Computer Vision – Transformations, Imaging Geometry and Stereo Vision

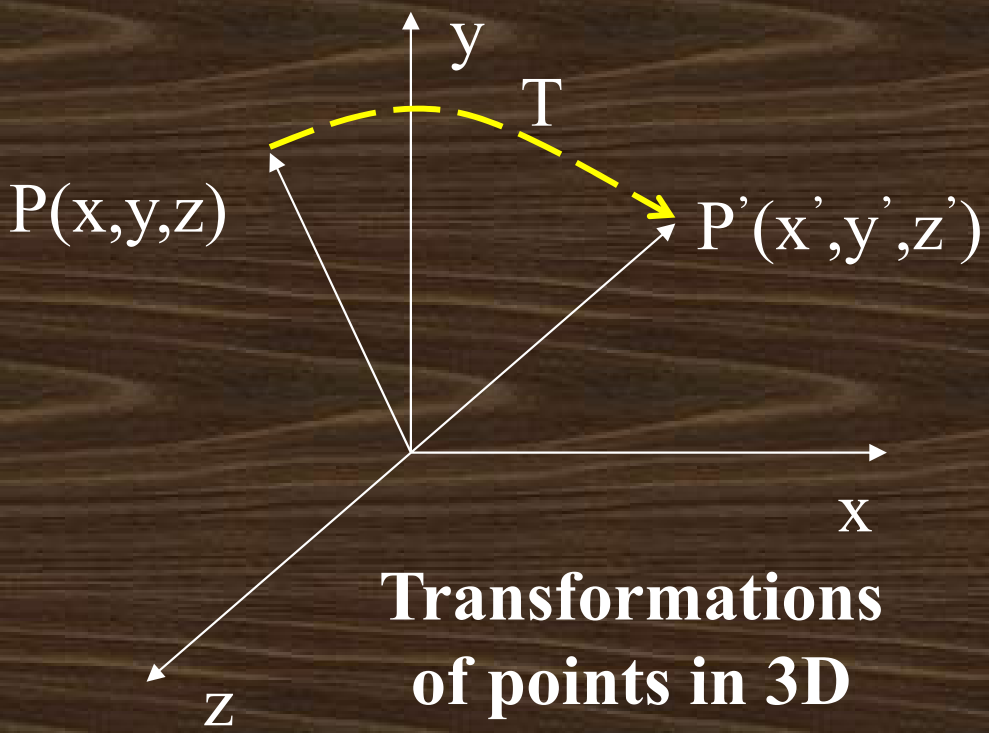
Dr. S. Das

IIT Madras, Chennai-36

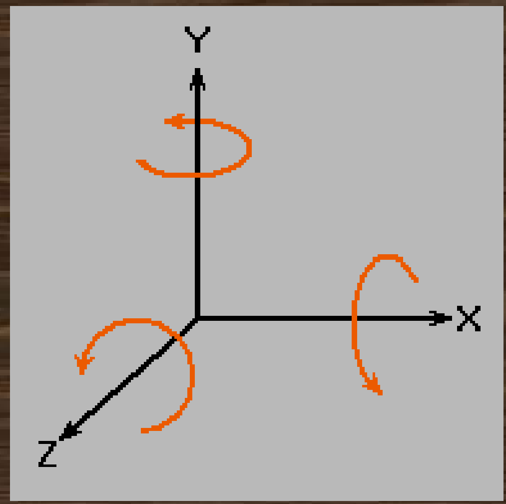
BASICS

Representation of Points in the 3D world: a vector of length 3

$$\overline{X} = [x \; y \; z]^T$$



Transformations of points in 3D



Right handed coordinate system

4 basic transformations

- **Translation**
- **Rotation**
- **Scaling**
- **Shear**

Affine transformations

Basics 3D Transformation equations

- Translation : P' = P + ΔP

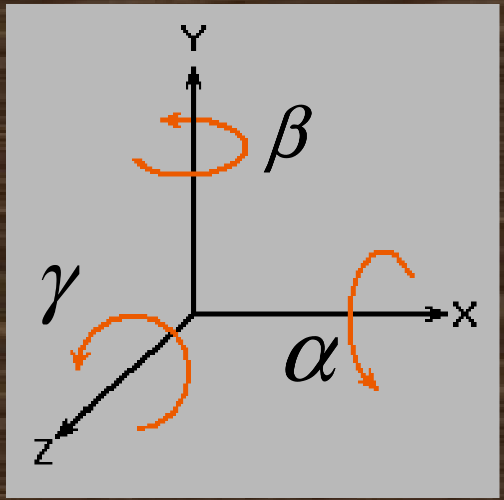
$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix}$$

- Scaling: P'= SP

$$S = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & S_z \end{bmatrix}$$

- Rotation : about an axis,

  P' = RP

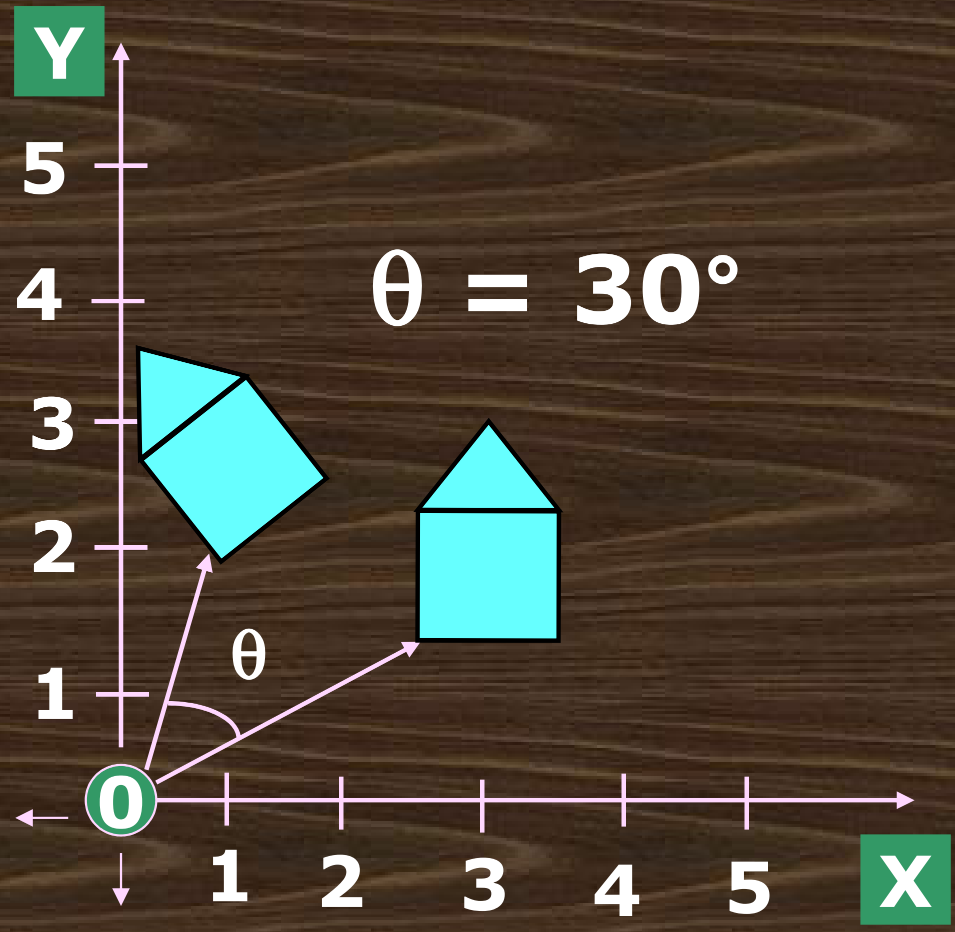


ROTATION - 2D

$$x' = x\cos(\theta) - y\sin(\theta)$$
$$y' = x\sin(\theta) + y\cos(\theta)$$

In matrix form, this is :

$$R = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$



Positive Rotations: counter clockwise about the origin

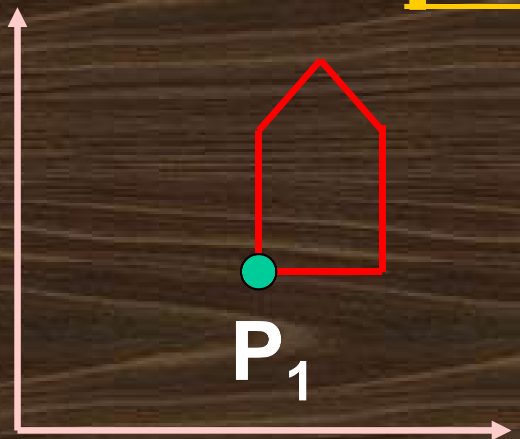
For rotations, |R| = 1 and $[R]^T = [R]^{-1}$. Rotation matrices are orthogonal.

Rotation about an arbitrary point P in space

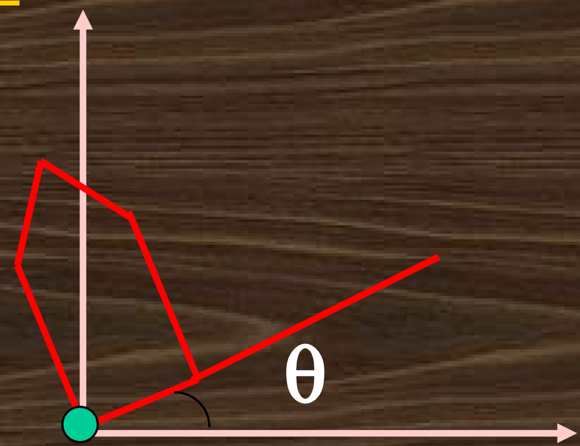
As we mentioned before, rotations are applied about the origin. So to rotate about any arbitrary point P in space, translate so that P coincides with the origin, then rotate, then translate back. Steps are:

- Translate by $(-P_x, -P_y)$
- Rotate
- Translate by (P_x, P_y)

Rotation about an arbitrary point P in space

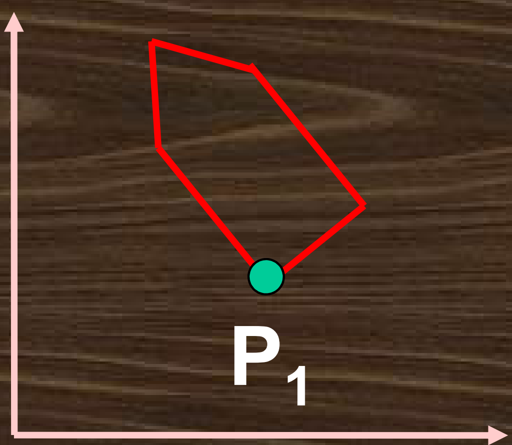


House at P_1



Rotation by θ

Translation of P_1 to Origin



Translation back to P_1

2D Transformation equations (revisited)

- Translation : P' = P + ΔP

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \Rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \end{bmatrix}^T \begin{bmatrix} x \\ y \end{bmatrix} ??$$

- Rotation : about an axis,

  P' = RP

$$\boldsymbol{R} = \begin{bmatrix} \mathbf{cos}(\boldsymbol{\theta}) & \mathbf{-sin}(\boldsymbol{\theta}) \\ \mathbf{sin}(\boldsymbol{\theta}) & \mathbf{cos}(\boldsymbol{\theta}) \end{bmatrix}$$

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

Rotation about an arbitrary point P in space

$R_{gen} = T_1(-P_x, -P_y) * R_2(\theta) * T_3(P_x, P_y)$

$$= \begin{bmatrix} 1 & 0 & -P_x \\ 0 & 1 & -P_y \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & P_x \\ 0 & 1 & P_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos(\theta) & -\sin(\theta) & P_x*(\cos(\theta)-1)-P_y*(\sin(\theta) \\ \sin(\theta) & \cos(\theta) & P_y*(\cos(\theta)-1)+P_x*\sin(\theta) \\ 0 & 0 & 1 \end{bmatrix}$$

Using Homogeneous system

Homogeneous representation of a point in 3D space:

$$P = |\, x\ y\ z\ w \,|^{T}$$

$$(w = 1, \text{for a 3D point})$$

Transformations will thus be represented by 4x4 matrices:

P′ = A.P

Homogenous Coordinate systems

- In order to Apply a sequence of transformations to produce composite transformations we introduce the fourth coordinate
- Homogeneous representation of 3D point:

$|x\ y\ z\ h|^T$ (h=1 for a 3D point, dummy coordinate)

- Transformations will be represented by 4x4 matrices.

$$T = \begin{bmatrix} 1 & 0 & 0 & \Delta x \\ 0 & 1 & 0 & \Delta y \\ 0 & 0 & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Homogenous Translation matrix

$$S = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Homogenous Scaling matrix

$$R_\alpha = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation about x axis by angle α

$$R_\beta = \begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation about y axis by angle β

Change of sign?

$$R_\gamma = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 & 0 \\ \sin\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation about z axis by angle γ

How can one do a Rotation about an arbitrary Axis in Space?

3D Transformation equations (3)

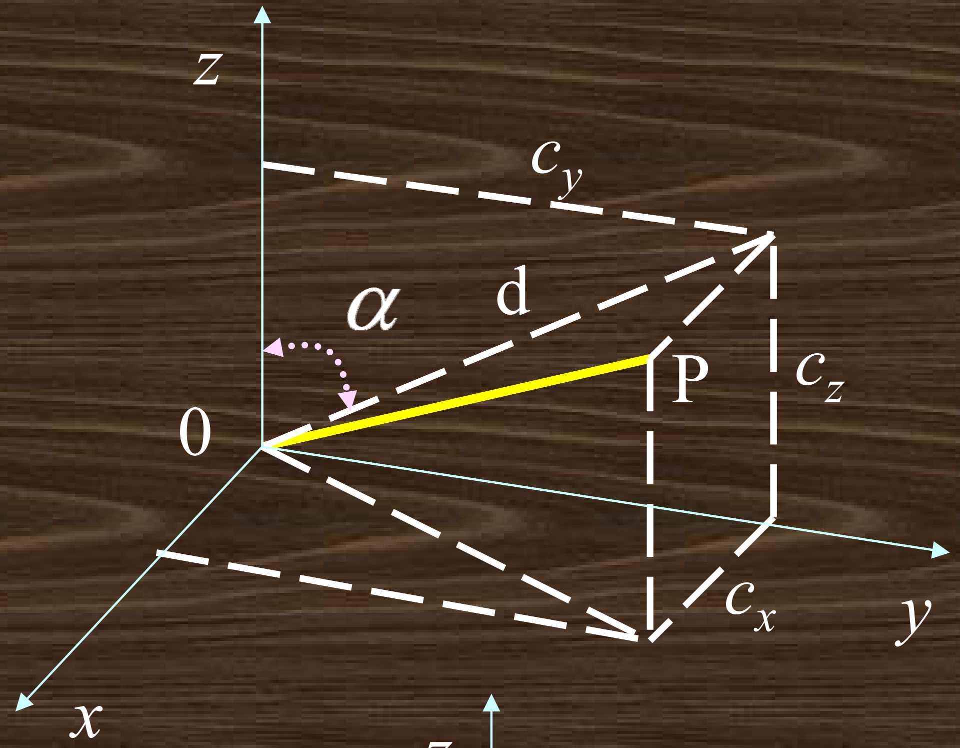
Rotation About an Arbitrary Axis in Space

Assume we want to perform a rotation about an axis in space, passing through the point (x_0, y_0, z_0) with direction cosines (c_x, c_y, c_z), by θ degrees.

1) **First of all, translate by: $-(x_0, y_0, z_0) = |T|$.**
2) **Next, we rotate the axis into one of the principle axes. Let's pick, Z ($|R_x|$, $|R_y|$).**
3) **We rotate next by θ degrees in Z ($|R_z(\theta)|$).**
4) **Then we undo the rotations to align the axis.**
5) **We undo the translation: translate by (x_0, y_0, z_0)**

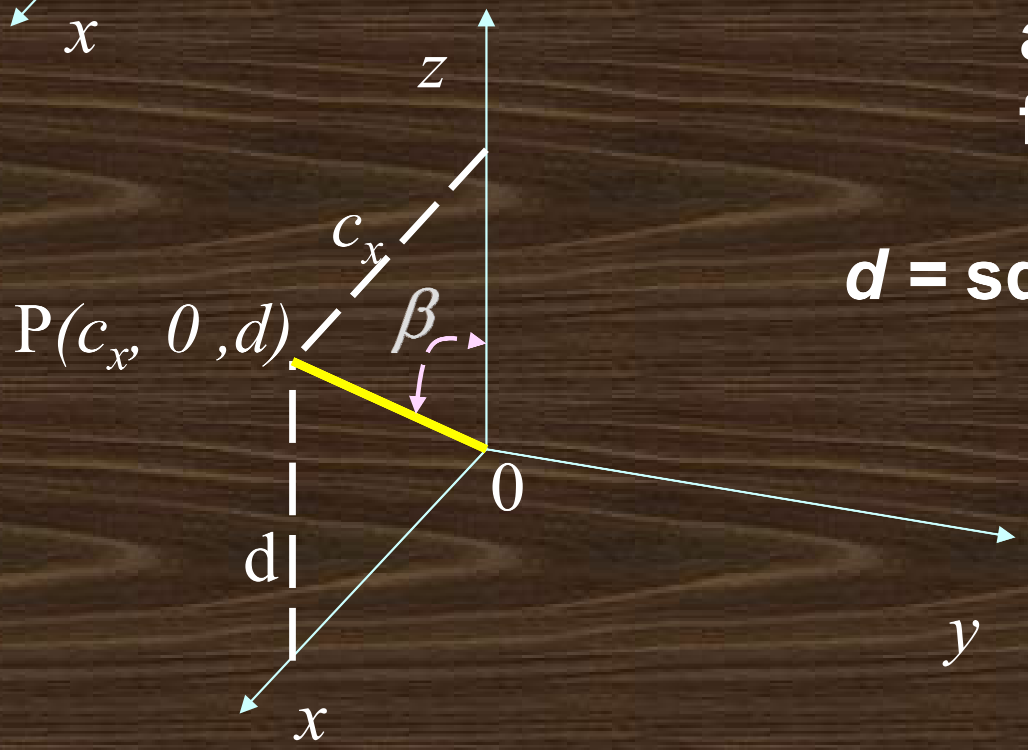
T*he tricky part is (2) above.*

This is going to take 2 rotations,
i) about x (to place the axis in the x-z plane)
and
ii) about y (to place the result coincident with the z axis).



Rotation about *x* by α:
How do we determine α?

Project the unit vector, along OP, into the y-z plane. The y and z components are c_y and c_z, the directions cosines of the unit vector along the arbitrary axis. It can be seen from the diagram above, that :



$$d = \mathrm{sqrt}(c_y^{\,2} + c_z^{\,2}), \qquad \cos(\alpha) = c_z / d$$

$$\sin(\alpha) = c_y / d$$

Rotation by β about y:
How do we determine β?
Similar to above:

Determine the angle β to rotate the result into the Z axis:

The *x* component is c_x and the *z* component is d.

$\cos(\beta) = d = d/(\text{length of the unit vector})$

$\sin(\beta) = c_x = c_x/(\text{length of the unit vector})$.

Final Transformation:

$M = |T|^{-1} \, |R_x|^{-1} \, |R_y|^{-1} \, |R_z| \, |R_y| \, |R_x| \, |T|$

If you are given 2 points instead, you can calculate the direction cosines as follows:

$V = | \, (x_1 - x_0) \;\; (y_1 - y_0) \;\; (z_1 - z_0) \, |^T$

$c_x = (x_1 - x_0) / |V|$

$c_y = (y_1 - y_0) / |V|$

$c_z = (z_1 - z_0) / |V|$,

where |V| is the length of the vector V.

Inverse transformations

$$T^{-1} = \begin{bmatrix} 1 & 0 & 0 & -\Delta x \\ 0 & 1 & 0 & -\Delta y \\ 0 & 0 & 1 & -\Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Inverse Translation

$$S^{-1} = \begin{bmatrix} 1/S_x & 0 & 0 & 0 \\ 0 & 1/S_y & 0 & 0 \\ 0 & 0 & 1/S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Inverse scaling

Inverse Rotation

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} \cos\beta & 0 & -\sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} \cos\gamma & \sin\gamma & 0 & 0 \\ -\sin\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

R_α^{-1} R_β^{-1} R_γ^{-1}

Concatenation of transformations

- The 4 X 4 representation is used to perform a sequence of transformations.
- Thus application of several transformations in a particular sequence can be presented by a single transformation matrix

$$v^* = R_\theta(S(Tv)) = Av; \ \mathrm{A} = \mathrm{R}_\theta.S.T$$

- The order of application is important… the multiplication may not be commutable.

Commutivity of Transformations

If we scale, then translate to the origin, and then translate back, is that equivalent to translate to origin, scale, translate back?

When is the order of matrix multiplication unimportant?

When does $T_1 * T_2 = T_2 * T_1$?

Cases where $T_1 * T_2 = T_2 * T_1$:

T_1	T_2
translation	translation
scale	scale
rotation	rotation
Scale (uniform)	rotation

COMPOSITE TRANSFORMATIONS

If we want to apply a series of transformations T_1, T_2, T_3 to a set of points, We can do it in two ways:

1) We can calculate $p'=T_1*p$, $p''= T_2*p'$, $p'''=T_3*p''$
2) Calculate $T= T_1*T_2*T_3$, then $p'''= T*p$.

Method 2, saves large number of additions and multiplications (computational time) – needs approximately 1/3 of as many operations. Therefore, we concatenate or compose the matrices into one final transformation matrix, and then apply that to the points.

Spaces

Object Space

definition of objects. Also called Modeling space.

World Space

where the scene and viewing specification is made

Eye space (Normalized Viewing Space)

where eye point (COP) is at the origin looking down the Z axis.

3D Image Space

A 3D Perspected space.

Dimensions: -1:1 in x & y, 0:1 in Z.

Where Image space hidden surface algorithms work.

Screen Space (2D)

Coordinates 0:width, 0:height

Projections

We will look at several planar geometric 3D to 2D projection:

- Parallel Projections
  - Orthographic
  - Oblique
- Perspective

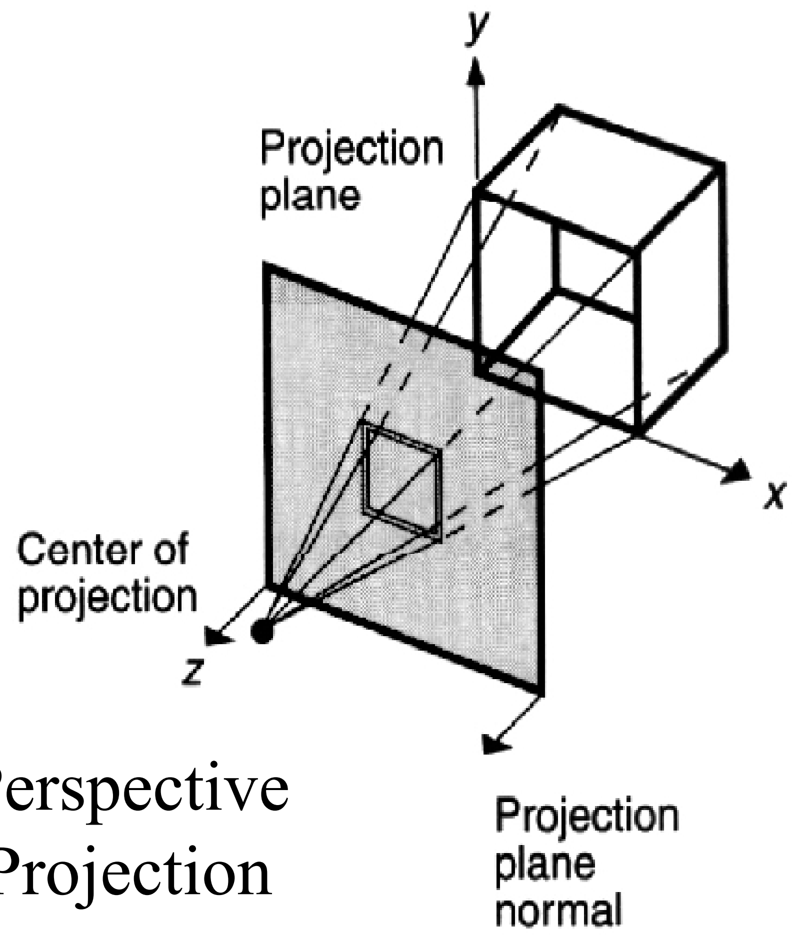
Projection of a 3D object is defined by straight projection rays (projectors) emanating from the center of projection (COP) passing through each point of the object and intersecting the projection plane.

Perspective Projections

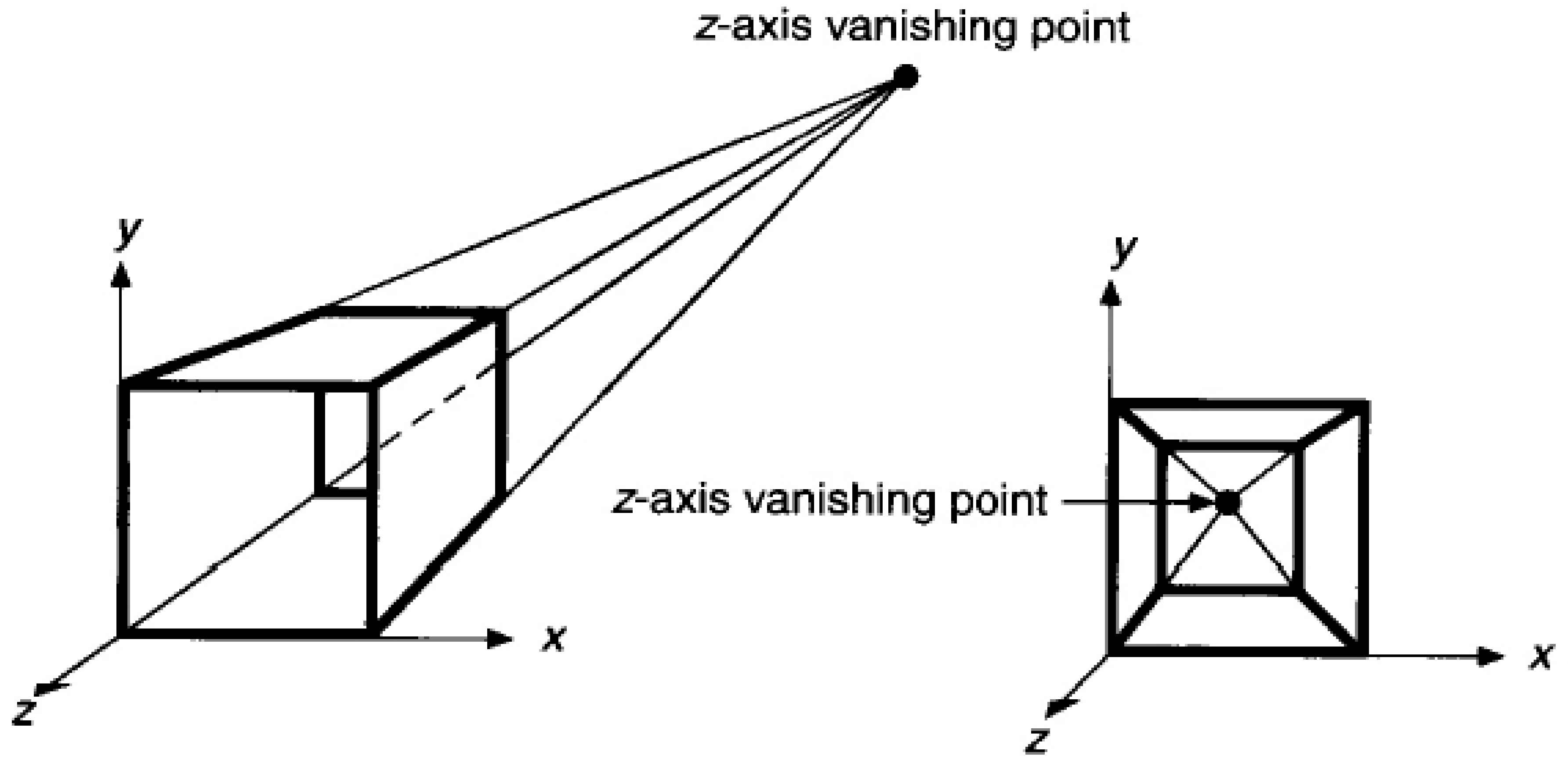
Distance from COP to projection plane is finite. The projectors are not parallel & we specify a center of projection.

Center of Projection is also called the Perspective Reference Point

COP = PRP

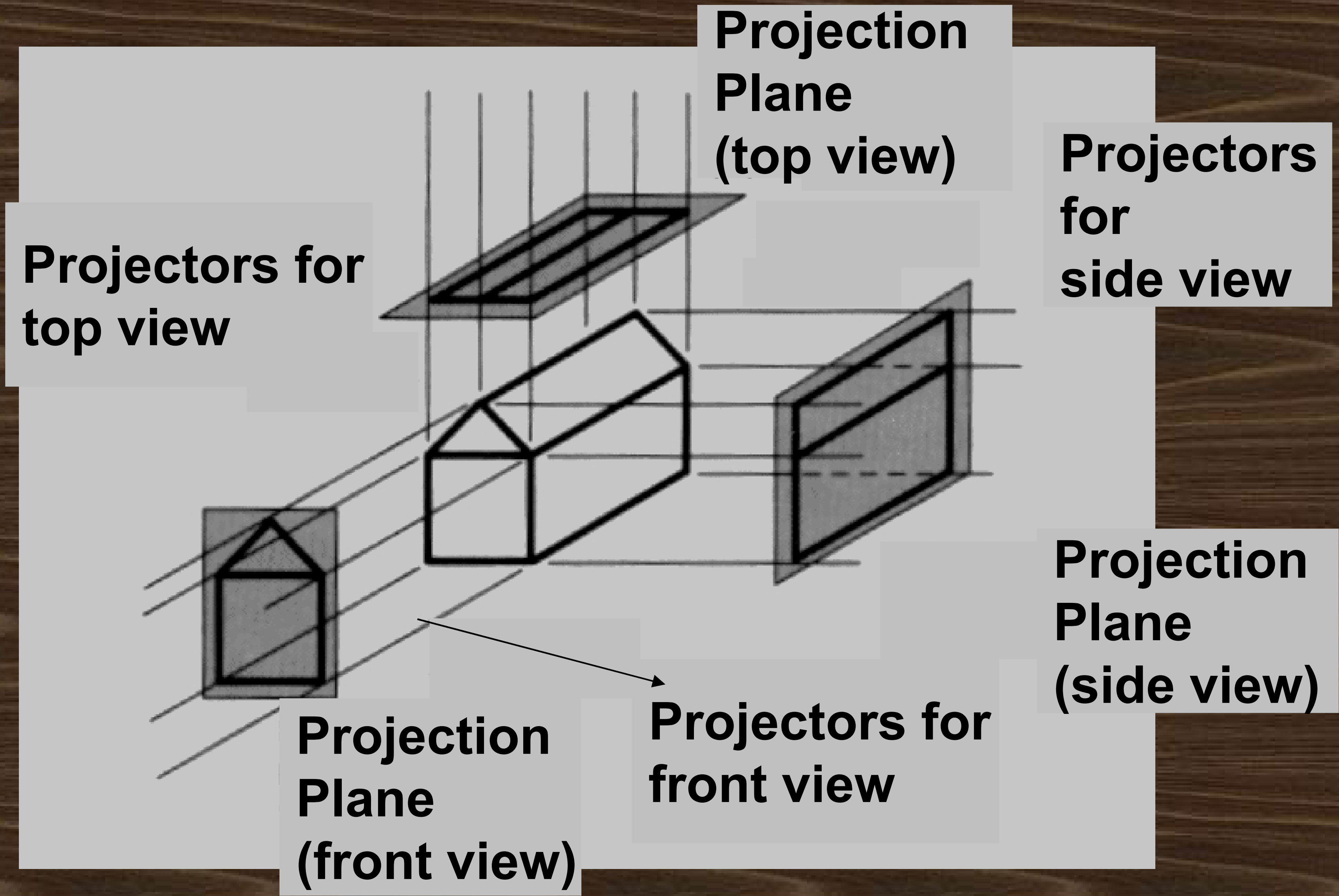


Perspective Projection



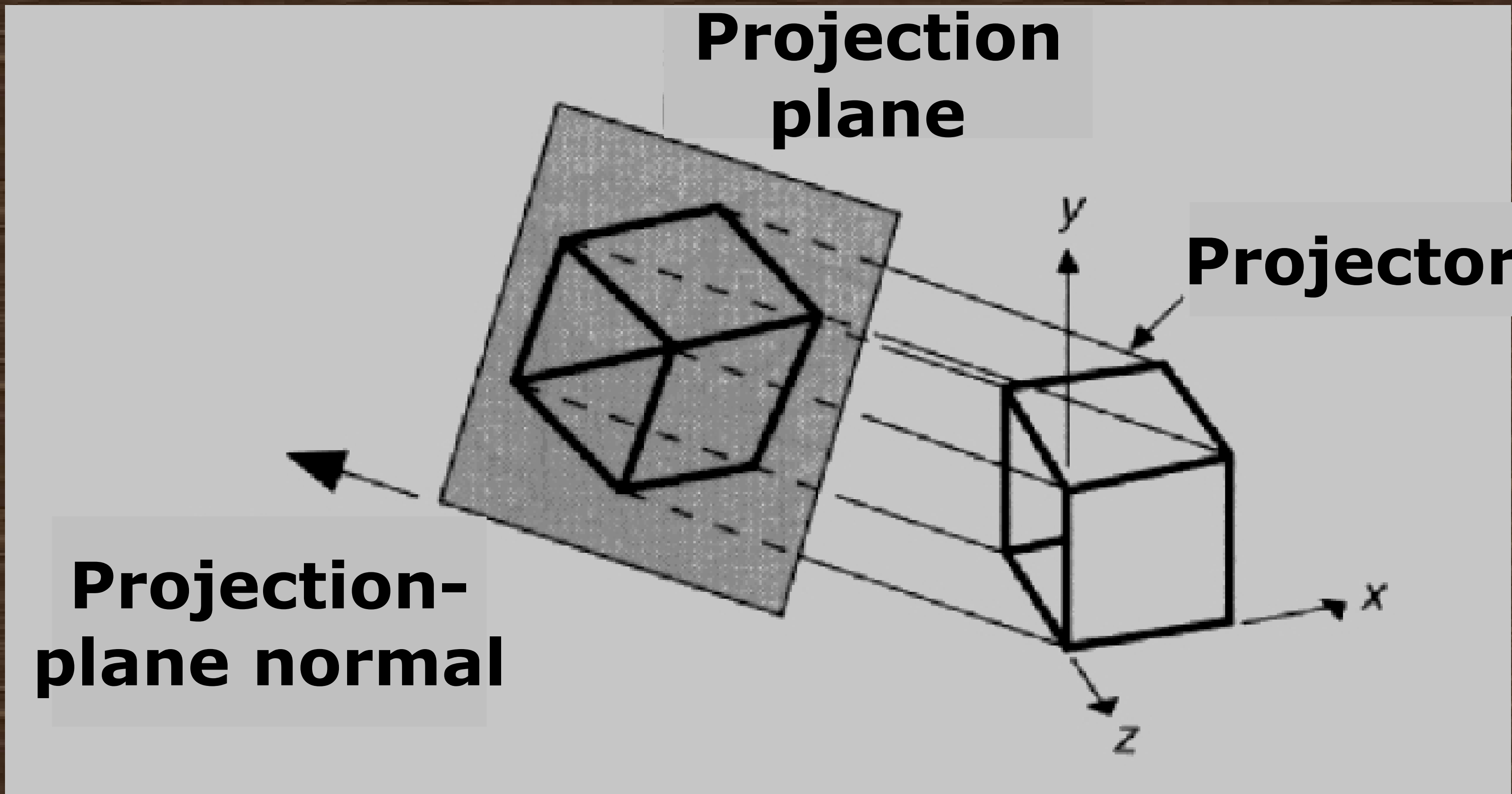
- **Perspective foreshortening:** the size of the perspective projection of the object varies inversely with the distance of the object from the center of projection.

- **Vanishing Point:** The perspective projections of any set of parallel lines that are not parallel to the projection plane converge to a vanishing point.

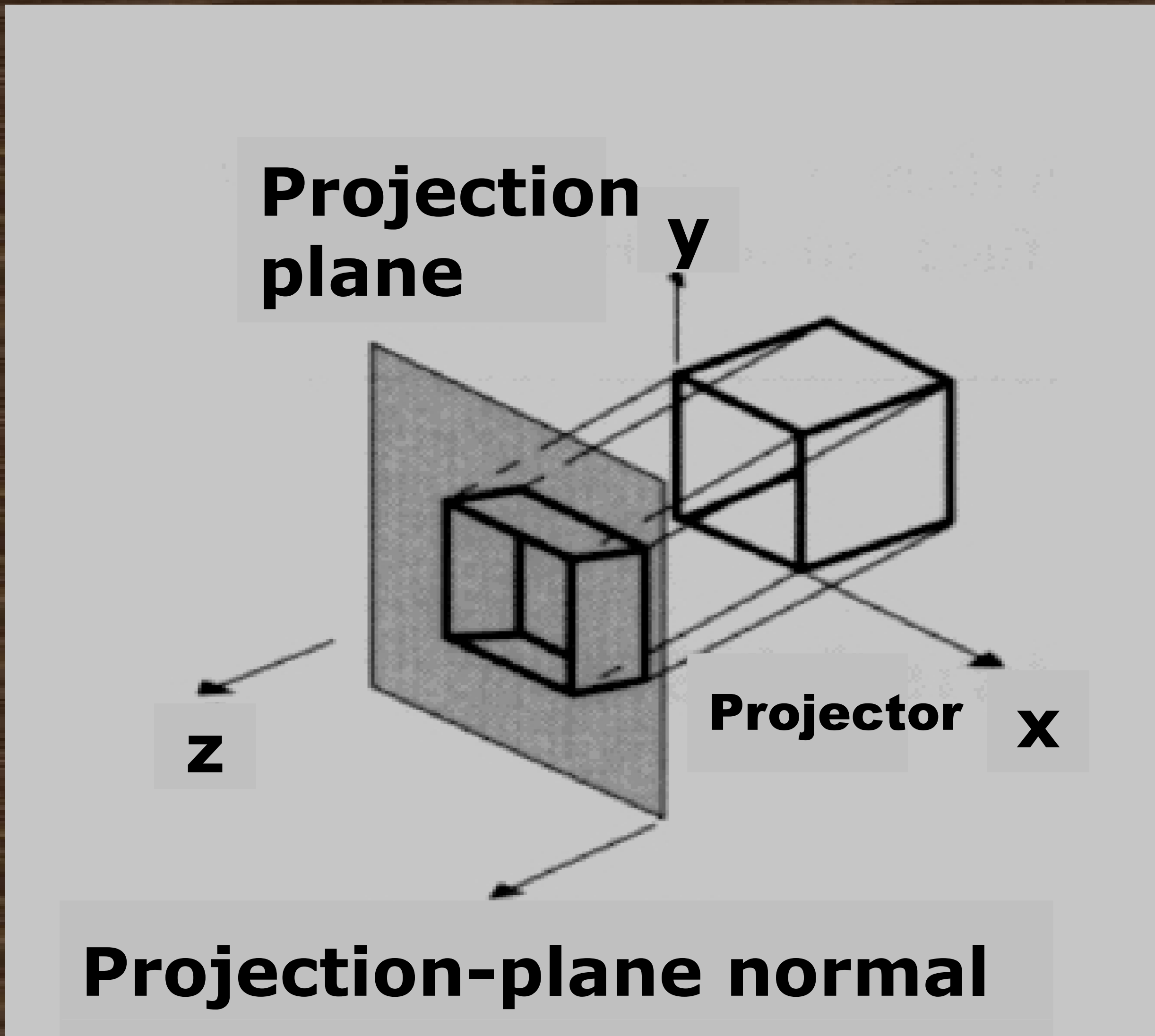


Example of Orthographic Projection

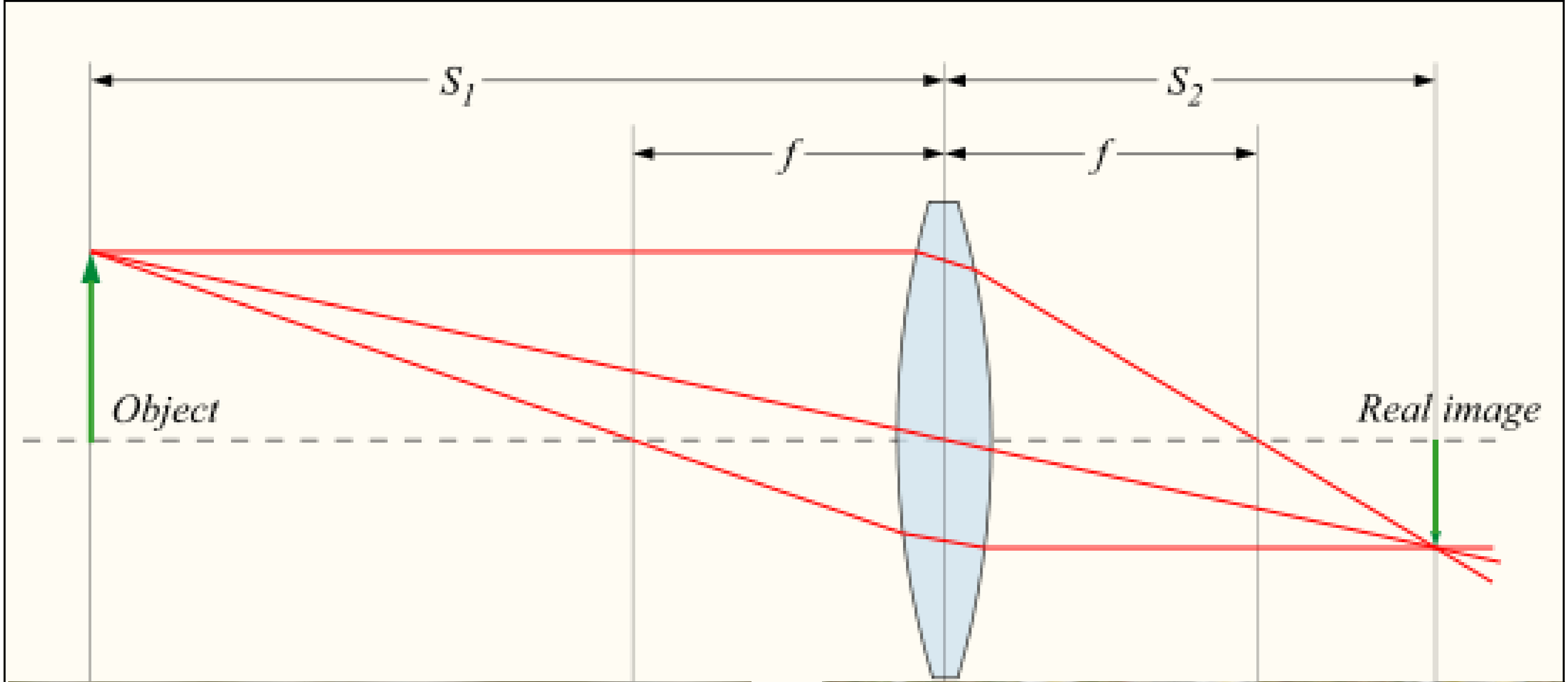
Example of Isometric Projection:



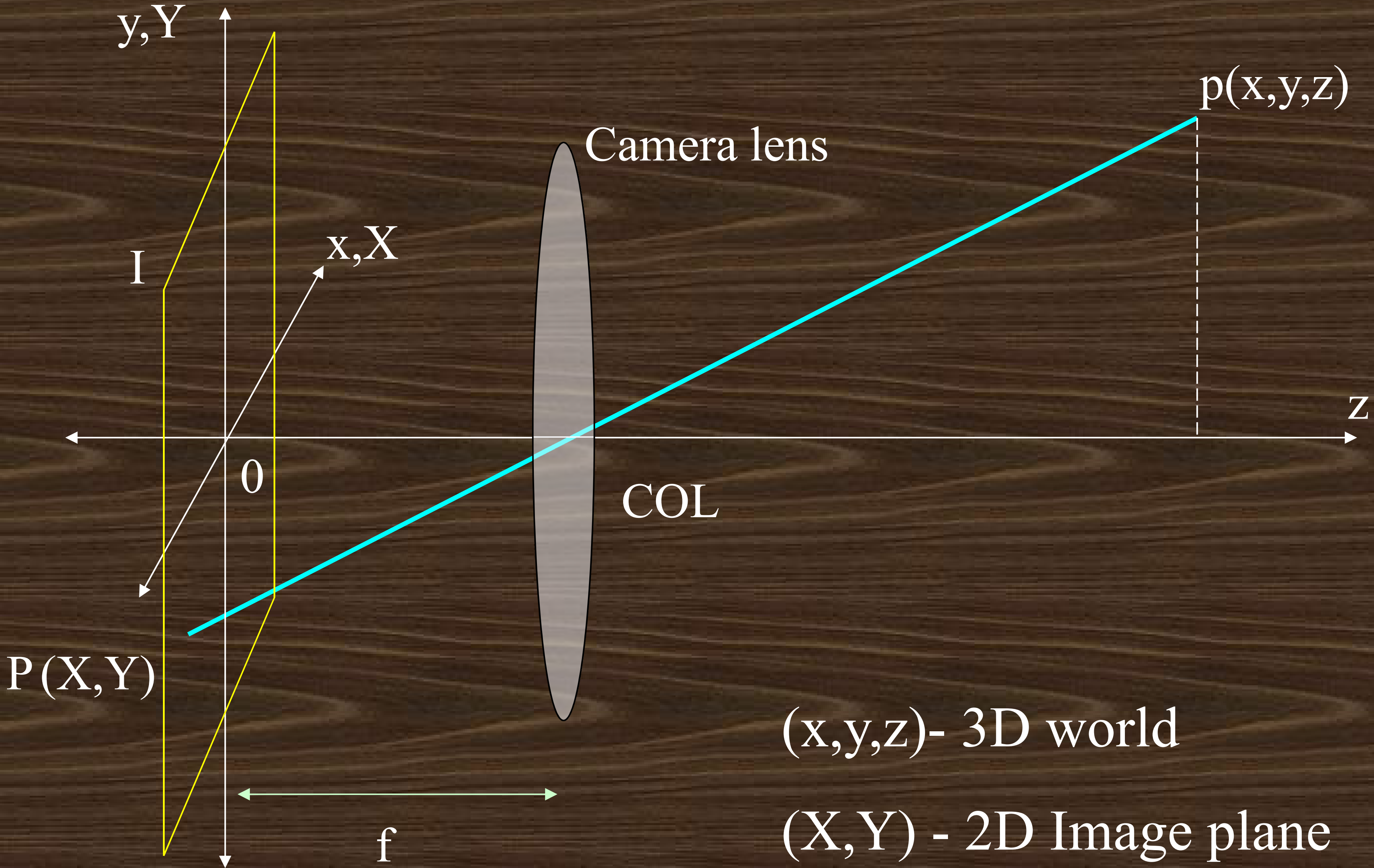
Example Oblique Projection



END OF BASICS


S_1
S_2
f
f
Object
Real image

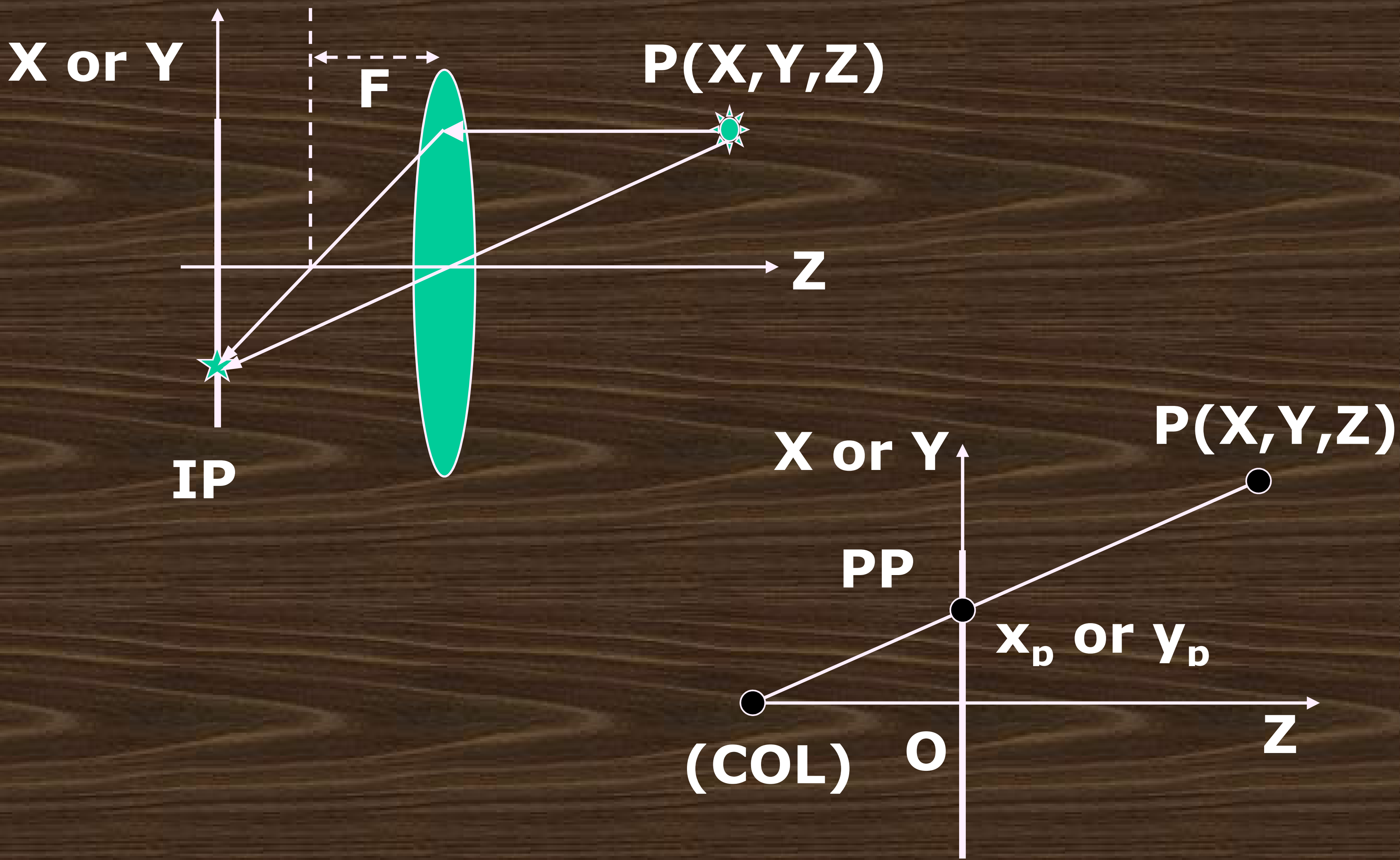
THE CAMERA MODEL: perspective projection

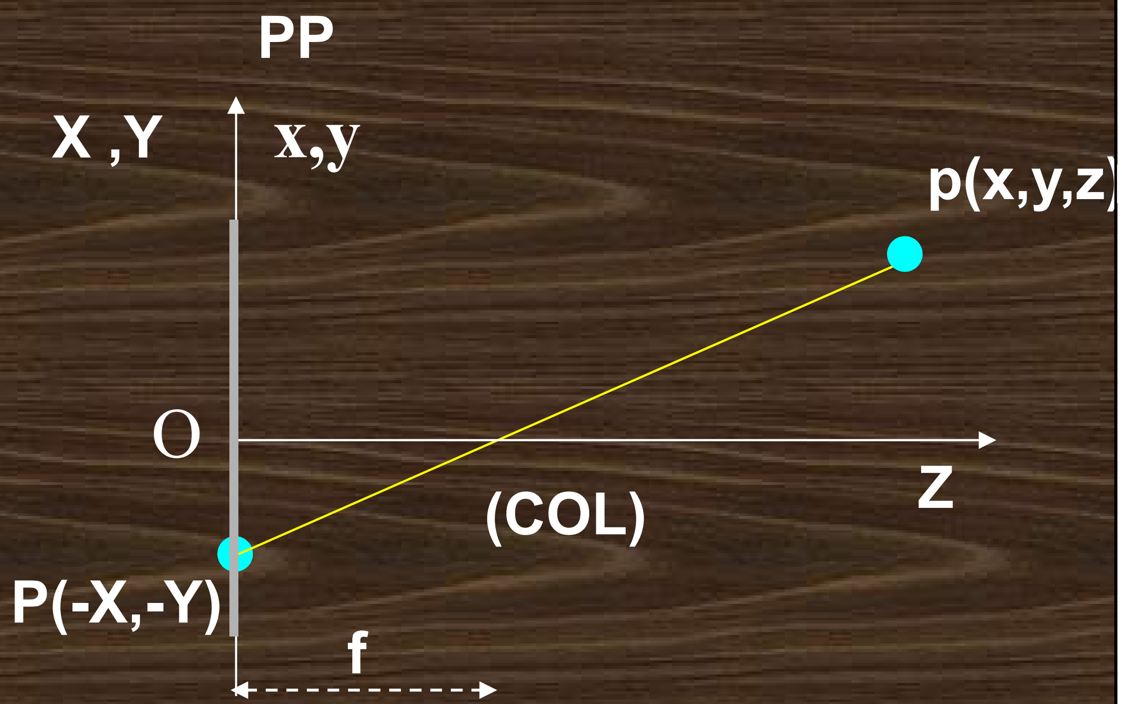


(x,y,z)- 3D world

(X,Y) - 2D Image plane

Perspective Geometry and Camera Models





CASE - 1

By similarity of triangles

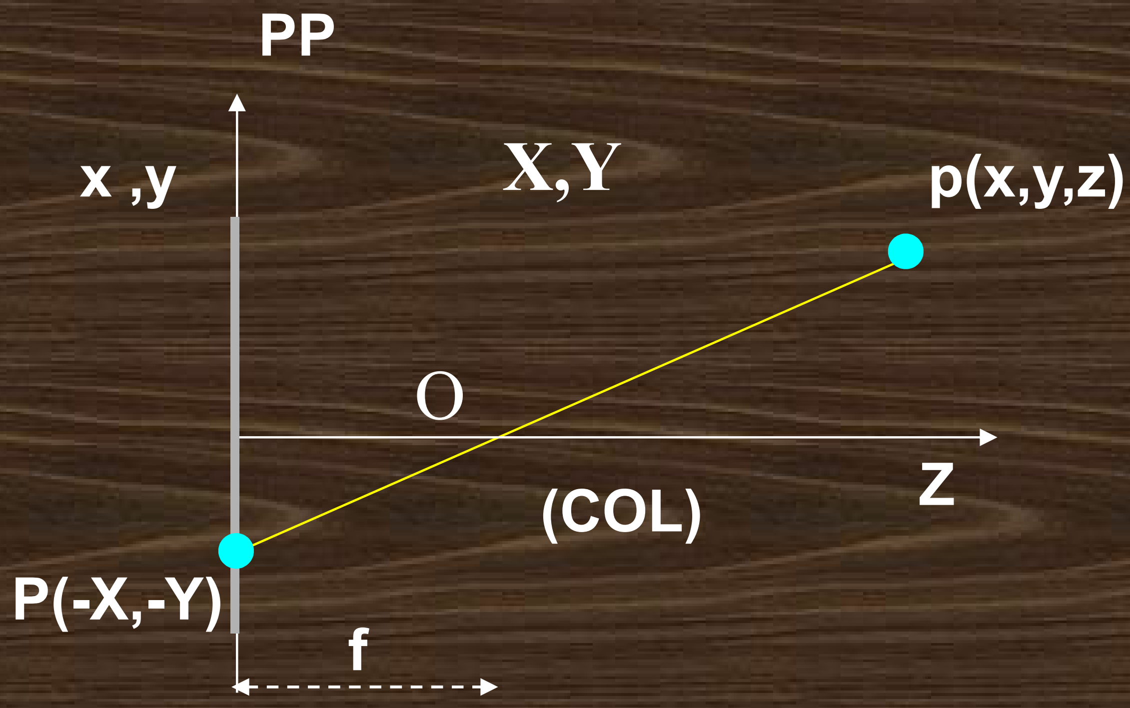
$$\frac{X}{f} = \frac{-x}{z-f}, \qquad \frac{Y}{f} = \frac{-y}{z-f}$$

$$X = \frac{xf}{f-z}, \qquad Y = \frac{yf}{f\text{-}z}$$

$$X = \frac{x}{1-z/f}, \qquad Y = \frac{y}{1-z/f}$$

- Image plane before the camera lens
- Origin of coordinate systems at the image plane
- Image plane at origin of coordinate system

CASE - 1.1



By similarity of triangles

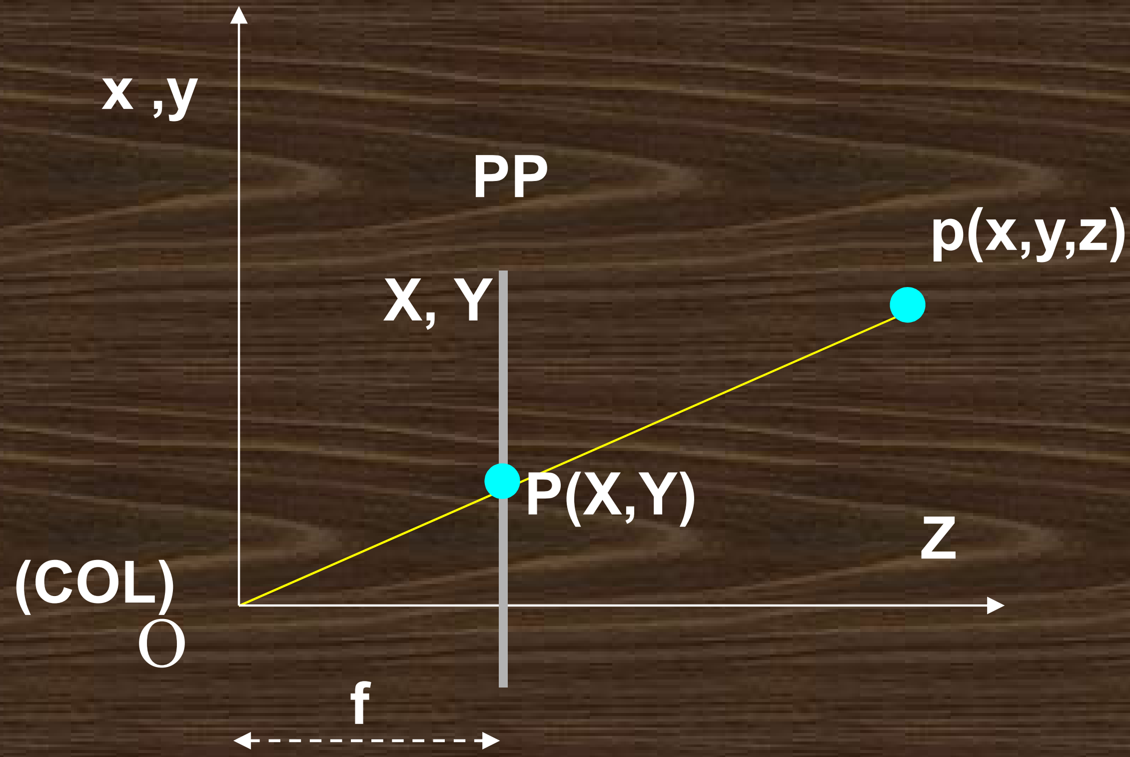
$$\frac{-X}{-f} = \frac{x}{z}, \qquad \frac{-Y}{-f} = \frac{y}{z}$$

$$X = \frac{xf}{z}, \qquad Y = \frac{yf}{z}$$

$$X = \frac{x}{z/f}, \quad Y = \frac{y}{z/f}$$

- Image plane before the camera lens
- Origin of coordinate systems at the camera lens
- Image plane at origin of coordinate system

CASE - 2



By similarity of triangles

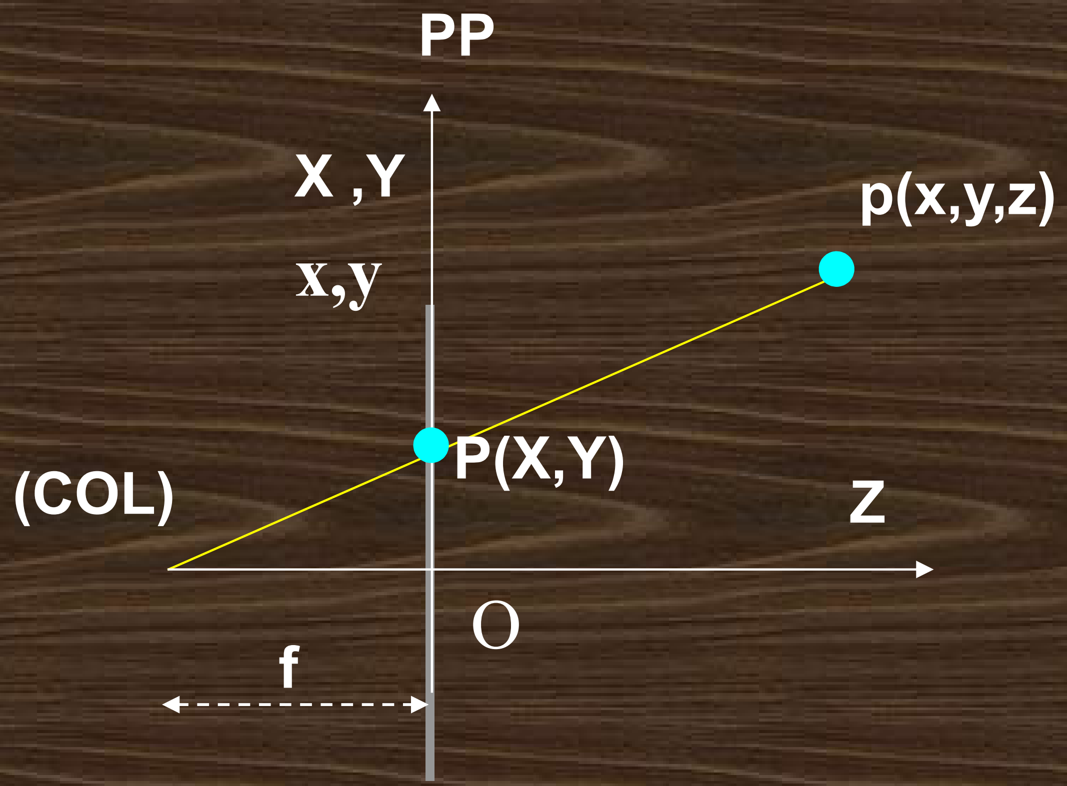
$$\frac{X}{f} = \frac{x}{z}, \qquad \frac{Y}{f} = \frac{y}{z}$$

$$X = \frac{xf}{z}, \qquad Y = \frac{yf}{z}$$

$$X = \frac{x}{z/f}, \qquad Y = \frac{y}{z/f}$$

- Image plane after the camera lens
- Origin of coordinate systems at the camera lens
- Focal length f

CASE – 2.1



By similarity of triangles

$$\frac{X}{f} = \frac{x}{f+z}, \qquad \frac{Y}{f} = \frac{y}{f+z}$$

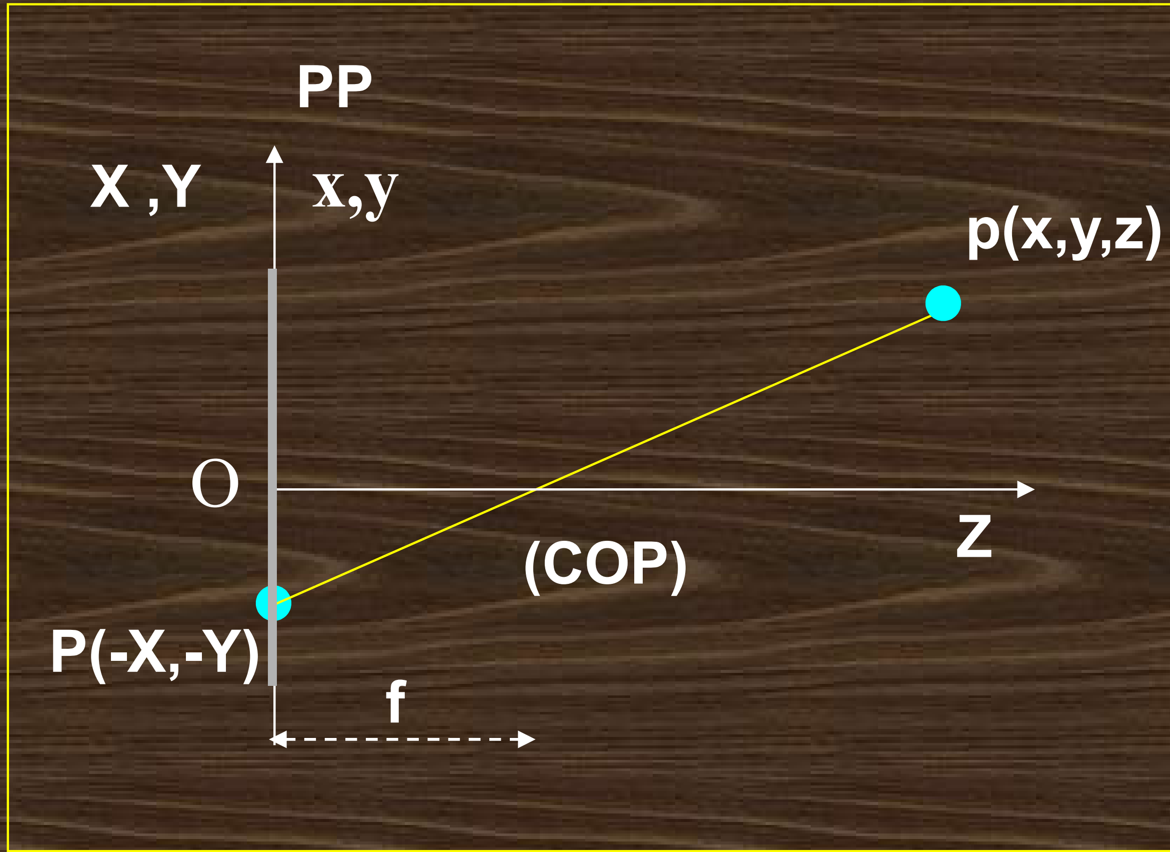
$$X = \frac{xf}{f+z}, \qquad Y = \frac{yf}{f+z}$$

$$X = \frac{x}{1+z/f}, \qquad Y = \frac{y}{1+z/f}$$

- Image plane after the camera lens
- Origin of coordinate system not at COP
- Image plane origin coincides with 3D world origin

Consider the first case ….

- Note that the equations are non-linear
- We can develop a matrix formulation of the equations given below



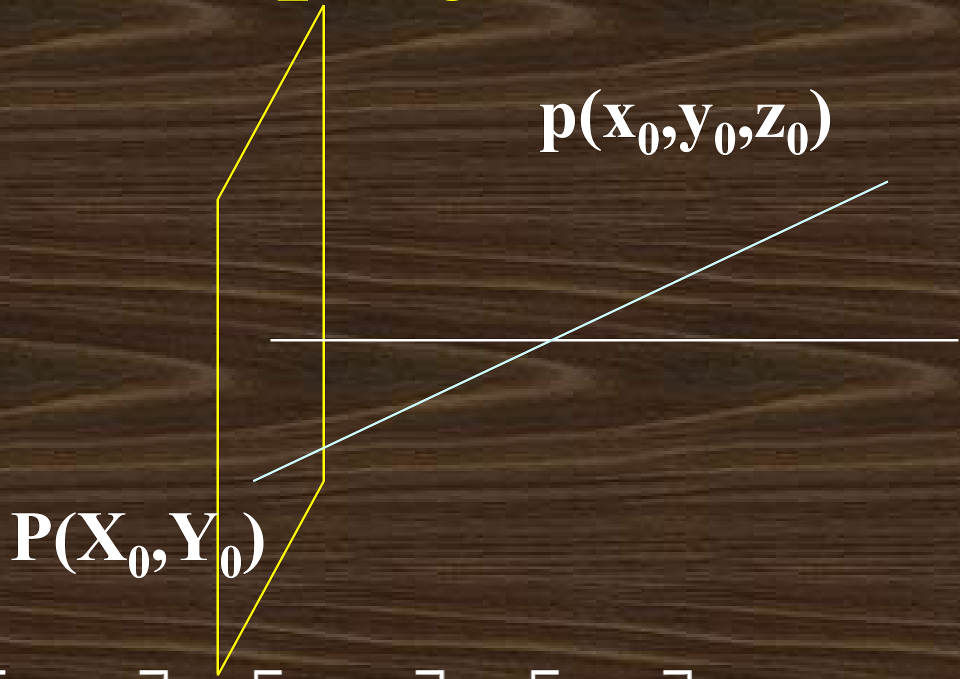
$$X = \frac{x}{1 - z/f}, \quad Y = \frac{y}{1 - z/f}$$

$$\begin{bmatrix} X \\ Y \\ Z \\ k' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/f & 1 \end{bmatrix} \begin{bmatrix} kx \\ ky \\ kz \\ k \end{bmatrix}$$

(Z is not important and is eliminated)

Inverse perspective projection

$$P^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/f & 1 \end{bmatrix}$$



$$w_h = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/f & 1 \end{bmatrix} \begin{bmatrix} kX_0 \\ kY_0 \\ 0 \\ k \end{bmatrix} = \begin{bmatrix} kX_0 \\ kY_0 \\ 0 \\ k \end{bmatrix} = \begin{bmatrix} X_0 \\ Y_0 \\ 0 \\ 1 \end{bmatrix}$$

Hence no 3D information can be retrieved with the inverse transformation

So we introduce the dummy variable i.e. the depth Z

Let the image point be represented as: $[kX_0 \;\; kY_0 \;\; kZ \;\; k]^T$

$$w_h = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/f & 1 \end{bmatrix} \begin{bmatrix} kX_0 \\ kY_0 \\ kZ \\ k \end{bmatrix} =$$

$$z_0 = \frac{fZ}{f+Z} \Rightarrow Z = \frac{fz_0}{f-z_0} \Rightarrow \frac{f}{f+Z} = \frac{z_0}{Z} = \frac{f-z_0}{f}$$

$$\Rightarrow x_0 = \frac{X_0}{f}(f-z_0), \quad y_0 = \frac{Y_0}{f}(f-z_0)$$

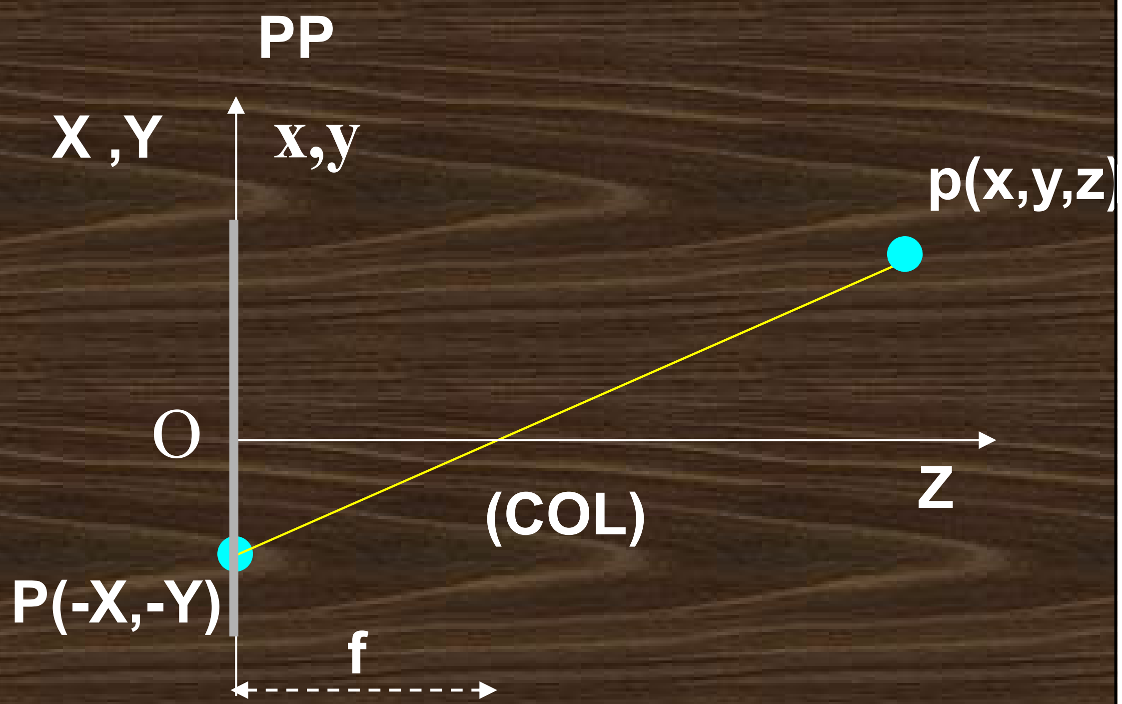
CASE - 1

Forward: 3D to 2D

$$\frac{X}{f} = \frac{-x}{z-f}, \quad \frac{Y}{f} = \frac{-y}{z-f}$$

$$X = \frac{xf}{f-z}, \quad Y = \frac{yf}{f\text{-}z}$$

$$X = \frac{x}{1-z/f}, \quad Y = \frac{y}{1-z/f}$$



Inverse: 2D to 3D

$$x_0 = \frac{X_0}{f}(f-z_0), \quad y_0 = \frac{Y_0}{f}(f-z_0)$$

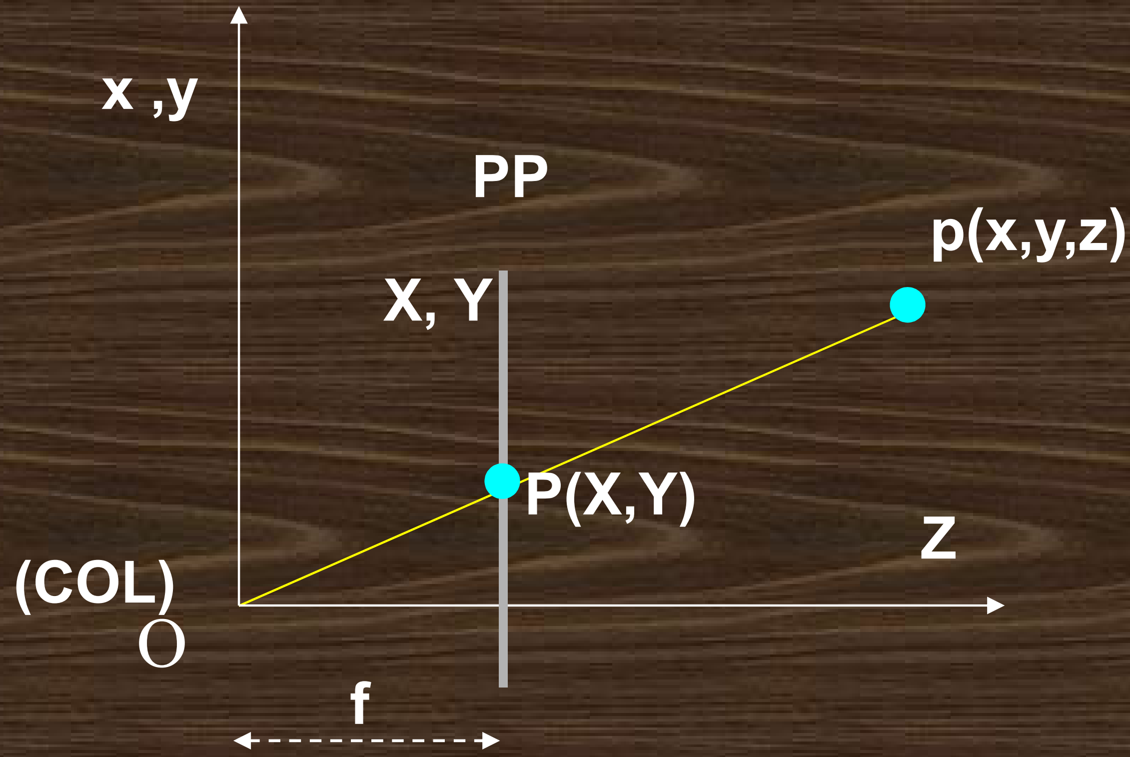
CASE - 2

Forward: 3D to 2D

$$\frac{X}{f} = \frac{x}{z}, \qquad \frac{Y}{f} = \frac{y}{z}$$

$$X = \frac{xf}{z}, \qquad Y = \frac{yf}{z}$$

$$X = \frac{x}{z/f}, \quad Y = \frac{y}{z/f}$$



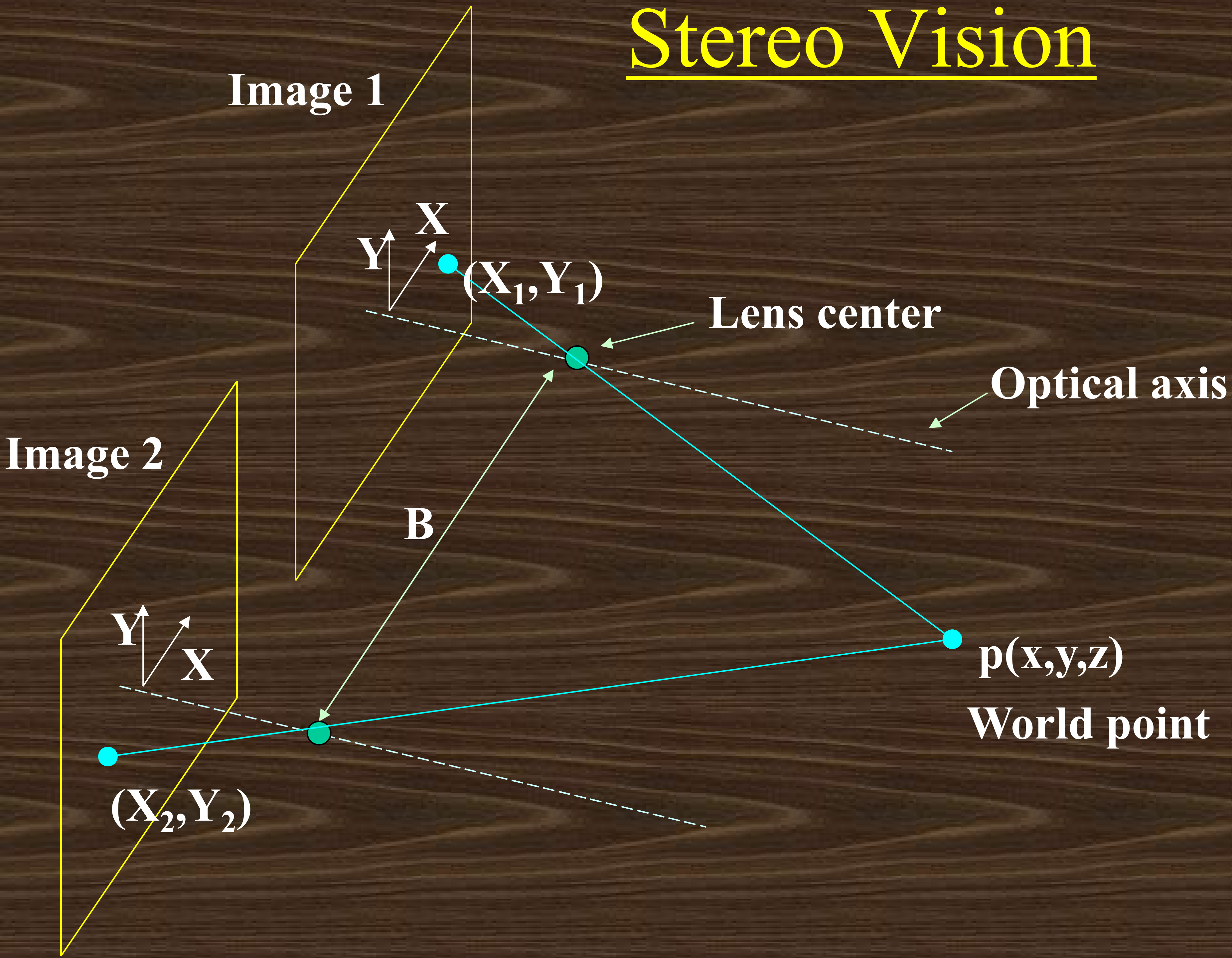
Inverse: 2D to 3D

$$x_0 = \frac{z_0.X_0}{f}, \qquad y_0 = \frac{z_0.Y_0}{f}$$

Observations about Perspective projection

- 3D scene to image plane is a one to one transformation (unique correspondence)
- For every image point no unique world coordinate can be found
- So depth information cannot be retrieved using a single image ? What to do?
- Would two (2) images of the same object (from different viewing angles) help?
- Termed - Stereo Vision

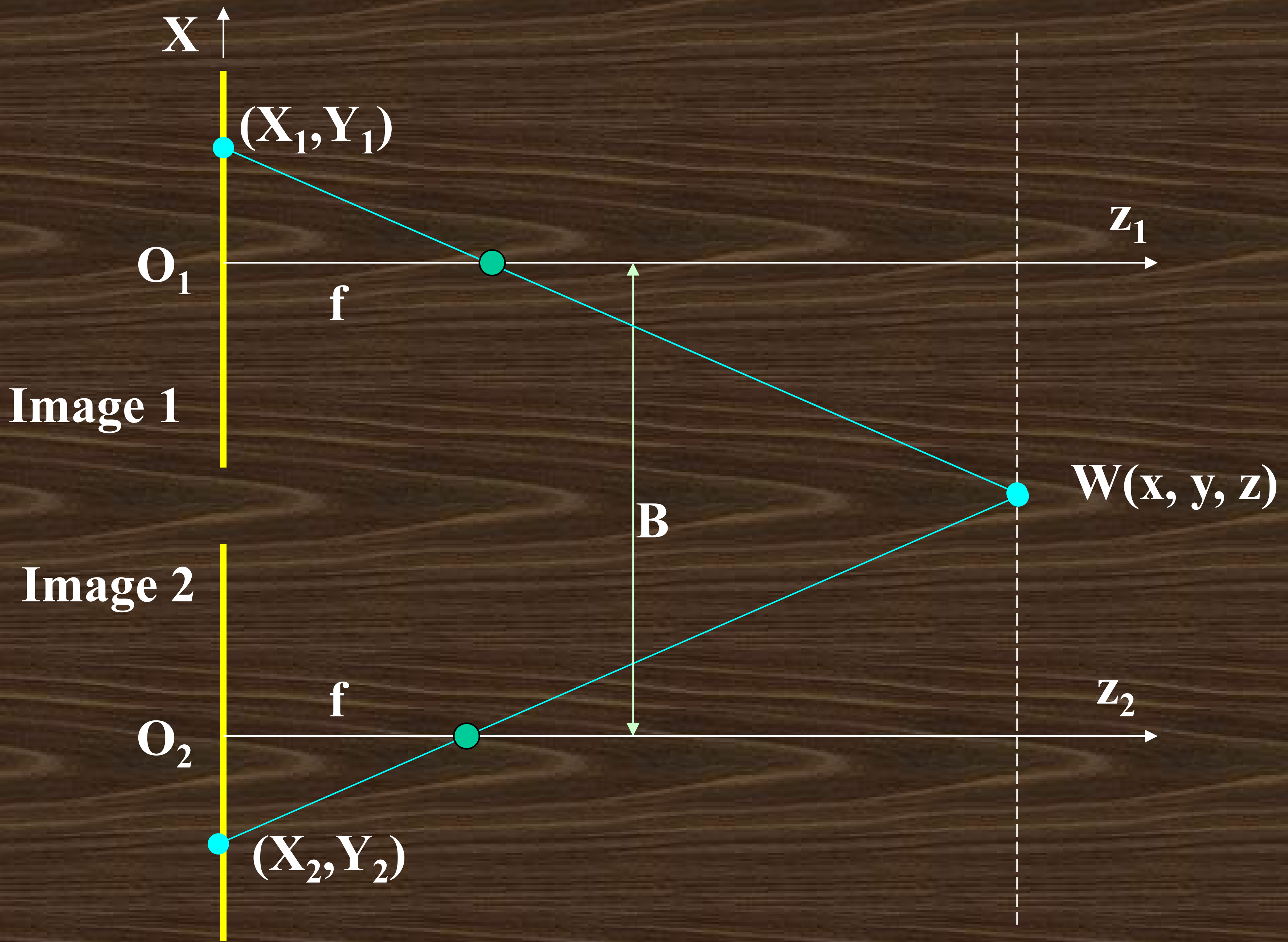
Stereo Vision



Stereo Vision (2)

- **Stereo imaging involves obtaining two separate image views of an object (in this discussion the world point)**
- **The distance between the centers of the two lenses is called the baseline width.**
- **The projection of the world point on the two image planes is (X_1, Y_1) and (X_2, Y_2)**
- **The assumption is that the cameras are identical**
- **The coordinate system of both cameras are perfectly aligned differing only in the x-coordinate location of the origin.**
- **The world coordinate system is also bought into the coincidence with one of the image X, Y planes (say image plane 1) . So y, z coordinates are same for both the camera coordinate systems.**

Top view of the stereo imaging system with origin at center of first imaging plane.



First bringing the first camera into coincidence with the world coordinate system and then using the second camera coordinate system and directly applying the formula we get:

$$x_1 = \frac{X_1}{f}(f - z_1), \quad x_2 = \frac{X_2}{f}(f - z_2)$$

Because the separation between the two cameras is B

$$x_2 = x_1 + B, \quad z_1 = z_2 = z(?) \quad /*\mathrm{Solve\ it\ now}*/$$

$$x_1 = \frac{X_1}{f}(f - z), \quad x_1 + B = \frac{X_2}{f}(f - z)$$

$$B = \frac{(X_2 - X_1)}{f}(f - z), \quad z = f - \frac{fB}{(X_2 - X_1)}$$

- The equation above gives the depth directly from the coordinate of the two points
- The quantity given below is called the disparity

$$D = (X_2 - X_1) = \frac{fB}{(f - z)}$$

- The most difficult task is to find out the two corresponding points in different images of the same scene – **the correspondence problem**.
- Once the correspondence problem is solved – (non-analytical), we get D. Then obtain depth using:

$$z = f - \frac{fB}{(X_2 - X_1)} = f[1 - B/D]$$

Alternate Model – Case III

$$\frac{X}{f} = \frac{x}{z}, \quad \frac{Y}{f} = \frac{y}{z}$$

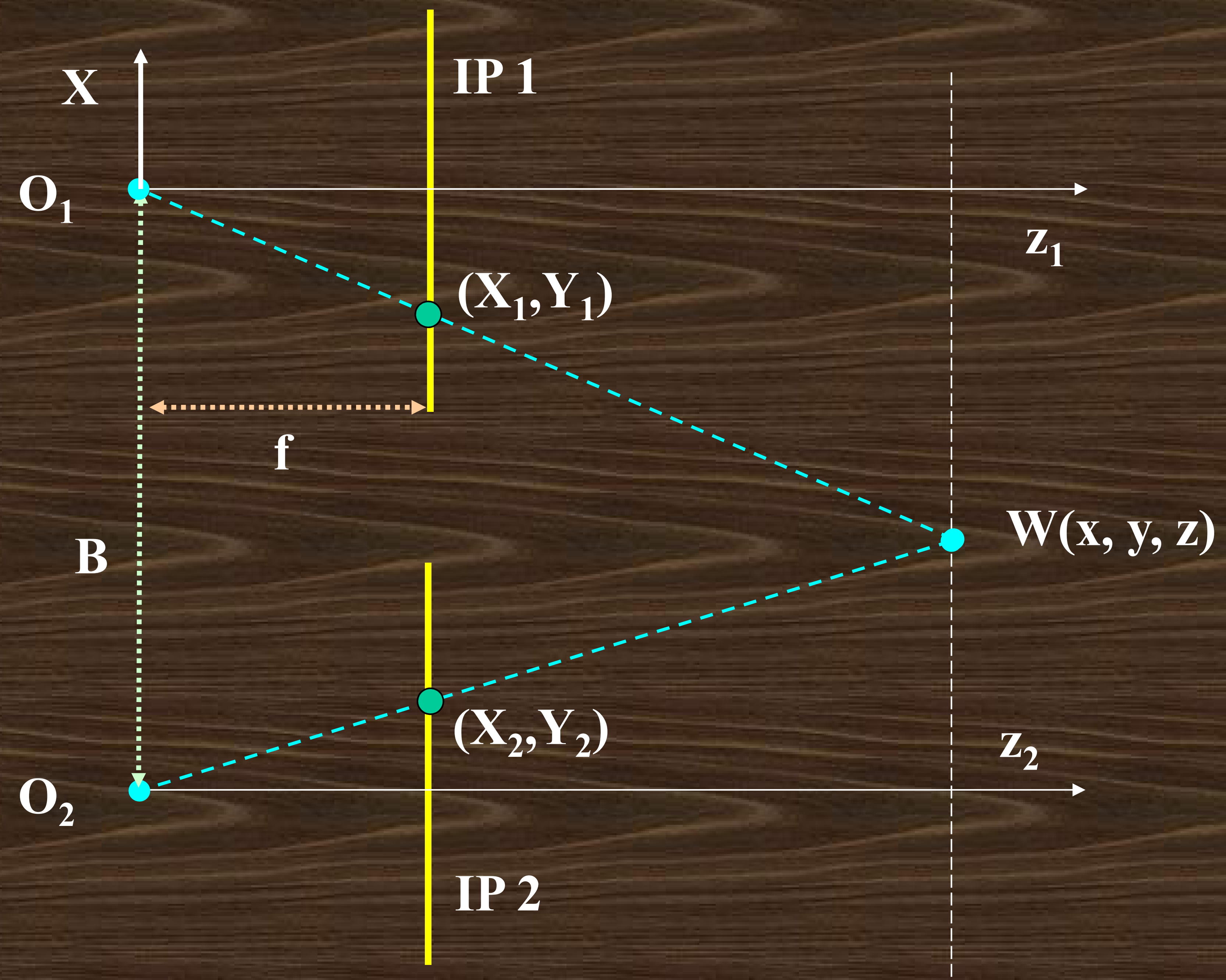
$$x = \frac{Xz}{f}, \quad y = \frac{Yz}{f}$$

$$x_2 = x_1 + B, \quad y_1 = y_2 = y; \quad z_1 = z_2 = z(?).$$

$$x_1 = \frac{X_1 z}{f}, \quad x_2 = x_1 + B = \frac{X_2 z}{f}$$

$$B = \frac{(X_2 - X_1)z}{f}; \quad z = \frac{fB}{(X_2 - X_1)} = {B.f}\Big/{D}$$

Top view of the stereo imaging system with origin at center of first camera lens.



Compare the two solutions

$$z = f - \frac{fB}{(X_2 - X_1)} = f[1 - B/D]$$

$$D = (X_2 - X_1) = \frac{fB}{(f - z)}$$

$$z = \frac{fB}{(X_2 - X_1)} = B.f/D$$

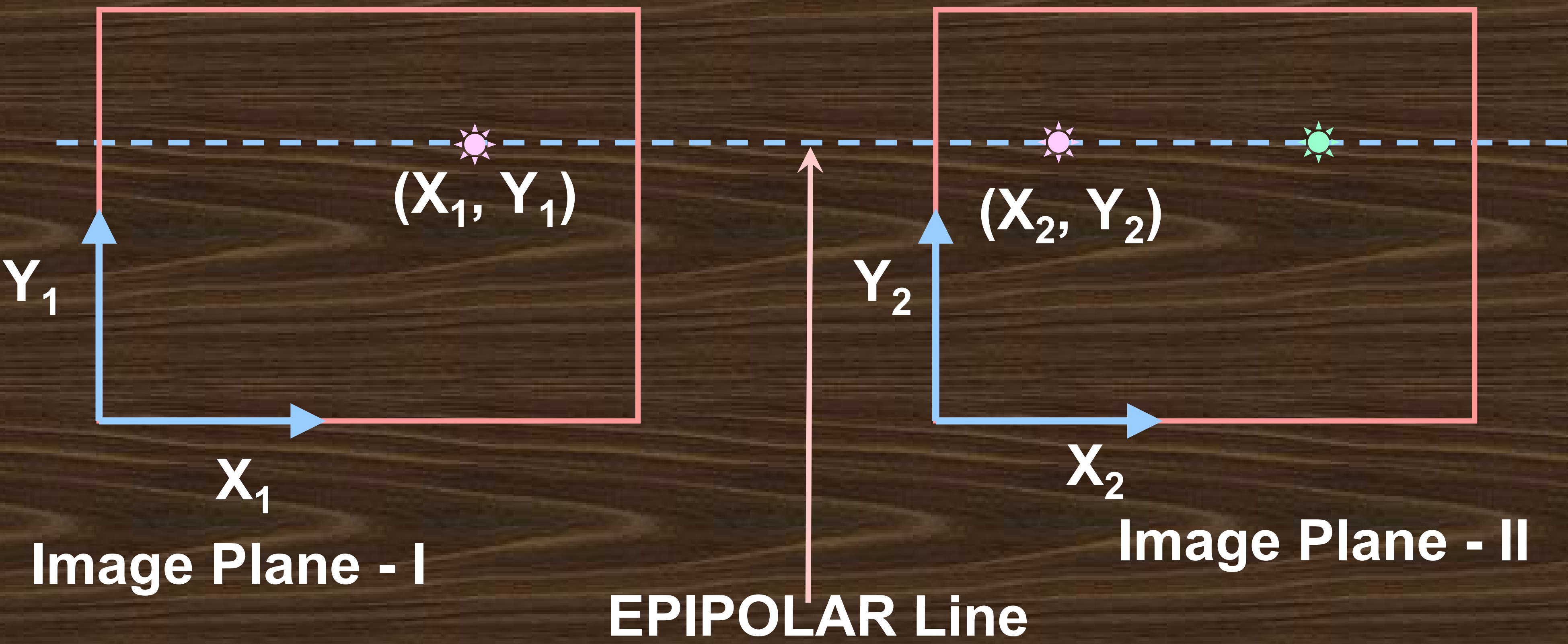
$$D = (X_2 - X_1) = \frac{fB}{z}$$

What do you think of D ?

The Correspondence Problem

$$z = {B.f}\big/{D} \qquad D = (X_1 - X_2) = \frac{fB}{z} \qquad Y_1 = Y_2$$

$$If \quad D > 0; \quad then \;\; X_2 < X_1$$



Error in Depth Estimation

$$z = {B.f}/{D} \qquad \delta(z)/\delta D = -{B.f}/{D^2}$$

Expressing in terms of depth (z), we have:

$$\delta(z)/\delta D = -\frac{B.f}{D^2} = -\frac{z}{D} = -\frac{z^2}{B.f}$$

What is the maximum value of depth (z), you can measure using a stereo setup ?

$$z_{\max} = B.f$$

Even if correspondence is solved correctly, the computation of D may have an error, with an upper bound of 0.5; i.e. $(\delta D)_{max} = 0.5$.

That may cause an error of: $$\delta(z) = -\frac{z^2}{2B.f}$$

Larger baseline width and Focal length (of the camera) reduces the error and increases the maximum value of depth that may be estimated.

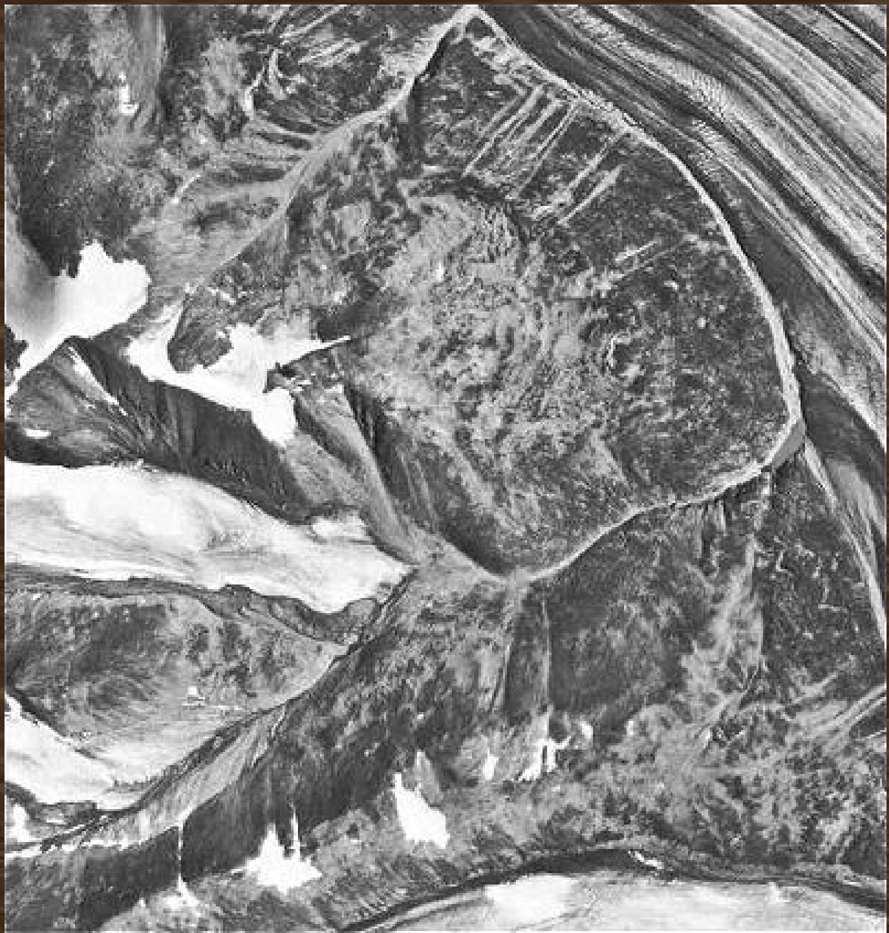
What about the minimum value of depth (object closest to the cameras) ?

$$z_{\mathbf{min}} = \boldsymbol{B.f\,/\,D}_{\mathbf{max}}$$

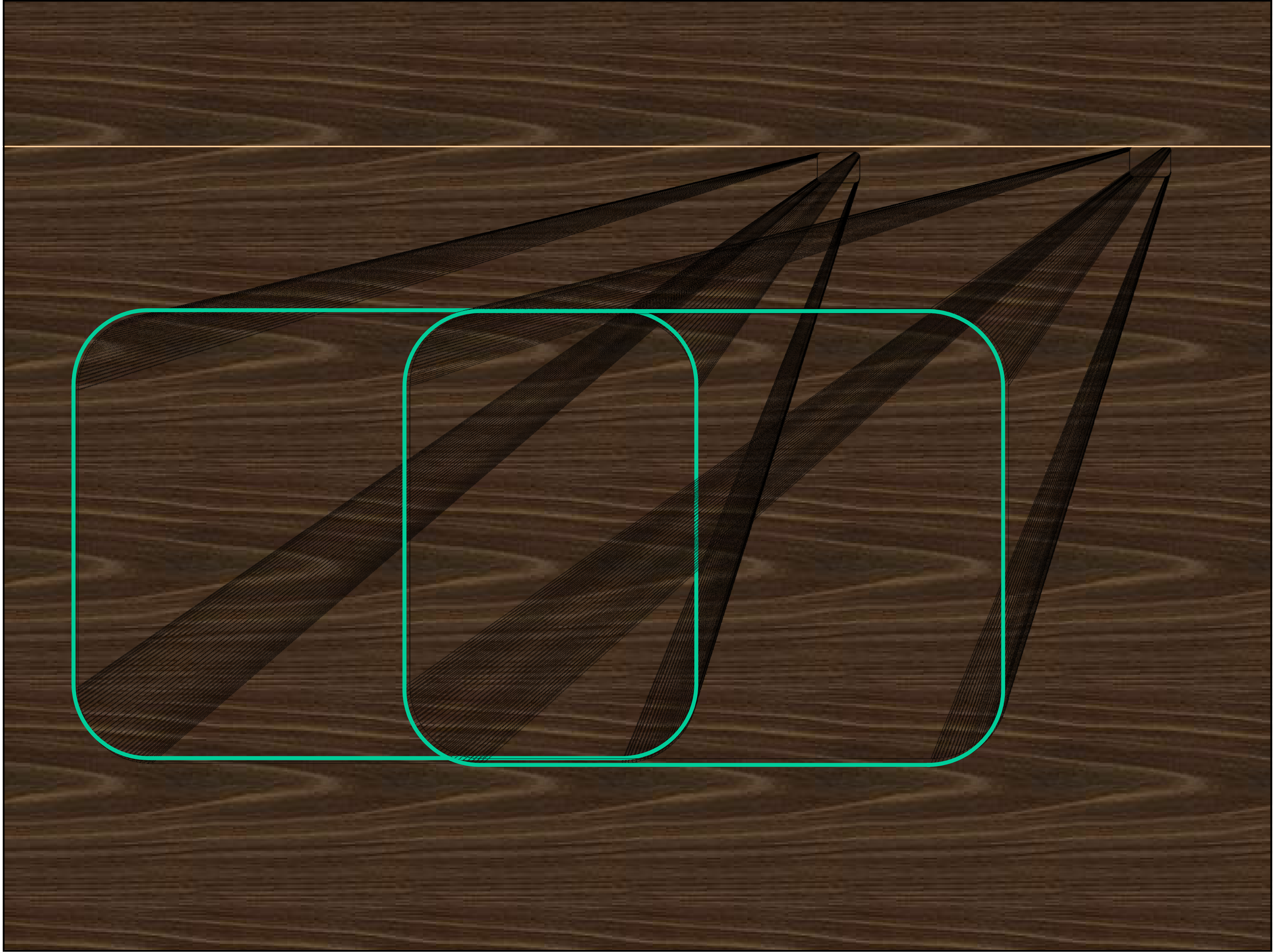
What is D_{max} ?

$$D_{\max} = X_{\max}$$

X_{max} depends on f and image resolution (in other words, angle of field-of-view or FOV).

General Stereo Views



We can also have arbitrary pair of views from two cameras.

- The baseline may not lie on any of the principle axis
- The viewing axes of the cameras may not be parallel
- Unequal focal lengths of the cameras
- The coordinate systems of the image planes may not be aligned

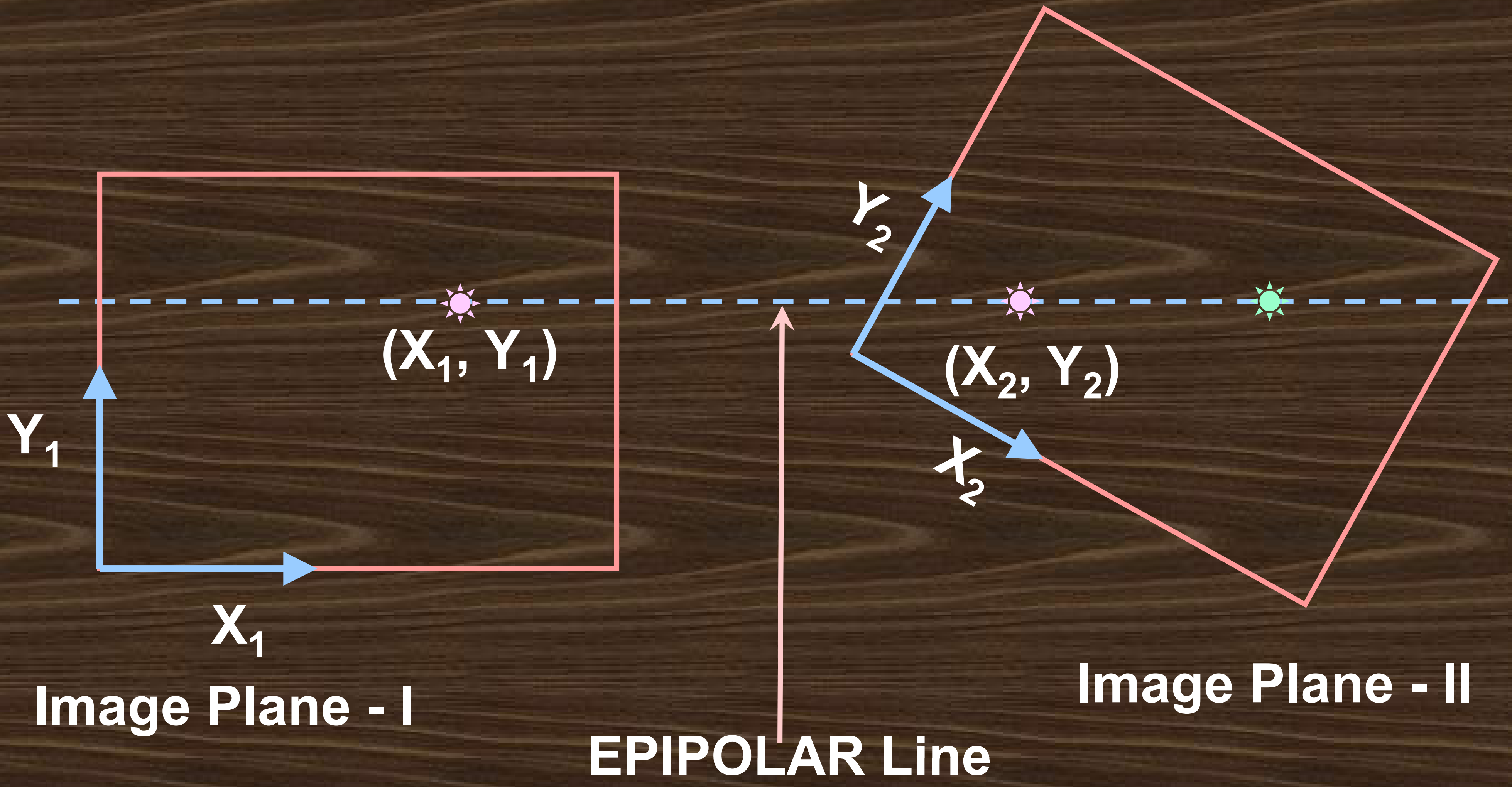
Take home exercises/problems:

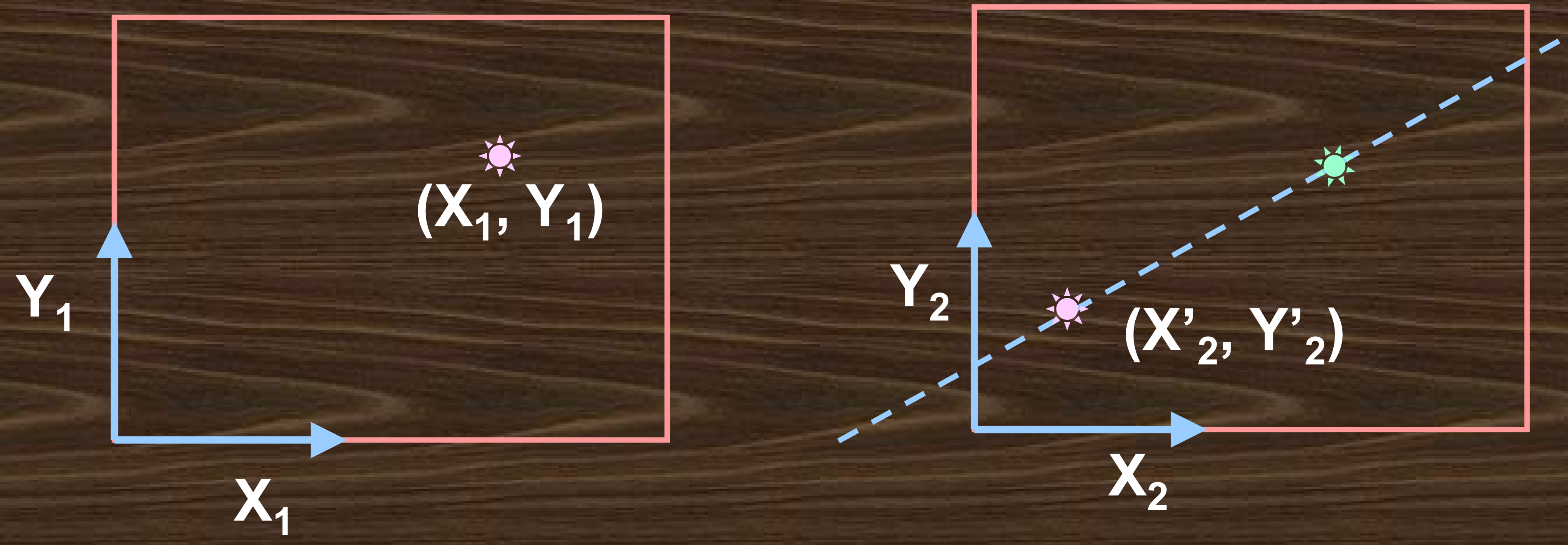
What about Epipolar line in cases above ?

How do you derive the equation of an epipolar line ?

In general we may have multiple views (2 or more) of a scene. Typically used for 3D surveillance tasks.

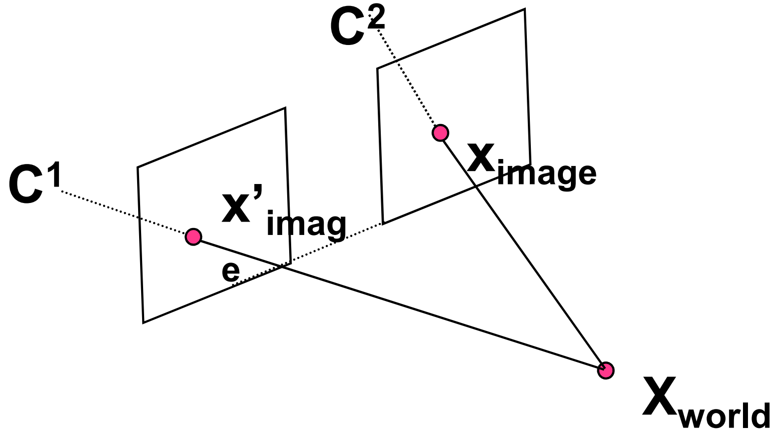
The Epipolar line in case of Arbitrary Views



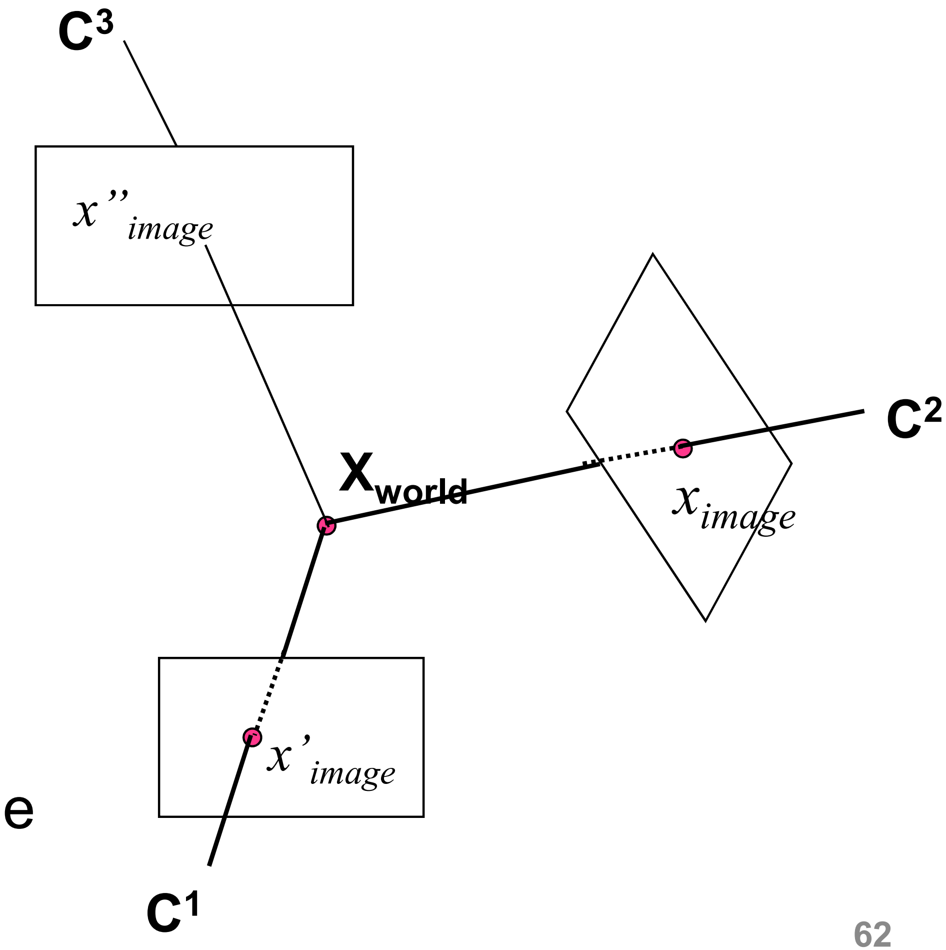

(X1, Y1)
Y1
X1
Y2
(X'2, Y'2)
X2

Classical Depth Estimation

- Depth estimation of image points – need at least two views of the same object



General Stereo

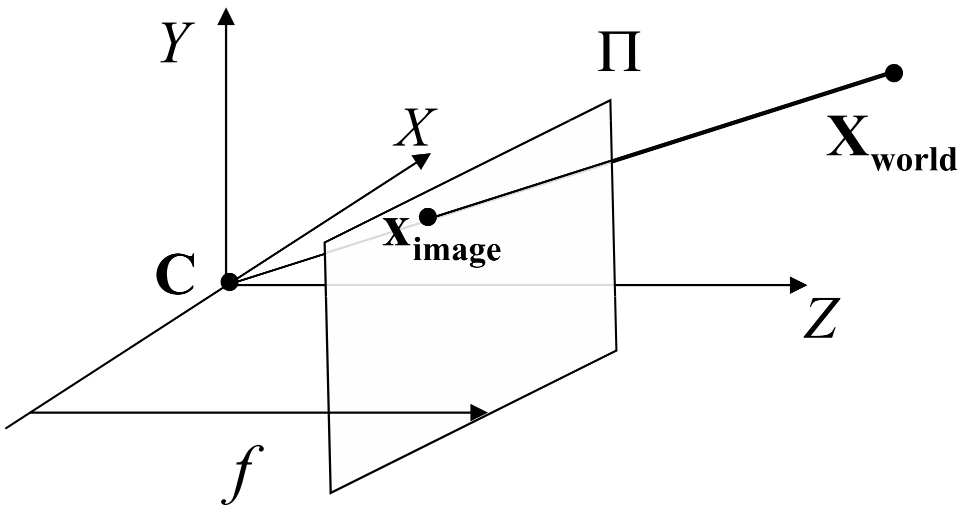


Arbitrary multiple view geometry

Camera Image formulation

- Action of eye is simulated by an abstract camera model (pinhole camera model)
- 3D real world is captured on the image plane. Image is projection of 3D object on a 2D plane.

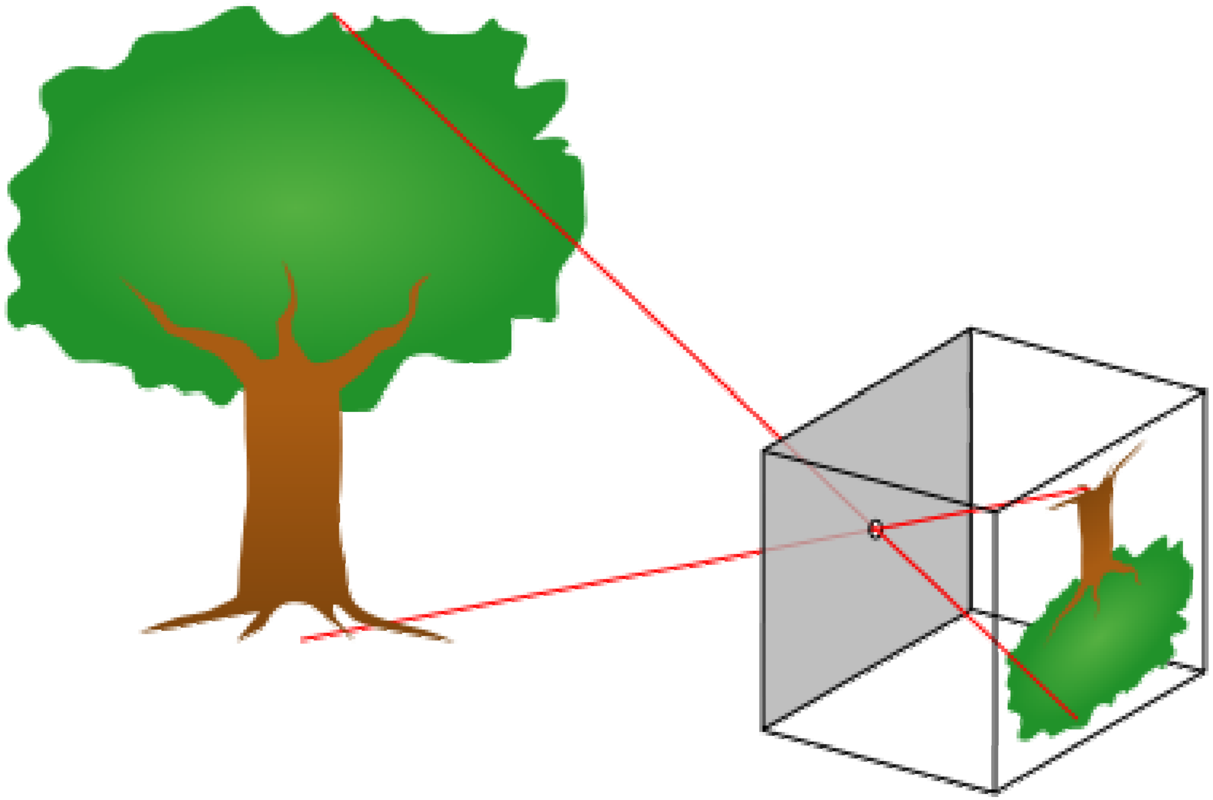
$$F : (X_w, Y_w, Z_w) \rightarrow (x_i, y_i)$$



$$\mathbf{C} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{f} & 0 \end{pmatrix} \sim \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\mathbf{X}_{\mathbf{world}} = (X_w, Y_w, Z_w)$$

$$\mathbf{x}_{\mathbf{image}} = \left(f \frac{X_w}{Z_w}, f \frac{Y_w}{Z_w}\right)$$



Pinhole Camera schematic diagram

Camera Geometry

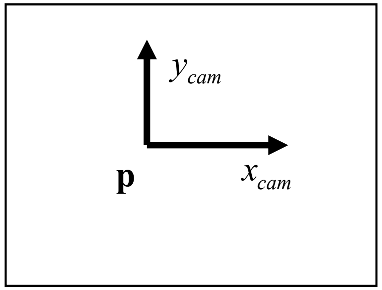
- Camera can be considered as a projection matrix, $\mathbf{x} = \mathrm{P}_{3*4}\mathbf{X}$
  - A pinhole camera has the projection matrix as

$$P = diag(f, f, 1)[I \quad | \quad 0]$$

  - Principal point offset

$$(X, Y, Z)^{\mathrm{T}} \rightarrow (fX / Z + p_x, fY / Z + p_y)^{\mathrm{T}}$$

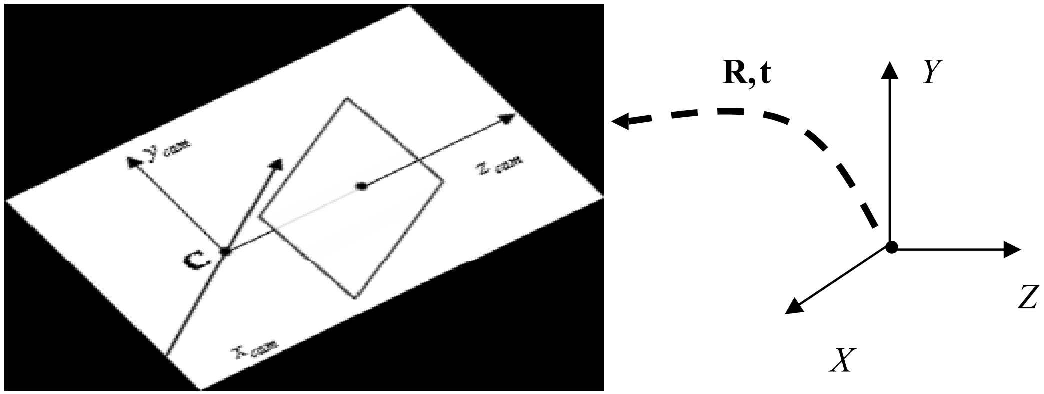
$$K = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad \mathbf{x} = \mathrm{K}[\mathrm{I} \quad | \quad \mathbf{0}]\mathbf{X}$$



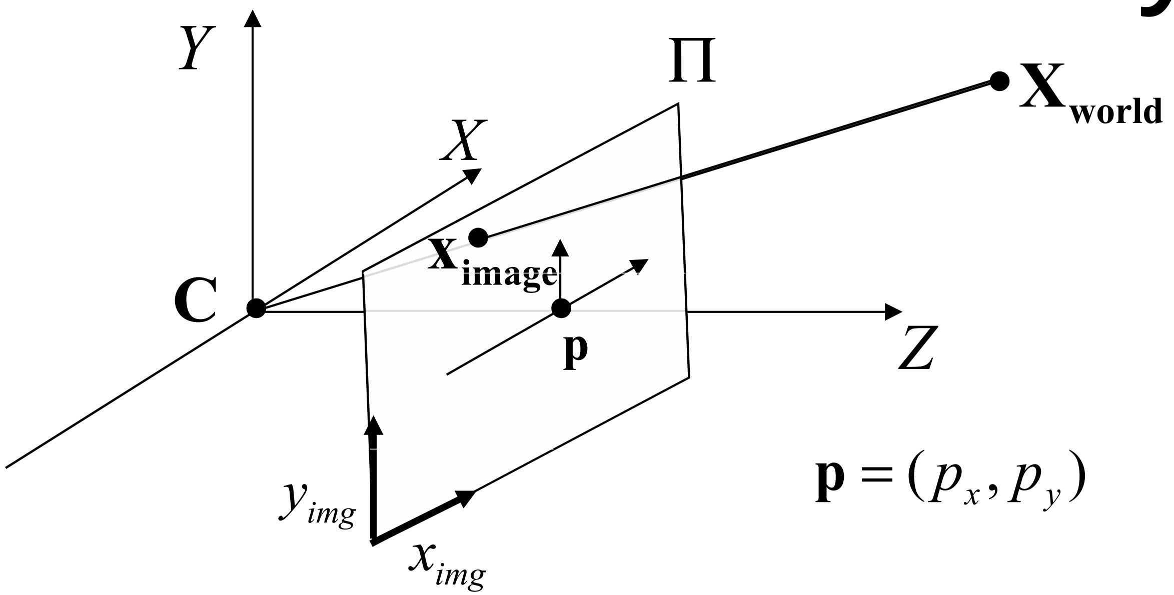
$\mathbf{p} = (p_x, p_y)$

  - Camera with rotation and translation

$$\mathbf{x} = \mathrm{K}[\mathrm{R} \mid \mathbf{t}]\mathbf{X}$$



Camera Geometry



$\mathbf{p} = (p_x, p_y)$

Camera internal parameters

$$K = \begin{bmatrix} \alpha_x & s & p_x \\ & \alpha_y & p_y \\ & & 1 \end{bmatrix}$$

α_x Scale factor in x- coordinate direction

α_y Scale factor in y- coordinate direction

s Camera skew

$\frac{\alpha_x}{\alpha_y}$ Aspect ratio

Camera matrix, $P = K[R \mid \mathbf{t}]$

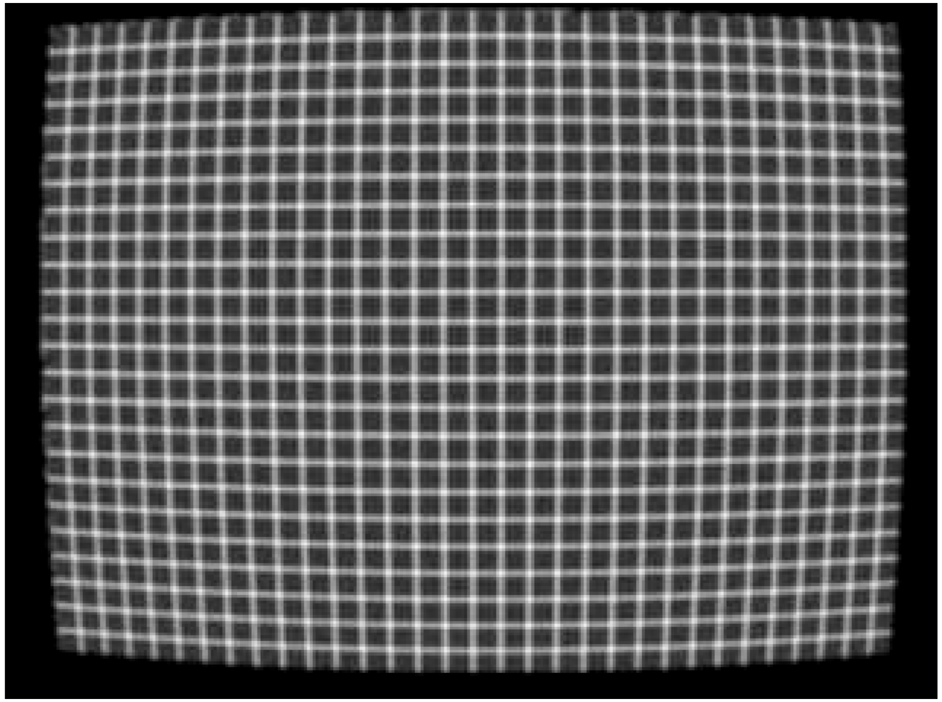
R Rotation

$\mathbf{t}$ Translation vector

Camera skew factor/parameter, s:

The parameter "s" accounts for a possible non-orthogonality of the axes in the image plane.

This might be the case if the rows and columns of pixels on the sensor are not perpendicular to each other.



Pincushion, non-linear distrortion

The Reconstruction Problem

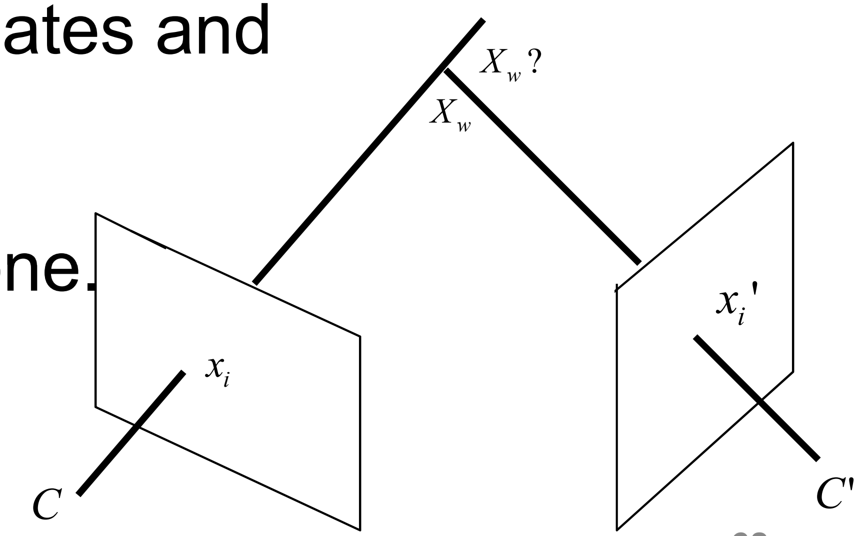
- Given a set of images of a particular 3D scene, can we reconstruct the scene back?
- 3D representation of an object is difficult because of the problem of depth estimation.
- Image is projection of 3D object on a 2D plane.

$$F:(X_w, Y_w, Z_w) \rightarrow (x\ , y)$$

(X_w, Y_w, Z_w) are real world coordinates and

(x, y) are Image coordinates

- Reverse mapping is not one to one.



3D Reconstruction

- Given a set of images of a particular 3D scene, can we reconstruct the scene back?

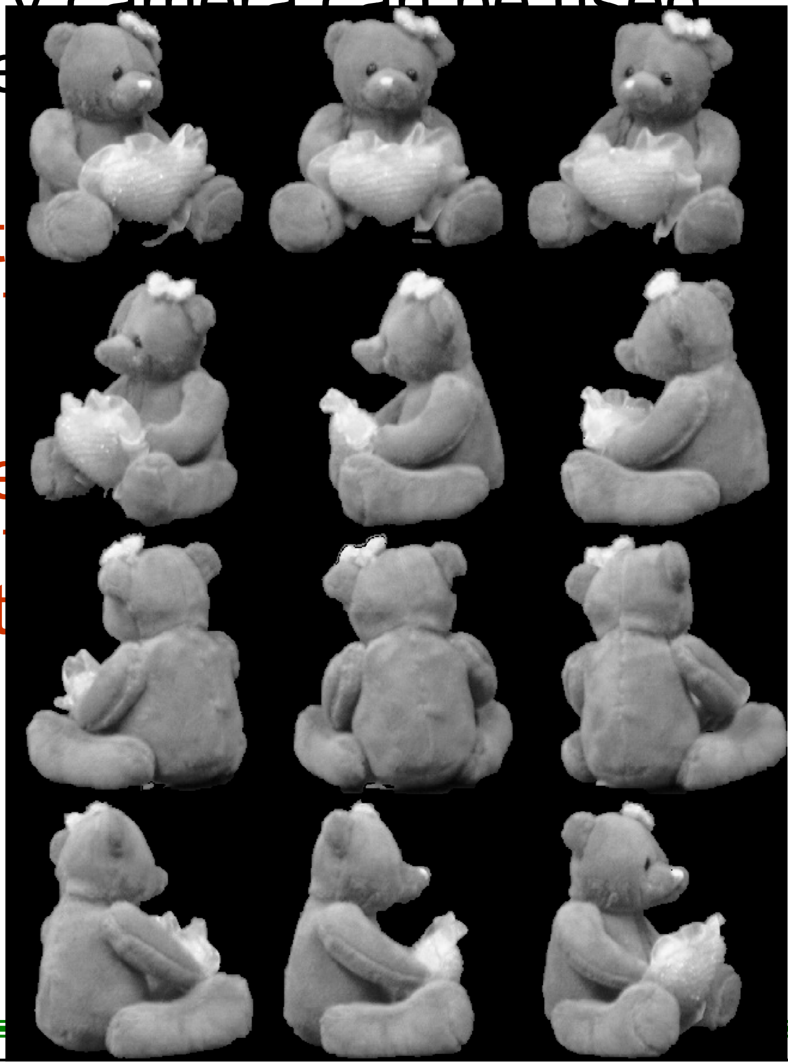
[a]

- Classical inverse problem of the computer vision

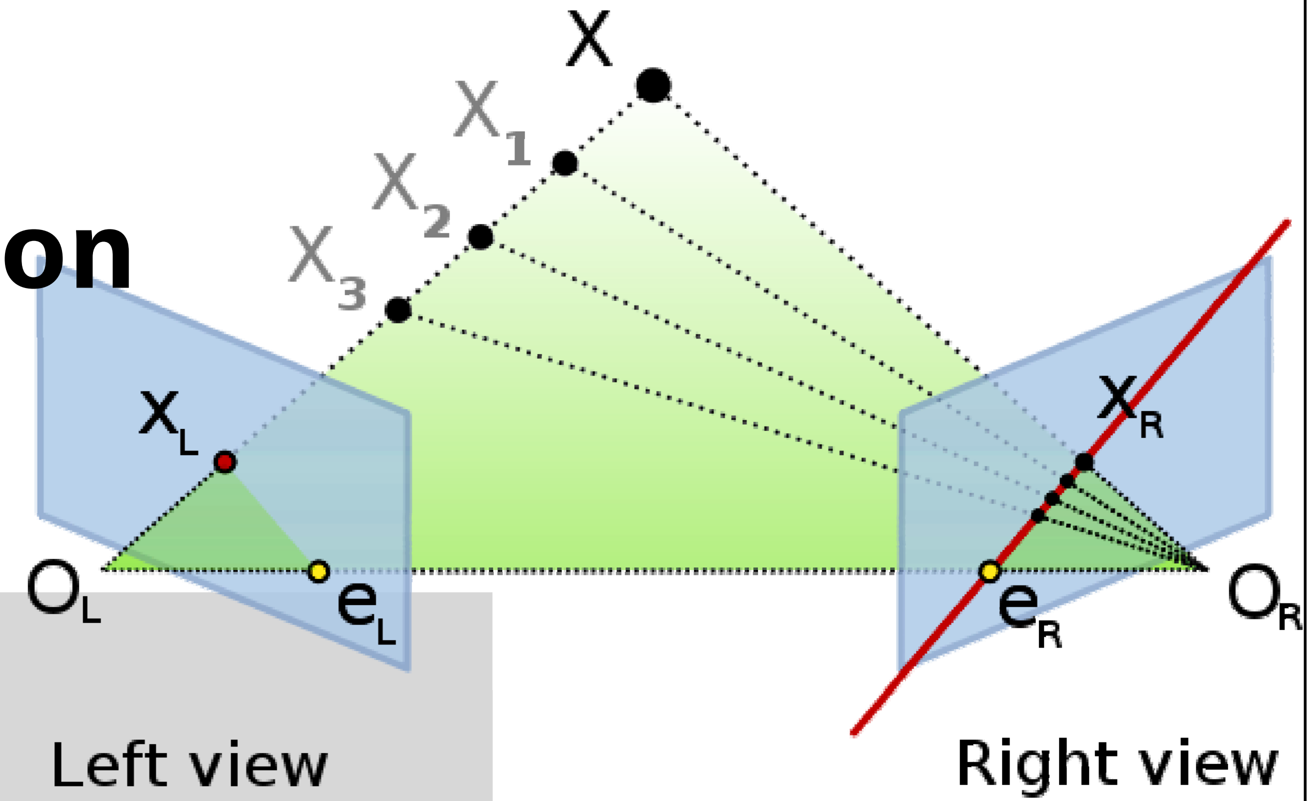
[a]. Oxford Keble College

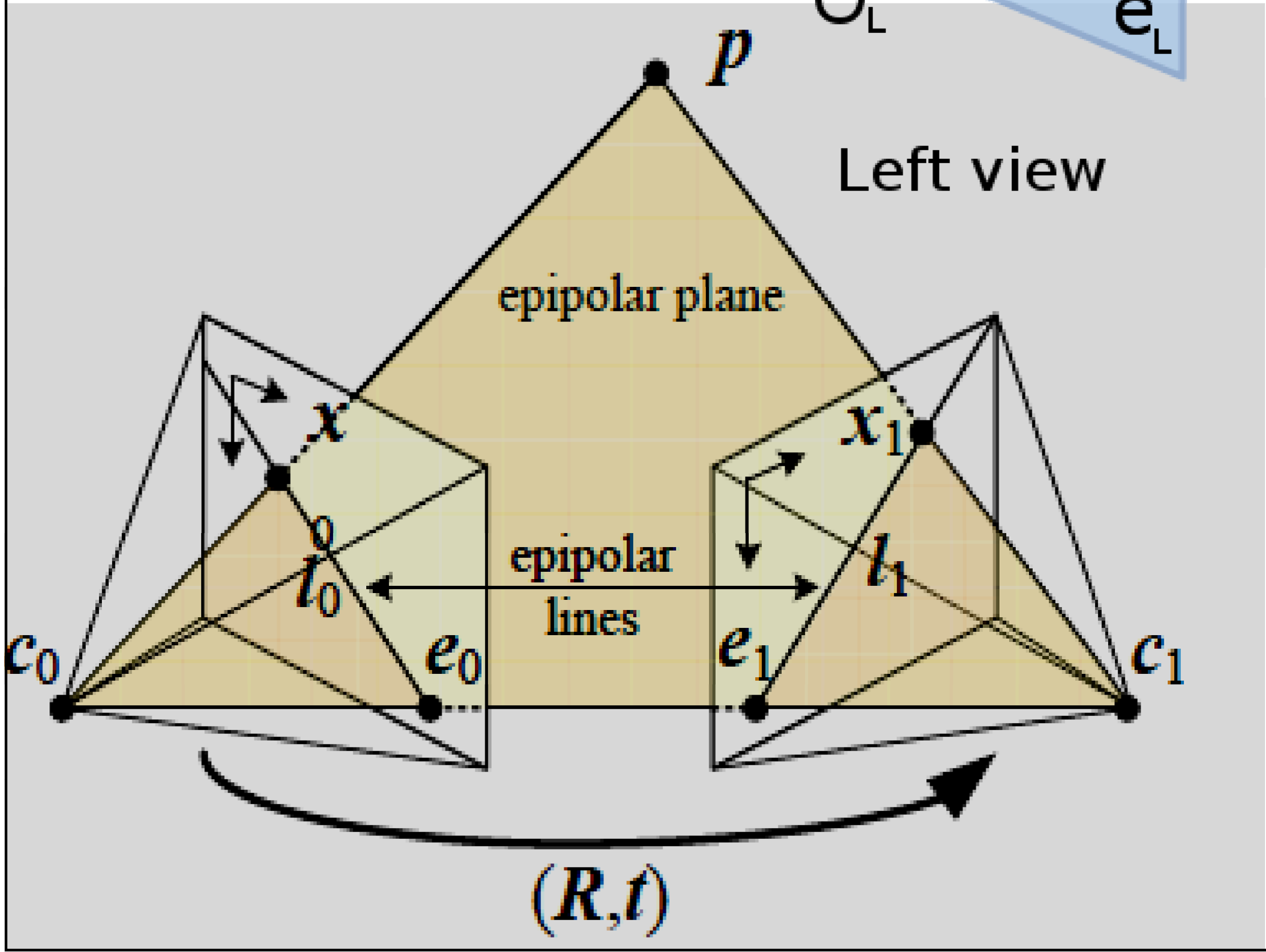
Reconstruction from turntable sequence

- The images acquired from various poses using an ordinary camera can be used to gene
- How st recons
- Is there where better t

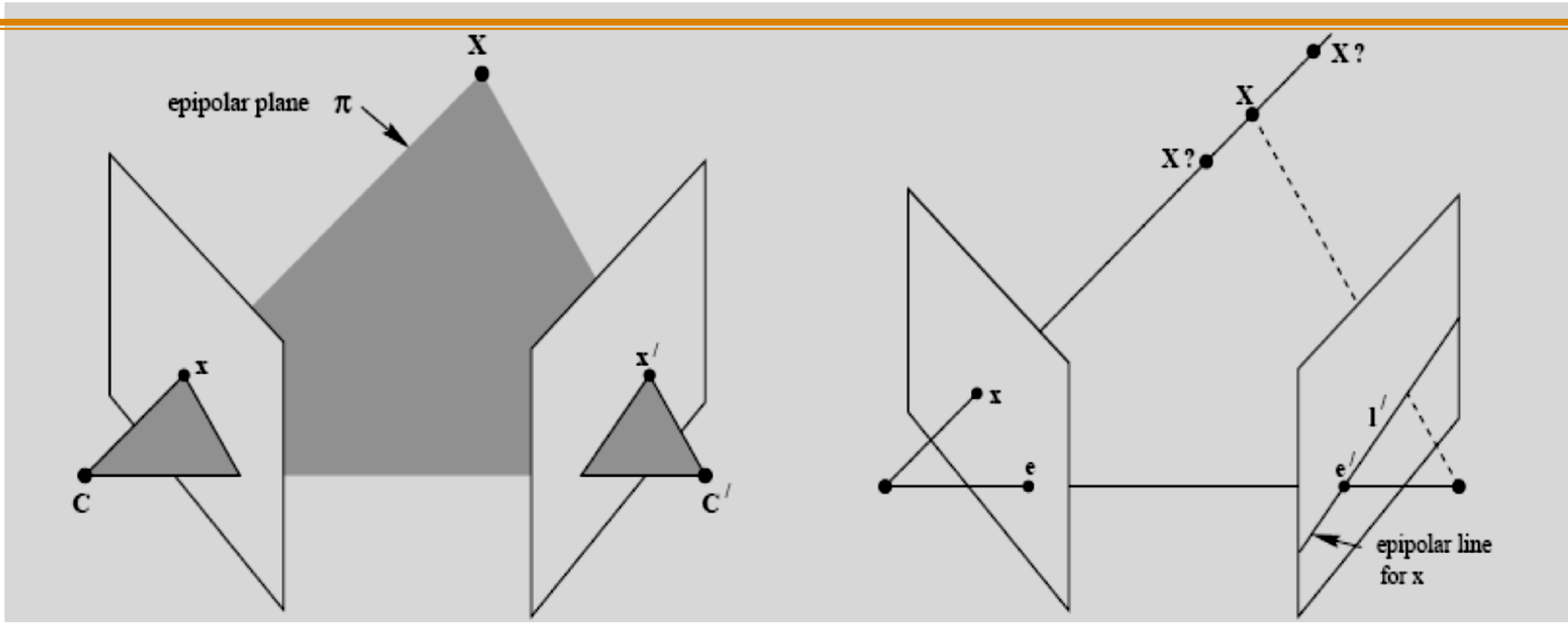


3D Reconstruction





Epipolar lines and Fundamental matrix



- An **epipolar plane** is a plane containing the camera centers (baseline) and the object point.
- An **epipolar line** is the intersection of an epipolar plane with the image plane.
- **Fundamental Matrix (F)** gives the constraint between corresponding image points of same 3D object point [a]

[a] A. Zisserman, Multiple View Geometry '02

Some Notations (*different; WATCH very carefully*)

$Point: \vec{x} = (x, y)^T;$

$Line: \vec{L} = (a, b, c)^T;$

A point x in line L is: $x^T L = L^T x = \vec{x} . \vec{L} = \vec{L} . \vec{x} = 0;$

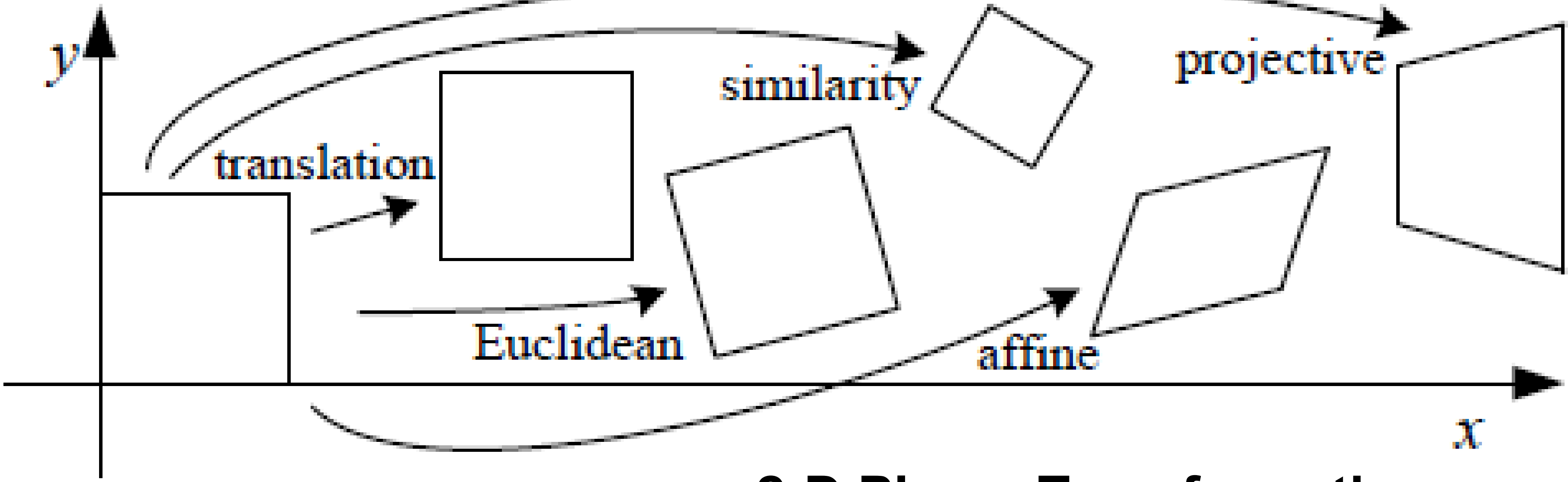
A line through two points is: $L = \vec{x} \mathbf{x} \vec{x}';$

Point as intersection of 2 lines: $\vec{x} = \vec{L} \mathbf{x} \vec{L}';$

$$\vec{A} \mathbf{x} \vec{B} = (a_2 b_3 - a_3 b_2, a_3 b_1 - a_1 b_3, a_1 b_2 - a_2 b_1)^T$$

Define: $$[A]_\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix};$$

$$Thus, \quad [A]_\times B = \vec{A} \mathbf{x} \vec{B} = \left(A^T [B]_\times\right)^T$$



2-D Planar Transformations

New:
Projective
Or
Homography

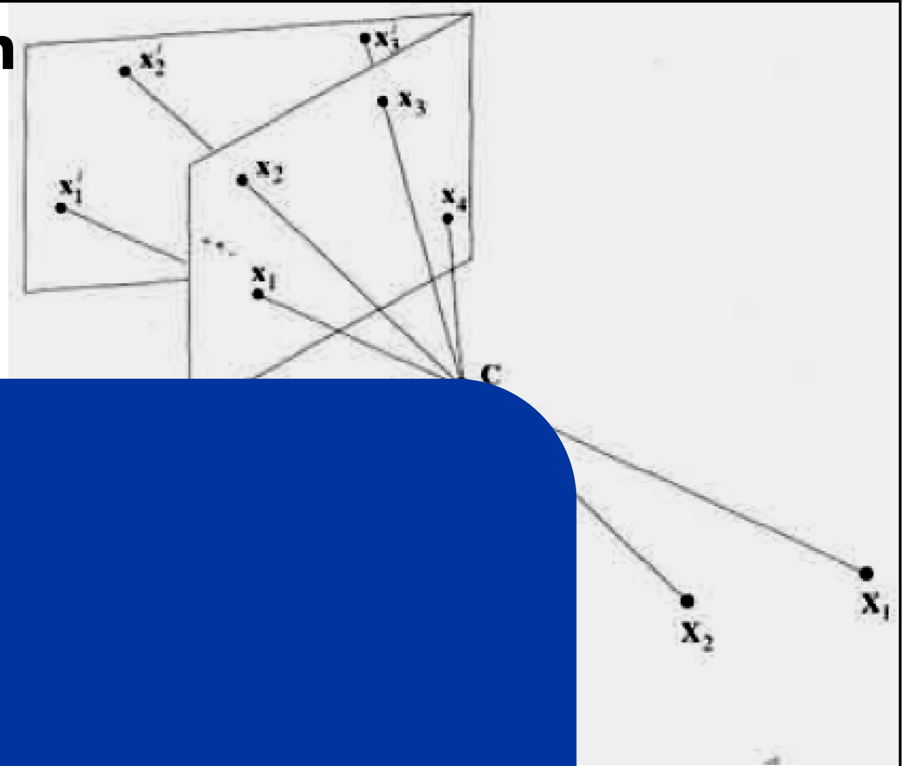
$$\mathrm{x}' = H\,\mathrm{x};$$

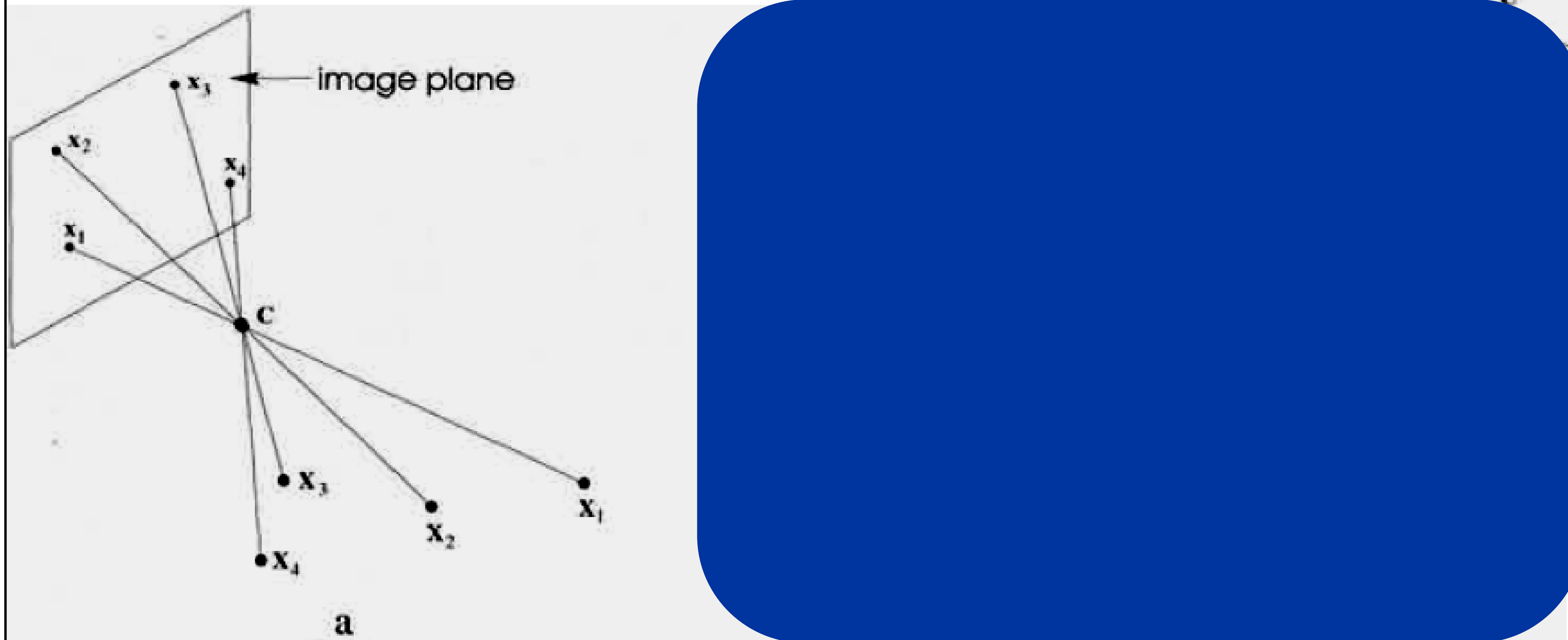
$$x' = \frac{h_{00}x + h_{01}y + h_{02}}{h_{20}x + h_{21}y + h_{22}};\ \text{and}\ y' = \frac{h_{10}x + h_{11}y + h_{12}}{h_{20}x + h_{21}y + h_{22}}$$

$$l.\mathrm{x} = 0; \qquad l'.\mathrm{x}' = \qquad = 0;$$

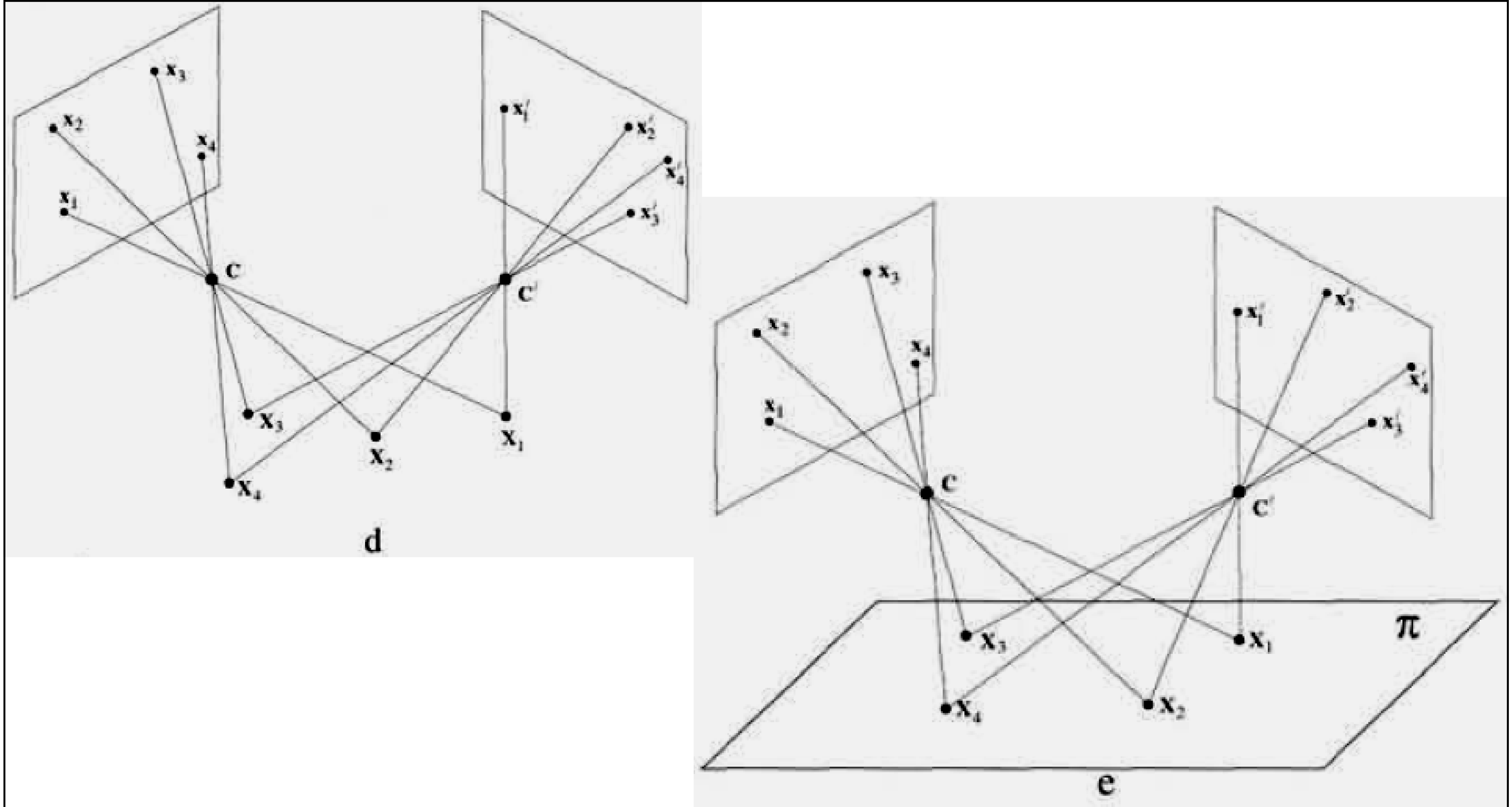
$$\text{Thus, } l' = $$

A projectivity (or homography) is an invertible mapping H from $\mathbb{P}^2$ to itself such that three points x1, x2 and x3 lie on the same line, iff H(x1), H(x2) and H(x3) do.



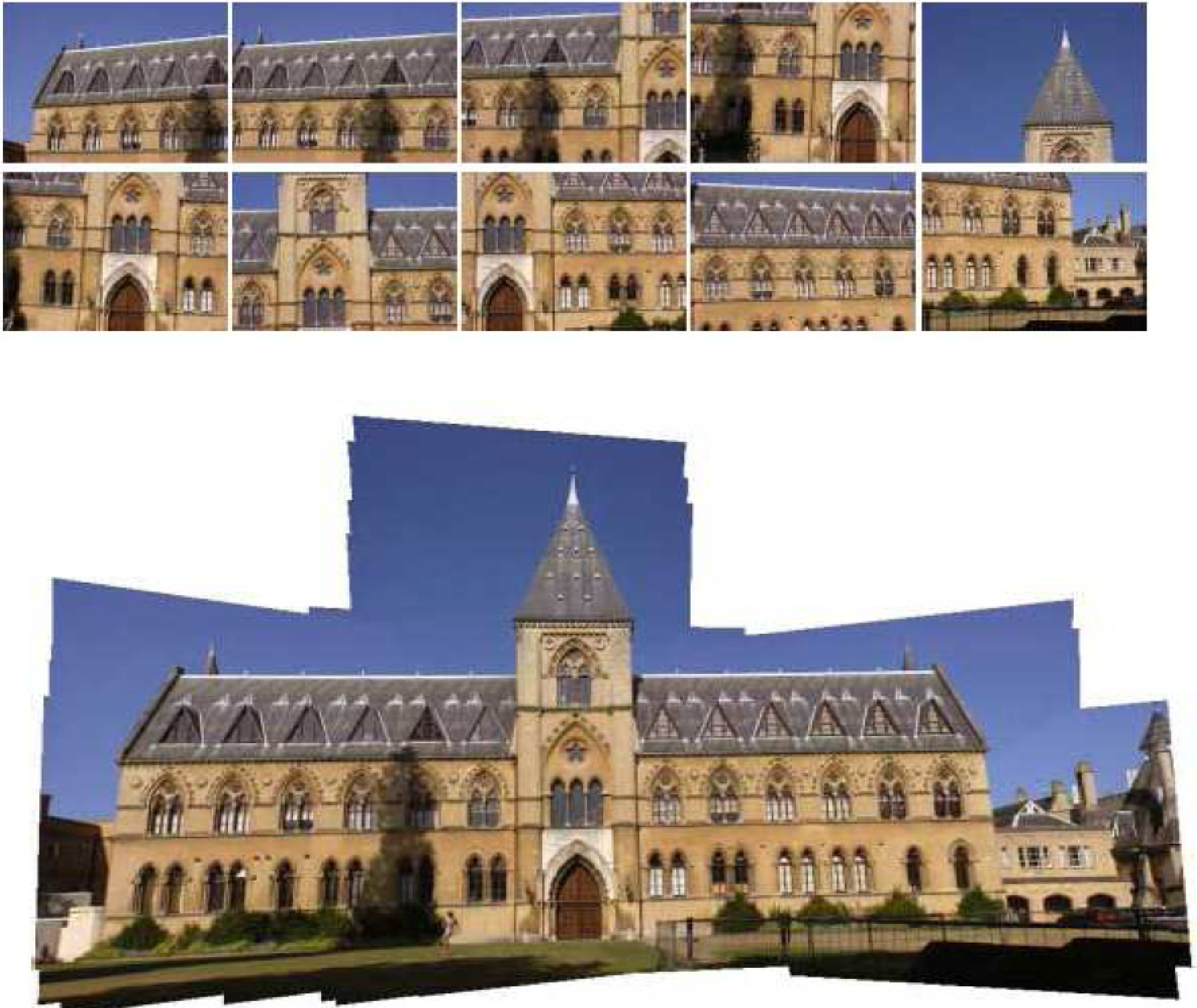


The camera centre is the essence, (a) Image formation: the image points x_i are the intersection of a plane with rays from the space points X_j through the camera centre C. (b) If the space points are coplanar then there is a projective transformation between the world and image planes: $x_i = H_{3X3}X_i$. (c) All images with the same camera centre are related by a projective transformation, $x'_i = H'_{3x3}x_i$. Compare (b) and (c) - in both cases planes are mapped to one another by rays through a centre. In (b) the mapping is between a scene and image plane, in (c) between two image planes.

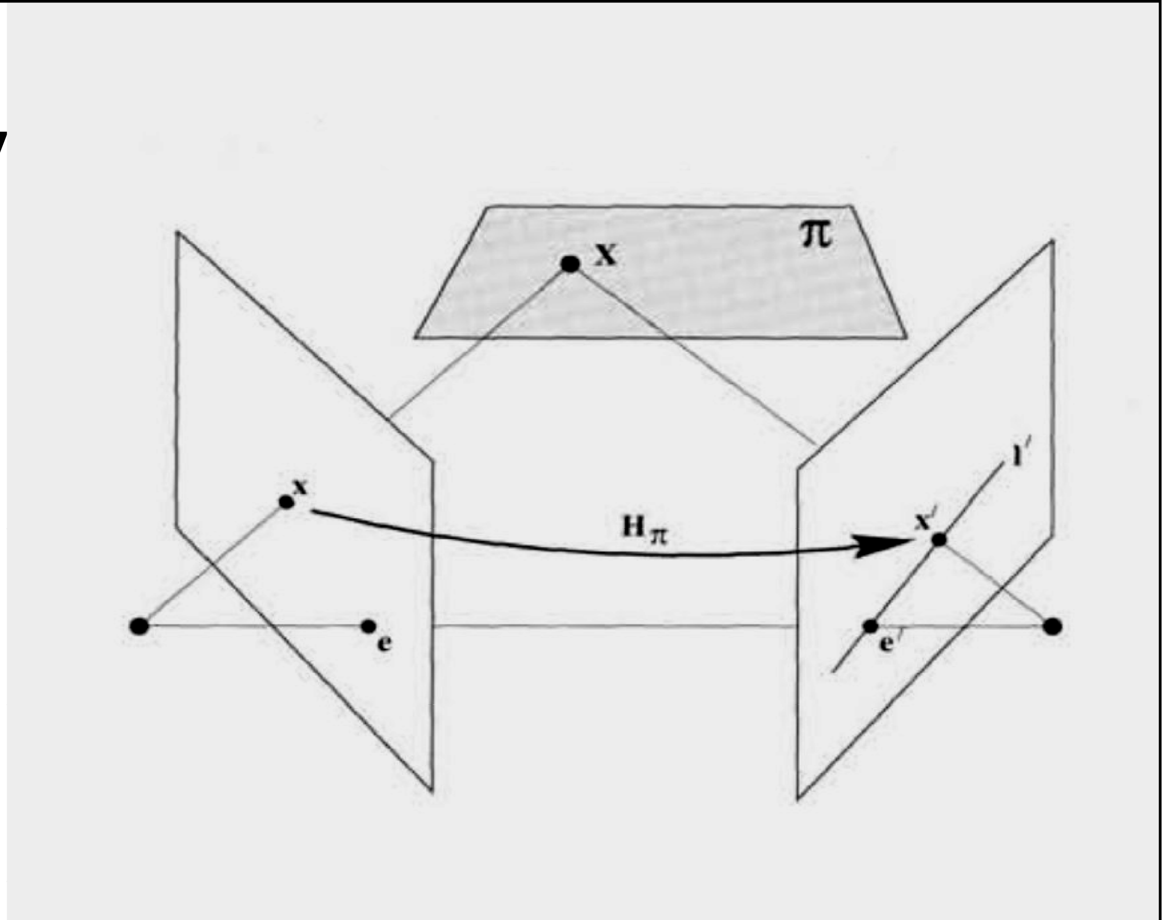


(d) If the camera centre moves, then the images are in general not related by a projective transformation, unless - (e) all the space points are coplanar.

H is non-singular, with 8 dof. It has applications in image/video mosaic, stereo reconstruction, camera calibration, scene modeling and understanding etc.



Homography of points



$$x' = H\ x;$$

$$\begin{aligned} l' &= e' \mathbf{x}\ x' \\ &= [e']_\times\ x' \\ &= [e']_\times H\ x \\ &= F\ x \end{aligned}$$

$$e = [e_1\ e_2\ e_3]$$

$$[e]_\times = \begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix}$$

$$x'^T . l' = 0;$$

F & *H* in terms of camera matrix.

$$\mathrm{x} = PX$$

$$X = P^{+}\,\mathrm{x}$$

$$\mathrm{x}' = P'X$$

$$= P'P^{+}\,\mathrm{x}$$

$$\therefore H = \text{?}$$

$$e' = P'C$$

$$l' = F\,\mathrm{x}$$

$$and,$$

$$F = [e']_{\times} H$$

$$l' = e' \times \mathrm{x}'$$

$$= [e']_{\times}\,\mathrm{x}'$$

$$= [P'C]_{\times}(P'P^{+}\,\mathrm{x})$$

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

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

This is, corresponding Epipolar Line for a point

H in terms of K

$$P = K[I \mid 0]$$

$$P' = KR[I \mid 0]$$

$$\mathrm{x} = PX$$
$$= K[I \mid 0]X$$
$$K^{-1}\mathrm{x} = [I \mid 0]X$$

$$\mathrm{x}' = P'X$$
$$= \text{?}$$
$$=$$

$$x' = Hx$$

$$\therefore H = \text{?}$$

Scene Homography (points)

A **homography** is an invertible mapping of points and lines on a projective plane. Its an invertible mapping to itself, such that collinearity is preserved. It is represented as:

$$\vec{p^h} = H\vec{q^h} \qquad \text{..................(1)}$$

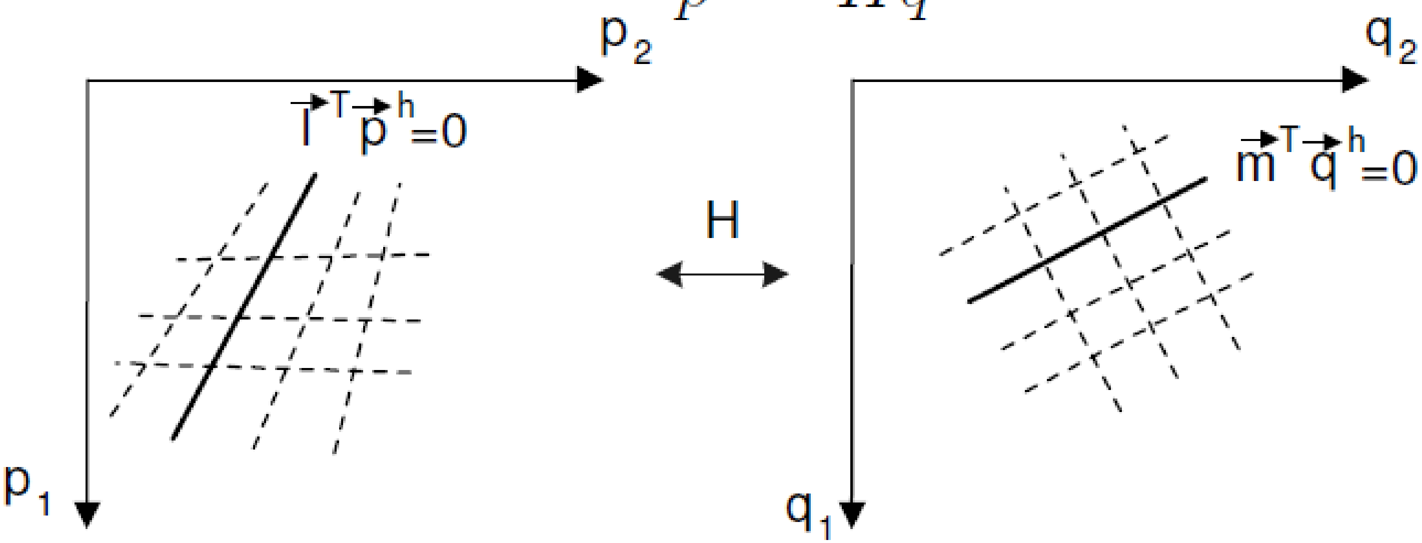
where:

- $\vec{p^h}$, $\vec{q^h}$ are homogeneous 3D vectors
- $H \in \Re^{3X3}$ is called a **homography matrix** and has 8 degrees of freedom, because it is defined up to a scaling factor ($H = cA^{-1}B$ where c is any arbitrary scalar)

- The mapping defined by (1) is called a 2D **homography**
- Since the homography matrix H has 8 degrees of freedom, 4 corresponding $(\vec{p}, \vec{q})$ pairs are enough to constrain the problem

Scene Homography (Lines)

$$\vec{p^h} = H\vec{q^h}$$



$$\vec{l^T}\vec{p^h} = 0 \quad \Leftarrow \quad \vec{m^T}\vec{q^h} = 0$$

From above, derive, *l* = f(H, m) ??

$$l^T p^h = 0 \Rightarrow l^T H q^h = 0 = m^T q^h;$$

$$l^T H = m^T$$

$$\Rightarrow l^T = m^T H^{-1}$$

Thus, $l = \left(H^{-1}\right)^T m$

What about H, from above ??

$$H = (l^T)^{-1} m^T$$

Possible to compute H, now ??

Solving Homography using point correspondences

$$c \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = H \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \qquad (2.1)$$

where c is any non-zero constant, $\begin{pmatrix} u & v & 1 \end{pmatrix}^T$ represents $\mathbf{x}'$, $\begin{pmatrix} x & y & 1 \end{pmatrix}^T$ represents $\mathbf{x}$, and $H = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix}$.

$$-h_1 x - h_2 y - h_3 + (h_7 x + h_8 y + h_9) u = 0 \qquad (2.2)$$

$$-h_4 x - h_5 y - h_6 + (h_7 x + h_8 y + h_9) u = 0 \qquad (2.3)$$

$$A_i \mathbf{h} = \mathbf{0} \qquad (2.4)$$

where $A_i =$

and $\mathbf{h} = \begin{pmatrix} h_1 & h_2 & h_3 & h_4 & h_5 & h_6 & h_7 & h_8 & h_9 \end{pmatrix}^T$.

Solution to a homogeneous system ?

The solution set to a homogeneous system is the same as the null space of the corresponding matrix A.

Singular Value Decomposition (SVD)

Singular value decomposition takes a matrix (defined as A, where A is a n x p matrix). The SVD theorem states:

$$A_{n\times p} = U_{n\times n}\, S_{n\times p}\, V^T{}_{p\times p}$$

where, $U^T U = I \;\&\; V^T V = I$

Calculating the SVD consists of :

- **Finding the eigenvalues and eigenvectors of AA^T and A^TA.**
- **The columns of V are orthonormal eigenvectors of A^TA**
- **The columns of U are orthonormal eigenvectors of AA^T**
- **Also, the singular values in S are square roots of eigenvalues from AA^T or A^TA in descending order.**

Some important observations:

$$M = U\, \Sigma\, V^*$$

- **The singular values are the diagonal entries of the S matrix and are arranged in descending order.**
- **The singular values are always real numbers.**
- **If the matrix M is a real matrix, then U and V are also real.**

The right-singular vectors corresponding to vanishing singular values of M span the null space of M. The left-singular vectors corresponding to the non-zero singular values of M span the range (space) of M.

$$A_i \mathbf{h} = \mathbf{0}$$

Since each point correspondence provides 2 equations, 4 correspondences are sufficient to solve for the 8 degrees of freedom of H. The restriction is that no 3 points can be collinear (i.e., they must all be in "general position"). Four 2×9 A_i matrices (one per point correspondence) can be stacked on top of one another to get a single 8×9 matrix A. The 1D null space of A is the solution space for $\mathbf{h}$.

If the homography is exactly determined, then $\sigma_9 = 0$, and there exists a homography that fits the points exactly.

This is the basic DLT algorithm, which only requires normalization (pixel coordinates) and de-normalization steps, prior and after the solution of the homogeneous system.

Also a cost minimization approach (use RANSAC) is used for a over-determined set of systems, for a robust solution.

For Homography using line correspondences:

$$A_i = \begin{pmatrix} -u & 0 & ux & -v & 0 & vx & -1 & 0 & x \\ 0 & -u & uy & 0 & -v & vy & 0 & -1 & y \end{pmatrix}$$

$\begin{pmatrix} u & v & 1 \end{pmatrix}^T$ represents $\mathbf{l}'$ and $\begin{pmatrix} x & y & 1 \end{pmatrix}^T$ represents $\mathbf{l}$

Estimate H (DLT, but with an alternate notation)

Given n>=4 2-D point pairs;

Algo:

$$\mathrm{x}'_i \times H\,\mathrm{x}_i = 0; \quad \mathrm{x}'_i = (x'_i, y'_i, w'_i)^T;$$

$$\Rightarrow \begin{bmatrix} 0^T & -w'_i\,\mathrm{x}_i^T & y'_i\,\mathrm{x}_i^T \\ w'_i\,\mathrm{x}_i^T & 0^T & -\mathrm{x}'_i\mathrm{x}_i^T \\ -y'_i\,\mathrm{x}_i^T & \mathrm{x}'_i\mathrm{x}_i^T & 0^T \end{bmatrix} \begin{pmatrix} h^1 \\ h^2 \\ h^3 \end{pmatrix} = 0 \Rightarrow A_i\mathrm{h} = 0$$

Use:

-

$$\begin{bmatrix} \mathbf{0}^{\mathbf{T}} & -w'_i\mathbf{x}_i^{\mathbf{T}} & y'_i\mathbf{x}_i^{\mathbf{T}} \\ w'_i\mathbf{x}_i^{\mathbf{T}} & \mathbf{0}^{\mathbf{T}} & -x'_i\mathbf{x}_i^{\mathbf{T}} \end{bmatrix} \begin{bmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{bmatrix} = \mathbf{0}.$$

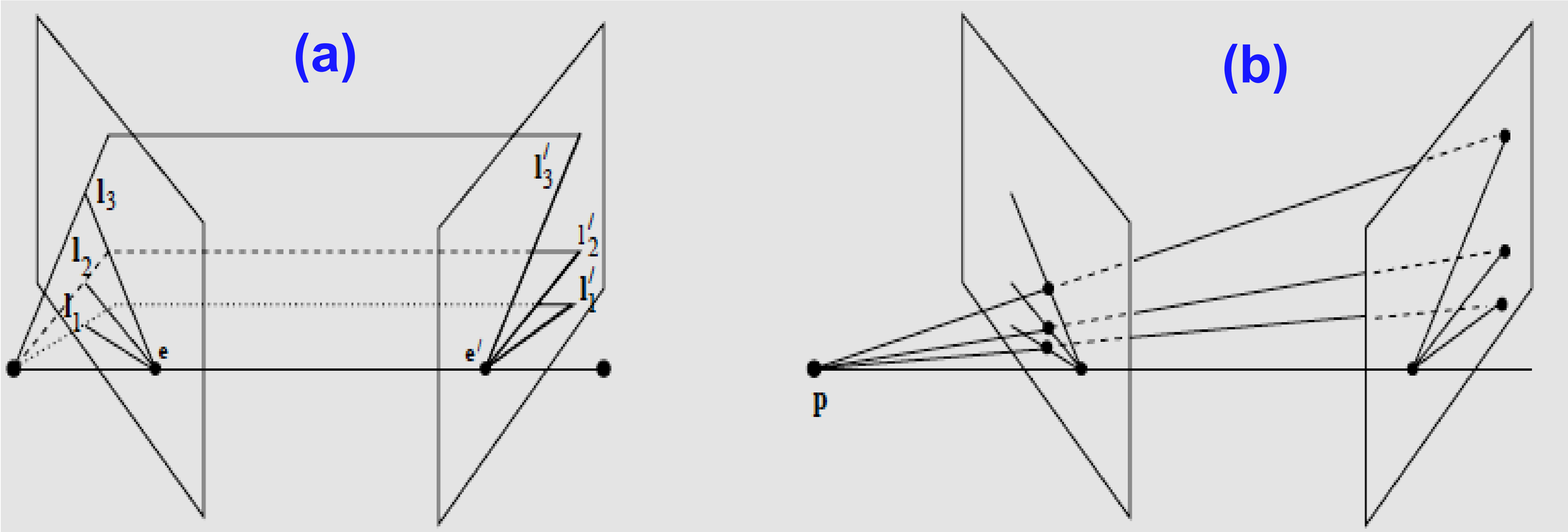
- Assemble n such 2*9 matrices A_i into a single 2n*9 matrix A, by stacking horizontally row-wise;

- SVD of A, gives : $A = UDV^T;$

- $\mathbf{h}_{9*1}$ is the last column of V (unit singular eigen-vector corresponding to smallest singular value)

- Form H_{3*3}, by arranging elements of $\mathbf{h}$

- May need normalization of coordinates



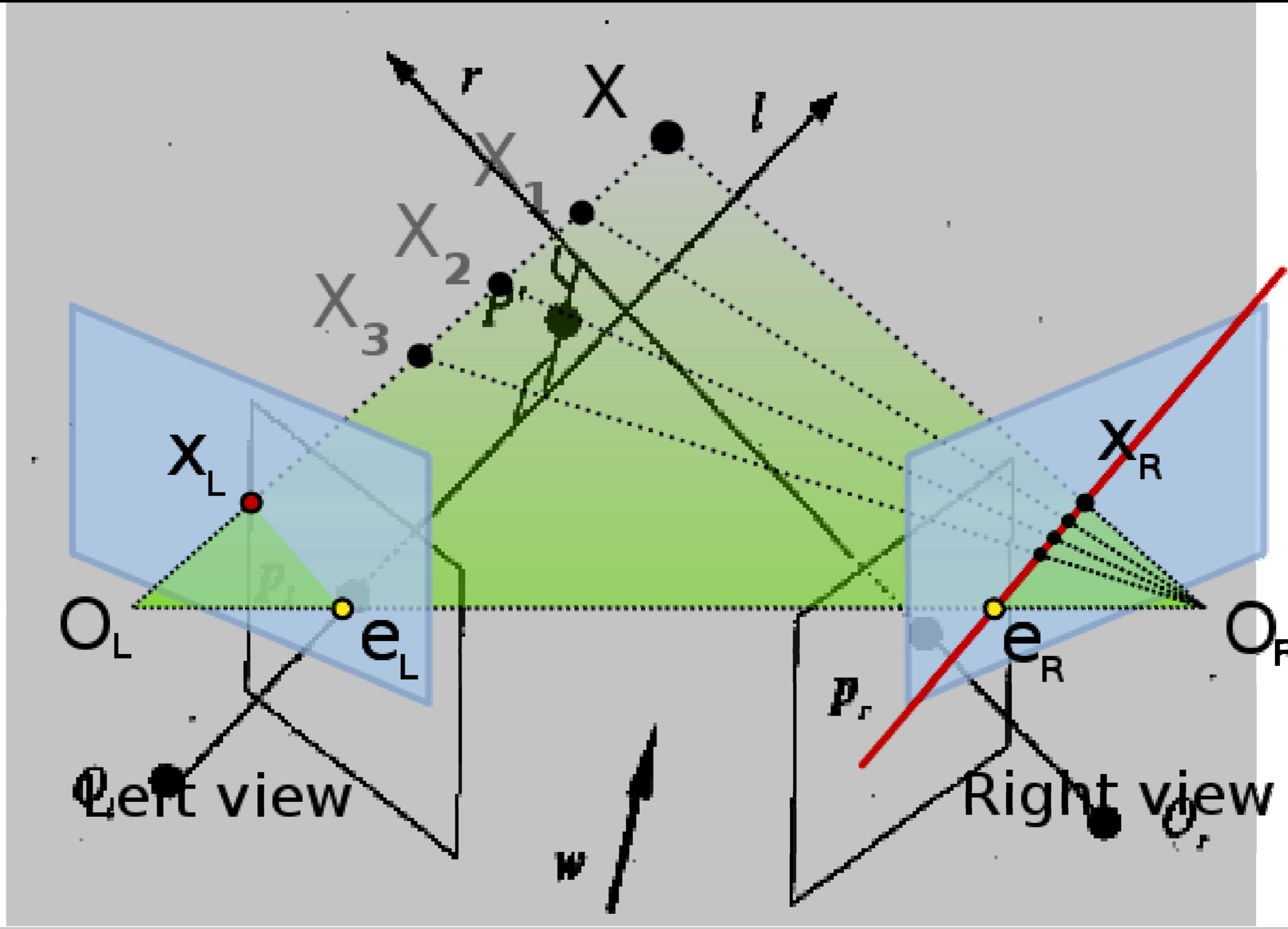
Epipolar line homography:

(a) There is a pencil of epipolar lines in each image centred on the epipole. The correspondence between epipolar lines, $l_i \leftrightarrow l'_i$ is defined by the pencil of planes with axis the baseline.

(b) The corresponding lines are related by a perspectivity, with centre at any point p on the baseline. It follows that the correspondence between epipolar lines in the pencils is a 1D homography.

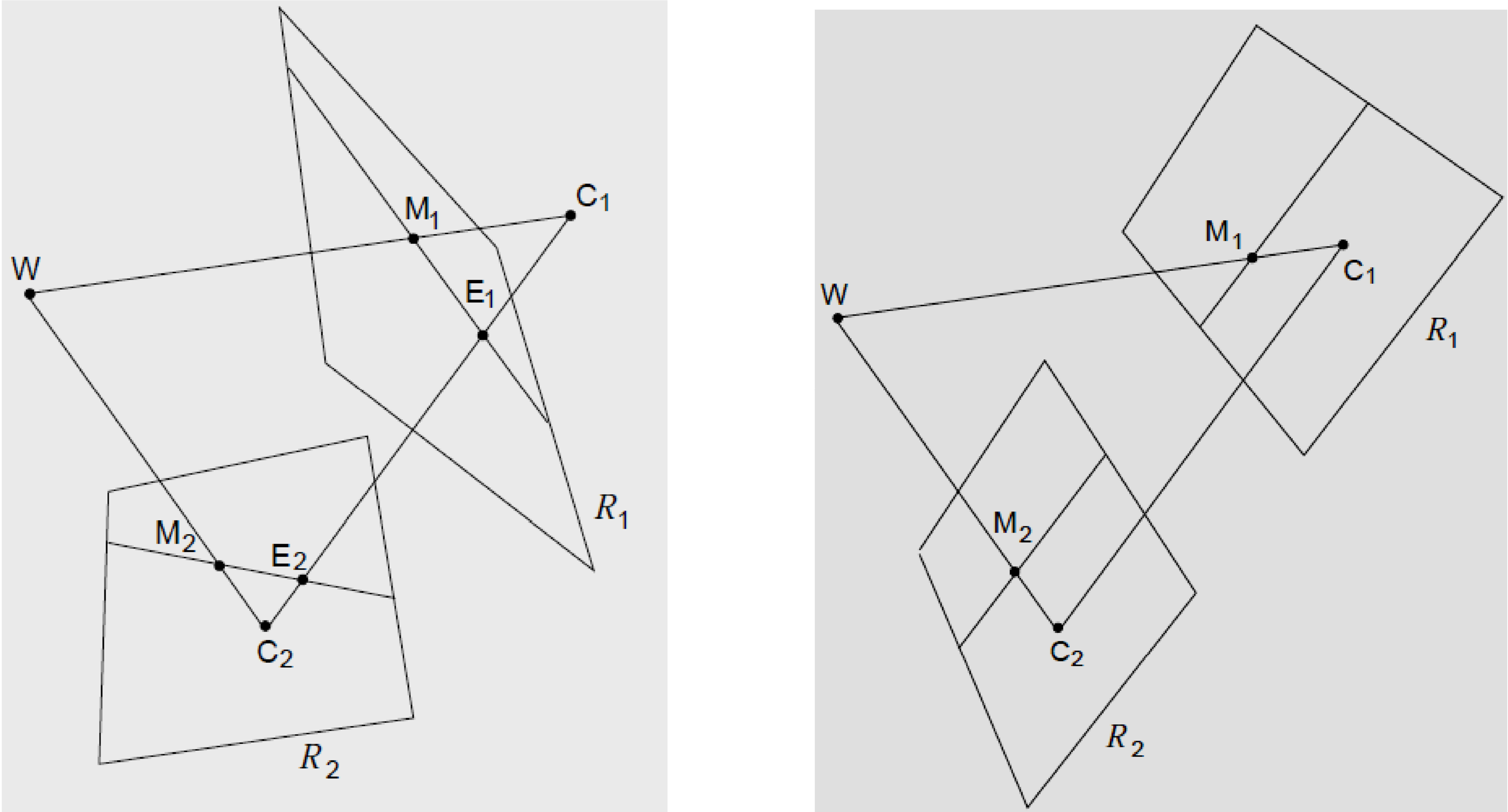
If the stereo is calibrated; i.e P and P' known, use:

A compact algorithm for rectification of stereo pairs; Andrea Fusiello, Emanuele Trucco, Alessandro Verri ; Machine Vision and Applications (2000) 12: 16–22 Machine Vision and Applications; Springer-Verlag 2000;



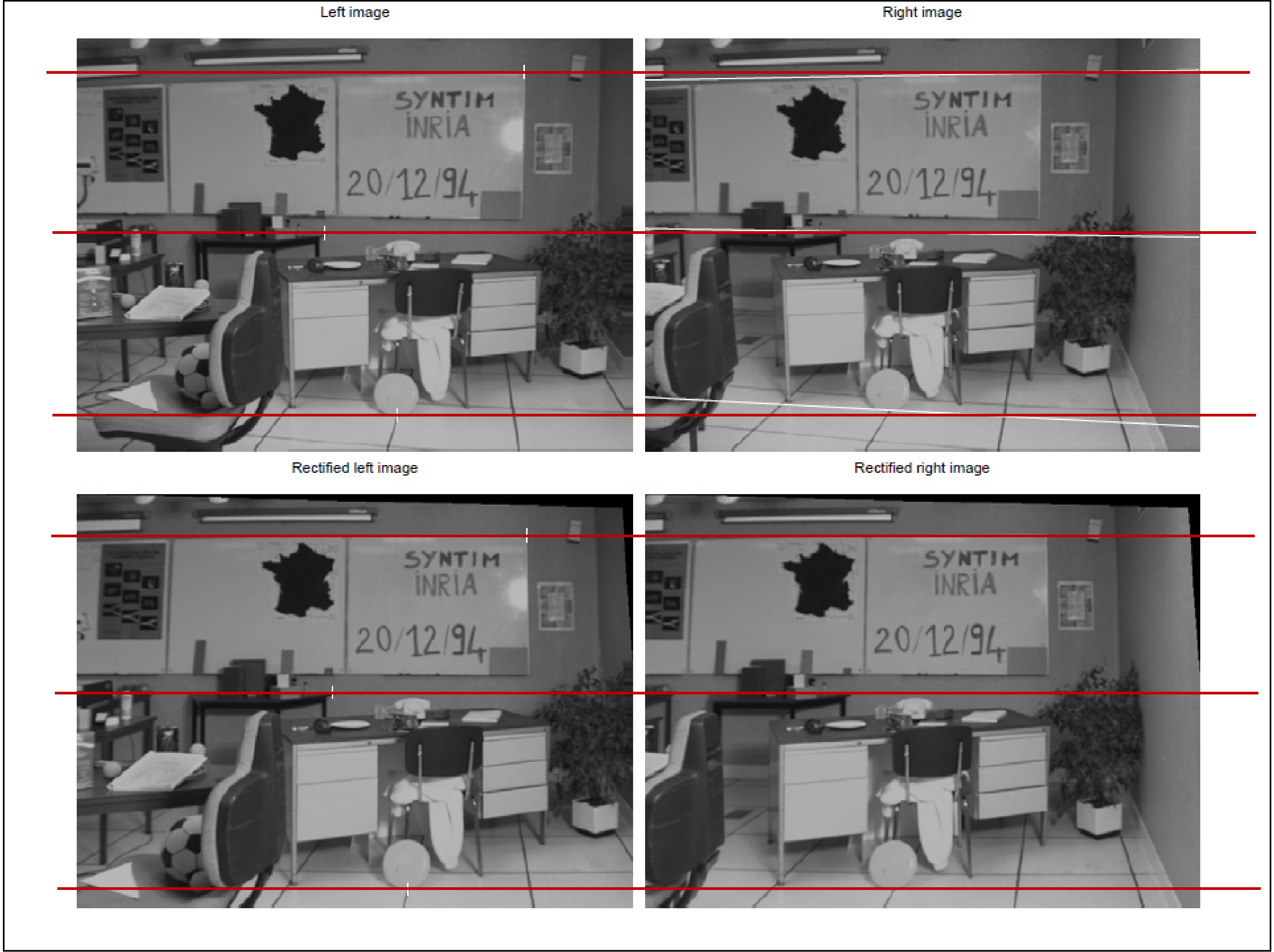
A Priori Knowledge	3-D Reconstruction from Two Views
Intrinsic and extrinsic parameters	Unambiguous (absolute coordinates)
Intrinsic parameters only	Up to an unknown scaling factor
No information on parameters	Up to an unknown projective transformation of the environment

W is orthogonal to both r & ***l; - formula ??***



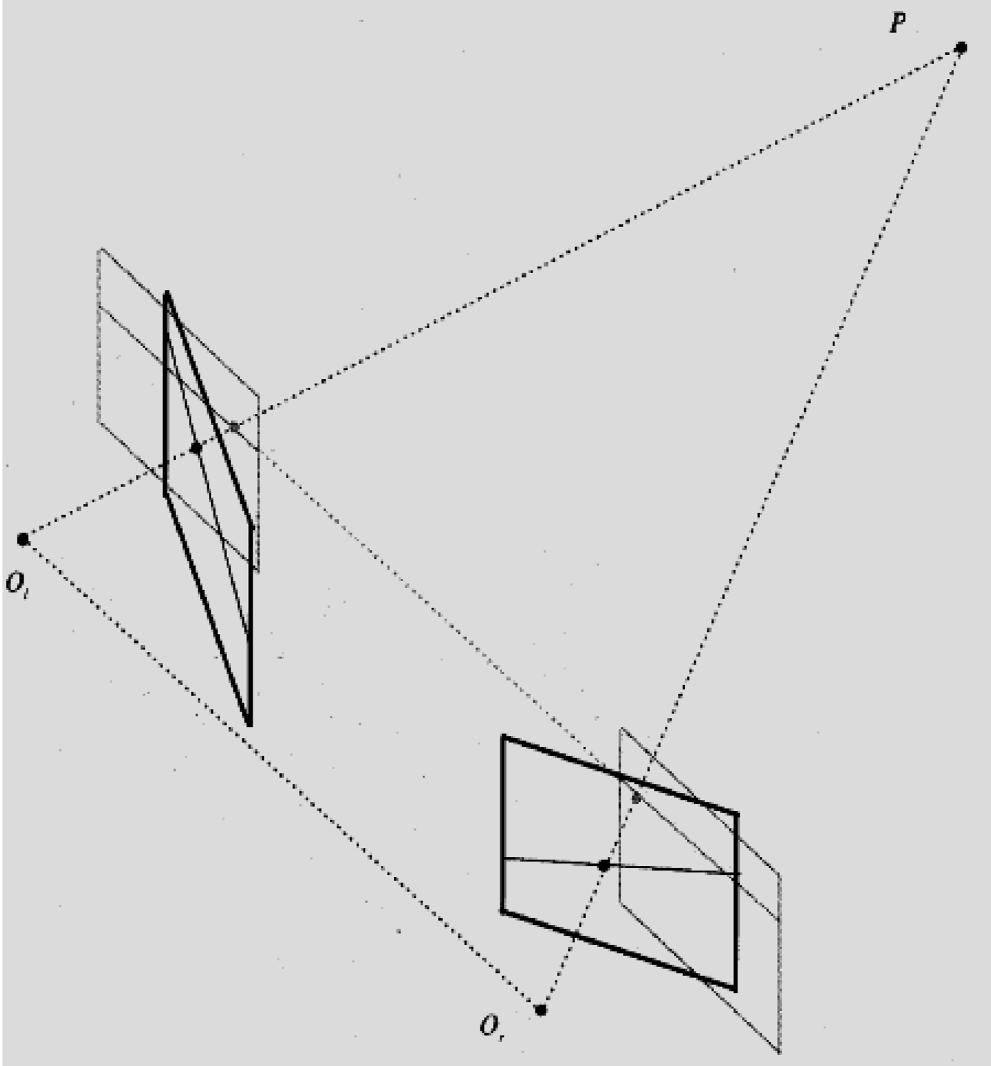
Process of Rectification

Image rectification is the process of applying a pair of 2 dimensional projective transforms, or homographies, to a pair of images whose epipolar geometry is known so that epipolar lines in the original images map to horizontally aligned lines in the transformed images.


Left image
Right image
Rectified left image
Rectified right image
SYNTIM
INRIA
20/12/94

Assumptions and Problem Statement of Rectification:

Given a stereo pair of images, the intrinsic parameters (K) of each camera, and the extrinsic parameters of the system, *R and T*; *compute the image transformation that makes conjugated epipolar* lines collinear and parallel to the horizontal image axis.

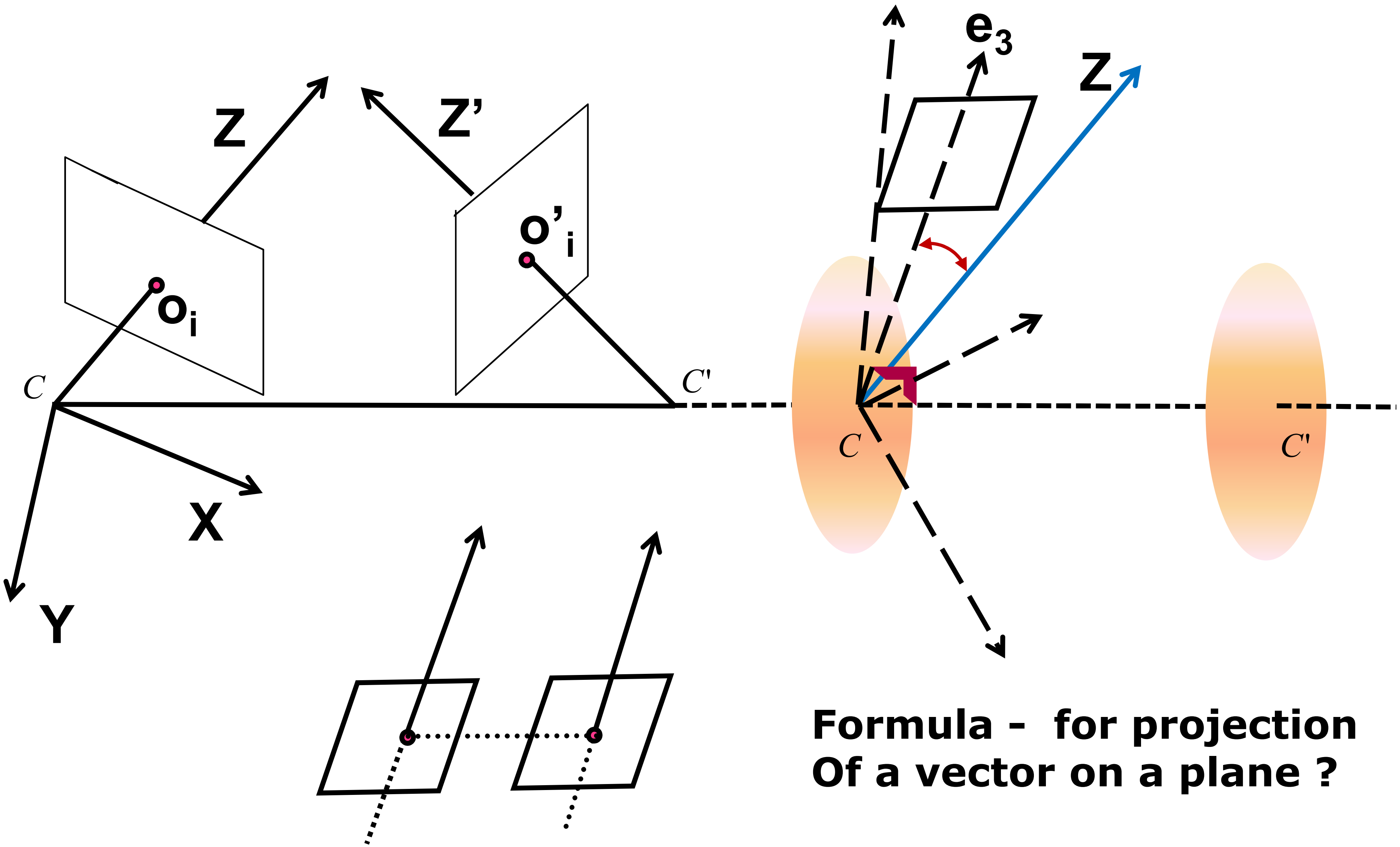


The algorithm (Trucco, Verri) consists of four steps:

- **Rotate the left camera so that the epipole goes to infinity along the horizontal axis.**
- **Apply the same rotation to the right camera to recover the original geometry.**
- **Rotate the right camera by *R.***
- **Adjust the scale in both camera reference frames.**

RECTIFICATION Illustrated

$$\vec{e}_1 = \vec{T};\ \vec{e}_2 =$$



Formula - for projection Of a vector on a plane ?

Rectification algo. (four steps), by Trucco, Verri:

- **Rotate the left camera so that the epipole goes to infinity along the horizontal axis.**
- **Apply the same rotation to the right camera to recover the original geometry.**

First rotate the left camera so that it looks perpendicular to the line joining the camera centers *c0 and c1. Since there is a degree* of freedom in the *tilt, the smallest rotations that achieve this should be used.* smallest rotation can be computed from the cross product between the original and desired optical axes.

To determine the desired twist around the optical axes, make the *up vector (the camera y axis)* perpendicular to the baseline. This ensures that corresponding epipolar lines are horizontal and that the disparity for points at infinity is 0. The cross product between the current *x-axis after the first* rotation and the line joining the cameras gives the rotation.

- **Rotate the right camera by R (or R^{-1}).**
- **Adjust the scale in both camera reference frames.**

If necessary, to account for different focal lengths, magnifying the smaller image to avoid aliasing. Now, both have the same resolution (and hence line-to-line correspondence).

Algorithm RECTIFICATION

The input is formed by the intrinsic and extrinsic parameters of a stereo system and a set of points in each camera to be rectified (which could be the whole images).

Also, in both cameras:

i). the origin of the image reference frame is the principal point;

ii). the focal length is equal to *f*.

Steps:

1. Build the matrix *R_{rect}* as: $R_{rect} = \left(e_1^T \quad e_2^T \quad e_3^T\right)^T$

$$\vec{e}_1 = \vec{T};\ \vec{e}_2 = \vec{Z} \times \vec{T} = (-T_y, T_x, 0)^T;\ \vec{e}_3 = \vec{e}_1 \times \vec{e}_2$$

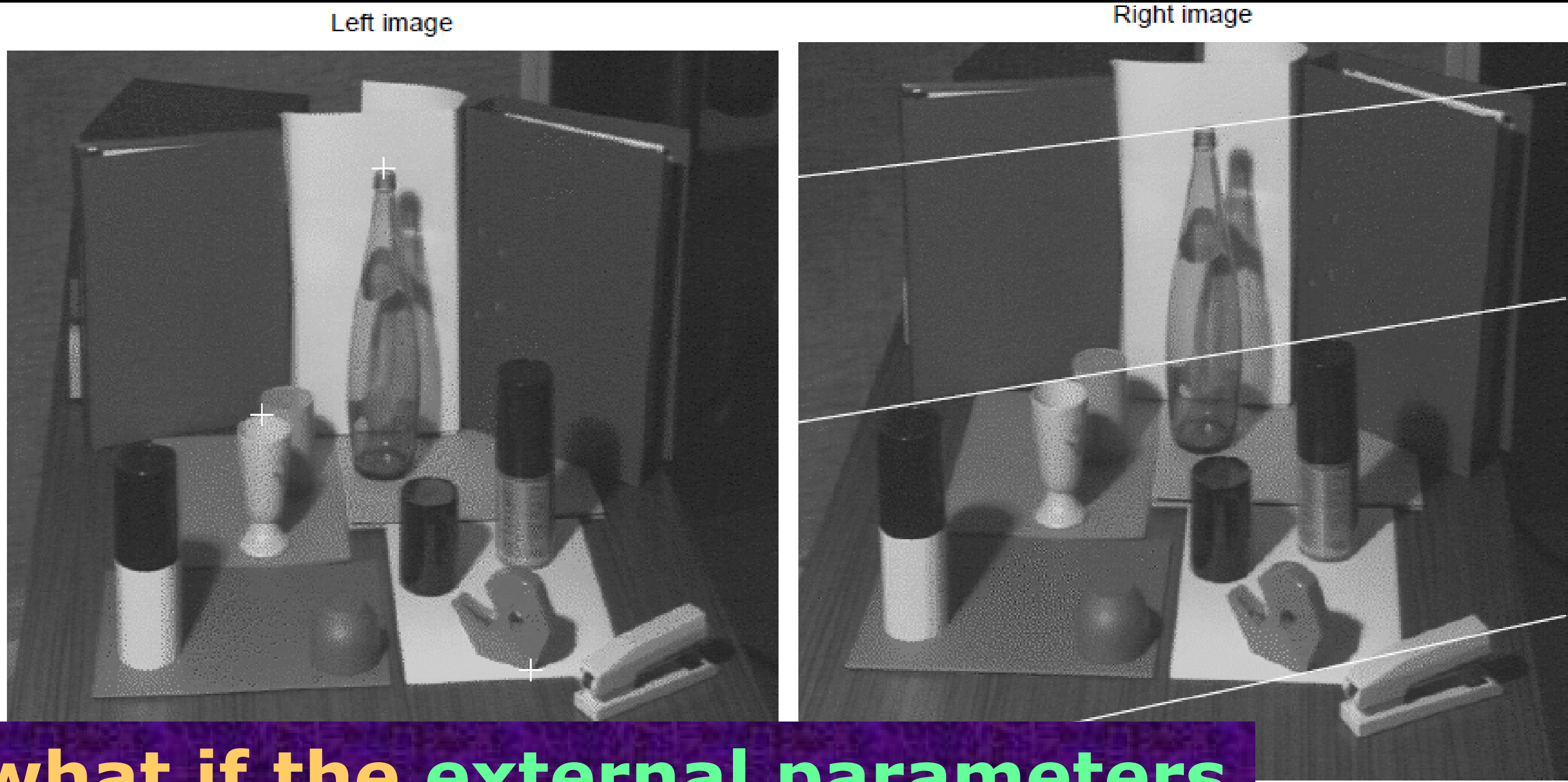
2. Set *$R_l = R_{rect}$ and $R_r = R^{-1}.R_{rect}$;*

3, 4: For Left and Right camera points, do:

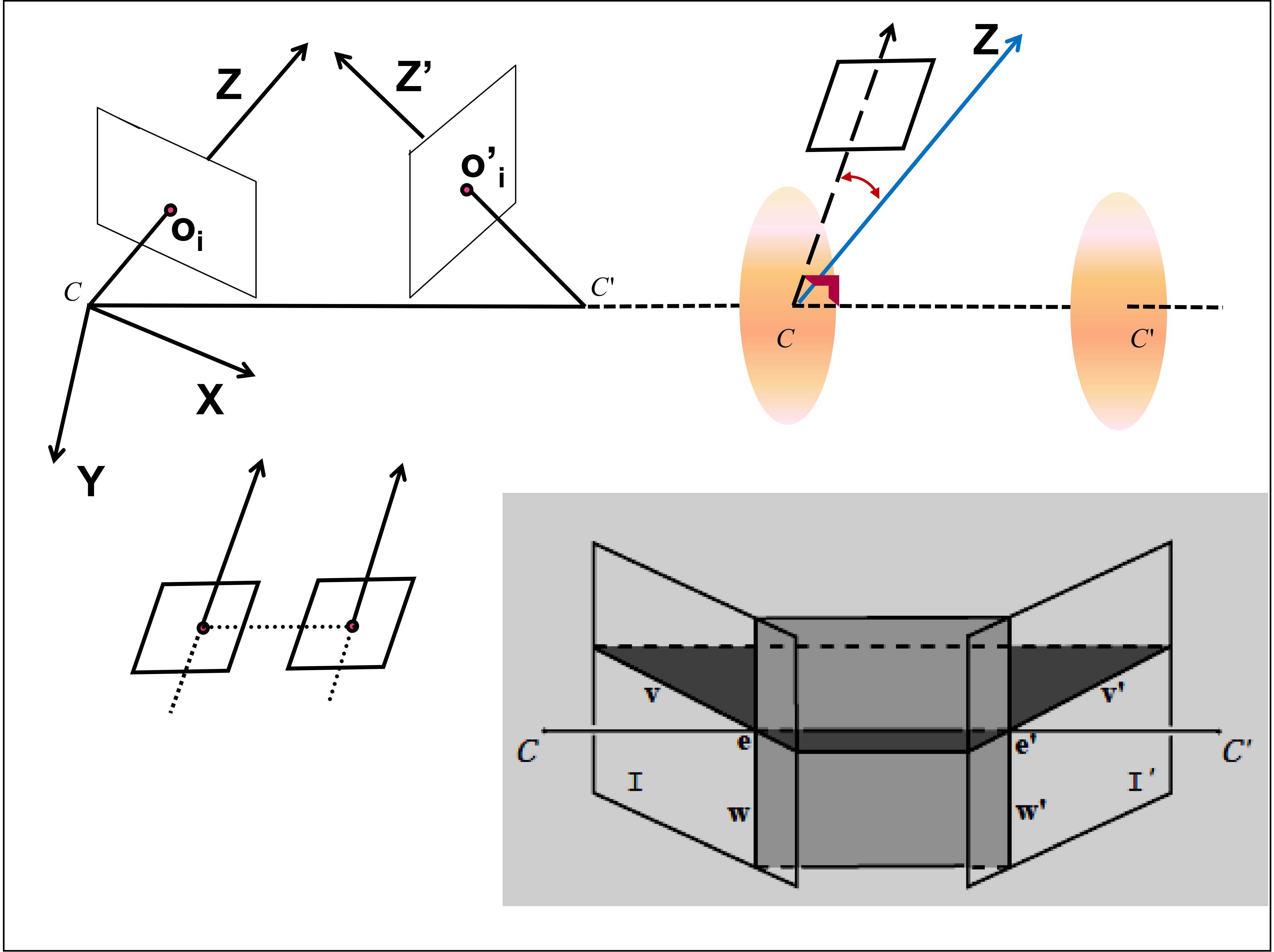
$$[x', y', z'] = R_c\,[x, y, f]^T;$$

$$\mathrm{x}' = \left(f/z\right)[x', y', z'].$$

This algorithm fails when the optical axis is parallel to the baseline, i.e., when there is a pure forward motion.

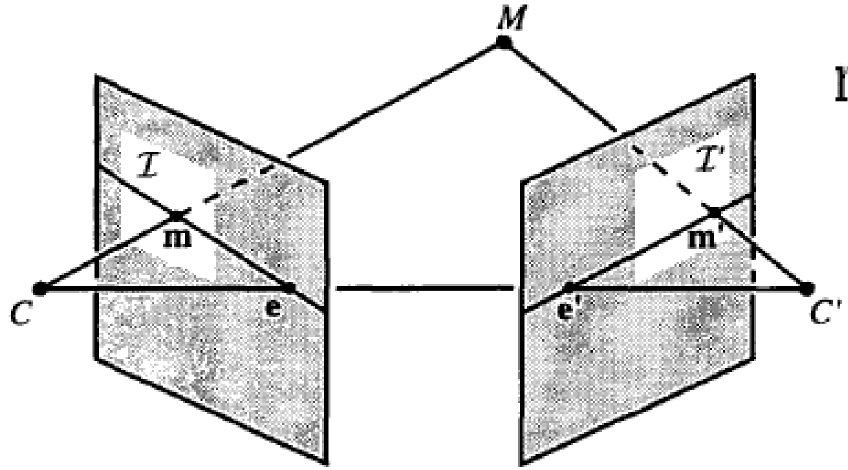


But, what if the external parameters are not known


Z
Z'
O'i
Oi
C
C'
X
Y
Z
C
C'
v
v'
e
e'
C
C'
I
I '
w
w'

Rectification (Zhang's), using Fundamental matrix

Work on entirely 2-D space;
Points and lines: $m = [m_u \quad m_v \quad m_w]^T; \; \mathbf{l} = [l_a \quad l_b \quad l_c]^T$



$$\mathbf{m}'^T \mathbf{F} \mathbf{m} = 0, \qquad (1)$$

F is a 3x3 rank-2 matrix, is known (?).

$$Fm = l'; \quad m'^T l' = 0;$$

$$Fe = 0 = F^T e';$$

Properties of rectified image pair:

- **All epipolar lines are parallel to horizontal (x- or u-axis)**
- **Corresponding points have identical y- or v-coordinates.**

Fundamental matrix for a rectified image pair:

$$\bar{\mathbf{F}} = [\mathbf{i}]_\times = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

What is i ??

where, i = [1 0 0]^T, is X-VP (at Inf.)

Rectification (Zhang's) – maps epipolar lines to image scan lines;

Let $\mathbf{H}$ and $\mathbf{H}'$ be the homographies to be applied to images $\mathcal{I}$ and $\mathcal{I}'$ respectively, and let $\mathbf{m} \in \mathcal{I}$ and $\mathbf{m}' \in \mathcal{I}'$ be a pair of points that satisfy Eq. (1). Consider rectified image points $\bar{\mathbf{m}}$ and $\bar{\mathbf{m}}'$ defined

$$\bar{\mathbf{m}} = \mathbf{H}\mathbf{m} \quad \text{and} \quad \bar{\mathbf{m}}' = \mathbf{H}'\mathbf{m}'.$$

It follows from Eq. (1) that

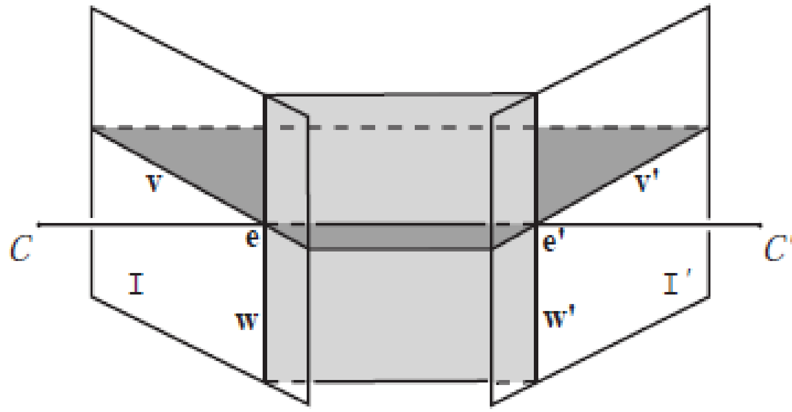
$$\mathbf{m}'^T \mathbf{F} \mathbf{m} = 0,$$

$$\bar{\mathbf{m}}'^T \bar{\mathbf{F}} \bar{\mathbf{m}} = 0,$$

$$\mathbf{m}'^T \underbrace{\mathbf{H}'^T \bar{\mathbf{F}} \mathbf{H}}_{\mathbf{F}} \mathbf{m} = 0,$$

resulting in the factorization

$$\mathbf{F} = \mathbf{H}'^T [\mathbf{i}]_\times \mathbf{H}.$$



$$\mathbf{H}\mathbf{e} = \mathbf{i},\ \mathbf{H}'\mathbf{e}' = \mathbf{i} \text{ and } \mathbf{H}'^T [\mathbf{i}]_\times \mathbf{H} = \mathbf{F}$$

Let,

$$\mathbf{H} = \begin{bmatrix} \mathbf{u}^T \\ \mathbf{v}^T \\ \mathbf{w}^T \end{bmatrix} = \begin{bmatrix} u_a & u_b & u_c \\ v_a & v_b & v_c \\ w_a & w_b & w_c \end{bmatrix}$$

and consider $\mathbf{H}\mathbf{e} = \begin{bmatrix} \mathbf{u}^T\mathbf{e} & \mathbf{v}^T\mathbf{e} & \mathbf{w}^T\mathbf{e} \end{bmatrix}^T = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$

Then, the corresponding lines v and v', w and w' must be epipolar lines (as, l'e=0), for minimal distortion due to rectification;

$\mathbf{H} = \mathbf{H}_{sh}.\mathbf{H}_{rs}.\mathbf{H}_p$

$$\mathbf{H}_s = \begin{bmatrix} s_a & s_b & s_c \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{H}_r = \begin{bmatrix} v_b - v_c w_b & v_c w_a - v_a & 0 \\ v_a - v_c w_a & v_b - v_c w_b & v_c \\ 0 & 0 & 1 \end{bmatrix} \mathbf{H}_p = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ w_a & w_b & 1 \end{bmatrix}$$

$$\mathbf{H}_s = \begin{bmatrix} s_a & s_b & s_c \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{H}_r = \begin{bmatrix} v_b - v_c w_b & v_c w_a - v_a & 0 \\ v_a - v_c w_a & v_b - v_c w_b & v_c \\ 0 & 0 & 1 \end{bmatrix} \mathbf{H}_p = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ w_a & w_b & 1 \end{bmatrix}.$$

Proposition 1. *If* $\mathbf{l} \sim \mathbf{l}'$ *and* $\mathbf{x} \in \mathcal{I}$ *is a direction (point at* ∞*) such that* $\mathbf{l} = [\mathbf{e}]_\times \mathbf{x}$ *then*

$$\mathbf{l}' = \mathbf{F}\mathbf{x}.$$

<- used earlier; Proof in Loop & Zhang '99.

Proposition 2. *If* H *and* H' *are homgraphies such that*

$$\mathbf{F} = \mathrm{H}'^T [\mathbf{i}]_\times \mathrm{H},$$

then $\mathbf{v} \sim \mathbf{v}'$ *and* $\mathbf{w} \sim \mathbf{w}'$.

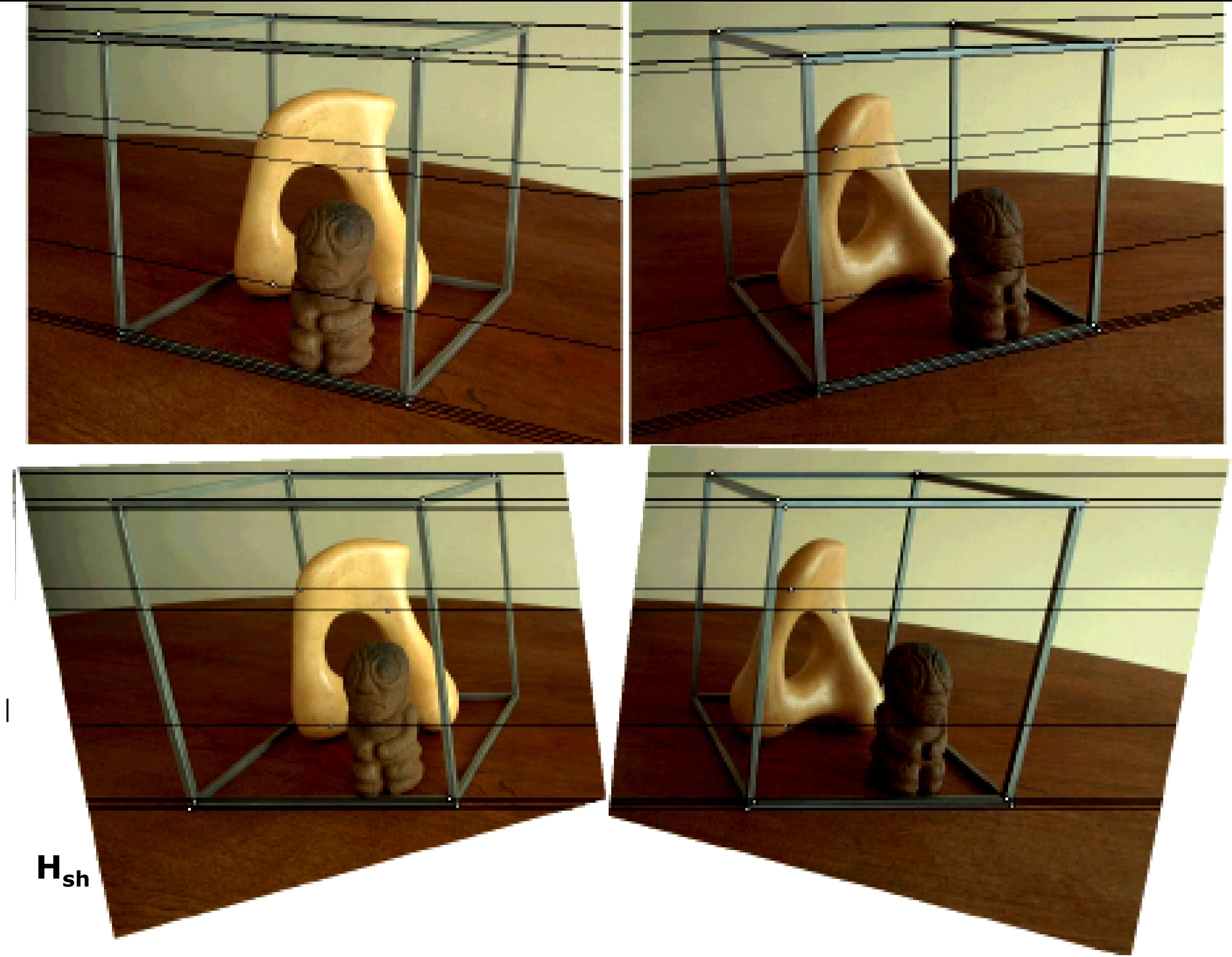
Minimization criteria used to compute H_p.

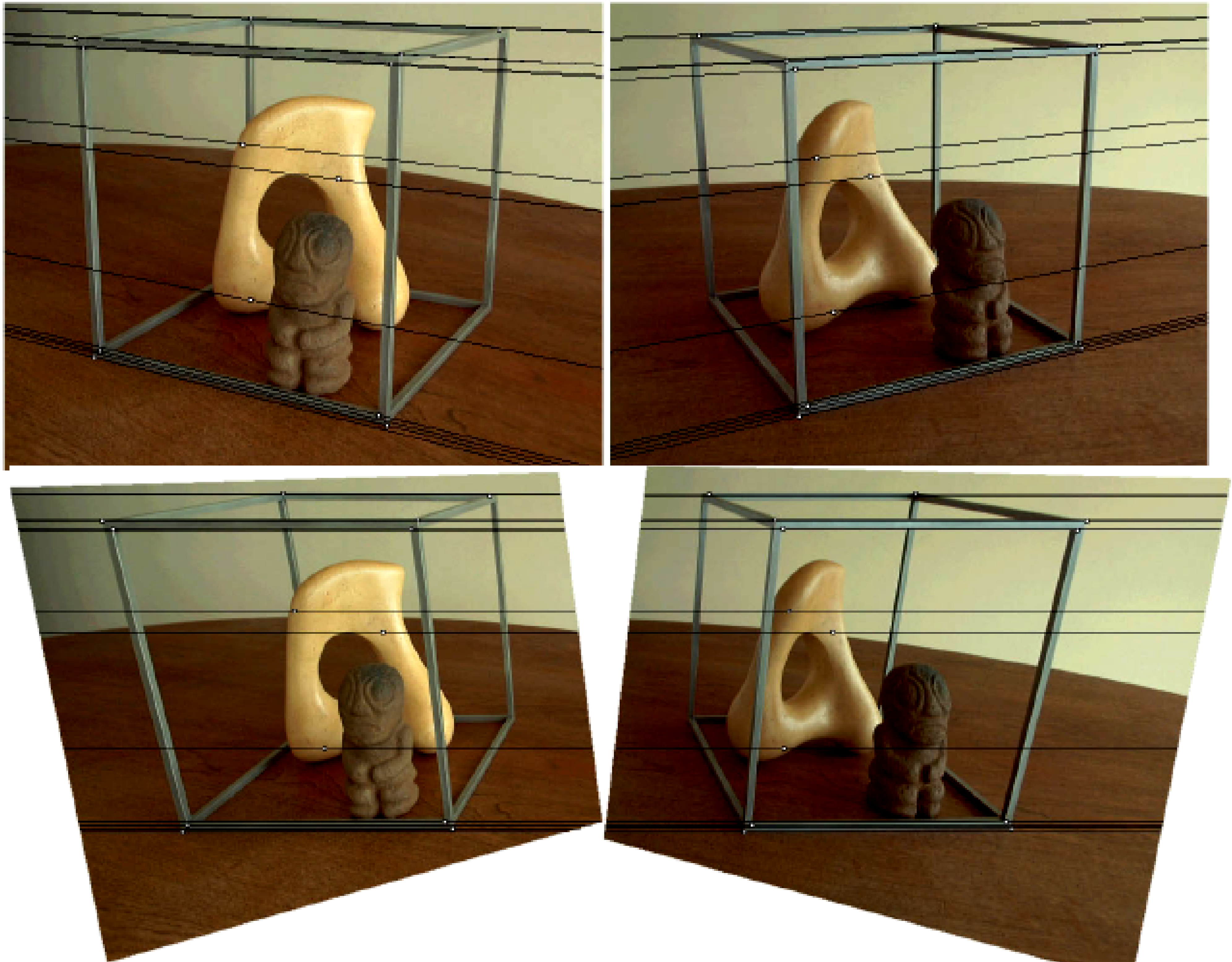
H_s (shearing) only effects the u-coordinate; hence rectification in unaffected. H_r is similarity; H_p is perspective.

$$\mathbf{H}_r = \begin{bmatrix} F_{32} - w_b F_{33} & w_a F_{33} - F_{31} & 0 \\ F_{31} - w_a F_{33} & F_{32} - w_b F_{33} & F_{33} + v_c' \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{H}_r' = \begin{bmatrix} F_{23} - w_b' F_{33} & w_a' F_{33} - F_{13} & 0 \\ F_{13} - w_a' F_{33} & F_{23} - w_b' F_{33} & v_c' \\ 0 & 0 & 1 \end{bmatrix}$$

$$a = \frac{h^2 x_v^2 + w^2 y_v^2}{hw(x_v y_u - x_u y_v)} \quad \text{and} \quad b = \frac{h^2 x_u x_v + w^2 y_u y_v}{hw(x_u y_v - x_v y_u)}$$

Figure The multi-stage stereo rectification algorithm of Loop and Zhang (1999) © 1999 IEEE. (a) Original image pair overlaid with several epipolar lines; (b) images transformed so that epipolar lines are parallel; (c) images rectified so that epipolar lines are horizontal and in vertial correspondence; (d) final rectification that minimizes horizontal distortions.


H_{sh}



Latest/Modern methods of Correspondence/Rectification/reconstruction include:

- Monassee et. al's Rectification – BMVC 2010;

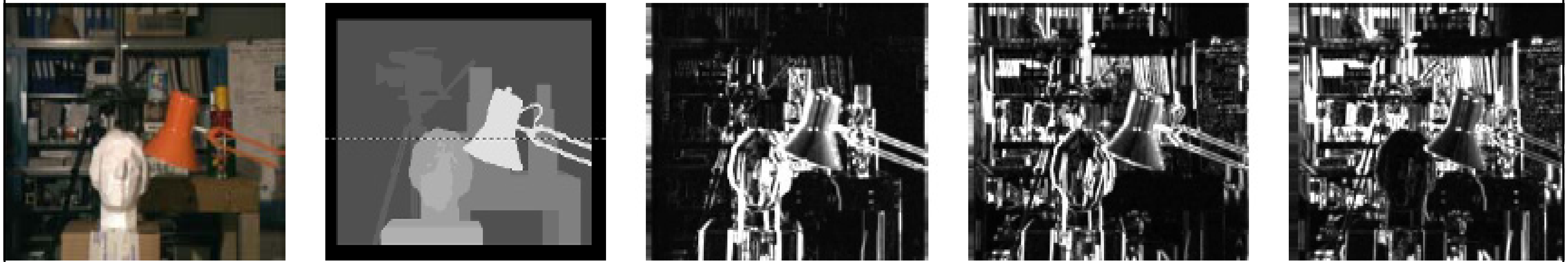
- Plane Sweep;

- Sparse feature set matching

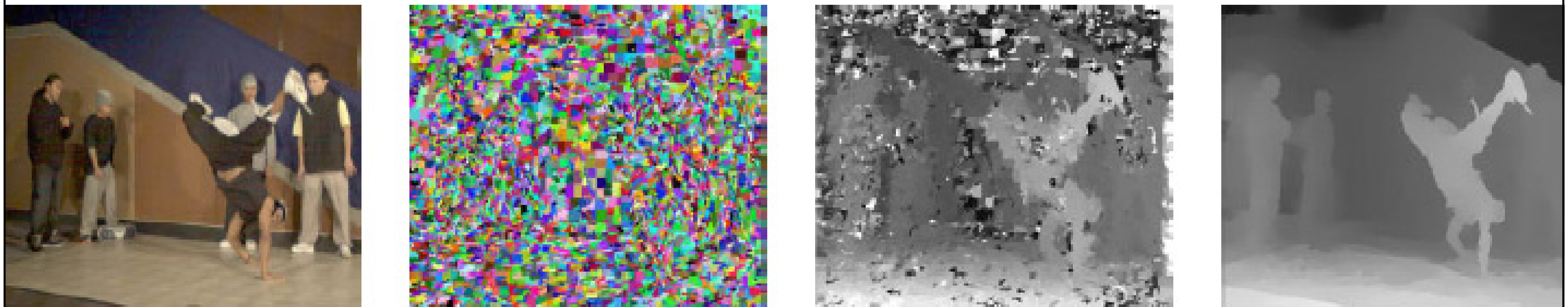
- Profile curves or contours (even occluding)

- Dense correspondences using : similarity measures (NCC, SAD, SSD, MSE, MAD), local methods;

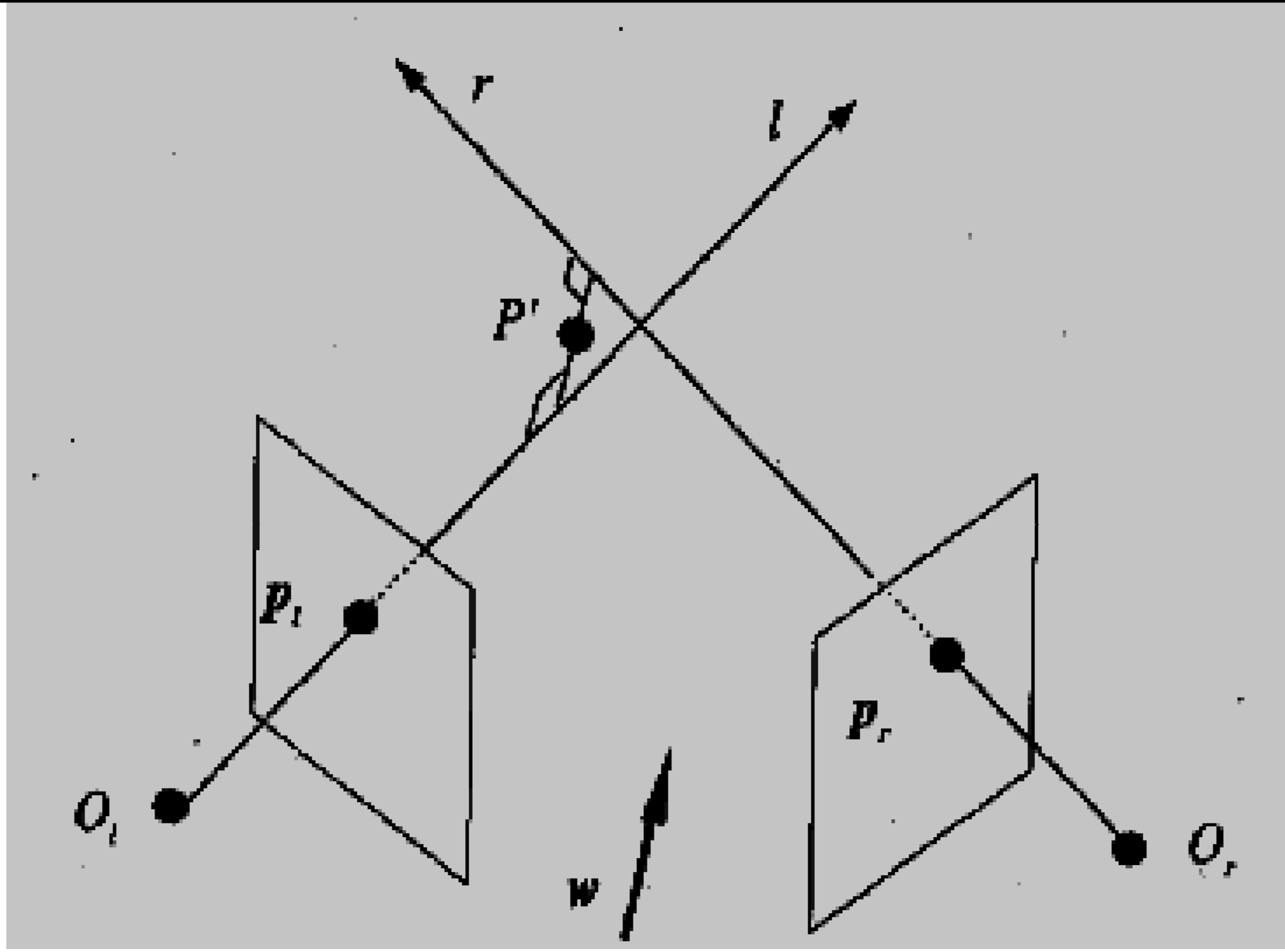
- Global optimization – Dynamic Prog., Segmentation based; etc.



Slices through a typical *disparity space image (DSI)* (Scharstein and Szeliski 2002) c 2002, Springer:
(a) original color image; (b) ground truth disparities;
(b) (c–e) three (*x, y) slices for d = 10, 16, 21;*



Segmentation-based stereo matching (Zitnick, Kang, Uyttendaele *et al. 2004) c 2004 ACM:*(a) input color image; (b) color-based segmentation; (c) initial disparity estimates; (d) final piecewise-smoothed disparities;



A Priori Knowledge	3-D Reconstruction from Two Views
Intrinsic and extrinsic parameters	Unambiguous (absolute coordinates)
Intrinsic parameters only	Up to an unknown scaling factor
No information on parameters	Up to an unknown projective transformation of the environment

Properties of F : $$x'^T Fx = 0$$

(i) Transpose: If F is the fundamental matrix of the pair of cameras (P, P'), then F^T is the fundamental matrix of the pair in the opposite order: (P', P).

(ii) Epipolar lines: For any point x in the first image, the corresponding epipolar line is l' = Fx. Similarly, l = F^Tx' represents the epipolar line corresponding to x' in the second image;

(iii) The epipole: for any point x (other than e) the epipolar line l' = Fx contains the epipole e'. Thus e' satisfies e'^T(Fx) = (e'^TF)x = 0 for all x. It follows that e'^TF = 0, i.e. e' is the left null-vector of F. Similarly Fe = 0, i.e. e is the right null-vector of F.

$$F = [P'C]_\times P'P^+$$

(iv) F is rank-2 homogenous matrix with 7 dof.

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

Canonical cameras, $P = [I \mid 0]$, $P' = [M \mid m]$,

$$[m]_\times M = F = [e']_\times M = M^{-T}[e]_\times, \text{ where } e' = m \text{ and } e = M^{-1}m.$$

F in terms of K

- Let K be the internal parameter matrix of the camera.
- Camera matrix of the second camera (P') is a rotation and translation of the first camera(P):

$$P = K[I \mid 0] \qquad P' = K'[R \mid t] \qquad P^{+} = \begin{bmatrix} K^{-1} \\ 0^{T} \end{bmatrix} \qquad C = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$P'C =$$

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

$$= [K't]_{\times} K'RK^{-1} = K'^{-T}[t]_{\times} RK^{-1} = K'^{-T} R[R^{T}t]_{\times} K^{-1} = K'^{-T} RK^{T}[KR^{T}t]_{\times}$$

Prove it.

- The epipoles, defined as the image of other camera centers are:

$$e = P\begin{bmatrix} -R^{T}t \\ 1 \end{bmatrix} = KR^{T}t \qquad e' = P'\begin{bmatrix} 0 \\ 1 \end{bmatrix} = K't$$

$$F = [e']_{\times} K'RK^{-1} = K'^{-T}[t]_{\times} RK^{-1} = K'^{-T} R[R^{T}t]_{\times} K^{-1} = K'^{-T} RK^{T}[e]_{\times}$$

For any vector t and non-singular matrix M:

$$[t]_\times M = M^{-T}\left[M^{-1}t\right]_\times$$

$$[\mathbf{K}'\mathbf{t}]_\times \mathbf{K}' R K^{-1} = \mathbf{K}'^{-T}[\mathbf{K}'^{-1}\mathbf{K}'\mathbf{t}]_\times R K^{-1} = \mathbf{K}'^{-T}[\mathbf{t}]_\times R K^{-1}$$

$$K'^{-T}[\mathbf{t}]_\times \mathbf{R} K^{-1} = K'^{-T}\mathbf{R}^{-T}[\mathbf{R}^{-1}\mathbf{t}]_\times K^{-1} = K'^{-T}\mathbf{R}[\mathbf{R}^T\mathbf{t}]_\times K^{-1}$$

$$K'^{-T} R[\mathbf{R}^T\mathbf{t}]_\times \mathbf{K}^{-1} = K'^{-T} R \mathbf{K}^T[\mathbf{K}\mathbf{R}^T\mathbf{t}]_\times$$

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* $P'^{\top}FP$ *is skew-symmetric.*

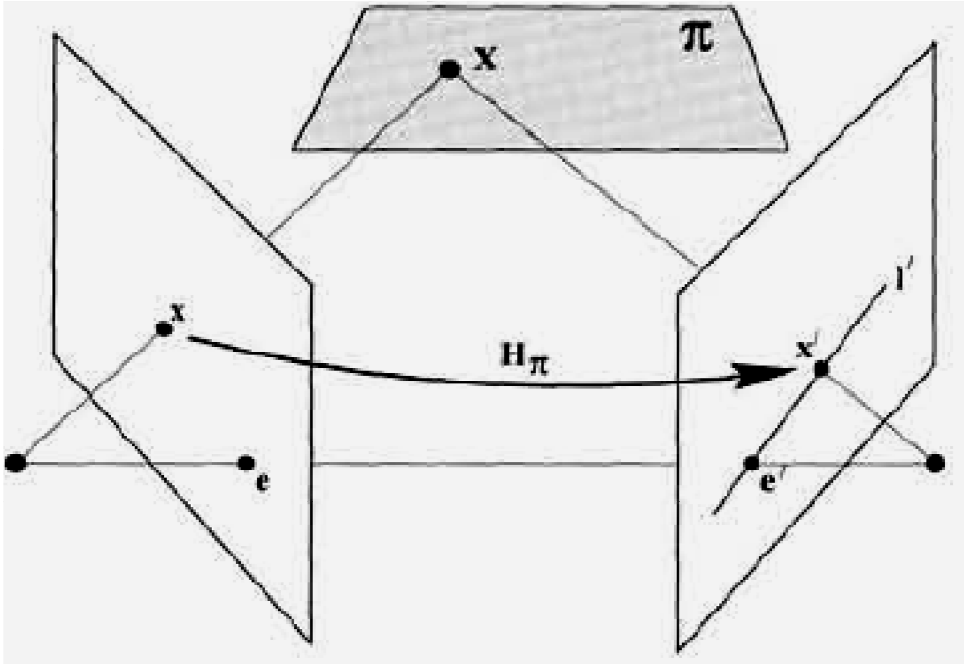
Proof. The condition that $P'^{\top}FP$ is skew-symmetric is equivalent to $\mathbf{X}^{\top}P'^{\top}FP\mathbf{X} = 0$ for all $\mathbf{X}$. Setting $\mathbf{x}' = P'\mathbf{X}$ and $\mathbf{x} = P\mathbf{X}$, this is equivalent to $\mathbf{x}'^{\top}F\mathbf{x} = 0$, which is the

Homography: x′ = Hx;

Relationship with Fundamental matrix, F: $\Leftarrow x'^T F x = 0$

Hx′ lies on the corresponding epipolar line: $F^T x'$

Thus, e′ = He; $H^{-1}e' = e$;

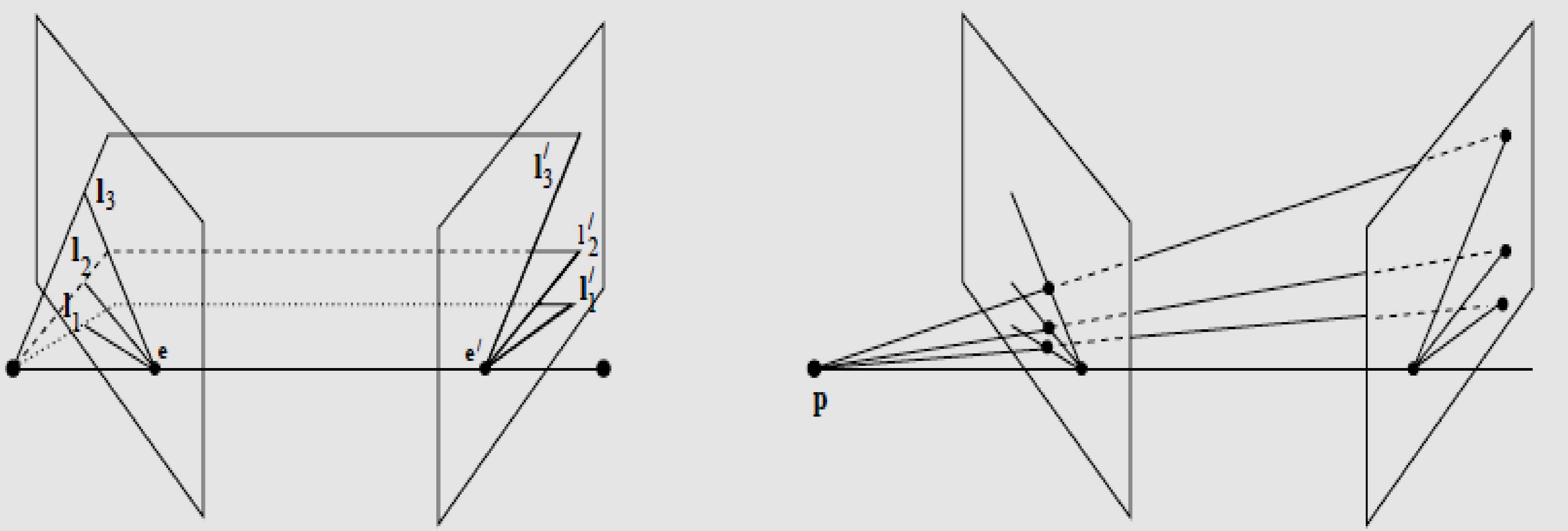


$$F = [P'C]_\times P'P^+$$
$$= [K't]_\times K'RK^{-1} = K'^{-T}[t]_\times RK^{-1} = K'^{-T}R[R^T t]_\times K^{-1} = K'^{-T}RK^T[KR^T t]_\times$$

$$F = [e']_\times K'RK^{-1} = K'^{-T}[t]_\times RK^{-1} = K'^{-T}R[R^T t]_\times K^{-1} = K'^{-T}RK^T[e]_\times$$

$$F = K'^{-T}RK^T[KR^T t]_\times = [e']_\times K'RK^{-1} = K'^{-T}RK^T[e]_\times = [e']_\times P'P^+ = [e']_\times H_\pi$$

where, H_π is the homography imposed by epipolar plane.



Result 9.5. Suppose $\boldsymbol{l}$ and $\boldsymbol{l'}$ are corresponding epipolar lines, and k is any line not passing through the epipole **e**, then $\boldsymbol{l}$ and $\boldsymbol{l'}$ are related by:

$$l' = F[k]_\times l;$$

Symmetrically, $l = F^T[k']_\times l';$

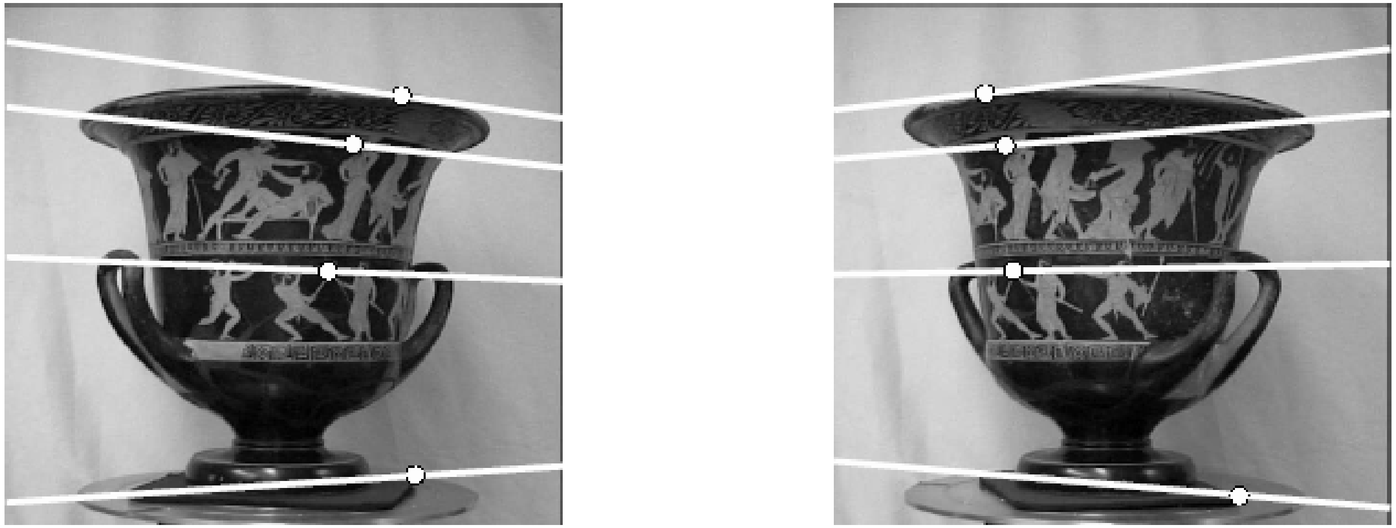
$$[k]_\times l = k \times l \Rightarrow \text{x (a point, as intersection of two lines)}; \quad F[k]_\times l = F\,\text{x} = l';$$

Let, line k be a line "e" :-(; as : $k^T e = \text{e}^T e \neq 0$;

Hence, line "e" does not pass thru epipole e.

$$l' = [e']_\times H_\pi \,\text{x} = F\,\text{x} = F[e]_\times l; \quad l = F^T[e']_\times l'$$

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



Typical methods used to estimate F:

- 8-pt DLT algo.

- RANSAC

$$\mathbf{m}'^T \mathbf{F} \mathbf{m} = 0,$$

=>

$$Af = 0;$$

- Normalize data, using Transformation matrix T_{TS}
- DLT; F is the "smallest singular" vector of A
- replace F by F~, using SVD, where det (F~) = 0
- Denormalize, as:

$$F = T'^T \tilde{F} T$$

Also, look at Gold Standard method based on MLE

E, the essential matrix

Maps a point from one image plane to a line in the corresponding image domain; Has 5 dof.

Two images of a single scene/object are related by the epipolar geometry, which can be described by a 3x3 singular matrix called the **essential matrix** if images' internal parameters are known, or the fundamental matrix otherwise. Mostly used in case of SFM problems.

$$P = K[R \mid t] \qquad \mathbf{x} = PX = \mathrm{K}[\mathrm{R} \mid \mathbf{t}]\mathbf{X}$$

$$let, \ \hat{\mathrm{x}} = K^{-1}\,\mathrm{x} = [R|t]X$$

x^ is in normalized coordns.

And <u>normalized camera matrix</u> is : (where the effect of known camera calibration matrix has been removed.)

$$K^{-1}P = [R \mid t]$$

$$\hat{\mathrm{x}}'^{T} E \,\hat{\mathrm{x}} = 0$$

The fundamental matrix corresponding to the pair of normalized cameras is customarily called the ***essential matrix***.

Thus for a pair of normalized cameras:

$$P = [I \mid 0]$$
$$P' = [R \mid t]$$

Using:

$$F = K'^{-T}[t]_{\times} R K^{-1} = K'^{-T} R [R^T t]_{\times} K^{-1}$$

and ignoring K & K′: $E =$

So actually:

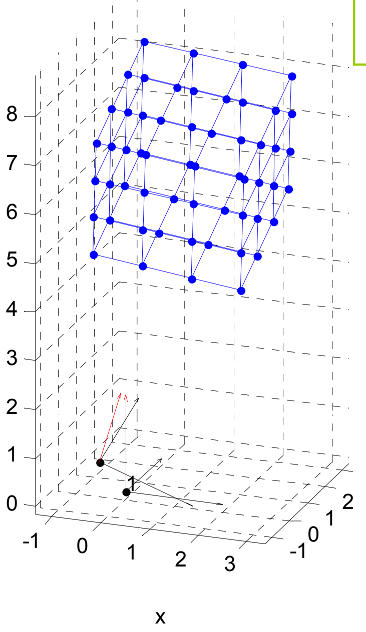
$$\hat{x}'^{T} E \, \hat{x} = 0$$

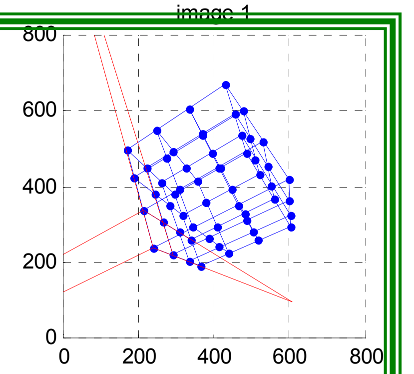
$$=$$

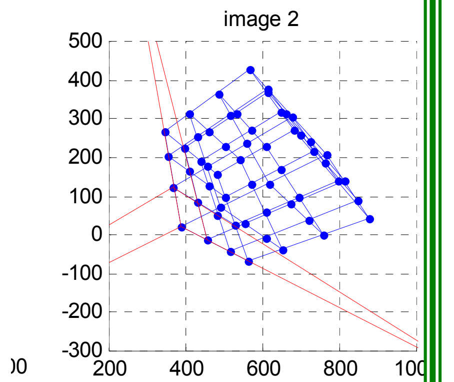
$$\Rightarrow F =$$

A 3 x 3 matrix is an essential matrix, E if and only if two of its singular values are equal, and the third is zero .

Reconstruction Framework







3D world object

2D View (Cam1) → Feature Extraction

2D View (Cam2) → Feature Extraction

Find Correspondence

↓

Fundamental / Essential Matrix

$$x'^T Fx = 0 \text{ or } x'^T Ex = 0$$

$$P = [I \mid 0] \text{ and } P' = [[e']_\times F \mid e']$$

$$E = K'^T FK;$$

↓

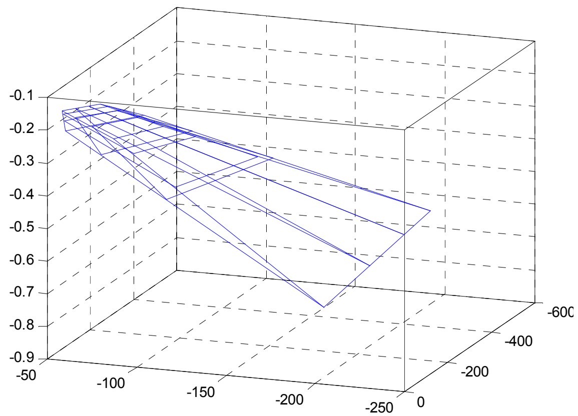
Projective Reconstruction

(Triangulation process) [a]

$$\mathrm{x} = PX$$

$$\mathrm{x} \times PX = 0$$

$$AX = 0$$

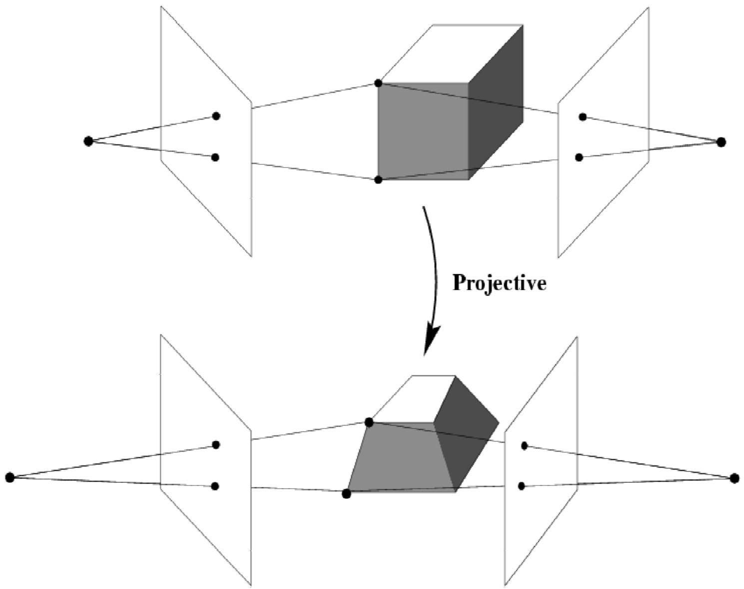


[a] R.Hartley, CVIU ‘94

Ambiguity in Reconstruction

- From Image correspondences, the scene and the camera can be reconstructed to a projective equivalent of the original scene and camera
- Projective Reconstruction theorem:

$$x_i = PX_i = \left(PH^{-1}\right)\left(H\ X_i\right)$$



- Additional information (scene parallel lines, camera internal parameters) required for metric reconstruction

GENERIC STEREO RECONSTRUCTION (sec. 10.6, pp 277; H&Z)

Input: Two Uncalibrated images;

Output: Reconstruction (metric) of the scene structure and camera

Algo. Steps:

- **Projective reconstruction**
  - **Compute Fundamental matrix, F**
  - **Compute P and P′ (camera matrices) using F**
  - **Use triangulation (with rectification) to get X, from x_i and x_i'**

- **Rectify from projective to Metric (M), using either**

  (a) Direct:

  Estimate homography H, from grnd. Control pts.,;

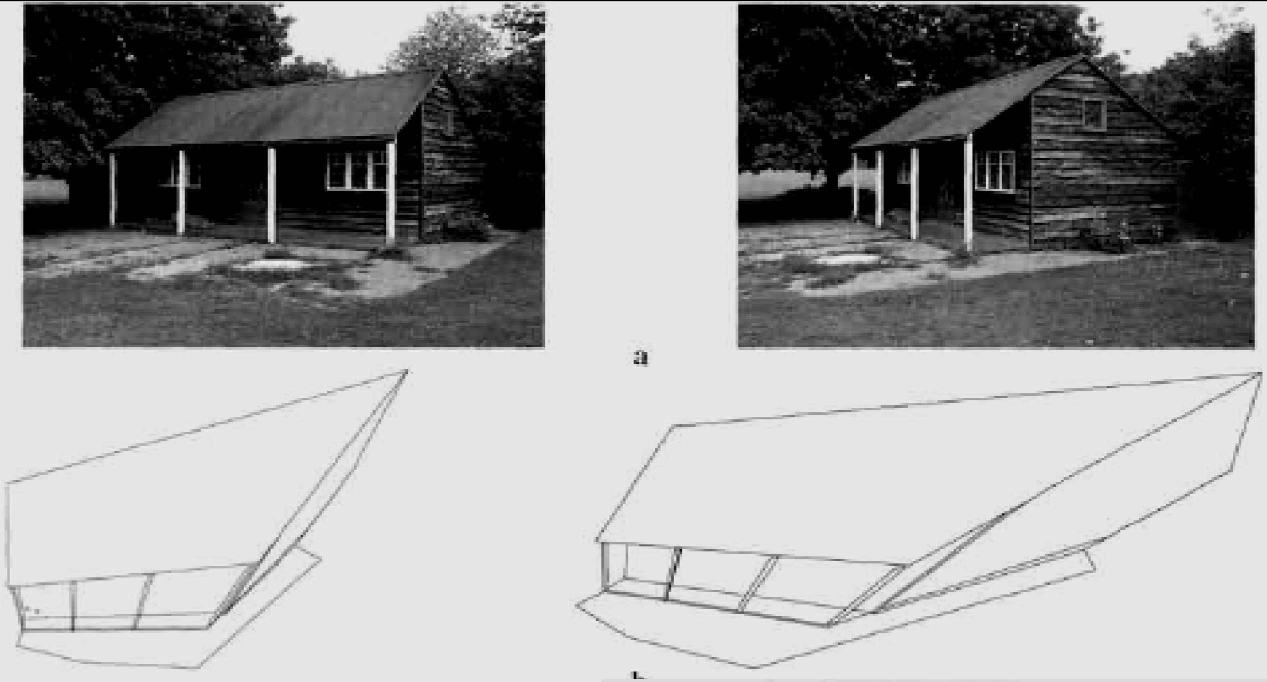
  $P_M = P.H^{-1}$; $P'_M = P'.H^{-1}$; $X_{Mi} = HX_i$.

  OR

  (b) Stratified (use, VP, VL, VPI, Homography, DIAC etc.):

  Affine;

  Metric

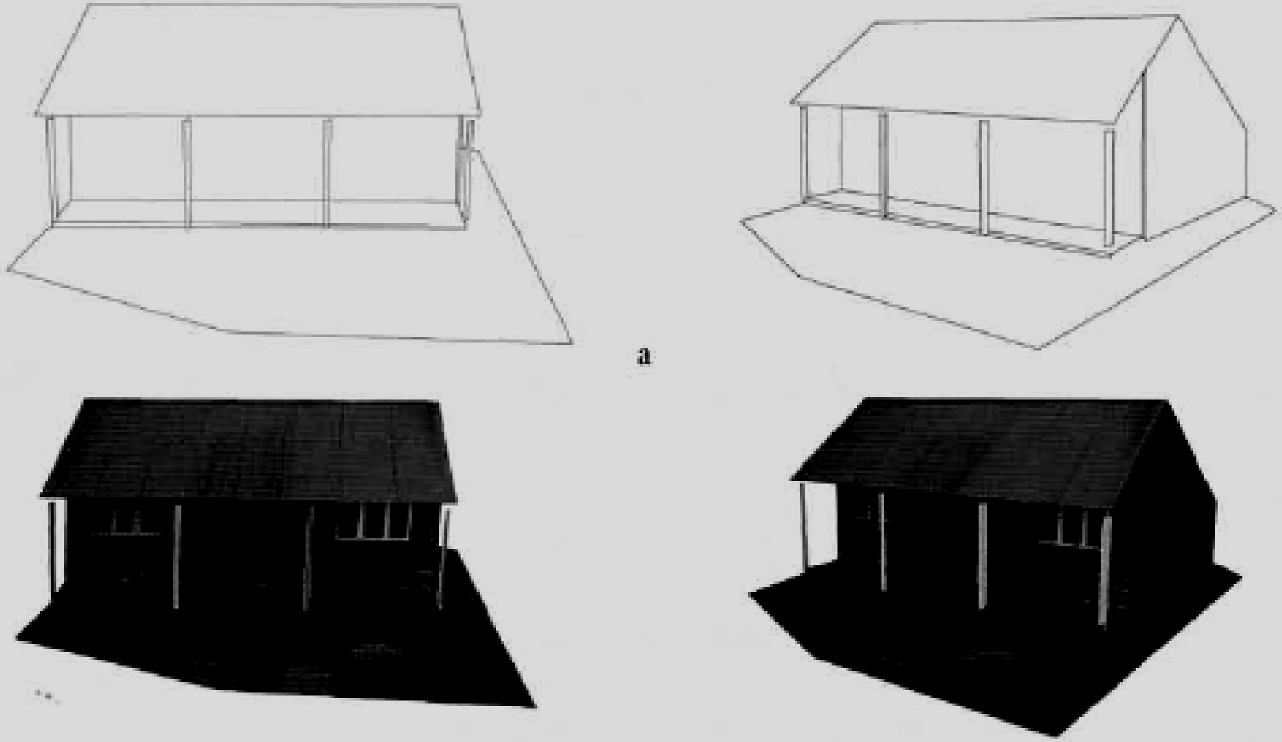


a

Fig. 10.3. **Projective reconstruction.** *(a) Orig*
construction of the scene. The reconstruction re
information about the scene geometry. The fund
dences between the images, camera matrices are
triangulation from the correspondences. The line

a

Fig. 10.6. **Direct reconstructio**
metric by specifying the position
responding points on the project
points are mapped to their world



a

b

Fig. 10.5. **Metric reconstruction.** *The affine reconstruction of figure 10.4 is upgraded to metric by computing the image of the absolute conic. The information used is the orthogonality of the directions*

Vanishing points

Points on a line in 3 space through point A and direction $D = (d^T, 0)^T$ are $X(\lambda) = A + \lambda D$. As λ goes from zero to infinity, then X(λ) varies from finite point A to point D at ∞.
Assume P = K [I 0], then image of X(λ) is given by

$$x(\lambda) = PX(\lambda) = PA + \lambda PD = a + \lambda Kd$$

$$v = \lim_{\lambda \to \infty} x(\lambda) = \lim_{\lambda \to \infty} (a + \lambda Kd) = Kd$$

note that v depends only on the direction d of the line, not on its position specified by A

➔ Conclusion: the vanishing point of lines with direction d in 3 space is the intersection v of the image plane with a ray through the camera center with direction d, namely v = Kd

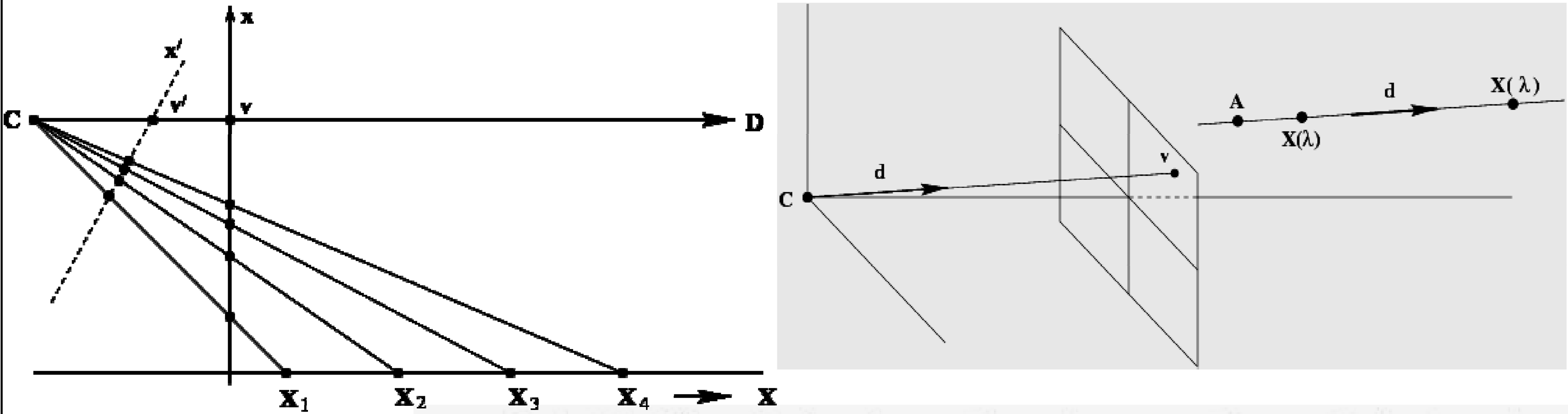
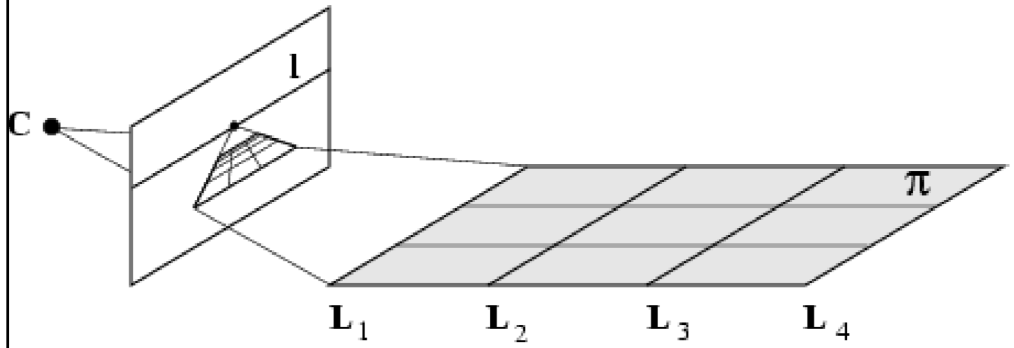
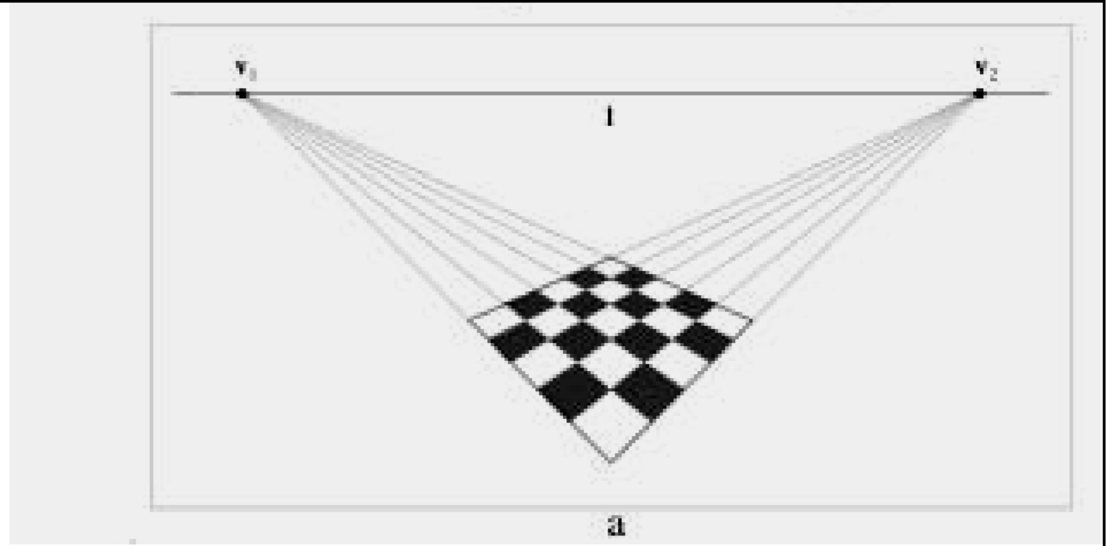


Fig. 8.14. **Vanishing point formation.** *(a) Plane to line camera. The points X_i, $i = 1, \ldots, 4$ are equally spaced on the world line, but their spacing on the image line monotonically decreases. In the limit X ∞ the world point is imaged at x = v on the vertical image line, and at x = v on the inclined image line. Thus the vanishing point of the world line is obtained by intersecting the image plane with a ray parallel to the world line through the camera centre C. (b) 3-space to plane camera. The vanishing point, v, of a line with direction d is the intersection of the image plane with a ray parallel to d through C. The world line may be parametrized as X( ) = A + D, where A is a point on the line, and* $D = (d^T, 0)^T$.

Vanishing lines





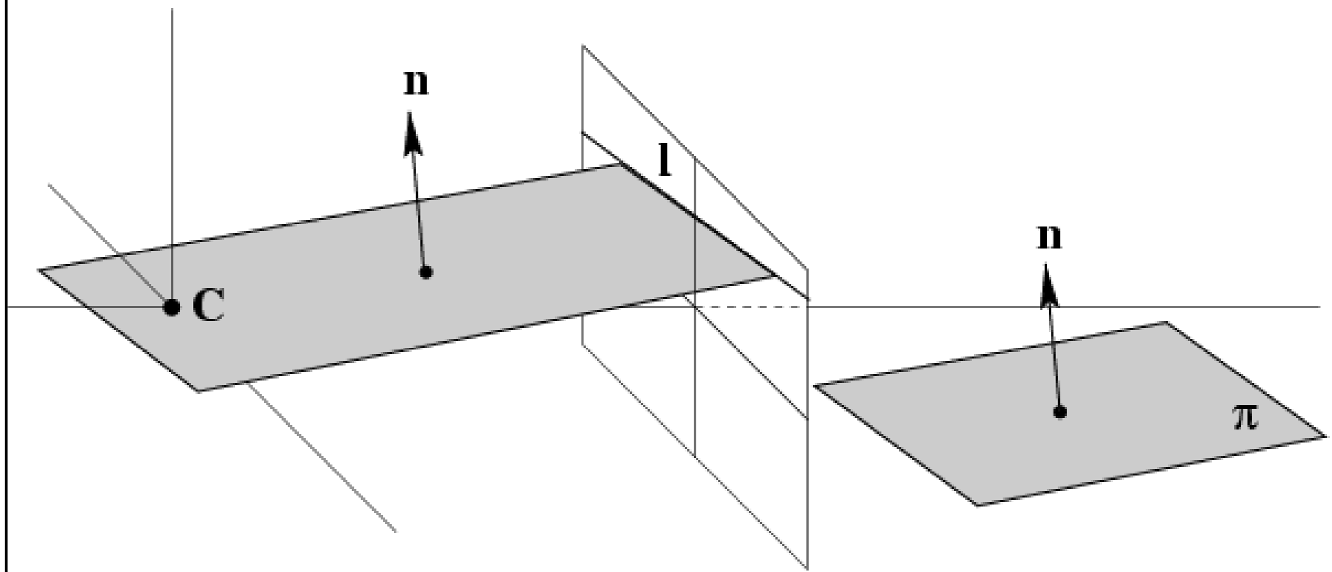
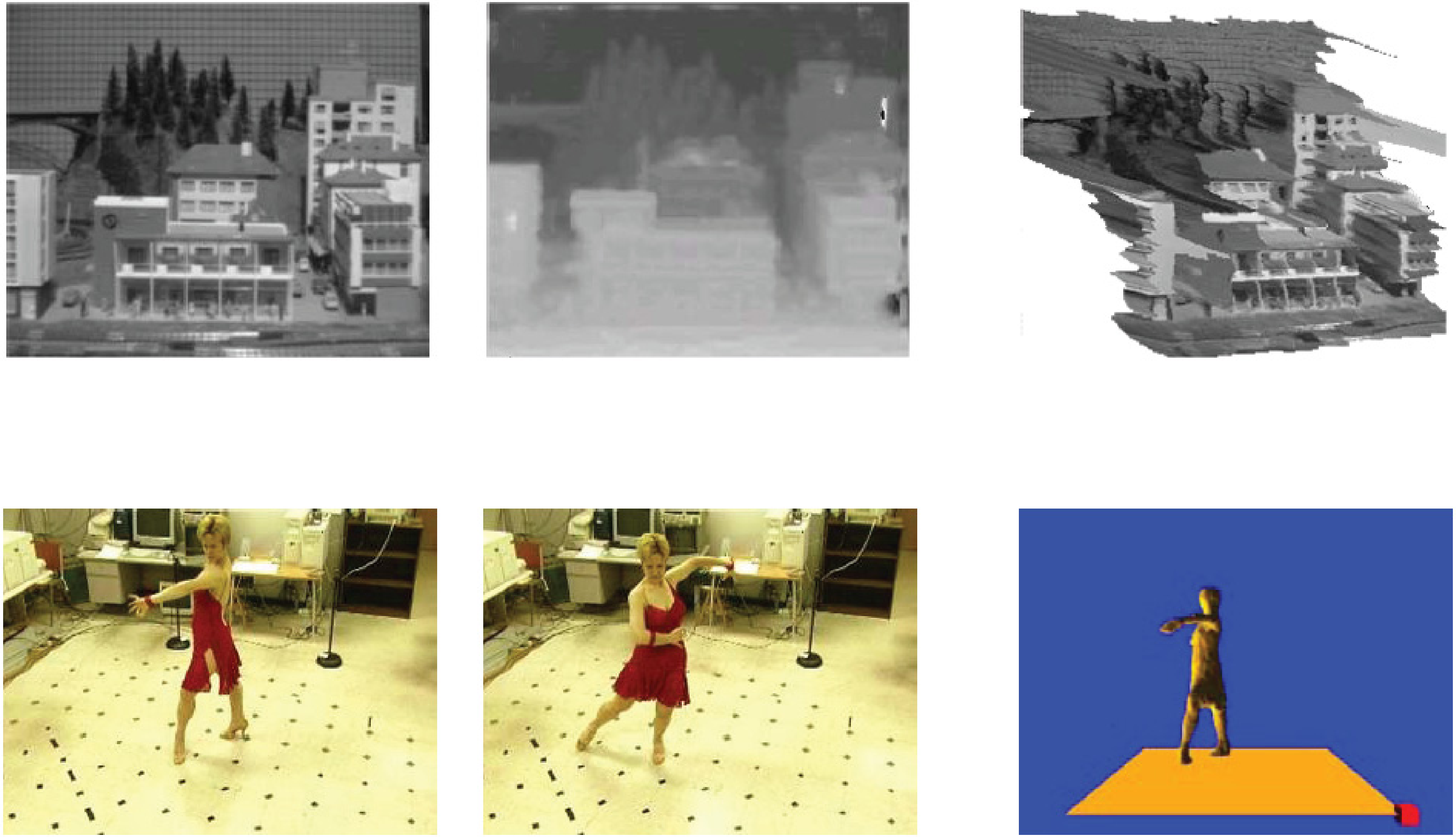


Fig. 8.16. **Vanishing line formation.** *(a) The two sets of parallel lines on the scene plane converge to the vanishing points* $\mathbf{v}_1$ *and* $\mathbf{v}_2$ *in the image. The line* $\mathbf{l}$ *through* $\mathbf{v}_1$ *and* $\mathbf{v}_2$ *is the vanishing line of the plane. (b) The vanishing line* $\mathbf{l}$ *of a plane* $\boldsymbol{\pi}$ *is obtained by intersecting the image plane with a plane through the camera centre* $\mathbf{C}$ *and parallel to* $\boldsymbol{\pi}$.

(a) (b) (c)

(d) (e) (f)

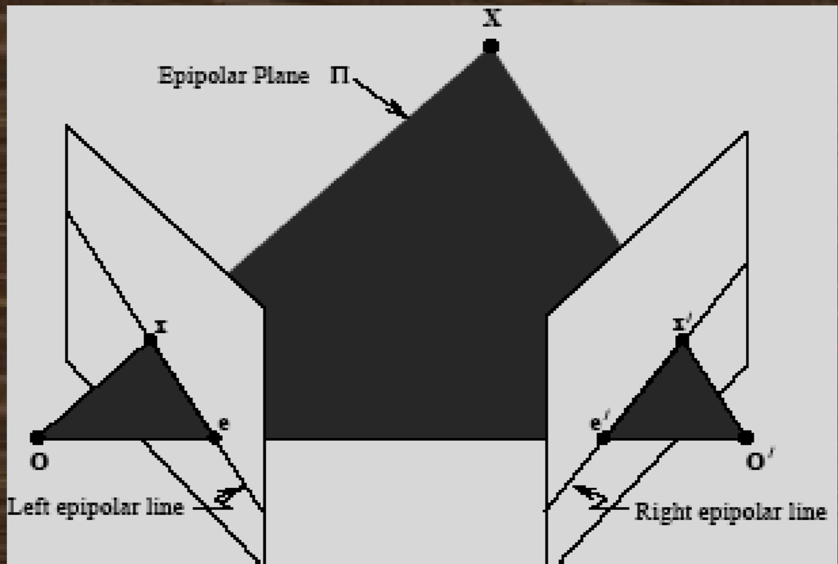
Courtesy: Szeliski

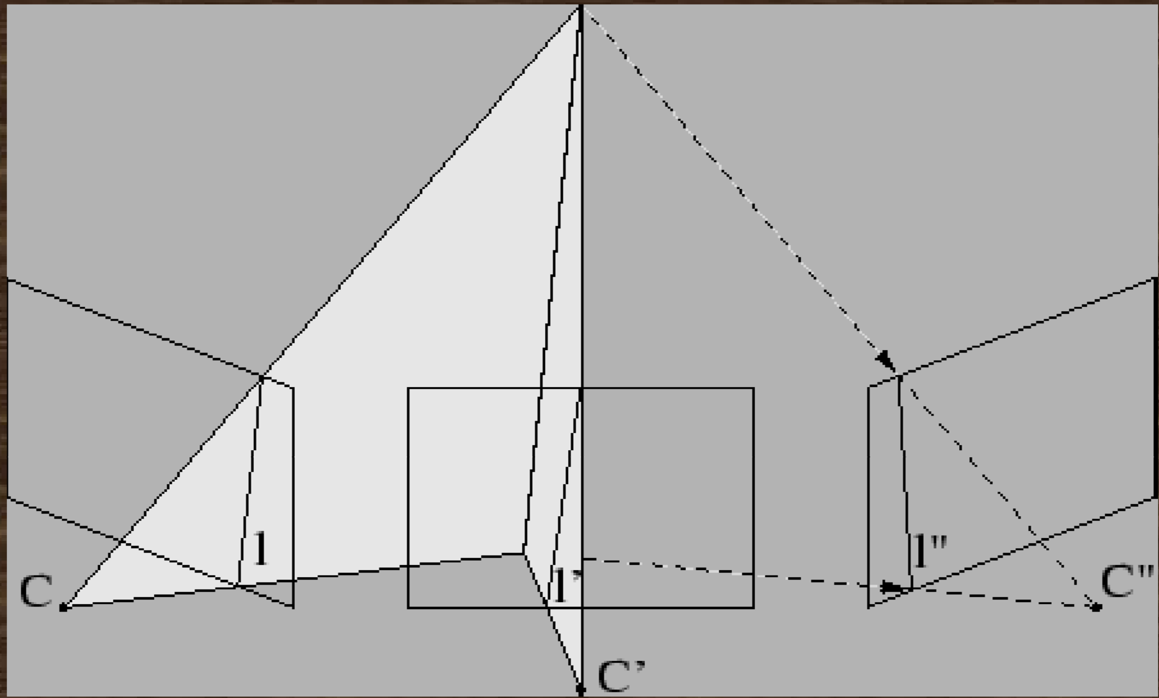


Courtesy: Szeliski

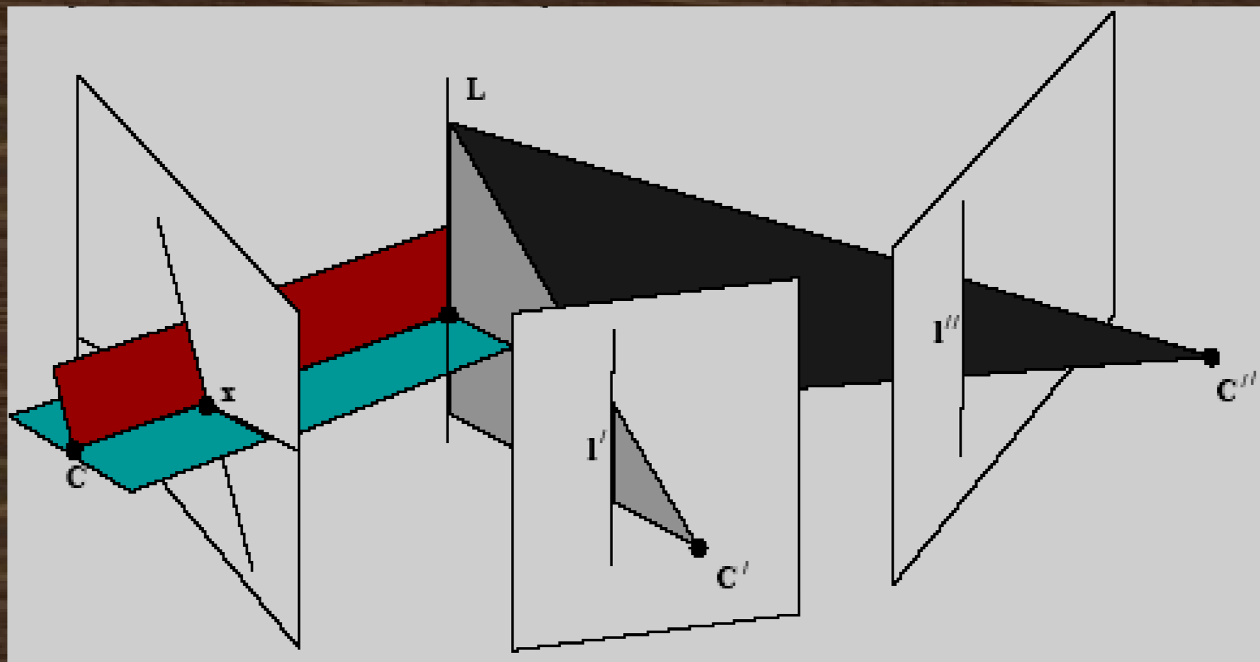
In case of a set of arbitrary views (multi-view geometry) used for 3-D reconstruction (object structure, surface geometry, modeling etc.), methods used involve:

- KLT (Kanade-Lucas-Tomasi)- tracker
- Bundle adjustment and RANSAC
- 8-point DLT algorithm
- Zhang's scene homography
- *Tri-focal tensors*
- Cheriality and DIAC
- Auto-calibration
- Affine to Metric reconstruction
- Stratification
- Kruppa's eqn. for infinite homography





l
C
l'
C'
l''
C''



L
x
C
l'
C'
l''
C''

Example of 3-D reconstruction

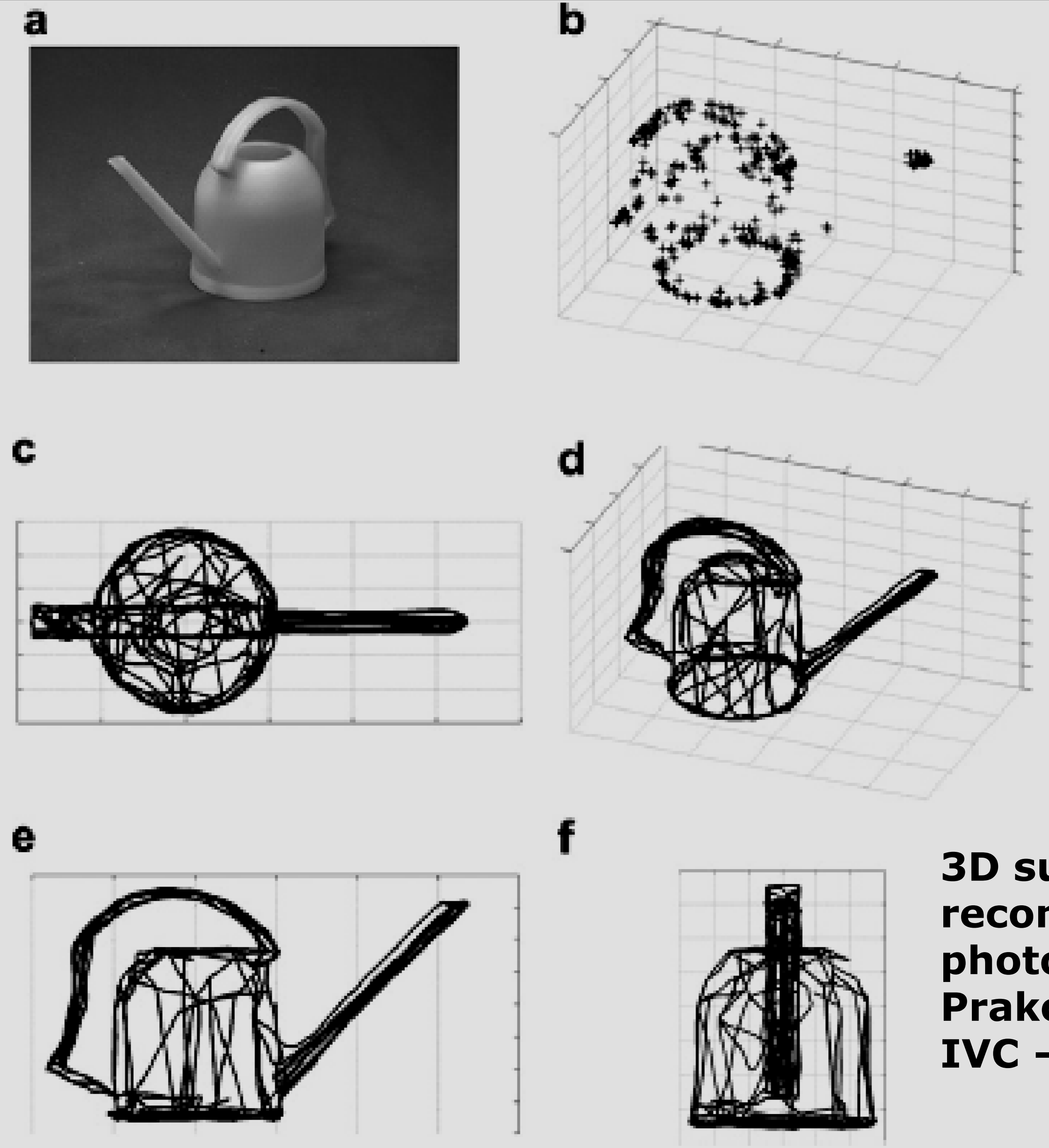
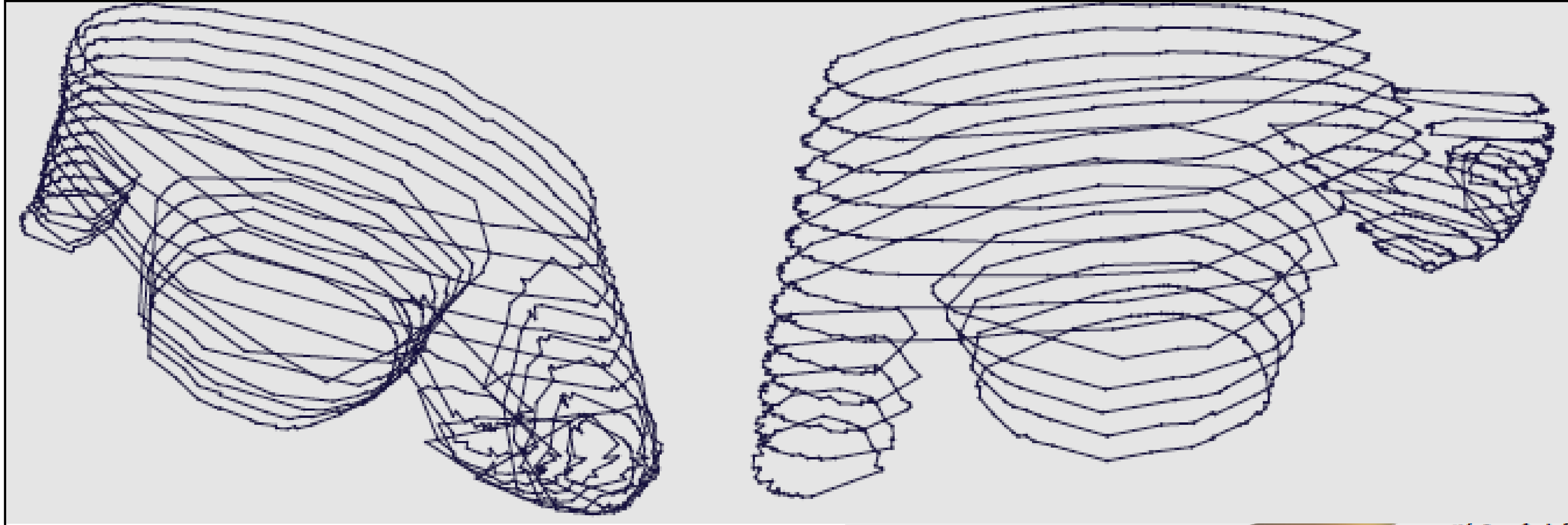
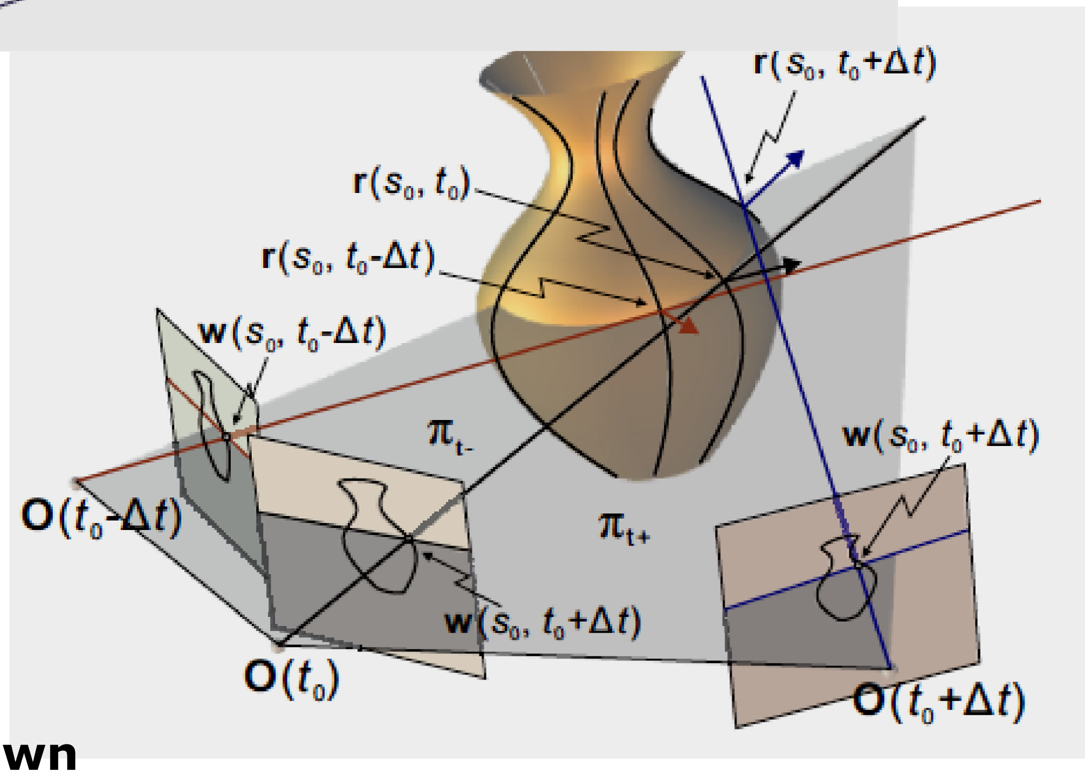


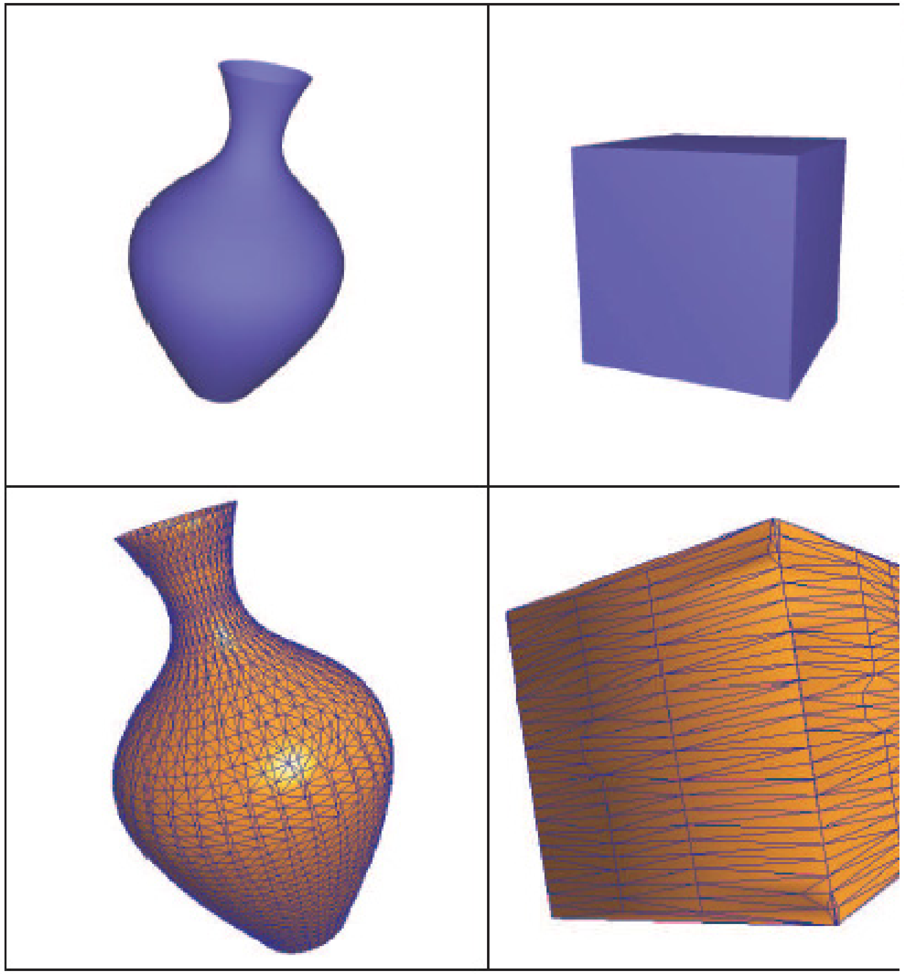
Fig. 18. (a) Real matt plastic watering pot. (b) The reconstructed 3D frontier points shown superimposed upon the pot. (c) - (f) Different views of the reconstructed 3D contour generators.

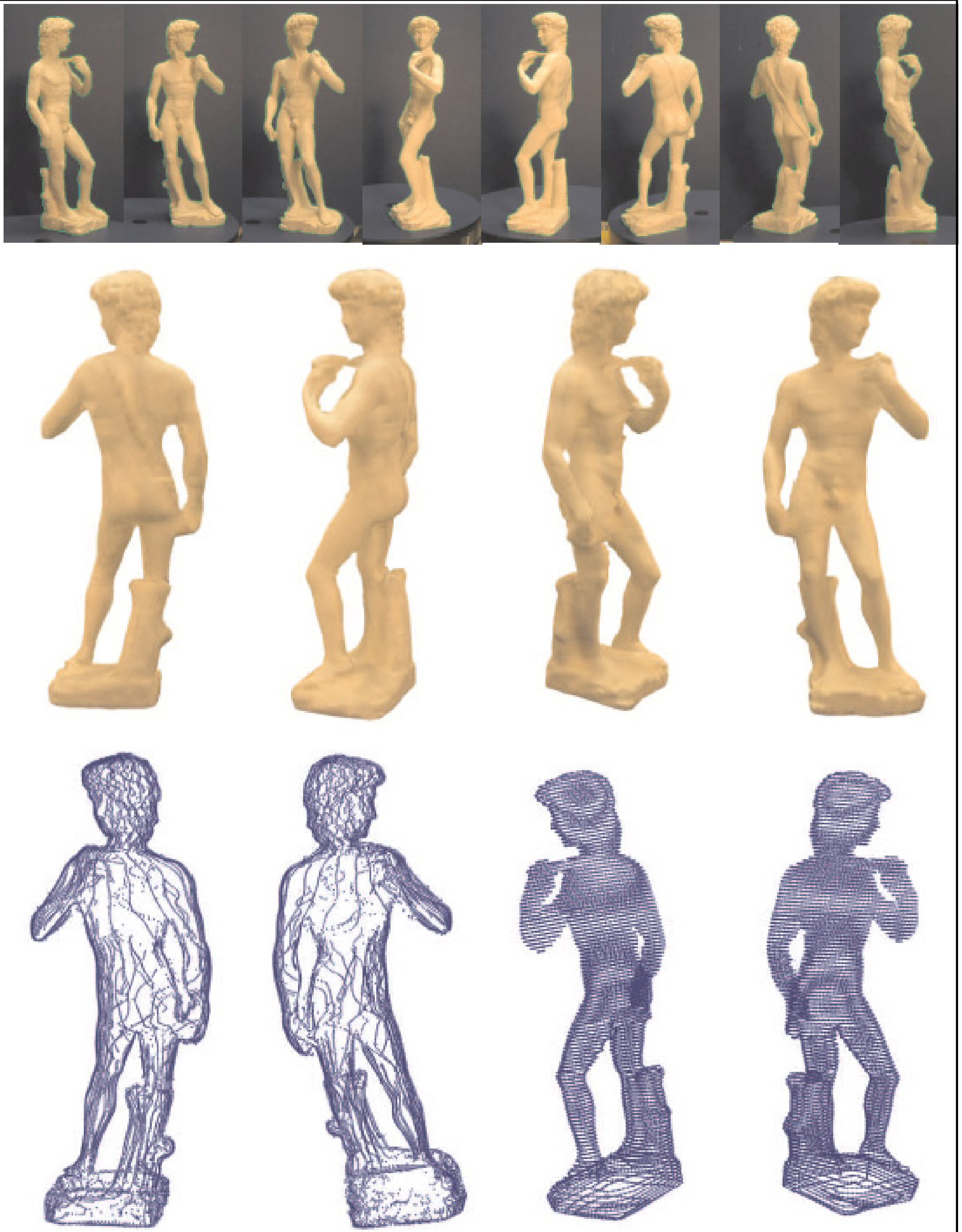
3D surface point and wireframe reconstruction from multiview photographic images; Simant Prakoonwit, Ralph Benjamin; IVC – 2008/9





Robust Recovery of Shapes with Unknown Topology from the Dual Space;
Chen Liang and Kwan-Yee K. Wong,
IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE.





References

1. Multiple View geometry. R. Hartley and A. Zisserman. 2002 Cambridge university Press
2. Invitation to 3D vision. Y.Ma, S.Soatto, J.Kosecka and S.Sastry. 2006 Springer
3. Stratification of 3-dimensional vision: Projective, affine, and metric representations. O. Faugeras, JOSA-A12(3), 465–484 (1995)
4. R. Hartley Triangulation. CVIU, 146-157, 1997
5. R. Hartley.: In defense of the eight-point algorithm. PAMI ‘97
6. R. Hartley. Chirality. IJCV 41-61, 1998
7. Mark Pollefeys et. al: Self-Calibration and Metric Reconstruction in spite of Varying and Unknown Intrinsic Camera Parameters. IJCV 99
8. Forsyth & Ponce – Modern CV…..

End of Lectures on -

Transformations,
Imaging Geometry,
Stereo Vision
and
3-D Reconstruction