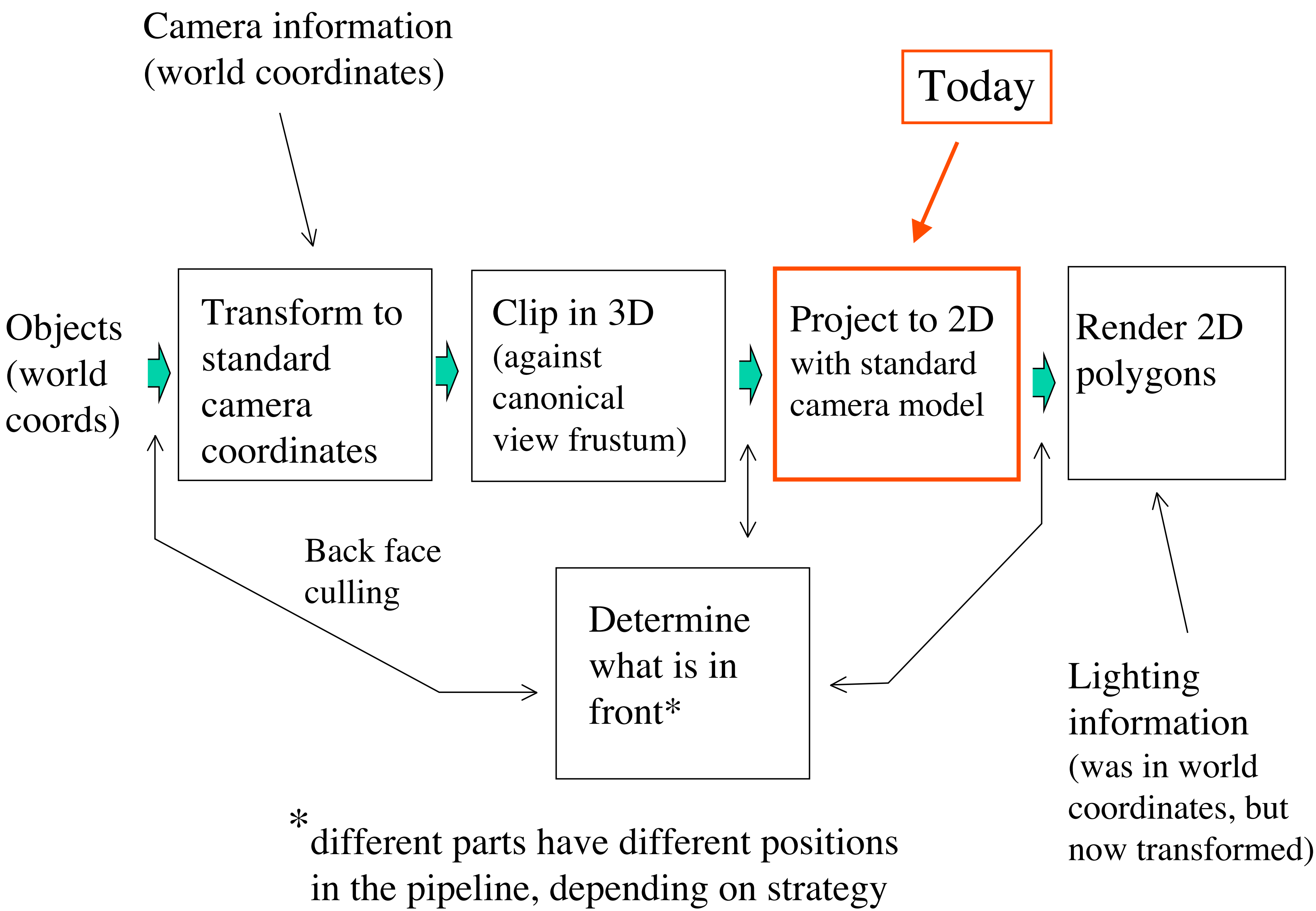
3D Graphics Concepts

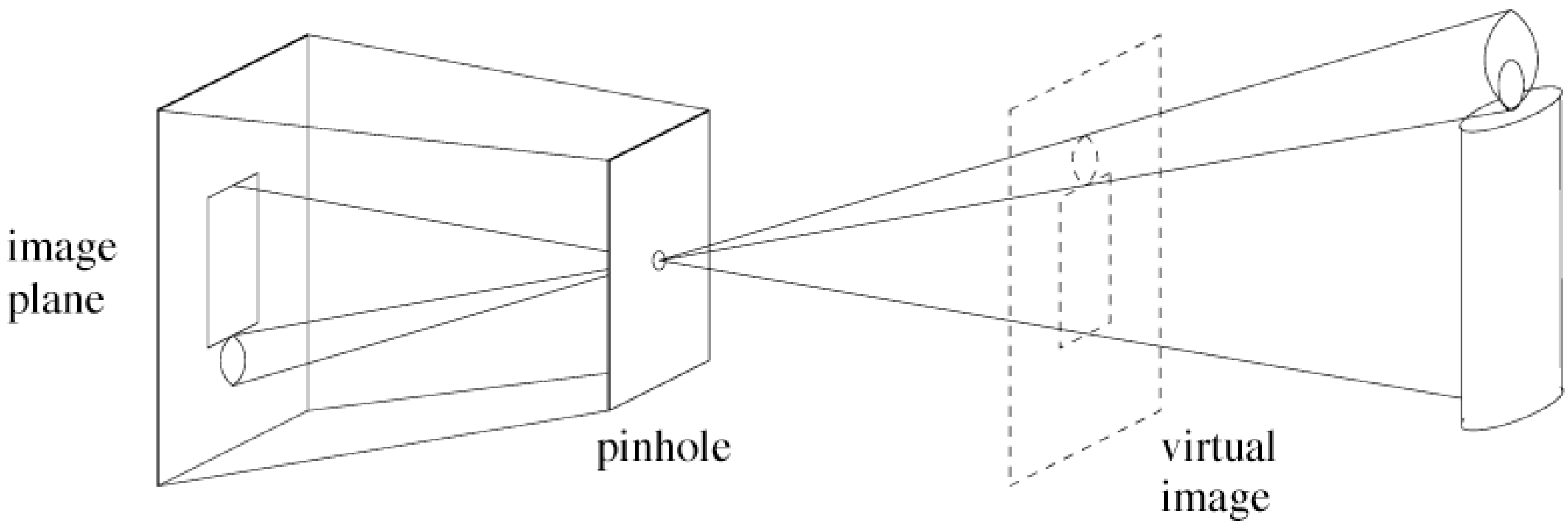
(Watt ch. 5, Foley et al ch. 6)

- Modeling: For now, objects will be collections of polygons in 3D. Complex shapes will be many small polygons.
- Issues:
  - Which polygons can be seen? (some polygons hide others, and some are outside the relevant volume of space and need to be clipped).
  - Where do they go in the 2D image? (key abstraction is a virtual camera)
  - How bright should they be? (for example, to make it look as if we are looking at a real surface)

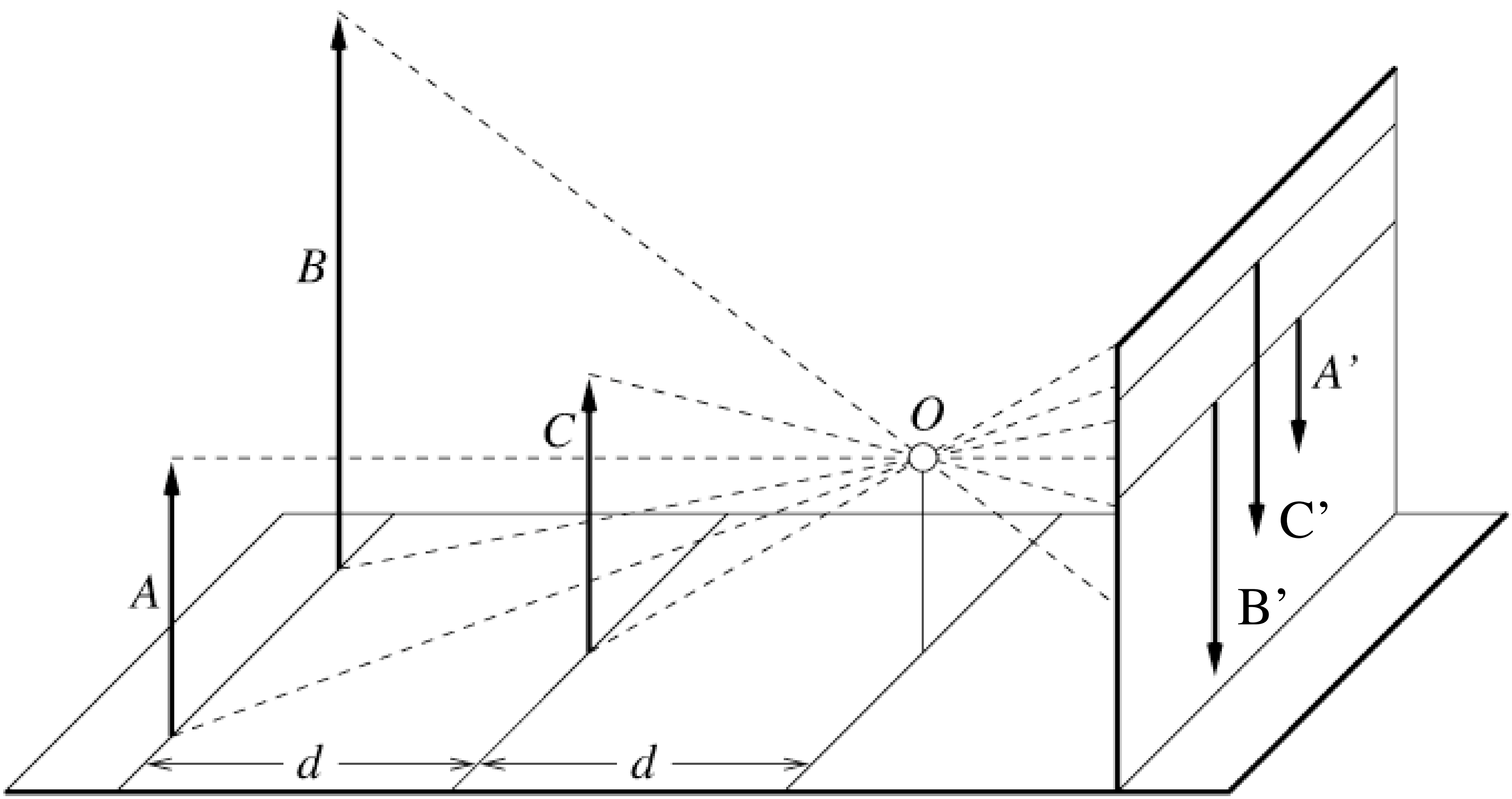

Camera information
(world coordinates)
Today
Objects
(world
coords)
Transform to
standard
camera
coordinates
Clip in 3D
(against
canonical
view frustum)
Project to 2D
with standard
camera model
Render 2D
polygons
Back face
culling
Determine
what is in
front*
Lighting
information
(was in world
coordinates, but
now transformed)
* different parts have different positions
in the pipeline, depending on strategy

Pinhole cameras

- Abstract camera model--box with a small hole in it
- Pinhole cameras work for deriving algorithms--a real camera needs a lens

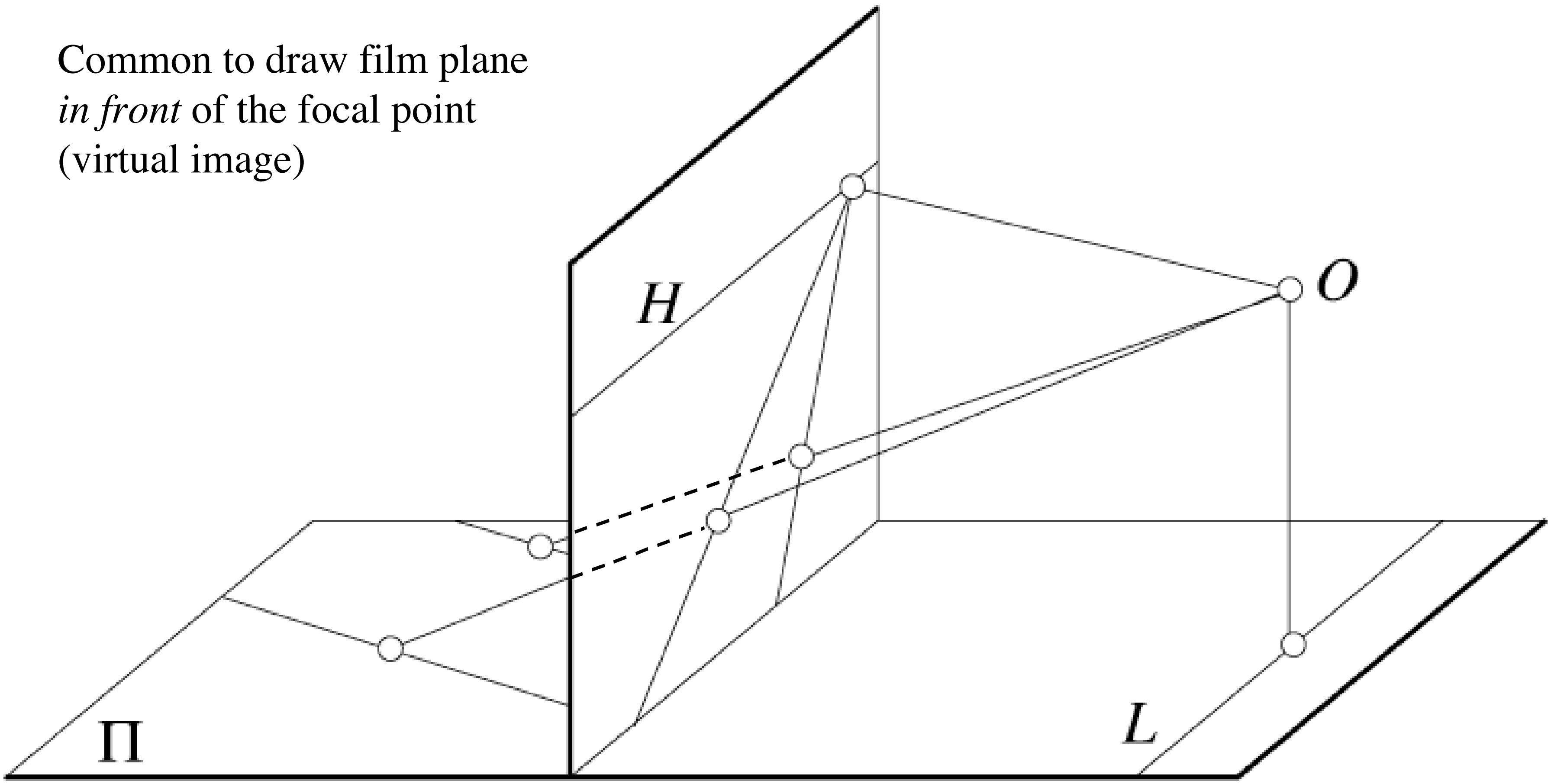


Distant objects are smaller



Parallel lines meet*

Common to draw film plane
in front of the focal point
(virtual image)



*Exceptions?

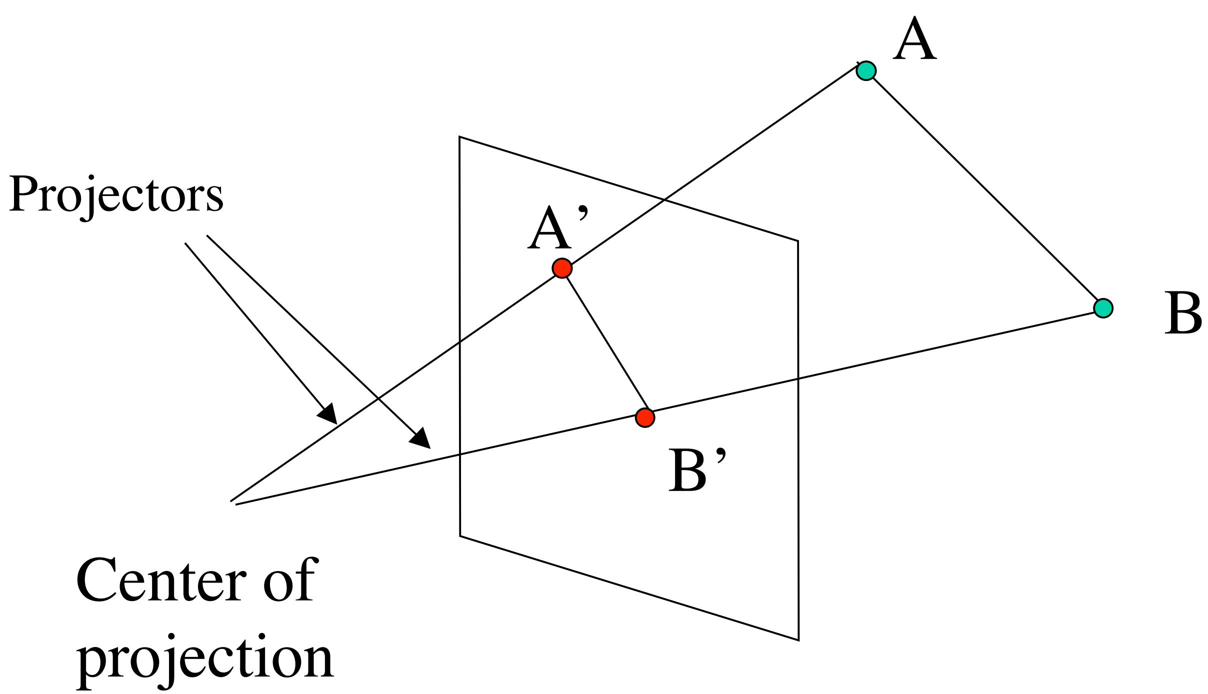
Vanishing points

- Each set of parallel lines (=direction) meets at a different point
  - The *vanishing point* for this direction
- Sets of parallel lines on the same plane lead to *collinear* vanishing points.
  - The line is called the *horizon* for that plane
  - Standard horizon is the horizon of the ground plane.
- One way to spot fake images

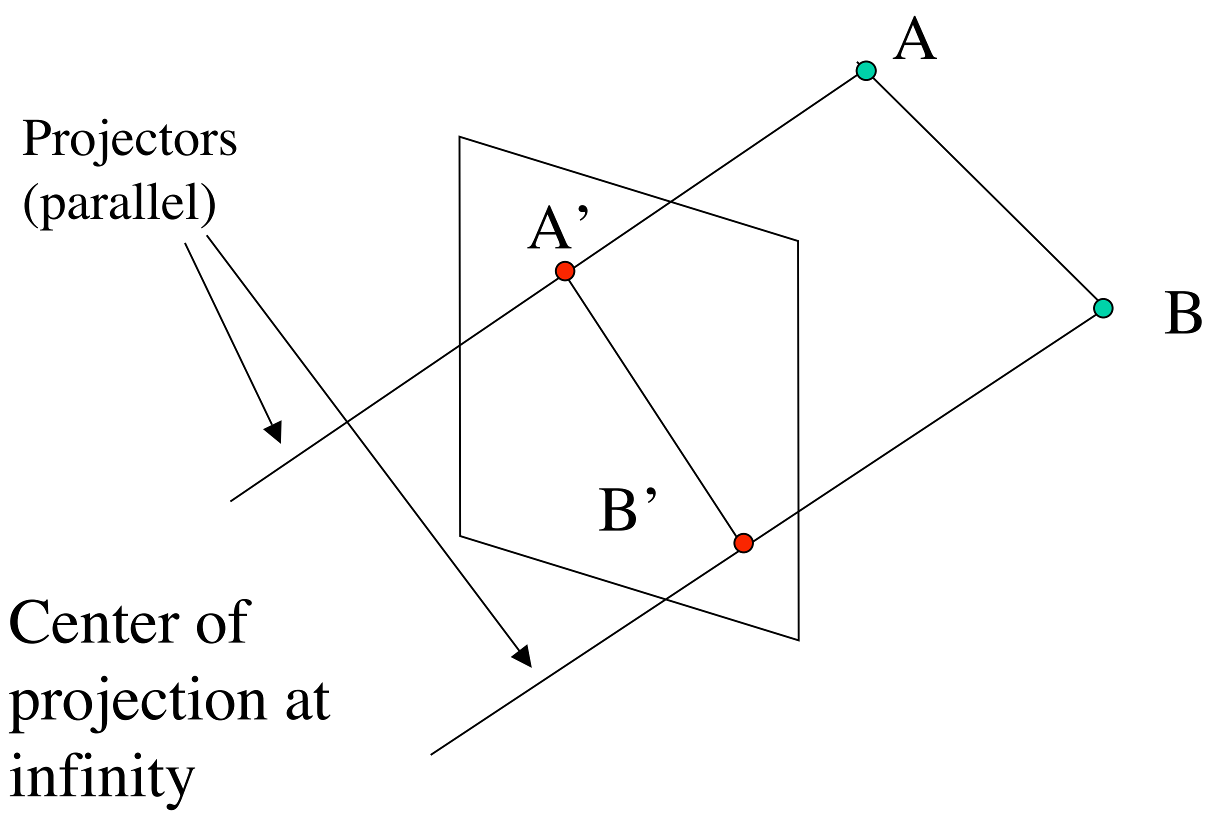
Projections

- Mathematical definition of a projection: PP=P
- (Doing it a second time has no effect).
- Generally rank deficient (non-invertable)--exception is P=I
- Transformation looses information (e.g., depth)
- Given a 2D image, there are many 3D worlds that could have lead to it.

Projections



Parallel Projection



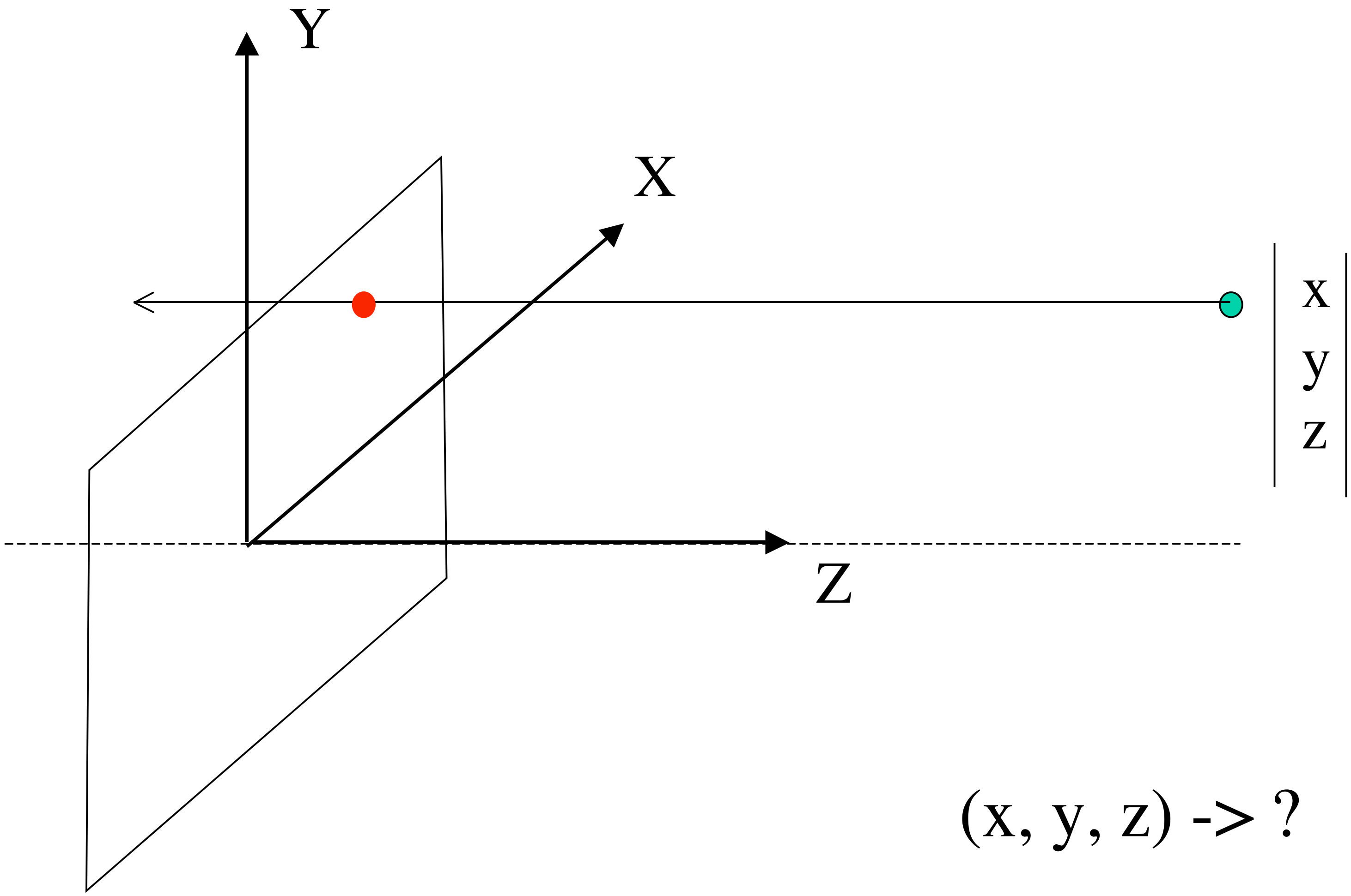
Parallel Projection

Parallel lines remain parallel, some 3D measurements can be made using 2D picture

Does not give realistic 3D view because eye is more like perspective projection.

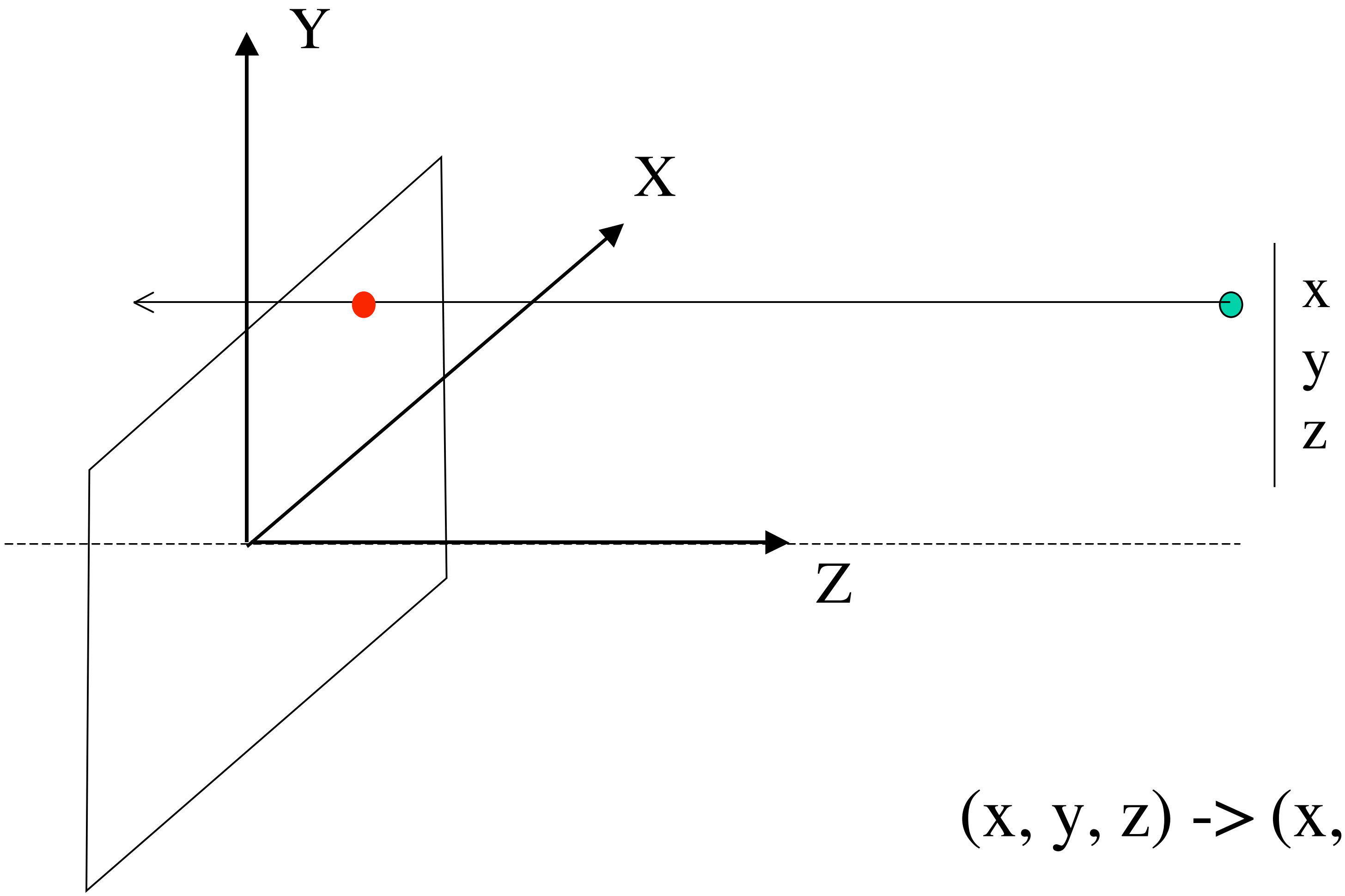
If projection plane is perpendicular to projectors the projection is *orthographic*

Orthographic example (onto z=0)



(x, y, z) -> ?

Orthographic example (onto z=0)



(x, y, z) -> (x, y, 0)

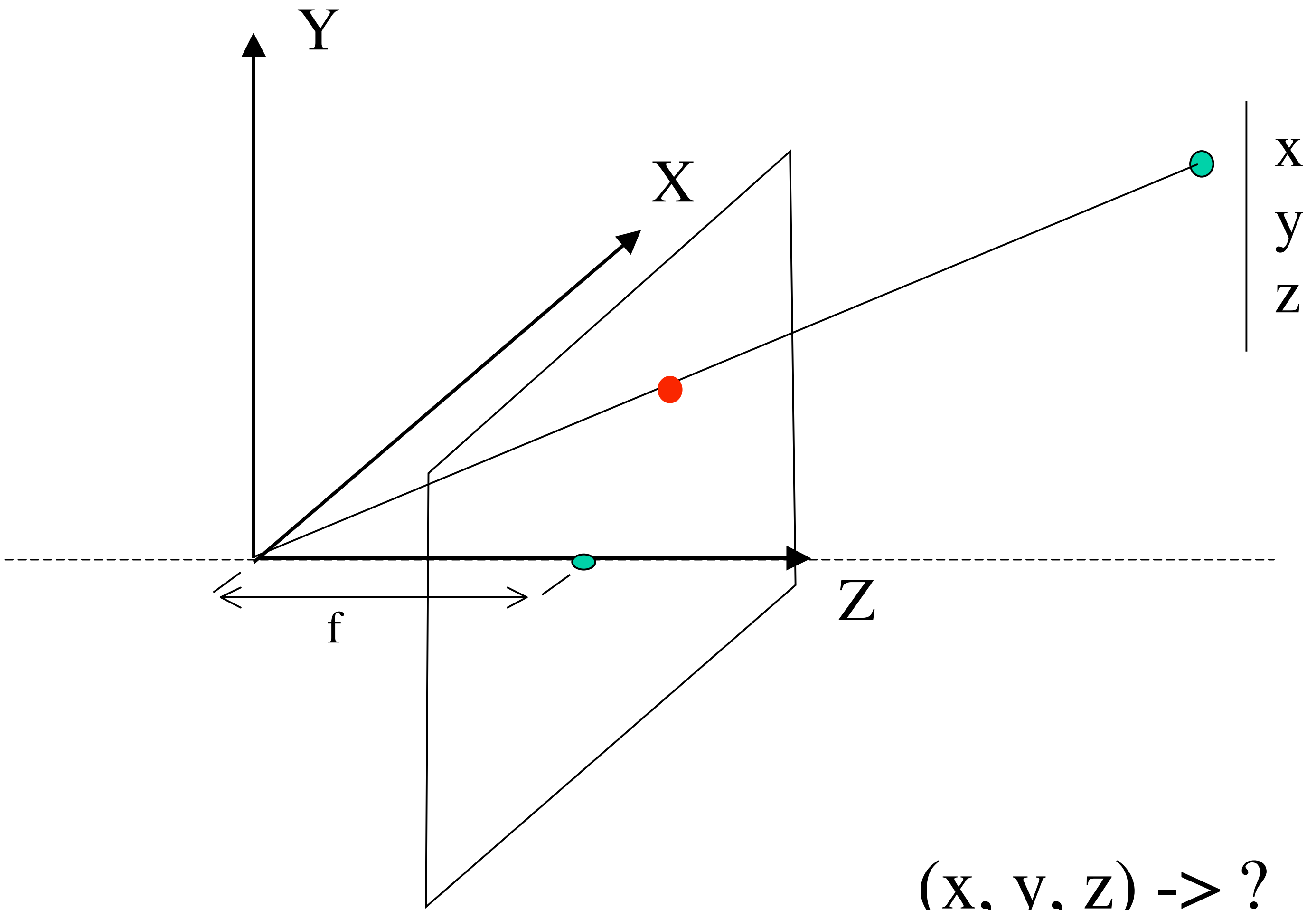
The camera matrix

$$\begin{bmatrix} x \\ y \\ 0 \\ w \end{bmatrix} = \begin{bmatrix} & & ? & & \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

The camera matrix

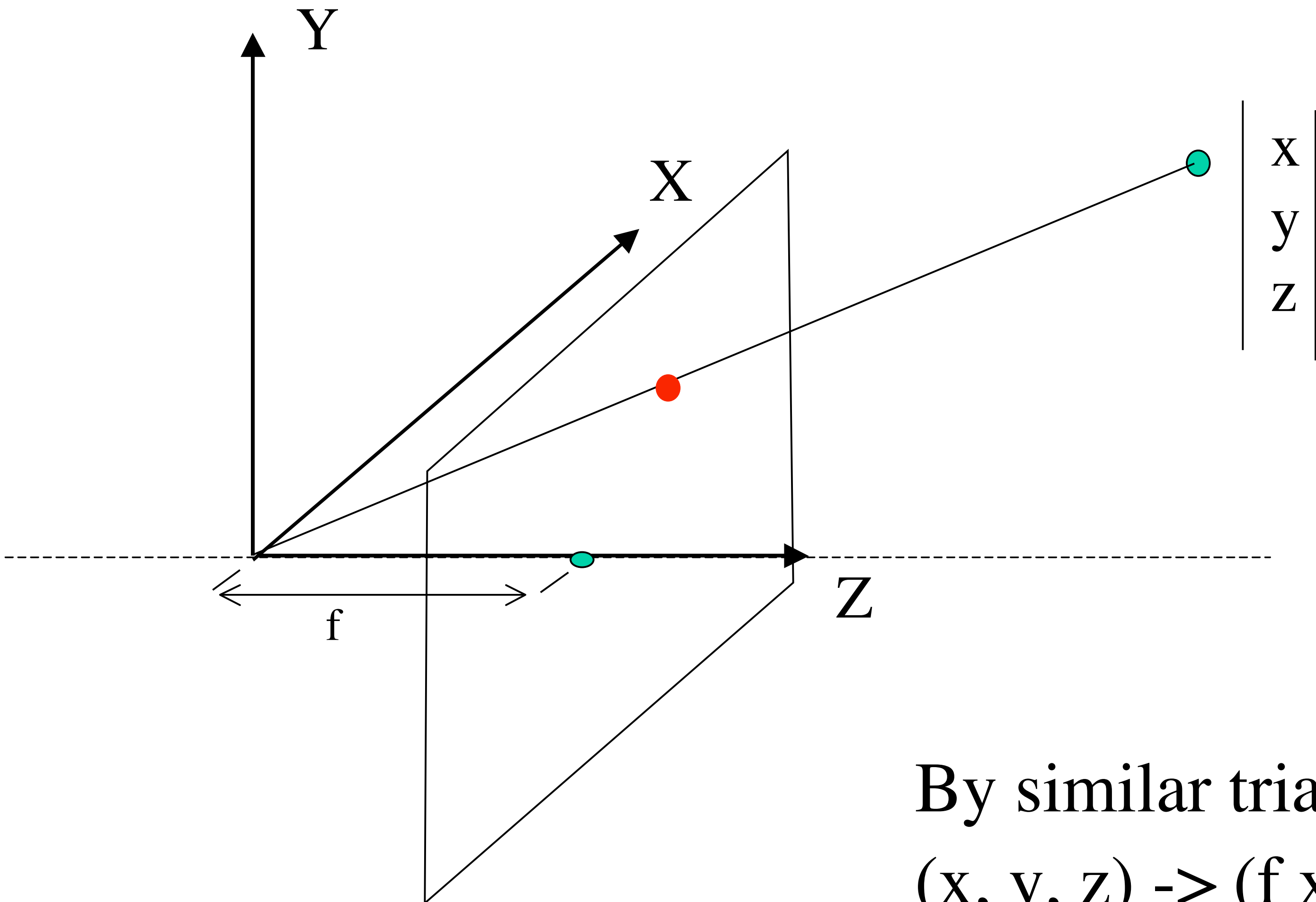
$$\begin{bmatrix} x \\ y \\ 0 \\ w \end{bmatrix} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 0 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

Perspective example (onto z=f)



(x, y, z) -> ?

Perspective example (onto z=f)



By similar triangles,

(x, y, z) -> (f x/z, f y/z, f)

The equation of projection

- In homogeneous coordinates

$$(x,\ y,\ z,\ 1) \rightarrow (f\frac{x}{z},\ f\frac{y}{z},\ f,\ 1)$$

- Equivalently

$$(x,\ y,\ z,\ 1) \rightarrow (x,\ y,\ z,\ \frac{z}{f})$$

- (Now H.C. are being used to store foreshortening)

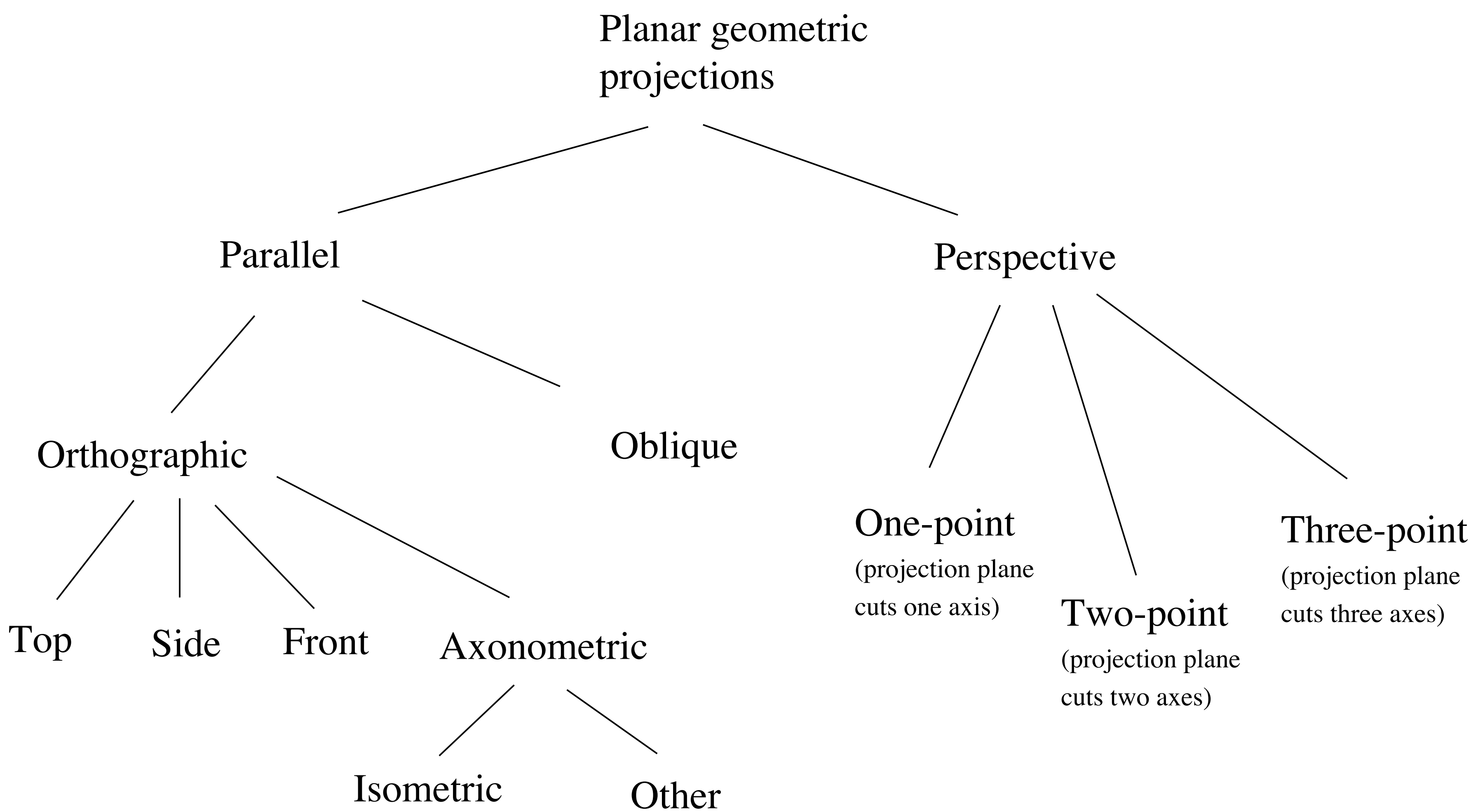
The camera matrix

$$\begin{bmatrix} x \\ y \\ z \\ \frac{z}{f} \end{bmatrix} = \begin{bmatrix} \; \\ \; \\ \; \\ \; \end{bmatrix} \quad ? \quad \begin{bmatrix} \; & \; \\ \; & \; \\ \; & \; \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

The camera matrix

$$\begin{pmatrix} x \\ y \\ z \\ \frac{z}{f} \end{pmatrix} = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & \frac{1}{f} & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

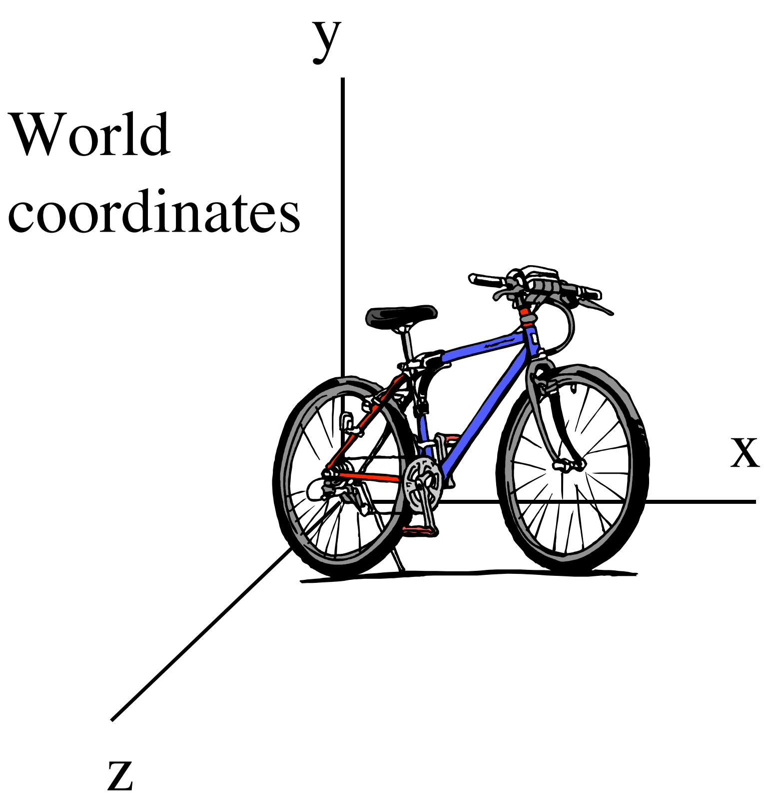
Projection Taxonomy

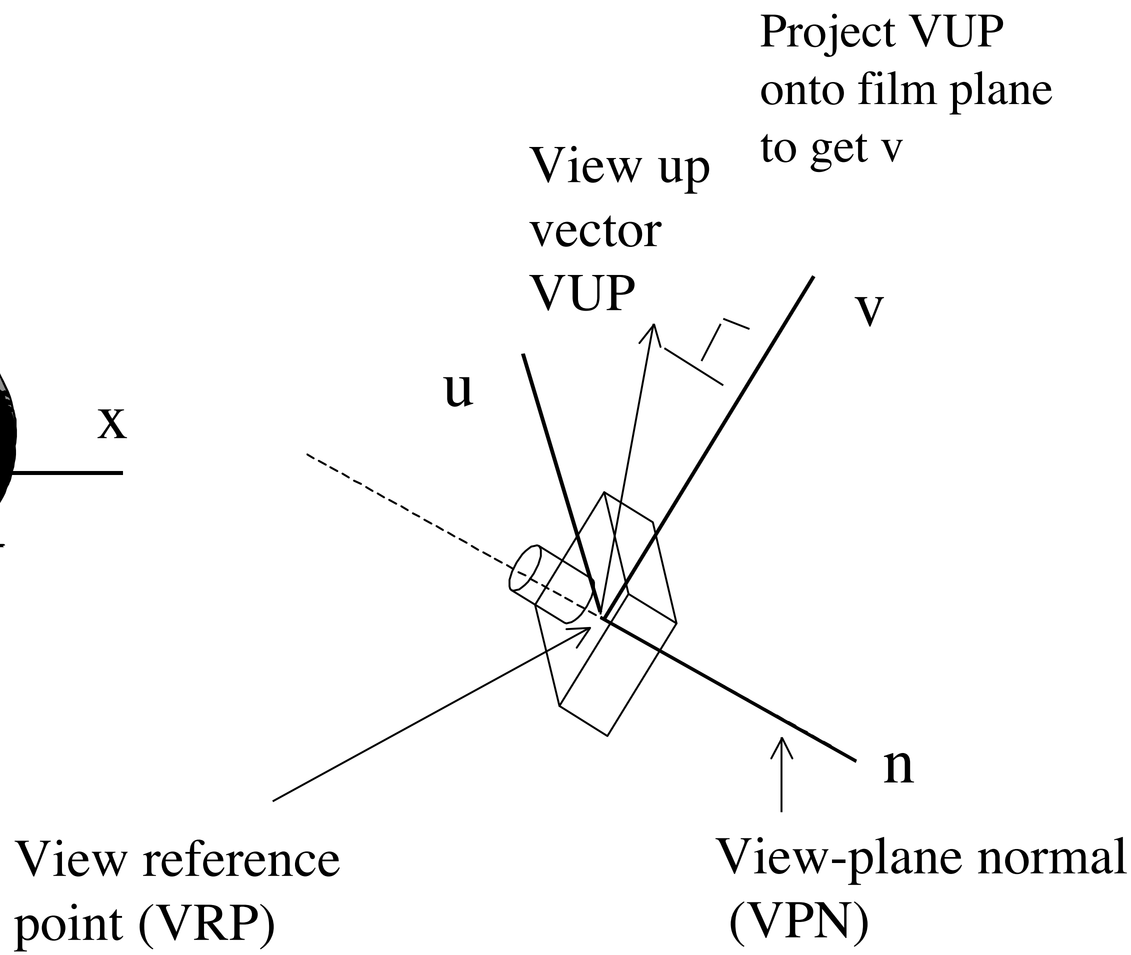


Specifying a camera

- Camera
  - Tell rendering system where camera is in world coordinates
  - We want to transform the world into camera coordinates so that projection is easy (i.e., we can use the projection matrix from a few slides back).
  - Need to specify focal point and film plane.
  - Convenient to construct a coordinate system for the camera with origin on film plane
- Clipping volume
  - we render only a window in the film plane
  - Things beyond any of four sides don't get rendered
  - Things that are too far away don't get rendered
  - Things that are too near don't get rendered.

Specifying a camera

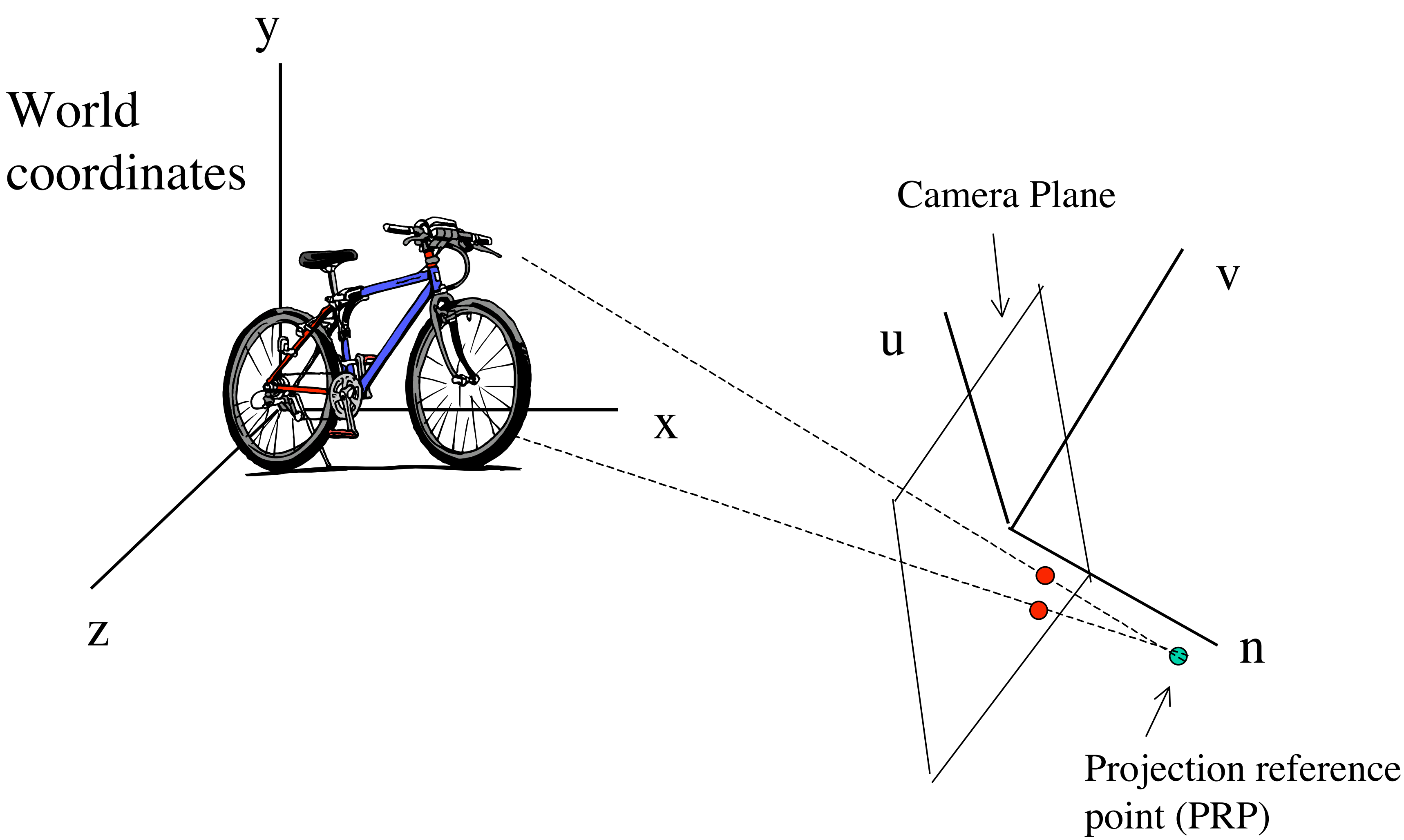




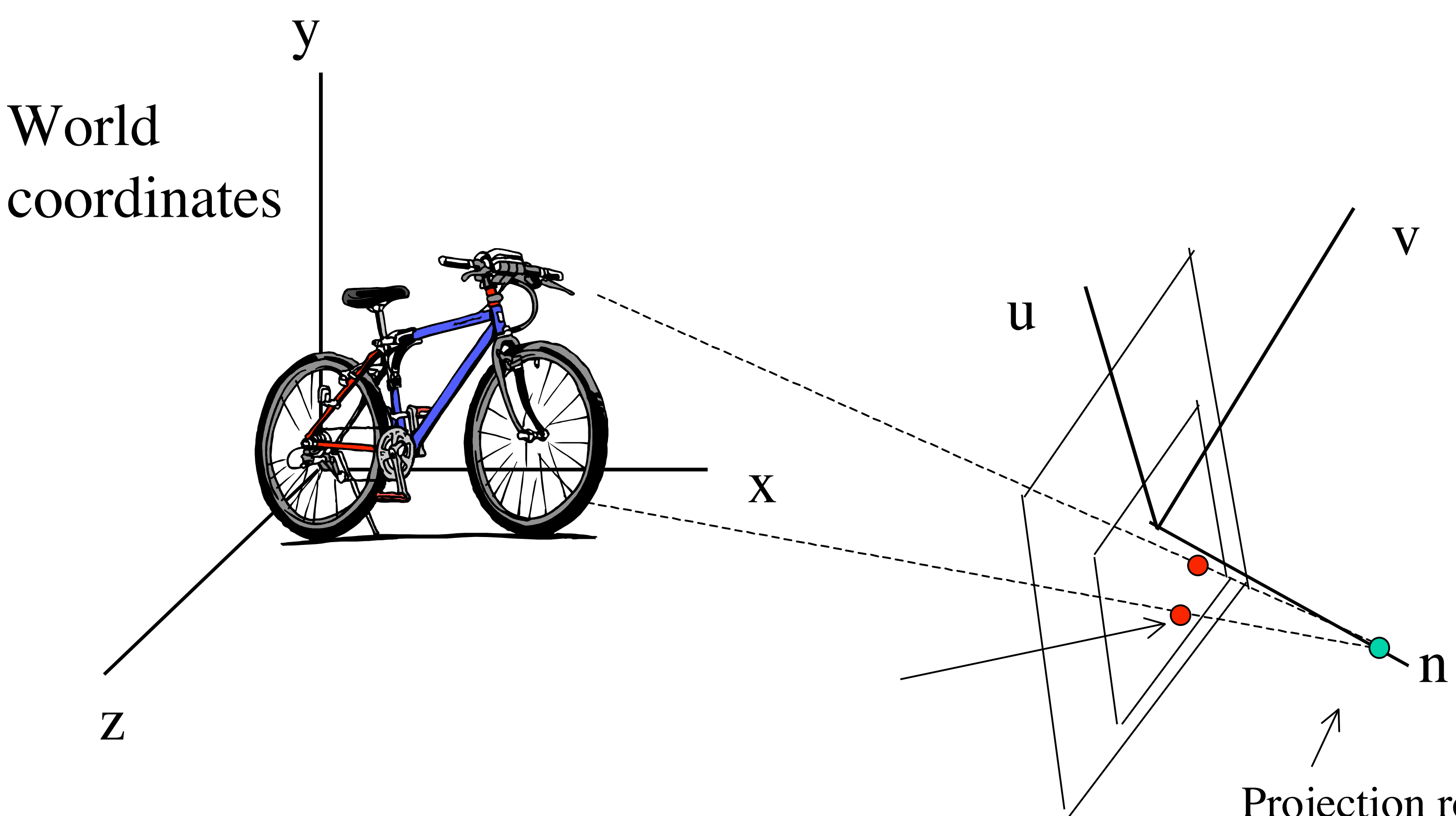
Specifying a camera

- Why use VUP?
  - Convenient for the user but there are other ways (OpenGL has several ways to negotiate camera parameters, including very much how we are doing it).
  - A world centric coordinate system is natural for the user. In particular, the user may think in terms of the camera rotation around the axis (**n**) relative to a natural horizon and/or "up" direction.
  - This will mean that VUP cannot be parallel to **n**. Often one will fix VUP (e.g. to the Y-axis) but this is too restrictive for some applications.
- Why use a "backwards" pointing **n**?
  - It is more natural to make the camera direction point the other way, but this makes the camera coordinates left handed. (You will see it done both ways).

Specifying a camera



Specifying a camera



Window given by (u_{min}, u_{max}), (v_{min}, v_{max}), denote center by CW

Projection reference point (PRP)