Computer Vision – Transformations, Imaging Geometry and Stereo Vision

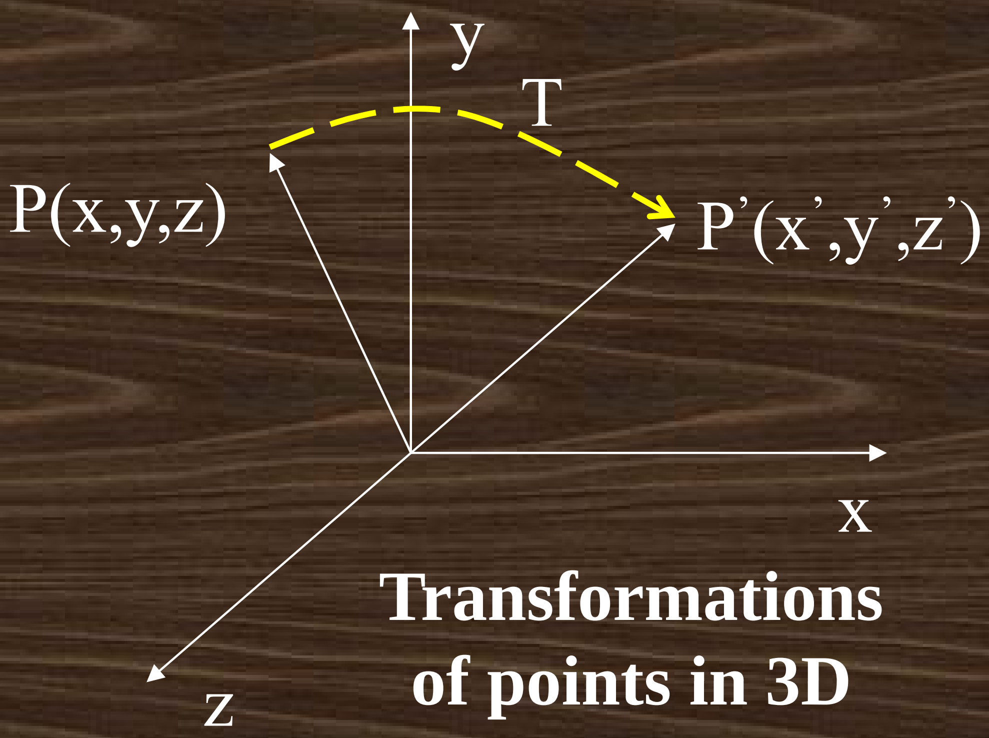
Dr. S. Das

IIT Madras, Chennai-36

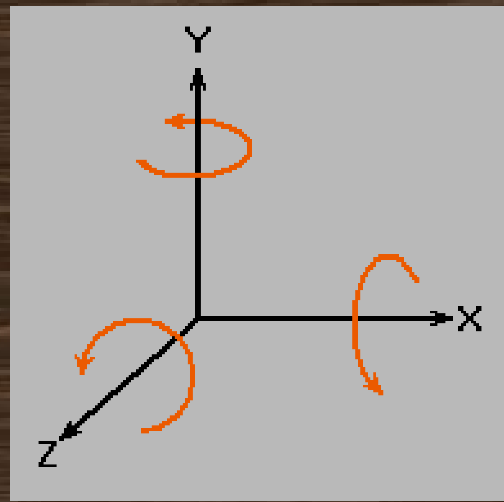
BASICS

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

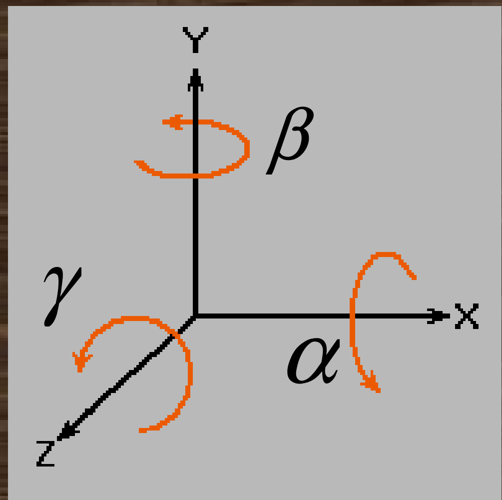
$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix}$$

- Scaling: P'= SP

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

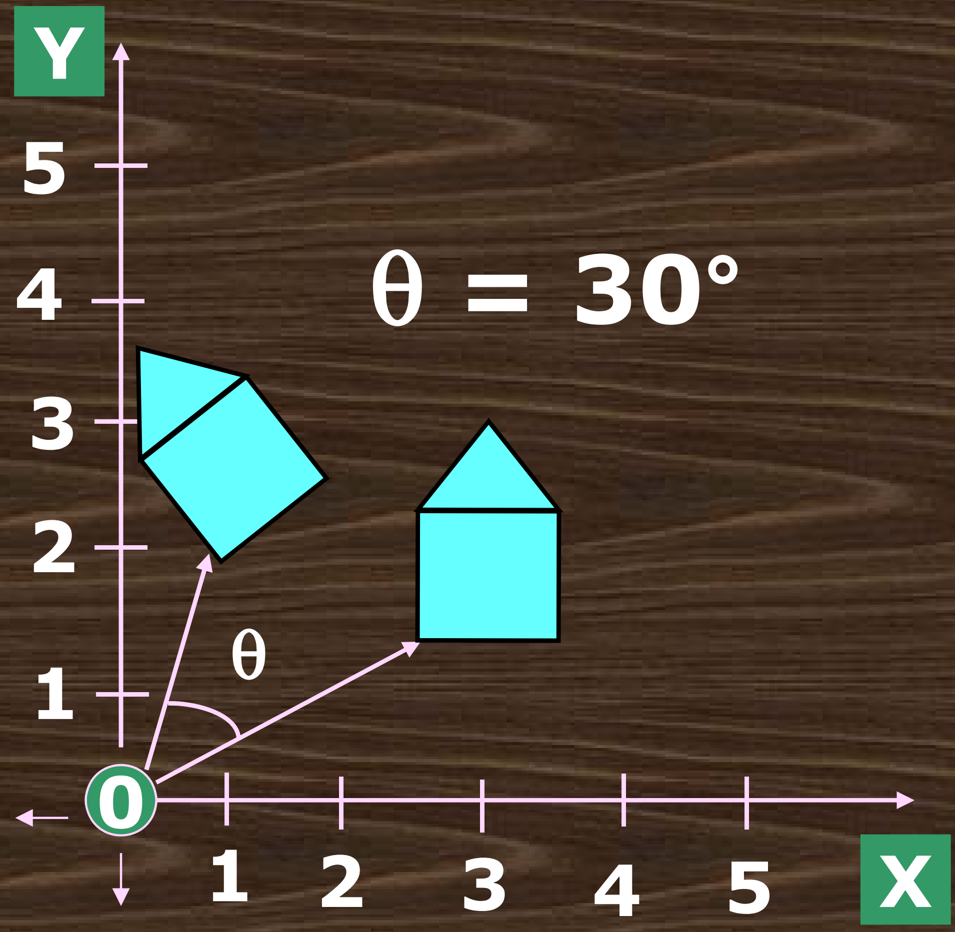
- Rotation : about an axis,

  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

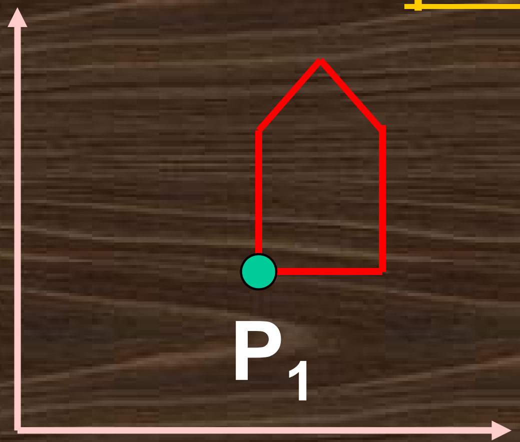
For rotations, $|R| = 1$ and $[R]^T = [R]^{-1}$. Rotation matrices are orthogonal.

Rotation about an arbitrary point P in space

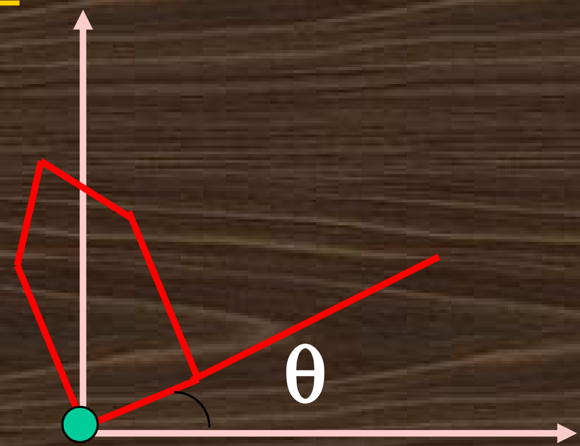
As we mentioned before, rotations are applied about the origin. So to rotate about any arbitrary point P in space, **translate** so that P coincides with the origin, then **rotate**, then **translate back**. Steps are:

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

Rotation about an arbitrary point P in space

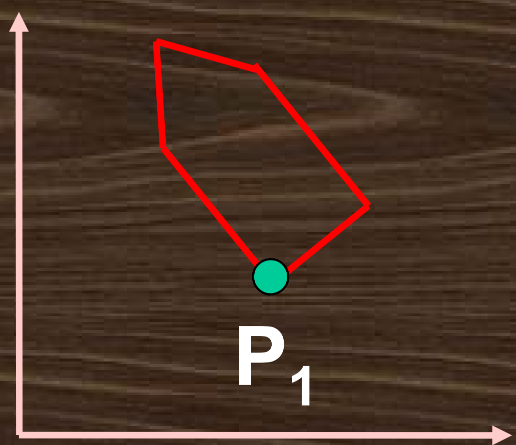


House at P_1



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

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

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

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:

$$P = \left| \mathrm{x\ y\ z\ w} \right|^{T}$$

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

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)

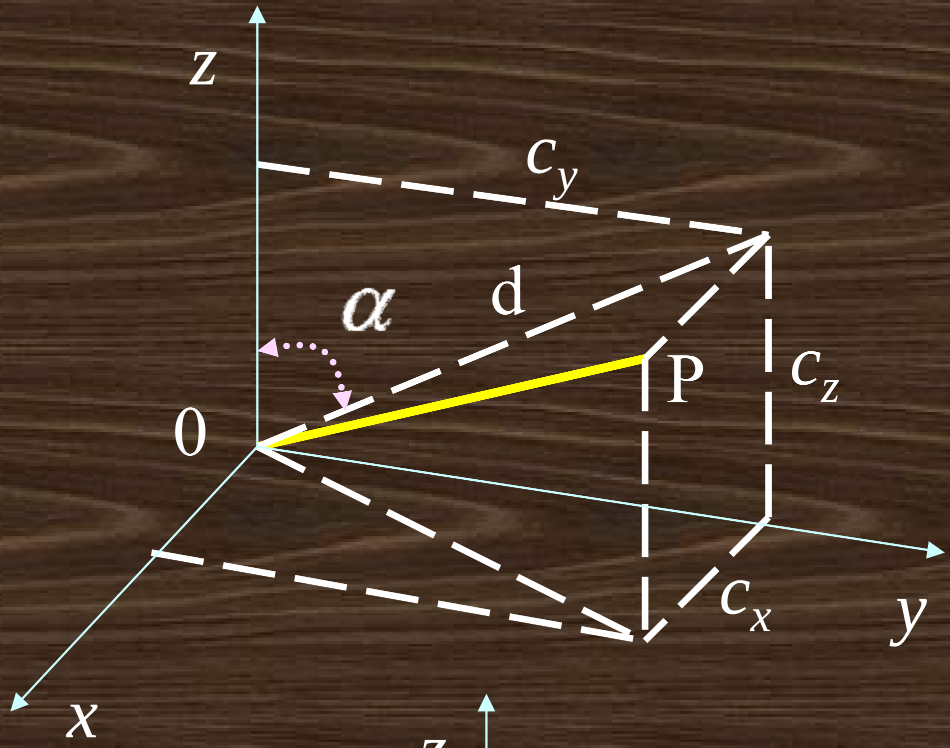
Rotation About an Arbitrary Axis in Space

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)**

The 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).

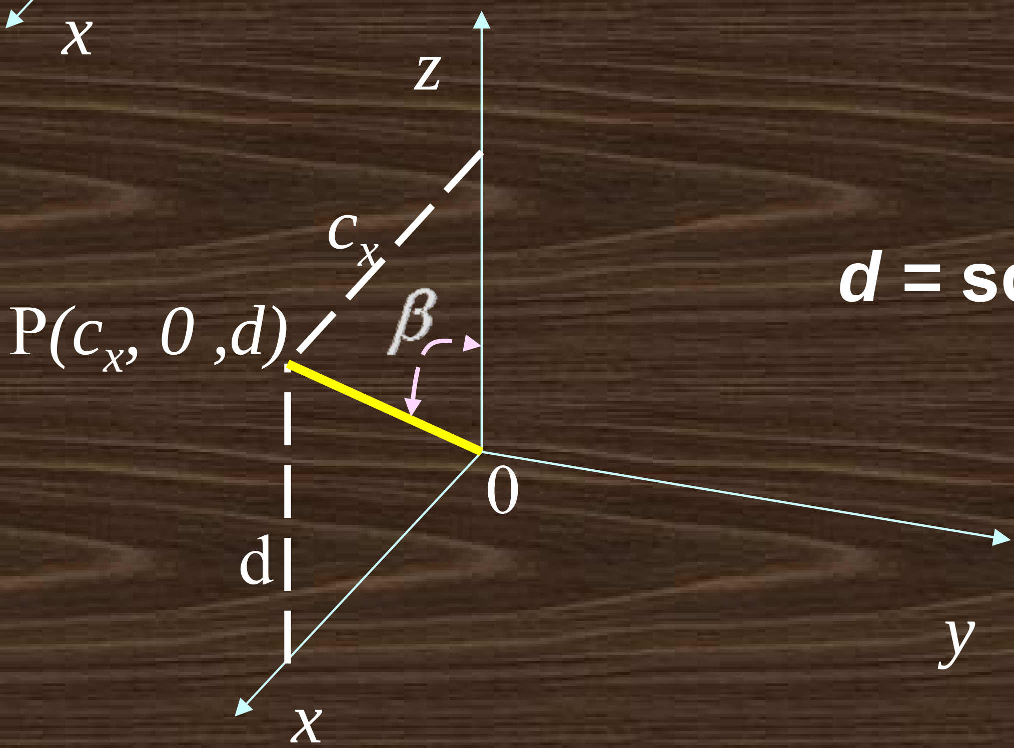


Rotation about x by α:
How do we determine α?

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} $\quad$ R_β^{-1} $\quad$ 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

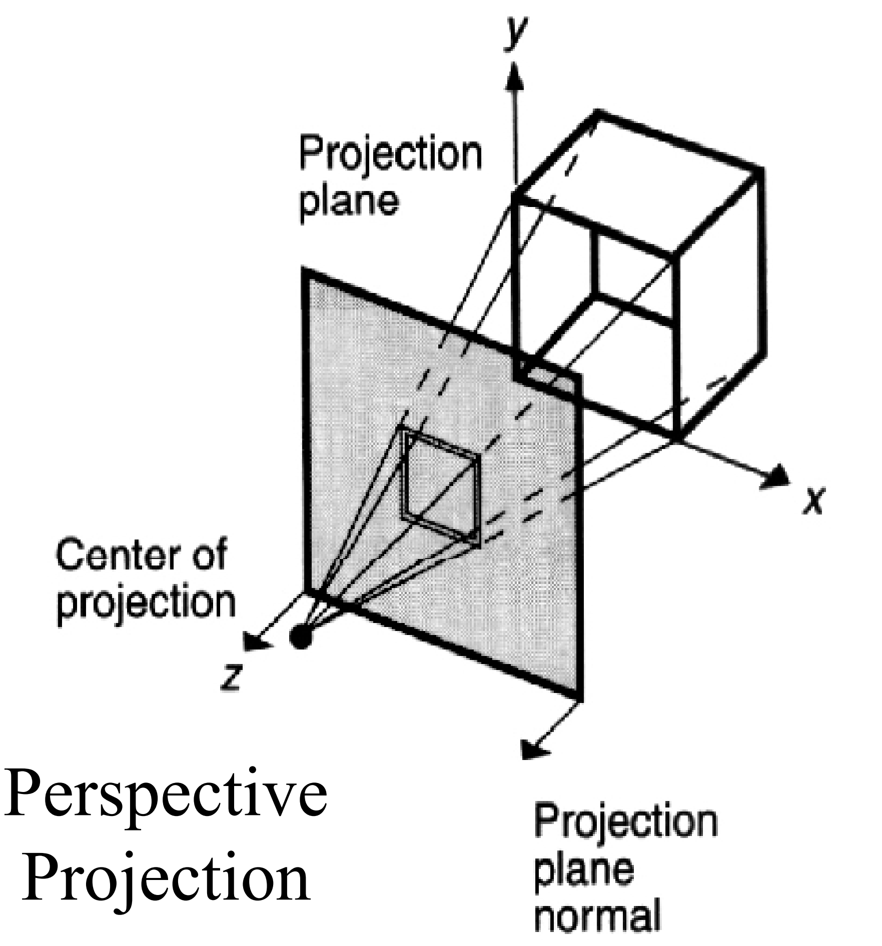
Projection of a 3D object is defined by straight projection rays (projectors) emanating from the center of projection (COP) passing through each point of the object and intersecting the projection plane.

Perspective Projections

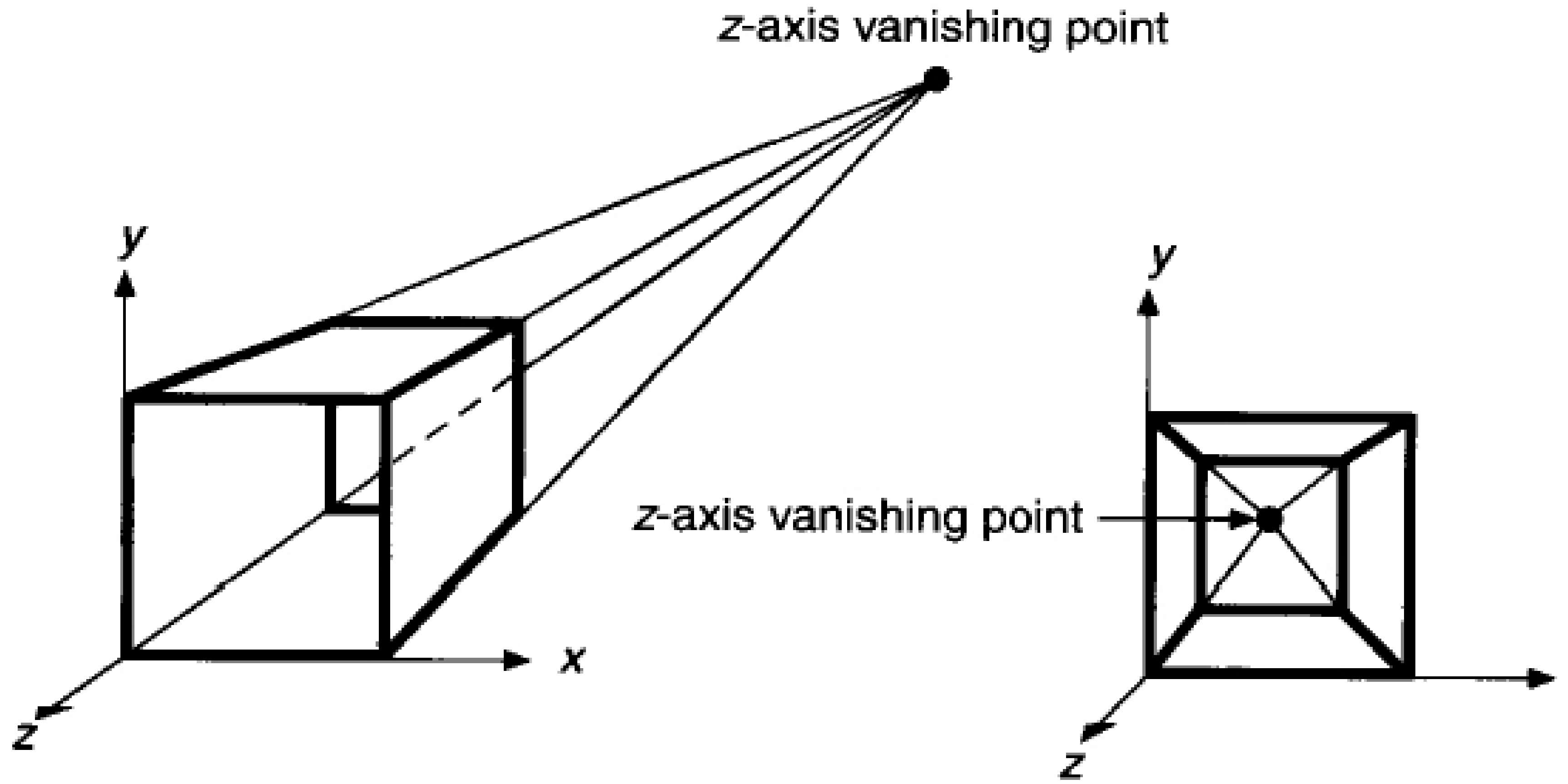
Distance from COP to projection plane is finite. The projectors are not parallel & we specify a center of projection.

Center of Projection is also called the Perspective Reference Point

COP = PRP

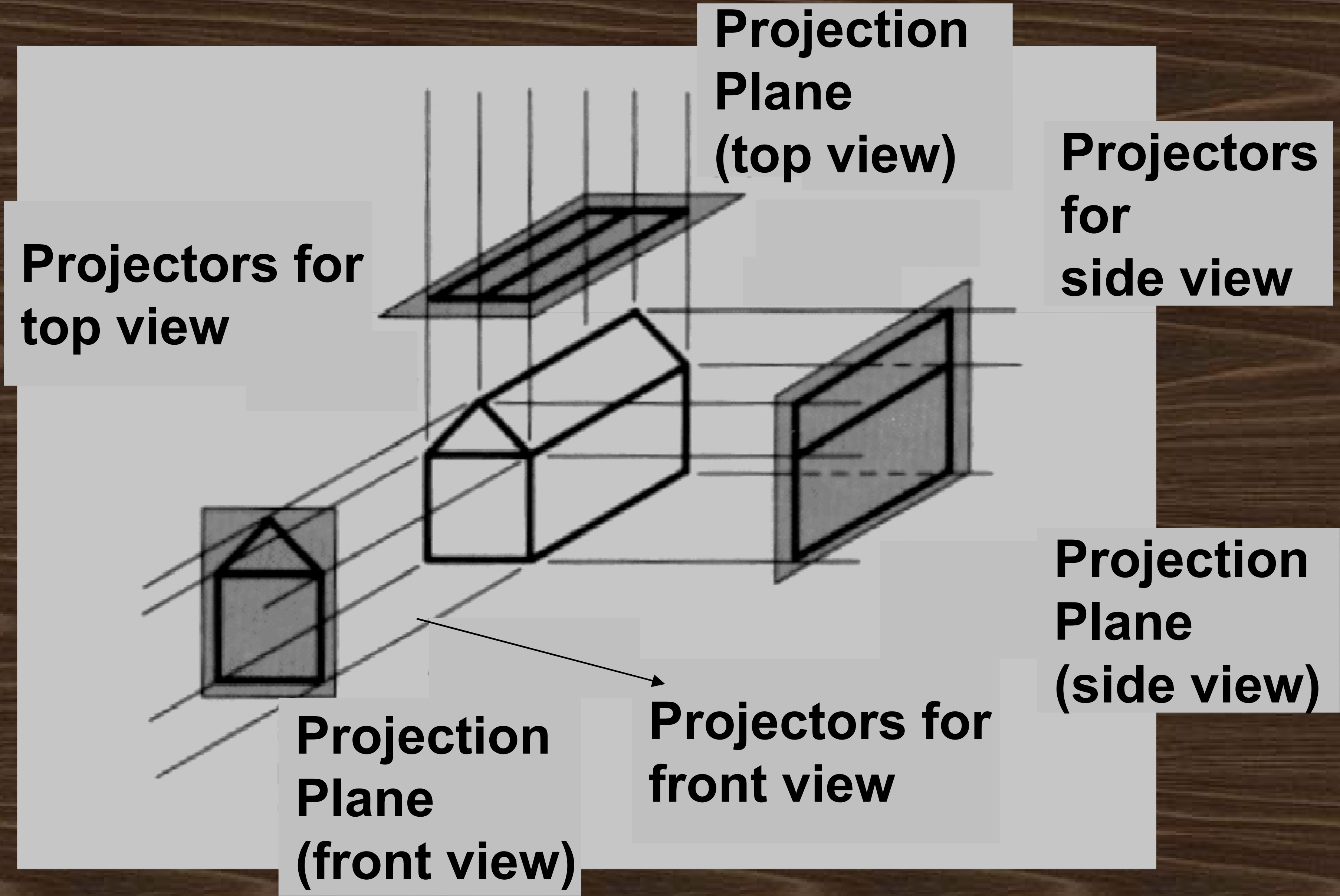


Perspective Projection



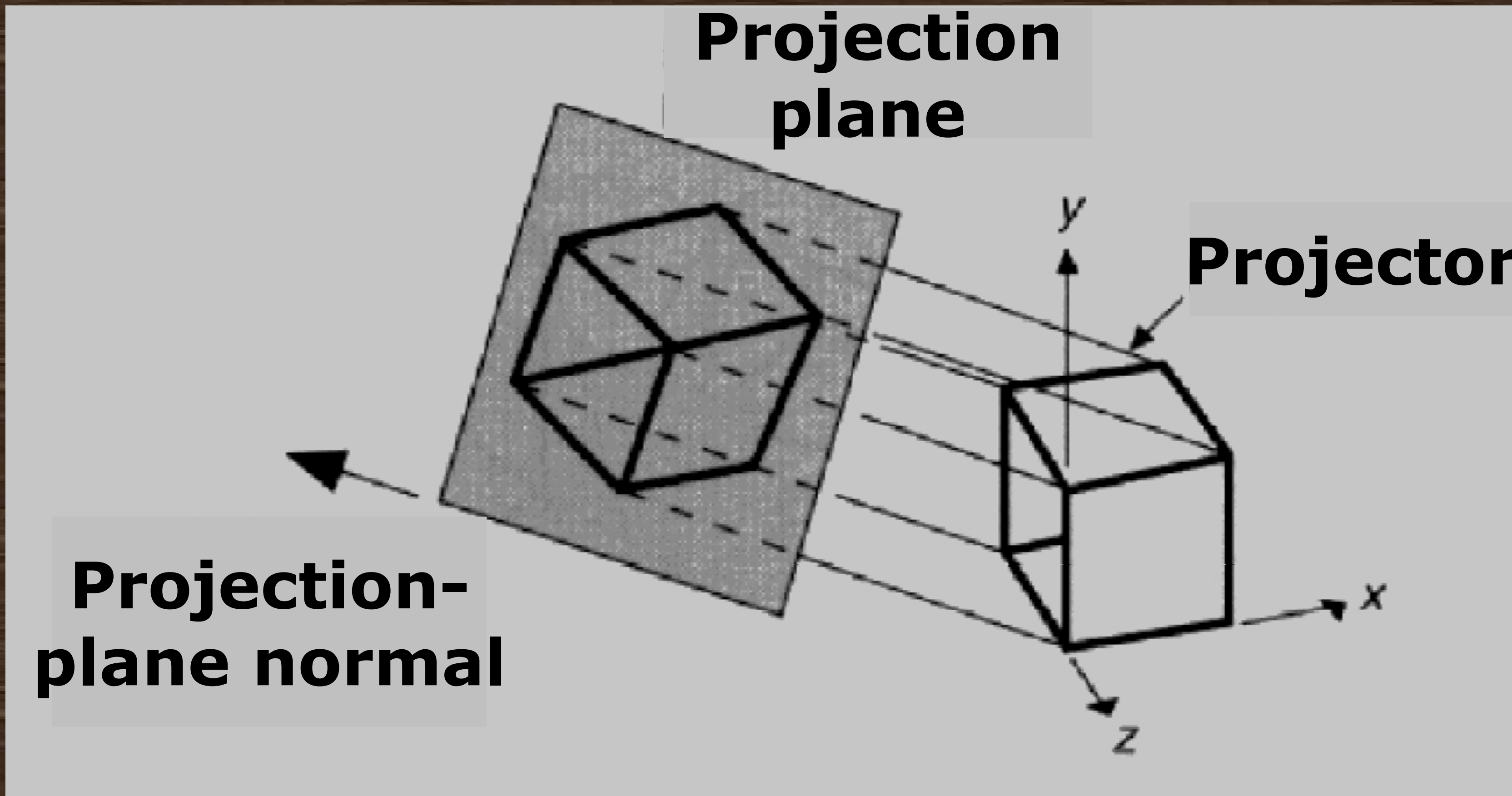
- **Perspective foreshortening:** the size of the perspective projection of the object varies inversely with the distance of the object from the center of projection.

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

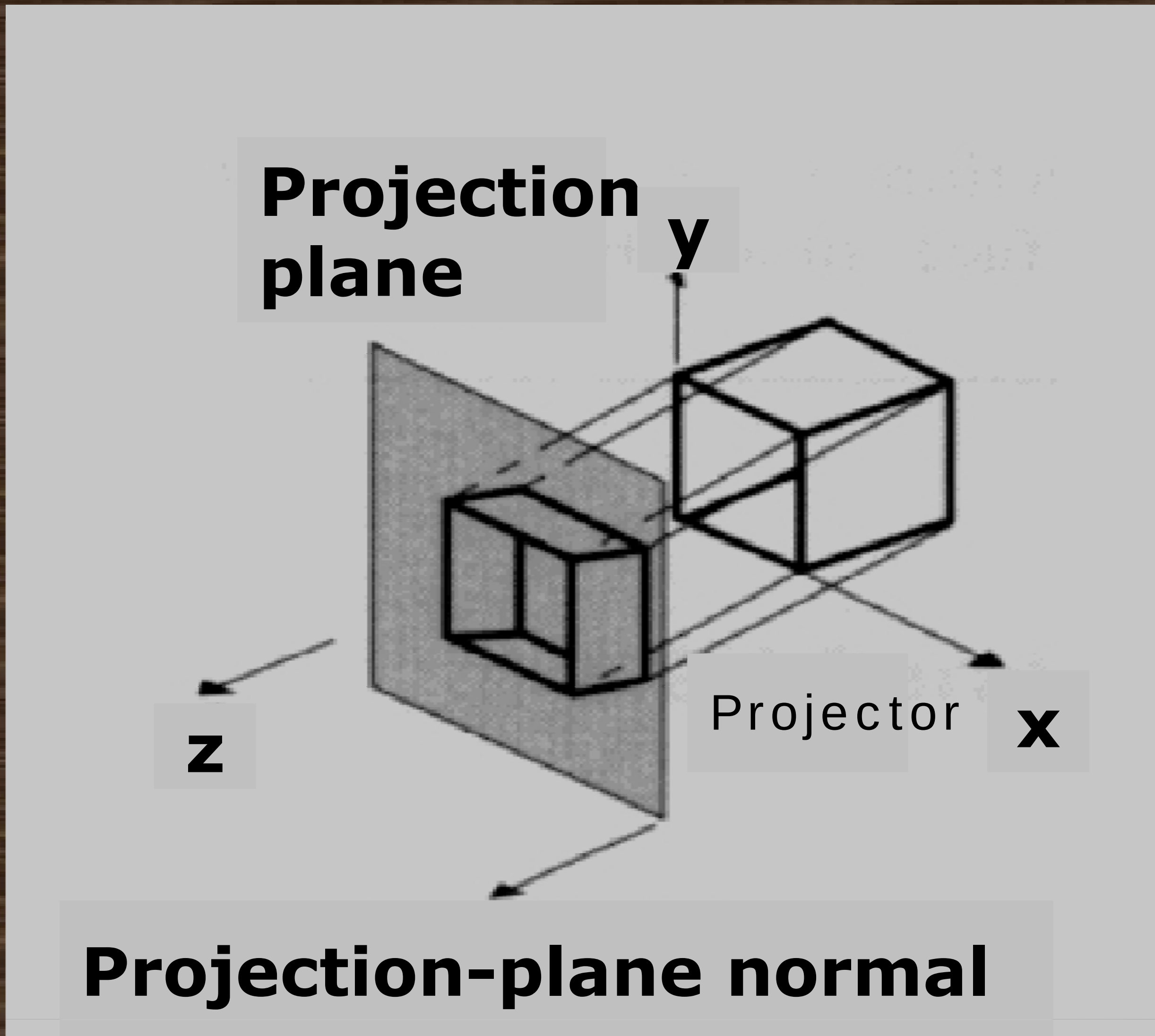


Example of Orthographic Projection

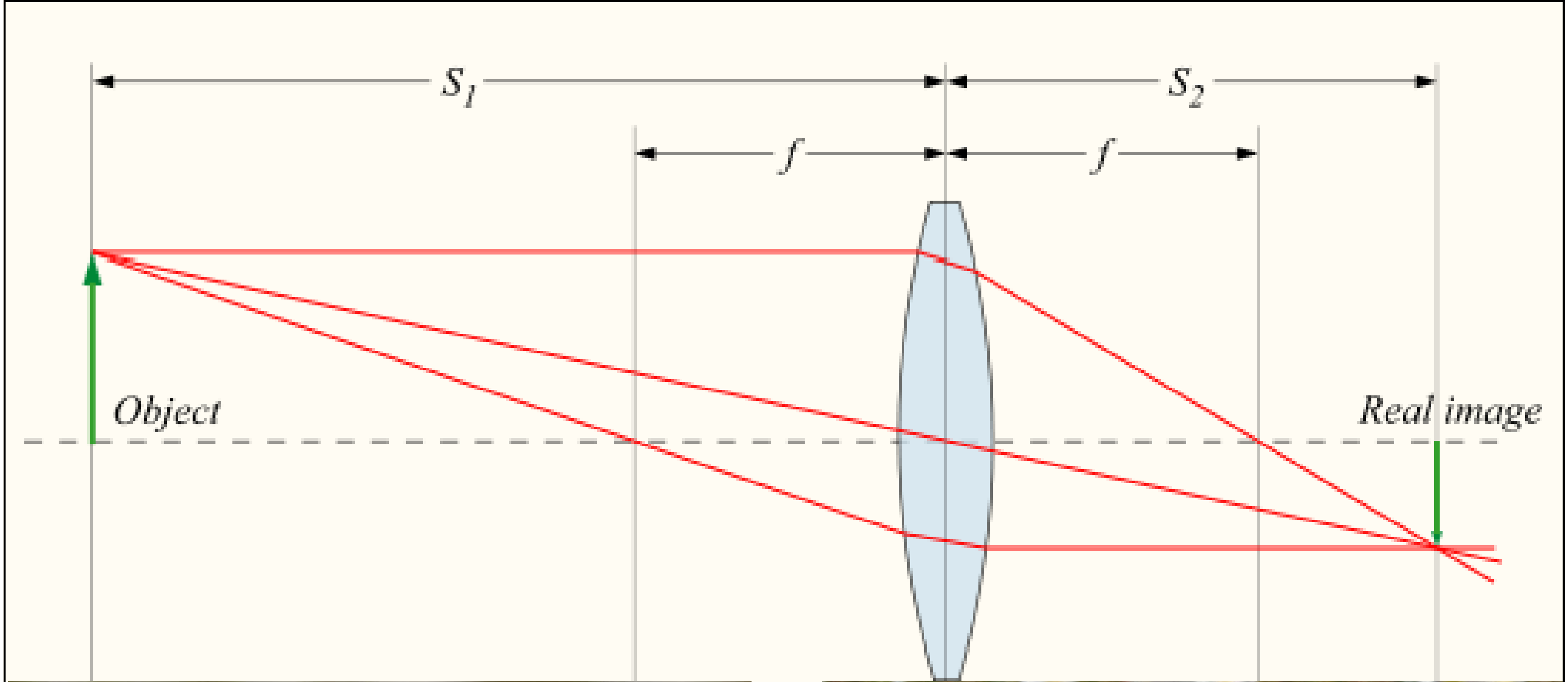
Example of Isometric Projection:



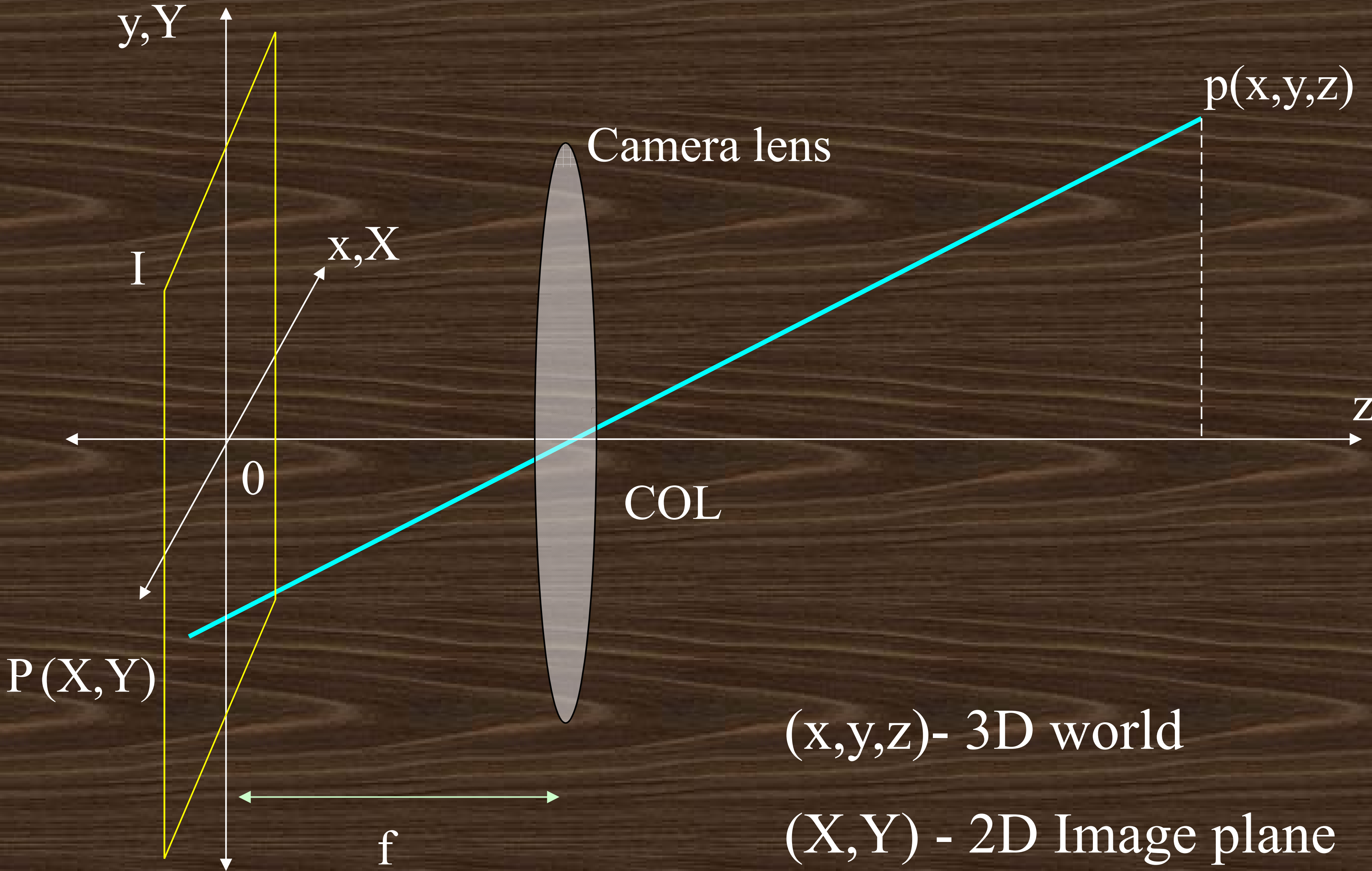
Example Oblique Projection



END OF BASICS


S_1
S_2
f
f
Object
Real image

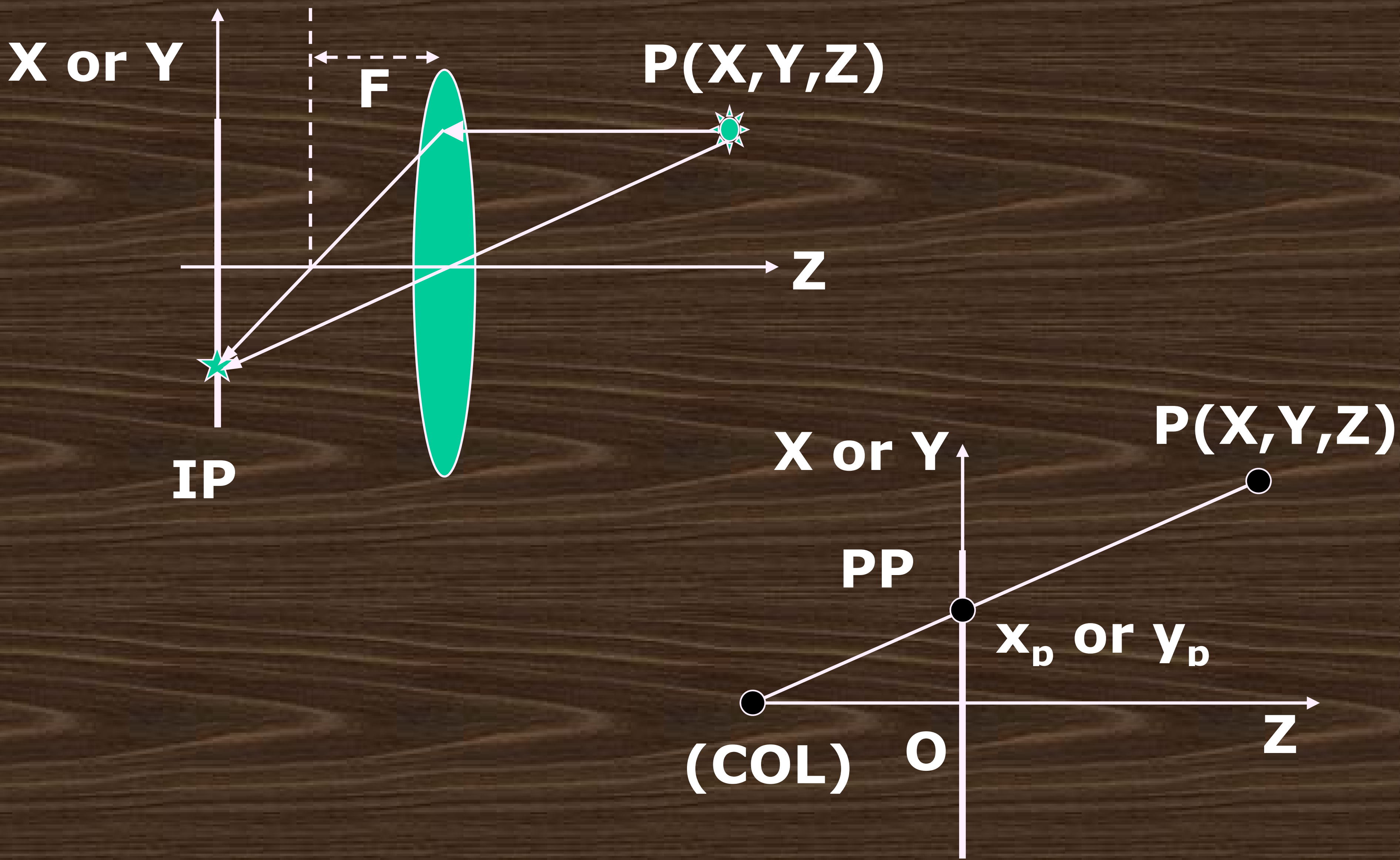
THE CAMERA MODEL: perspective projection

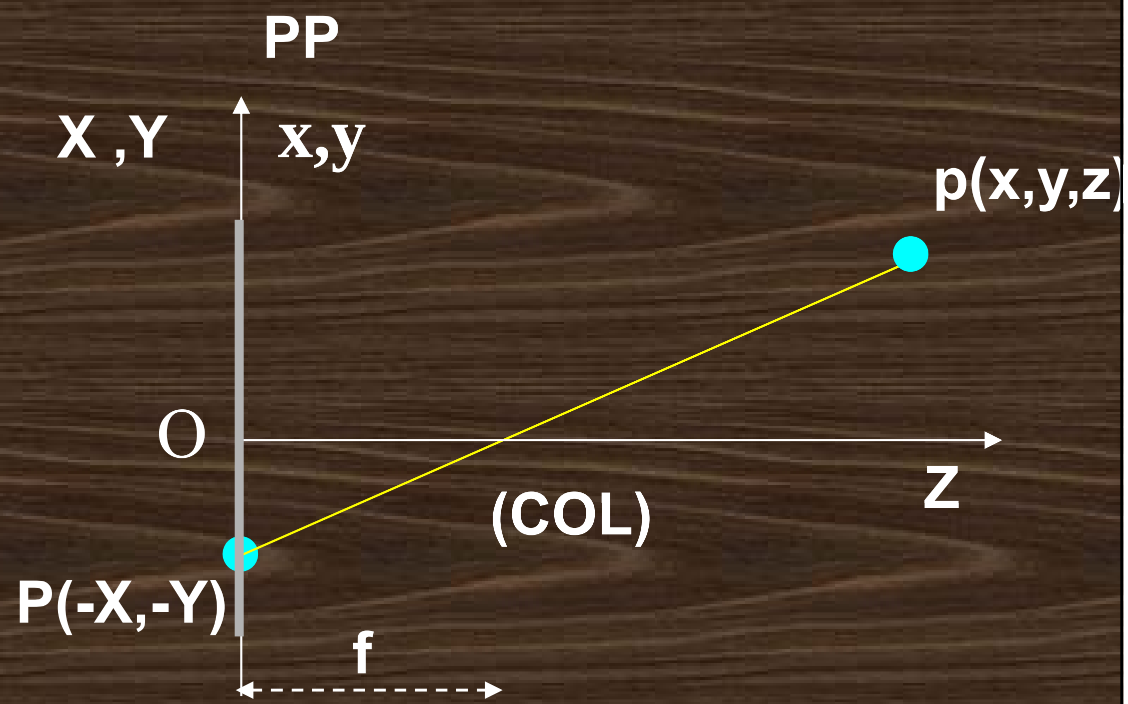


(x,y,z)- 3D world

(X,Y) - 2D Image plane

Perspective Geometry and Camera Models





CASE - 1

By similarity of triangles

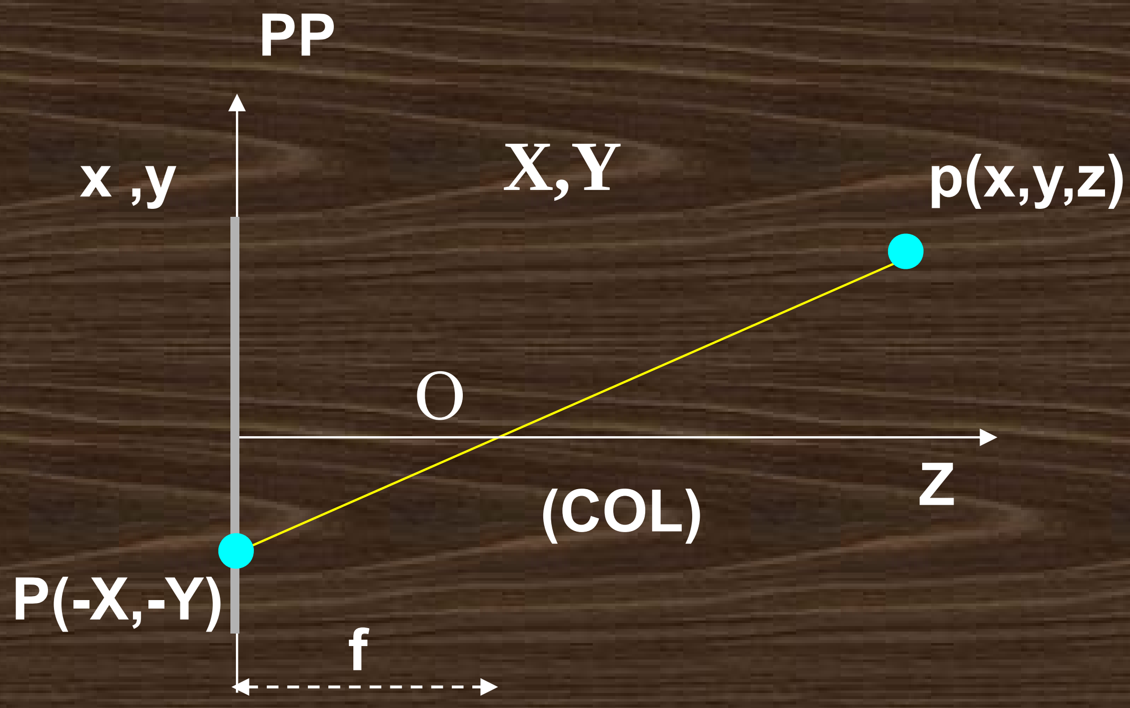
$$\frac{X}{f} = \frac{-x}{z-f}, \qquad \frac{Y}{f} = \frac{-y}{z-f}$$

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

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

- 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

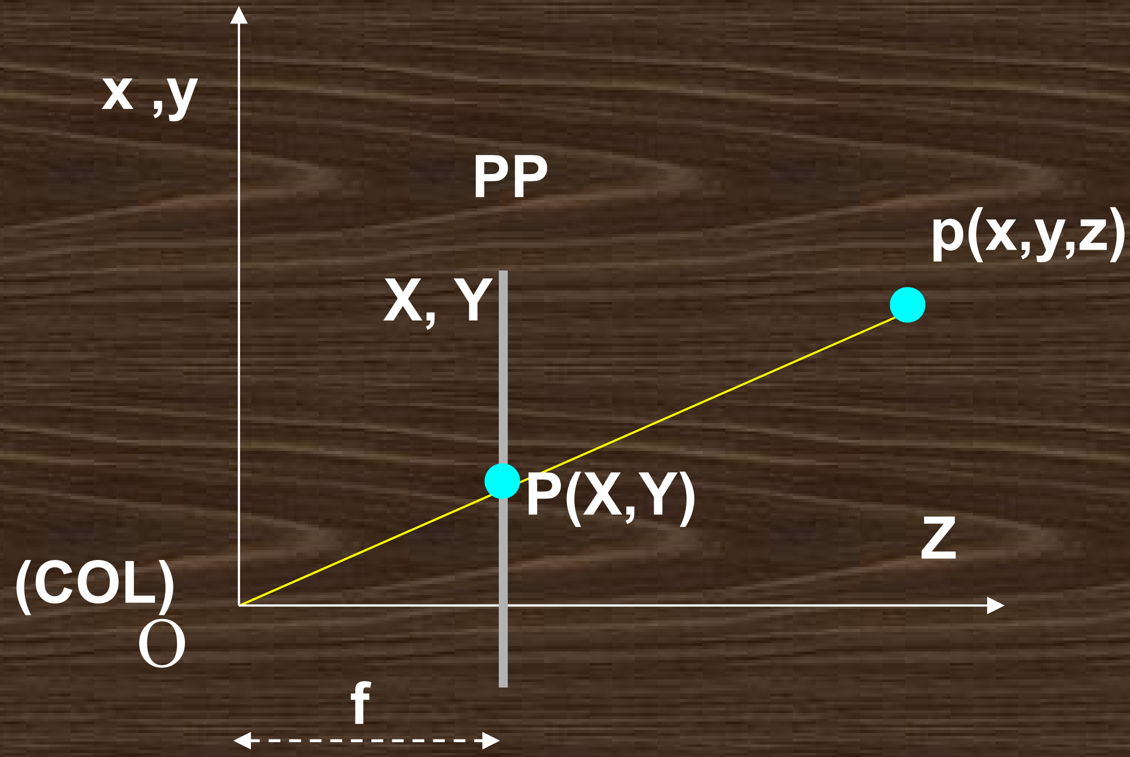
$$\frac{-X}{-f} = \frac{x}{z}, \qquad \frac{-Y}{-f} = \frac{y}{z}$$

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

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

- 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

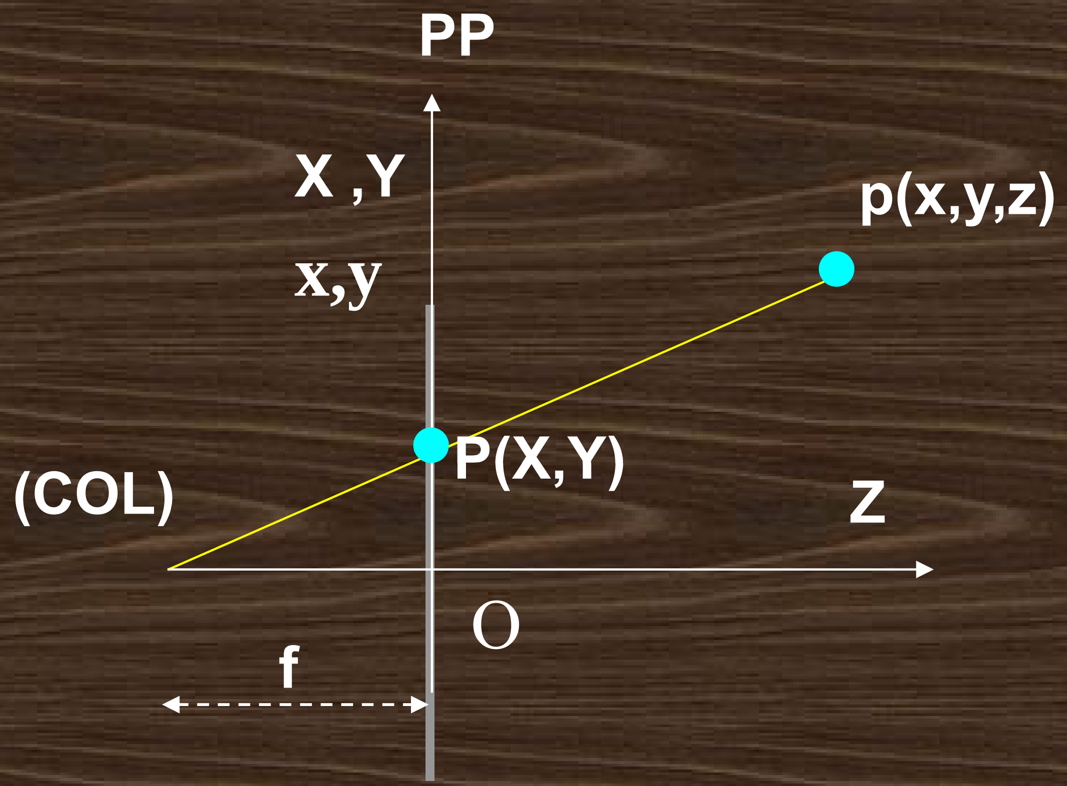
$$\frac{X}{f} = \frac{x}{z}, \qquad \frac{Y}{f} = \frac{y}{z}$$

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

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

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

CASE – 2.1



By similarity of triangles

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

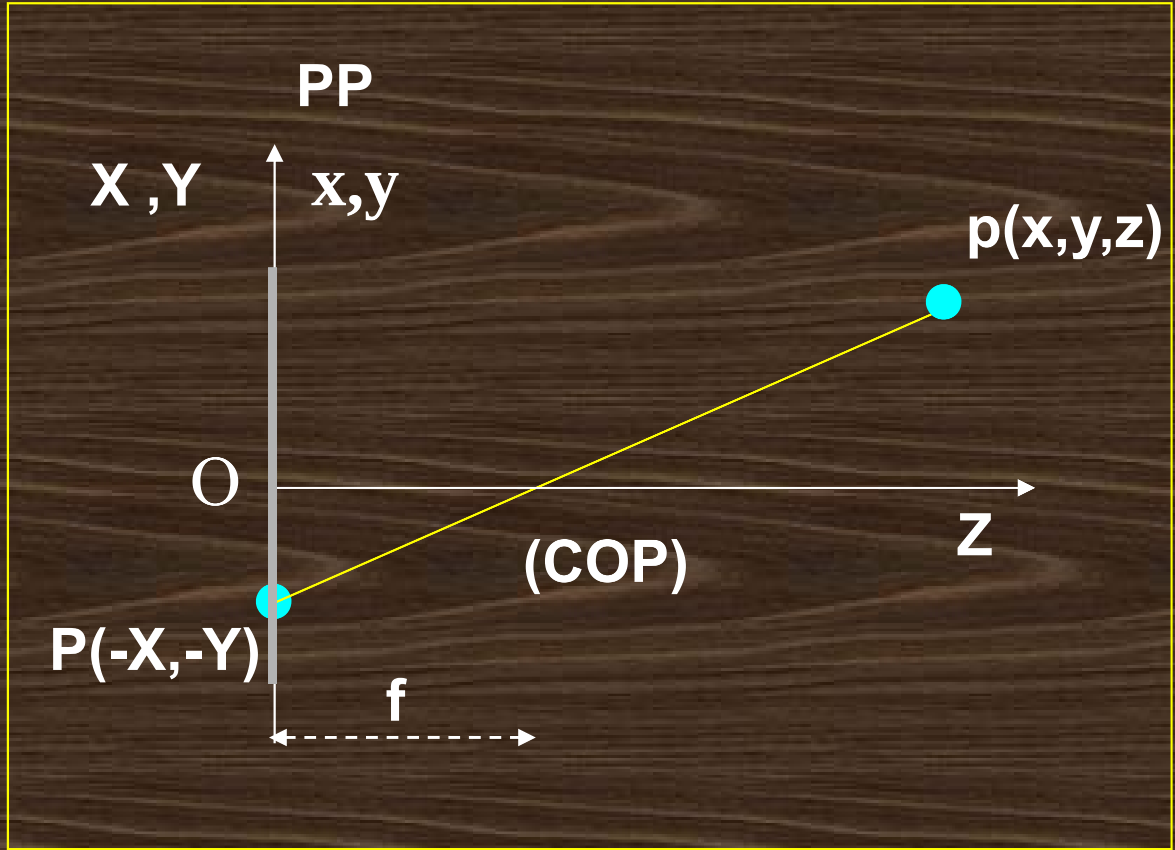
$$X = \frac{xf}{f+z}, \qquad Y = \frac{yf}{f+z}$$

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

- 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



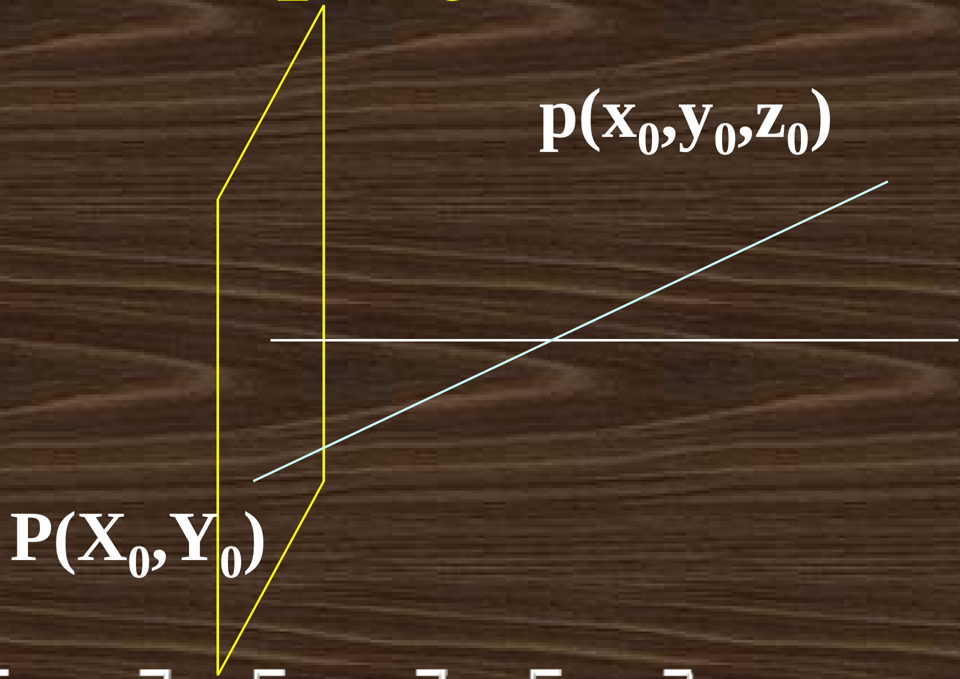
$$X = \frac{x}{1 - z/f}, \quad Y = \frac{y}{1 - z/f}$$

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

(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

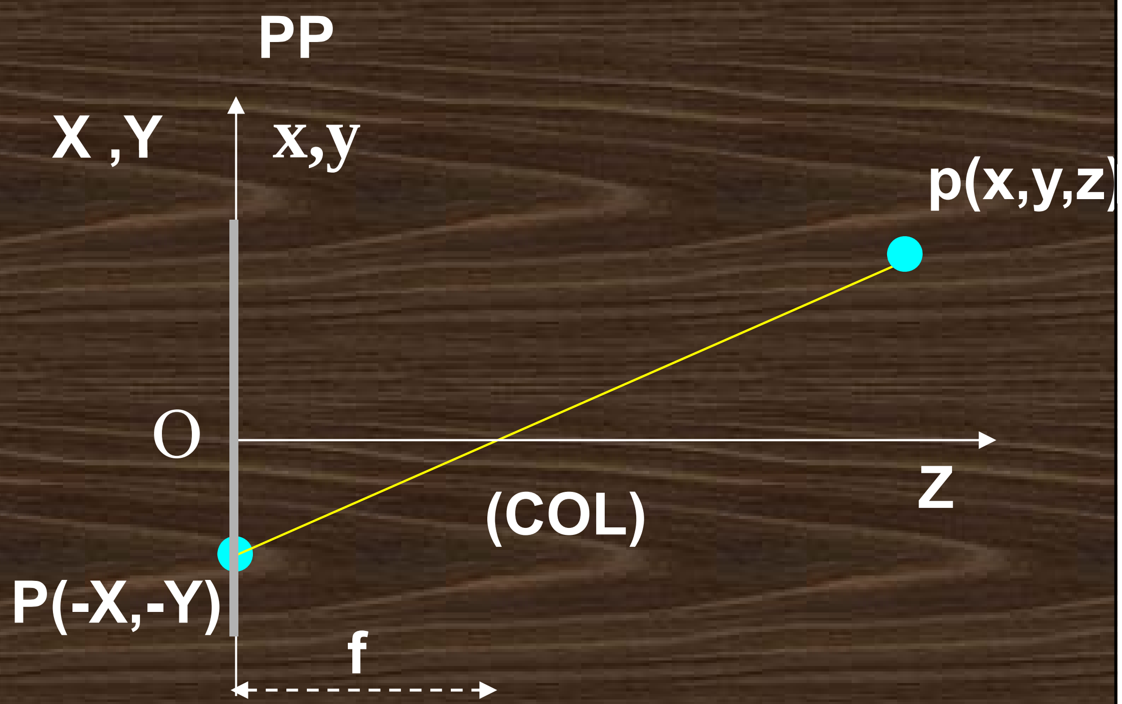
Let the image point be represented as: $[kX_0 \;\; kY_0 \;\; kZ \;\; k]^T$

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

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

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

CASE - 1



Forward: 3D to 2D

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

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

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

Inverse: 2D to 3D

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

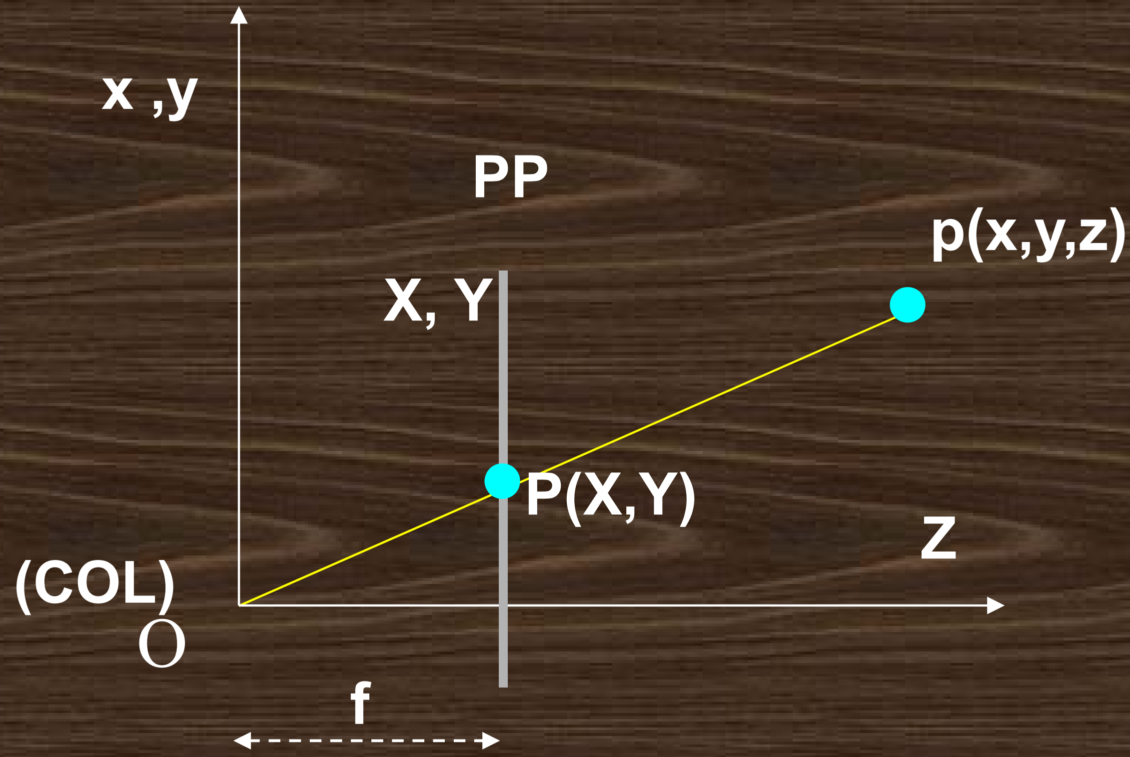
CASE - 2

Forward: 3D to 2D

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

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

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



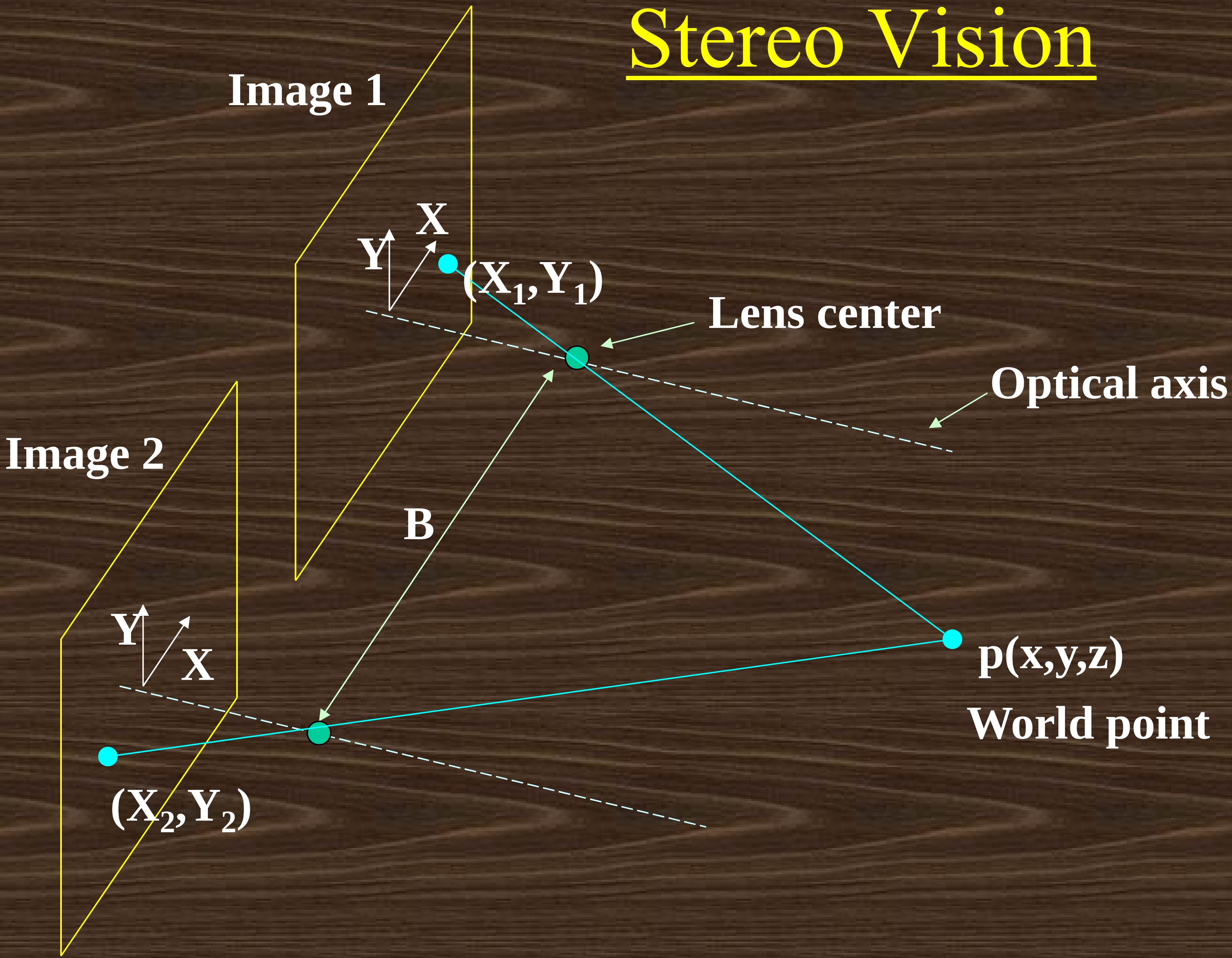
Inverse: 2D to 3D

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

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

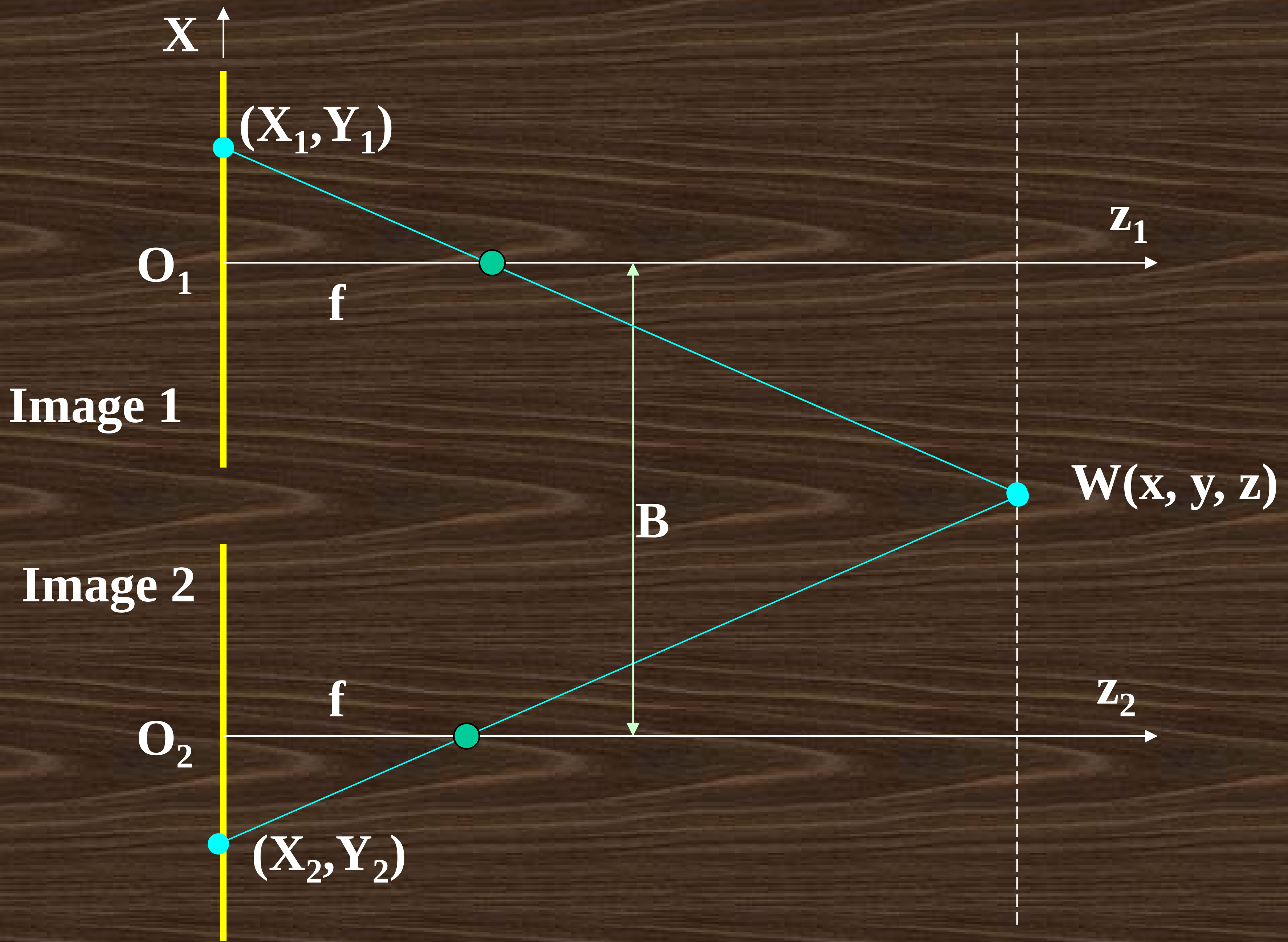
Stereo Vision



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

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

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

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

- 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

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

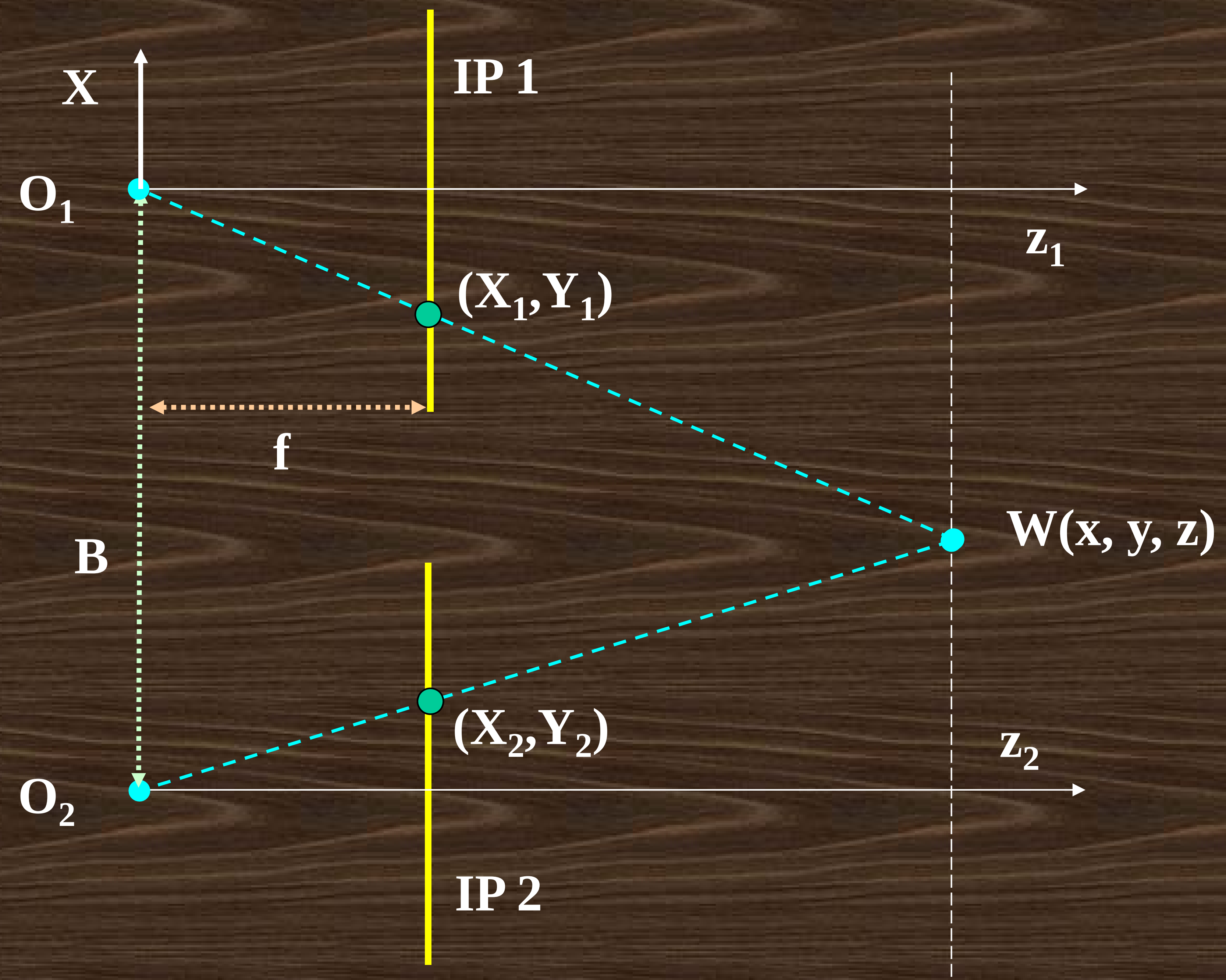
$$x = \frac{Xz}{f}, \quad y = \frac{Yz}{f}$$

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

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

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

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



Compare the two solutions

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

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

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

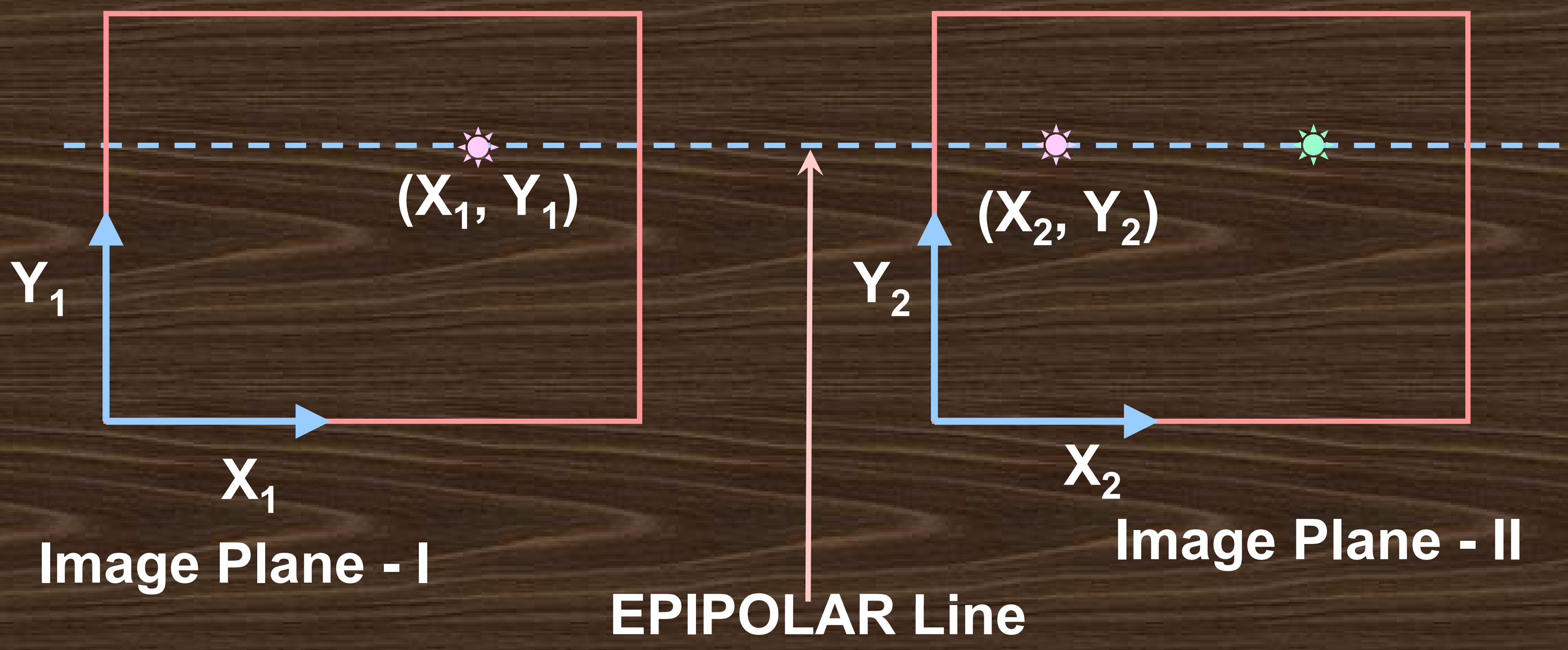
$$D = (X_2 - X_1) = \frac{fB}{z}$$

What do you think of D ?

The Correspondence Problem

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

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



Error in Depth Estimation

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

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

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

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) ?

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

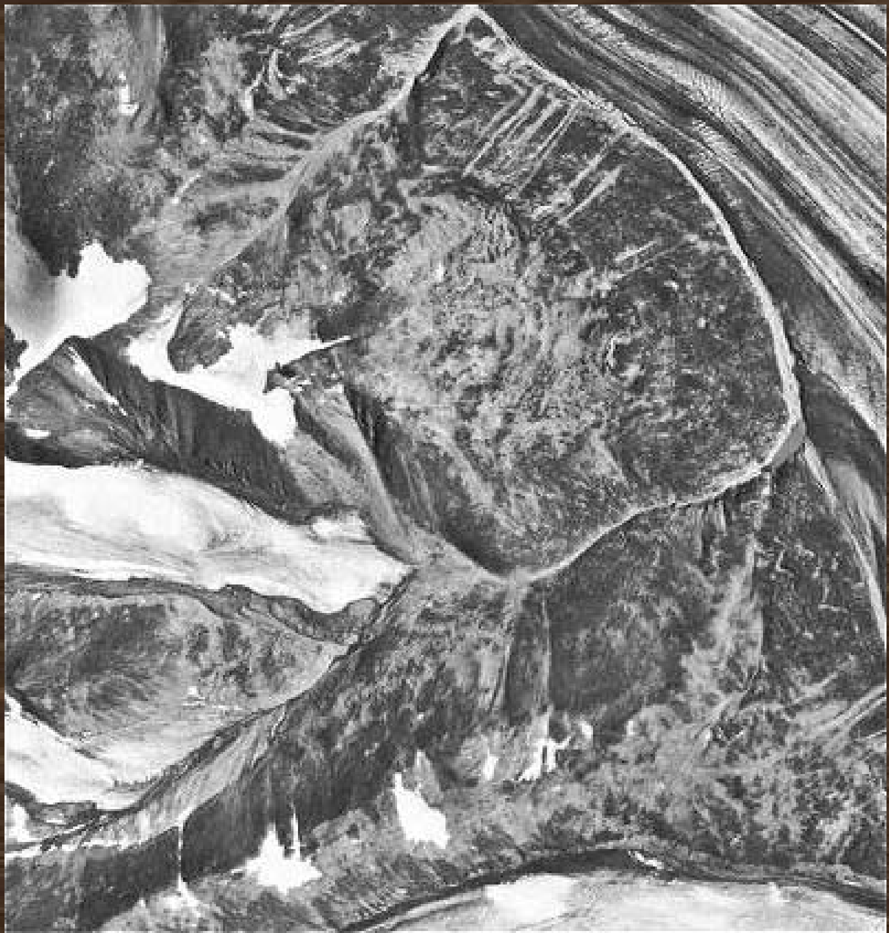
What is D_{max} ?

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

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







General Stereo Views

Perfect 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

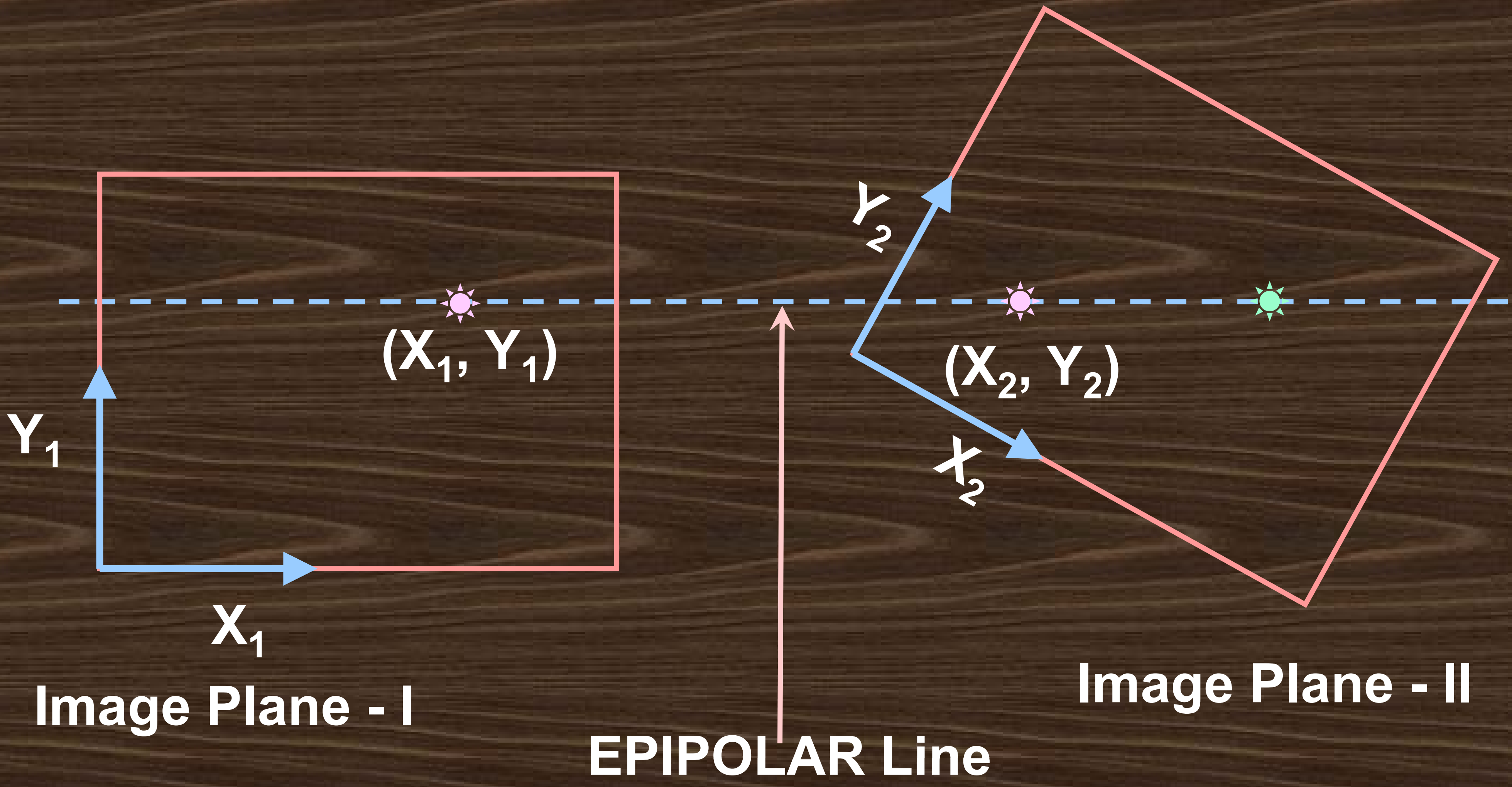
Take home exercises/problems:

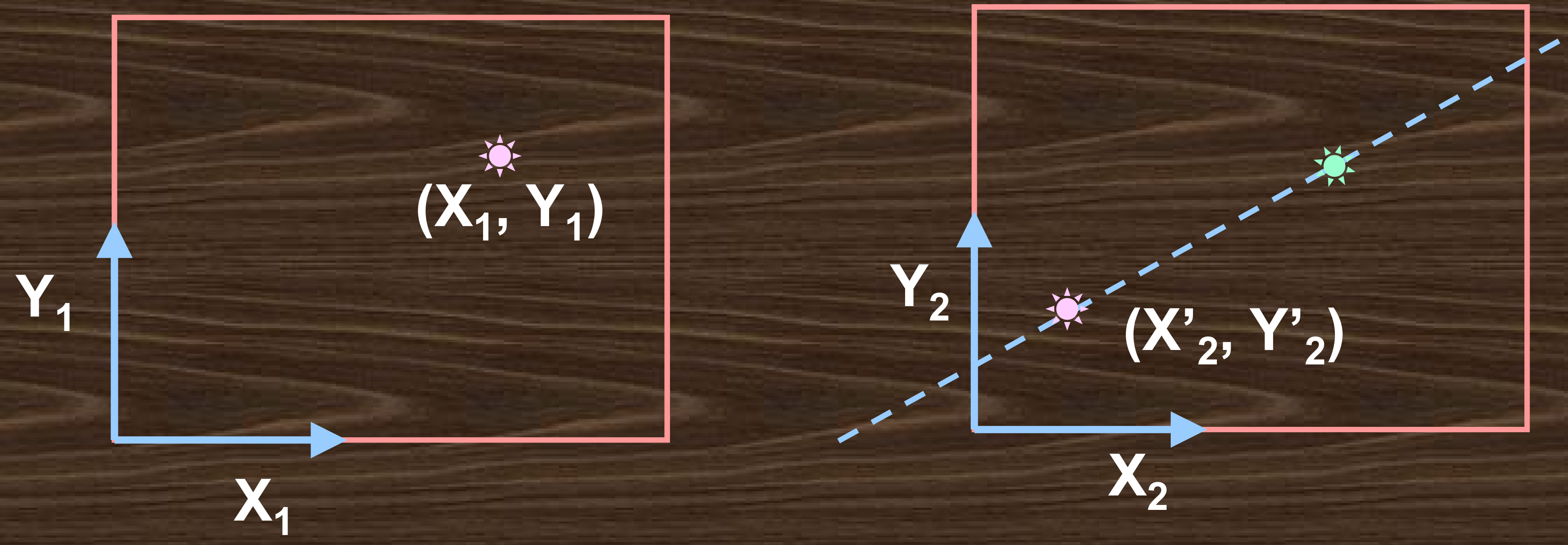
What about Epipolar line in cases above ?

How do you derive the equation of an epipolar line ?

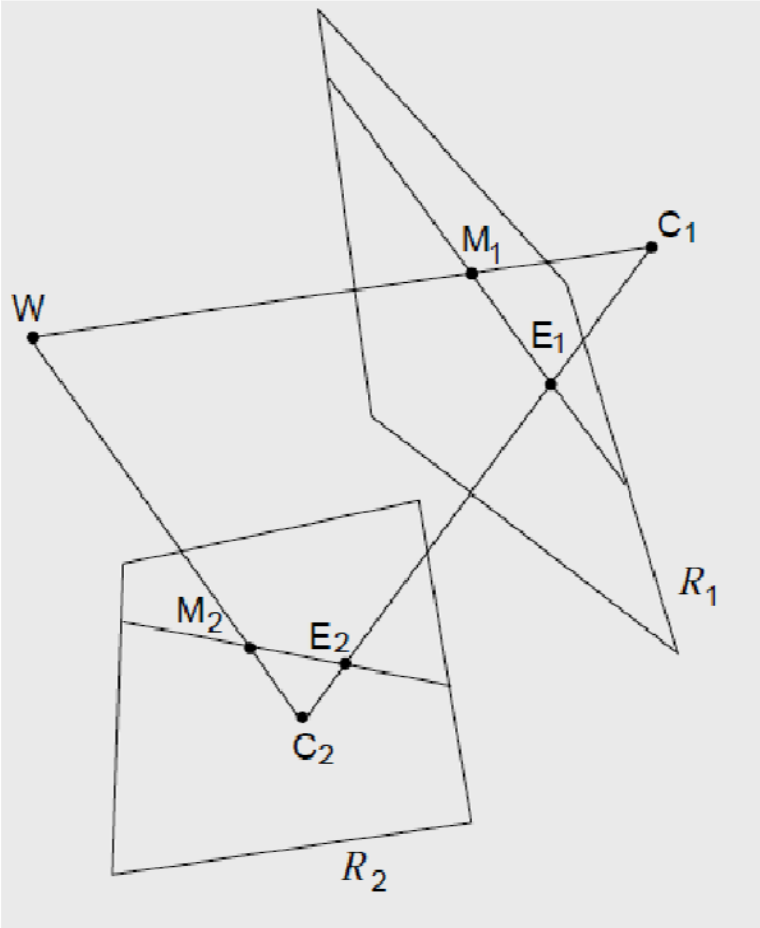
In general we may have multiple views (2 or more) of a scene. Typically used for 3D surveillance tasks.

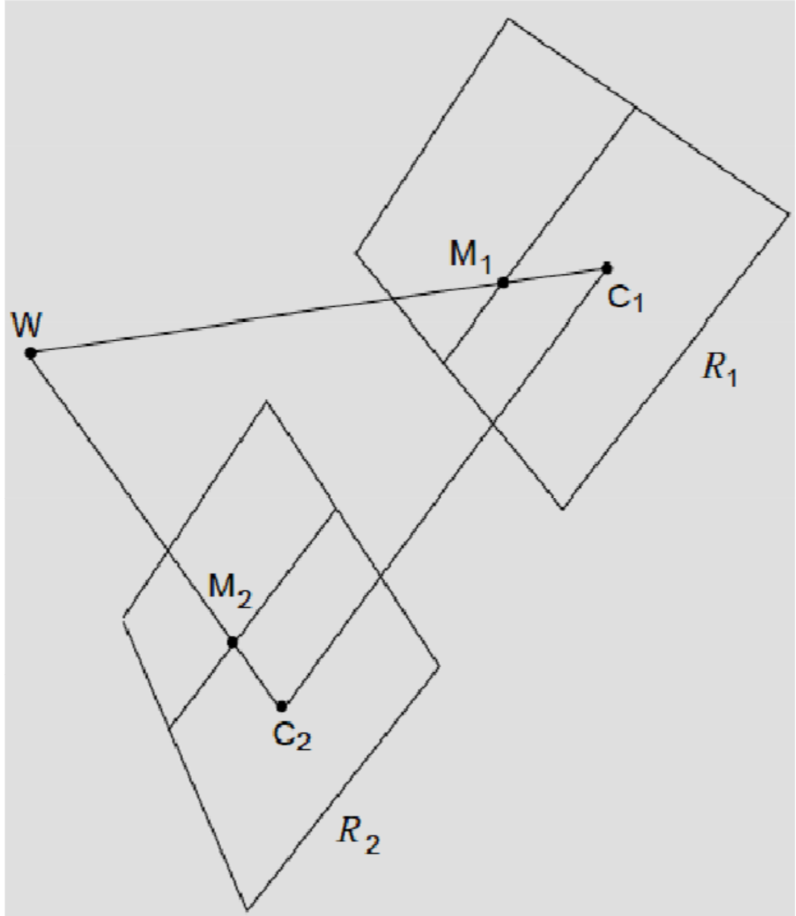
The Epipolar line in case of Arbitrary Views





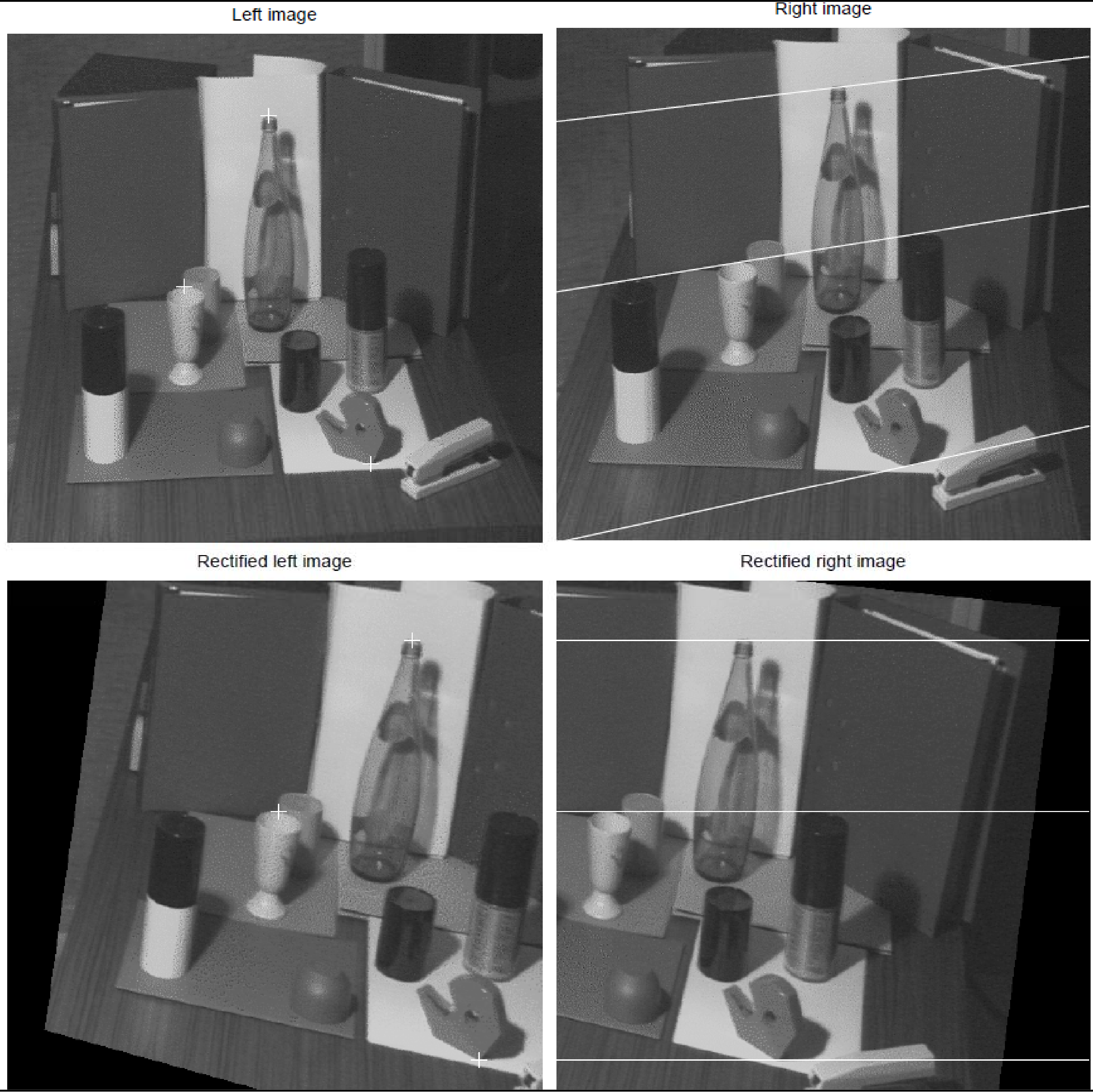
(X1, Y1)
Y1
X1
Y2
(X'2, Y'2)
X2

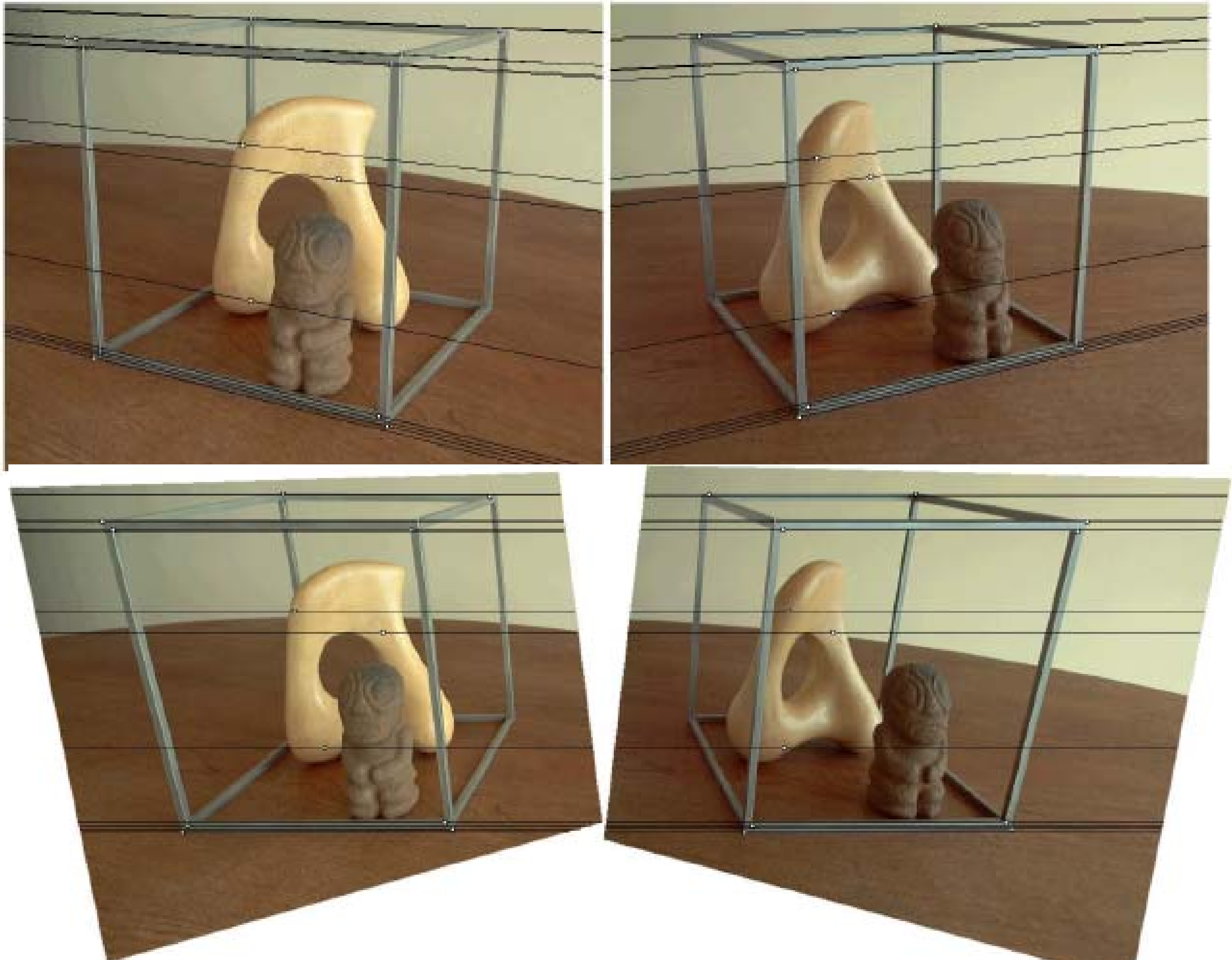




Process of Rectification

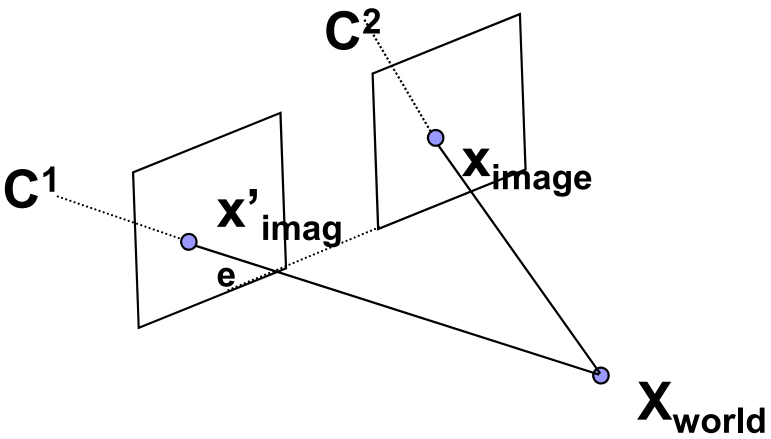
Image rectification is the process of applying a pair of 2 dimensional projective transforms, or homographies, to a pair of images whose epipolar geometry is known so that epipolar lines in the original images map to horizontally aligned lines in the transformed images.


Left image
Right image
Rectified left image
Rectified right image

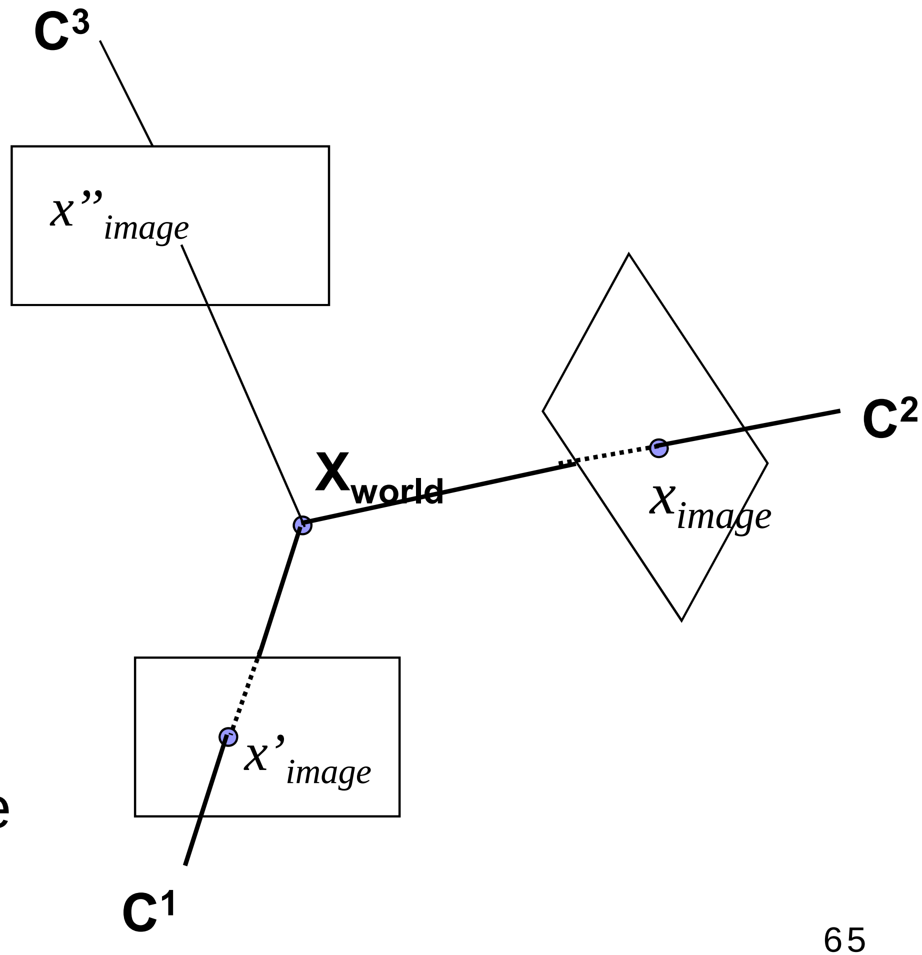


Classical Depth Estimation

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



General Stereo

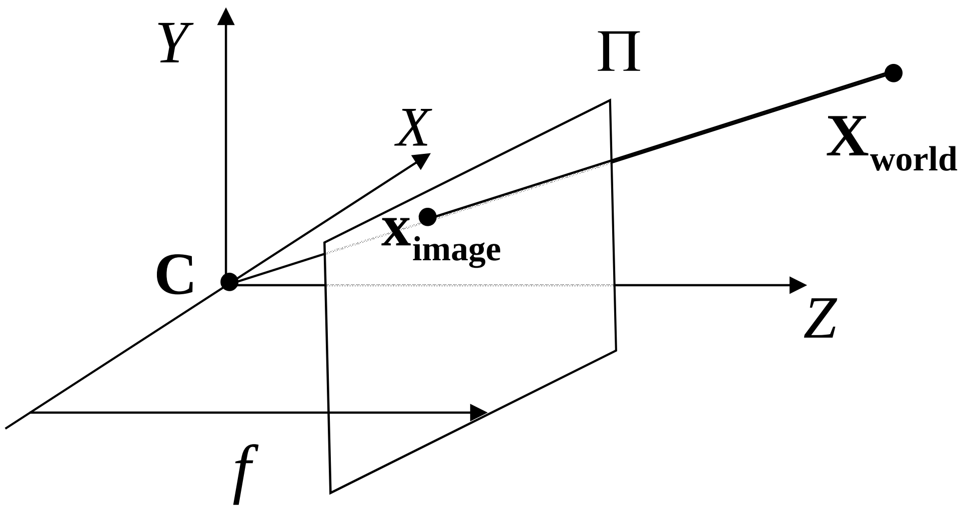


Arbitrary multiple view geometry

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.

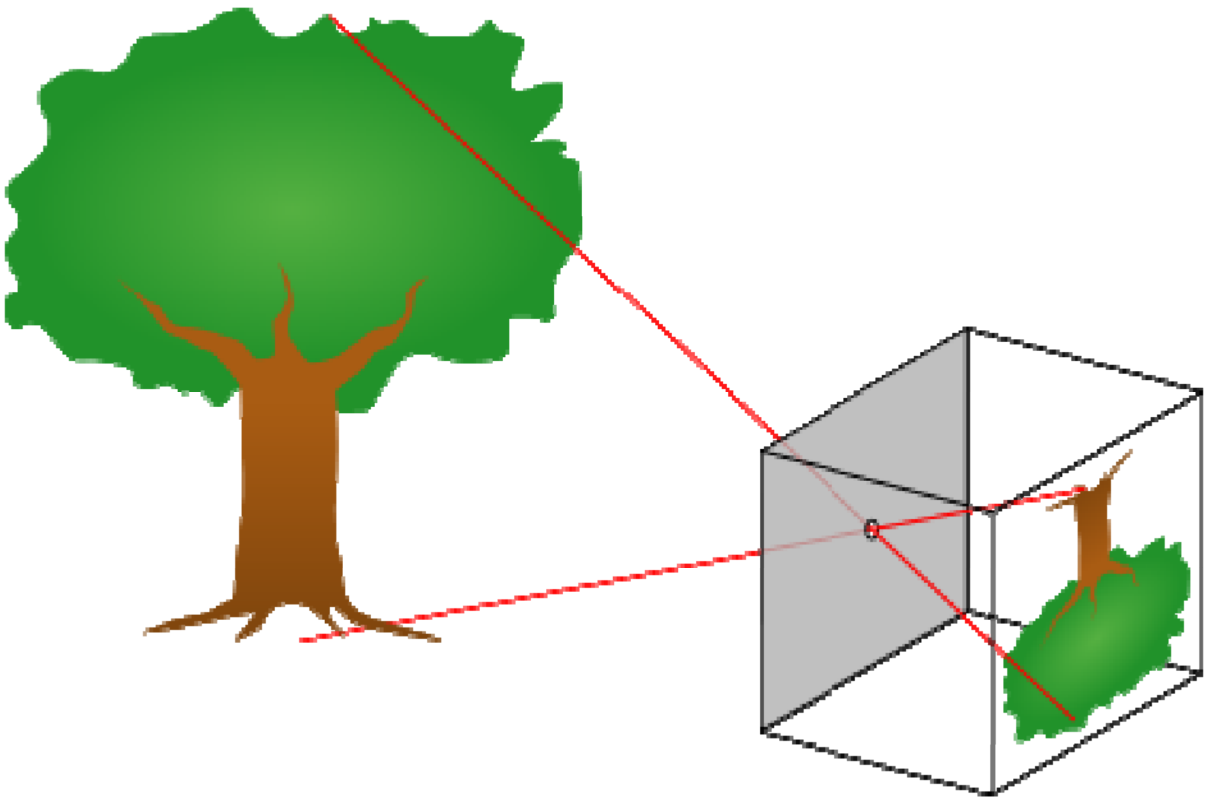
$$F:(X_w, Y_w, Z_w) \rightarrow (x_i, y_i)$$



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

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

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



Pinhole Camera schematic diagram

Camera Geometry

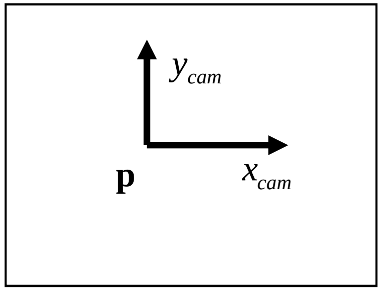
- Camera can be considered as a projection matrix, $\mathbf{x} = \mathrm{P}_{3*4}\mathbf{X}$
  - A pinhole camera has the projection matrix as

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

  - Principal point offset

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

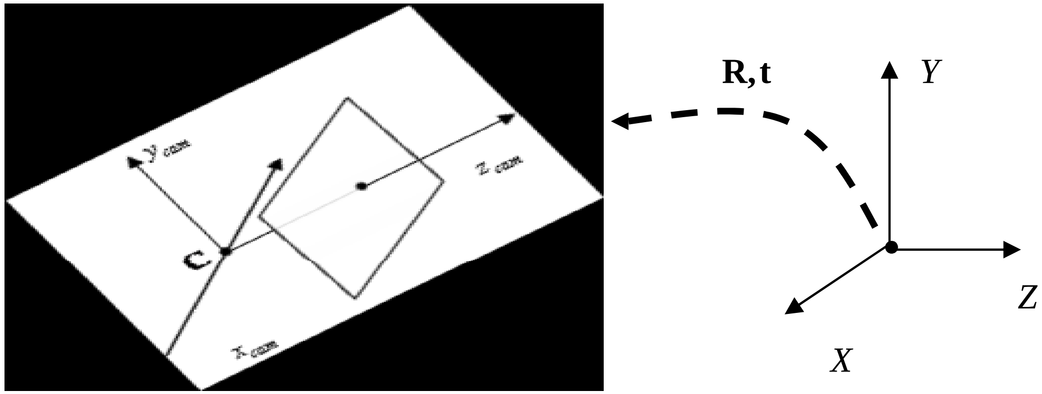
$$K = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad \mathbf{x} = \mathrm{K}[\mathrm{I} \quad | \quad \mathbf{0}]\mathbf{X}$$



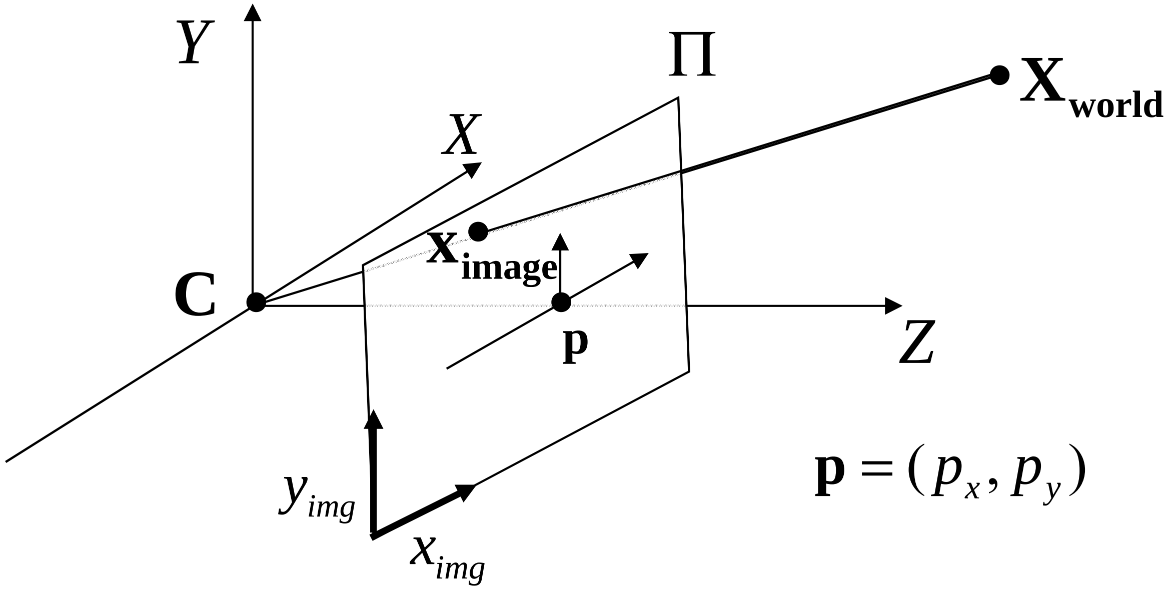
$$\mathbf{p} = (p_x, p_y)$$

  - Camera with rotation and translation

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



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

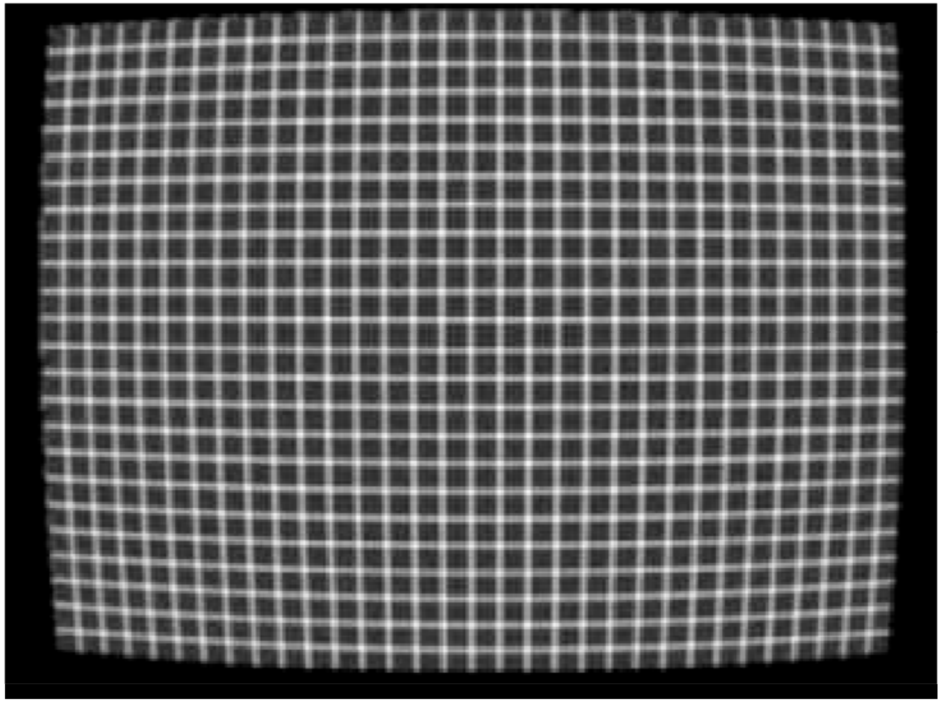
R Rotation

$\mathbf{t}$ 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

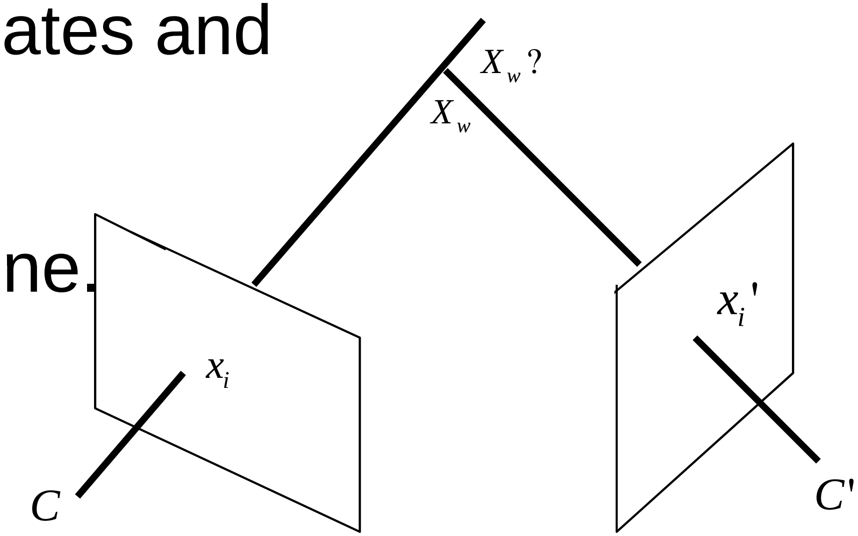
- Given a set of images of a particular 3D scene, can we reconstruct the scene back?
- 3D representation of an object is difficult because of the problem of depth estimation.
- Image is projection of 3D object on a 2D plane.

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

(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?

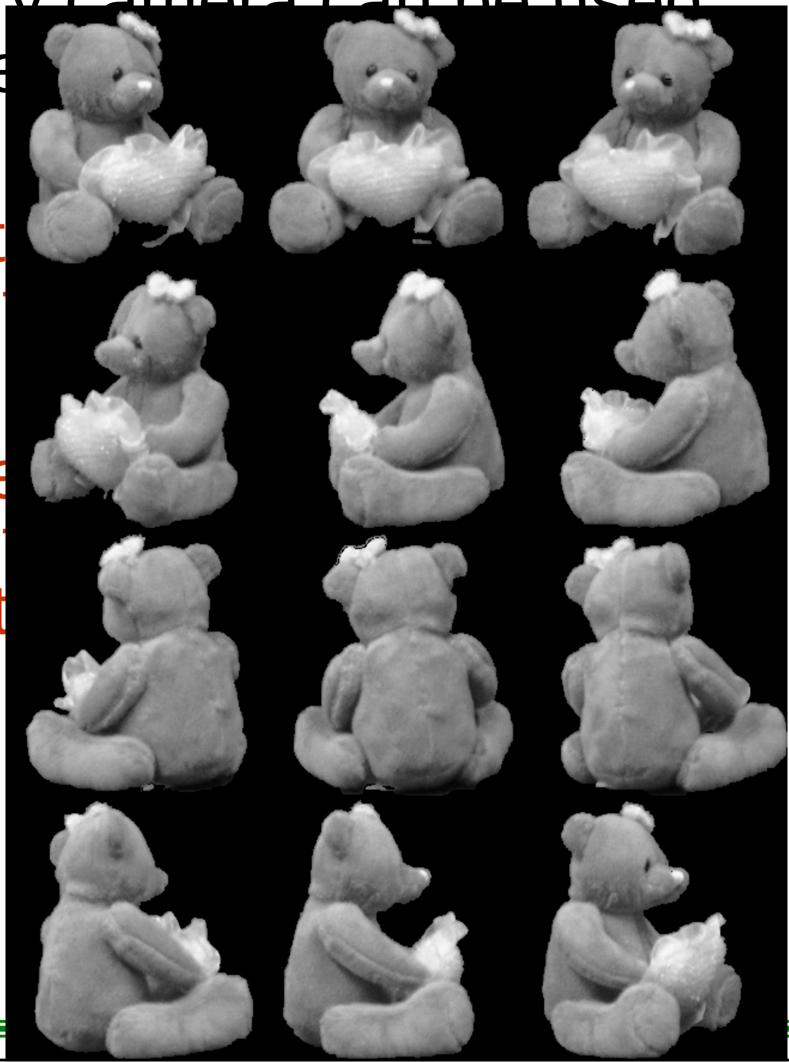
[a]

- Classical inverse problem of the computer vision

[a]. Oxford Keble College

Reconstruction from turntable sequence

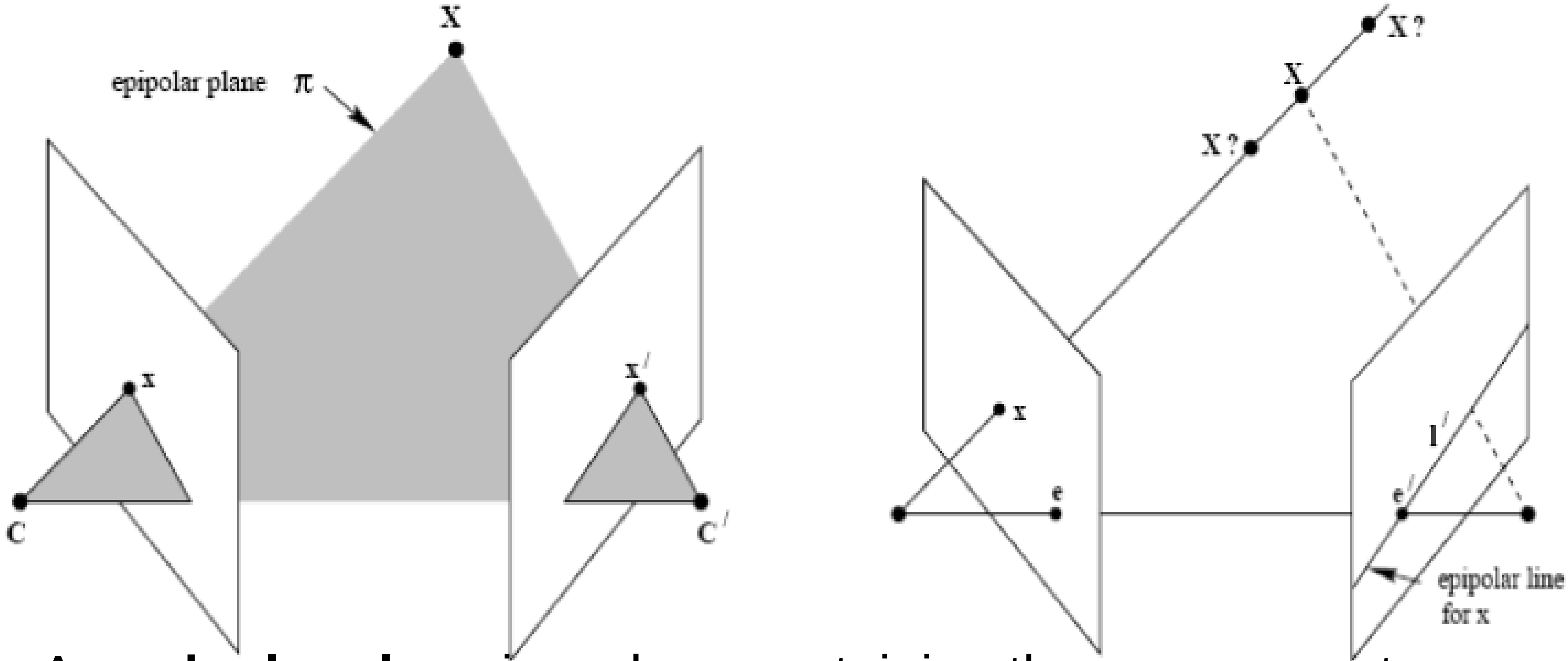
- The images acquired from various poses using an ordinary camera can be used to gene



- How st
recons

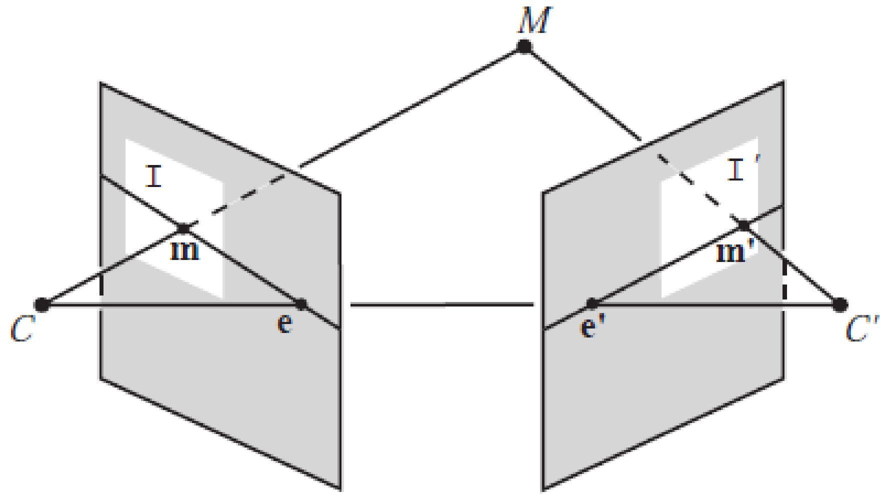
- Is there
where
better t

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



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

Point m in I, produces a line, l' (=Fm), as: $m'^T l' = 0;$

Epipoles satisfy: $Fe = \vec{0} = F^T e';$

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

$$\boxed{P' = K'[R \mid \mathbf{t}]}$$

Read Camera Calibration methods for P estimation

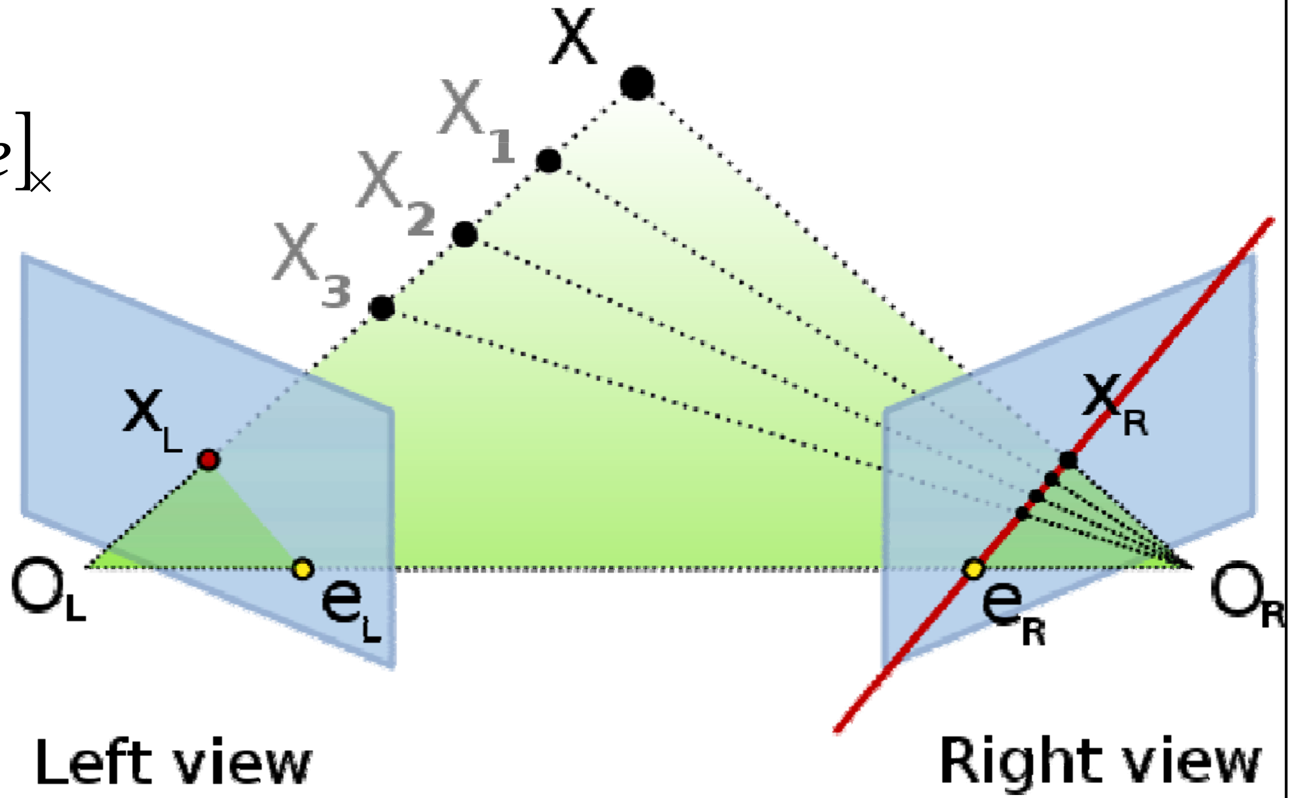
$$F = K'^{-T} RK^T [KR^T t]_\times$$

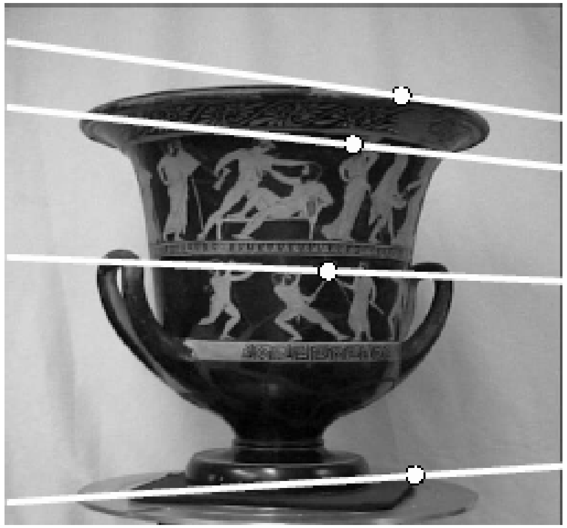
$$= [e']_\times K' RK^{-1} = K'^{-T} RK^T [e]_\times$$

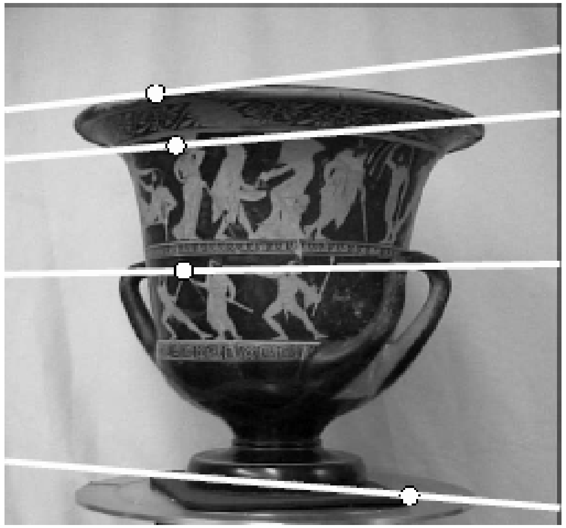
$$\mathbf{m}'^T\mathbf{F}\mathbf{m} = 0,$$

$=>$

$$Af = 0;$$







Typical methods used to estimate F:

- 8-pt DLT algo.

- RANSAC

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

$$\Rightarrow$$

$$Af = 0;$$

- **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

Scene Homography (points)

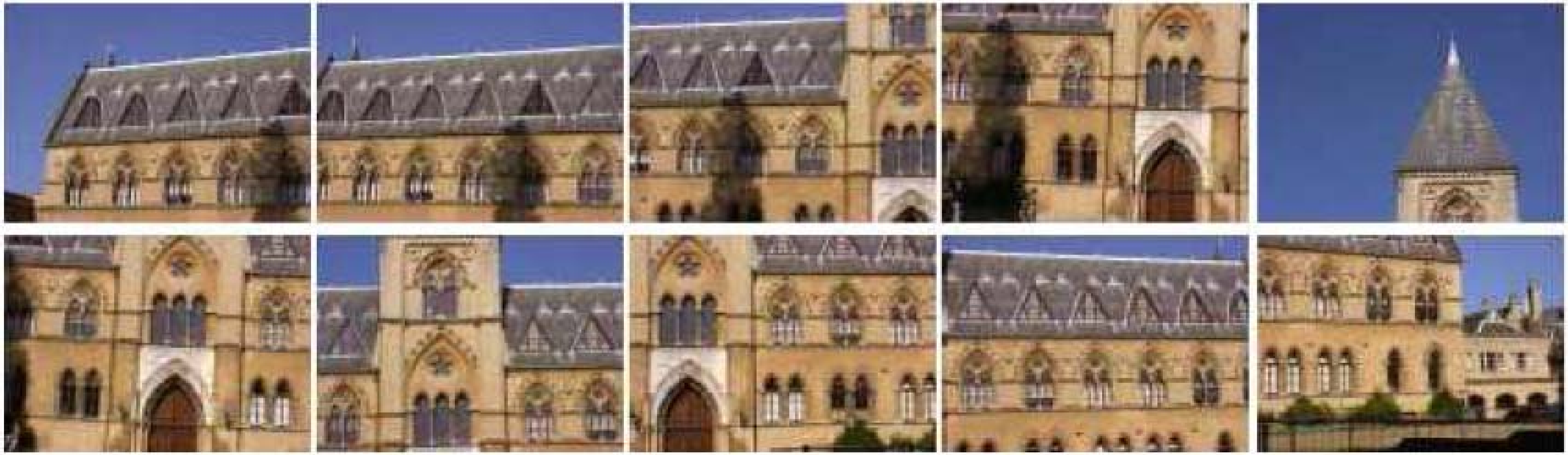
A **homography** is an invertible mapping of points and lines on a projective plane. Its an invertible mapping to itself, such that collinearity is preserved. It is represented as:

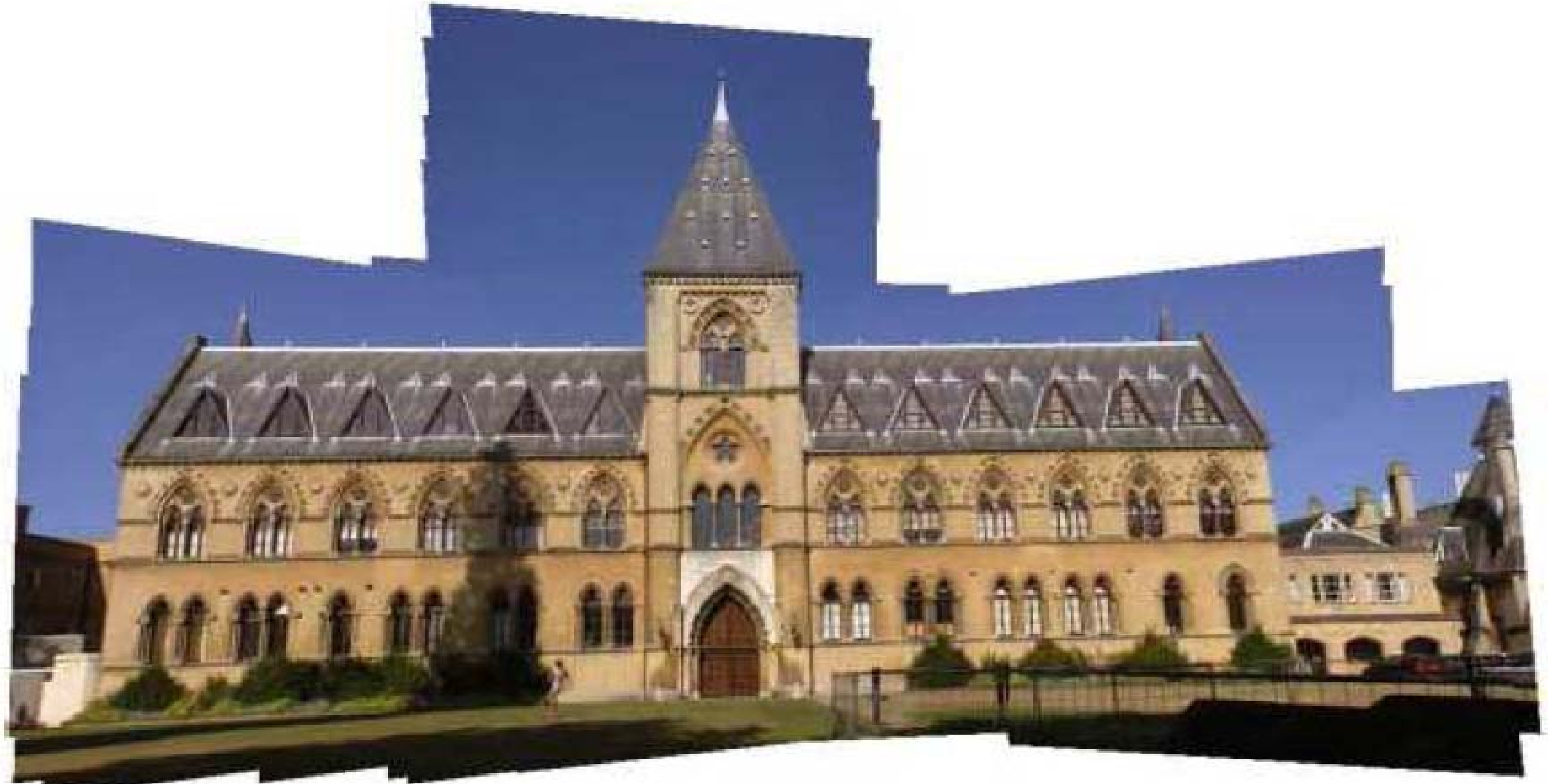
$$\vec{p^h} = H\vec{q^h}$$

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)

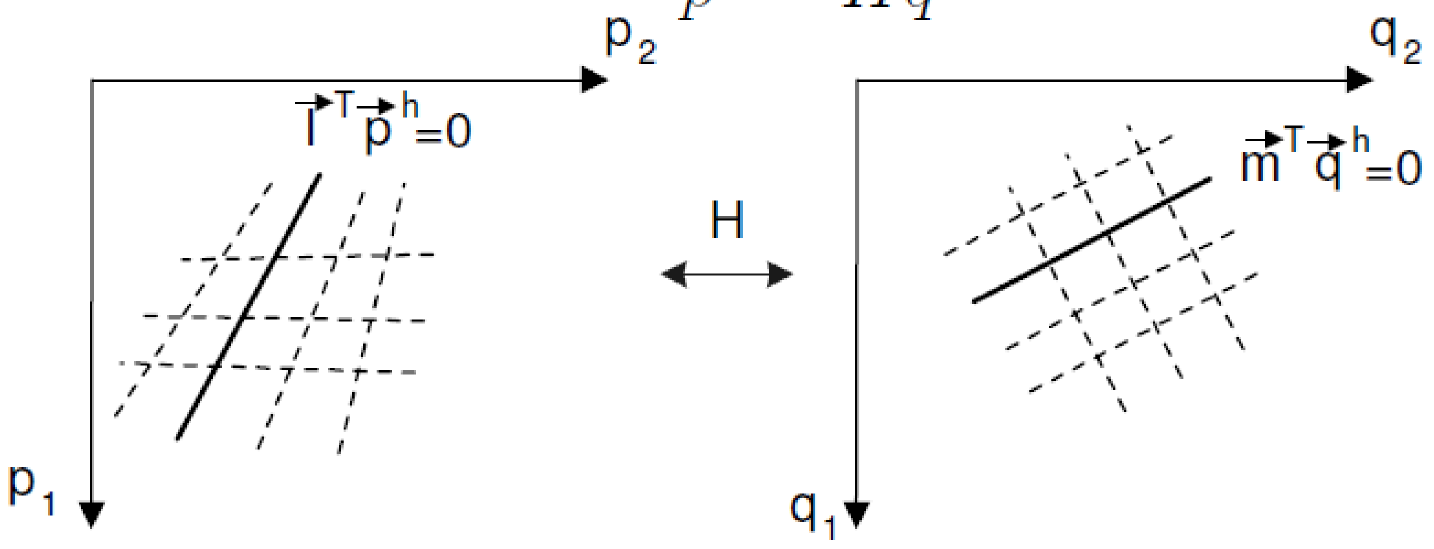
- The mapping defined by (1) is called a 2D **homography**
- Since the homography matrix H has 8 degrees of freedom, 4 corresponding $(\vec{p}, \vec{q})$ pairs are enough to constrain the problem





Scene Homography (Lines)

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



$$\vec{l^T}\vec{p^h} = 0 \qquad \longleftarrow \qquad \vec{m^T}\vec{q^h} = 0$$

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

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

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

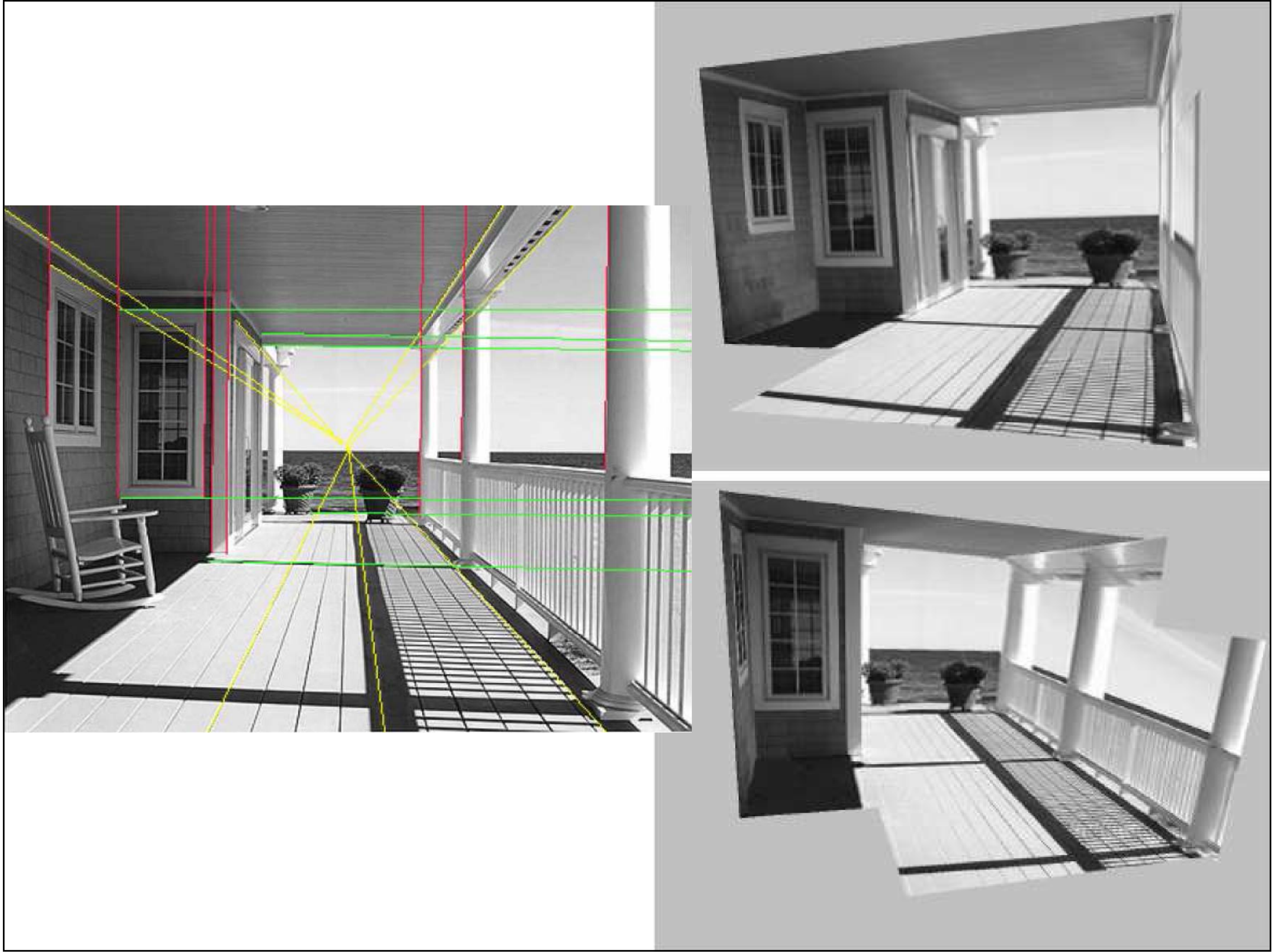
$$c = l^T H$$

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

What about H, from above ??

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

Possible to compute H, now ??



Rectification (Zhang's)

Properties of rectified image pair:

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

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

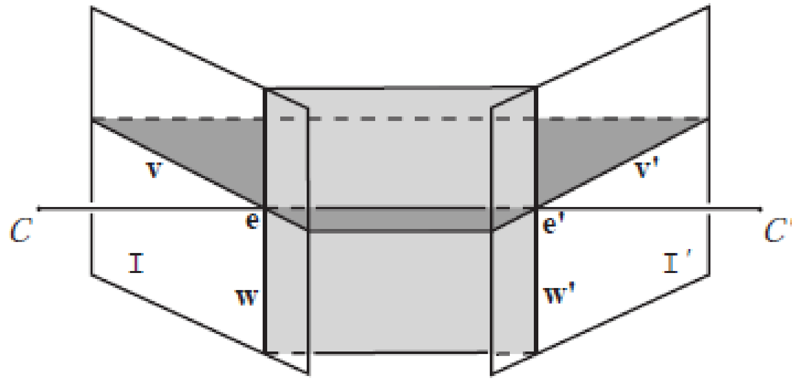
$$\bar{\mathbf{m}} = \mathbf{H}\mathbf{m} \quad \text{and} \quad \bar{\mathbf{m}}' = \mathbf{H}'\mathbf{m}'.$$

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

It follows from Eq. (1) that

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

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



resulting in the factorization

$$\mathbf{F} = \mathbf{H}'^T [\mathbf{i}]_\times \mathbf{H}. \qquad (2)$$

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.)

Let,

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

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

Then, the corresponding lines v and v', w and w' must be epipolar lines, for minimal distortion due to rectification; $\mathbf{H} = \mathbf{H}_{sh}.\mathbf{H}_{rs}.\mathbf{H}_p$

Homography: x = Hx';

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

Hx' lies on the corresponding epipolar line: Fx';

Thus, He'=e; H^{-1}e=e';

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

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

where, H_π is the homography imposed by epipolar plane

$$l' = [e']_\times H_\pi x = Fx = F[e]_\times l;$$

$$l = F^T [e']_\times l'$$

Simple steps of Rectification, given F:

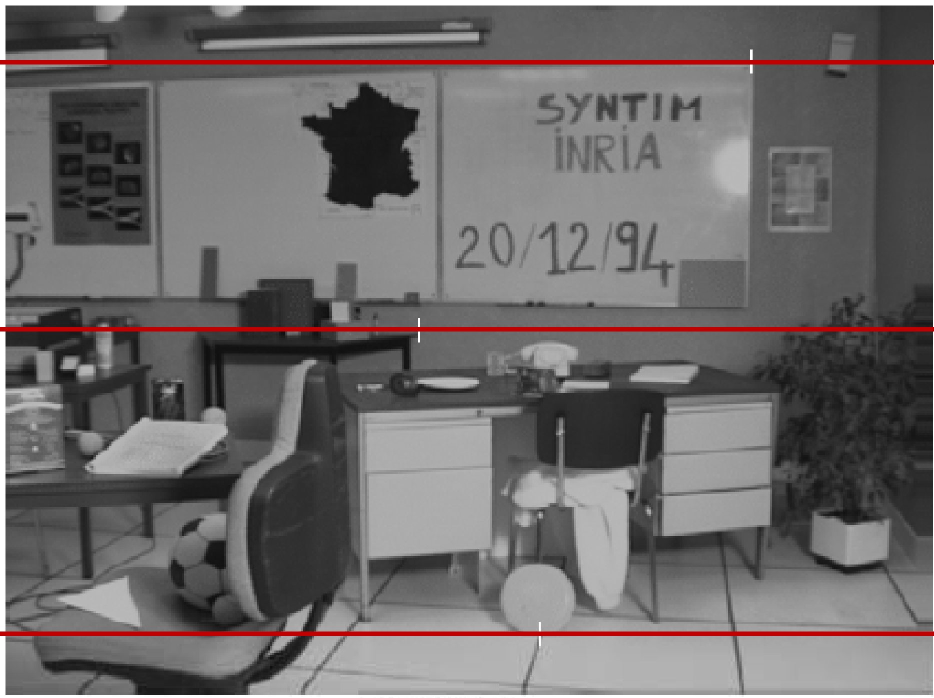
- Estimate H, to any one of the image pairs, for making epipolar lines coincident

(4-point pairs and RANSAC -RANdom SAmple Consensus);

- Epipolar lines are made parallel to x-axis;

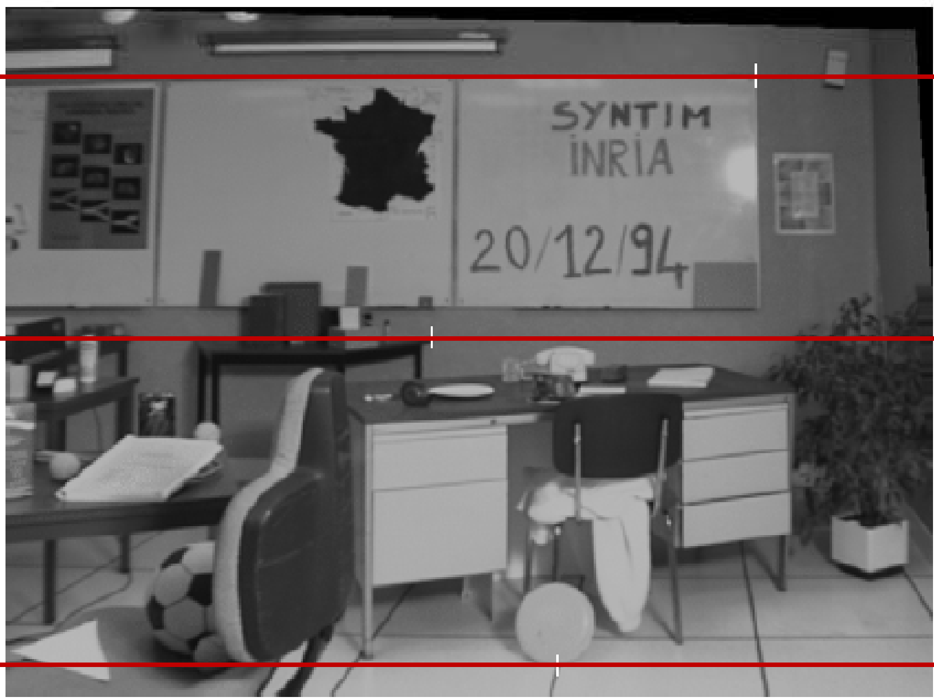
Three constraints for rectification: $\mathbf{He} = \mathbf{i},\ \mathbf{H}'\mathbf{e}' = \mathbf{i}$ and $\mathbf{H}'^T [\mathbf{i}]_\times \mathbf{H} = \mathbf{F}$

Left image



Right image

Rectified left image



Rectified right image

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 $A_i =$ [redacted]

and $\mathbf{h} = ($ [redacted]

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

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 $\mathbf{h}$.

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 (Altn. Method)

Given n>=4 2-D point pairs;

Algo:

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

Use:

-

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

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

- Assemble n 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

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;

 BMVC 2010 doi:10.5244/C.24.89

The idea is to transform both images so that the fundamental matrix gets the form $[\mathbf{i}]_\times$. Unlike the other methods which directly parameterize the homographies from the constraints $\mathbf{He} = \mathbf{i}$, $\mathbf{H'e'} = \mathbf{i}$ and $\mathbf{H'}^T[\mathbf{i}]_\times\mathbf{H} = \mathbf{F}$ and find an optimal pair by minimizing a measure of distortion, we shall compute the homography by explicitly rotating each camera around its optical center. The algorithm is decomposed into three steps (Fig. 1):

1. Com
   left
   (e'_x,
2. Rota
   (1,0
3. Rota
   tion
   deno

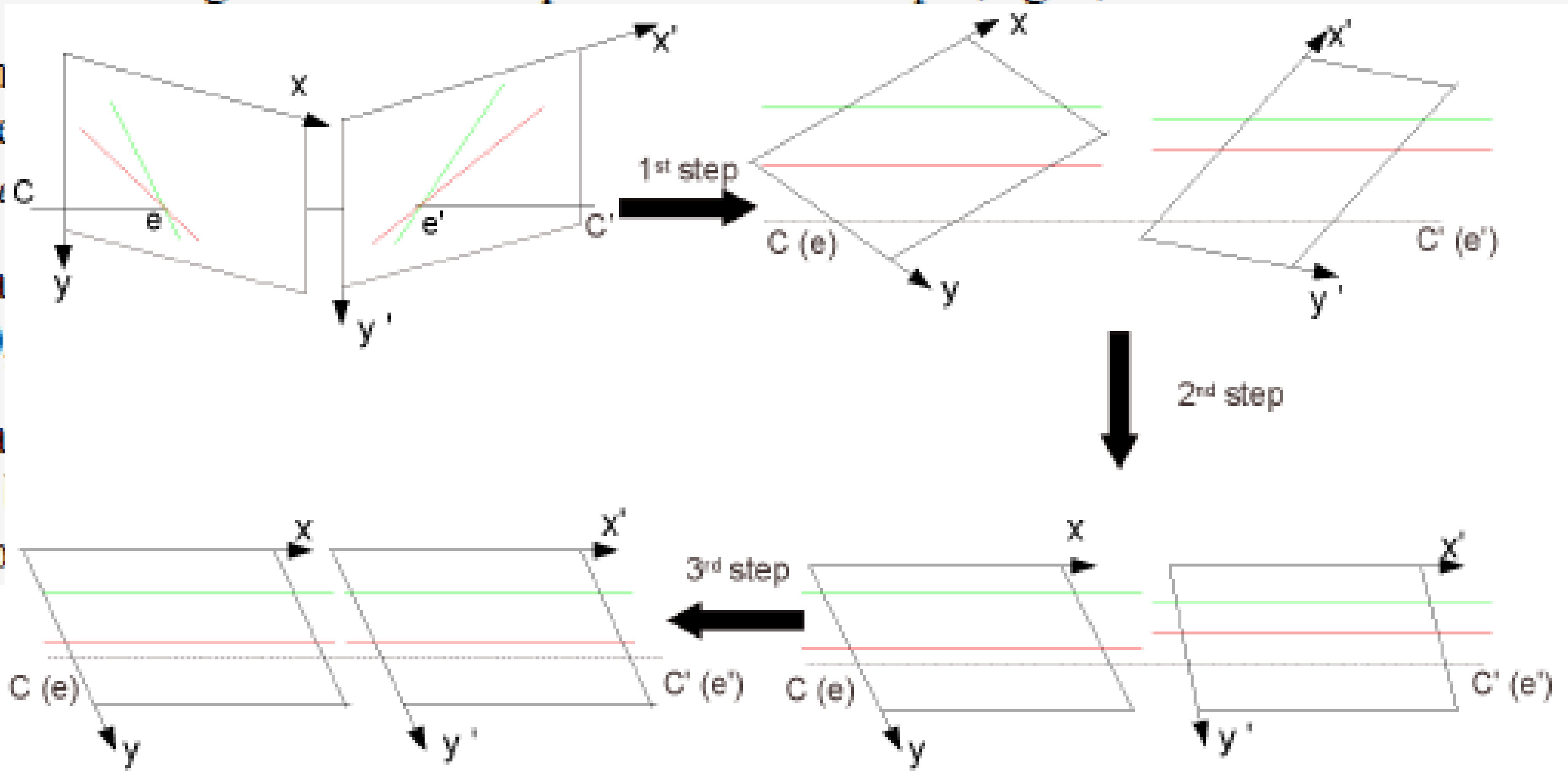


Figure 1: Three-step rectification. First step: the image planes become parallel to CC'. Second step: the images rotate in their own plane to have their epipolar lines also parallel to CC'. Third step: a rotation of one of the image planes around CC' aligns corresponding epipolar lines in both images. Note how the pairs of epipolar lines become aligned.

 BMVC 2010 doi:10.5244/C.24.89

Input: F, computed using correspondences; which gives epipoles e and e';

Let,

$$\mathbf{x}_1 = K[\mathbf{I} \mid 0]\mathbf{X}; \Rightarrow K^{-1}\mathbf{x}_1 = [\mathbf{I} \mid 0]\mathbf{X}$$

$$\mathbf{x}_2 = K.\mathbf{R}[\mathbf{I} \mid 0]\mathbf{X};$$

$$\Rightarrow \mathbf{x}_2 = K.\mathbf{R}K^{-1}\mathbf{x}_1 = H\mathbf{x}_1;$$

where, **Homography is:**

$$H = K.\mathbf{R}K^{-1}$$

Steps: 1 & 2:

$$\mathbf{H}_1\mathbf{e} = (e_x, e_y, 0)^T \text{ and } \mathbf{H}_1'\mathbf{e}' = (e_x', e_y', 0)^T$$

$$\mathbf{H}_1 = \mathbf{KRK}^{-1} \text{ and } \mathbf{H}_1' = \mathbf{KR}'\mathbf{K}^{-1}$$

$$\mathbf{RK}^{-1}\mathbf{e} = \mathbf{K}^{-1}(e_x, e_y, 0)^T$$

rotates the vector $\mathbf{a} = \mathbf{K}^{-1}\mathbf{e}$ to $\mathbf{b} = \mathbf{K}^{-1}(e_x, e_y, 0)^T$

$$\mathbf{K} = \begin{bmatrix} f & 0 & \frac{w}{2} \\ 0 & f & \frac{h}{2} \\ 0 & 0 & 1 \end{bmatrix}$$

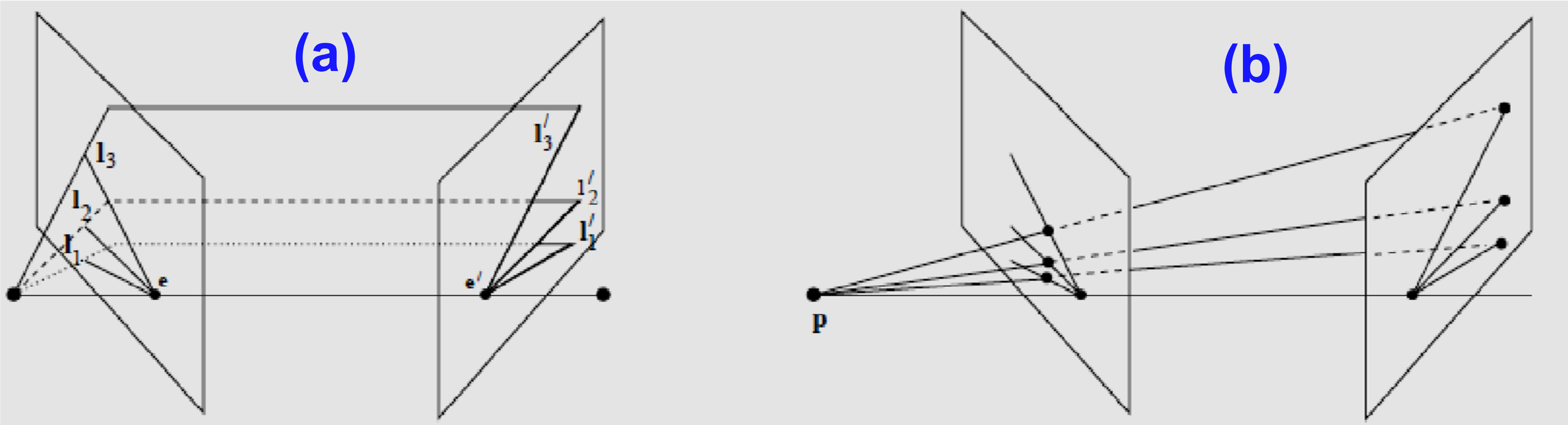
$$\mathbf{R}(\theta, \mathbf{t}) = \mathbf{I} + \sin\theta[\mathbf{t}]_\times + (1 - \cos\theta)[\mathbf{t}]_\times^2$$

minimal angle θ is $\operatorname{acos}(\frac{\mathbf{a}\cdot\mathbf{b}}{|\mathbf{a}||\mathbf{b}|})$ and the rotation axis $\mathbf{t}$ is $\frac{\mathbf{a}\times\mathbf{b}}{|\mathbf{a}||\mathbf{b}|}$

$\mathbf{H}_1, \mathbf{H}_1', \mathbf{H}_2$ and $\mathbf{H}_2'$ are all parametrized by f

Step 3: Rotation R^, of one camera about baseline: $\hat{\mathbf{F}} = \mathbf{K}^{-T}[\mathbf{i}]_\times\hat{\mathbf{R}}\mathbf{K}^{-1}$

H_3 is obtained after obtaining optimal K (or f)



Epipolar line homography:

(a) There is a pencil of epipolar lines in each image centred on the epipole. The correspondence between epipolar lines, li ↔ 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.

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 $X(\lambda)$ varies from finite point A to point D at ∞.
Assume $P = K[I \mid 0]$, then image of $X(\lambda)$ 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 $v = Kd$

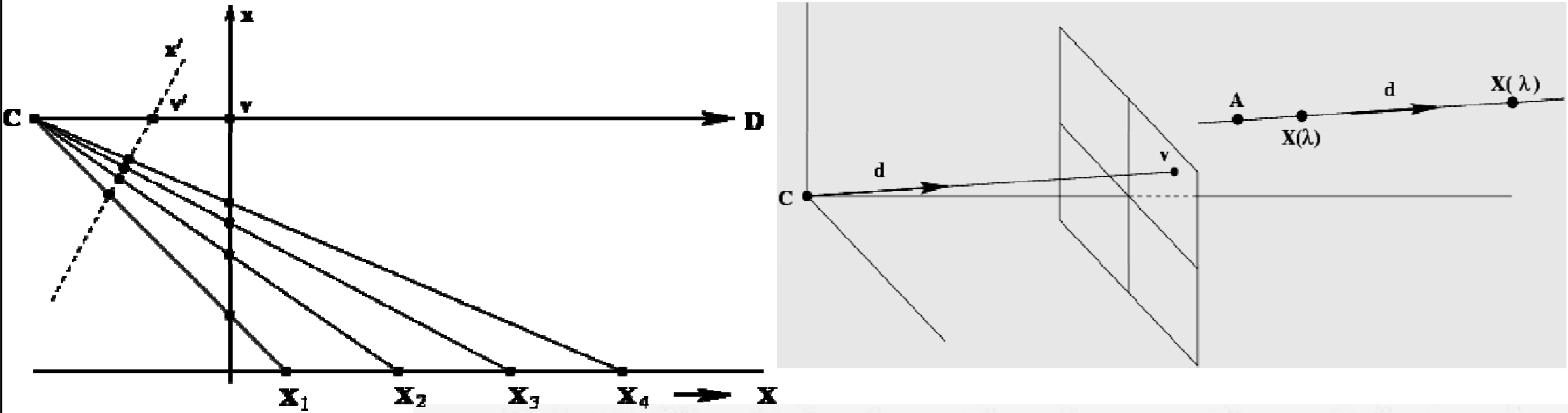
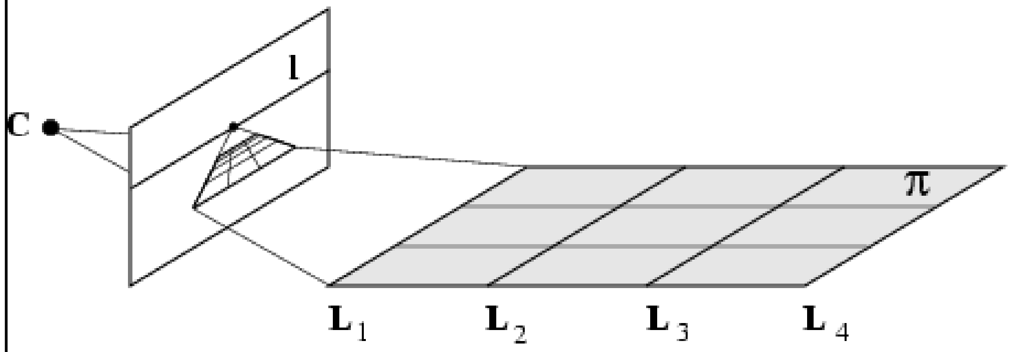
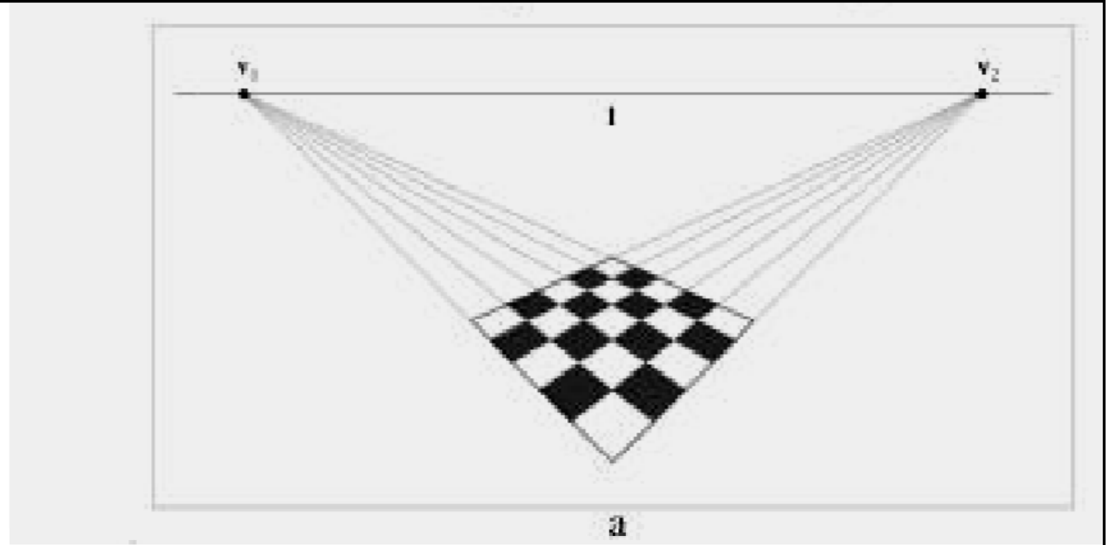


Fig. 8.14. **Vanishing point formation.** *(a) Plane to line camera. The points $\mathbf{X}_i$, $i = 1, \ldots, 4$ are equally spaced on the world line, but their spacing on the image line monotonically decreases. In the limit $\mathbf{X}$ ∞ the world point is imaged at $\mathbf{x} = \mathbf{v}$ on the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

Vanishing lines





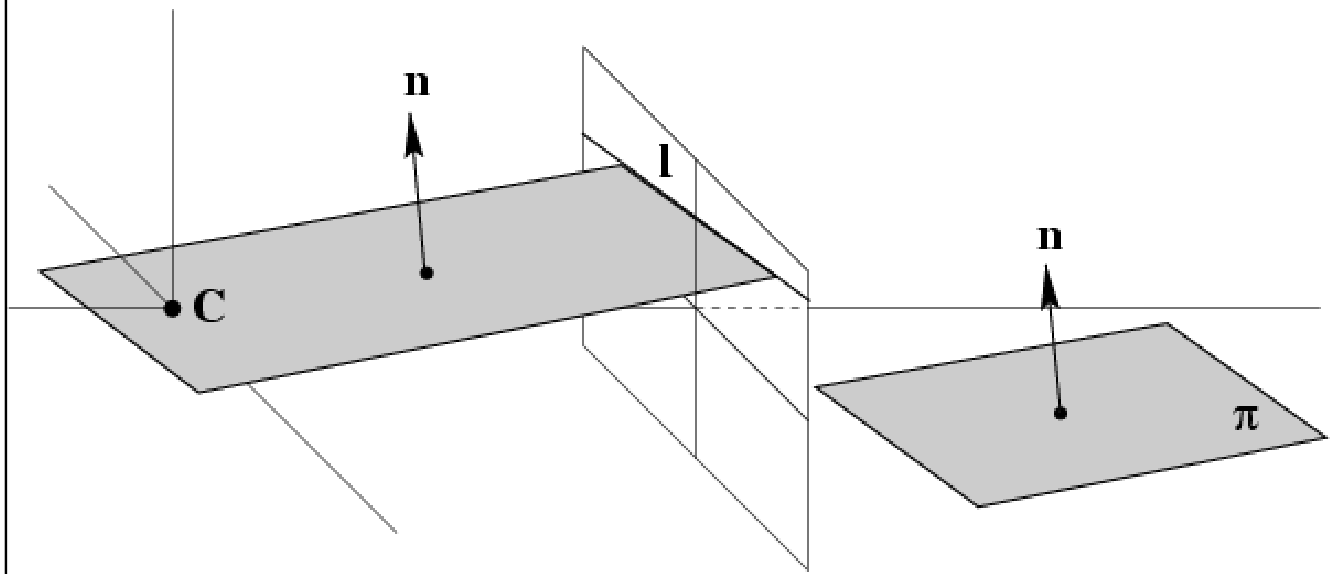
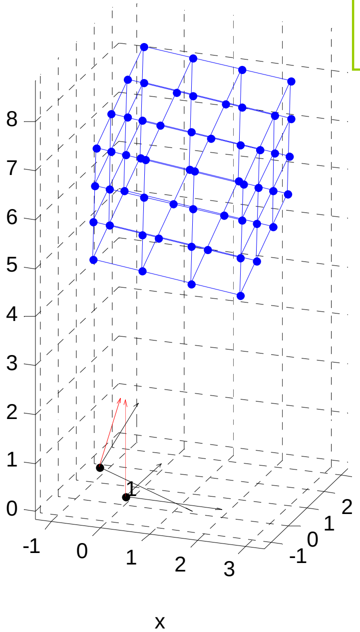
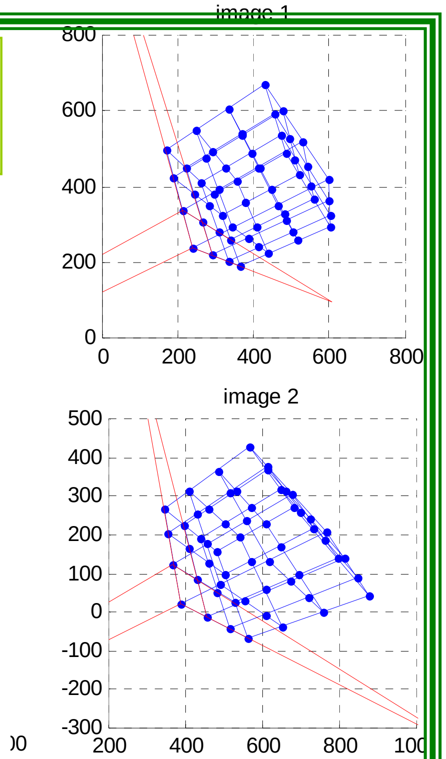


Fig. 8.16. **Vanishing line formation.** *(a) The two sets of parallel lines on the scene plane converge to the vanishing points* $\mathbf{v}_1$ *and* $\mathbf{v}_2$ *in the image. The line* $\mathbf{l}$ *through* $\mathbf{v}_1$ *and* $\mathbf{v}_2$ *is the vanishing line of the plane. (b) The vanishing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

Reconstruction Framework





3D world object

2D View (Cam1) → Feature Extraction

2D View (Cam2) → Feature Extraction

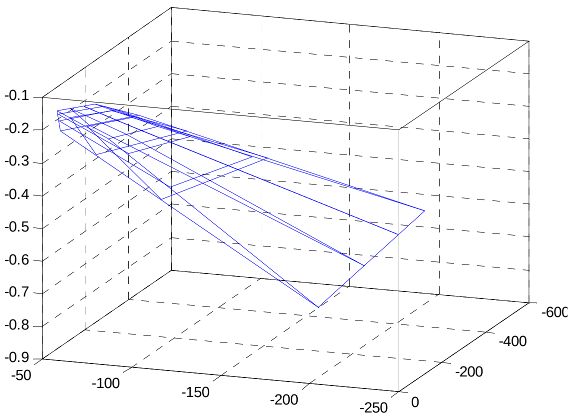
Find Correspondence

Fundamental / Essential Matrix

$$x'^{T} F x = 0 \text{ or } x'^{T} E x = 0$$

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

$$E = K'^{T} F K;$$



Projective Reconstruction

(Triangulation process) [a]

$$x = PX$$

$$x_{\times} PX = 0$$

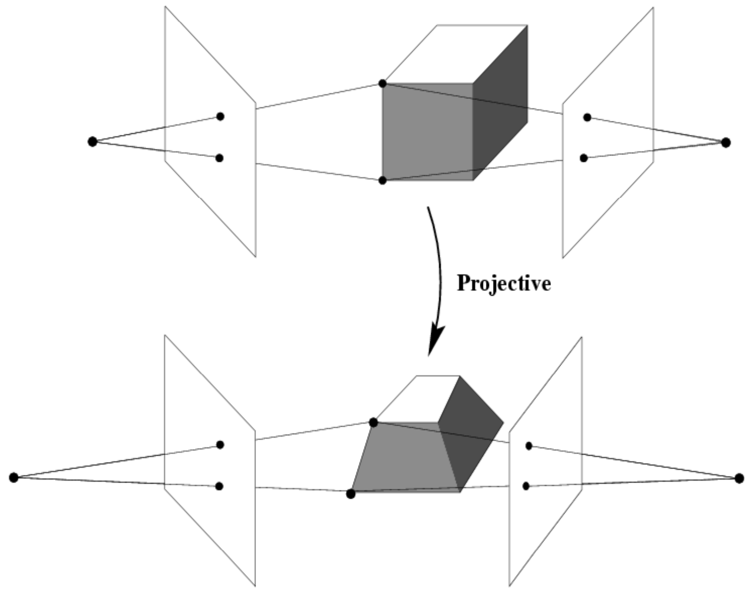
$$AX = 0$$

[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

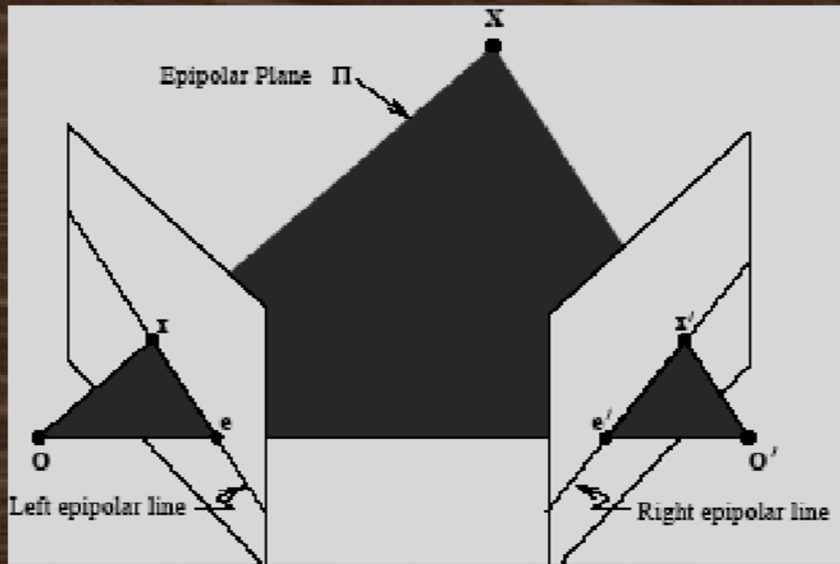
  (b) Stratified (use, VP, VL, Homography, Abs. conic etc.):

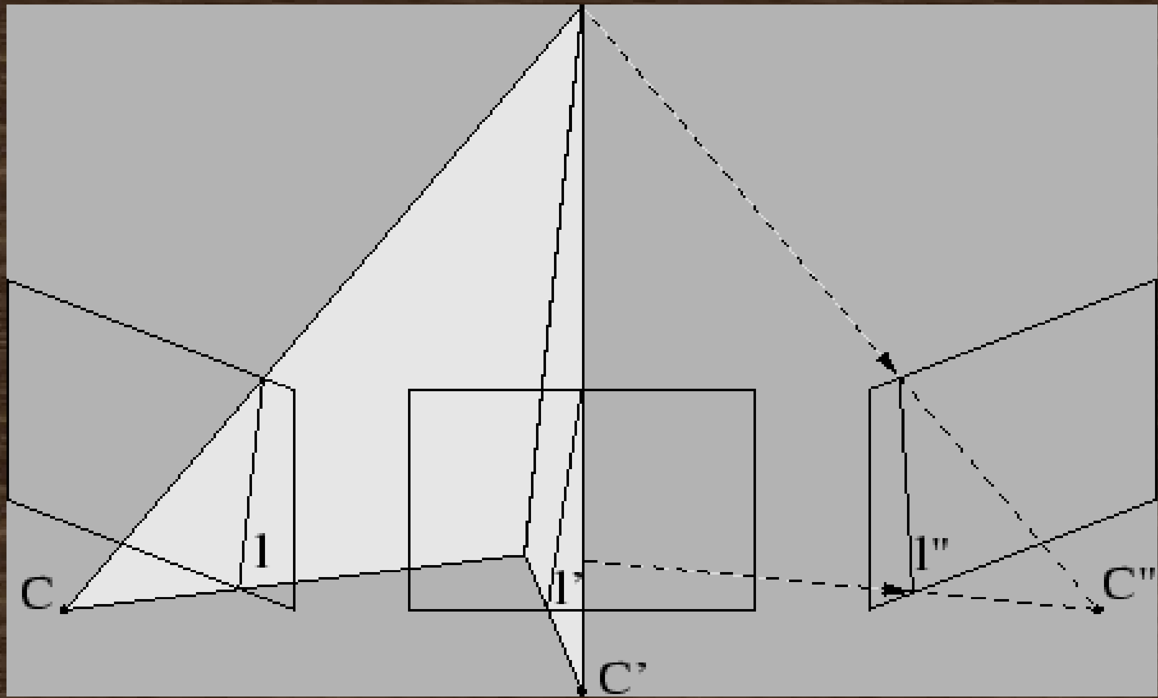
  Affine;

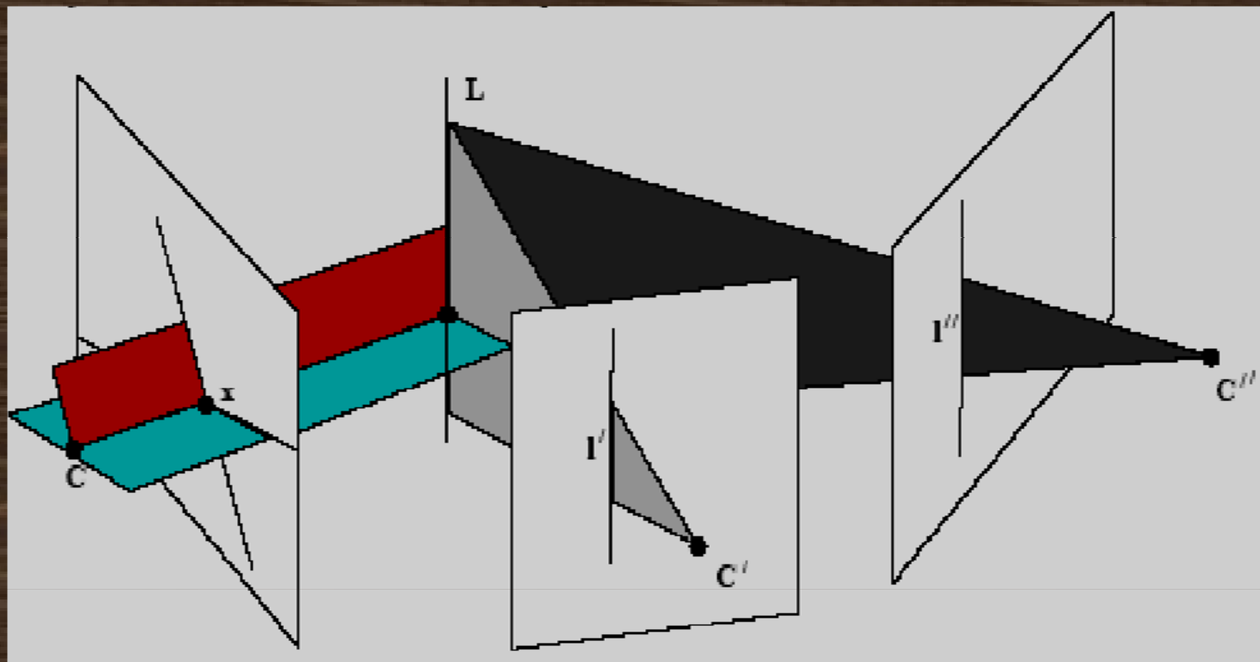
  Metric

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




C
l
l'
l"
C"
C'


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

Example of 3-D reconstruction

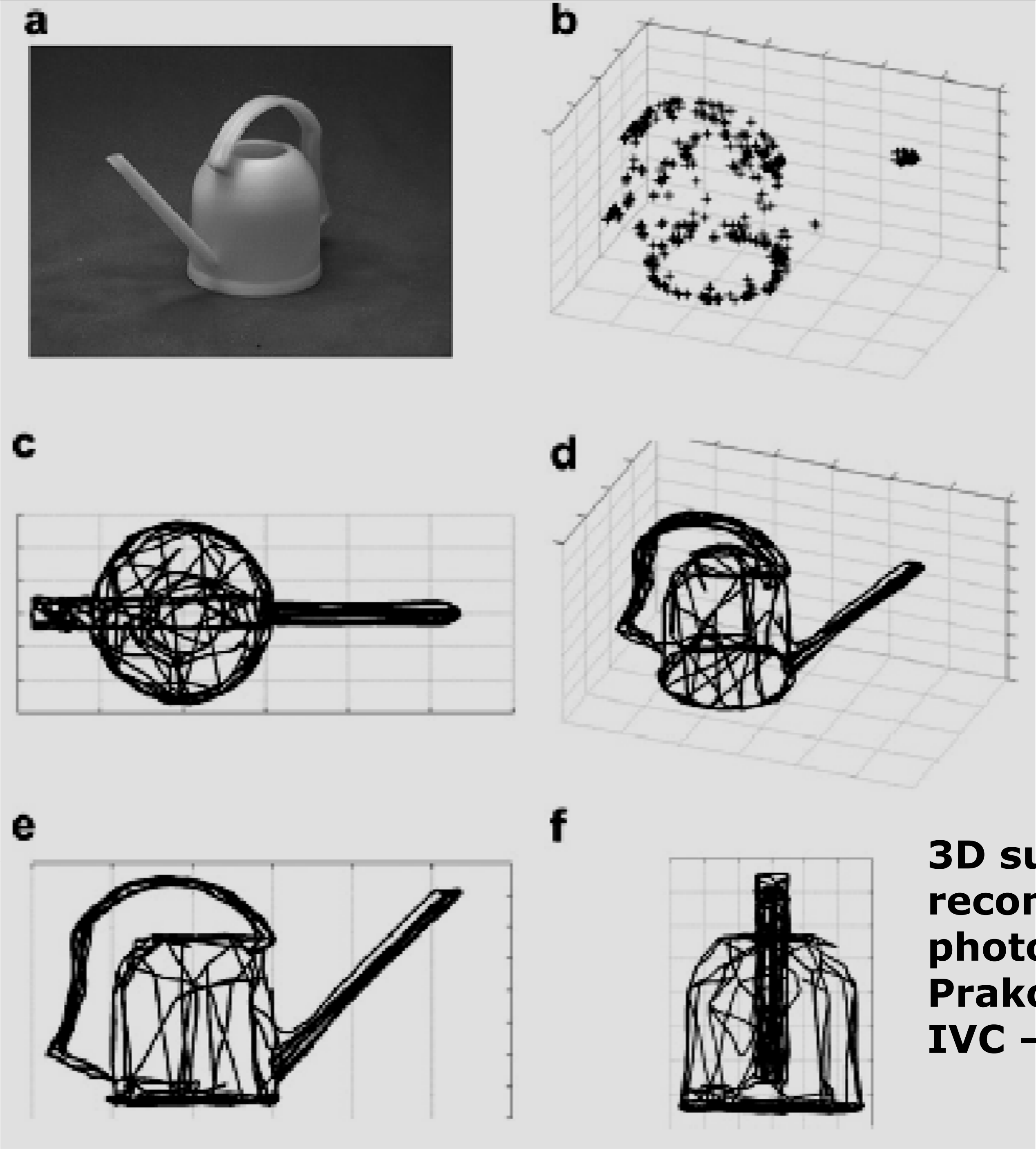
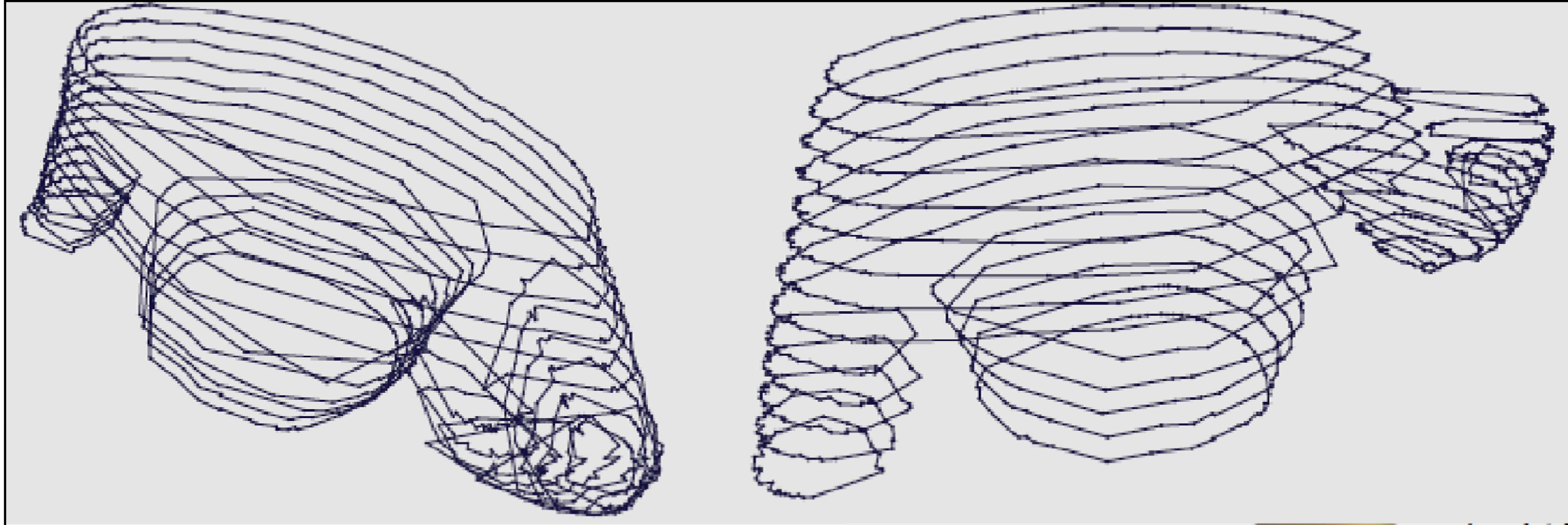
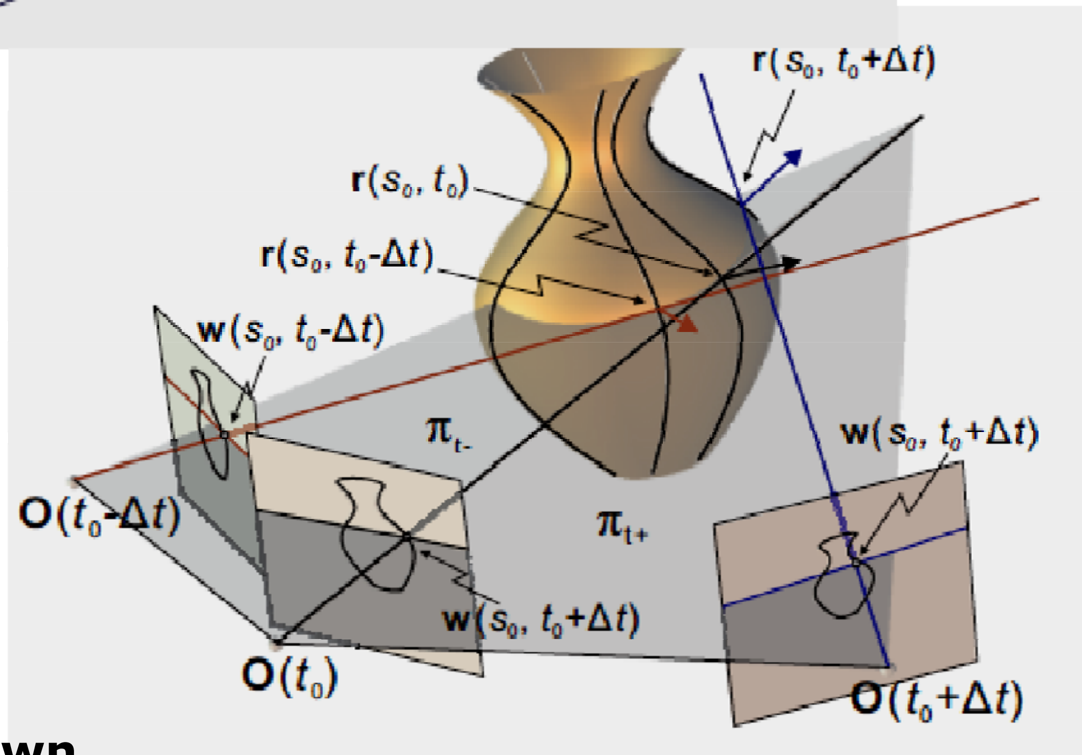


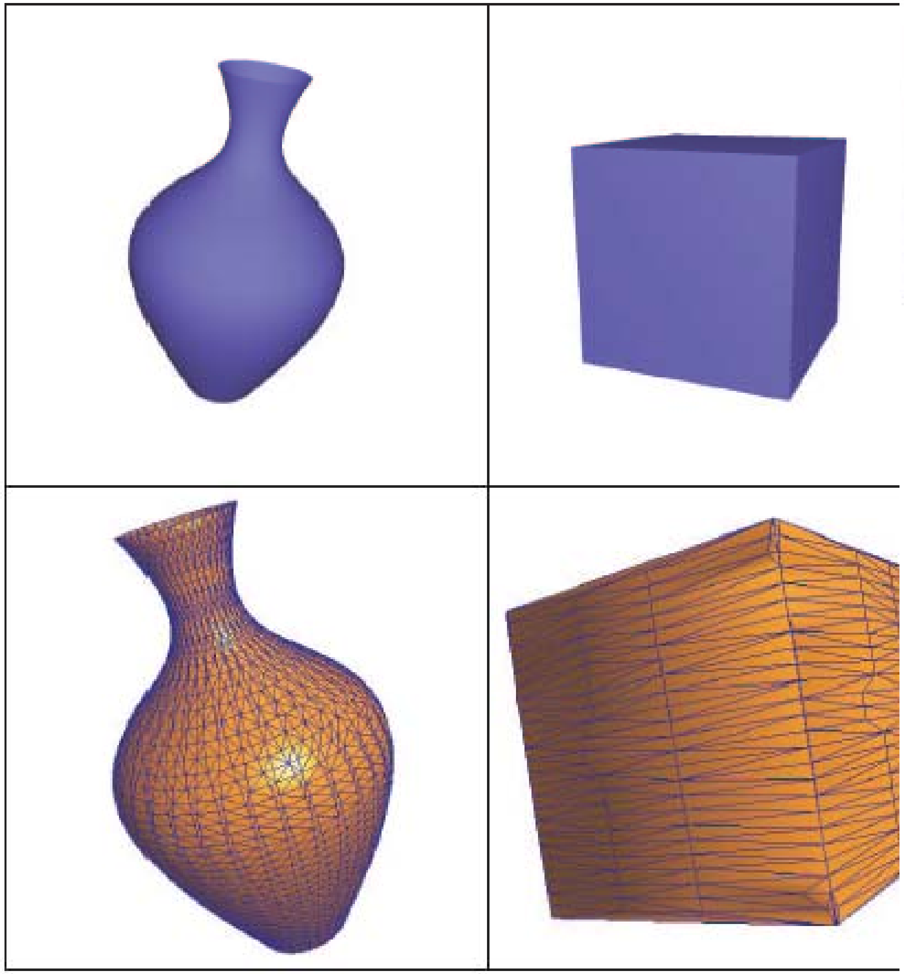
Fig. 18. (a) Real matt plastic watering pot. (b) The reconstructed 3D frontier points shown superimposed upon the pot. (c) - (f) Different views of the reconstructed 3D contour generators.

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