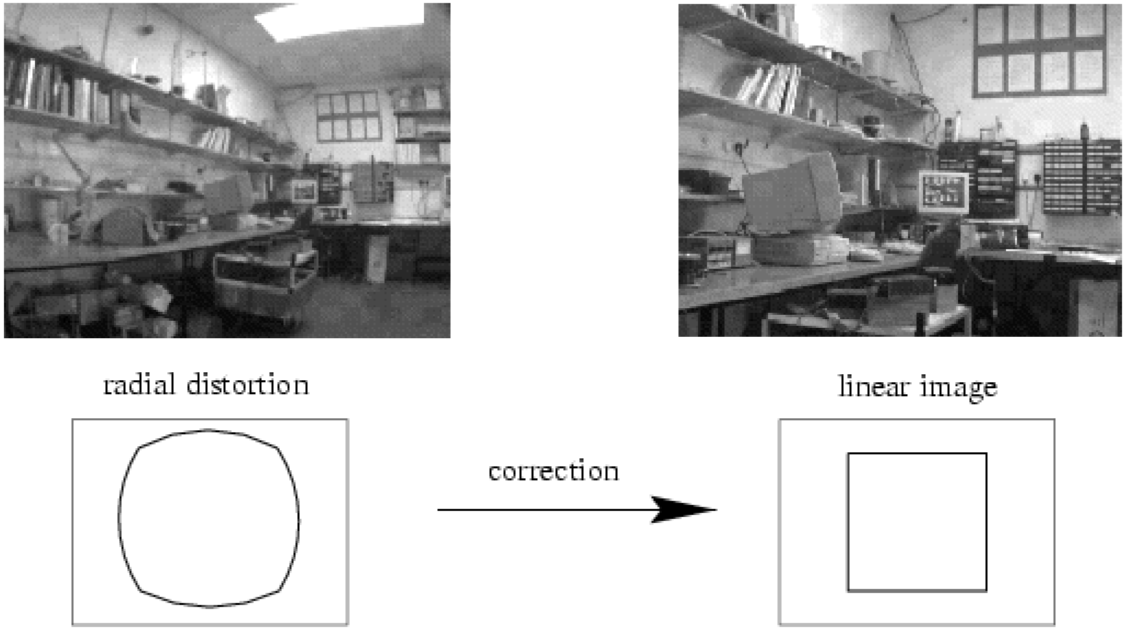
Fisheye lenses

Fisheye photo © 2006 Jarle Aasland
Nikkor 6mm mounted on Nikon F3 body © 2006 Kazuo Koga

Camera parameters

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

$$K = \begin{pmatrix} \alpha_x & \gamma & \beta_x \\ 0 & \alpha_y & \beta_y \\ 0 & 0 & 1 \end{pmatrix}$$



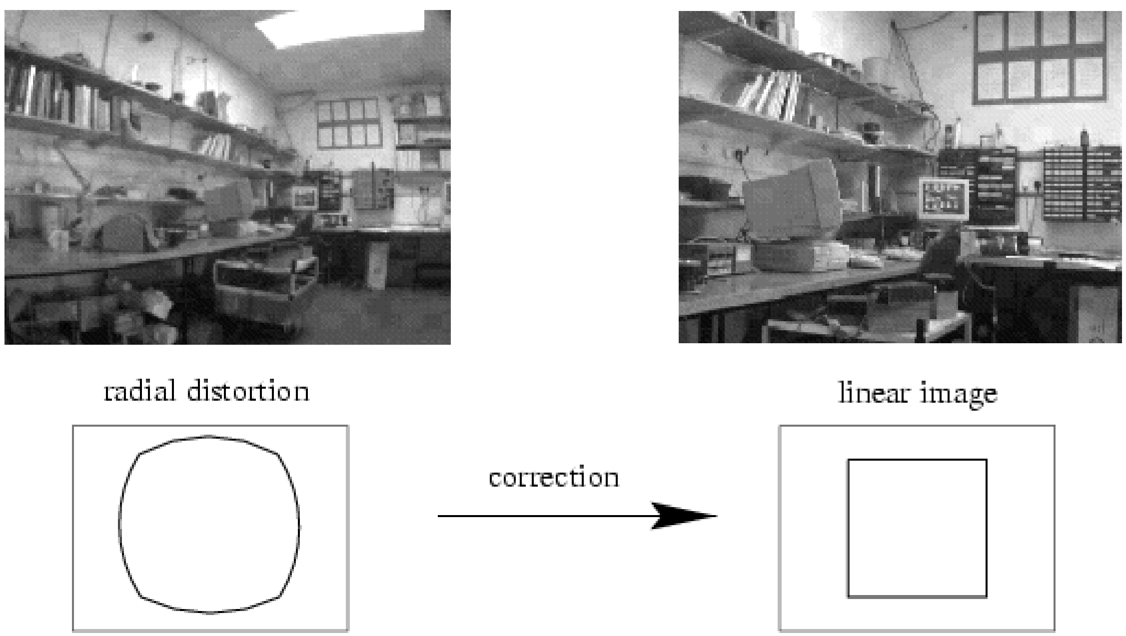
Lazebnik

Radial distortion

Polynomial model

$$\begin{pmatrix} x_d \\ y_d \end{pmatrix} = L(\tilde{r}) \begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix}$$

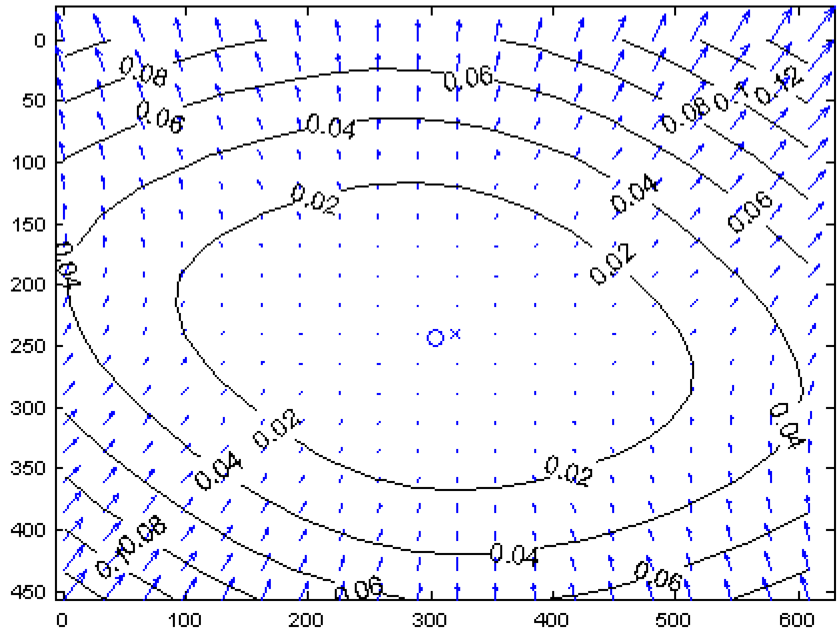
$$L(r) = 1 + \kappa_1 r + \kappa_2 r^2 + \kappa_3 r^3 + \ldots$$

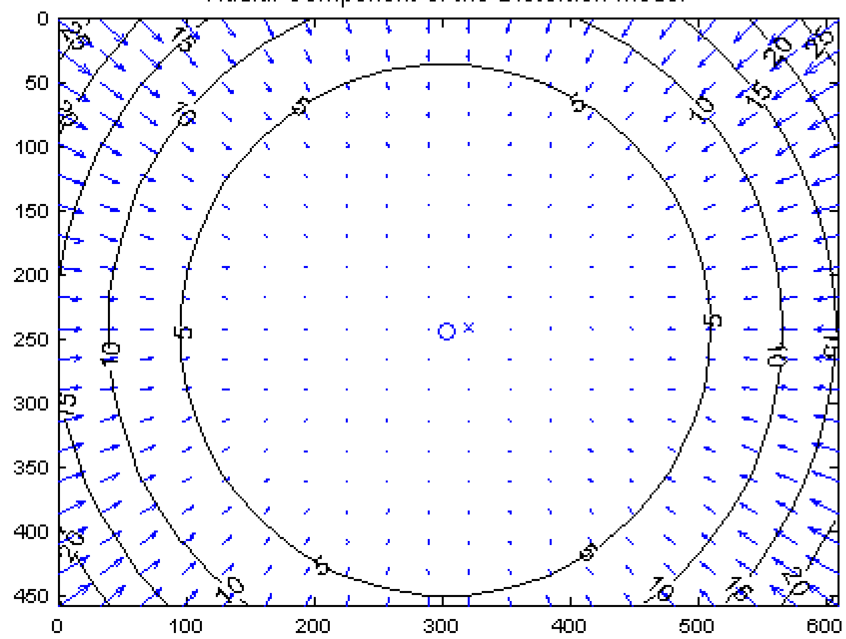


Distortion

UNIVERSITÄT BONN

Other models combine radial and tangential distortion





$$x_d = \begin{bmatrix} x_d(1) \\ x_d(2) \end{bmatrix} = \left(1 + kc(1)\, r^2 + kc(2)\, r^4 + kc(5)\, r^6\right) x_n + dx$$

$$dx = \begin{bmatrix} 2\, kc(3)\, x\, y + kc(4)\left(r^2 + 2x^2\right) \\ kc(3)\left(r^2 + 2y^2\right) + 2\, kc(4)\, x\, y \end{bmatrix}$$

Camera parameters

UNIVERSITÄT BONN

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

Lazebnik

Camera calibration

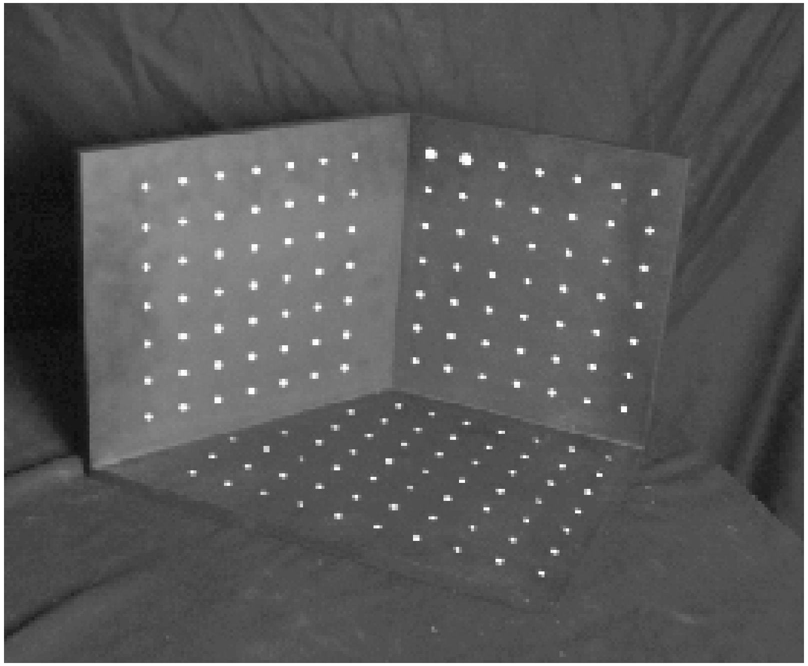
$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

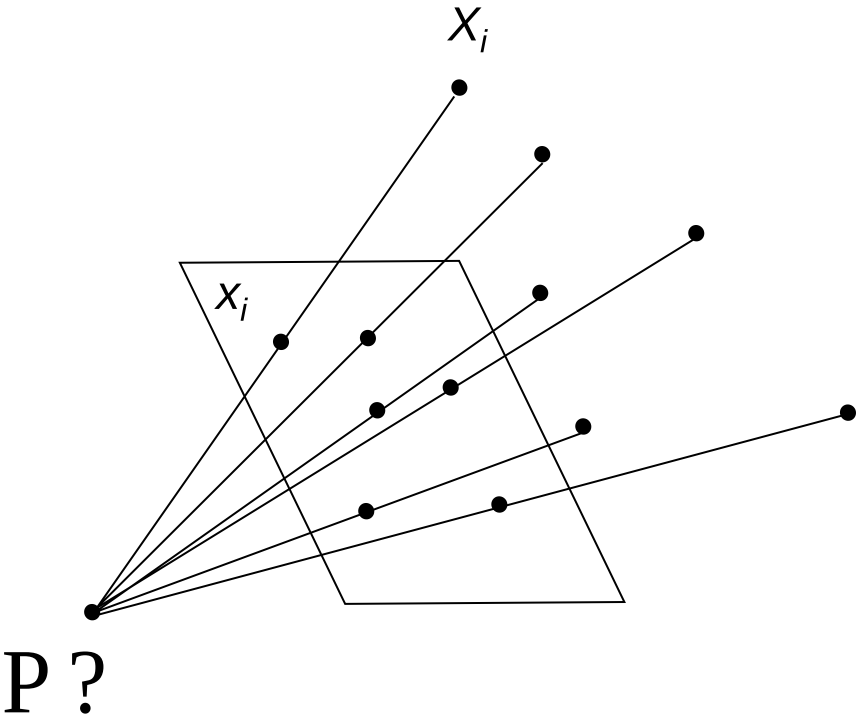
$$\begin{bmatrix}\lambda x \\ \lambda y \\ \lambda\end{bmatrix} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1\end{bmatrix}$$

Source: D. Hoiem

Camera calibration

Given n points with known 3D coordinates X_i and known image projections x_i, estimate the camera parameters





Lazebnik

Camera calibration: Linear method

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

$$\mathbf{a} \times \mathbf{b} = [\mathbf{a}]_\times \mathbf{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

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

Two linearly independent equations

Lazebnik

Camera calibration: Linear method

UNIVERSITÄT BONN

$$\begin{bmatrix} 0^T & X_1^T & -y_1X_1^T \\ X_1^T & 0^T & -x_1X_1^T \\ \cdots & \cdots & \cdots \\ 0^T & X_n^T & -y_nX_n^T \\ X_n^T & 0^T & -x_nX_n^T \end{bmatrix} \begin{pmatrix} P_1 \\ P_2 \\ P_3 \end{pmatrix} = 0 \qquad \mathrm{Ap} = 0$$

- P has 11 degrees of freedom (12 parameters, but scale is arbitrary)
- One 2D/3D correspondence gives us two linearly independent equations
- Homogeneous least squares
- 6 correspondences needed for a minimal solution

Lazebnik

Camera calibration: Linear method

Advantages: easy to formulate and solve

Disadvantages

- Doesn’t directly tell you camera parameters
- Doesn’t model radial distortion
- Can’t impose constraints, such as known focal length and orthogonality

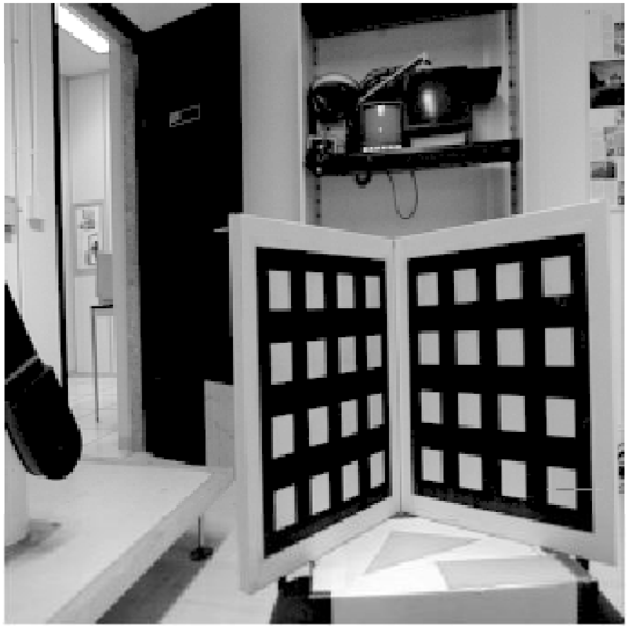
Non-linear methods are preferred

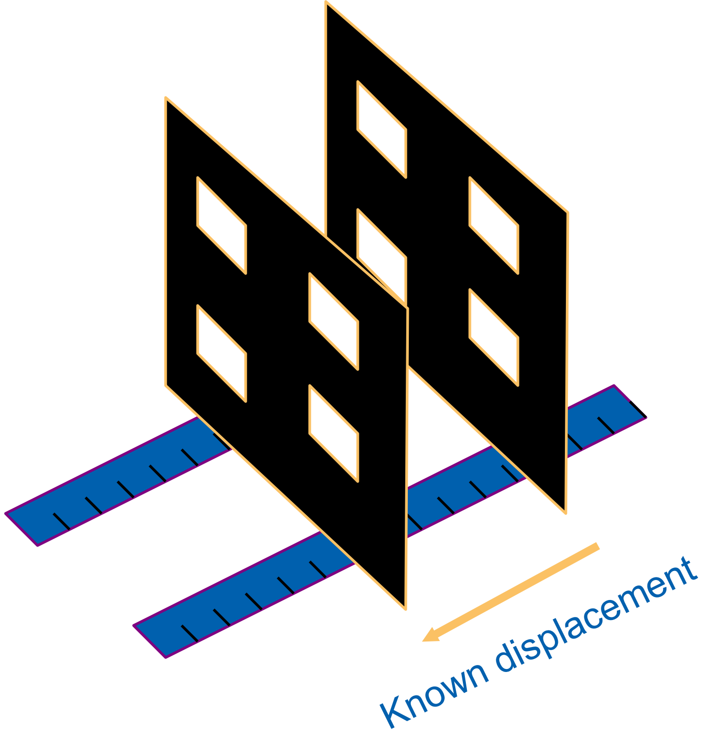
- Define error as difference between projected points and measured points
- Minimize error using Newton’s method or other non-linear optimization

Source: D. Hoiem

Calibration Object

Use precisely known 3D points

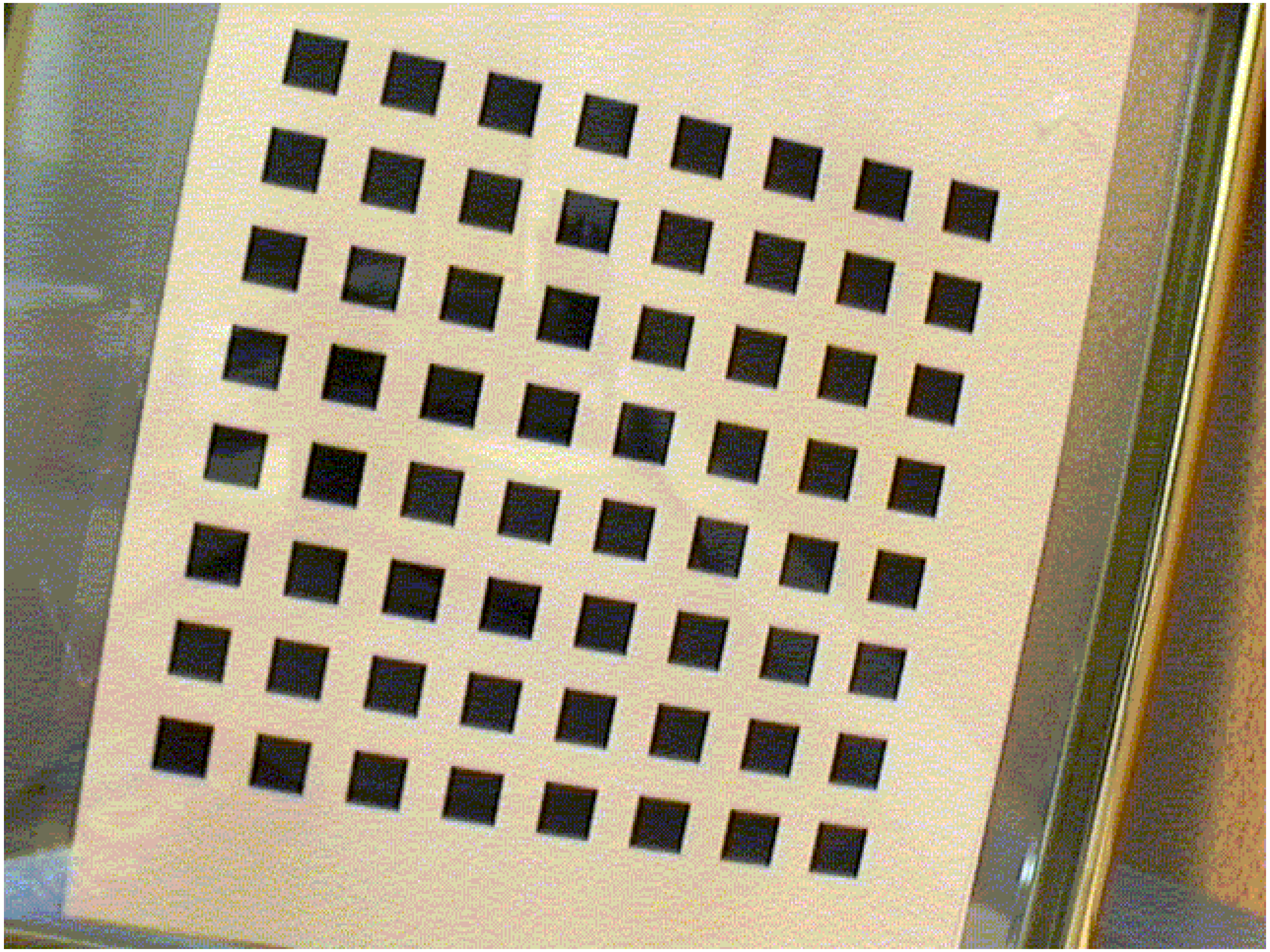




Shortcoming: Not flexible

Zhengyou Zhang

Intrinsic Calibration with Planes



Zhengyou Zhang

Intrinsic Calibration with Planes

Use only one plane

- Print a pattern on a paper
- Attach the paper on a planar surface
- Show the plane freely a few times to the camera

Advantages

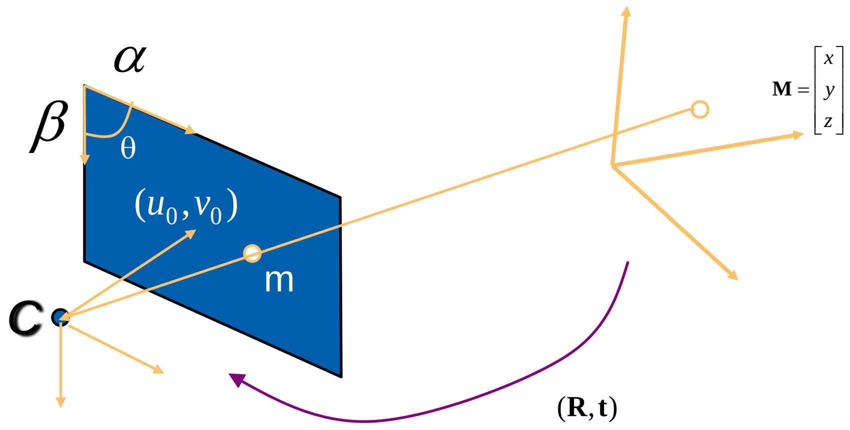
- Flexible
- Robust

Implementation in OpenCV or Matlab toolbox:
http://www.vision.caltech.edu/bouguetj/calib_doc/

[Z. Zhang. Flexible Camera Calibration by Viewing a Plane from Unknown Orientations. ICCV99]

Zhengyou Zhang

Camera Model



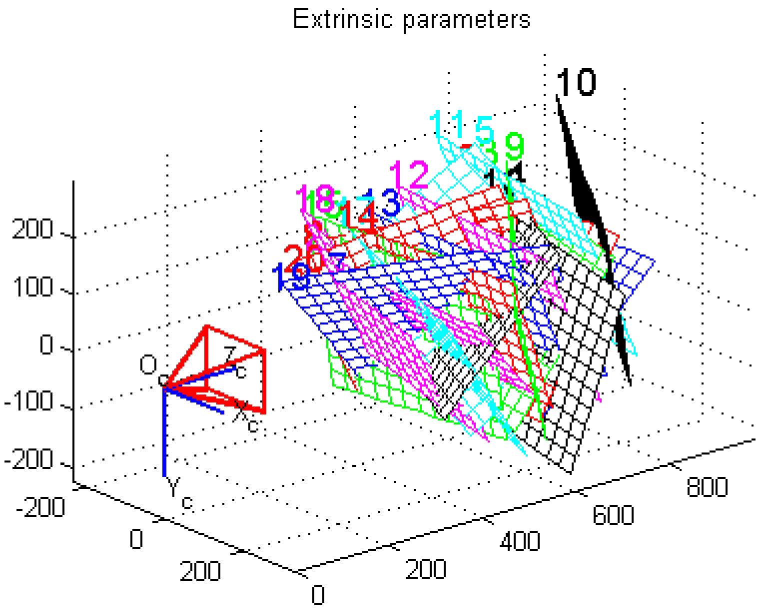
$$\underbrace{s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}}_{\tilde{\mathbf{m}}} = \underbrace{\begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t} \end{bmatrix}}_{[\mathbf{R} \ \ \mathbf{t}]} \underbrace{\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}}_{\tilde{\mathbf{M}}}$$

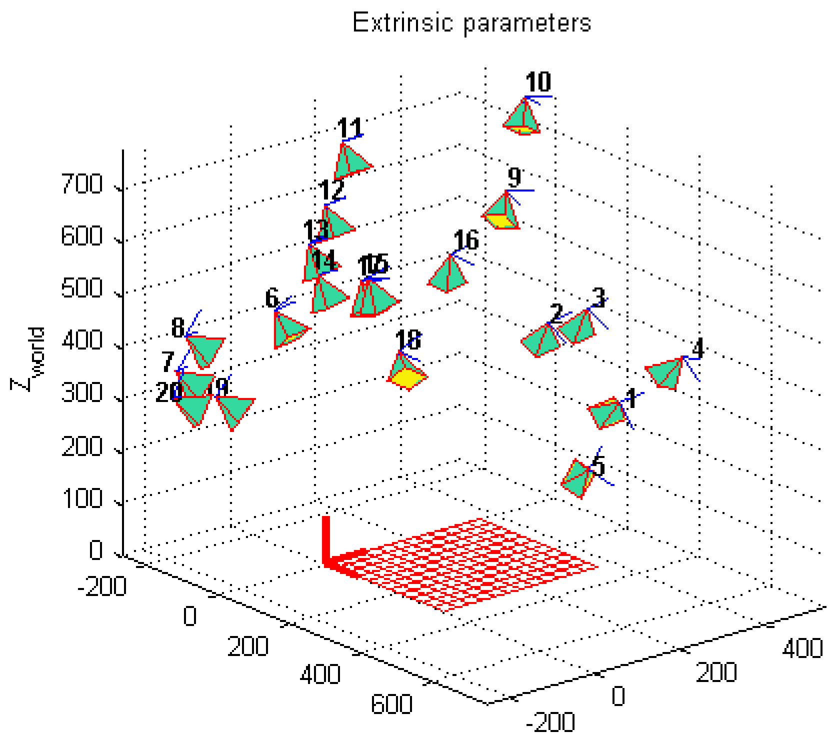
Zhengyou Zhang

Calibration process

UNIVERSITÄT BONN

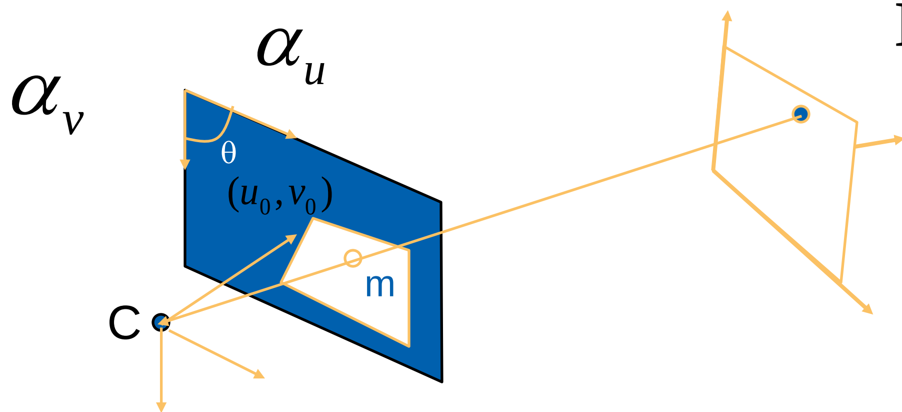
Extrinsic parameters





Plane projection

For convenience, assume the plane at $z = 0$.



$$\mathbf{M} = \begin{bmatrix} x \\ y \end{bmatrix}$$

$$z = 0$$

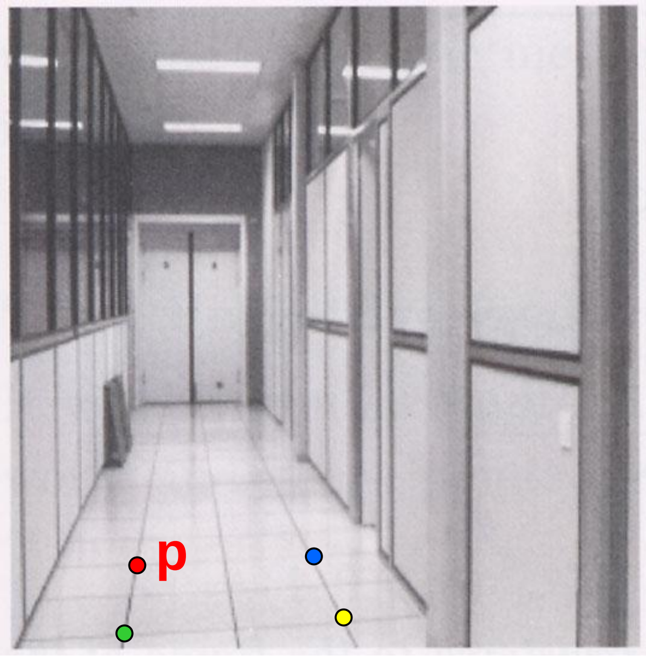
The relation between image points and model points is then given by a homography **H**:

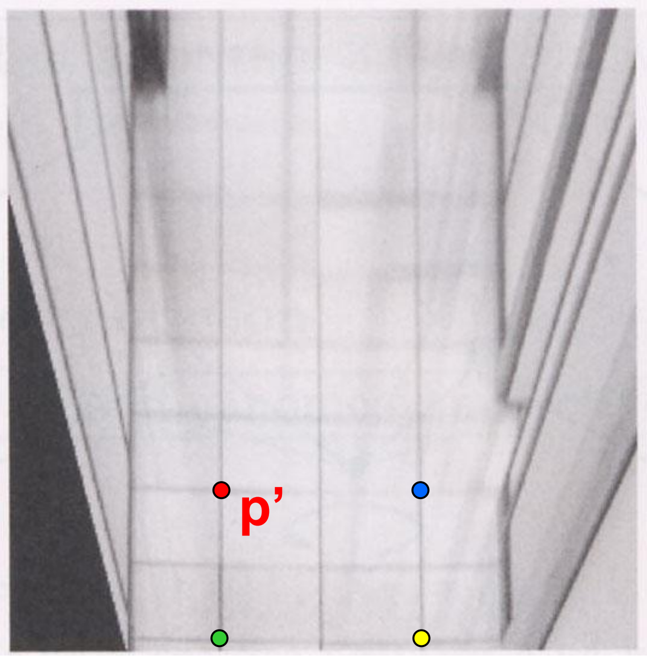
$$s\tilde{\mathbf{m}} = \mathbf{H}\tilde{\mathbf{M}} \quad \text{with} \quad \mathbf{H} = \mathbf{A}\begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix}$$

$$\mathbf{A}\begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = \mathbf{A}\begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

Zhengyou Zhang

Recall: Image rectification





Source: Steve Seitz

What do we get from one image?

We can obtain two equations in 6 intermediate homogeneous parameters.

Given **H**, which is defined up to a scale factor,

And let $\mathbf{H} = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3 \end{bmatrix}$, we have

$$\begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3 \end{bmatrix} = \lambda \mathbf{A} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix}$$

This yields

$$\mathbf{h}_1^T \mathbf{A}^{-T} \mathbf{A}^{-1} \mathbf{h}_2 = 0$$
$$\mathbf{h}_1^T \mathbf{A}^{-T} \mathbf{A}^{-1} \mathbf{h}_1 = \mathbf{h}_2^T \mathbf{A}^{-T} \mathbf{A}^{-1} \mathbf{h}_2$$

Zhengyou Zhang

Linear Equations

UNIVERSITÄT BONN

Let

$$\mathbf{B} = \mathbf{A}^{-T}\mathbf{A}^{-1} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix} \longleftarrow \text{symmetric}$$

Define $\mathbf{b} = [B_{11} \quad B_{12} \quad B_{22} \quad B_{13} \quad B_{23} \quad B_{33}]$ up to a scale factor

Rewrite

$$\mathbf{h}_1^T \mathbf{A}^{-T}\mathbf{A}^{-1}\mathbf{h}_2 = 0$$
$$\mathbf{h}_1^T \mathbf{A}^{-T}\mathbf{A}^{-1}\mathbf{h}_1 = \mathbf{h}_2^T \mathbf{A}^{-T}\mathbf{A}^{-1}\mathbf{h}_2$$

as linear equations:

$$\begin{bmatrix} \mathbf{v}_{12}^T \\ (\mathbf{v}_{11} - \mathbf{v}_{22})^T \end{bmatrix} \mathbf{b} = \mathbf{0} \qquad \begin{aligned} \mathbf{v}_{ij} = [&h_{i1}h_{j1}, h_{i1}h_{j2} + h_{i2}h_{j1}, h_{i2}h_{j2}, \\ &h_{i3}h_{j1} + h_{i1}h_{j3}, h_{i3}h_{j2} + h_{i2}h_{j3}, h_{i3}h_{j3}]^T \end{aligned}$$

Zhengyou Zhang

Camera parameters

UNIVERSITÄT BONN

Intrinsic camera parameters

$$
\begin{aligned}
v_0 &= (B_{12}B_{13} - B_{11}B_{23})/(B_{11}B_{22} - B_{12}^2) \\
\lambda &= B_{33} - [B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})]/B_{11} \\
\alpha &= \sqrt{\lambda/B_{11}} \\
\beta &= \sqrt{\lambda B_{11}/(B_{11}B_{22} - B_{12}^2)} \\
c &= -B_{12}\alpha^2\beta/\lambda \\
u_0 &= cv_0/\alpha - B_{13}\alpha^2/\lambda\,.
\end{aligned}
$$

$$
\mathsf{A} = \begin{pmatrix} \alpha & c & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{pmatrix}
$$

Rotation and translation

$$
\mathbf{r}_1 = \lambda \mathbf{A}^{-1}\mathbf{h}_1,\ \mathbf{r}_2 = \lambda \mathbf{A}^{-1}\mathbf{h}_2,\ \mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2,\ \mathbf{t} = \lambda \mathbf{A}^{-1}\mathbf{h}_3
$$

$$
\lambda = 1/\|\mathbf{A}^{-1}\mathbf{h}_1\| = 1/\|\mathbf{A}^{-1}\mathbf{h}_2\|
$$

Distortion

Distortion model ((x,y) undistorted image coordinates):

$$\breve{x} = x + x[k_1(x^2+y^2) + k_2(x^2+y^2)^2]$$
$$\breve{y} = y + y[k_1(x^2+y^2) + k_2(x^2+y^2)^2]$$

Converted to pixels by:

$$\breve{u} = u_0 + \alpha\breve{x} + c\breve{y}$$
$$\breve{v} = v_0 + \beta\breve{y}$$

$$\mathrm{A} = \begin{pmatrix} \alpha & c & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{pmatrix}$$

gives

$$\breve{u} = u + (u-u_0)[k_1(x^2+y^2) + k_2(x^2+y^2)^2]$$
$$\breve{v} = v + (v-v_0)[k_1(x^2+y^2) + k_2(x^2+y^2)^2]$$

i.e. distortion model centered at u_0 and v_0.

Distortion

Distortion model ((x,y) undistorted image coordinates):

$$\breve{x} = x + x[k_1(x^2+y^2) + k_2(x^2+y^2)^2]$$
$$\breve{y} = y + y[k_1(x^2+y^2) + k_2(x^2+y^2)^2]$$

Centered at u_0 and v_0:

$$\breve{u} = u + (u - u_0)[k_1(x^2+y^2) + k_2(x^2+y^2)^2]$$
$$\breve{v} = v + (v - v_0)[k_1(x^2+y^2) + k_2(x^2+y^2)^2]$$

Solution:

$$\underbrace{\begin{bmatrix} (u-u_0)(x^2+y^2) & (u-u_0)(x^2+y^2)^2 \\ (v-v_0)(x^2+y^2) & (v-v_0)(x^2+y^2)^2 \end{bmatrix}}_{\text{D}} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \breve{u}-u \\ \breve{v}-v \end{bmatrix}}_{\text{d}}$$

$$\mathbf{k} = (\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T\mathbf{d}$$

Non-linear optimization

In practice, closed-form solution is used for initialization of non-linear optimization problem

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\|\mathbf{m}_{ij} - \breve{\mathbf{m}}(\mathbf{A}, k_1, k_2, \mathbf{R}_i, \mathbf{t}_i, \mathtt{M}_j)\|^2$$

Solved with Levenberg-Marquardt algorithm.

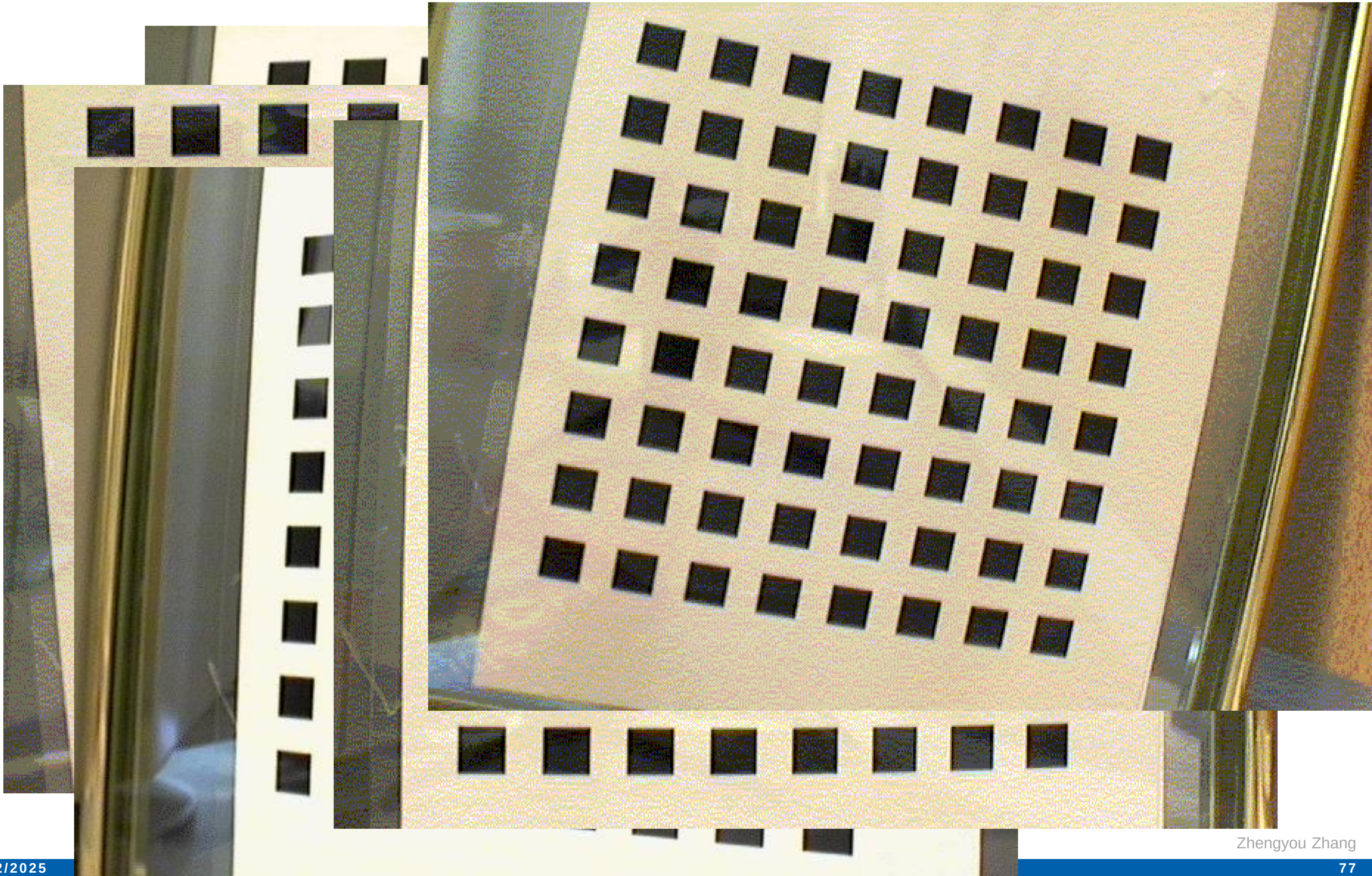
Without skew at least 2 images are needed, the more the better.

Summary

- Show the plane under *n* different orientations (n > 1)
- Estimate the n homography matrices
  (analytic solution followed by MLE)
- Solve analytically the 6 intermediate parameters *(defined up to a scale factor*)
- Extract the five intrinsic parameters
- Compute the extrinsic parameters
- Refine all parameters with MLE

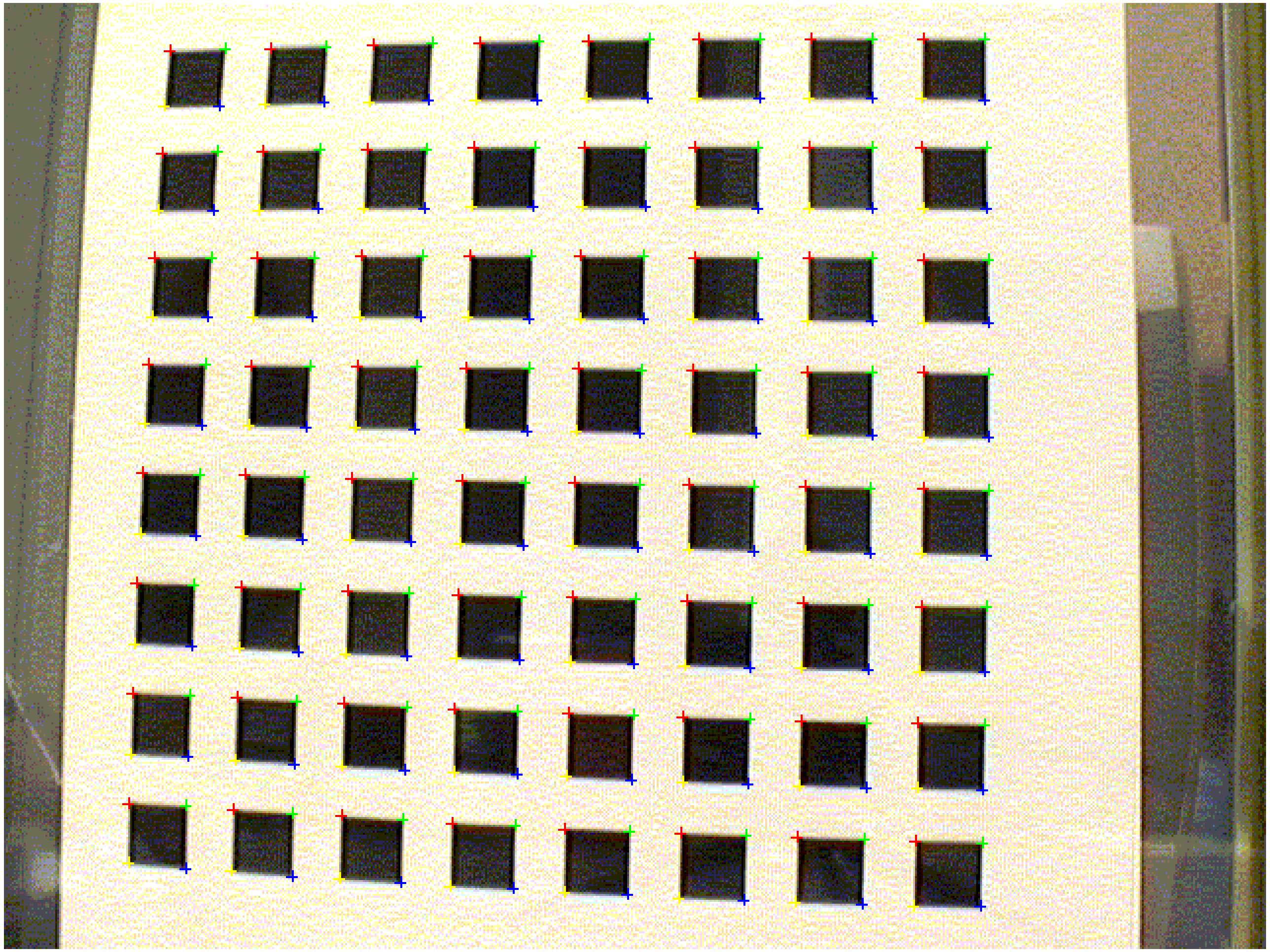
Zhengyou Zhang

Experimental results



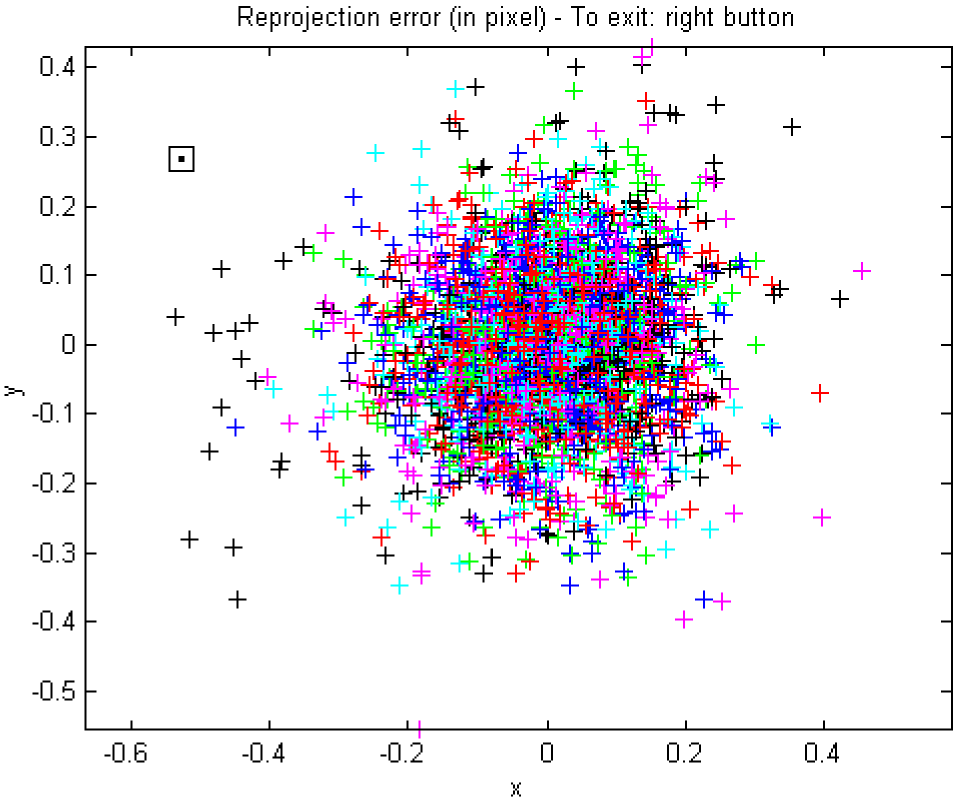
Zhengyou Zhang

Extracted corner points



Zhengyou Zhang

Reprojection error



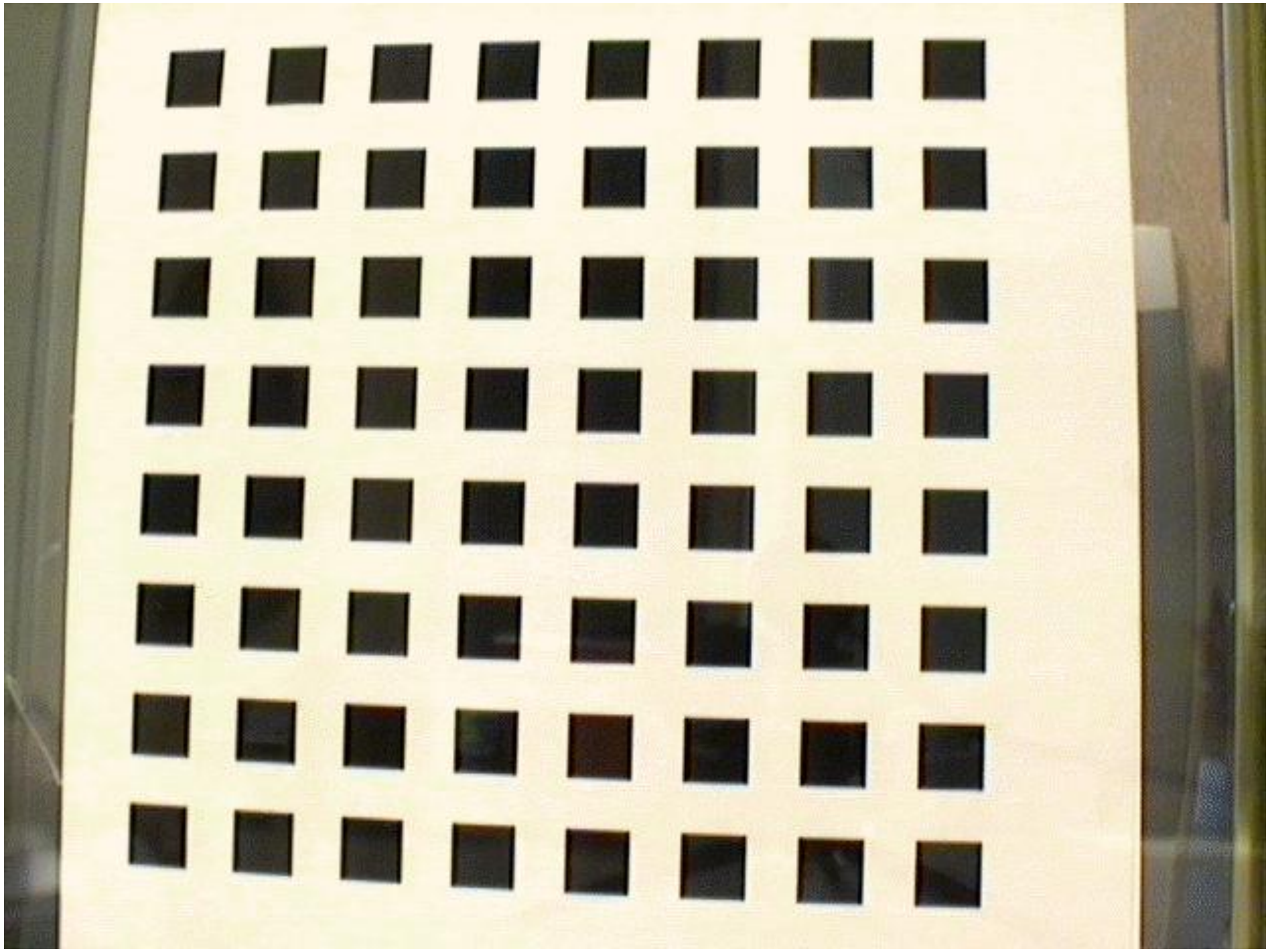
Number of images

Table 1: Results with real data of 2 through 5 images

nb	2 images			3 images			4 images			5 images		
	initial	final	σ	initial	final	σ	initial	final	σ	initial	final	σ
α	825.59	830.47	4.74	917.65	830.80	2.06	876.62	831.81	1.56	877.16	832.50	1.41
β	825.26	830.24	4.85	920.53	830.69	2.10	876.22	831.82	1.55	876.80	832.53	1.38
γ	0	0	0	2.2956	0.1676	0.109	0.0658	0.2867	0.095	0.1752	0.2045	0.078
u_0	295.79	307.03	1.37	277.09	305.77	1.45	301.31	304.53	0.86	301.04	303.96	0.71
v_0	217.69	206.55	0.93	223.36	206.42	1.00	220.06	206.79	0.78	220.41	206.59	0.66
k_1	0.161	−0.227	0.006	0.128	−0.229	0.006	0.145	−0.229	0.005	0.136	−0.228	0.003
k_2	−1.955	0.194	0.032	−1.986	0.196	0.034	−2.089	0.195	0.028	−2.042	0.190	0.025
RMS	0.761	0.295		0.987	0.393		0.927	0.361		0.881	0.335	

Zhengyou Zhang

Correction of Radial Distortion

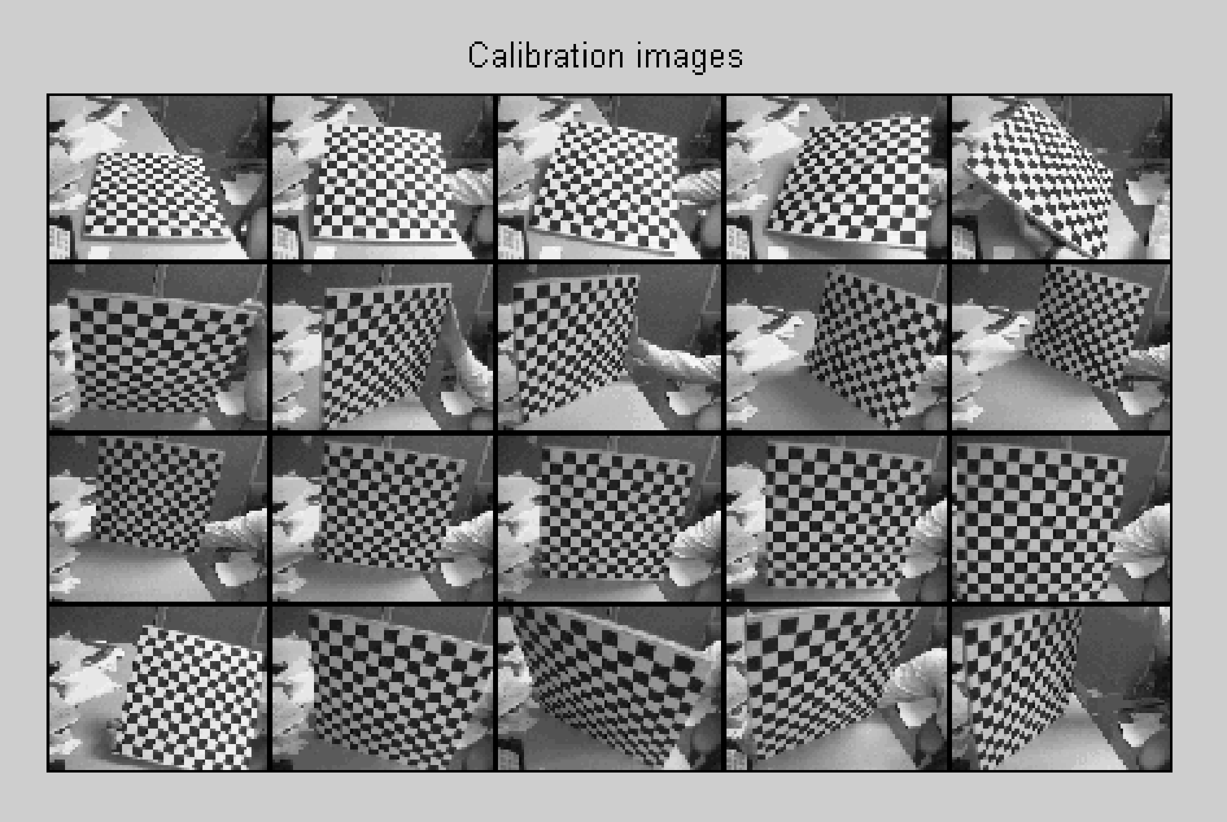


Original image

Zhengyou Zhang

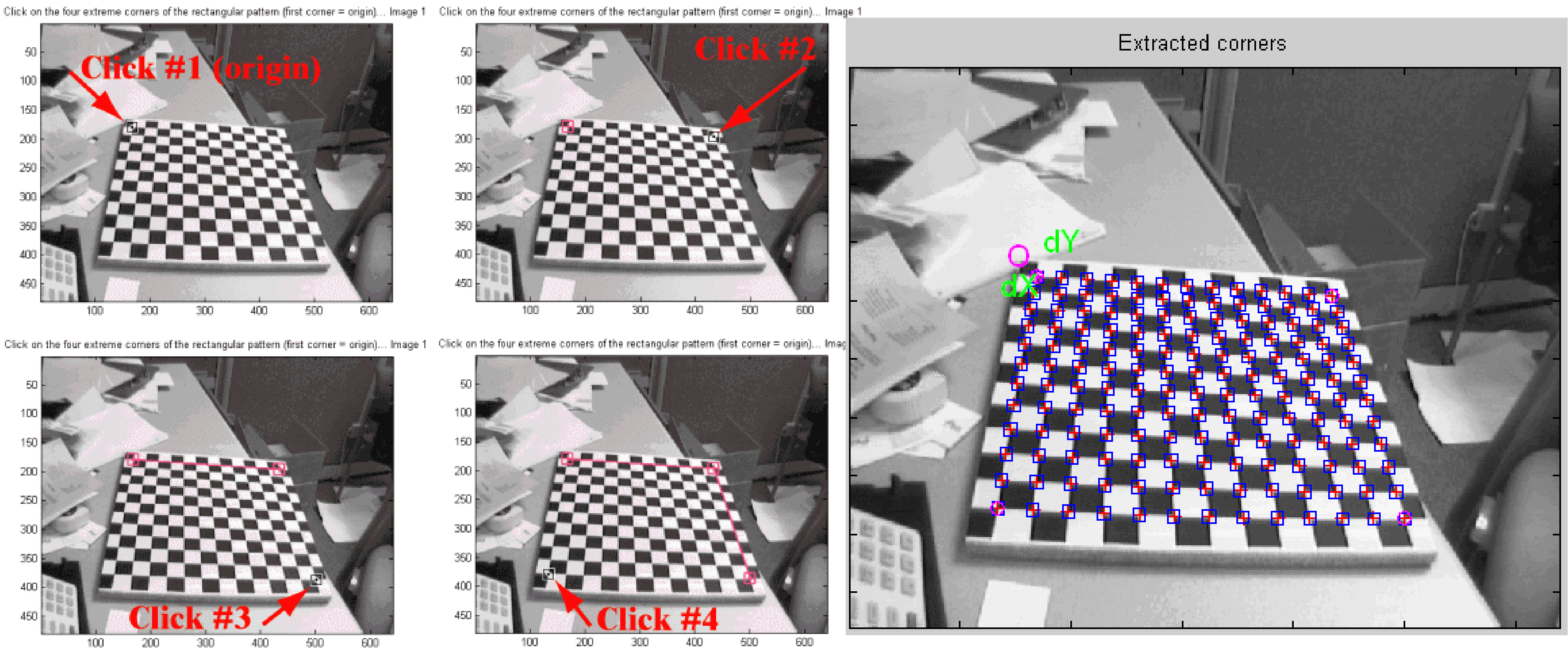
Calibration process

Capture images



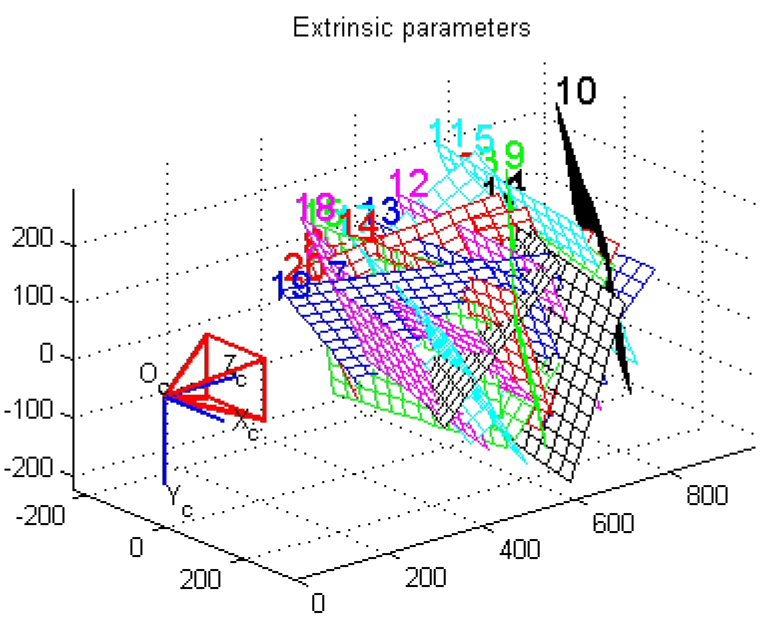
Calibration process

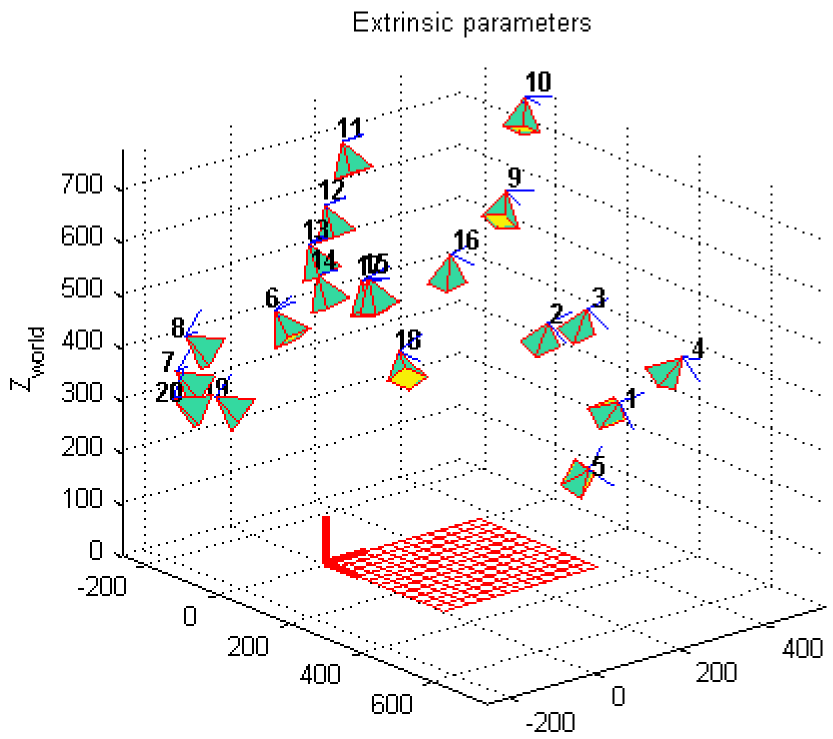
Click on four corners, corners extracted automatically



Calibration process

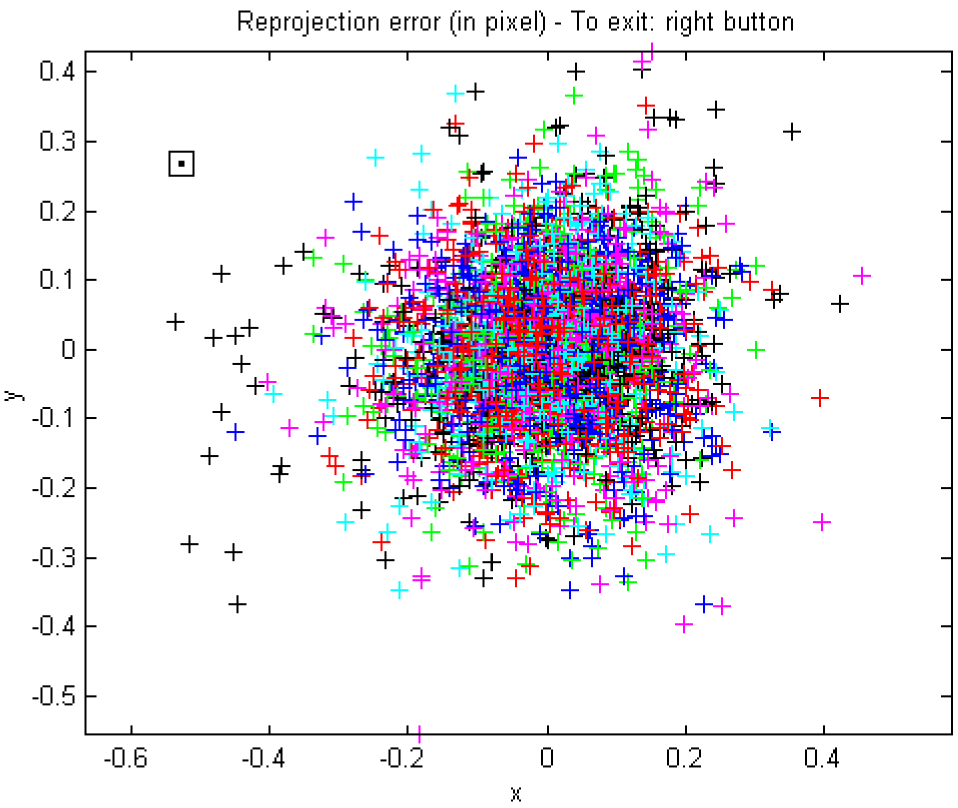
Extrinsic parameters





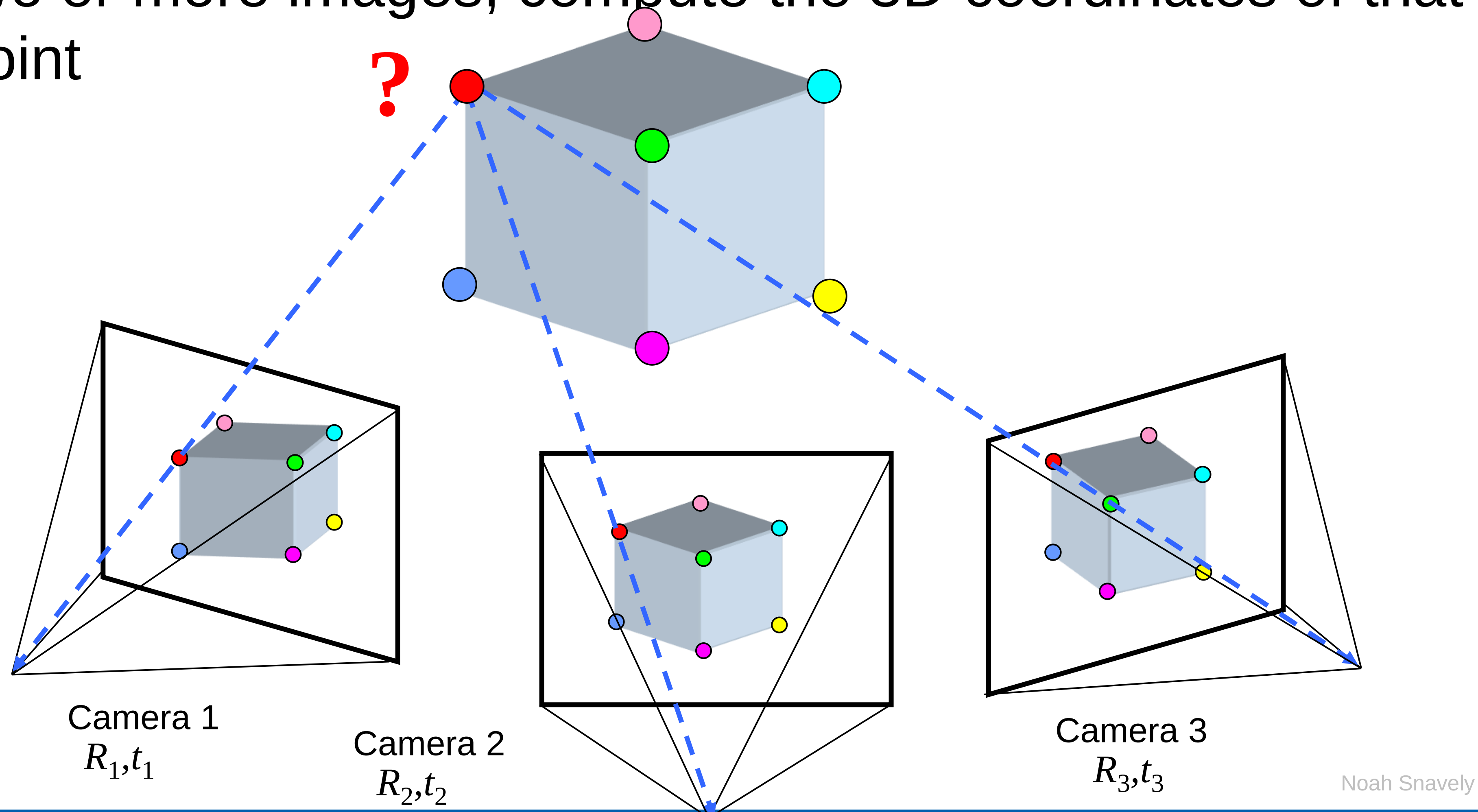
Calibration process

Reprojection error



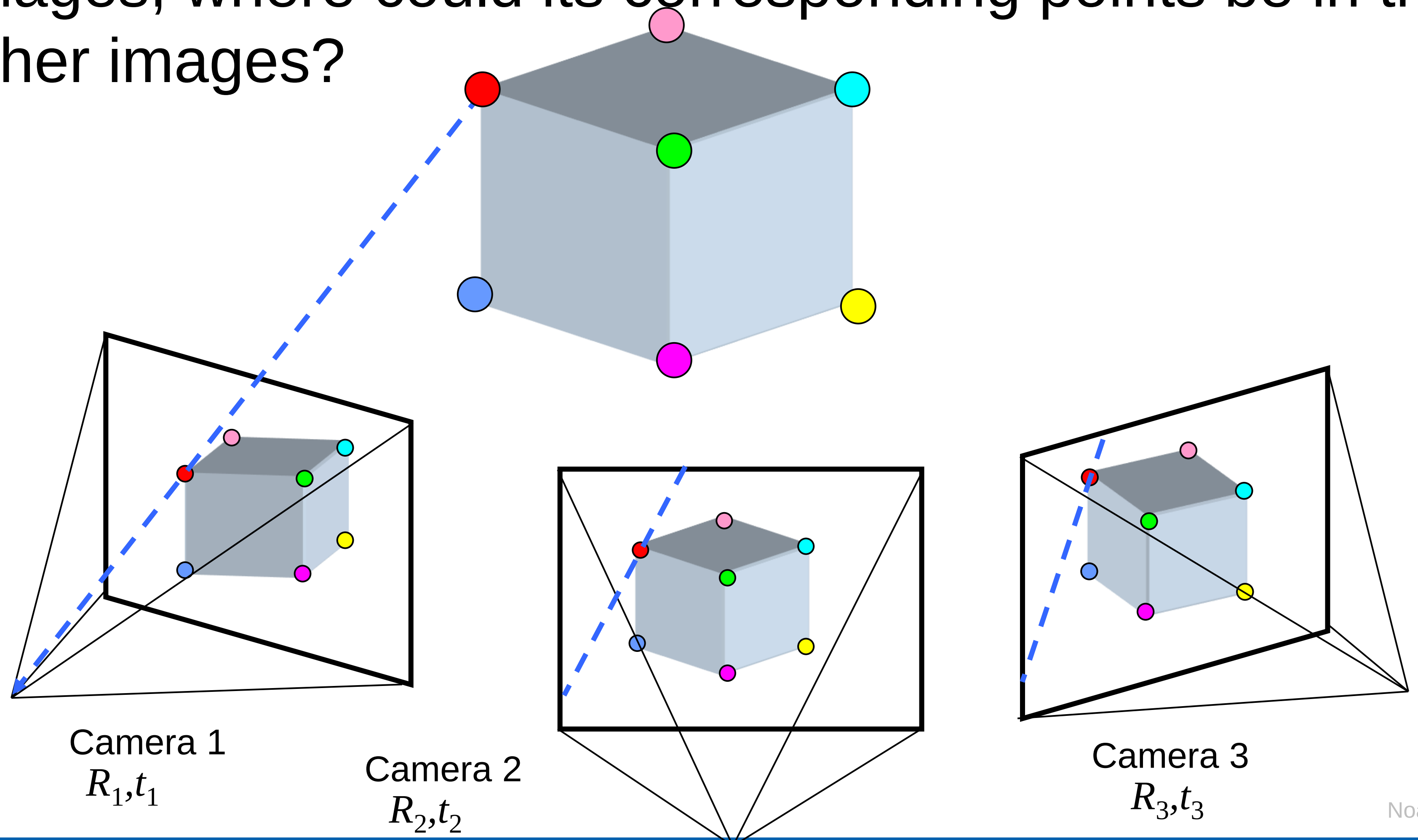
Multi-view geometry problems

Structure: Given projections of the same 3D point in two or more images, compute the 3D coordinates of that point



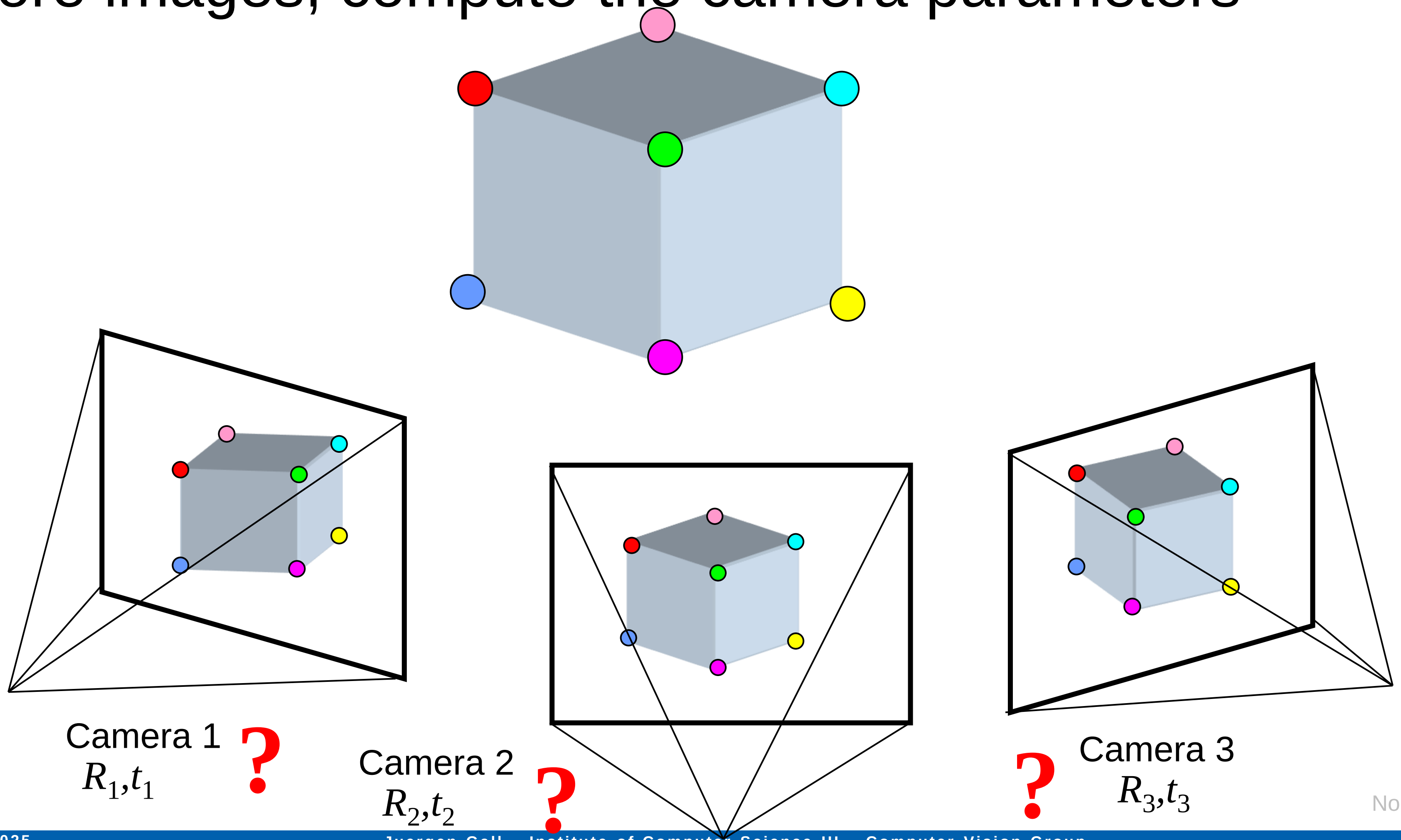
Multi-view geometry problems

Stereo correspondence: Given a point in one of the images, where could its corresponding points be in the other images?



Multi-view geometry problems

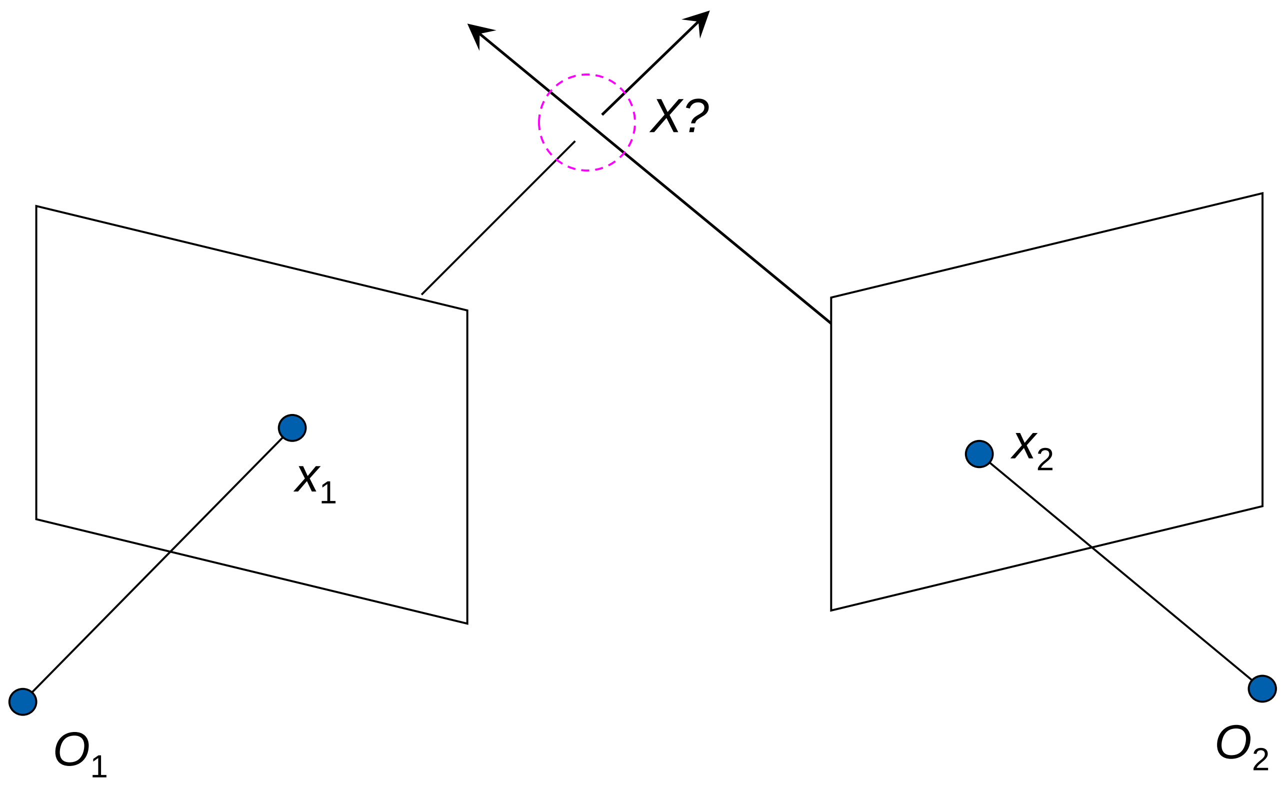
Motion: Given a set of corresponding points in two or more images, compute the camera parameters



Noah Snavely

Triangulation

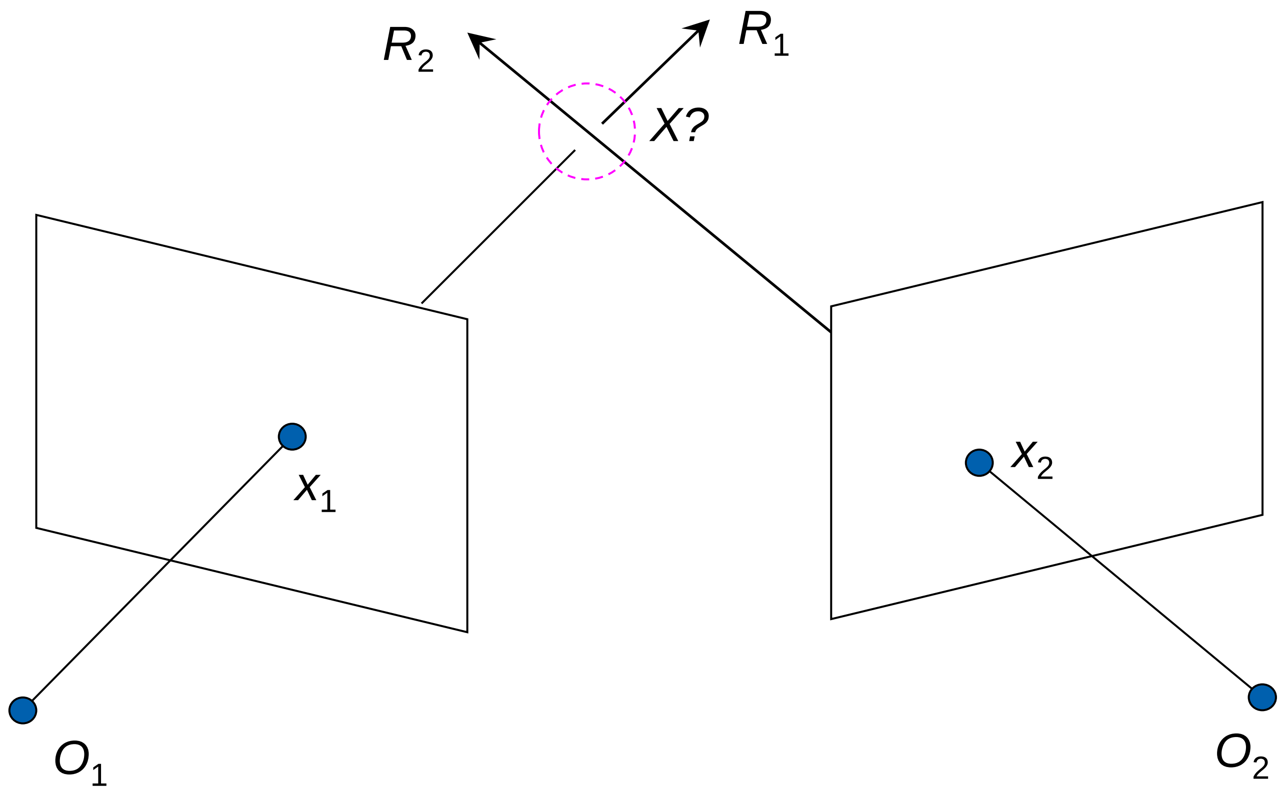
Given projections of a 3D point in two or more images (with known camera matrices), find the coordinates of the point



Lazebnik

Triangulation

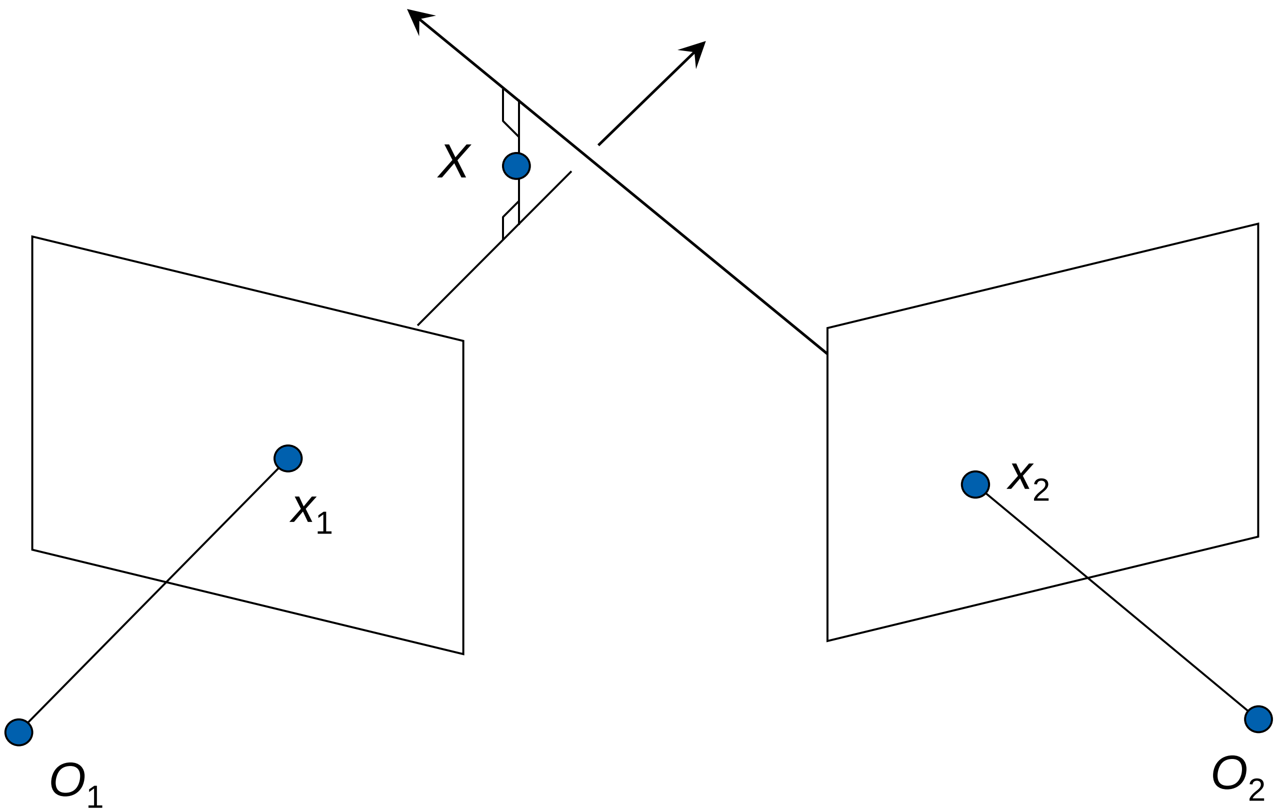
We want to intersect the two visual rays corresponding to x_1 and x_2, but because of noise and numerical errors, they don't meet exactly



Lazebnik

Triangulation: Geometric approach

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



Lazebnik

Triangulation: Linear approach

$$\lambda_1 \mathrm{x}_1 = \mathrm{P}_1 \mathrm{X} \qquad \mathrm{x}_1 \times \mathrm{P}_1 \mathrm{X} = 0 \qquad [\mathrm{x}_{1\times}]\mathrm{P}_1 \mathrm{X} = 0$$

$$\lambda_2 \mathrm{x}_2 = \mathrm{P}_2 \mathrm{X} \qquad \mathrm{x}_2 \times \mathrm{P}_2 \mathrm{X} = 0 \qquad [\mathrm{x}_{2\times}]\mathrm{P}_2 \mathrm{X} = 0$$

Cross product as matrix multiplication:

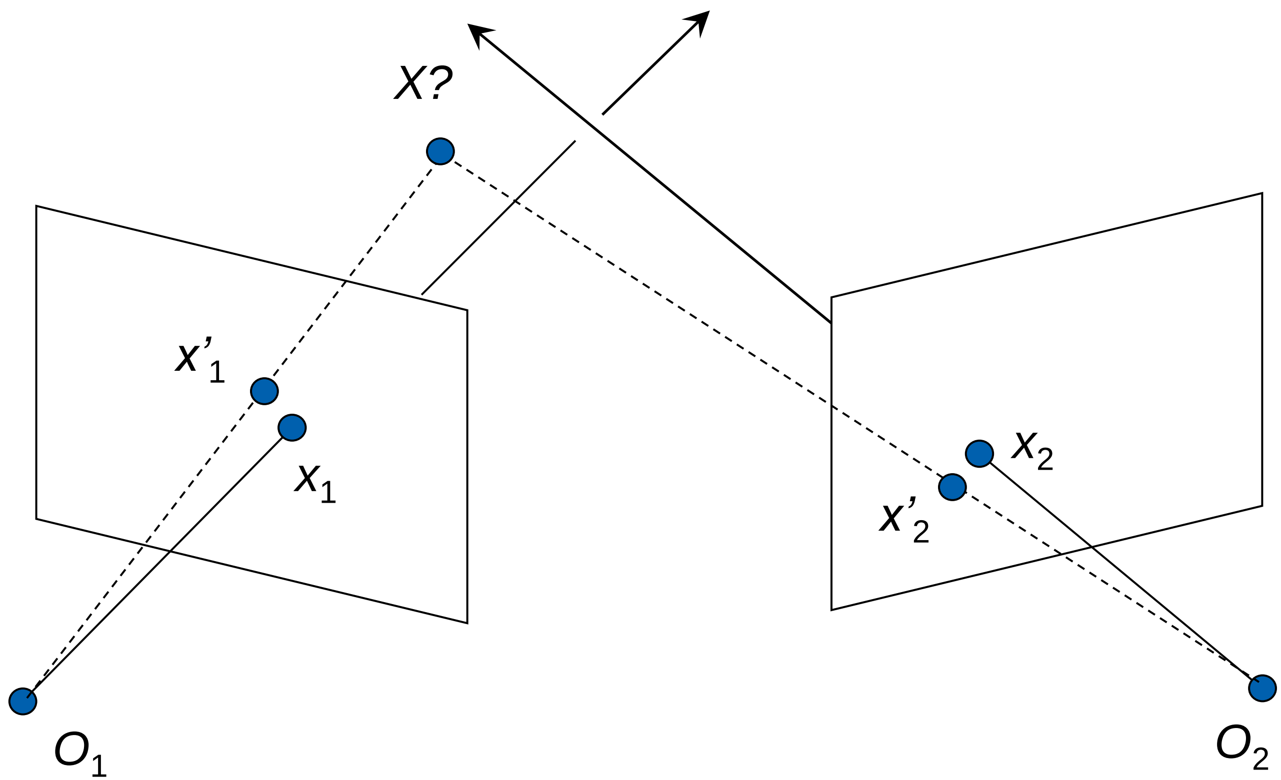
$$\mathbf{a} \times \mathbf{b} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = [\mathbf{a}_\times]\mathbf{b}$$

Lazebnik

Triangulation: Nonlinear approach

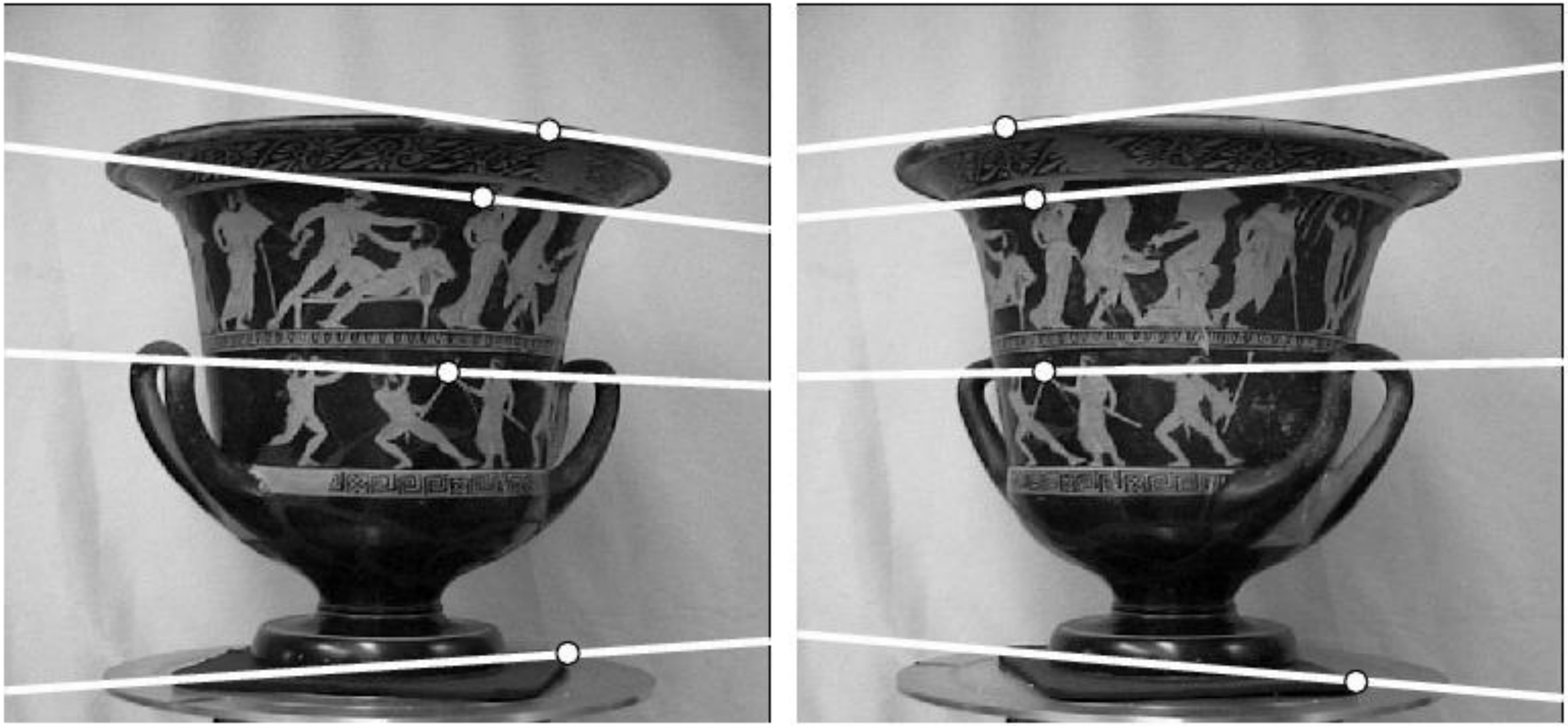
- Find X that minimizes

$$d^2(x_1, P_1 X) + d^2(x_2, P_2 X)$$



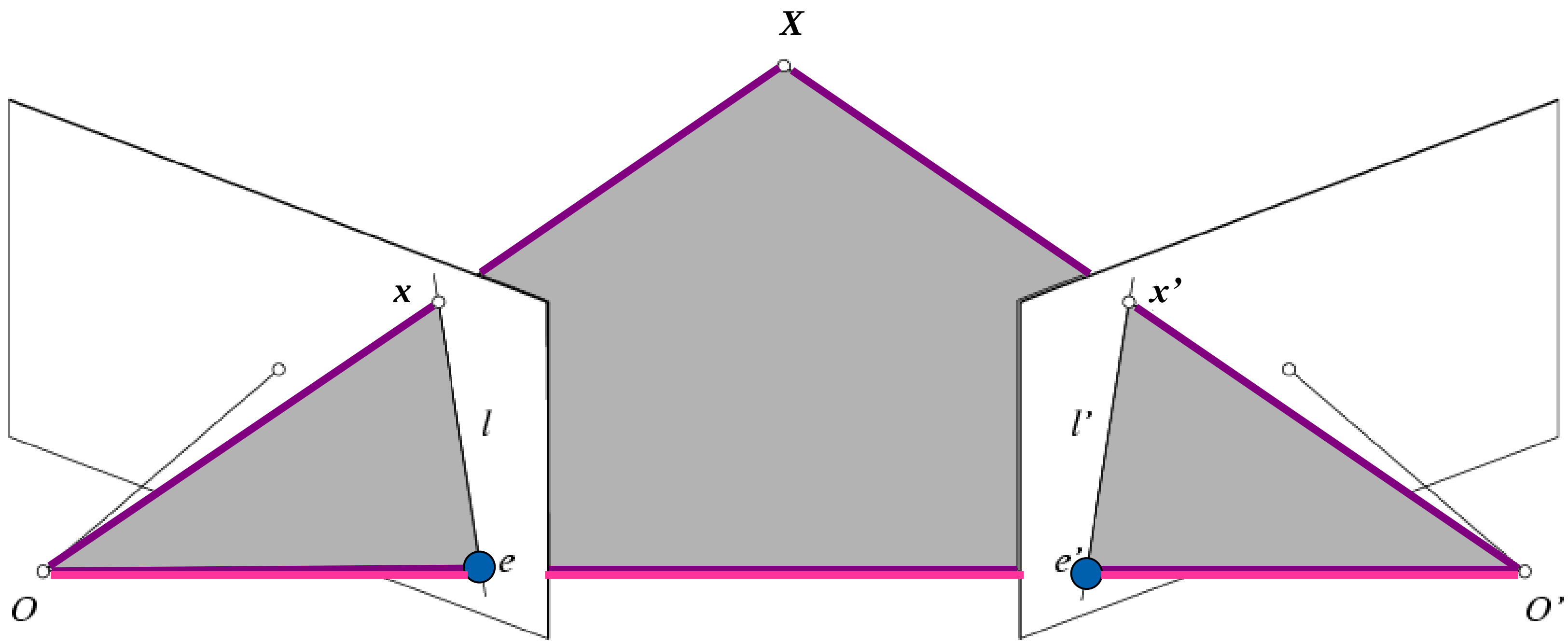
Lazebnik

Two-view geometry



Lazebnik

Epipolar geometry

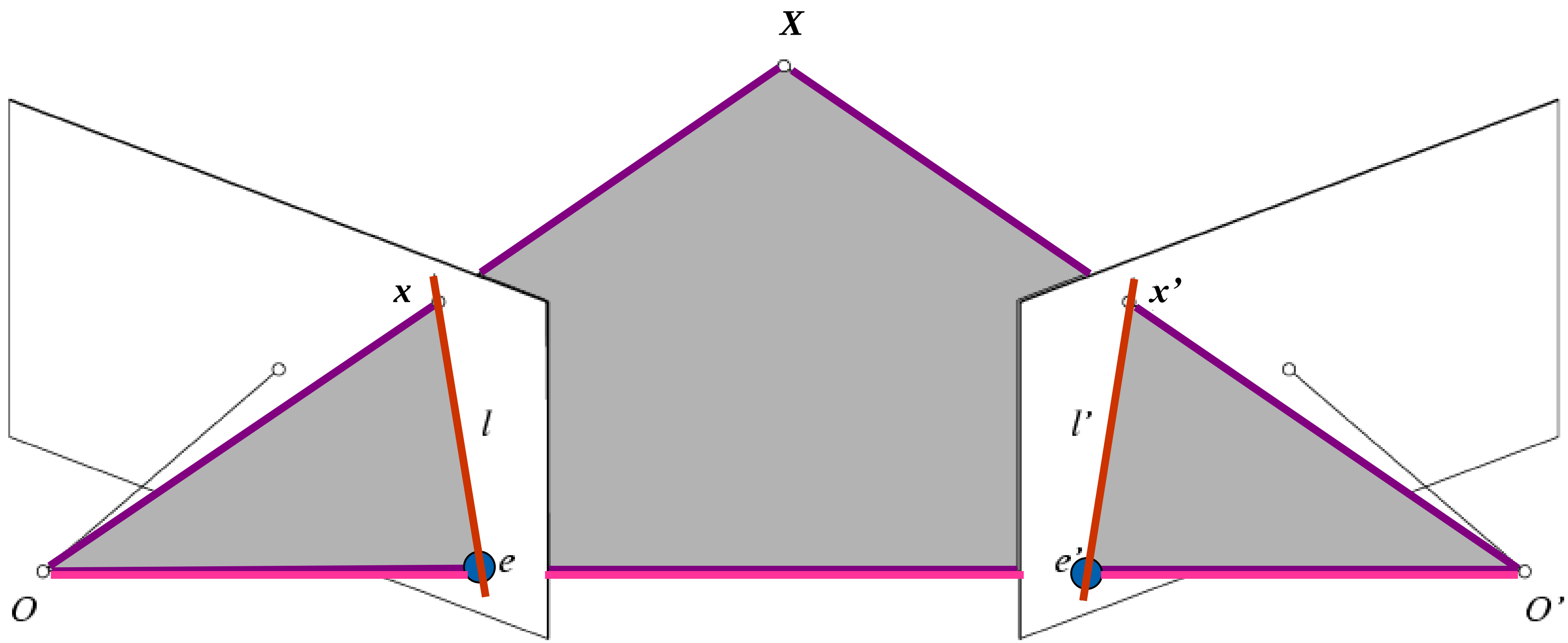


- **Baseline** – line connecting the two camera centers
- **Epipolar Plane** – plane containing baseline (1D family)
- **Epipoles**

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

Lazebnik

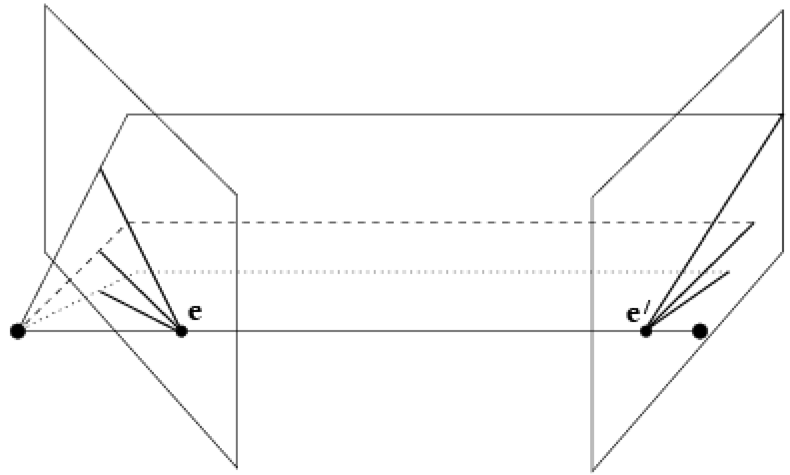
Epipolar geometry

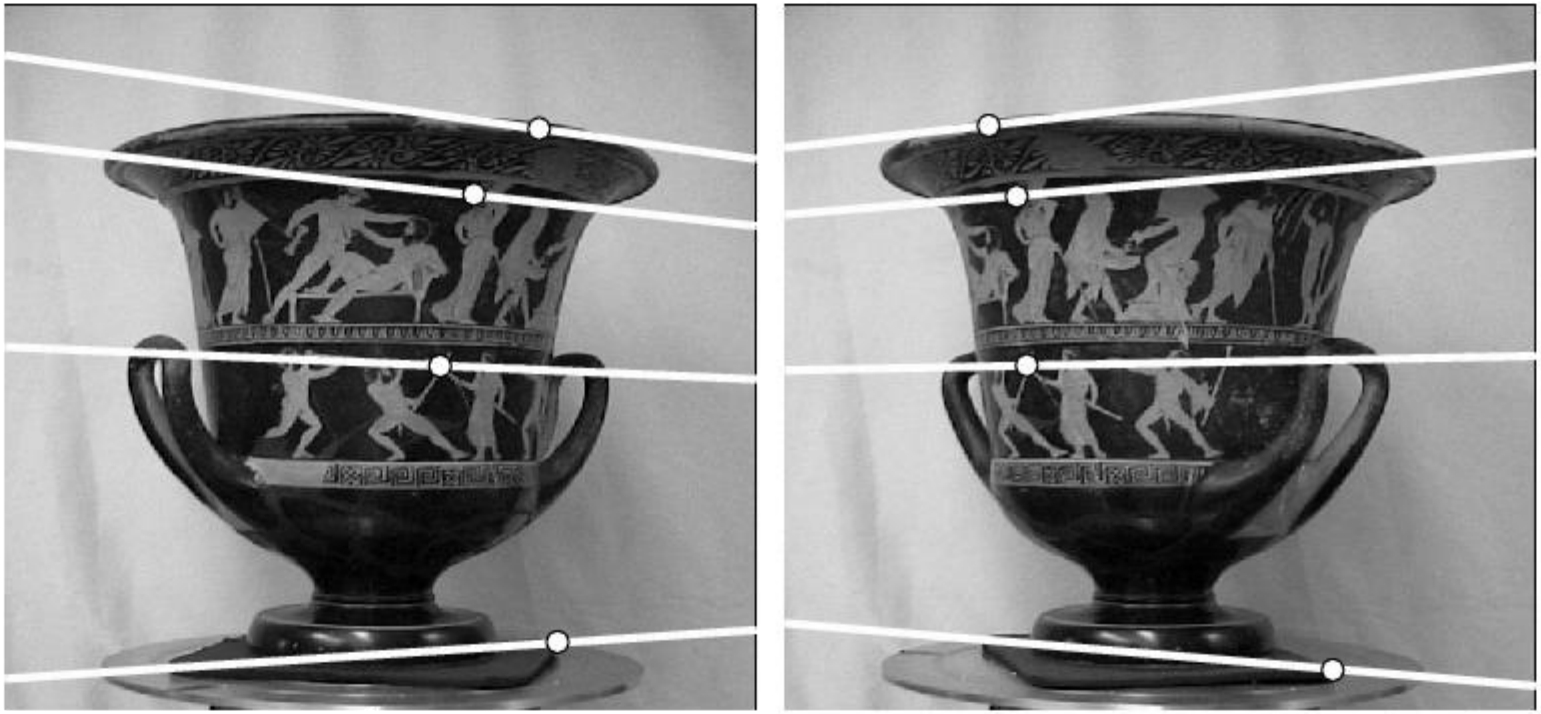


- **Baseline** – line connecting the two camera centers
- **Epipolar Plane** – plane containing baseline (1D family)
- **Epipoles**
  = intersections of baseline with image planes
  = projections of the other camera center
- **Epipolar Lines** - intersections of epipolar plane with image planes (always come in corresponding pairs)

Lazebnik

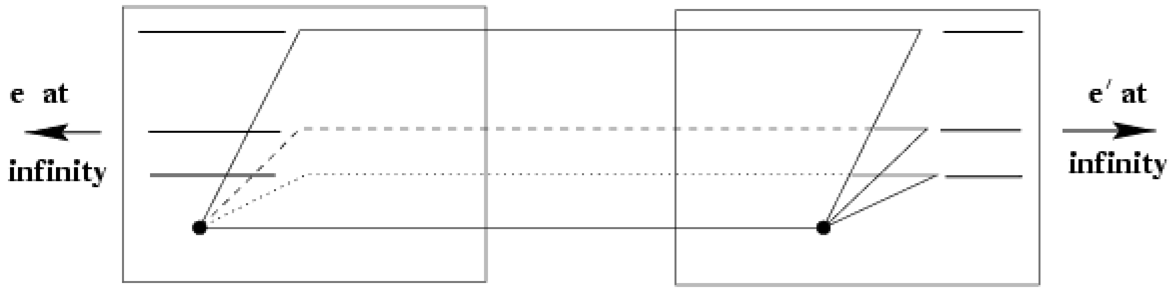
Example: Converging cameras





Lazebnik

Example: Motion parallel to image plane



Lazebnik

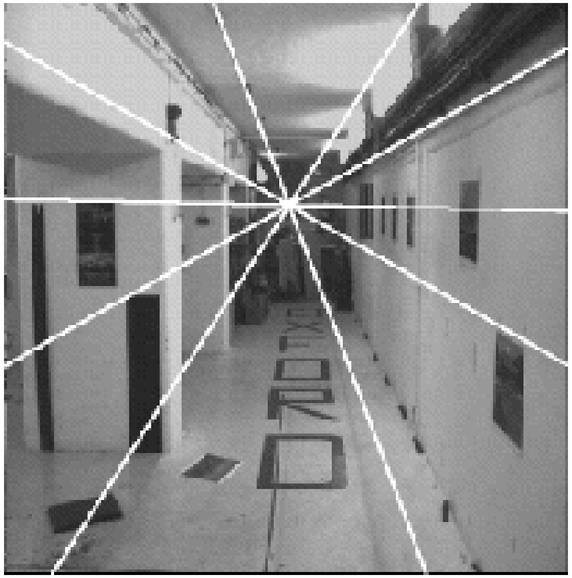
Example: Motion perpendicular to image plane

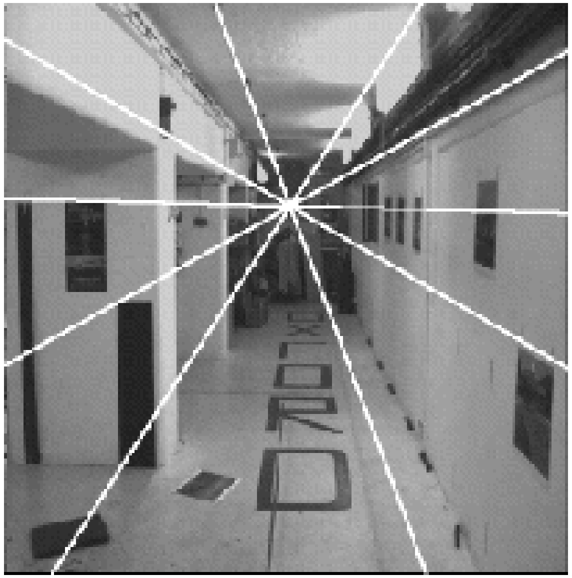
Lazebnik

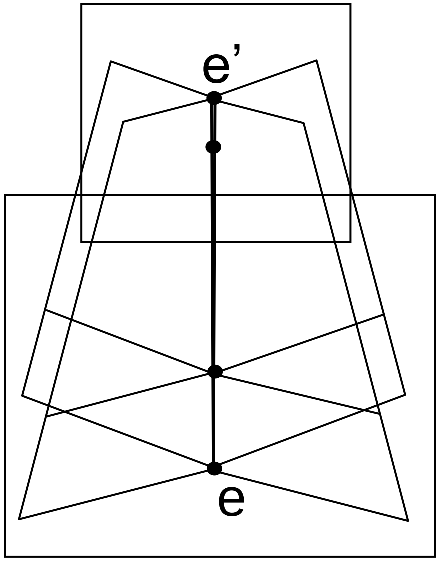
Example: Motion perpendicular to image plane

Lazebnik

Example: Motion perpendicular to image plane



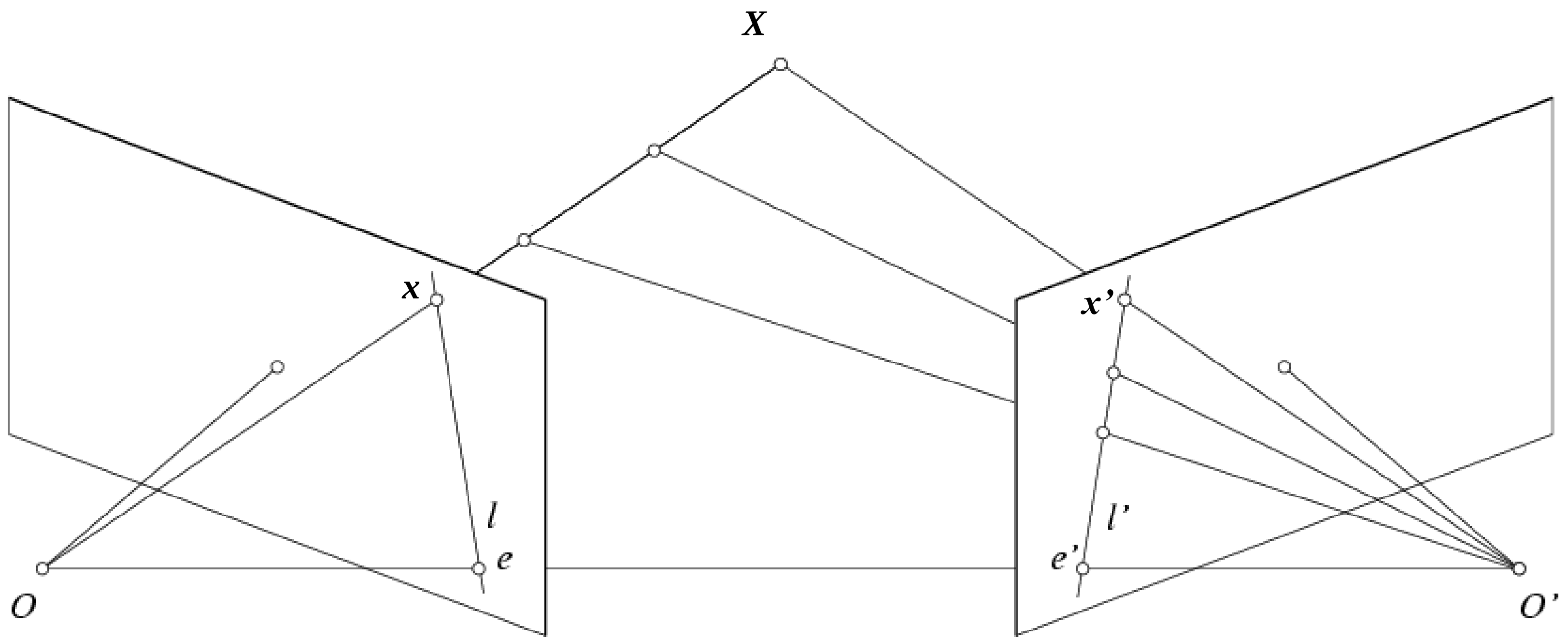




Epipole has same coordinates in both images.
Points move along lines radiating from e: “Focus of expansion”

Lazebnik

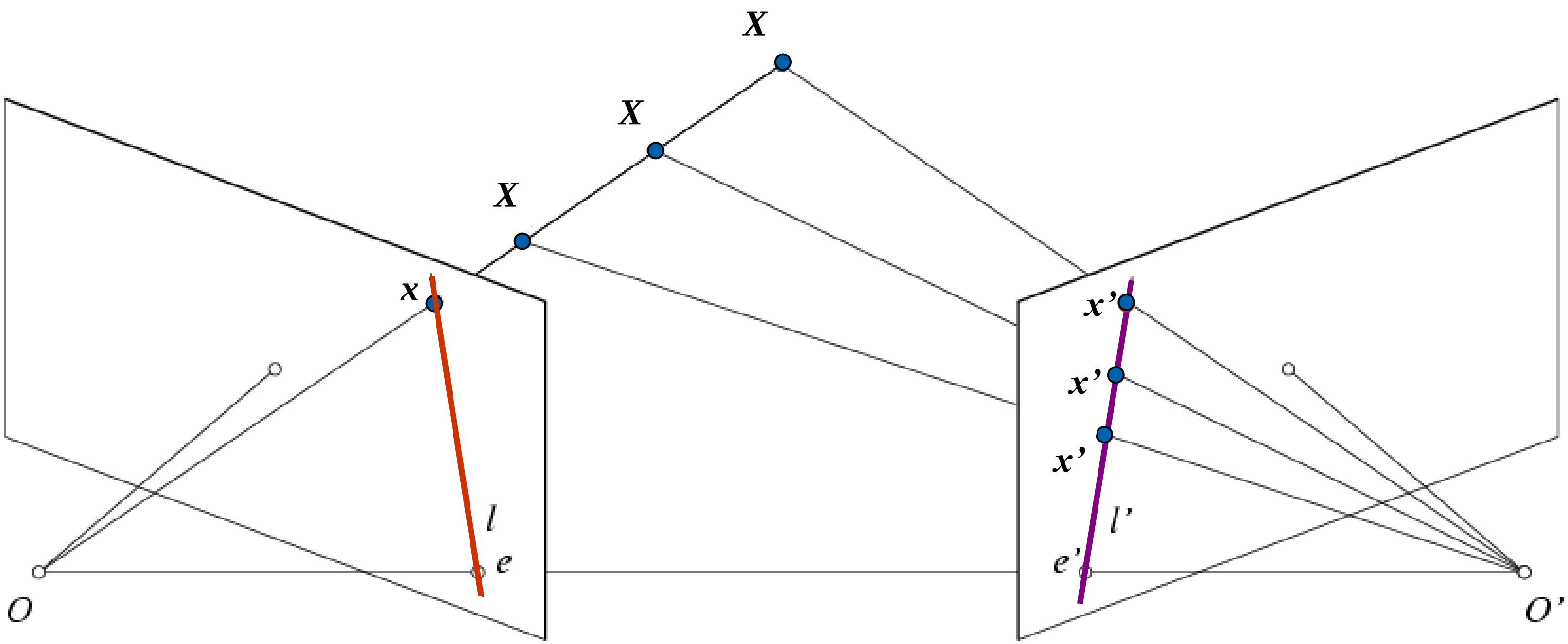
Epipolar constraint



If we observe a point x in one image, where can the corresponding point x’ be in the other image?

Lazebnik

Epipolar constraint



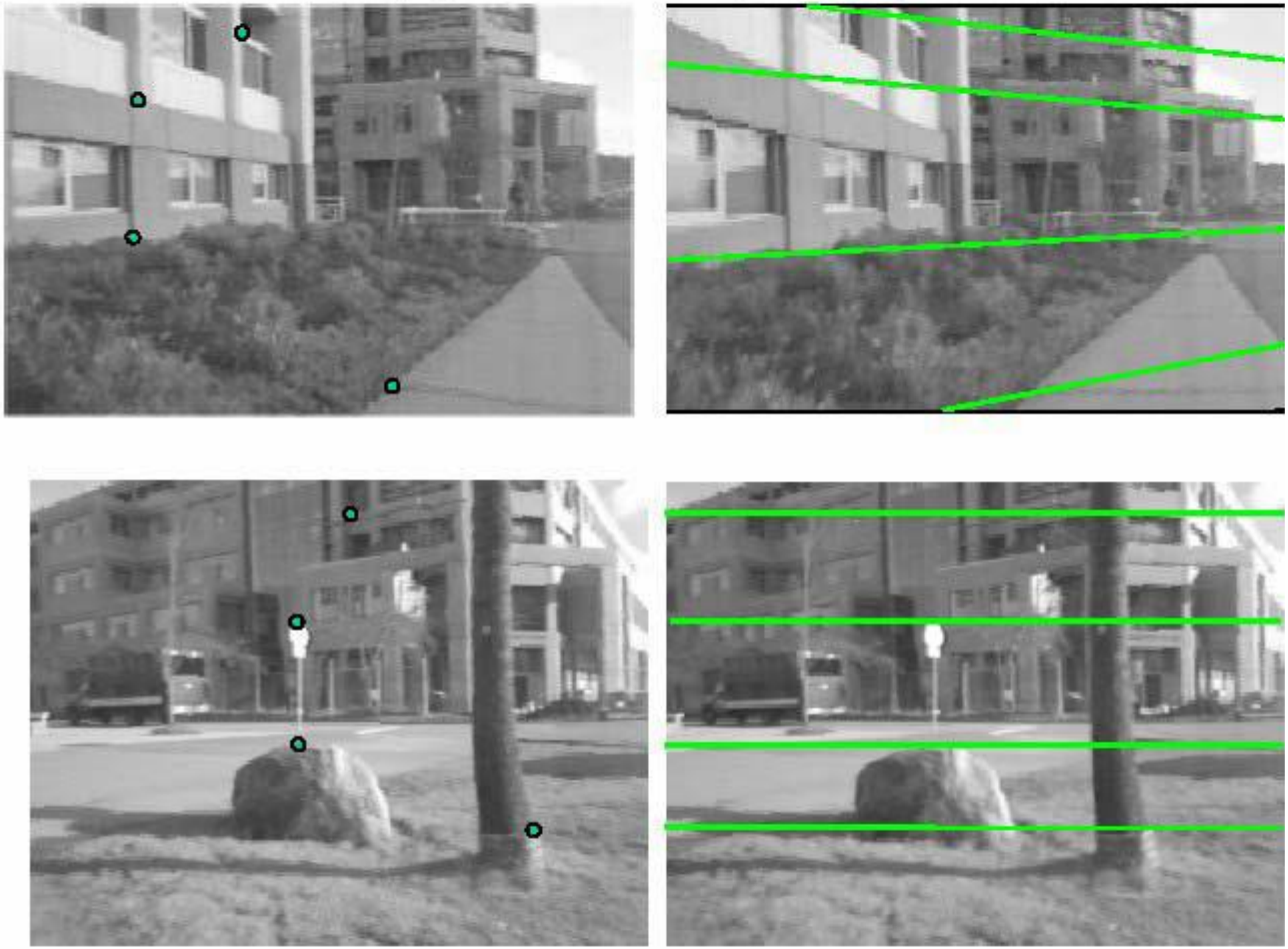
Potential matches for *x* have to lie on the corresponding epipolar line *l'*.

Potential matches for *x'* have to lie on the corresponding epipolar line *l*.

Lazebnik

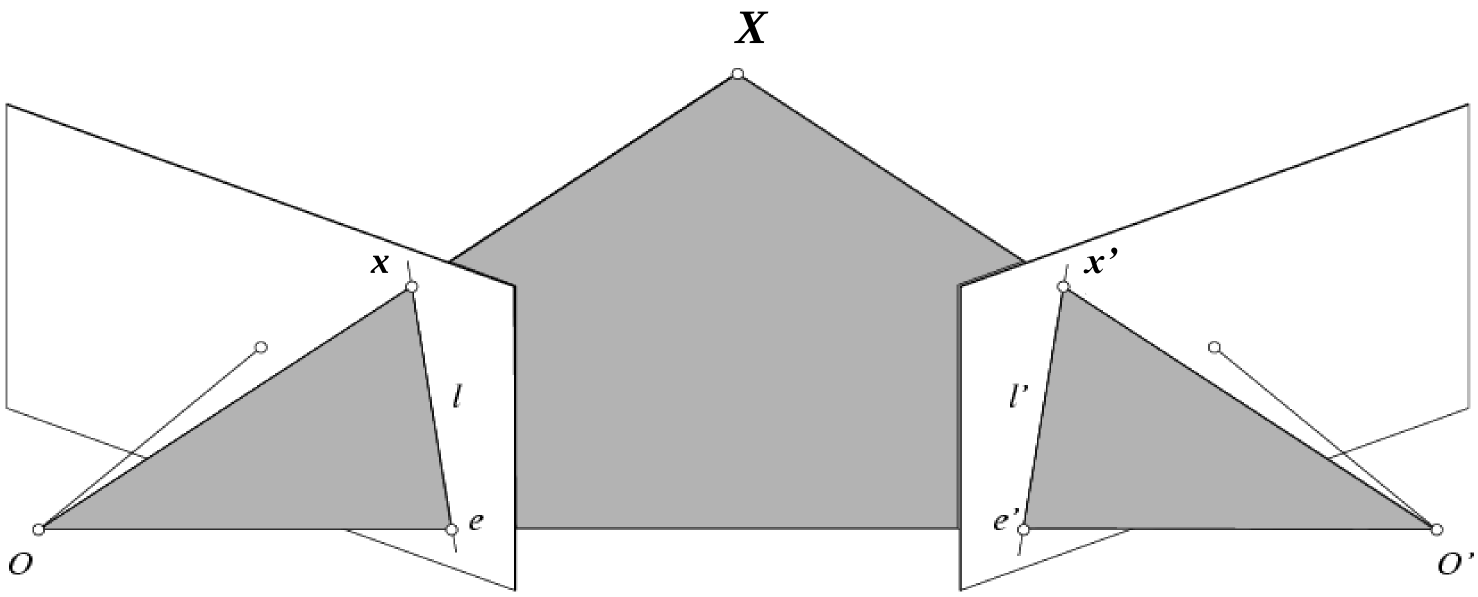
Epipolar constraint example

UNIVERSITÄT BONN



Lazebnik

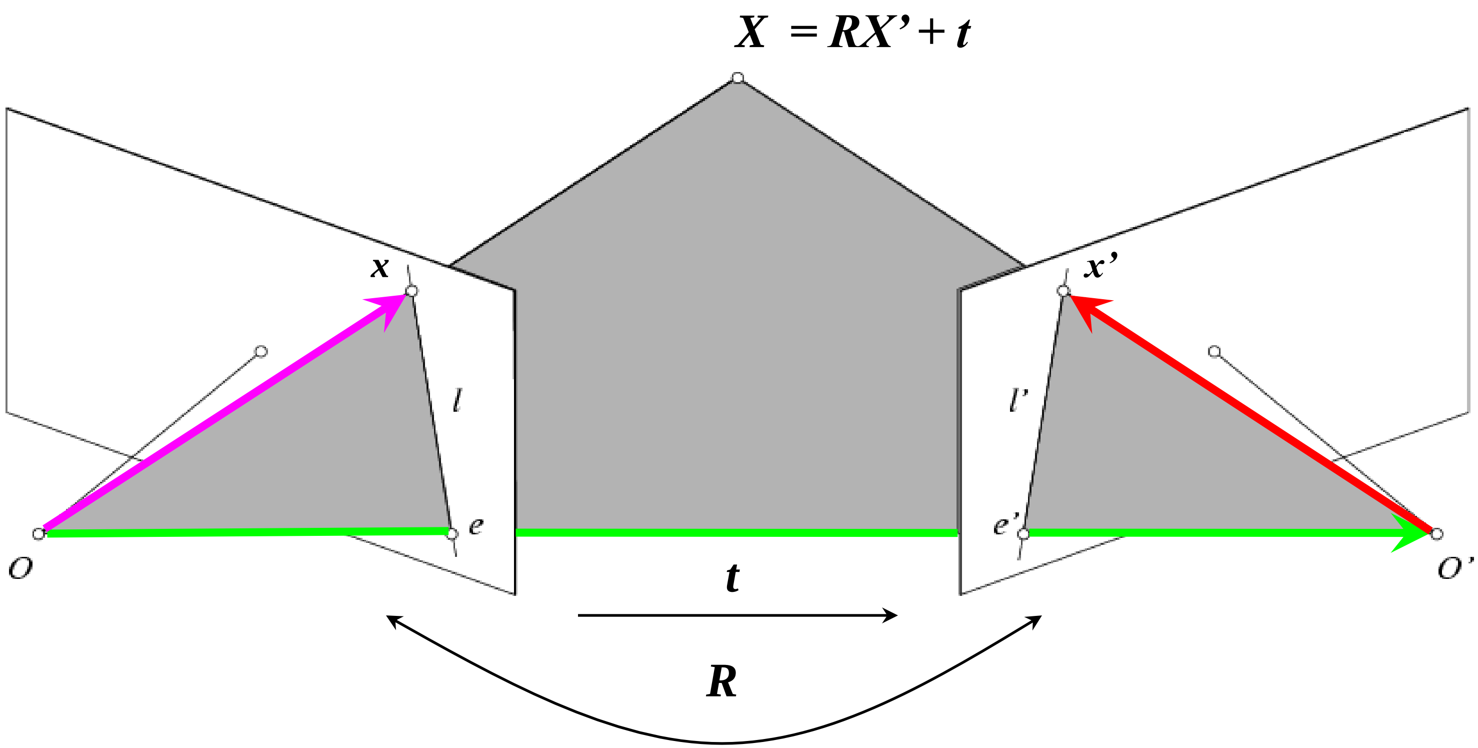
Epipolar constraint: Calibrated case



- Assume that the intrinsic and extrinsic parameters of the cameras are known
- We can multiply the projection matrix of each camera (and the image points) by the inverse of the calibration matrix to get normalized image coordinates
- We can also set the global coordinate system to the coordinate system of the first camera. Then the projection matrix of the first camera is **[I | 0]**.

Lazebnik

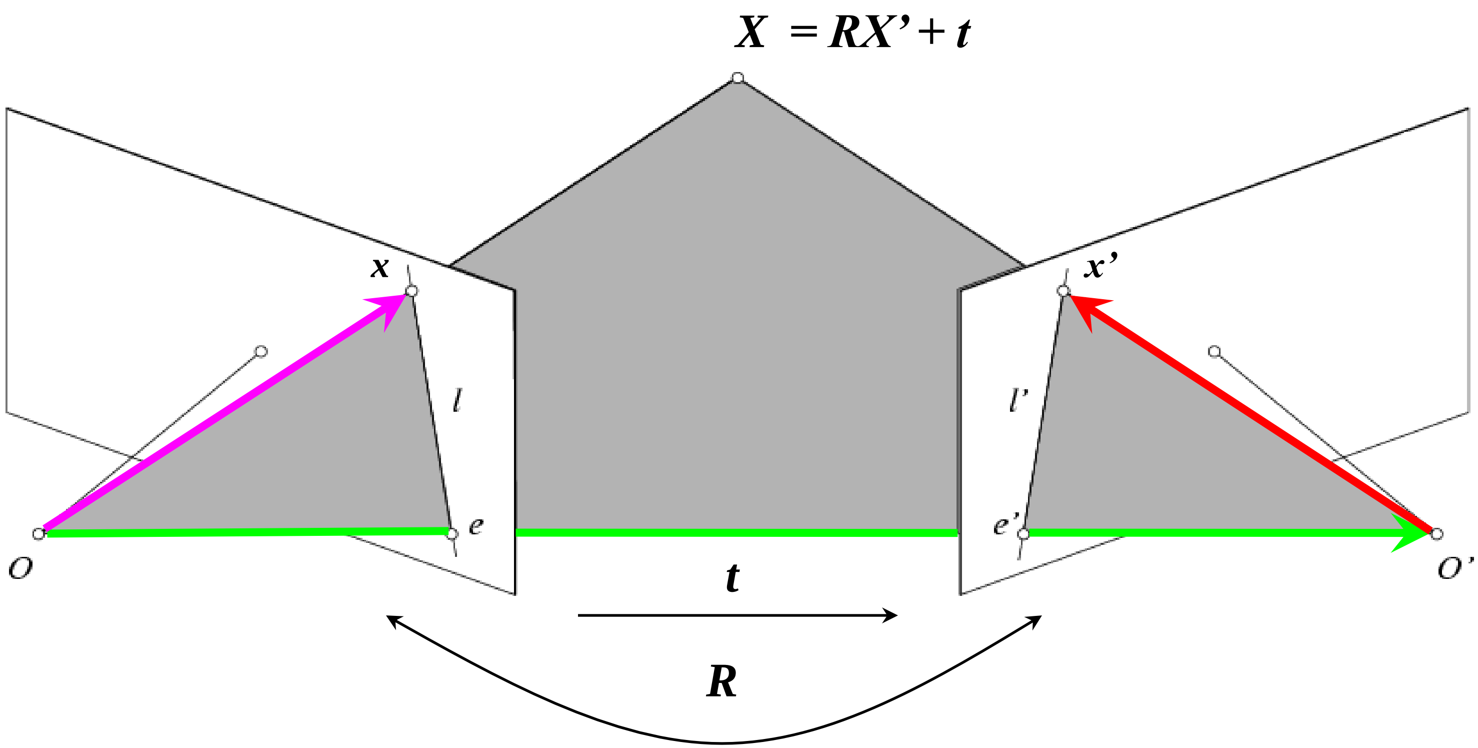
Epipolar constraint: Calibrated case



- X' is X in the second camera's coordinate system
- Projections of X and X' by homogeneous vectors x and x'
- The vectors ***x***, ***t***, and ***Rx'*** are coplanar

Lazebnik

From geometry to algebra



$$\boxed{\mathbf{X}} = \mathbf{R}\boxed{\mathbf{X}'} + \boxed{\mathbf{T}}$$

$$\underbrace{\mathbf{T} \times \mathbf{X}}_{\text{Normal to the plane}} =$$

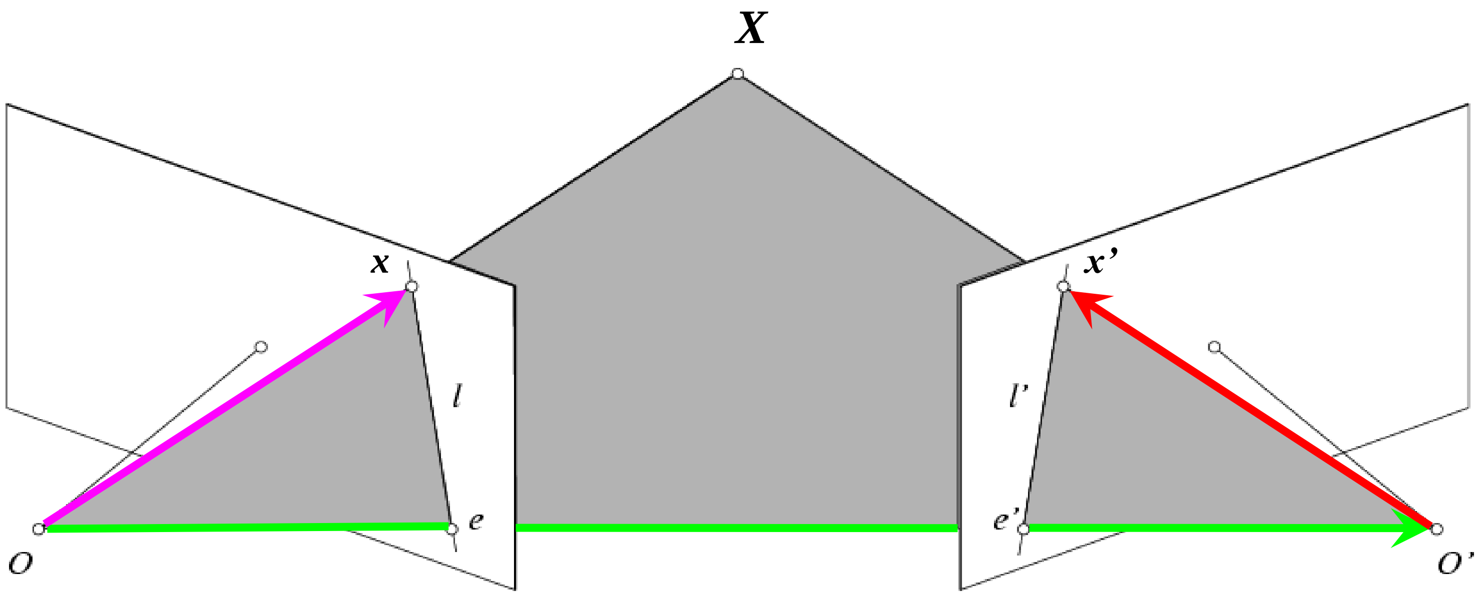
$$= \mathbf{T} \times \mathbf{R}\mathbf{X}'$$

$$\mathbf{X} \cdot (\mathbf{T} \times \mathbf{X}) = \mathbf{X} \cdot (\mathbf{T} \times \mathbf{R}\mathbf{X}')$$

$$= 0$$

Kristen Grauman

Epipolar constraint: Calibrated case

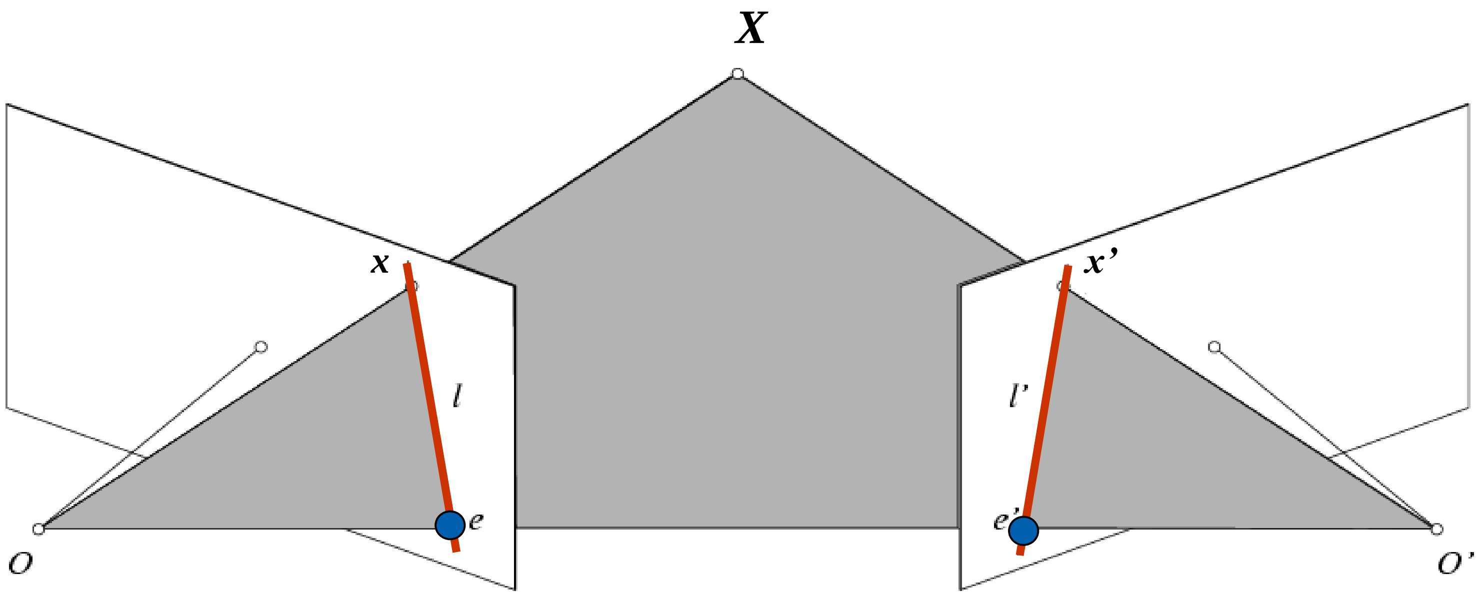


$$x \cdot [t \times (Rx')] = 0 \quad \Rightarrow \quad x^T E x' = 0 \quad \text{with} \quad E = [t_\times]R$$

Essential Matrix
(Longuet-Higgins, 1981)

Lazebnik

Epipolar constraint: Calibrated case

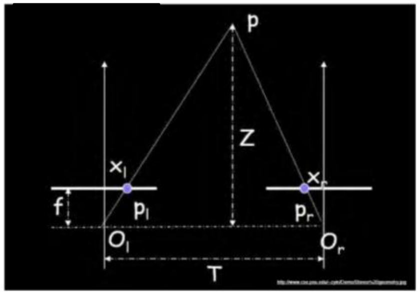


$$x \cdot [t \times (R x')] = 0 \quad \Rightarrow \quad x^T E x' = 0 \quad \text{with} \quad E = [t_\times] R$$

- E x' is the epipolar line associated with x' (l = E x')
- $E^T x$ is the epipolar line associated with x ($l' = E^T x$)
- E e' = 0 and $E^T e = 0$
- E is singular (rank two)
- E has five degrees of freedom

Lazebnik

Essential matrix example: parallel cameras



$$\mathbf{R} =$$

$$\mathbf{T} =$$

$$\mathbf{E} = [\mathrm{T}_{\mathbf{x}}]\mathbf{R} =$$

$$\mathbf{p} = [x, y, f]$$

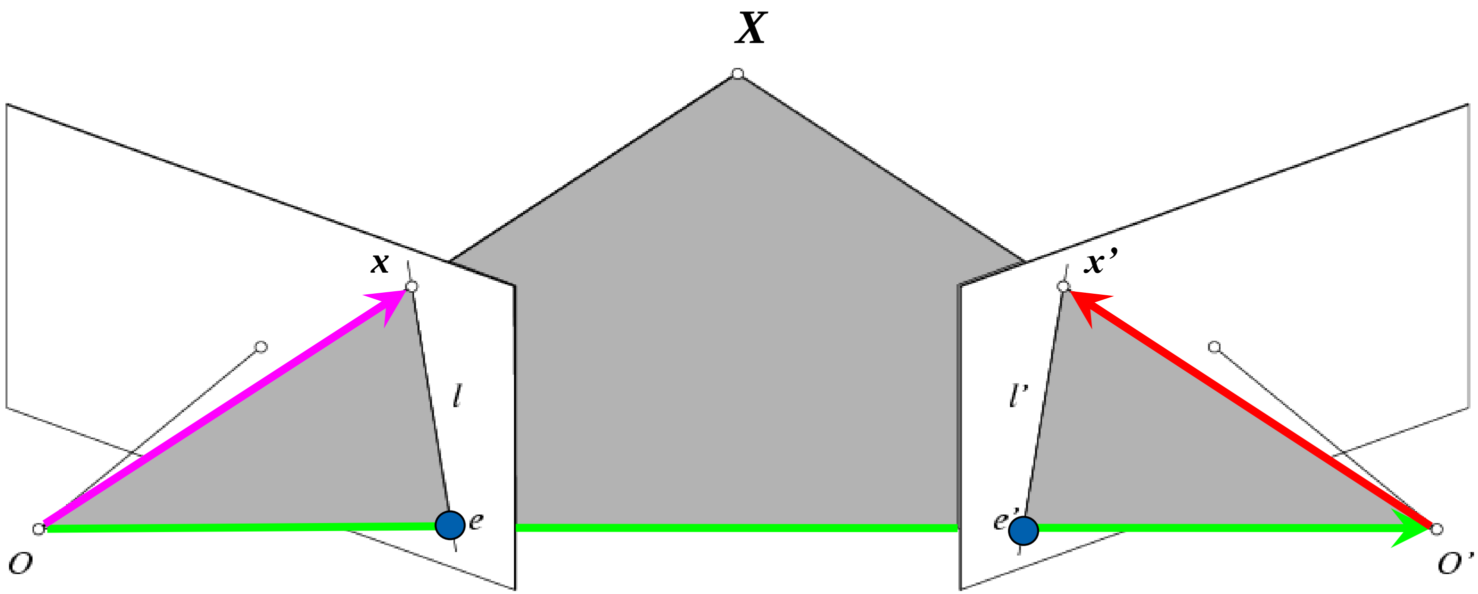
$$\mathbf{p}' = [x', y', f]$$

$$\mathbf{p}'^{\mathrm{T}}\mathbf{E}\mathbf{p} = 0$$

For the parallel cameras, image of any point must lie on same horizontal line in each image plane.

Kristen Grauman

Epipolar constraint: Uncalibrated case



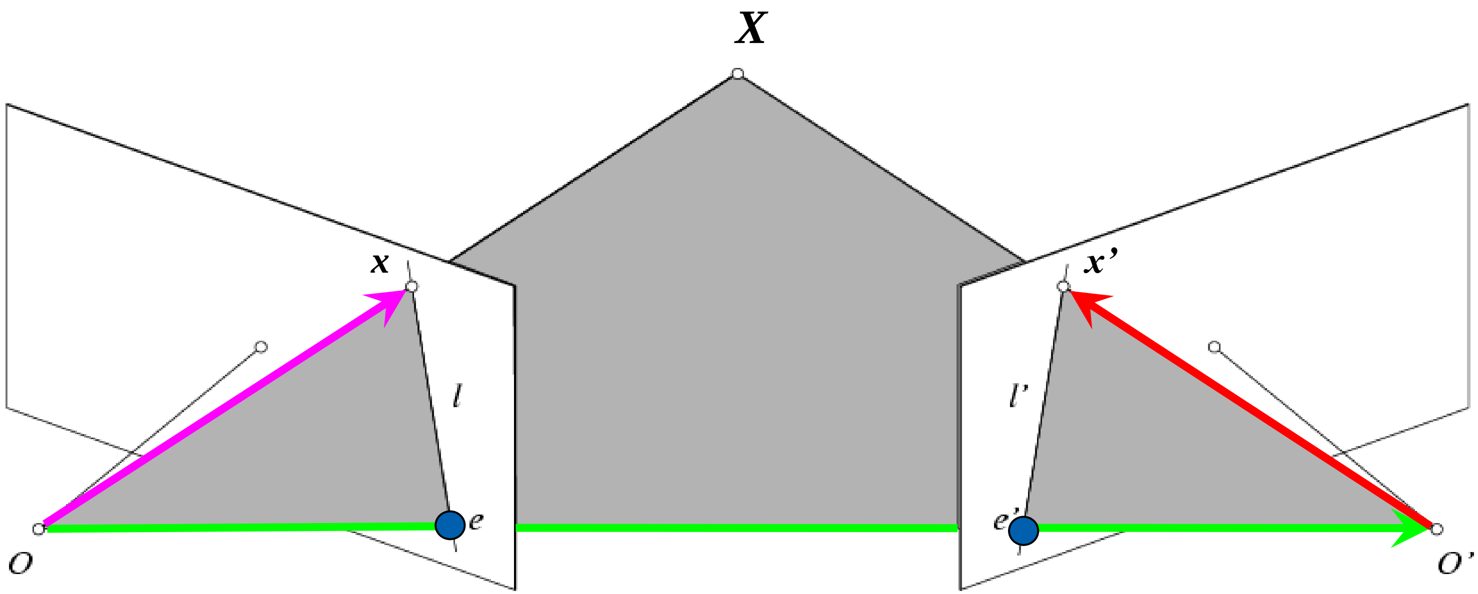
The calibration matrices K and K' of the two cameras are unknown

We can write the epipolar constraint in terms of unknown normalized coordinates:

$$\hat{x}^T E \hat{x}' = 0 \qquad\qquad x = K\hat{x}, \quad x' = K'\hat{x}'$$

Lazebnik

Epipolar constraint: Uncalibrated case



$$x^T F x' = 0 \quad \text{with} \quad F = K^{-T} E K'^{-1}$$

$$\hat{x} = K^{-1} x$$

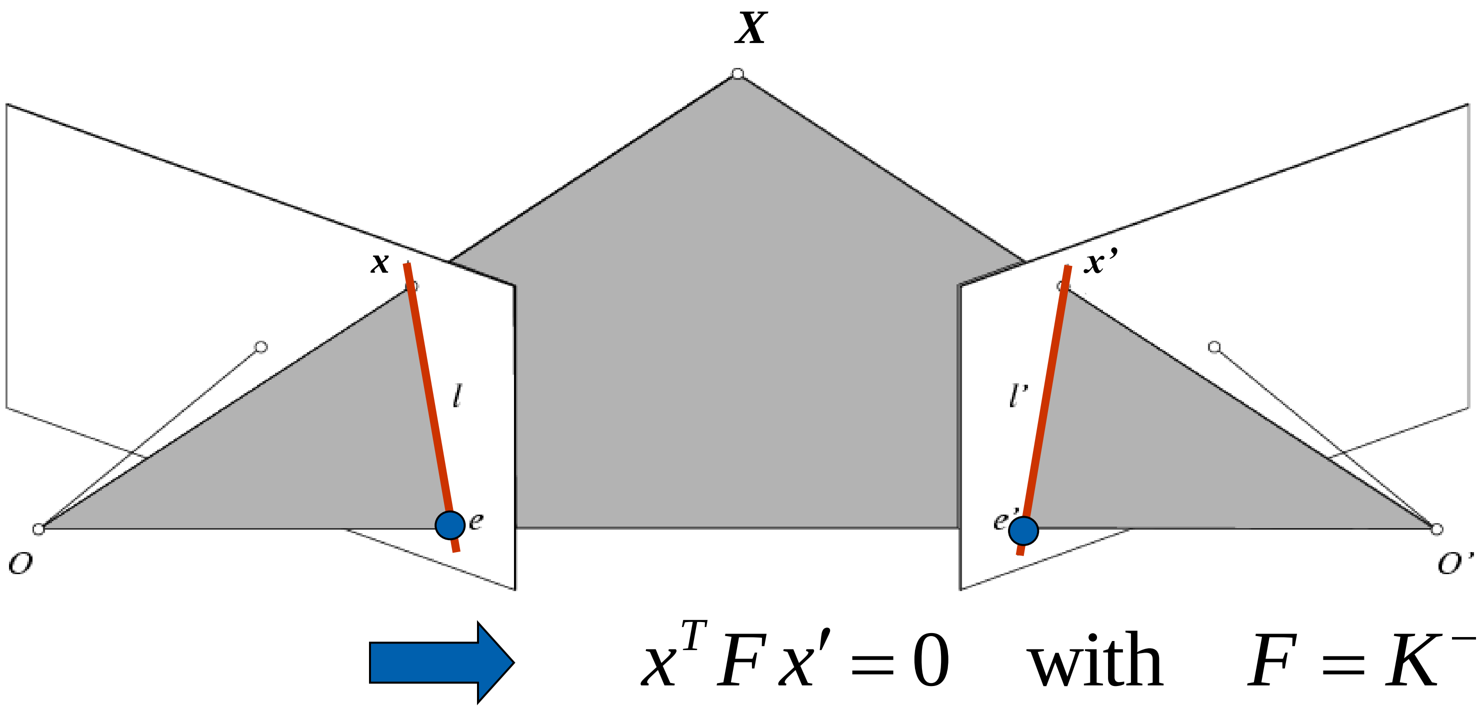
$$\hat{x}' = K'^{-1} x'$$

Fundamental Matrix
(Faugeras and Luong, 1992)

Lazebnik

Epipolar constraint: Uncalibrated case

UNIVERSITÄT BONN



$$x^T F x' = 0 \quad \text{with} \quad F = K^{-T} E K'^{-1}$$

- *F x'* is the epipolar line associated with *x'* (*l* = *F x'*)
- $F^T x$ is the epipolar line associated with *x* ($l' = F^T x$)
- *F e'* = 0 and $F^T e = 0$
- *F* is singular (rank two)
- *F* has seven degrees of freedom

Lazebnik

The eight-point algorithm

$\boldsymbol{x} = (u, v, 1)^T, \quad \boldsymbol{x'} = (u', v', 1)^T$

$$(u, v, 1)\begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix}\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = 0 \quad \Rightarrow \quad (uu', uv', u, vu', vv', v, u', v', 1)\begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix} = 0$$

$$\begin{pmatrix} u_1u_1' & u_1v_1' & u_1 & v_1u_1' & v_1v_1' & v_1 & u_1' & v_1' \\ u_2u_2' & u_2v_2' & u_2 & v_2u_2' & v_2v_2' & v_2 & u_2' & v_2' \\ u_3u_3' & u_3v_3' & u_3 & v_3u_3' & v_3v_3' & v_3 & u_3' & v_3' \\ u_4u_4' & u_4v_4' & u_4 & v_4u_4' & v_4v_4' & v_4 & u_4' & v_4' \\ u_5u_5' & u_5v_5' & u_5 & v_5u_5' & v_5v_5' & v_5 & u_5' & v_5' \\ u_6u_6' & u_6v_6' & u_6 & v_6u_6' & v_6v_6' & v_6 & u_6' & v_6' \\ u_7u_7' & u_7v_7' & u_7 & v_7u_7' & v_7v_7' & v_7 & u_7' & v_7' \\ u_8u_8' & u_8v_8' & u_8 & v_8u_8' & v_8v_8' & v_8 & u_8' & v_8' \end{pmatrix}\begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \end{pmatrix} = -\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \quad \Rightarrow$$

Minimize:

$$\sum_{i=1}^{N} (x_i^T F x_i')^2$$

under the constraint

$F_{33} = 1$

Lazebnik

The eight-point algorithm

- Meaning of error $\sum_{i=1}^{N}(x_i^T F x_i')^2$:

  sum of Euclidean distances between points x_i and epipolar lines Fx'_i (or points x'_i and epipolar lines $F^T x_i$) multiplied by a scale factor
- Nonlinear approach: minimize

$$\sum_{i=1}^{N}\left[\mathrm{d}^2(x_i, F x_i') + \mathrm{d}^2(x_i', F^T x_i)\right]$$

Lazebnik

Problem with eight-point algorithm

$$\begin{pmatrix} u_1u_1' & u_1v_1' & u_1 & v_1u_1' & v_1v_1' & v_1 & u_1' & v_1' \\ u_2u_2' & u_2v_2' & u_2 & v_2u_2' & v_2v_2' & v_2 & u_2' & v_2' \\ u_3u_3' & u_3v_3' & u_3 & v_3u_3' & v_3v_3' & v_3 & u_3' & v_3' \\ u_4u_4' & u_4v_4' & u_4 & v_4u_4' & v_4v_4' & v_4 & u_4' & v_4' \\ u_5u_5' & u_5v_5' & u_5 & v_5u_5' & v_5v_5' & v_5 & u_5' & v_5' \\ u_6u_6' & u_6v_6' & u_6 & v_6u_6' & v_6v_6' & v_6 & u_6' & v_6' \\ u_7u_7' & u_7v_7' & u_7 & v_7u_7' & v_7v_7' & v_7 & u_7' & v_7' \\ u_8u_8' & u_8v_8' & u_8 & v_8u_8' & v_8v_8' & v_8 & u_8' & v_8' \end{pmatrix} \begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \end{pmatrix} = - \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Lazebnik

Problem with eight-point algorithm

UNIVERSITÄT BONN

$$
\begin{bmatrix}
250906.36 & 183269.57 & 921.81 & 200931.10 & 146766.13 & 738.21 & 272.19 & 198.81 \\
2692.28 & 131633.03 & 176.27 & 6196.73 & 302975.59 & 405.71 & 15.27 & 746.79 \\
416374.23 & 871684.30 & 935.47 & 408110.89 & 854384.92 & 916.90 & 445.10 & 931.81 \\
191183.60 & 171759.40 & 410.27 & 416435.62 & 374125.90 & 893.65 & 465.99 & 418.65 \\
48988.86 & 30401.76 & 57.89 & 298604.57 & 185309.58 & 352.87 & 846.22 & 525.15 \\
164786.04 & 546559.67 & 813.17 & 1998.37 & 6628.15 & 9.86 & 202.65 & 672.14 \\
116407.01 & 2727.75 & 138.89 & 169941.27 & 3982.21 & 202.77 & 838.12 & 19.64 \\
135384.58 & 75411.13 & 198.72 & 411350.03 & 229127.78 & 603.79 & 681.28 & 379.48
\end{bmatrix}
\begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \end{pmatrix}
= -\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}
$$

- Poor numerical conditioning
- Can be fixed by rescaling the data

Lazebnik

The normalized eight-point algorithm

(Hartley, 1995)

- Center the image data at the origin, and scale it so the mean squared distance between the origin and the data points is 2 pixels
- Use the eight-point algorithm to compute F from the normalized points
- Enforce the rank-2 constraint (for example, take SVD of F and throw out the smallest singular value)
- Transform fundamental matrix back to original units: if T and T' are the normalizing transformations in the two images, than the fundamental matrix in original coordinates is $T^T F T'$

Lazebnik

Comparison of estimation algorithms

	8-point	Normalized 8-point	Nonlinear least squares
Av. Dist. 1	2.33 pixels	0.92 pixel	0.86 pixel
Av. Dist. 2	2.18 pixels	0.85 pixel	0.80 pixel

Lazebnik

From epipolar geometry to camera calibration

- Estimating the fundamental matrix is known as "weak calibration"
- If we know the calibration matrices of the two cameras, we can estimate the essential matrix: $E = K^T F K'$
- The essential matrix gives us the relative rotation and translation between the cameras, or their extrinsic parameters

Lazebnik