$$\begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} \times \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} = \begin{pmatrix} a_y b_z - b_y a_z \\ a_z b_x - b_z a_x \\ a_x b_y - b_x a_y \end{pmatrix}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

CS 532: 3D Computer Vision
Lecture 2

Enrique Dunn
edunn@stevens.edu
Lieb 310

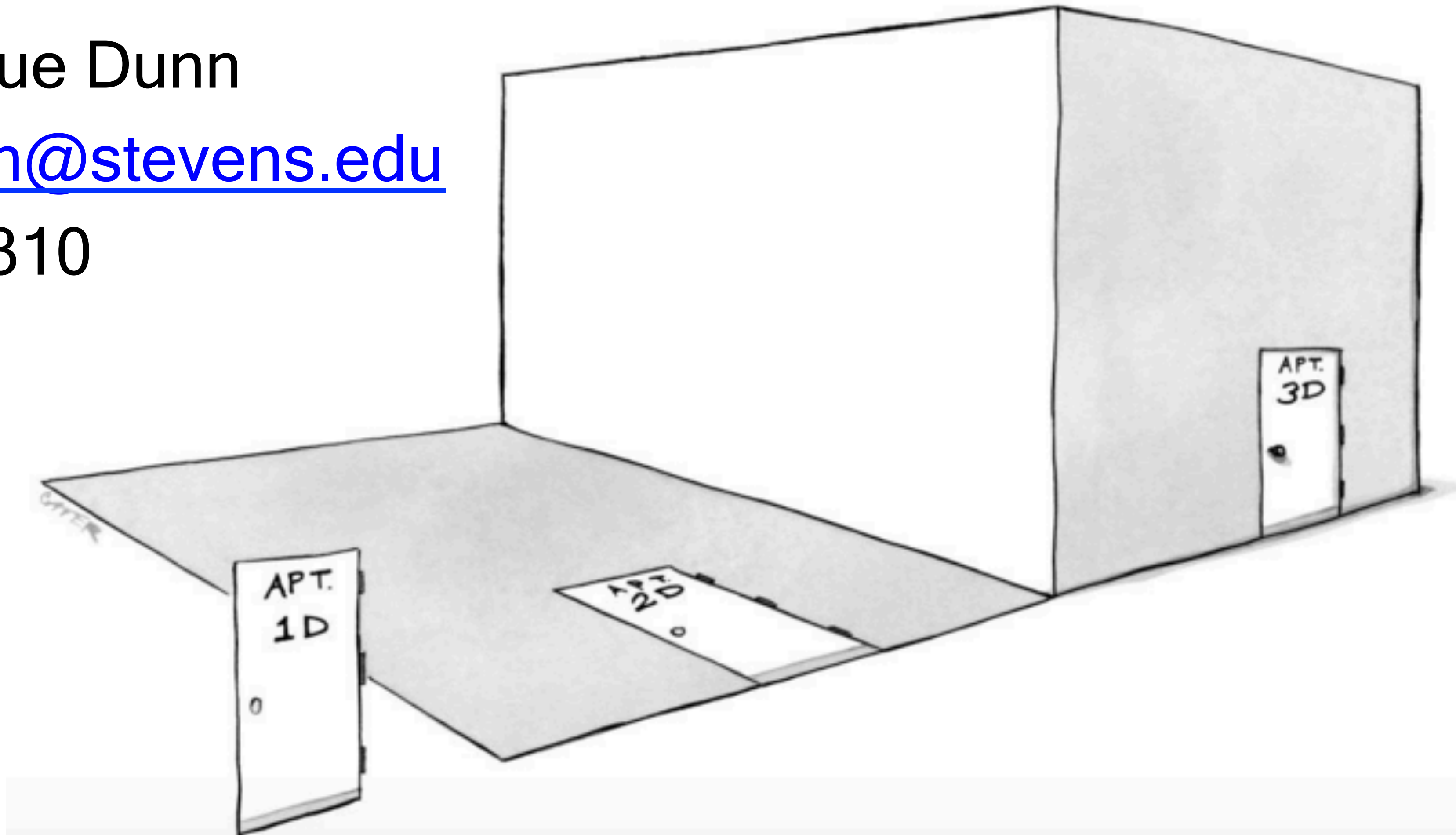


Image Formation

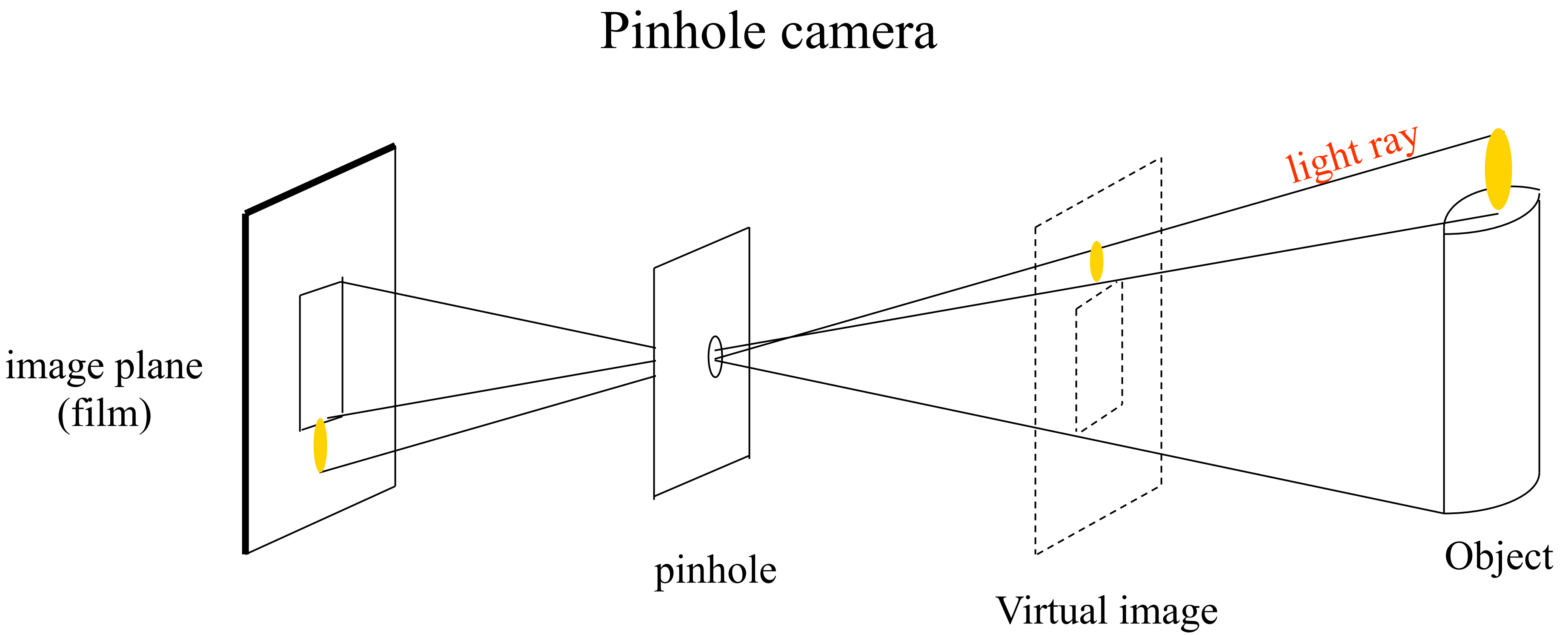
Based on slides by John Oliensis

Lecture Outline

- Single View Geometry
- 2D projective transformations
  – Homographies
- Robust estimation
  – RANSAC
- Radial distortion
- Two-view geometry

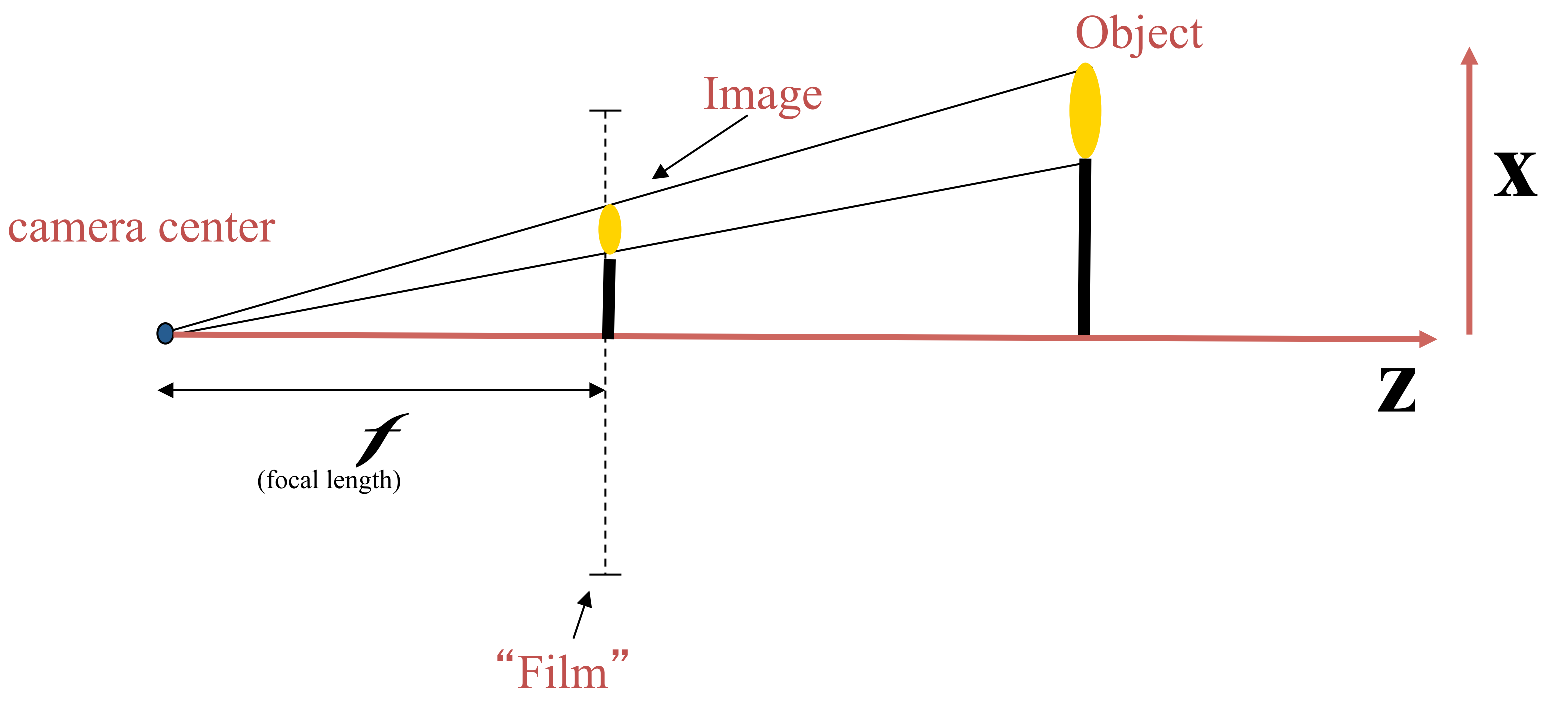
Based on slides by R. Hartley, A. Zisserman, M. Pollefeys and S. Seitz

Image Formation

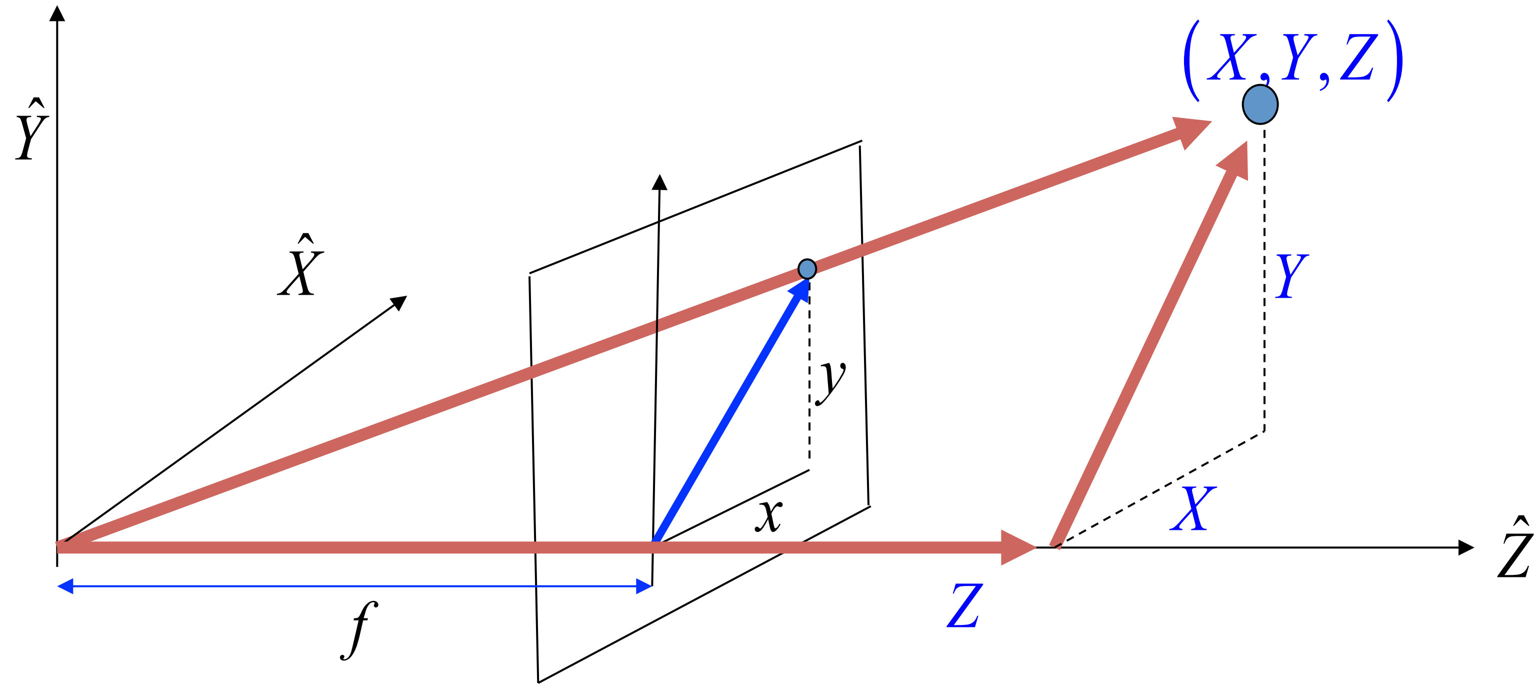


Projection Equation

- 2D world ➔ 1D image



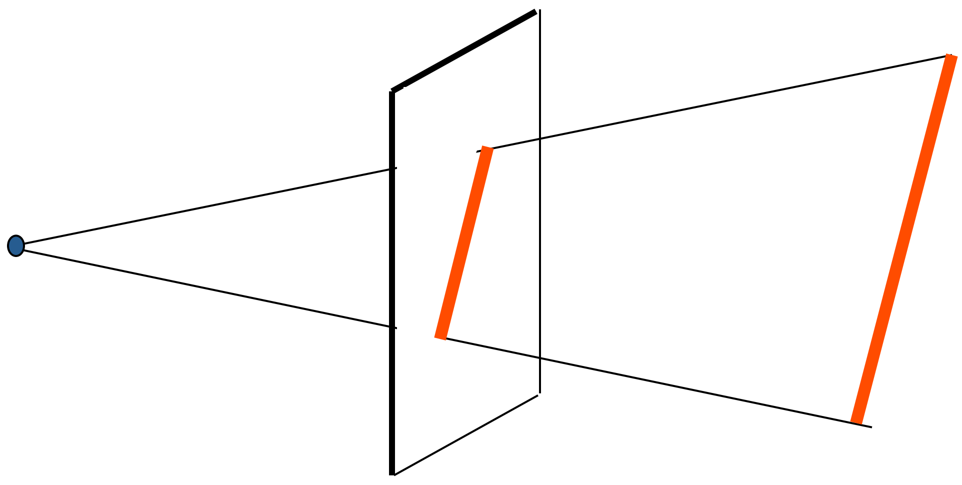
Projection Equation: 3D



Similar triangles: $$\frac{x}{X} = \frac{y}{Y} = \frac{f}{Z} \longrightarrow \boxed{(x,y) = \frac{f}{Z}(X,Y)}$$

Perspective Projection: Properties

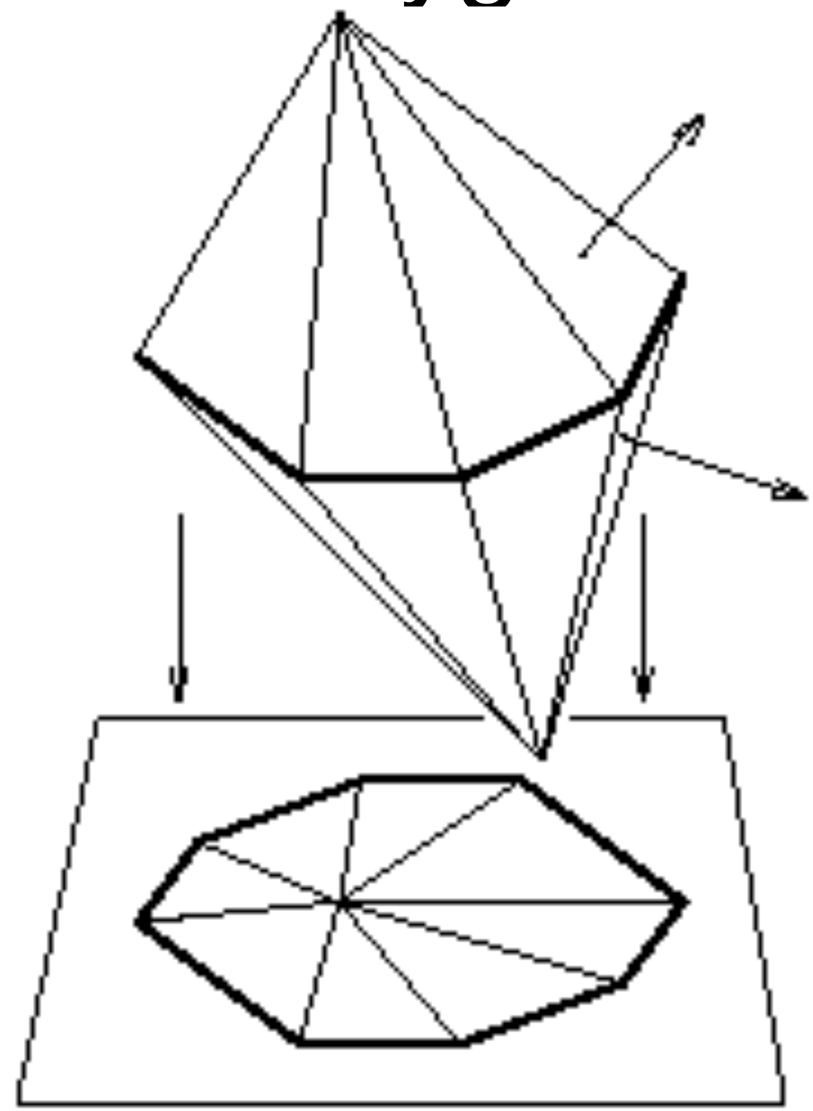
- 3D points ➔ image points
- 3D straight lines ➔ image straight lines



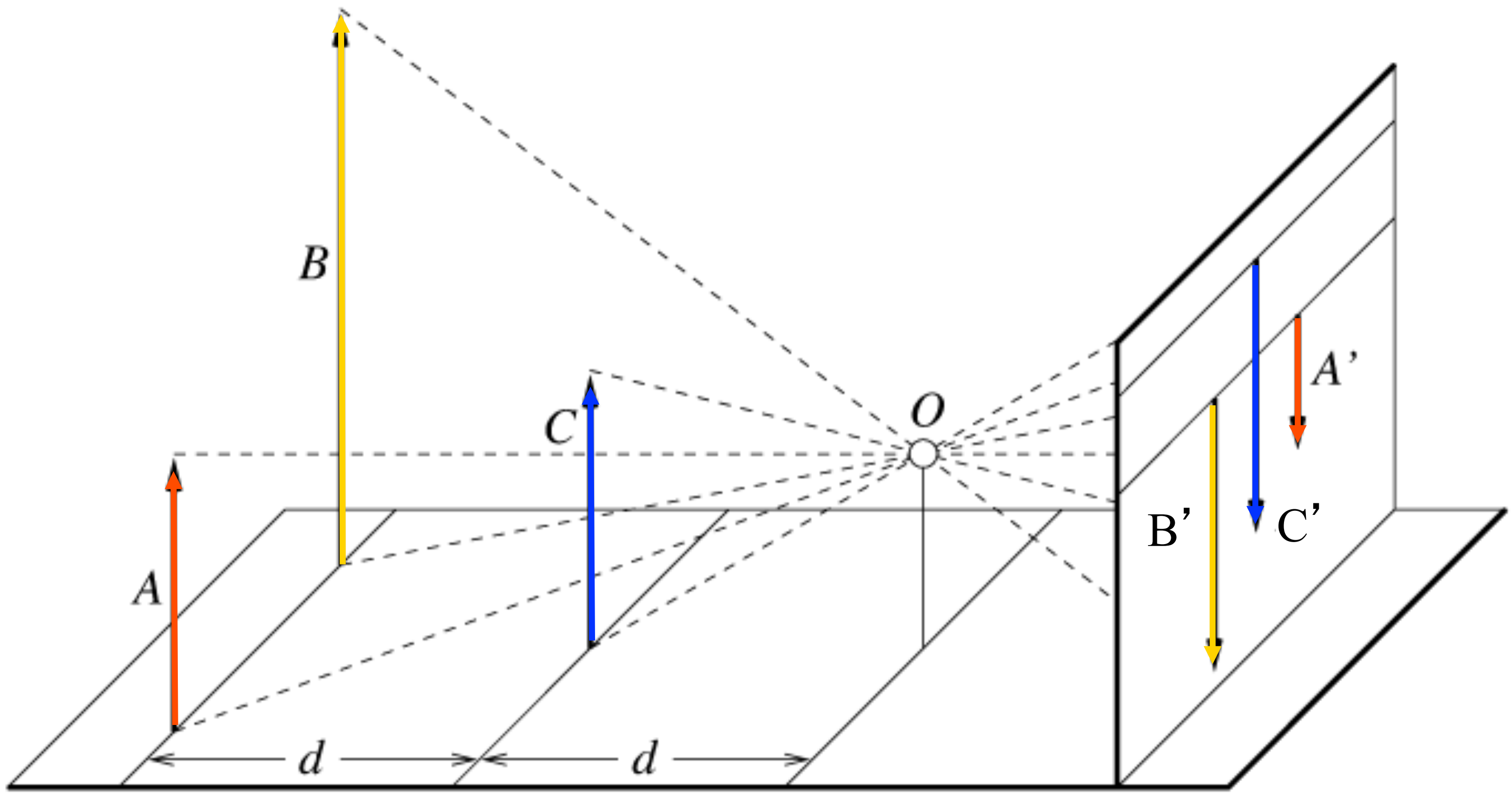
- 3D Polygons ➔ image polygons

Polyhedra Project to Polygons

(since lines project to lines)



Properties: Distant objects are smaller



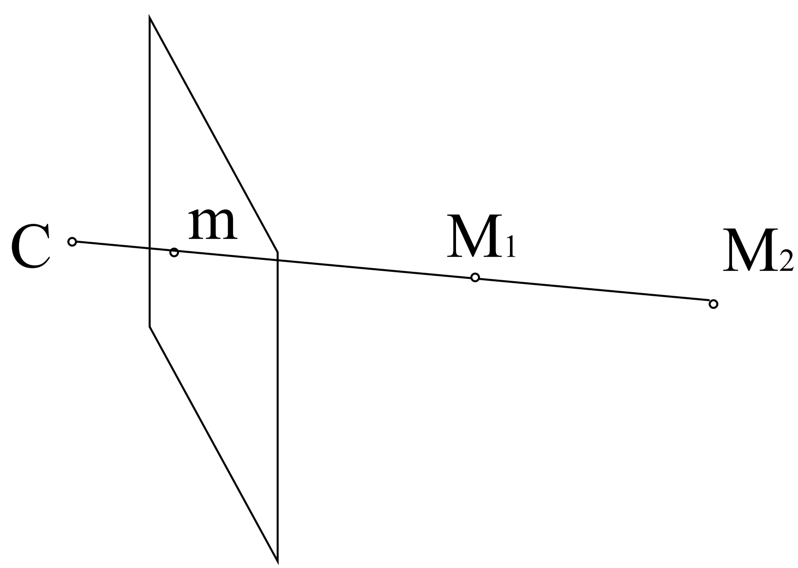
Single View Geometry

Richard Hartley and Andrew Zisserman
Marc Pollefeys

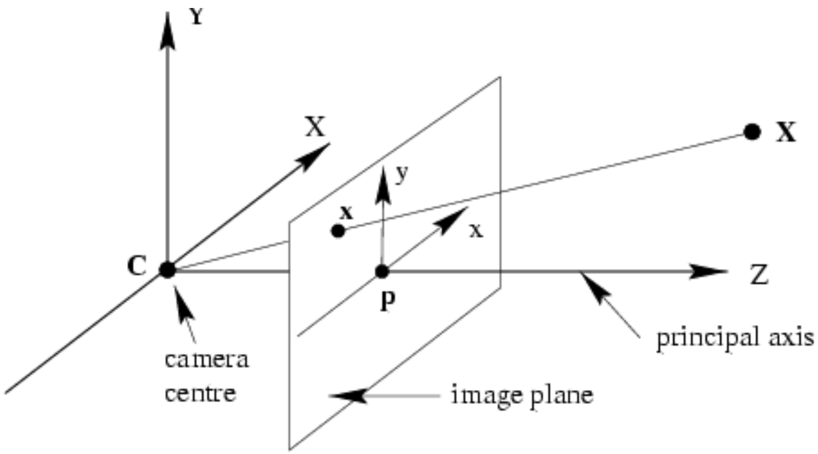
Modified by Philippos Mordohai

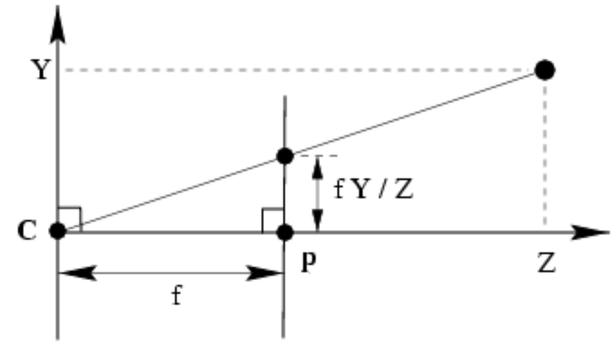
Homogeneous Coordinates

- 3-D points represented as 4-D vectors $(X\ Y\ Z\ 1)^T$
- Equality defined up to scale
  - $(X\ Y\ Z\ 1)^T \sim (WX\ WY\ WZ\ W)^T$
- Useful for perspective projection → makes equations linear



Pinhole camera model



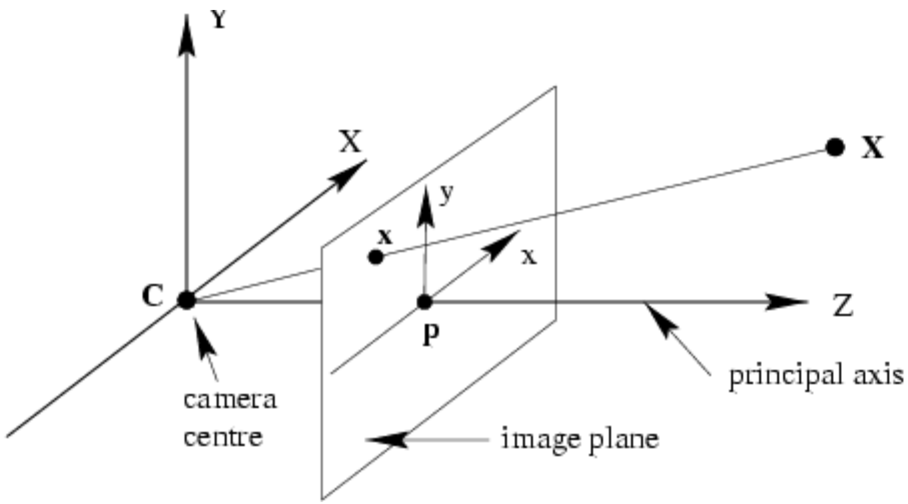


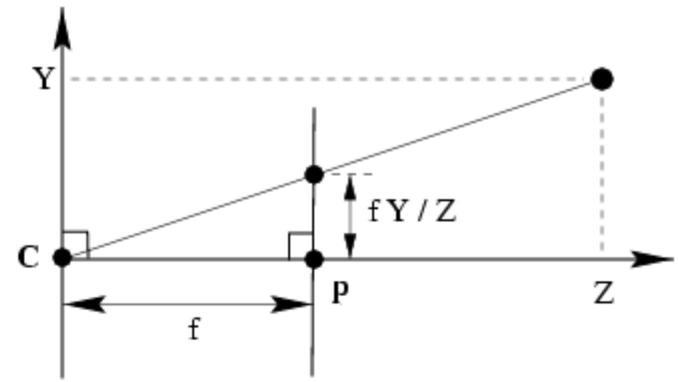
$$(X,Y,Z)^T \mapsto (fX/Z, fY/Z)^T$$

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

linear projection in homogeneous coordinates!

The Pinhole Camera



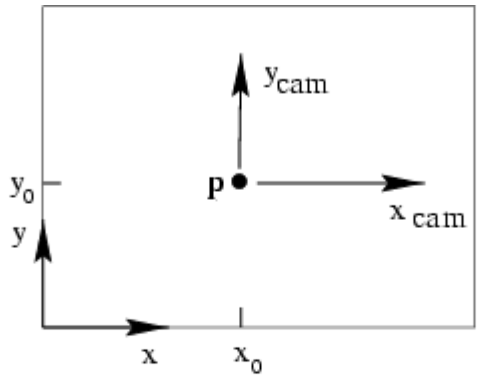


$$x = \frac{fX}{Z}$$

$$y = \frac{fY}{Z}$$

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Principal Point Offset

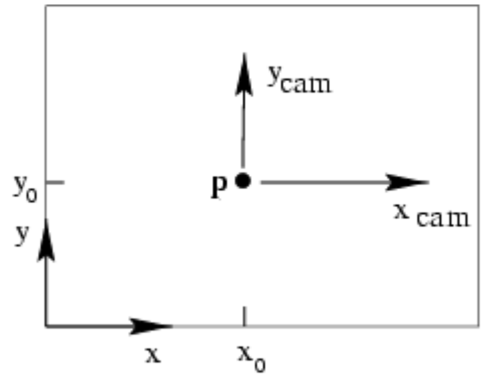


$$(X,Y,Z)^T \mapsto (fX/Z + p_x, fY/Z + p_y)^T$$

$(p_x, p_y)^T$ principal point

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

Principal Point Offset



$$\mathrm{x} = \mathrm{K}\left[\mathrm{I} \mid 0\right]\mathrm{X}_{\mathrm{cam}}$$

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

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

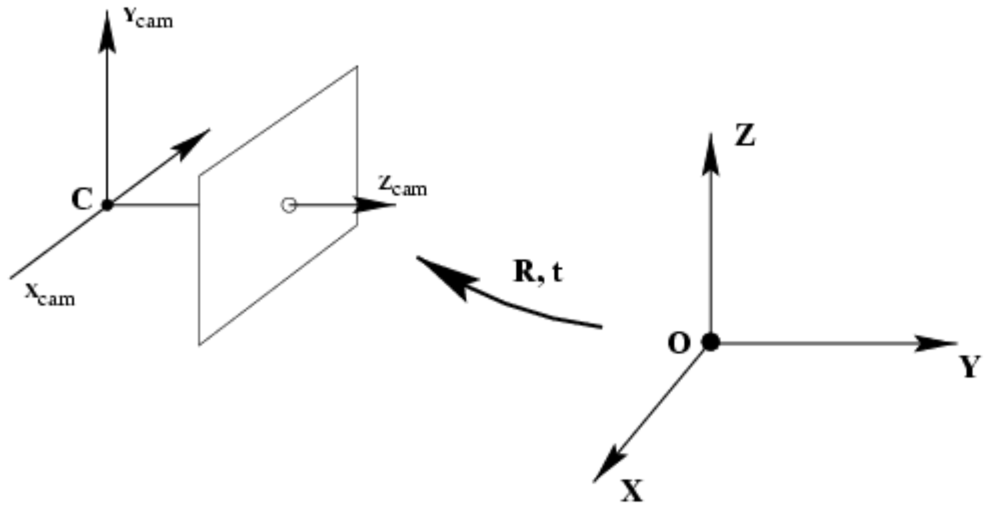
Hands On: Image Formation

- For a 640 by 480 image with focal length equal to 640 pixels, find 3D points that are marginally visible at the four borders of the image

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

- Increase and decrease the focal length. What happens?

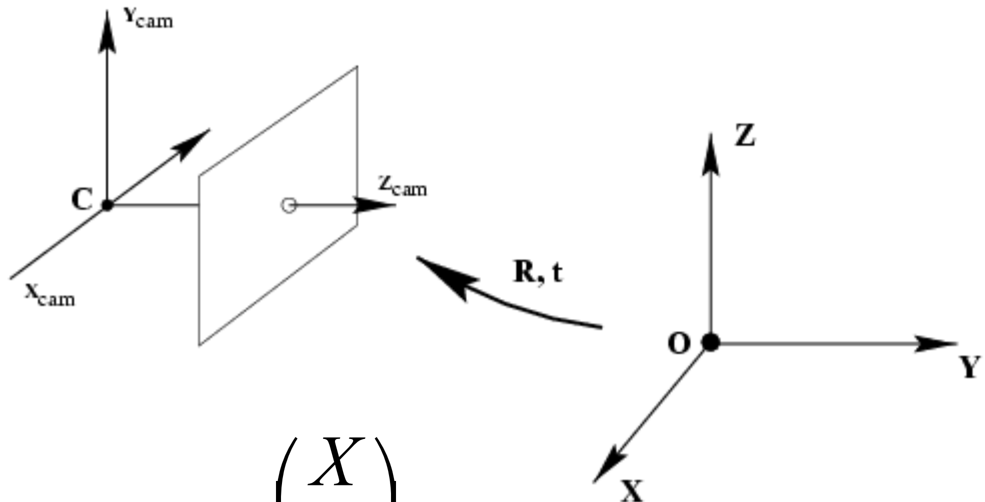
Camera Rotation and Translation



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

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

Camera Rotation and Translation



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

$$x = K[I \mid 0]X_{cam}$$

$$x = KR[I \mid -\tilde{C}]X$$

$$x = PX$$

$$P = K[R \mid t]$$

$$t = -R\tilde{C}$$

Intrinsic Parameters

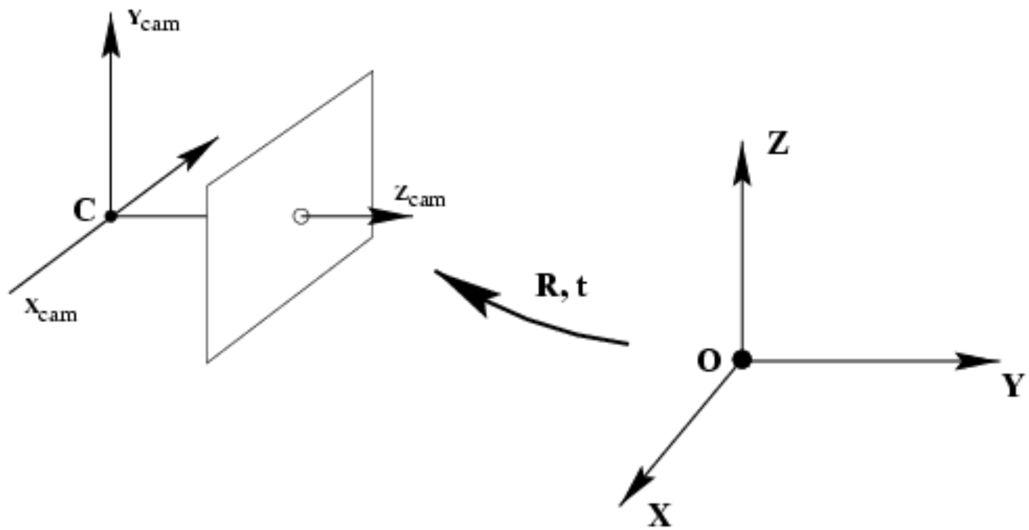
$$\mathbf{K} = \begin{bmatrix} f_x & s & c_x \\ & f_y & c_y \\ & & 1 \end{bmatrix} \quad \text{or} \quad \mathbf{K} = \begin{bmatrix} af & f\cos(s) & u_o \\ & f & v_o \\ & & 1 \end{bmatrix}$$

- Camera deviates from pinhole
  - *s*: skew
  - $f_x \neq f_y$: different magnification in *x* and *y*
  - $(c_x\ c_y)$: optical axis does not pierce image plane exactly at the center

$$\mathbf{K} = \begin{bmatrix} \gamma f & sf & x_0 \\ & f & y_0 \\ & & 1 \end{bmatrix}$$

- Usually:
  - rectangular pixels: $s = 0$
  - square pixels: $f_x = f_y$
  - principal point known: $(c_x, c_y) = \left(\frac{w}{2}, \frac{h}{2}\right)$

Extrinsic Parameters



Scene motion

$$M = \begin{bmatrix} R_{(3x3)} & t_{(3x1)} \\ 0_{(1x3)} & 1 \end{bmatrix}$$

Camera motion

$$M' = \begin{bmatrix} R^{T}{}_{(3x3)} & -(R^{T}t)_{3x1} \\ 0_{(1x3)} & 1 \end{bmatrix}$$

Projection matrix

- Includes coordinate transformation and camera intrinsic parameters

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

- Everything we need to know about a pinhole camera
- Unambiguous
- Can be decomposed into parameters

Projection matrix

- Mapping from 2-D to 3-D is a function of internal and external parameters

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R^{\top} \mid -R^{\top} t \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\lambda x = K \begin{bmatrix} R^{\top} \mid -R^{\top} t \end{bmatrix} X$$

$$\lambda x = PX$$

Hands On: Camera Motion

- Choose a few 3D points visible to a camera at the origin. (f=500, w=500, h=500)
- Now, move the camera by 2 units of length on the z axis. What happens to the images of the points?
- Rotate the points by 45 degrees about the z axis of the camera and then translate them by 5 units on the z axis away from the camera. What are the new images of the points?

Projective Transformations in 2D

Definition:

A *projectivity* is an invertible mapping h from P^2 to itself such that three points x_1,x_2,x_3 lie on the same line if and only if $h(x_1),h(x_2),h(x_3)$ do.

Theorem:

A mapping $h:P^2 \rightarrow P^2$ is a projectivity if and only if there exist a non-singular 3x3 matrix $\mathbf{H}$ such that for any point in P^2 reprented by a vector x it is true that $h(x)=\mathbf{H}x$

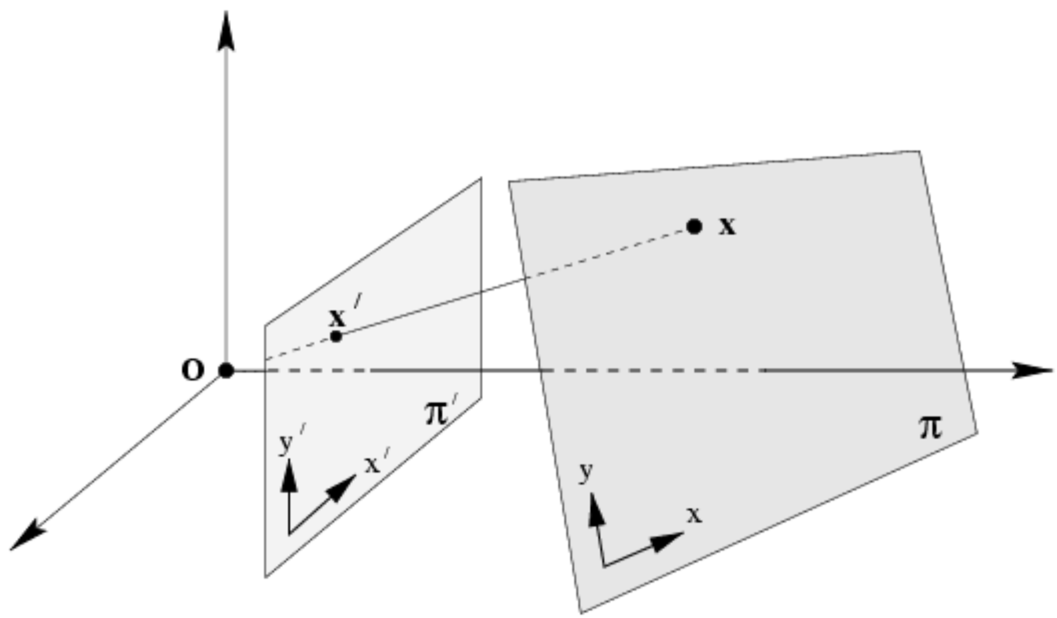
Definition: Projective transformation

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad \text{or} \quad \mathrm{x}' = \mathbf{H}\,\mathrm{x}$$

8DOF

projectivity=collineation=projective transformation=homography

Mapping between planes



central projection may be expressed by x'=Hx

(application of theorem)

Removing Projective Distortion

select four points in a plane with known coordinates

$$x' = \frac{x'_1}{x'_3} = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}} \qquad y' = \frac{x'_2}{x'_3} = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}$$

$$x'\left(h_{31}x + h_{32}y + h_{33}\right) = h_{11}x + h_{12}y + h_{13}$$
$$y'\left(h_{31}x + h_{32}y + h_{33}\right) = h_{21}x + h_{22}y + h_{23}$$

(linear in h_{ij})

(2 constraints/point, 8DOF $\Rightarrow$ 4 points needed)

Remarks: no calibration at all necessary,
better ways to compute (see later)

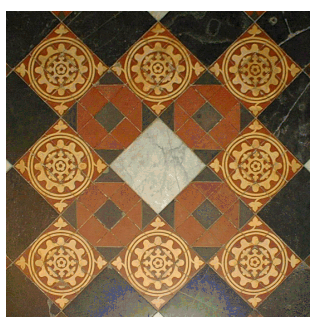
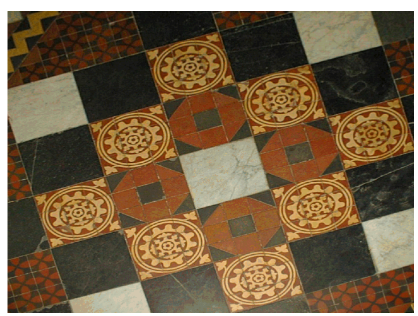
A Hierarchy of Transformations

Projective linear group

- Affine group (last row (0,0,1))
  - Euclidean group (upper left 2x2 orthogonal)
    - Oriented Euclidean group (upper left 2x2 det 1)

Alternatively, characterize transformation in terms of elements or quantities that are preserved or *invariant*

e.g. Euclidean transformations leave distances unchanged


Class I: Isometries

(*iso*=same, *metric*=measure)

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} \varepsilon\cos\theta & -\sin\theta & t_x \\ \varepsilon\sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \qquad \varepsilon = \pm 1$$

orientation preserving: $\varepsilon = 1$
orientation reversing: $\varepsilon = -1$

$$\mathrm{x}' = \mathbf{H}_E\,\mathrm{x} = \begin{bmatrix} \mathbf{R} & \mathrm{t} \\ 0^\mathsf{T} & 1 \end{bmatrix} \mathrm{x} \qquad \mathbf{R}^\mathsf{T}\mathbf{R} = \mathbf{I}$$

3DOF (1 rotation, 2 translation)

special cases: pure rotation, pure translation

Invariants: length, angle, area

Class II: Similarities

(*isometry* + *scale*)

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} s\cos\theta & -s\sin\theta & t_x \\ s\sin\theta & s\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$\mathrm{x}' = \mathbf{H}_S\,\mathrm{x} = \begin{bmatrix} s\mathbf{R} & \mathrm{t} \\ 0^\mathsf{T} & 1 \end{bmatrix} \mathrm{x}$$

$$\mathbf{R}^\mathsf{T}\mathbf{R} = \mathbf{I}$$

4DOF (1 scale, 1 rotation, 2 translation)

also know as *equi-form* (shape preserving)

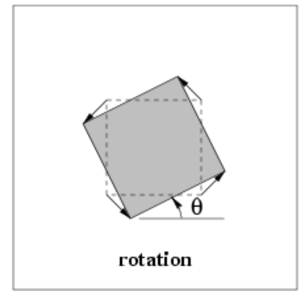
metric structure = structure up to similarity (in literature)

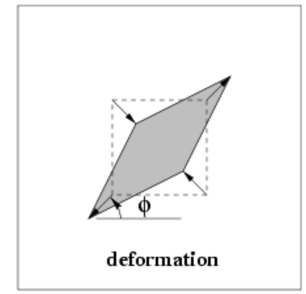
Invariants: ratios of length, angle, ratios of areas, parallel lines

Class III: Affine Transformations

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$\mathbf{x}' = \mathbf{H}_A \, \mathbf{x} = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ 0^{\mathsf{T}} & 1 \end{bmatrix} \mathbf{x}$$





$$\mathbf{A} = \mathbf{R}(\theta)\mathbf{R}(-\phi)\mathbf{D}\mathbf{R}(\phi) \qquad \mathbf{D} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

6DOF (2 scale, 2 rotation, 2 translation)

non-isotropic scaling! (2DOF: scale ratio and orientation)

Invariants: parallel lines, ratios of parallel lengths,
ratios of areas

Class VI: Projective Transformations

$$\mathrm{x}' = \mathbf{H}_P\,\mathrm{x} = \begin{bmatrix} \mathbf{A} & \mathrm{t} \\ \mathrm{v}^{\mathsf{T}} & v \end{bmatrix} \mathrm{x} \qquad \mathrm{v} = (v_1, v_2)^{\mathsf{T}}$$

8DOF (2 scale, 2 rotation, 2 translation, 2 line at infinity)

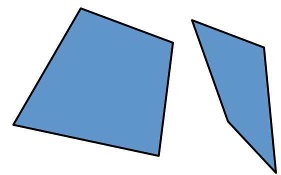
Action is non-homogeneous over the plane

Invariants: cross-ratio of four points on a line
(ratio of ratios)

Overview of Transformations

Projective
8dof

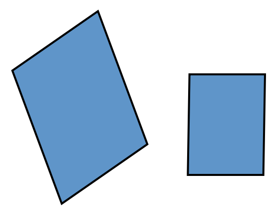
$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$



Concurrency, collinearity, order of contact (intersection, tangency, inflection, etc.), cross ratio

Affine
6dof

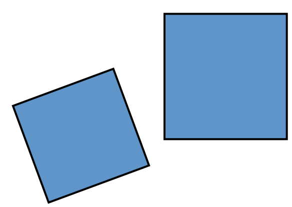
$$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$$



Parallellism, ratio of areas, ratio of lengths on parallel lines (e.g midpoints), linear combinations of vectors (centroids).
The line at infinity l_∞

Similarity
4dof

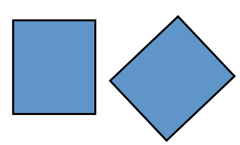
$$\begin{bmatrix} sr_{11} & sr_{12} & t_x \\ sr_{21} & sr_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$$



Ratios of lengths, angles.
The circular points I,J

Euclidean
3dof

$$\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$$



lengths, areas.

Homework 1

Warp the basketball court from this image to a new image so that it appears as if the new image was taken from directly above

What are we missing?

Image Warping

Slides by Steve Seitz

Image Transformations

image filtering: change ***range*** of image

$$g(x) = T(f(x))$$

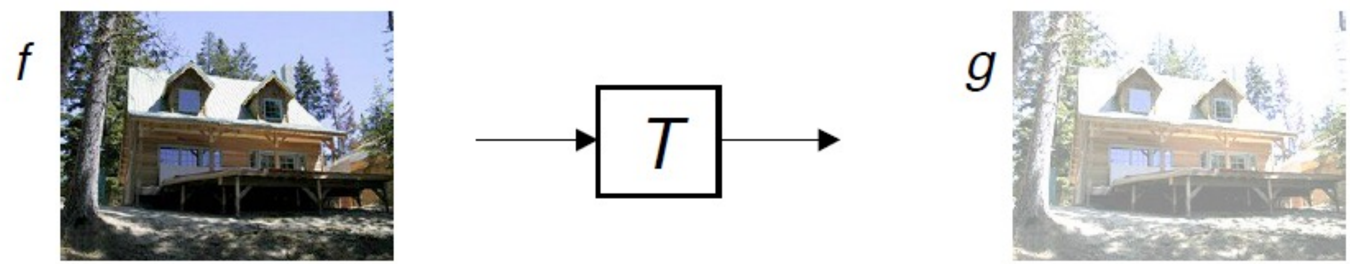
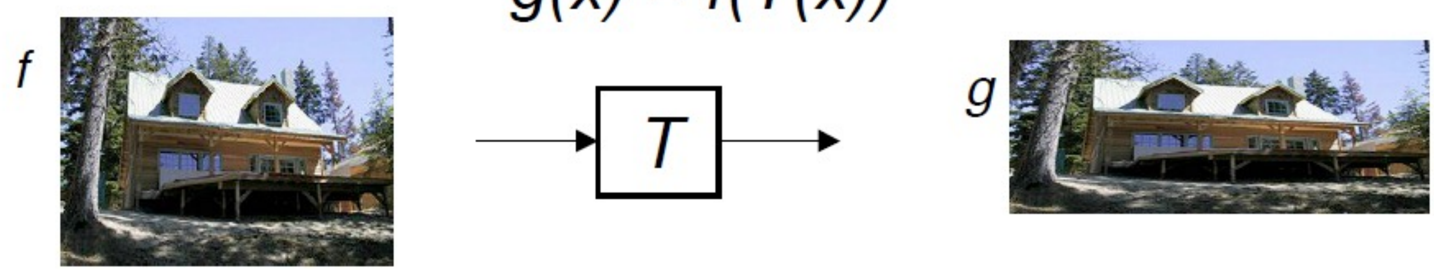
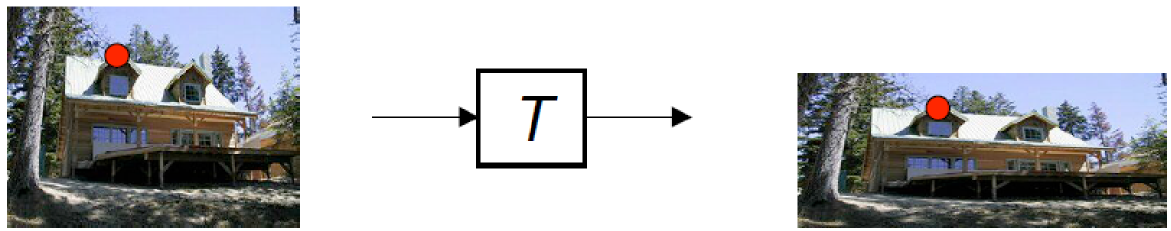


image warping: change ***domain*** of image

$$g(x) = f(T(x))$$



Parametric (Global) Warping



p = (x,y) **p’** = (x’,y’)

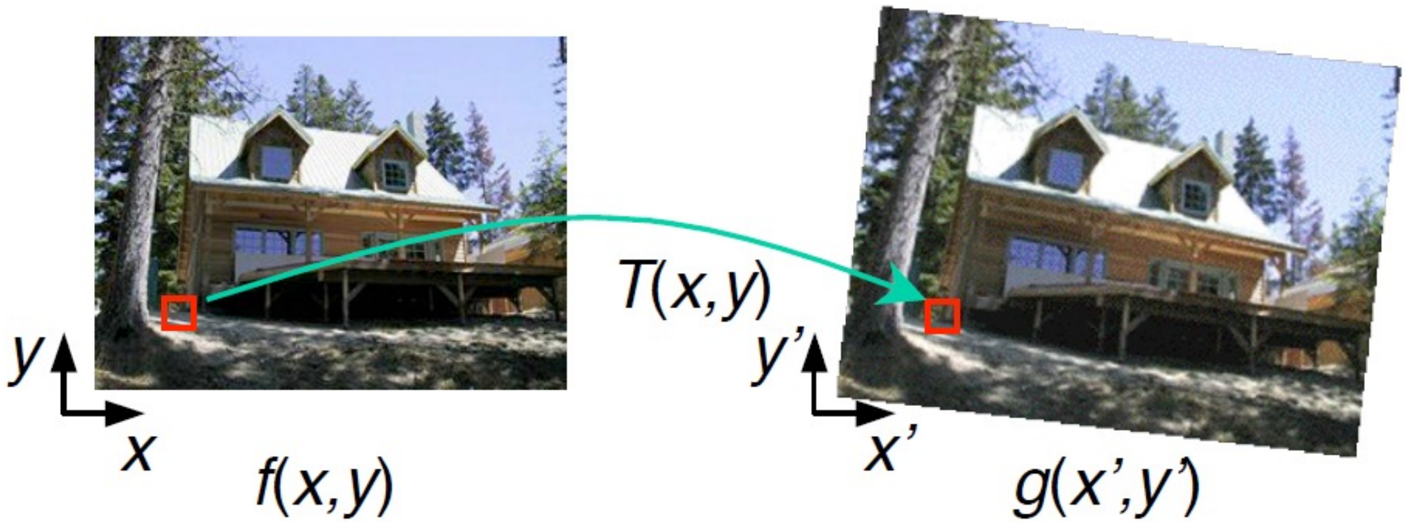
- Transformation T is a coordinate-changing machine:

  p’ = T(p)

- What does it mean that T is global?
  - It is the same for any point p
  - It can be described by just a few numbers (parameters)
- T is represented as a matrix (see prev. slides):

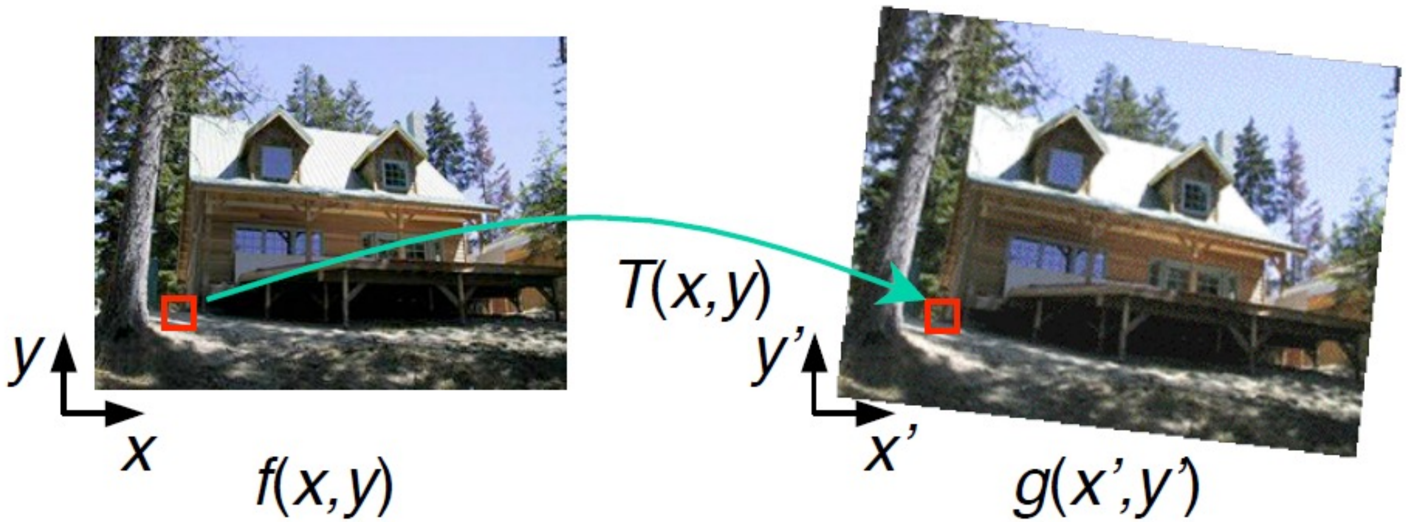
  p’ = M*p

Image Warping



Given a coordinate transform (x',y') = h(x,y) and a source image f(x,y), how do we compute a transformed image g(x',y') = f(T(x,y))?

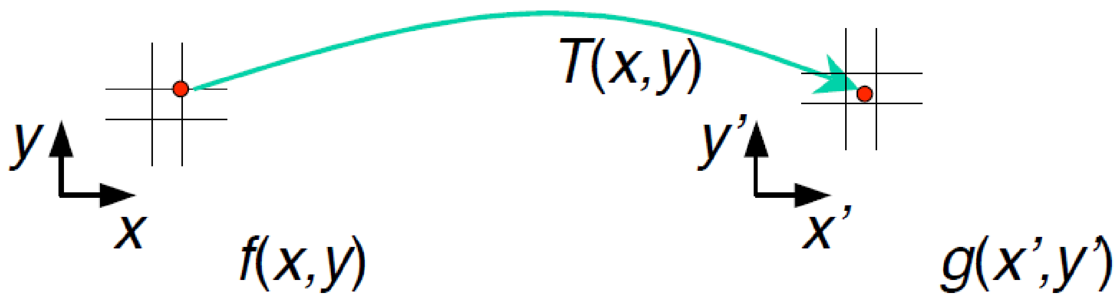
Forward Warping



Send each pixel f(x,y) to its corresponding location (x',y') = T(x,y) in the second image

Q: what if the pixel lands "between" two pixels?

Forward Warping

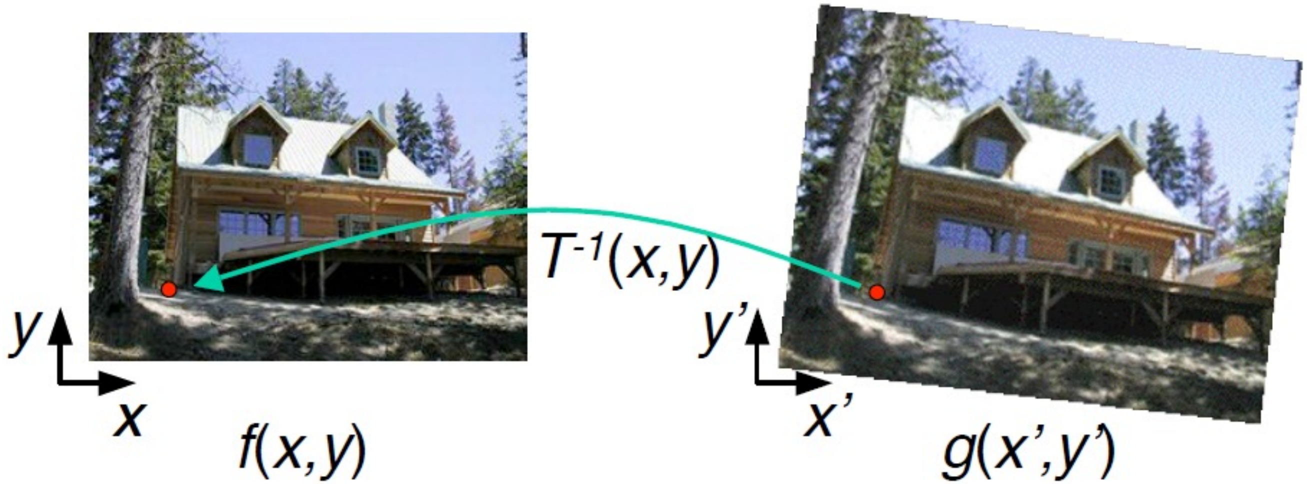


Send each pixel f(x,y) to its corresponding location (x',y') = T(x,y) in the second image

Q: what if the pixel lands "between" two pixels?

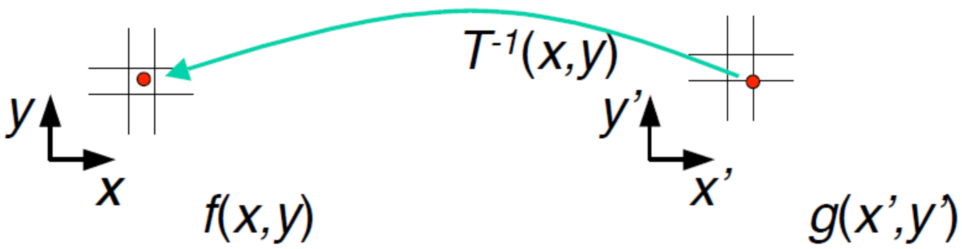
A: Distribute color among neighboring pixels (splatting)

Inverse Warping



- Get each pixel $g(x',y')$ from its corresponding location $(x,y) = T^{-1}(x',y')$ in the first image
- Q: what if pixel comes from “between” two pixels?

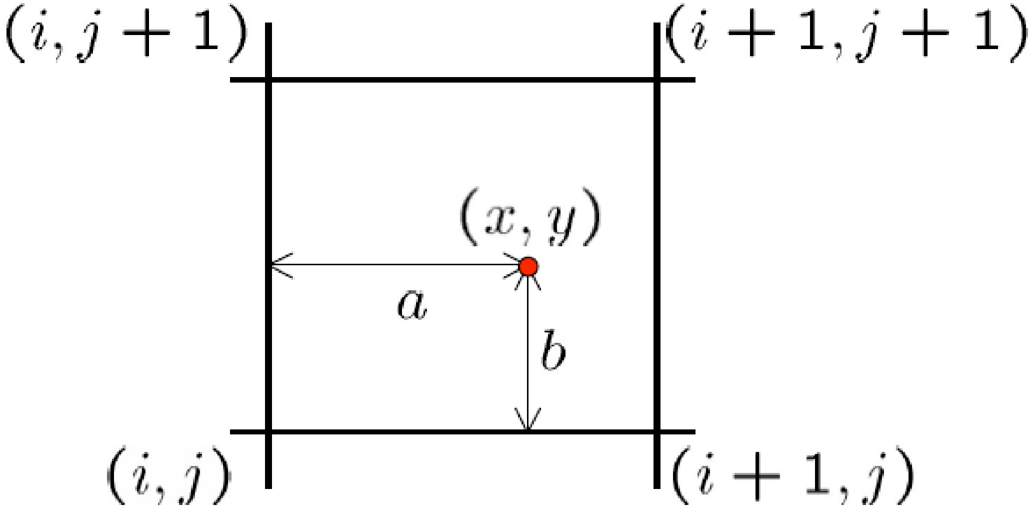
Inverse Warping



- Get each pixel g(x',y') from its corresponding location (x,y) = T^{-1}(x',y') in the first image

- Q: what if pixel comes from "between" two pixels?
- A: interpolate color value from neighbors
  - Bilinear interpolation typically used

Bilinear Interpolation



$$
\begin{aligned}
f(x,y) = \quad & (1-a)(1-b) && f[i,j] \\
& +a(1-b) && f[i+1,j] \\
& +ab && f[i+1,j+1] \\
& +(1-a)b && f[i,j+1]
\end{aligned}
$$

Forward vs. Inverse Warping

- Which is better?
- ...

Parameter Estimation

Slides by R. Hartley, A. Zisserman
and M. Pollefeys

Homography: Number of Measurements Required

- At least as many independent equations as degrees of freedom required
- Example:

$$\mathrm{x}' = \mathrm{Hx}$$

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

2 independent equations / point
8 degrees of freedom

$4 \mathrm{x} 2 \geq 8$

Approximate solutions

- Minimal solution

  4 points yield an exact solution for H

- More points
  - No exact solution, because measurements are inexact (“noise”)
  - Search for “best” according to some cost function
  - Algebraic or geometric/statistical cost

Direct Linear Transformation (DLT)

$$\mathrm{x}_i' = \mathrm{H}\mathrm{x}_i \qquad \mathrm{x}_i' \times \mathrm{H}\mathrm{x}_i = 0$$

$$\mathrm{x}_i' = (x_i', y_i', w_i')^\mathsf{T} \qquad \mathrm{H}\mathrm{x}_i = \begin{pmatrix} \mathrm{h}^{1\mathsf{T}}\mathrm{x}_i \\ \mathrm{h}^{2\mathsf{T}}\mathrm{x}_i \\ \mathrm{h}^{3\mathsf{T}}\mathrm{x}_i \end{pmatrix}$$

$$\mathrm{x}_i' \times \mathrm{H}\mathrm{x}_i = \begin{pmatrix} y_i'\mathrm{h}^{3\mathsf{T}}\mathrm{x}_i - w_i'\mathrm{h}^{2\mathsf{T}}\mathrm{x}_i \\ w_i'\mathrm{h}^{1\mathsf{T}}\mathrm{x}_i - x_i'\mathrm{h}^{3\mathsf{T}}\mathrm{x}_i \\ x_i'\mathrm{h}^{2\mathsf{T}}\mathrm{x}_i - y_i'\mathrm{h}^{1\mathsf{T}}\mathrm{x}_i \end{pmatrix}$$

$$\begin{bmatrix} 0^\mathsf{T} & -w_i'\mathrm{x}_i^\mathsf{T} & y_i'\mathrm{x}_i^\mathsf{T} \\ w_i'\mathrm{x}_i^\mathsf{T} & 0^\mathsf{T} & -x_i'\mathrm{x}_i^\mathsf{T} \\ -y_i'\mathrm{x}_i^\mathsf{T} & x_i'\mathrm{x}_i^\mathsf{T} & 0^\mathsf{T} \end{bmatrix} \begin{pmatrix} \mathrm{h}^1 \\ \mathrm{h}^2 \\ \mathrm{h}^3 \end{pmatrix} = 0$$

$$\mathrm{A}_i\mathrm{h} = 0$$

Direct Linear Transformation (DLT)

Equations are linear in h

$$\mathrm{A}_i \mathrm{h} = 0$$

Only 2 of 3 are linearly independent

(indeed, 2 eq/pt)

$$\begin{bmatrix} 0^\mathsf{T} & -w_i'\mathrm{x}_i^\mathsf{T} & y_i'\mathrm{x}_i^\mathsf{T} \\ w_i'\mathrm{x}_i^\mathsf{T} & 0^\mathsf{T} & -x_i'\mathrm{x}_i^\mathsf{T} \\ -y_i'\mathrm{x}_i^\mathsf{T} & x_i'\mathrm{x}_i^\mathsf{T} & 0^\mathsf{T} \end{bmatrix} \begin{pmatrix} \mathrm{h}^1 \\ \mathrm{h}^2 \\ \mathrm{h}^3 \end{pmatrix} = 0$$

$$x_i'\mathrm{A}_i^1 + y_i'\mathrm{A}_i^2 + w_i'\mathrm{A}_i^3 = 0$$

Direct Linear Transformation (DLT)

$$\begin{bmatrix} 0^\mathsf{T} & -w_i'\mathrm{x}_i^\mathsf{T} & y_i'\mathrm{x}_i^\mathsf{T} \\ w_i'\mathrm{x}_i^\mathsf{T} & 0^\mathsf{T} & -x_i'\mathrm{x}_i^\mathsf{T} \end{bmatrix} \begin{pmatrix} \mathrm{h}^1 \\ \mathrm{h}^2 \\ \mathrm{h}^3 \end{pmatrix} = 0$$

(only drop third row if $w_i' \neq 0$)

- Holds for any homogeneous representation, e.g. $(x_i', y_i', 1)$

Direct Linear Transformation (DLT)

- Solving for H $\quad \mathrm{A}\mathrm{h} = 0$

$$\begin{bmatrix} \mathrm{A}_1 \\ \mathrm{A}_2 \\ \mathrm{A}_3 \\ \mathrm{A}_4 \end{bmatrix} \mathrm{h} = 0$$

Size of A is 8x9, but rank 8

Trivial solution is $\mathrm{h}=0_9^T$ is not interesting

1-D null-space yields solution of interest, pick for example the one with $\|\mathrm{h}\| = 1$

Direct Linear Transformation (DLT)

- Over-determined solution

$$\begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{bmatrix} h = 0$$

No exact solution because of inexact measurement i.e. “noise”

Find approximate solution

- Additional constraint needed to avoid 0, e.g. $\|h\| = 1$
- $Ah = 0$ not possible, so minimize $\|Ah\|$

DLT Algorithm

Objective

Given $n \geq 4$ 2D to 2D point correspondences $\{x_i \leftrightarrow x_i'\}$, determine the 2D homography matrix H such that $x_i' = Hx_i$

Algorithm

(i) For each correspondence $x_i \leftrightarrow x_i'$ compute A_i. Usually only two first rows needed.

(ii) Assemble n 2x9 matrices A_i into a single $2n$x9 matrix A

(iii) Obtain SVD of A. Solution for h is last column of V

(iv) Determine H from h

Inhomogeneous solution

Since h can only be computed up to scale, pick h_j=1, e.g. h_9=1, and solve for 8-vector $\tilde{\mathrm{h}}$

$$\begin{bmatrix} 0 & 0 & 0 & -x_i w_i' & -y_i w_i' & -w_i w_i' & x_i y_i' & y_i y_i' \\ x_i w_i' & y_i w_i' & w_i w_i' & 0 & 0 & 0 & -x_i x_i' & -y_i x_i' \end{bmatrix} \tilde{\mathrm{h}} = \begin{pmatrix} -w_i y_i' \\ w_i x_i' \end{pmatrix}$$

Solve using Gaussian elimination (4 points) or using linear least-squares (more than 4 points)

However, if h_9=0 this approach fails

Also poor results if h_9 close to zero

Therefore, not recommended

Normalizing Transformations

- Since DLT is not invariant to transformations, what is a good choice of coordinates?

  e.g.

  – Translate centroid to origin

  – Scale to a $\sqrt{2}$ average distance to the origin

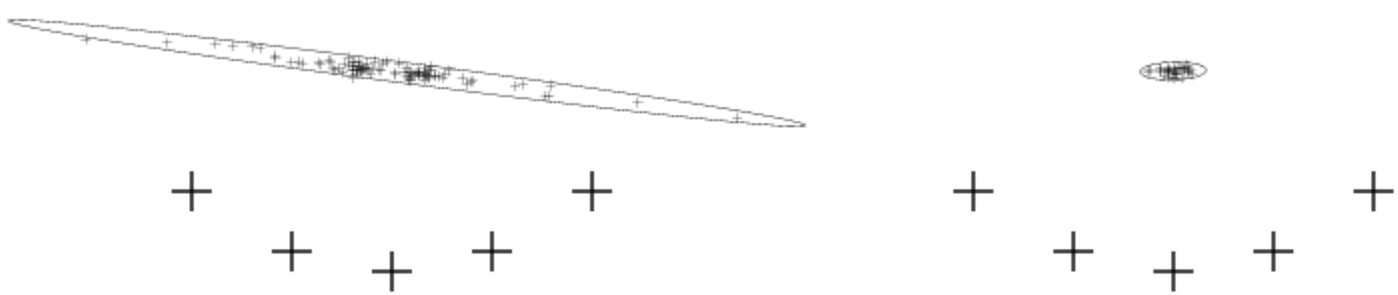
  – Independently on both images

$$\mathrm{T}_{\mathrm{norm}} = \begin{bmatrix} w+h & 0 & w/2 \\ 0 & w+h & h/2 \\ 0 & 0 & 1 \end{bmatrix}^{-1}$$

Importance of Normalization

$$\begin{array}{c}\begin{bmatrix} 0 & 0 & 0 & -x_i' & -y_i' & -1 & y_i'x_i & y_i'y_i & y_i' \\ x_i & y_i & 1 & 0 & 0 & 0 & -x_i'x_i & -x_i'y_i & -x_i' \end{bmatrix} \\ \begin{array}{ccccccccc} \sim 10^2 & \sim 10^2 & 1 & \sim 10^2 & \sim 10^2 & 1 & \sim 10^4 & \sim 10^4 & \sim 10^2 \end{array}\end{array} \begin{pmatrix} \mathrm{h}^1 \\ \mathrm{h}^2 \\ \mathrm{h}^3 \end{pmatrix} = 0$$

orders of magnitude difference!



Monte Carlo simulation
for identity computation based on 5 points
(not normalized ↔ normalized)

Normalized DLT Algorithm

<u>Objective</u>

Given $n \geq 4$ 2D to 2D point correspondences $\{x_i \leftrightarrow x_i'\}$, determine the 2D homography matrix H such that $x_i' = Hx_i$

<u>Algorithm</u>

(i) Normalize points $\tilde{x}_i = T_{norm} x_i, \tilde{x}_i' = T'_{norm} x_i'$

(ii) Apply DLT algorithm to $\tilde{x}_i \leftrightarrow \tilde{x}_i',$

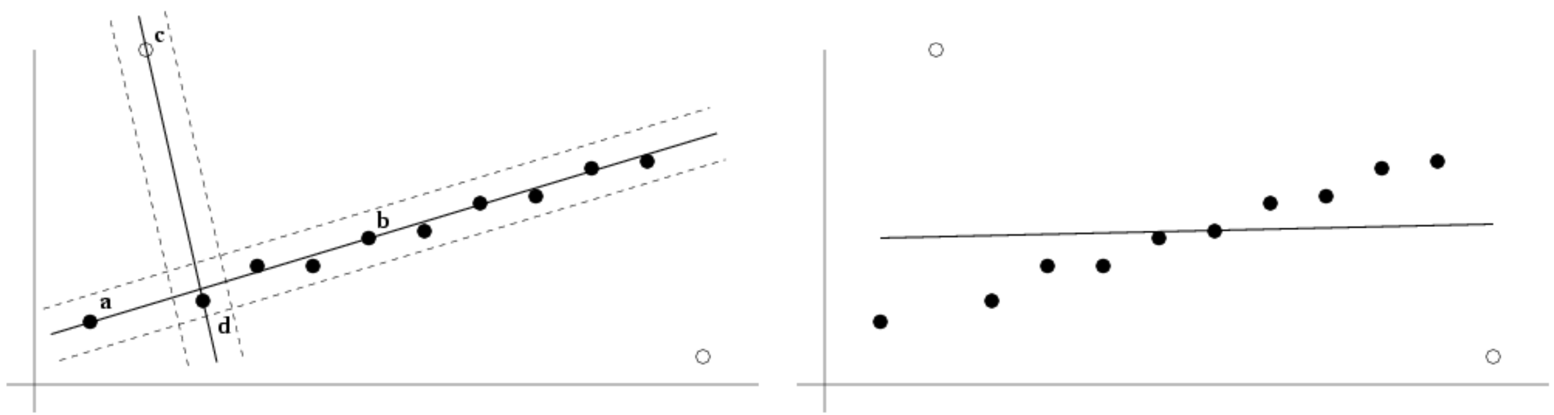
(iii) Denormalize solution $H = T'^{-1}_{norm} \tilde{H} T_{norm}$

RANSAC

Slides by R. Hartley, A. Zisserman and M. Pollefeys

Robust Estimation

- What if set of matches contains gross outliers?



RANSAC

Objective

Robust fit of model to data set S which contains outliers

Algorithm

(i) Randomly select a sample of *s* data points from S and instantiate the model from this subset.

(ii) Determine the set of data points S_i which are within a distance threshold *t* of the model. The set S_i is the consensus set of samples and defines the inliers of S.

(iii) If the subset of S_i is greater than some threshold *T*, re-estimate the model using all the points in S_i and terminate

(iv) If the size of S_i is less than *T*, select a new subset and repeat the above.

(v) After *N* trials the largest consensus set S_i is selected, and the model is re-estimated using all the points in the subset S_i

How Many Samples?

Choose N so that, with probability p, at least one random sample is free from outliers. e.g. p=0.99

$$\left(1-\left(1-e\right)^{s}\right)^{N}=1-p$$

$$N=\log\left(1-p\right)/\log\left(1-\left(1-e\right)^{s}\right)$$

	proportion of outliers e						
s	5%	10%	20%	25%	30%	40%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

Acceptable Consensus Set

- Typically, terminate when inlier ratio reaches expected ratio of inliers

$$T = (1 - e)n$$

Adaptively Determining the Number of Samples

e is often unknown a priori, so pick worst case, e.g. 50%, and adapt if more inliers are found, e.g. 80% would yield e=0.2

- N=∞, *sample_count* =0
- While N >*sample_count* repeat
  - Choose a sample and count the number of inliers
  - Set e=1-(number of inliers)/(total number of points)
  - Recompute N from e
  - Increment the *sample_count* by 1
- Terminate

$$\left(N = \log\left(1-p\right) / \log\left(1-\left(1-e\right)^s\right)\right)$$

Other robust algorithms

- RANSAC maximizes number of inliers
- LMedS minimizes median error

- Not recommended: case deletion, iterative least-squares, etc.

Automatic Computation of H

Objective

Compute homography between two images

Algorithm

(i) **Interest points:** Compute interest points in each image

(ii) **Putative correspondences:** Compute a set of interest point matches based on some similarity measure

(iii) **RANSAC robust estimation:** Repeat for *N* samples

(a) Select 4 correspondences and compute H

(b) Calculate the distance $d_{\perp}$ for each putative match

(c) Compute the number of inliers consistent with H ($d_{\perp}<t$)

Choose H with most inliers

(iv) **Optimal estimation:** re-estimate H from all inliers by minimizing ML cost function with Levenberg-Marquardt

(v) Guided matching: Determine more matches using prediction by computed H

Optionally iterate last two steps until convergence

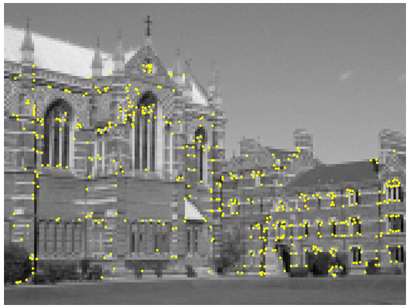
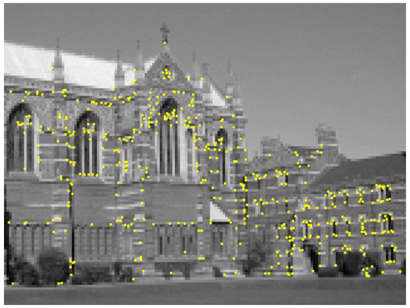
Determine Putative Correspondences

- Compare interest points
  Similarity measure:
  - SAD, SSD, ZNCC in small neighborhood

- If motion is limited, only consider interest points with similar coordinates

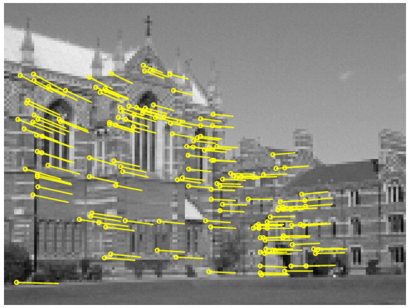
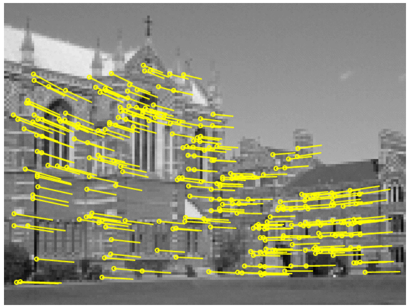
Example: robust computation

#in	1-e	adapt. *N*
6	2%	20M
10	3%	2.5M
44	16%	6,922
58	21%	2,291
73	26%	911
151	56%	43

Interest points
(500/image)
(640x480)


Putative correspondences (268)
(Best match,SSD<20,±320)
Outliers (117)
(t=1.25 pixel; 43 iterations)

Inliers (151)

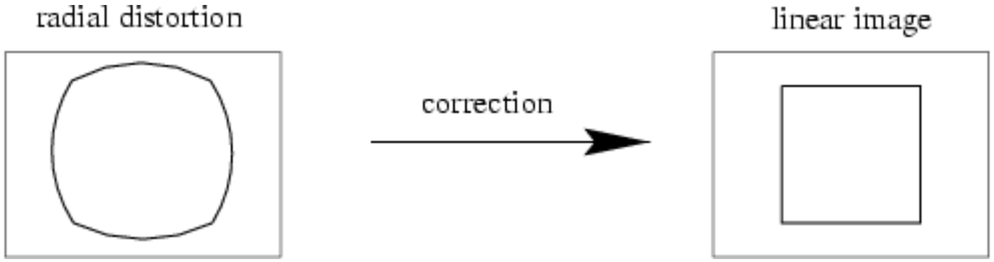
Final inliers (262)

Radial Distortion and Undistortion

Slides by R. Hartley, A. Zisserman and M. Pollefeys

Radial Distortion

short and long focal length



Typical Undistortion Model

Correction of distortion

$$\hat{x} = x_c + L(r)(x - x_c) \qquad \hat{y} = y_c + L(r)(y - y_c)$$

Choice of the distortion function and center

$$x = x_o + (x_o - c_x)(K_1 r^2 + K_2 r^4 + \ldots)$$
$$y = y_o + (y_o - c_y)(K_1 r^2 + K_2 r^4 + \ldots)$$

$$r = (x_o - c_x)^2 + (y_o - c_y)^2 \ .$$

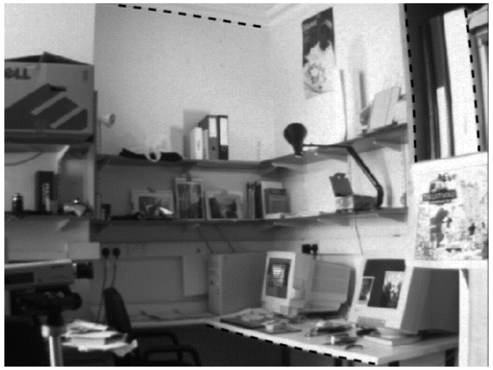
Computing the parameters of the distortion function

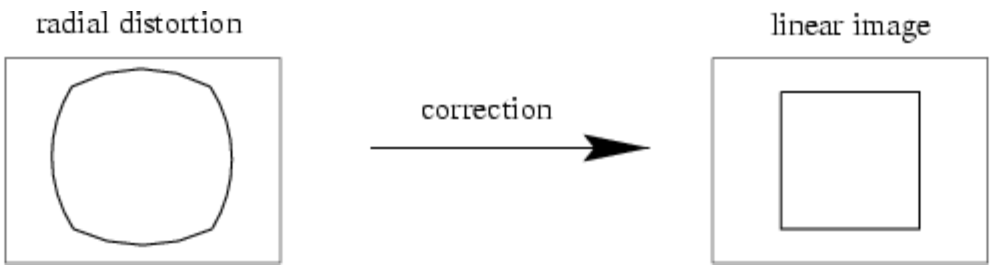
(i) Minimize with additional unknowns

(ii) Straighten lines

(iii) …

Why Undistort?





$$(\tilde{x}, \tilde{y}, 1)^\top = [\mathtt{I} \mid \mathbf{0}]\mathbf{X}_{\mathrm{cam}}$$

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

Two-View Geometry

Slides by R. Hartley, A. Zisserman and M. Pollefeys

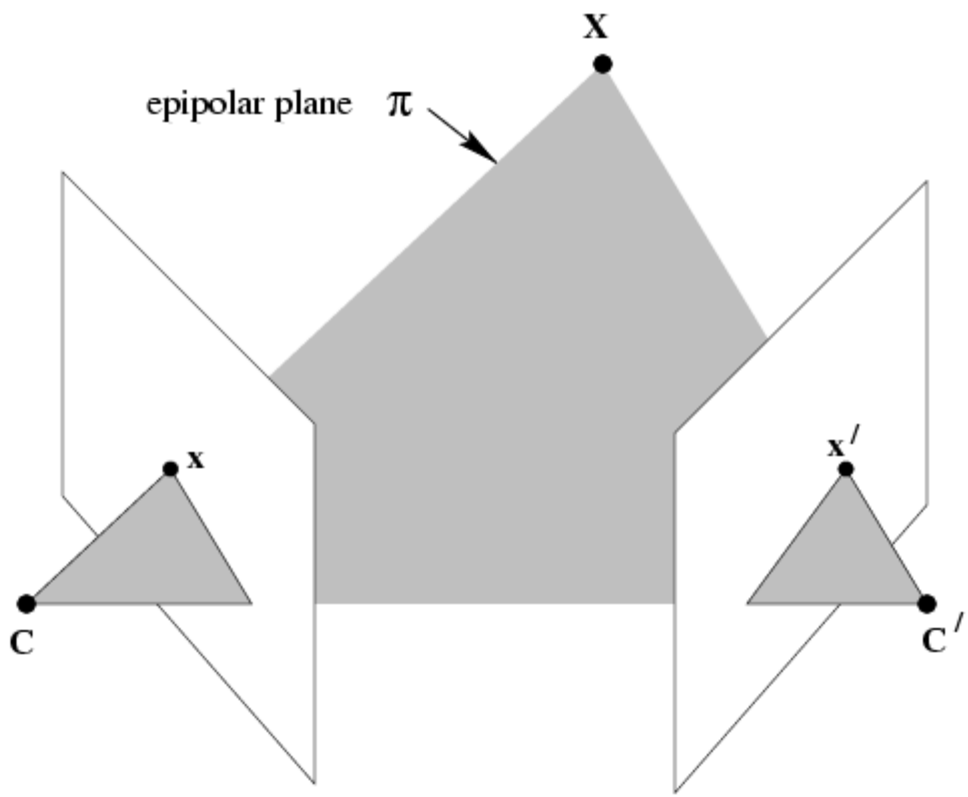
Three questions:

(i) **Correspondence geometry:** Given an image point x in the first image, how does this constrain the position of the corresponding point x' in the second image?

(ii) **Camera geometry (motion):** Given a set of corresponding image points $\{x_i \leftrightarrow x'_i\}$, $i=1,\ldots,n$, what are the cameras P and P' for the two views?

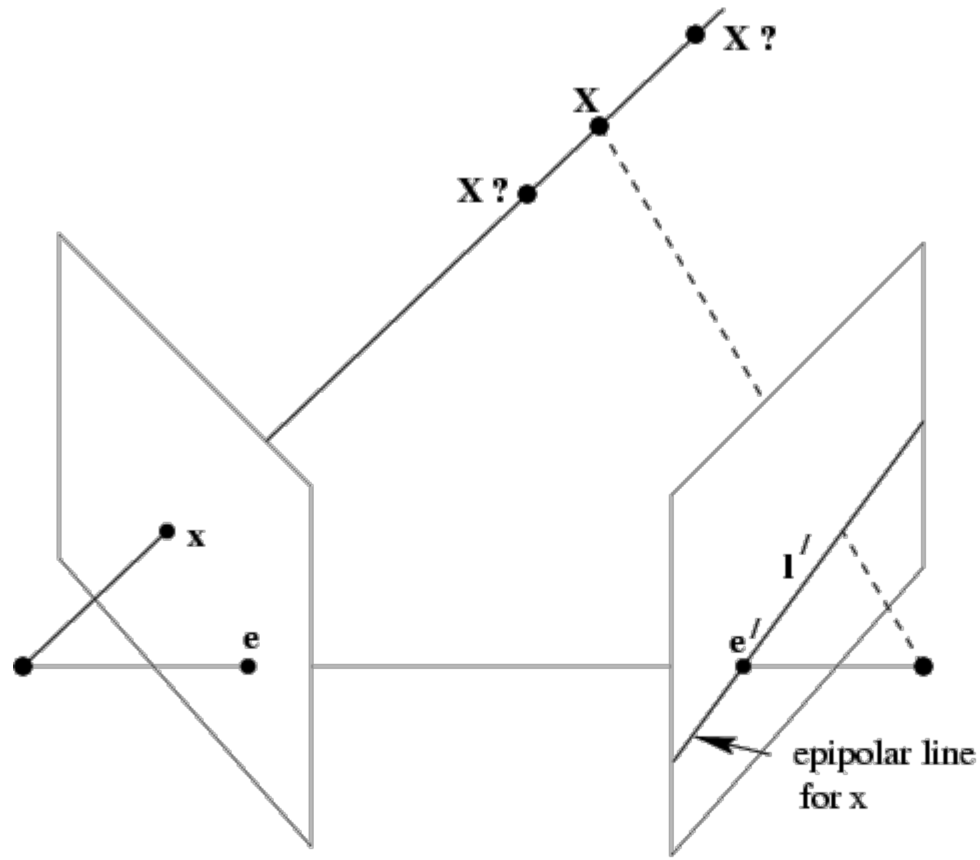
(iii) **Scene geometry (structure):** Given corresponding image points $x_i \leftrightarrow x'_i$ and cameras P, P', what is the position of (their pre-image) X in space?

The Epipolar Geometry



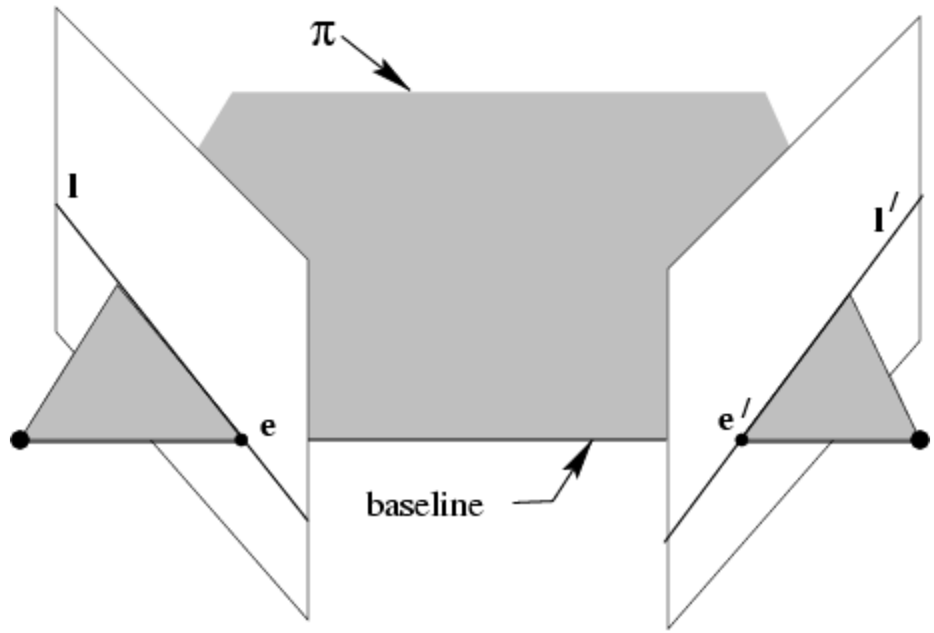
C, C', x, x' and X are coplanar

The Epipolar Geometry



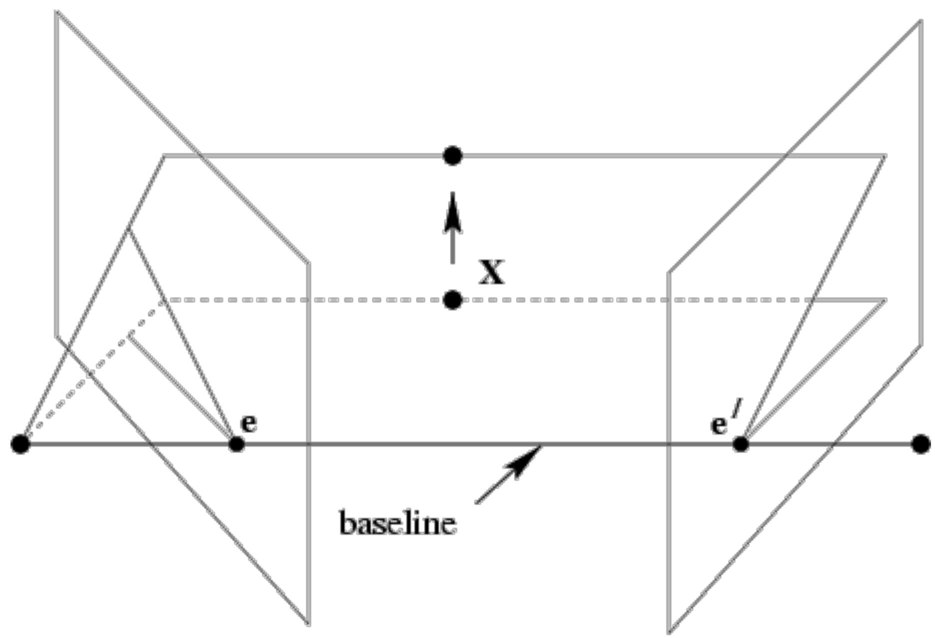
What if only C,C’,x are known?

The Epipolar Geometry



All points on π project on l and l’

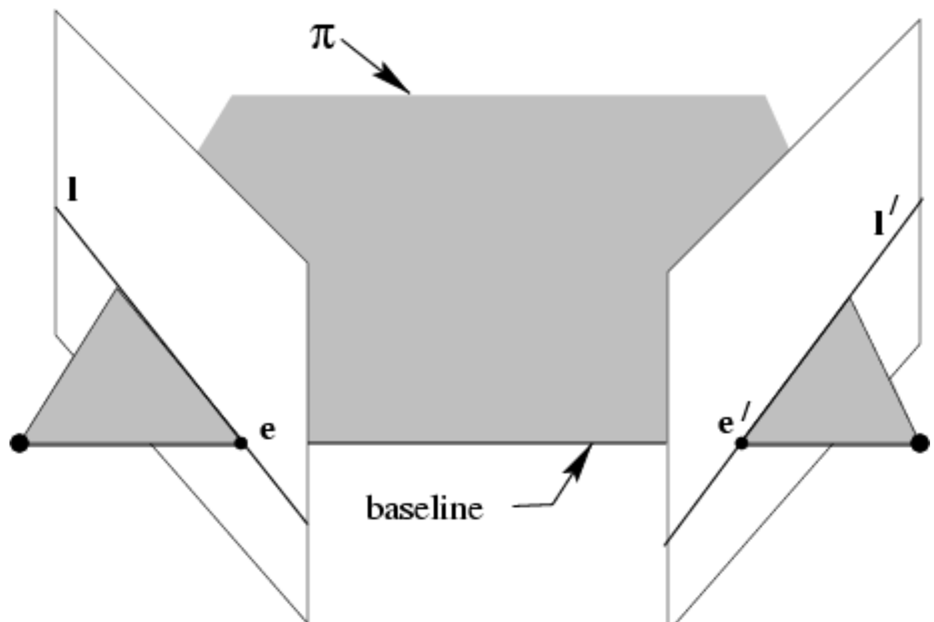
The Epipolar Geometry



Family of planes π and lines l and l'
Intersection in e and e'

The Epipolar Geometry

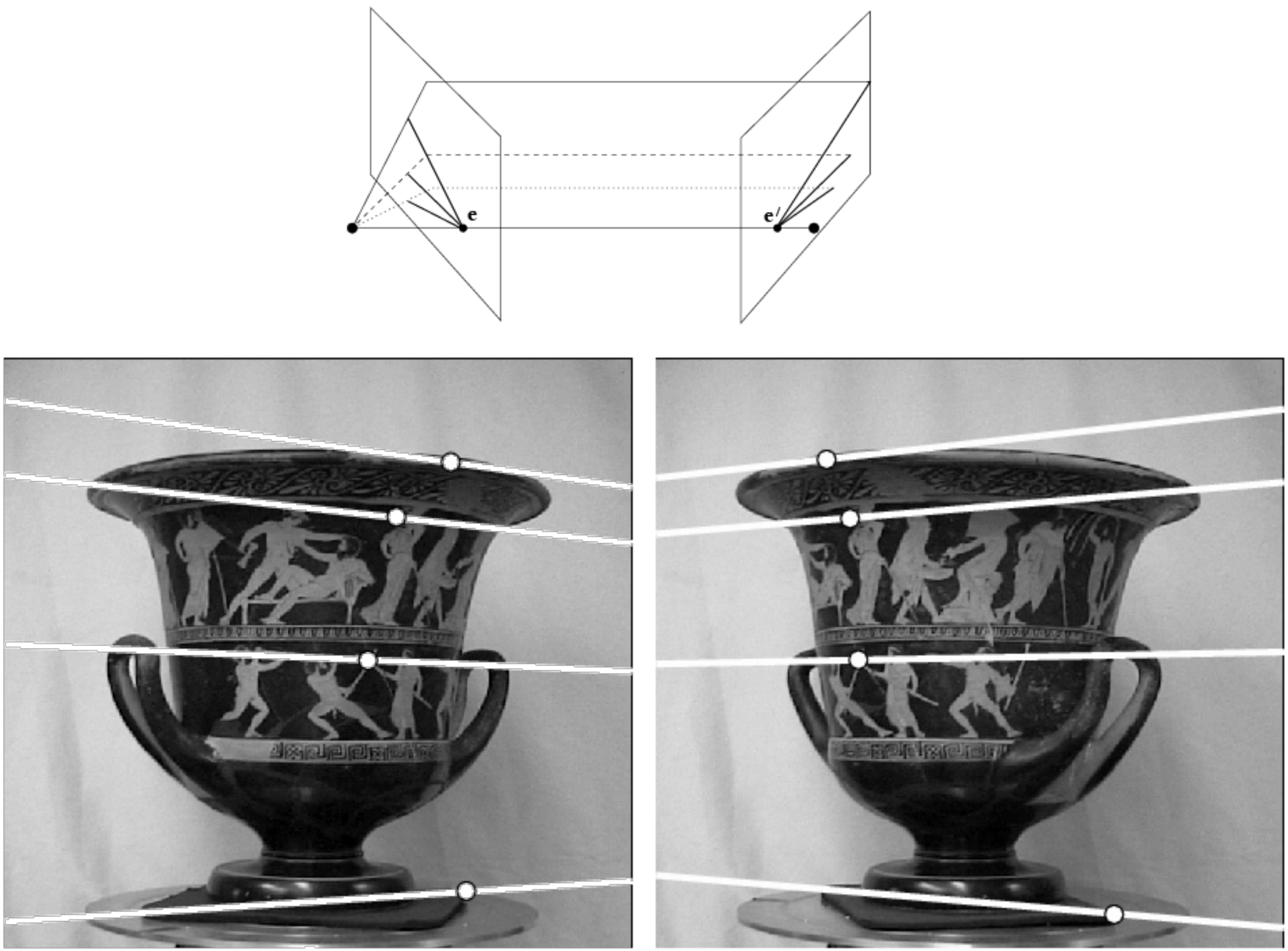
epipoles e, e'
= intersection of baseline with image plane
= projection of projection center in other image
= vanishing point of camera motion direction



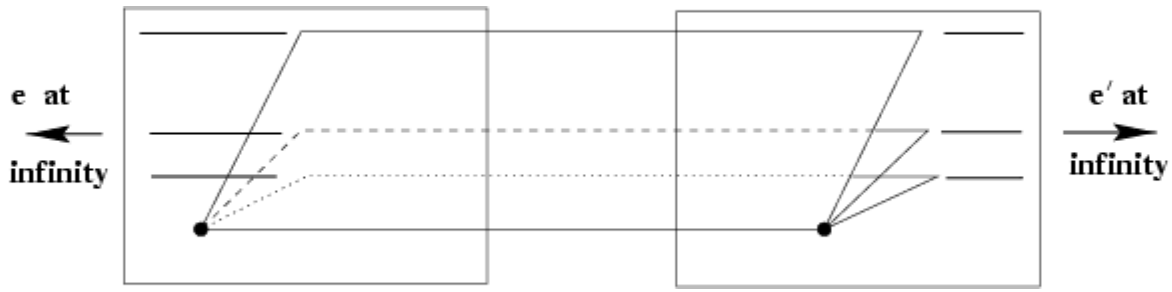
an epipolar plane = plane containing baseline (1-D family)

an epipolar line = intersection of epipolar plane with image
(always come in corresponding pairs)

Example: Converging Cameras

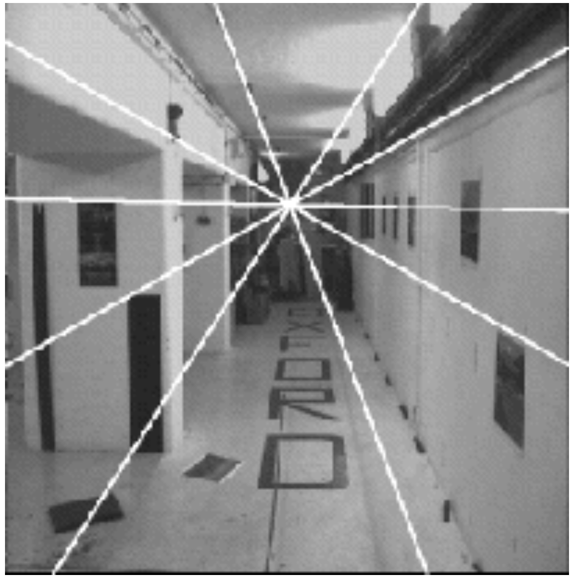


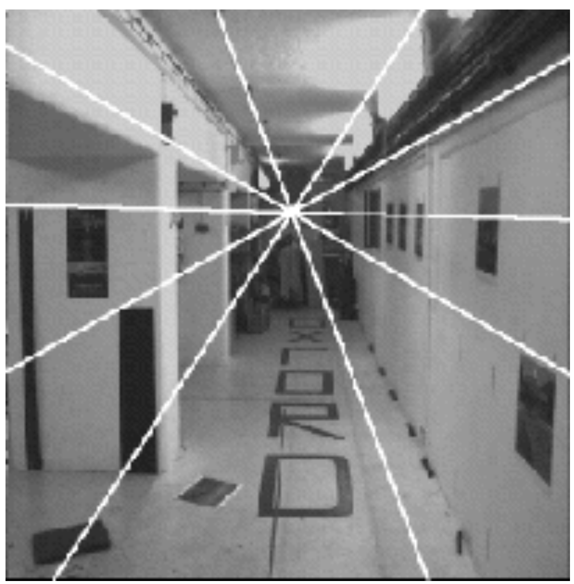
Example: Motion Parallel to Image Plane

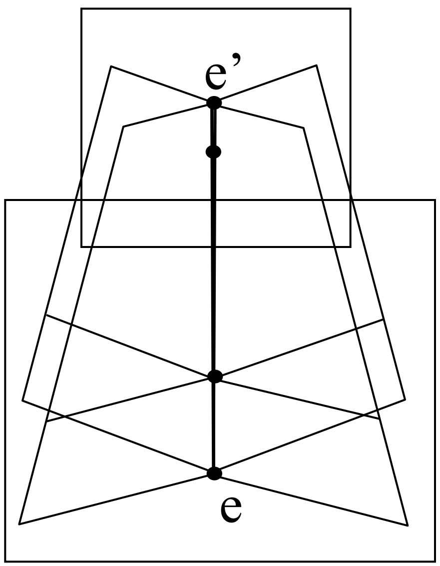


(simple for stereo → rectification)

Example: Forward Motion







The Fundamental Matrix F

algebraic representation of epipolar geometry

$$x \mapsto l'$$

we will see that mapping is a (singular) correlation
(i.e. projective mapping from points to lines)
represented by the fundamental matrix F

The Fundamental Matrix F

correspondence condition

The fundamental matrix satisfies the condition that for any pair of corresponding points $x \leftrightarrow x'$ in the two images

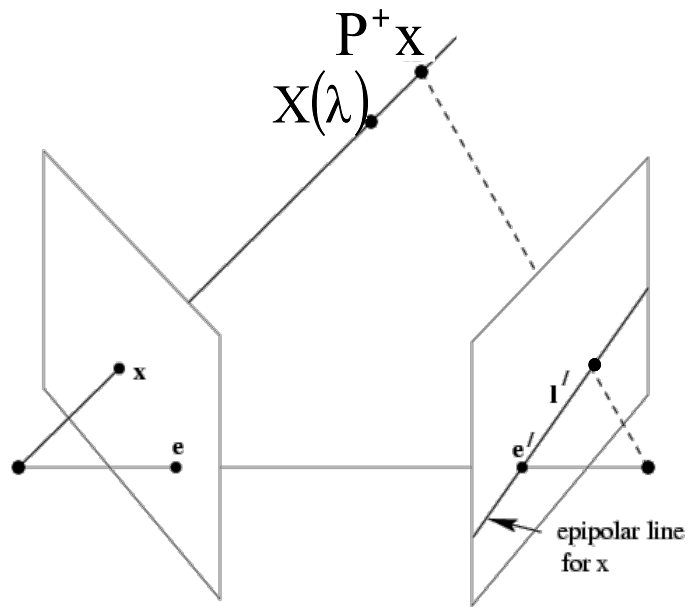
$$x'^{T} F x = 0 \qquad \left(x'^{T} l' = 0\right)$$

The Fundamental Matrix F

$$X(\lambda) = P^{+}x + \lambda C \qquad \left(PP^{+} = I\right)$$

$$l = P'C \times P'P^{+}x$$

$$F = [e']_{\times} P'P^{+}$$



(note: doesn’t work for C=C’ $\Rightarrow$ F=0)

The Fundamental Matrix F

F is the unique 3x3 rank 2 matrix that satisfies $x'^T F x = 0$ for all $x \leftrightarrow x'$

(i) Transpose: if F is fundamental matrix for (P,P'), then F^T is fundamental matrix for (P',P)

(ii) Epipolar lines: $l' = Fx$ & $l = F^T x'$

(iii) Epipoles: on all epipolar lines, thus $e'^T F x = 0, \forall x \Rightarrow e'^T F = 0$, similarly $Fe = 0$

(iv) F has 7 d.o.f. , i.e. 3x3-1(homogeneous)-1(rank2)

(v) F is a correlation, projective mapping from a point x to a line $l' = Fx$ (not a proper correlation, i.e. not invertible)

Two View Geometry Computation: Linear Algorithm

For every match (m,m´): $\mathrm{x}'^{\mathrm{T}}\mathrm{F}\mathrm{x}=0$

$$x'xf_{11}+x'yf_{12}+x'f_{13}+y'xf_{21}+y'yf_{22}+y'f_{23}+xf_{31}+yf_{32}+f_{33}=0$$

separate known from unknown

$$[x'x, x'y, x', y'x, y'y, y', x, y, 1][f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33}]^{\mathrm{T}}=0$$

(data) (unknowns)

(linear)

$$\begin{bmatrix} x'_1 x_1 & x'_1 y_1 & x'_1 & y'_1 x_1 & y'_1 y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_n x_n & x'_n y_n & x'_n & y'_n x_n & y'_n y_n & y'_n & x_n & y_n & 1 \end{bmatrix} \mathrm{f}=0$$

$$\mathrm{Af}=0$$

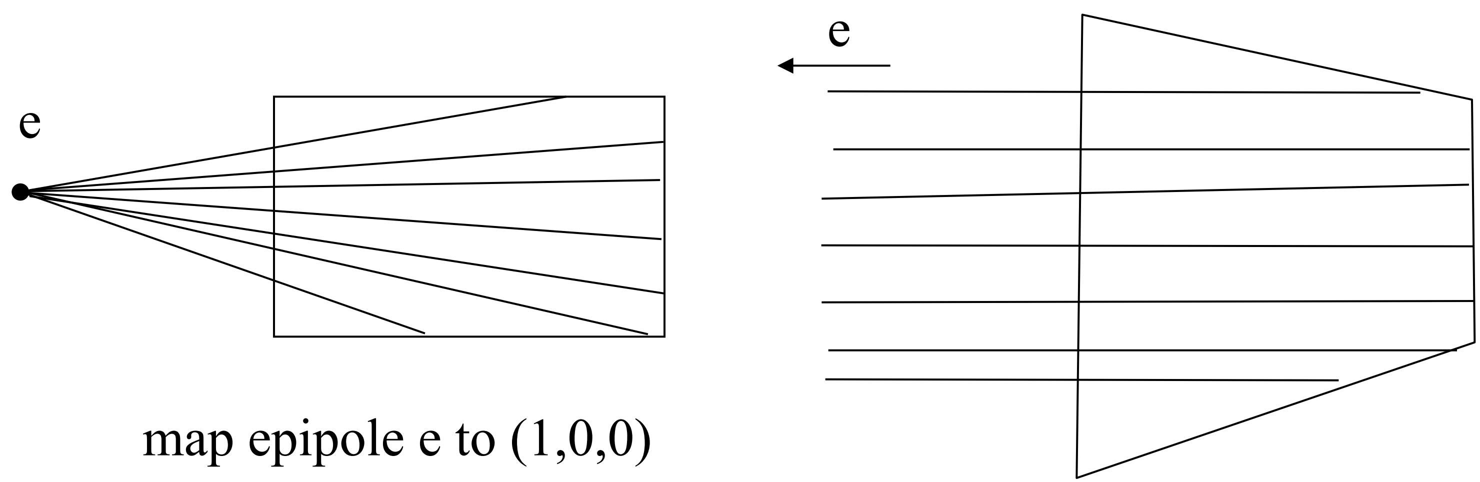
Benefits from having F

- Given a pixel in one image, the corresponding pixel has to lie on epipolar line
- Search space reduced from 2-D to 1-D

Image Pair Rectification

simplify stereo matching
by warping the images

Apply projective transformation so that epipolar lines
correspond to horizontal scanlines



$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = He$$

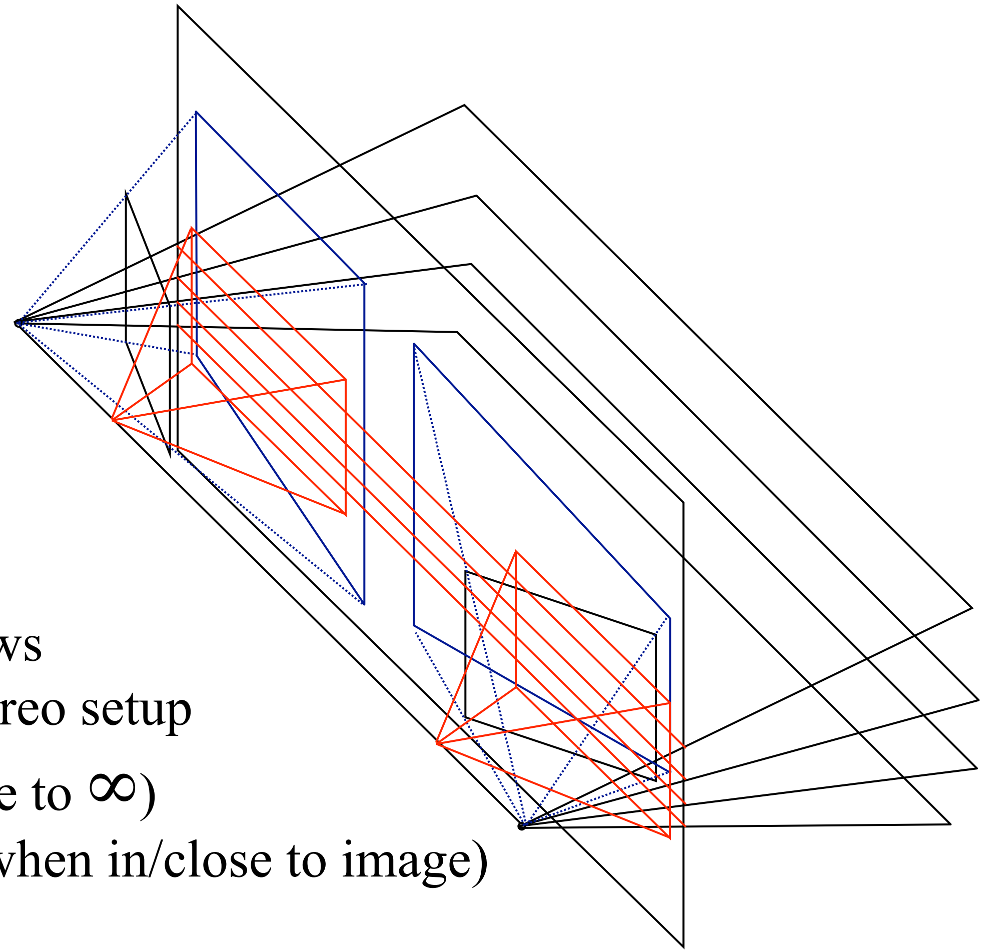
map epipole e to (1,0,0)

try to minimize image distortion

problem when epipole in (or close to) the image

Planar Rectification

(standard approach)



Bring two views
to standard stereo setup

(moves epipole to ∞)
(not possible when in/close to image)