# Introduction à l'Informatique Graphique

## Lecture 3. Viewing in 3D

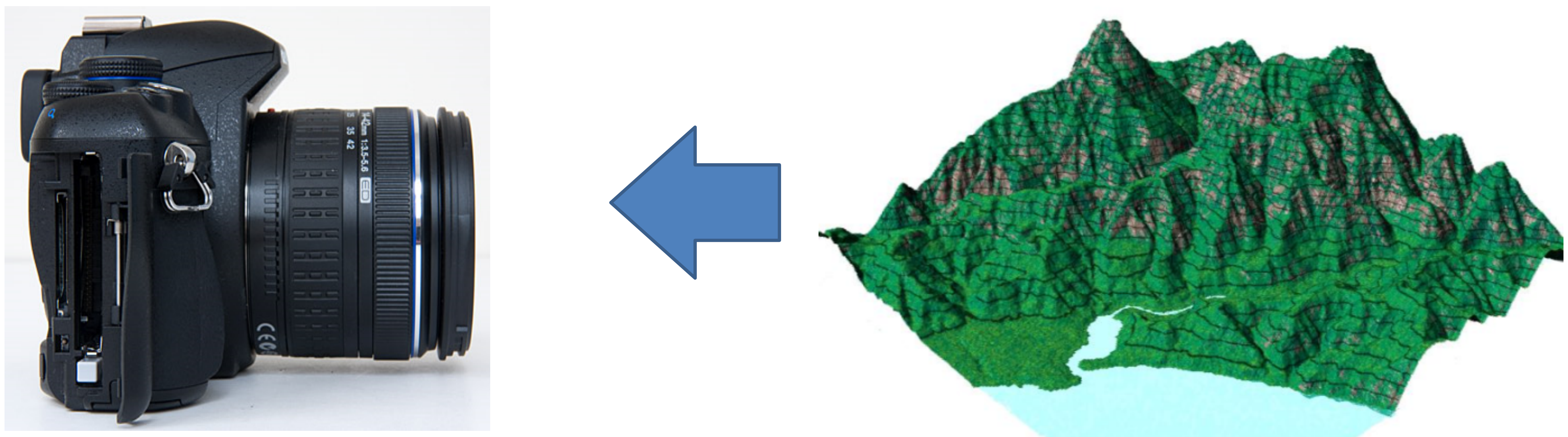
Caroline Larboulette

# Overview

- Camera
- Projections
- Specification of an Arbitrary 3D View
- Planar Geometric Projections
- 3D Transformations
- Coordinate Systems

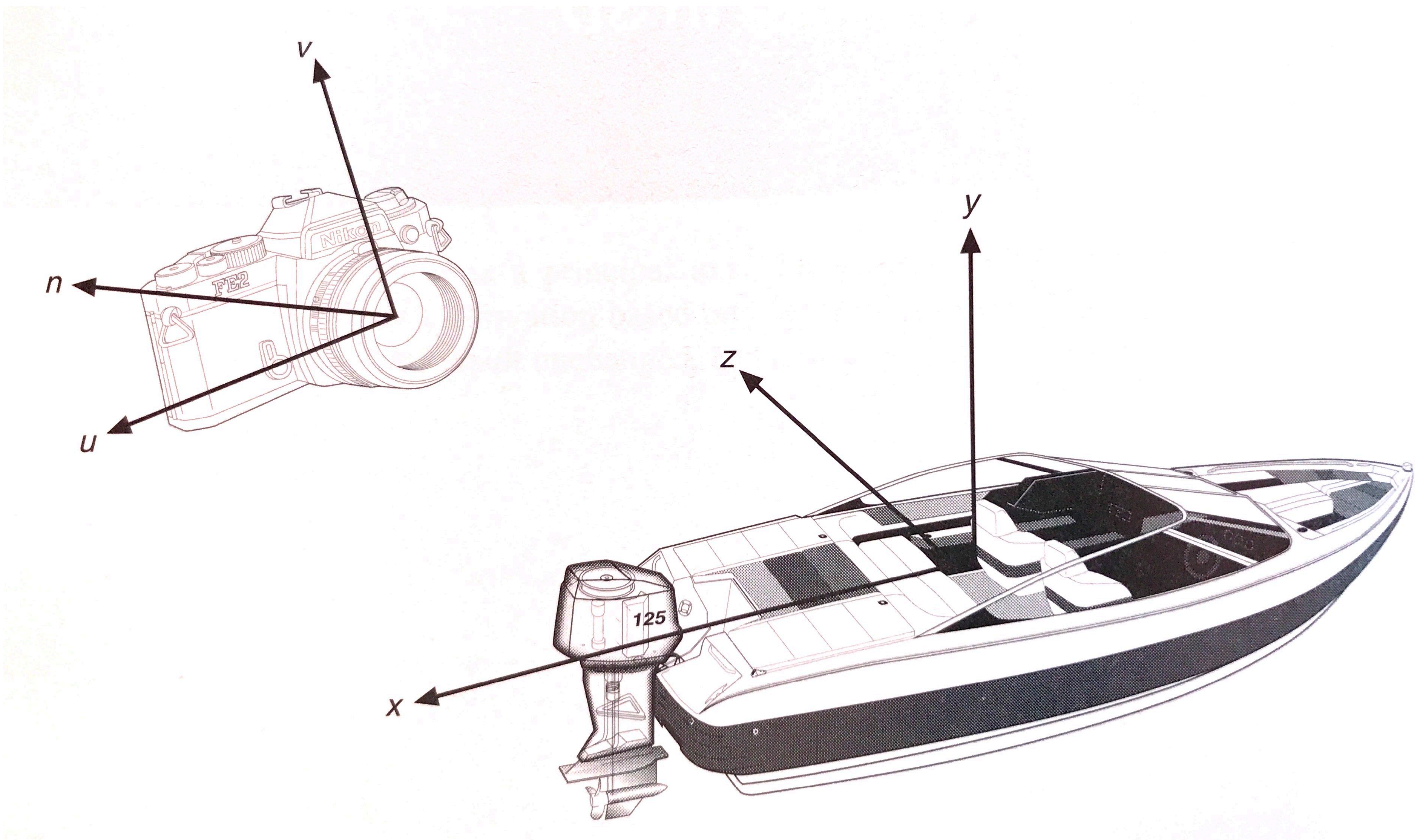
# Synthetic Camera
# &
# Steps in Viewing in 3D

# Camera and Scene



- 3D scene -> 2D medium (film)
- Synthetic Camera: programmer's model to specify how a 3D scene is projected onto a 2D screen

# Synthetic Camera



# Synthetic Camera

- Camera and 3D object have their own coordinate system
  - u,v,n for the camera
  - x,y,z for the object

# Steps for Creating an Image

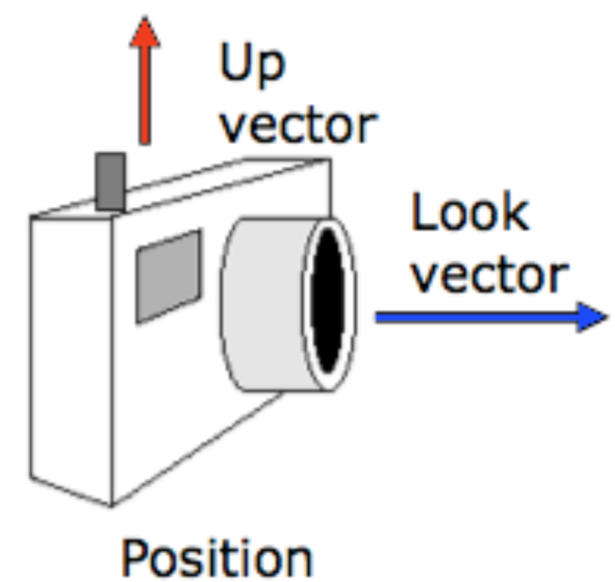
1. Specification of projection type (from 3D to 2D)
2. Specification of viewing parameters
3. Clipping in 3D
4. Projection and display

# Steps for Creating an Image

1. Specification of projection type (from 3D to 2D)
   - Perspective, parallel orthographic
2. Specification of viewing parameters
3. Clipping in 3D
4. Projection and display

# Steps for Creating an Image

1. Specification of projection type (from 3D to 2D)
2. Specification of viewing parameters
   - Viewing / Eye Coordinate System
     - Camera position
     - Camera orientation
     - Field of view (angle of view)
     - Depth of field / focal distance (near/far distance)
3. Clipping in 3D
4. Projection and display



# Steps for Creating an Image

1. Specification of projection type (from 3D to 2D)
2. Specification of viewing parameters
3. Clipping in 3D
    - View volume
        - Eliminates parts behind of too far
4. Projection and display

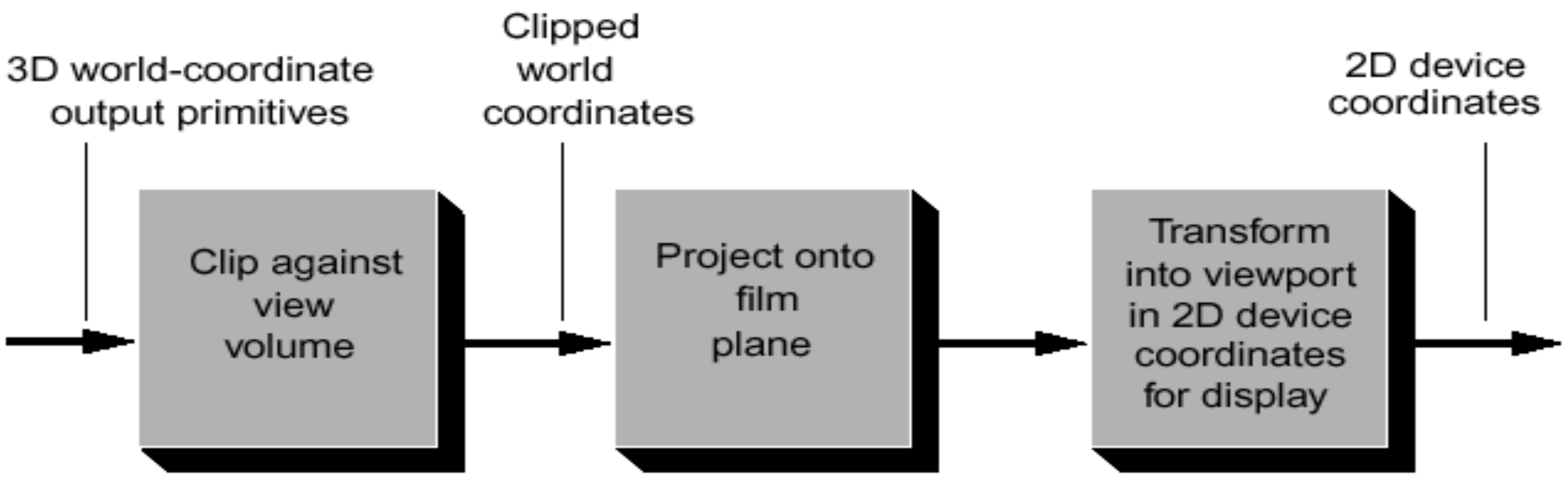
# Steps for Creating an Image

1. Specification of projection type (from 3D to 2D)
2. Specification of viewing parameters
3. Clipping in 3D
4. Projection and display
   - Window
     - Projection of the view volume onto the projection plane (window)
     - Transforming into viewport

# Window - Viewport

- Window = 2D clip rectangle on a 2D world coordinate drawing
- Viewport = rectangular area of screen where the scene is rendered (2D integer coordinates / pixels)
- Clipped window contents are mapped to screen viewport

# 3D Viewing Process



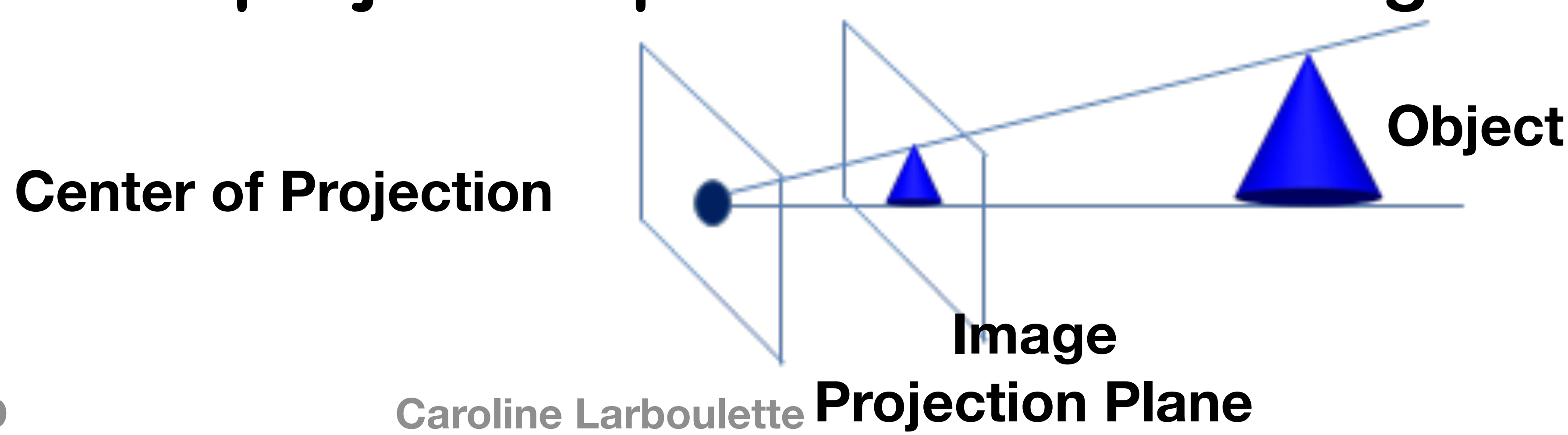
# Projections

# Projection: Definition

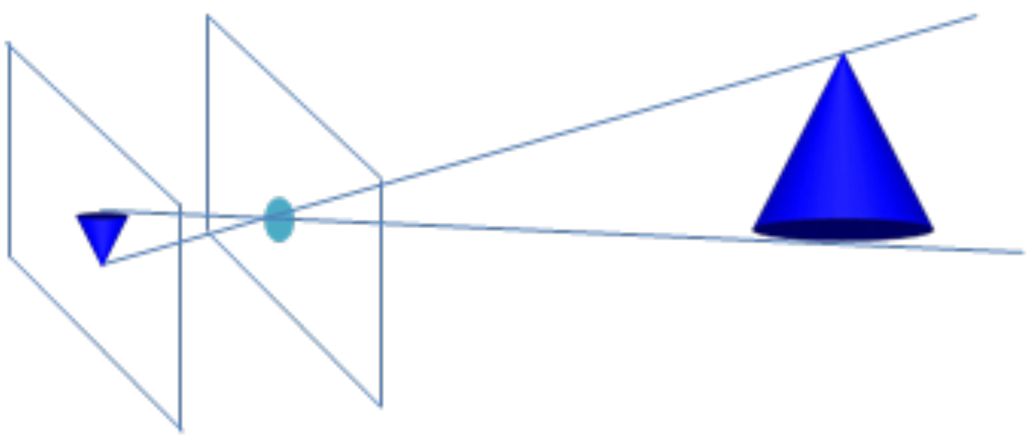
- A projection transforms points in a coordinate system of dimension $n$ into points in a coordinate system of dimension less than $n$
- From $3D$ to $2D$

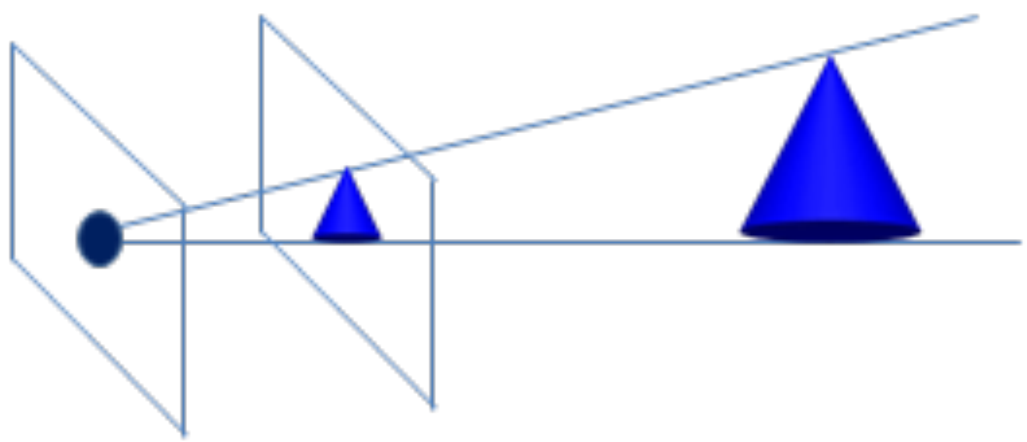
# Center of Projection

- Projection rays are straight and are called projectors
- Projection rays
  - emanate from center of projection PRP (viewpoint)
  - pass through each point of the 3D model
  - intersect the projection plane where the image forms



# Real vs. Synthetic Camera

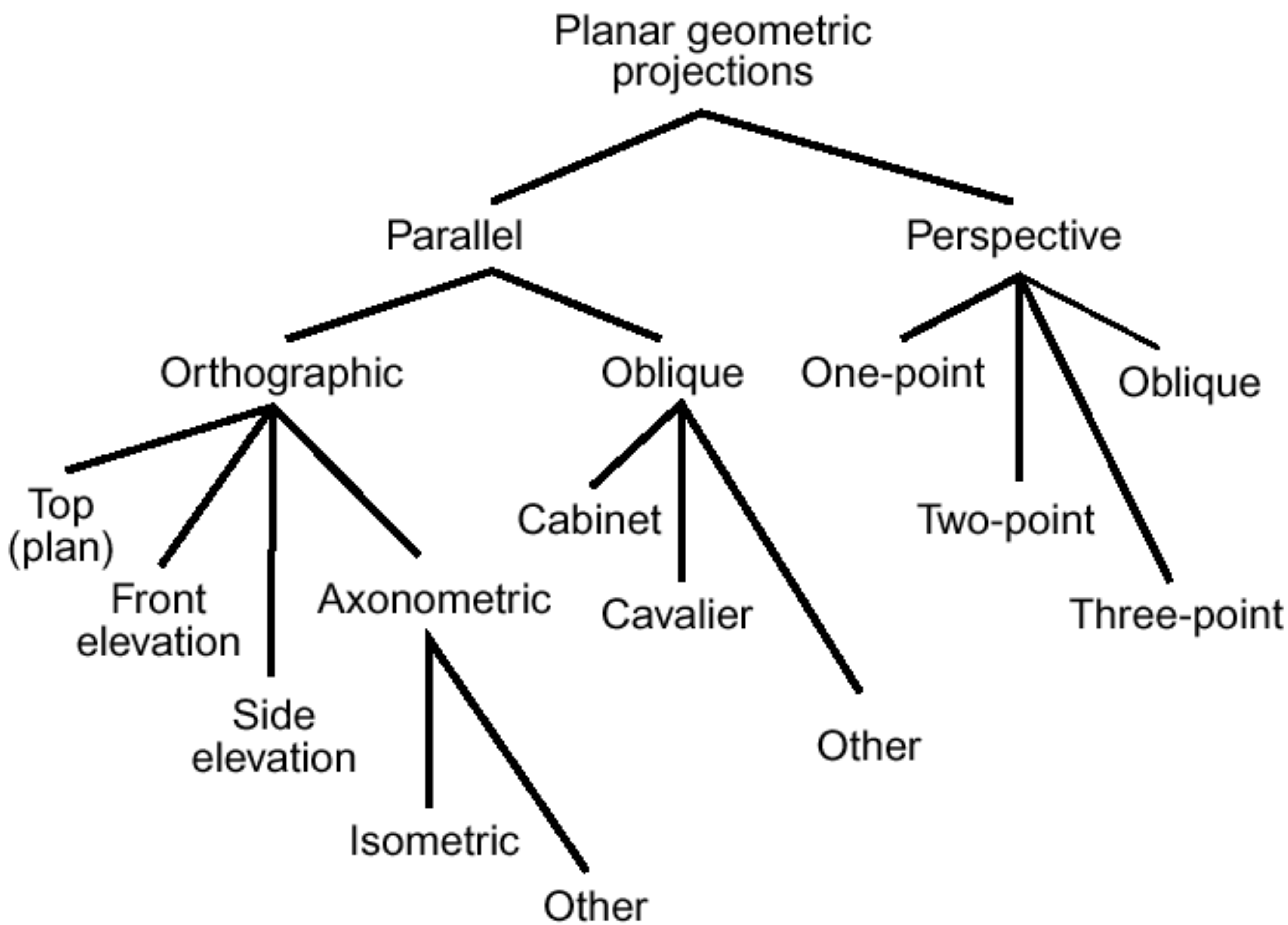




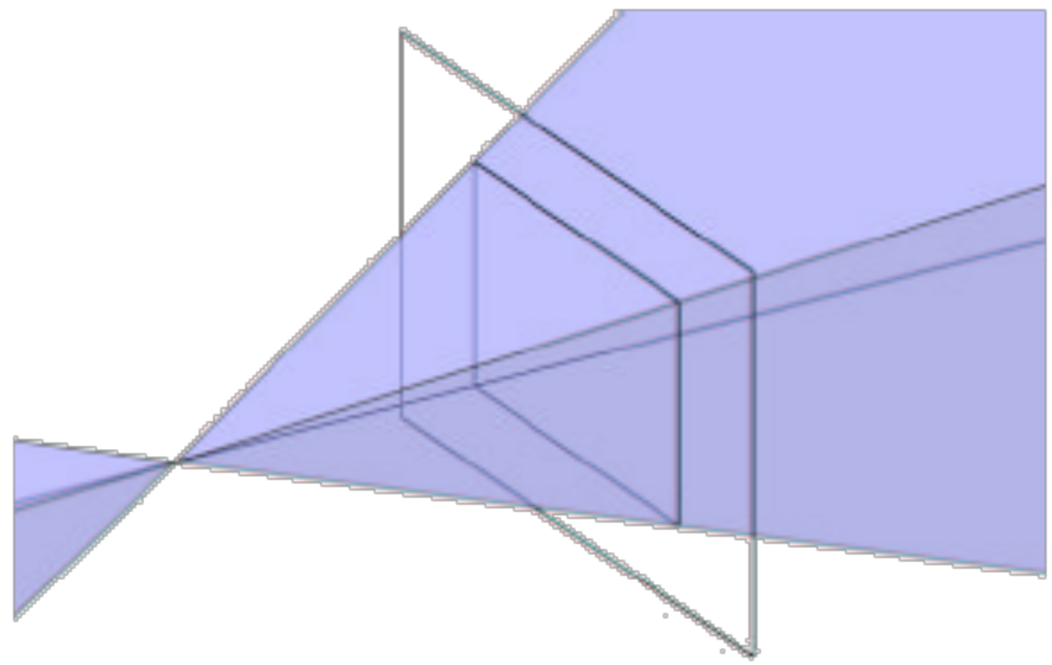
# Planar Geometric Projection

- Projection onto a plane (not a curved surface)
- Straight projectors (not curved)
- Cartographic projections are usually non-planar or non-geometric
- Planar geometric projections
  - Perspective (projectors converge)
  - Parallel (projectors are parallel)

# Types of Projections

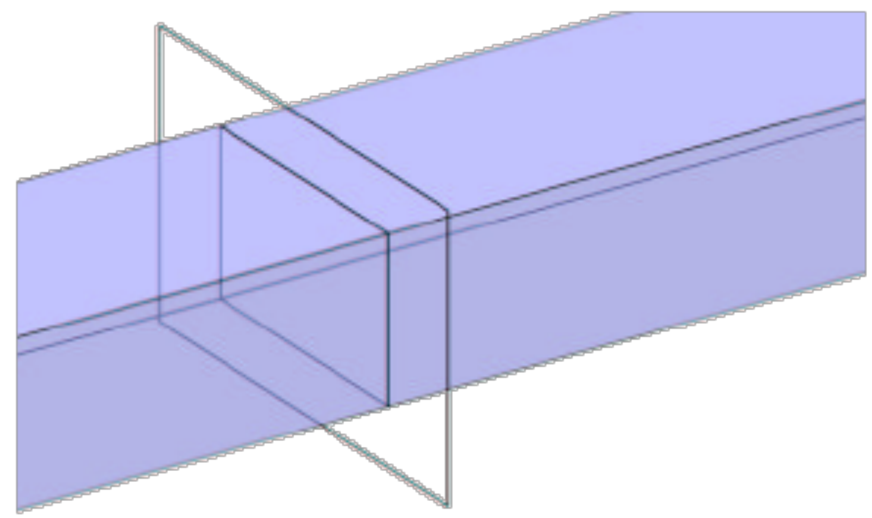


# Perspective Projection



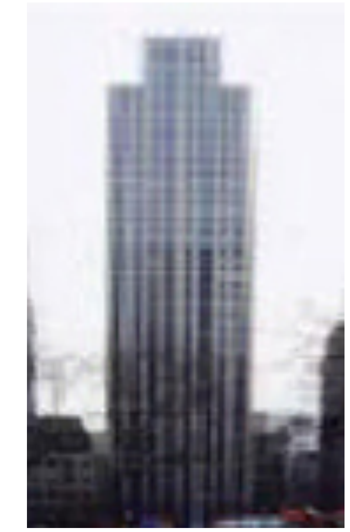
- Distance from center of projection to projection plane is finite
- Projectors converge into center of projection

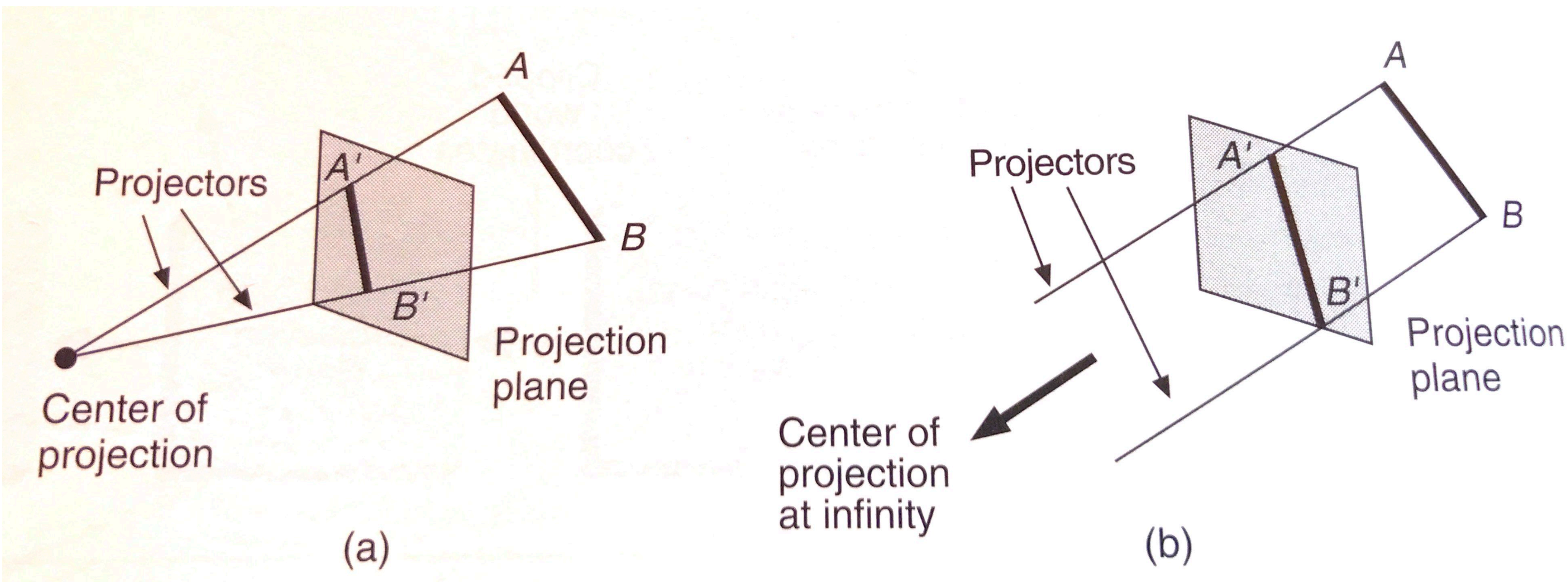
# Parallel Projection



- The center of projection is infinitely far away
- Projectors are parallel
- A perspective projection whose center of projection tends to infinity becomes a parallel projection

# Planar Geometric Projections





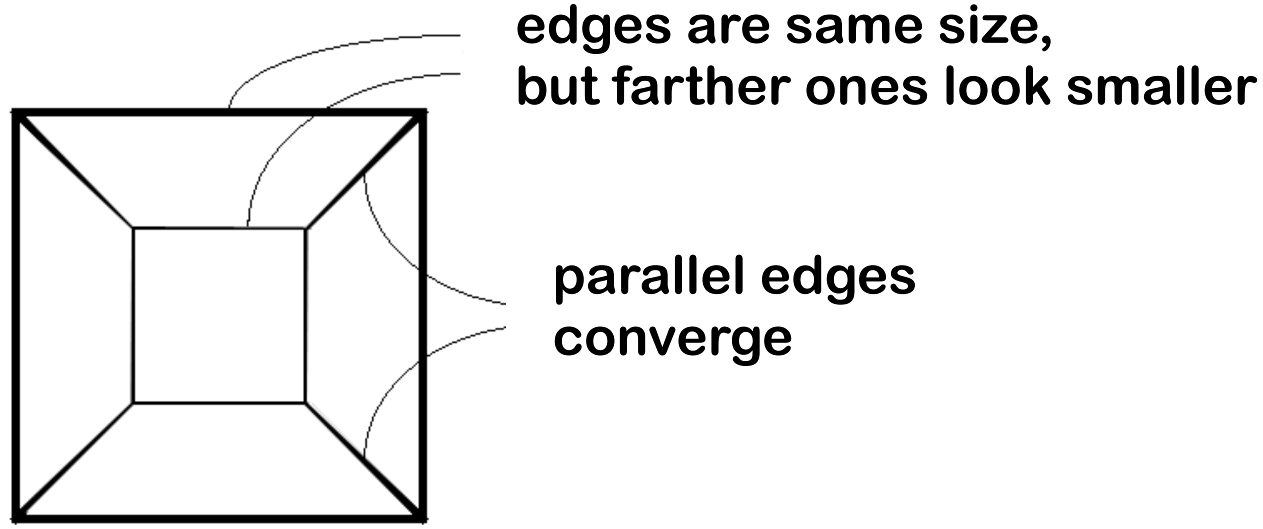

**Perspective**
**Defined by center of projection**
**PRP has coordinate (x,y,z,1)**
**Direction of projection**

$d = (x,y,z,1)-(x',y',z',1) = (a,b,c,0)$

**Parallel**
**Defined by direction of projection**

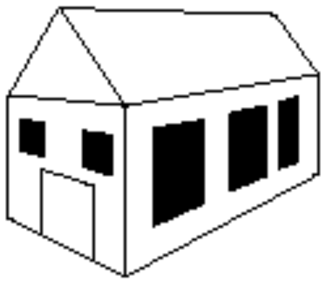
$d : (a,b,c,0)$

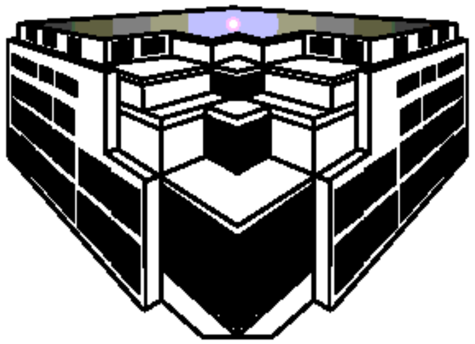
# Perspective Projections




- Parallel lines, that are not parallel to the projection plane converge to a vanishing point
- Further objects are smaller (size inverse of distance)
- Used in Computer Graphics, Art, Human visual system

# Perspective Projections

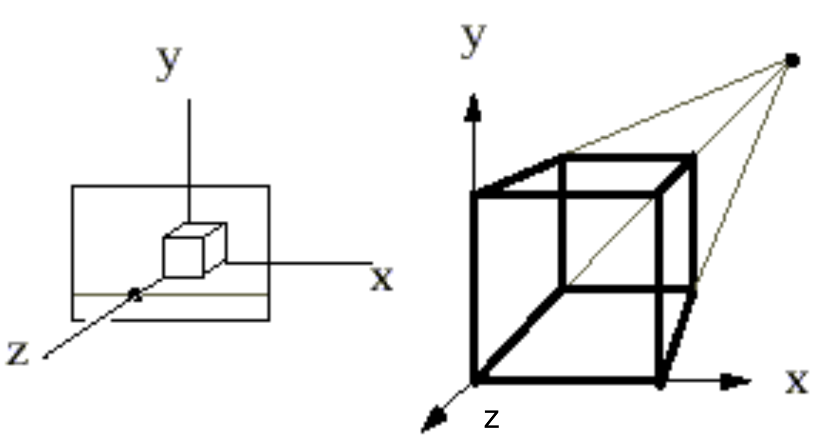




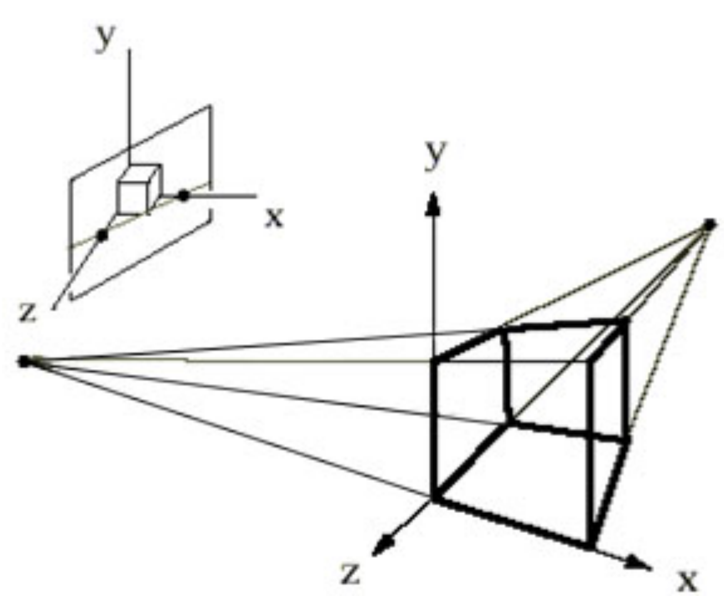
- Advantage
  - Gives a realistic view and feeling of 3D form of objects
- Drawback
  - Does not preserve shape or scale

# Perspective Projections

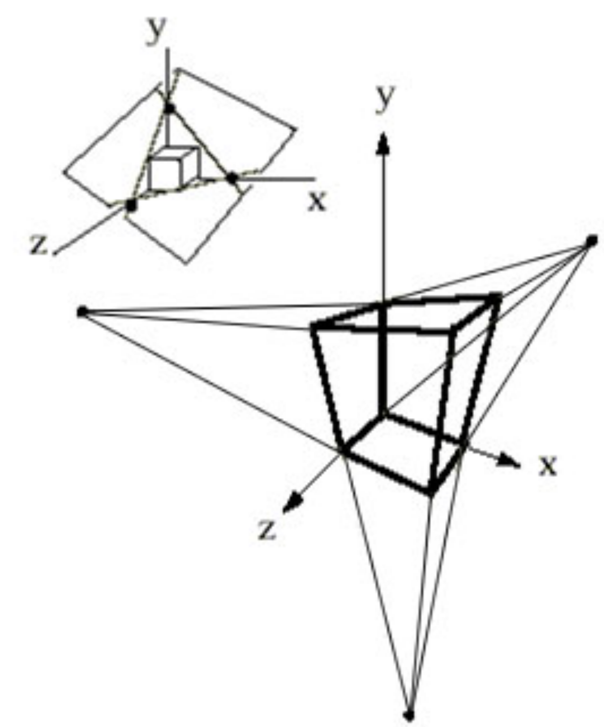
- If set of parallel lines is parallel to an axis, they converge into an axis vanishing point
- Maximum of 3 vanishing points ($x, y, z$ coordinate axes)
- Number of vanishing points equals number of principal coordinate axes intersected by projection plane




One Point Perspective
(z-axis vanishing point)



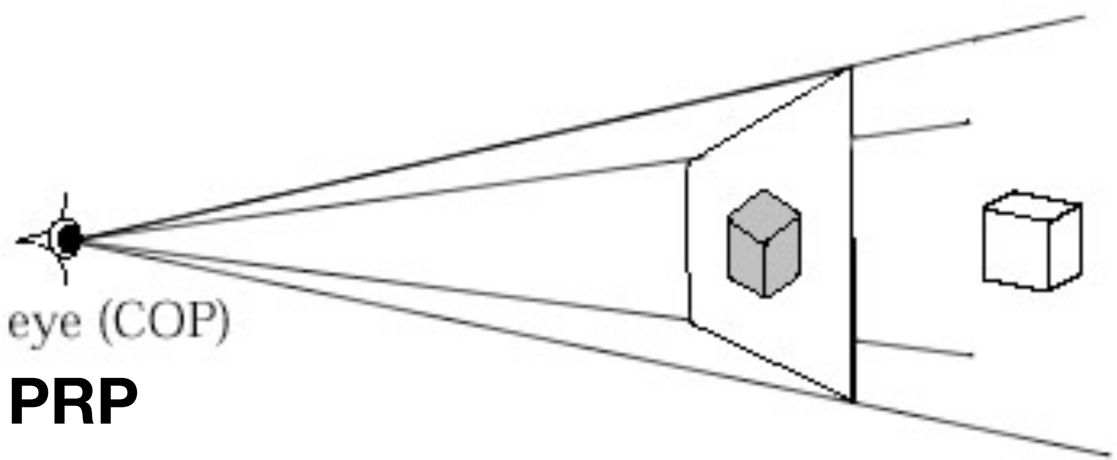

Two Point Perspective
(z and x-axis vanishing points)

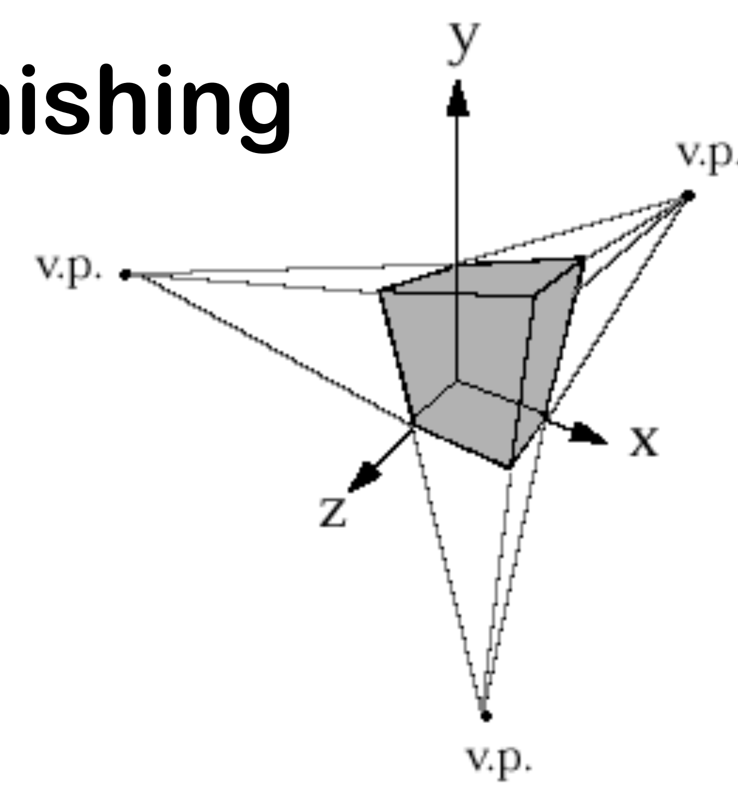



Three Point Perspective
(z, x, and y-axis vanishing points)

# Perspective Projections

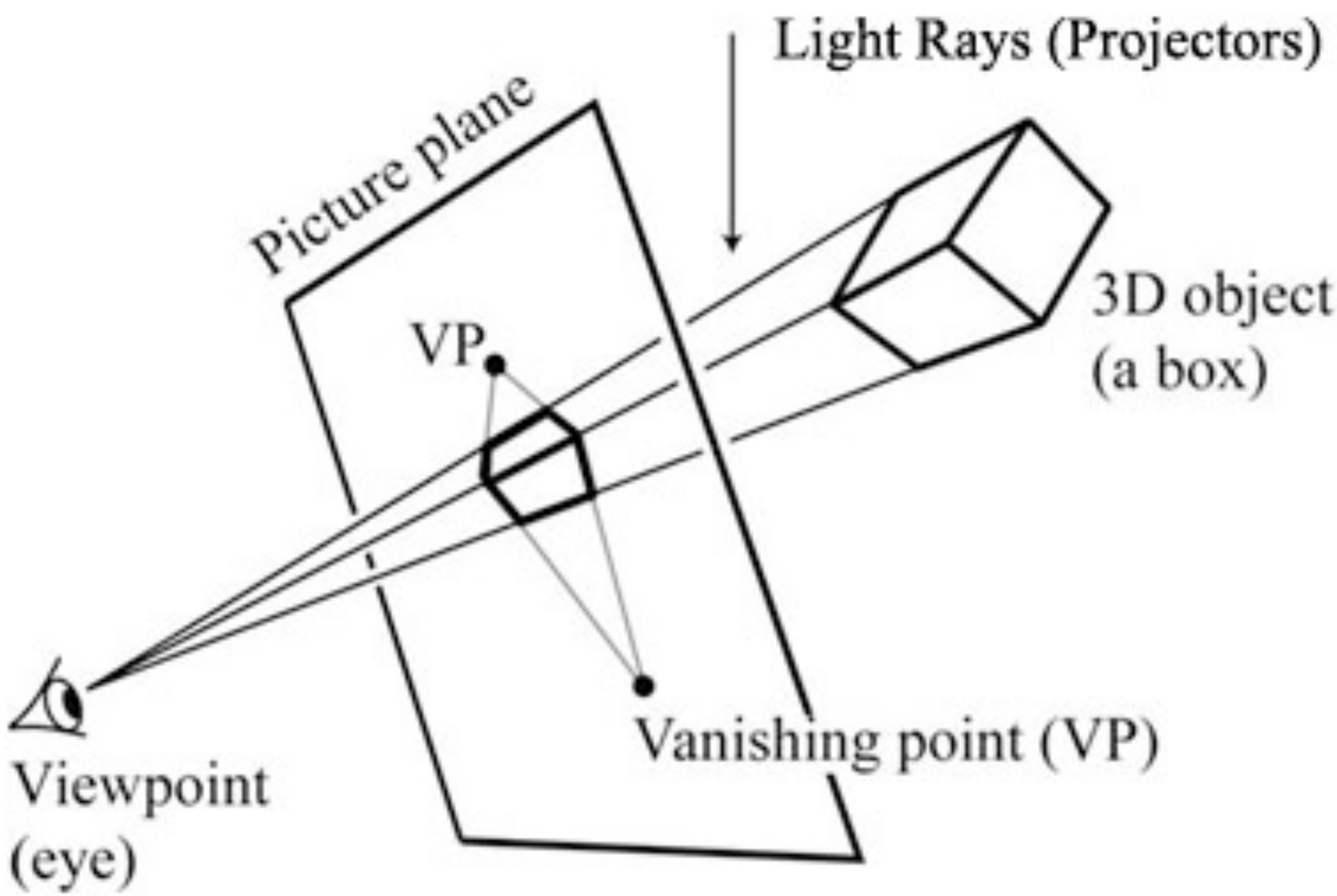
- Parallel lines converge towards vanishing points



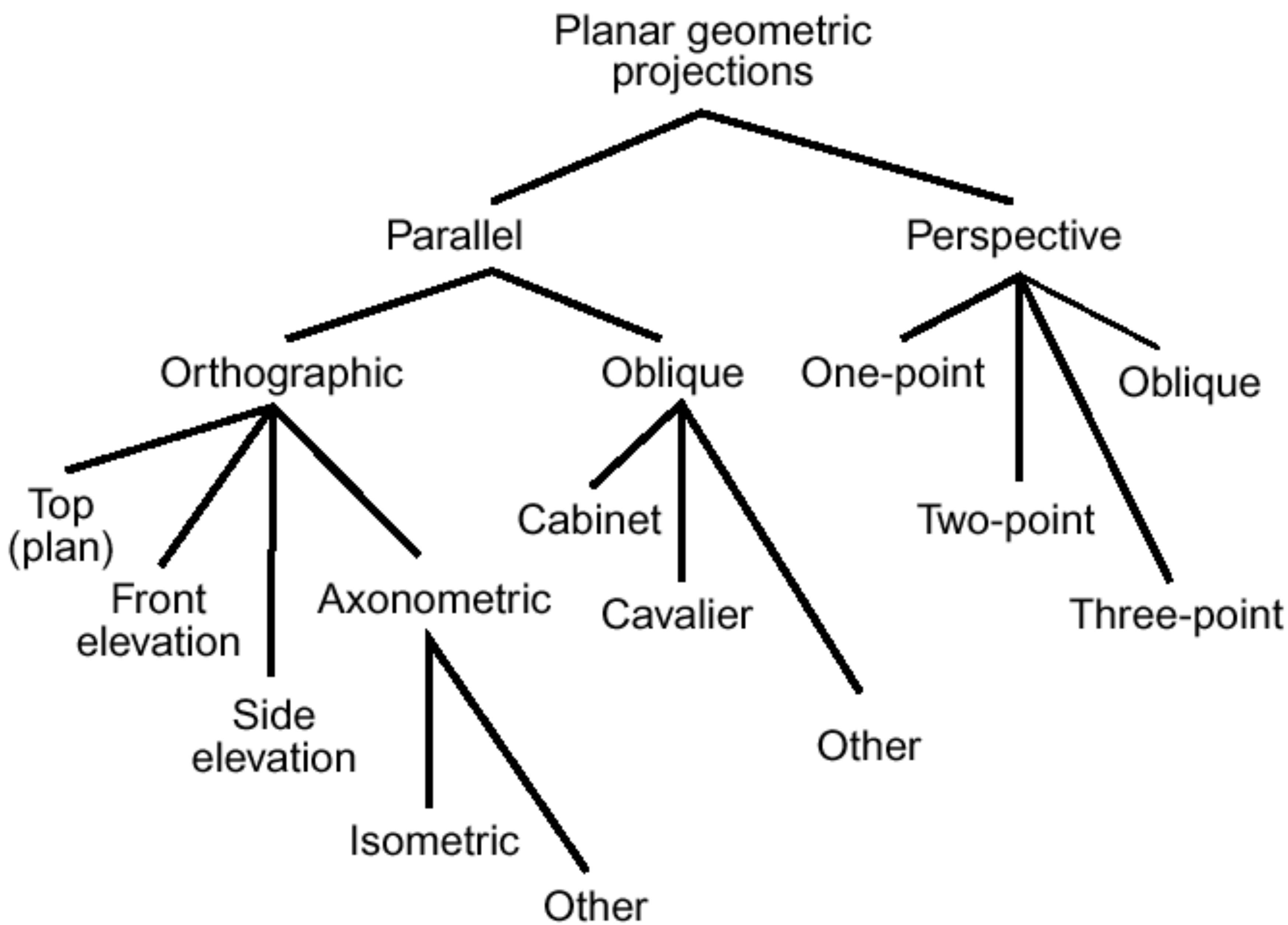




- Perspective image is the result of the intersection of projectors (from object to viewpoint / eye) with image plane

# Perspective Projections

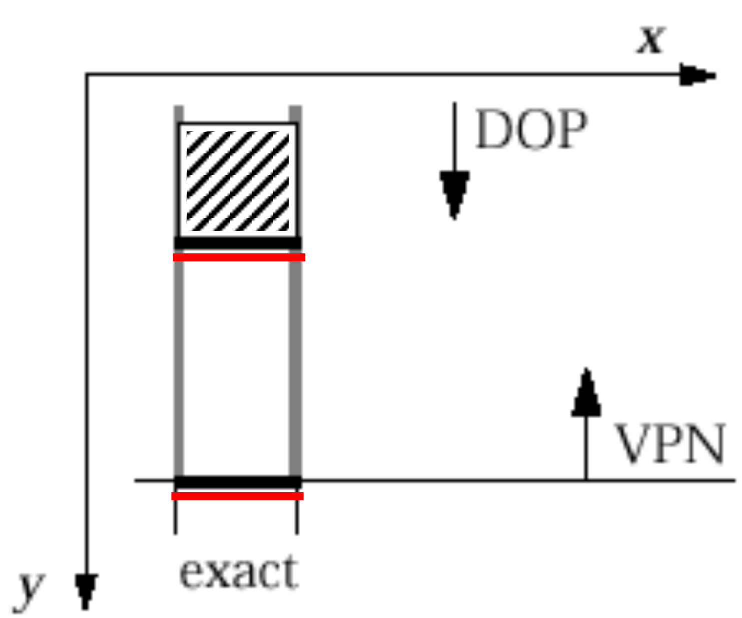


# Types of Projections

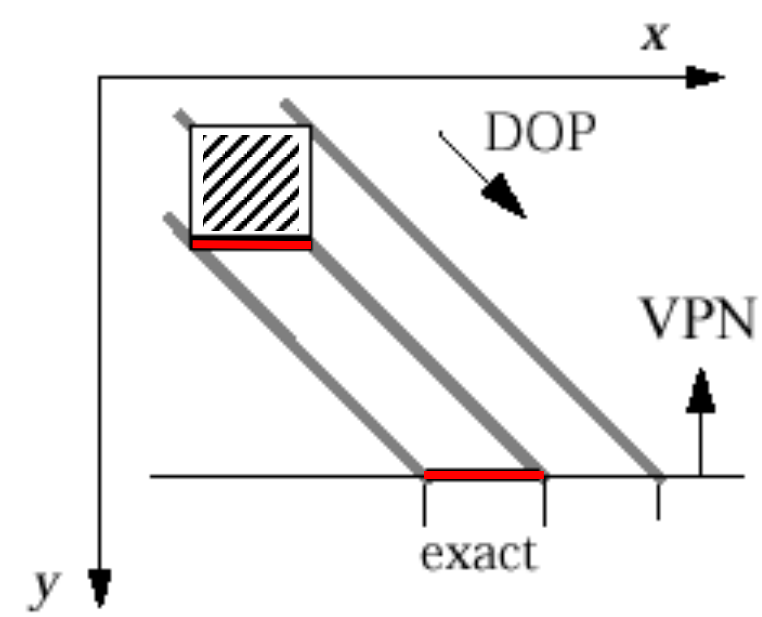


# Parallel Projections

- Two types
  - Orthographic : direction of projection and normal to projection plane are collinear (DOP || VPN)
  - Oblique : direction of projection is not normal to the projection plane



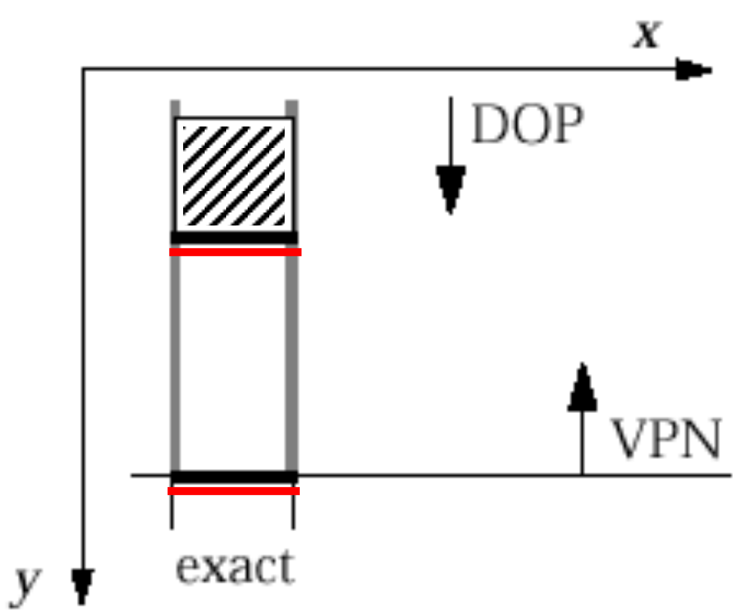

Orthographic

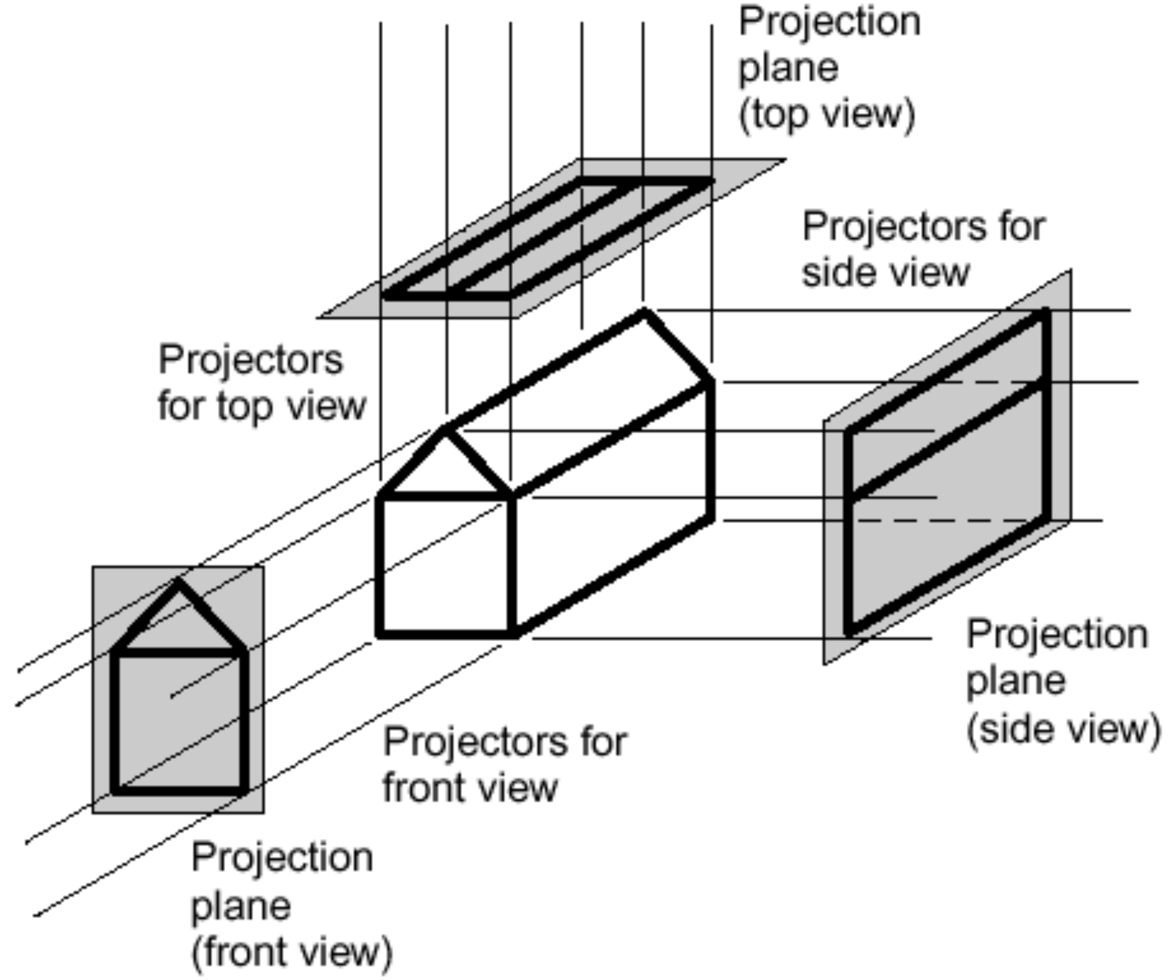



Oblique

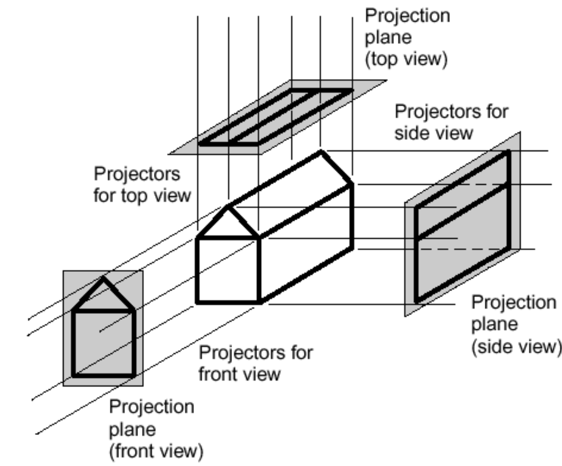
# Orthographic Projections

- Projection from front, top and side views



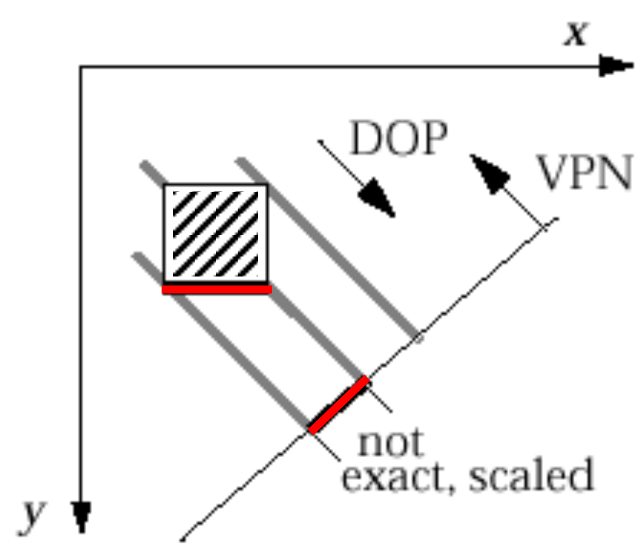
# Orthographic Projections

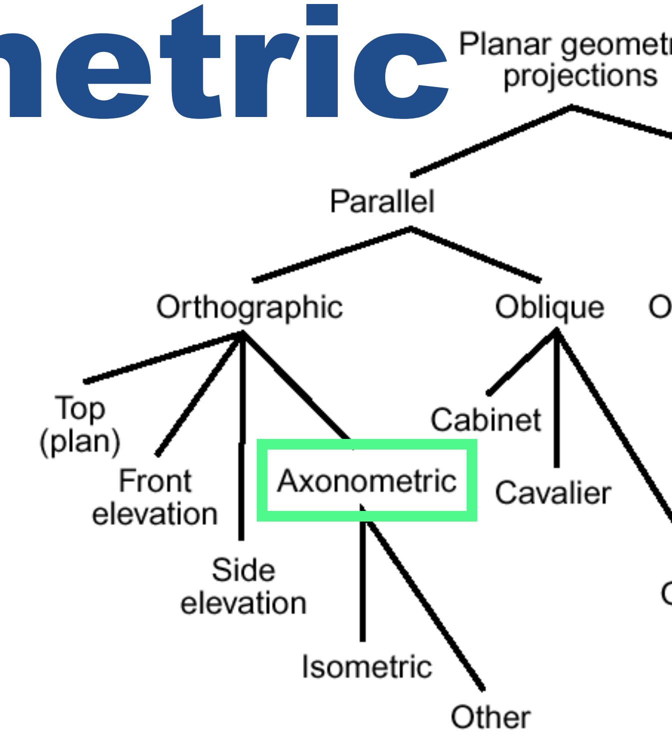



- Used in engineering, architecture
- Advantages
  - Accurate measurement (segment size are conserved)
  - All views are the same scale
- Drawbacks
  - Not realistic, no real 3D feeling
  - Often need multiple views to get the 3D feeling for object

# Orthographic Axonometric

**DOP || VPN**

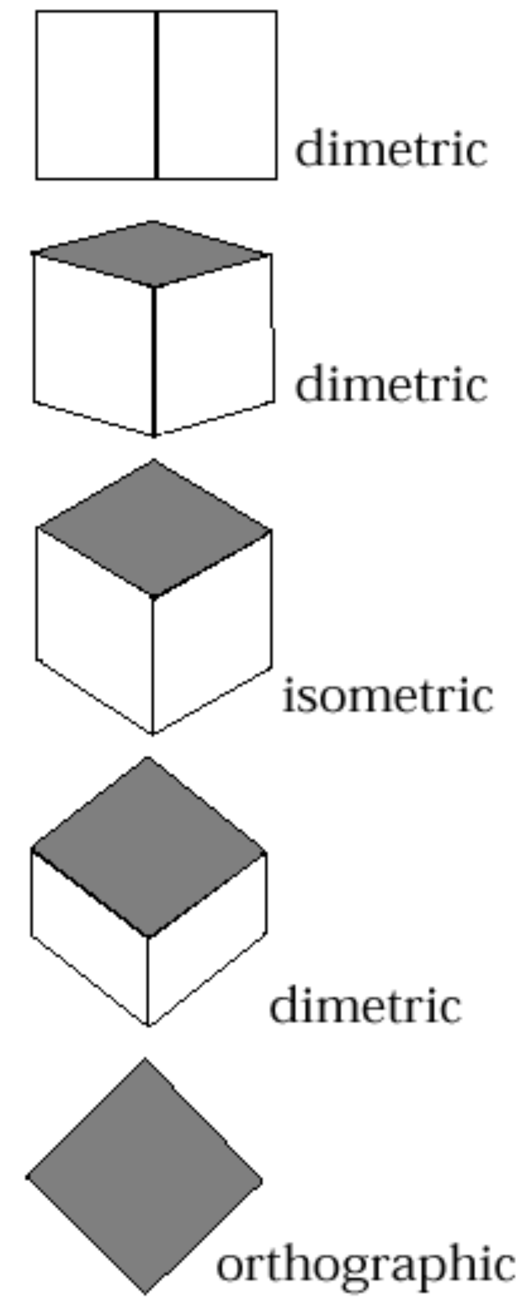



- Projection plane is not normal to any of the axes
- Shows several faces of an object at once
- Parallelism of lines is preserved (angles are not)
- Foreshortening is uniform (not related to the distance from the center of projection — that is at infinite distance)

# Orthographic Axonometric

- Isometric (direction of projection makes equal angle with each principal axis)
  - 120 degrees
  - Same scale ratio applies along each axis
- Dimetric (equal angle with 2 axes)
  - 2 scale ratios
- Trimetric (all angles are different)
  - 3 scale ratios

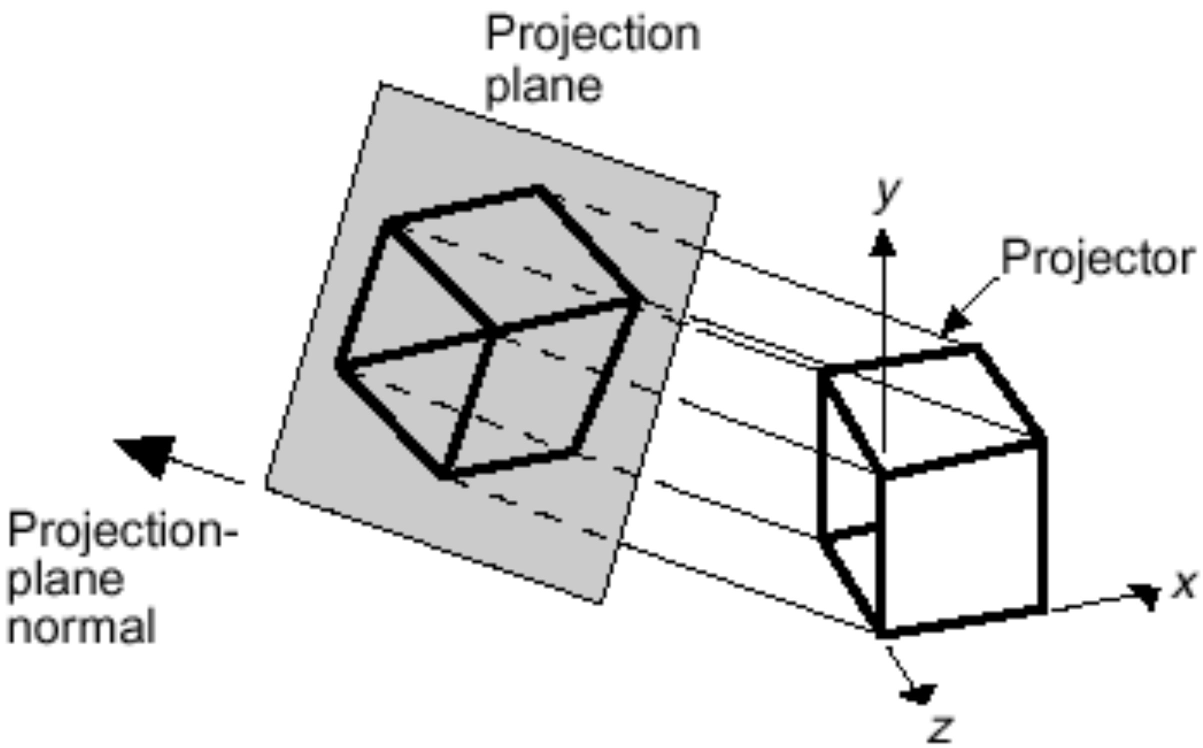


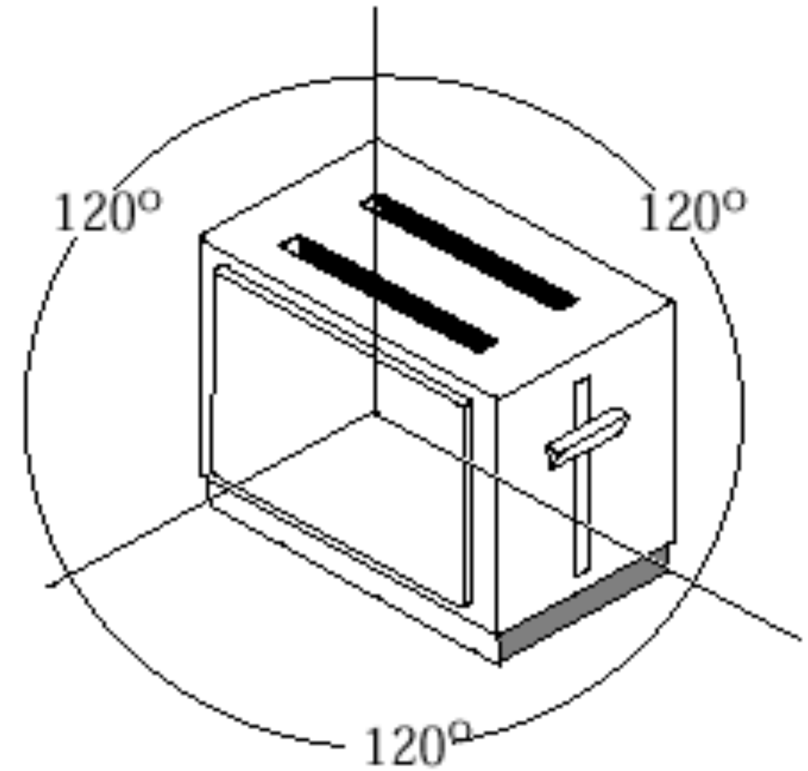

Carlbom Fig.3-8

# Isometric Projection

- Used for catalog illustrations, furniture design, structural design
- Advantages
  - Don’t need multiple views
  - Measurements can be made along principal axes
- Drawbacks
  - Distorted appearance
  - Better suited for rectangular (not curved) shapes

# Isometric Projection




Example

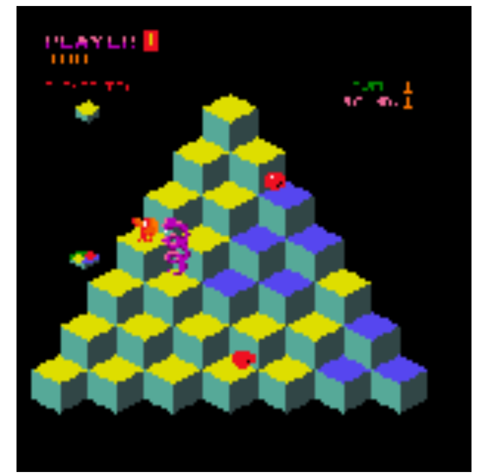
Carlbom Fig.2.2

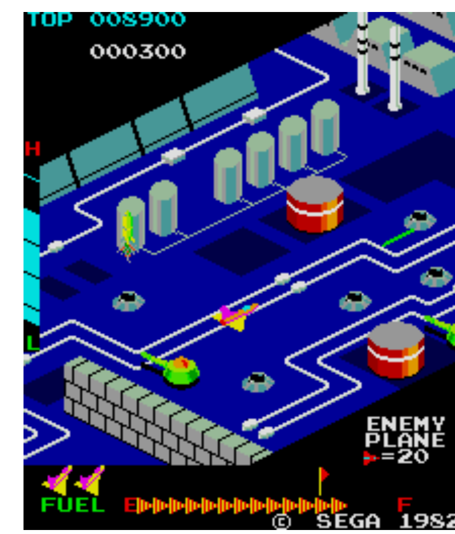
**Construction of an isometric projection of a unit cube: projection plane cuts each principal axis by 45°**

**8 possible projection plane directions: on the picture, direction of projection is (-1, 1, 1), plan normal is (1,-1,-1)**

# Axonometric Projection in Games

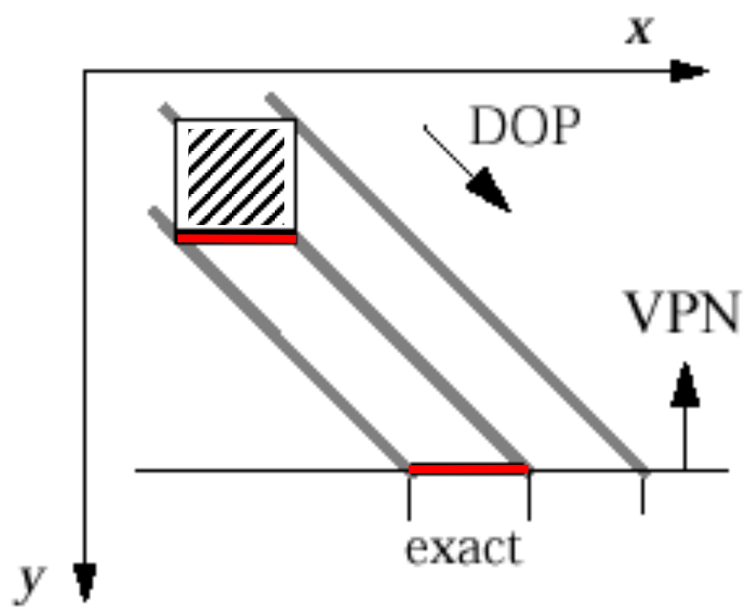
- Video games have been using isometric projection for ages.
  - It all started in 1982 with *Q*Bert* and *Zaxxon* which were made possible by advances in raster graphics hardware.
- Still in use today when you want to see things in distance as well as things close up (e.g. strategy, simulation games).
  - StarCraft II, Transistor
- While many games technically use axonometric views, the general style is still referred to isometric or, inappropriately, "2.5D"/ "three quarter".

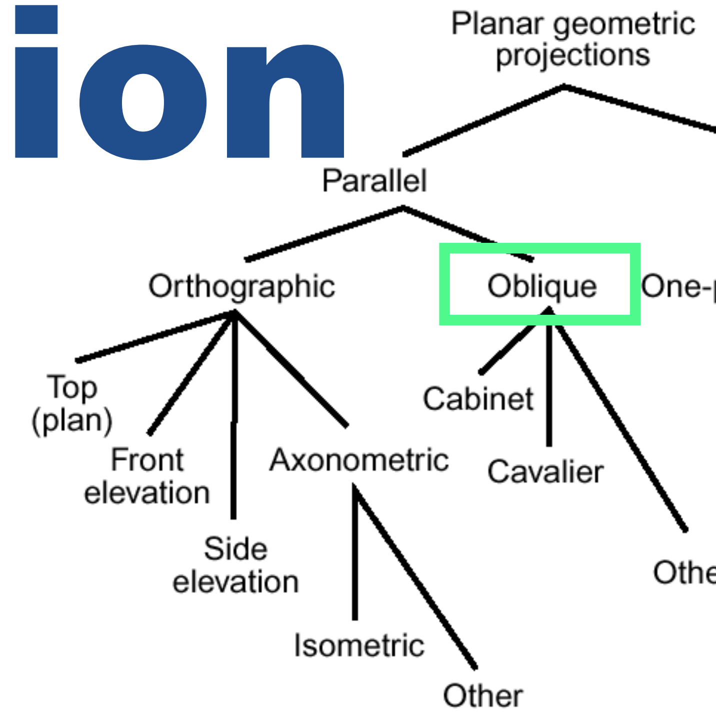





9/22/2015

# Oblique Projection



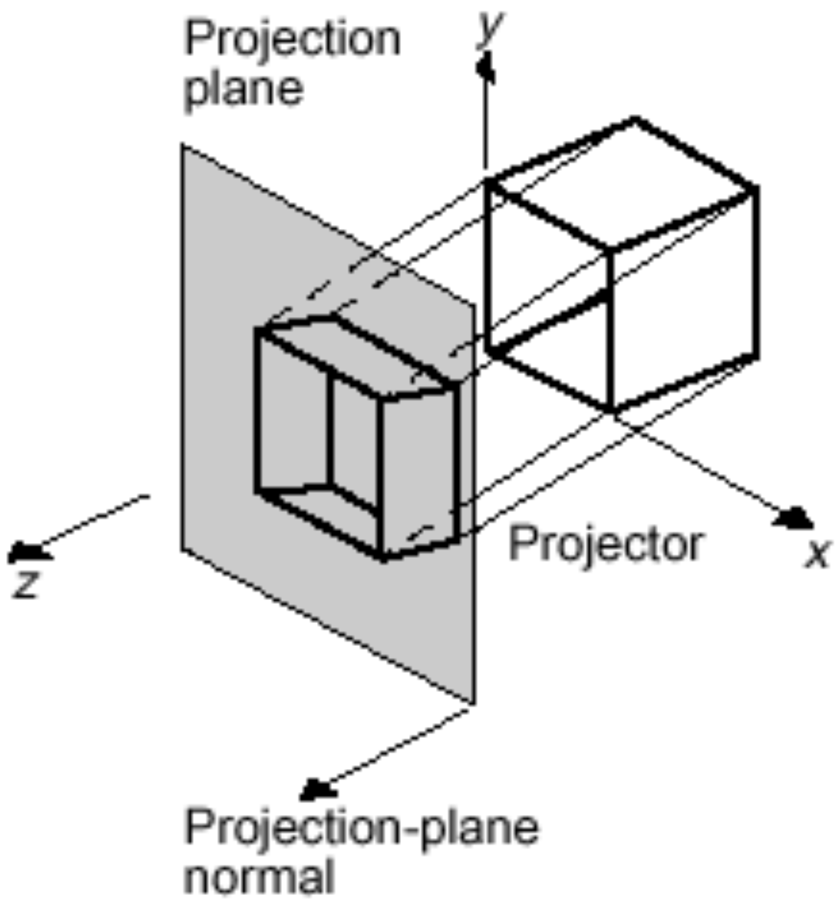




- Normal to projection plane and direction of projection differ
- Combines the properties of orthographic multi-view projections and axonometric projection
  - Projection plane is normal to an axis
  - Shows several faces of an object

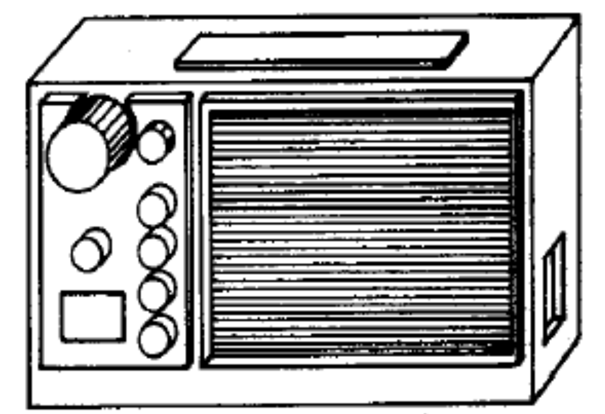
# Oblique Projection

- Advantages
  - Allows for measurement of
    - distances and angles of one face of the object
    - distances but not angles of the other faces
  - Gives some hint of 3D (several faces)
- Drawbacks
  - Objects can look distorted if projection plane is not well-chosen
  - No foreshortening (unrealistic)

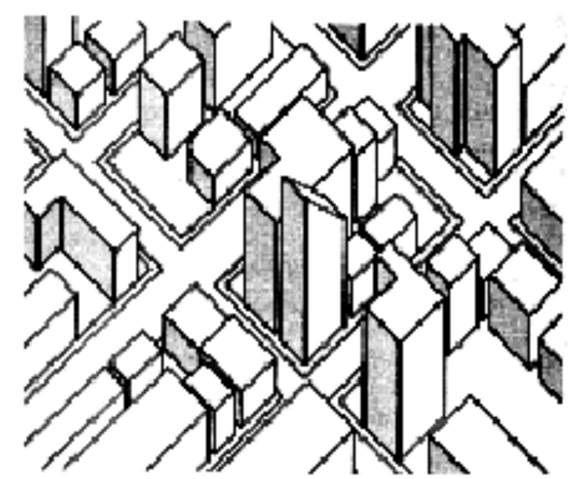
# Oblique Projection




**Construction of oblique parallel projection DOP and VPN differ**



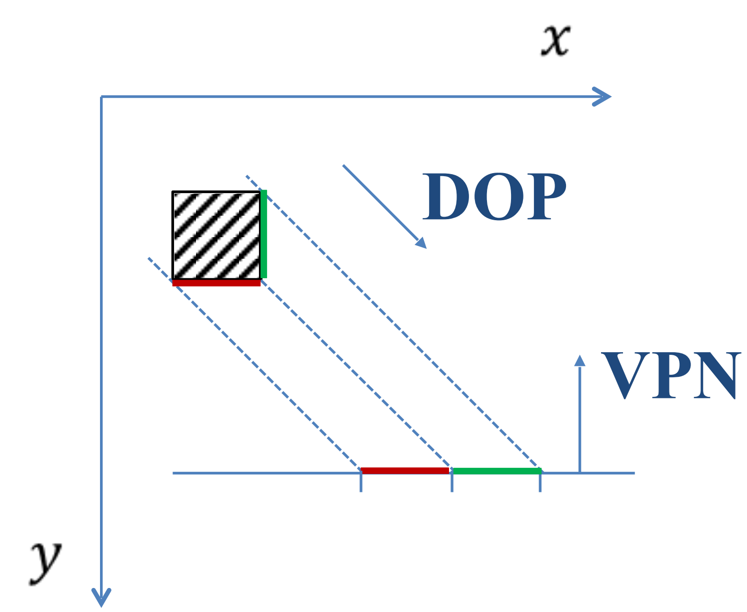
Front oblique projection of radio
(Carlbom Fig. 2-4)



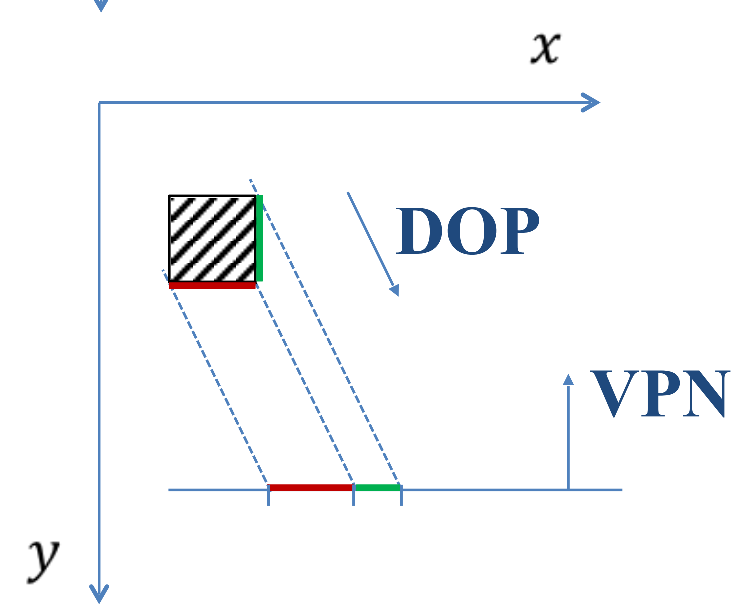
(Carlbom Fig. 2-6)
Plan oblique projection of city

## Main Types of Oblique Projections

- *Cavalier:* Angle between projectors and projection plane is 45. Perpendicular faces projected at full scale.

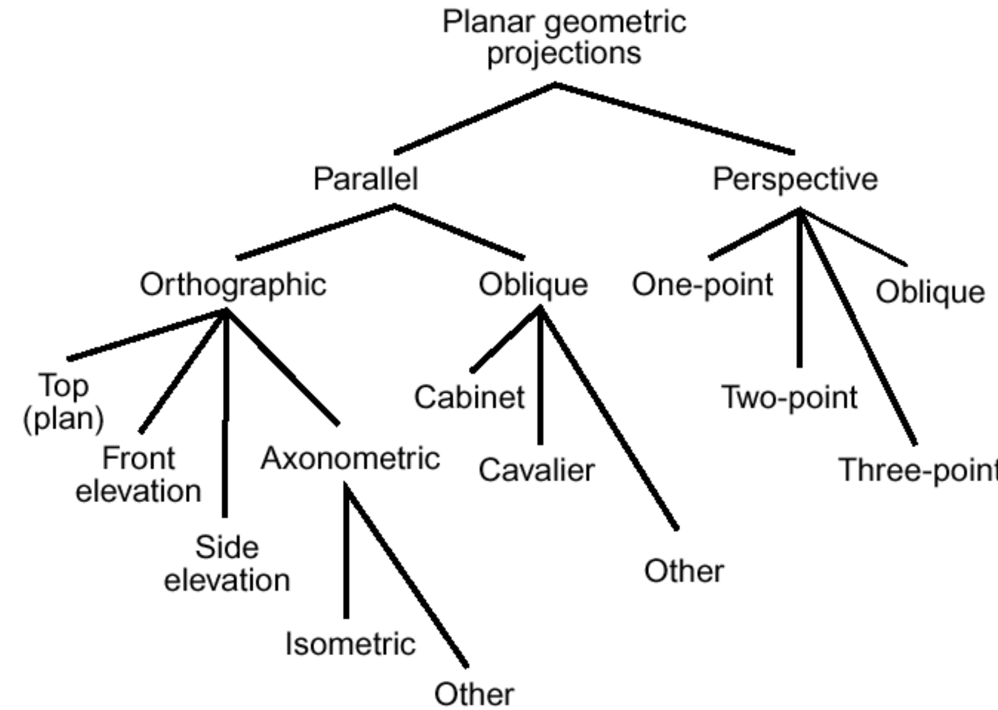


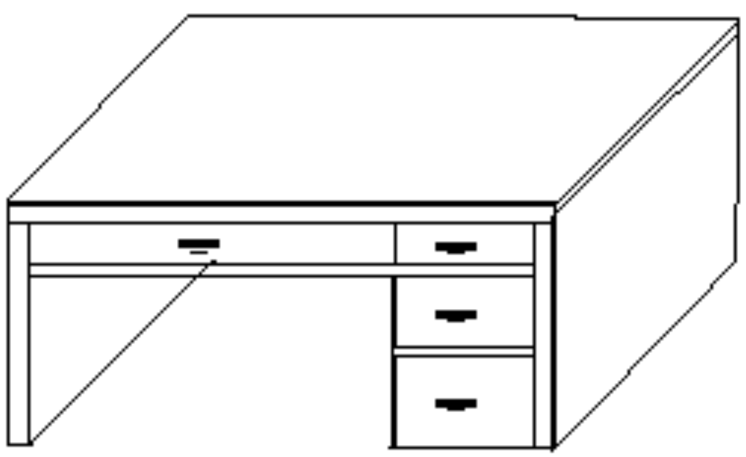

- *Cabinet:* Angle between projectors and projection plane: $tan^{-1}(2) = 63.4^o$. Perpendicular faces projected at 50% scale



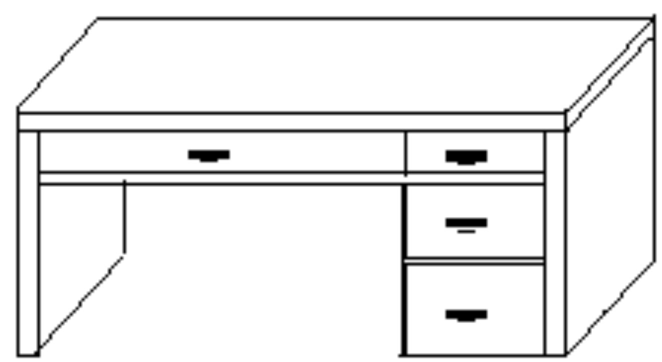

9/22/2015

# A Desk in Parallel Projection



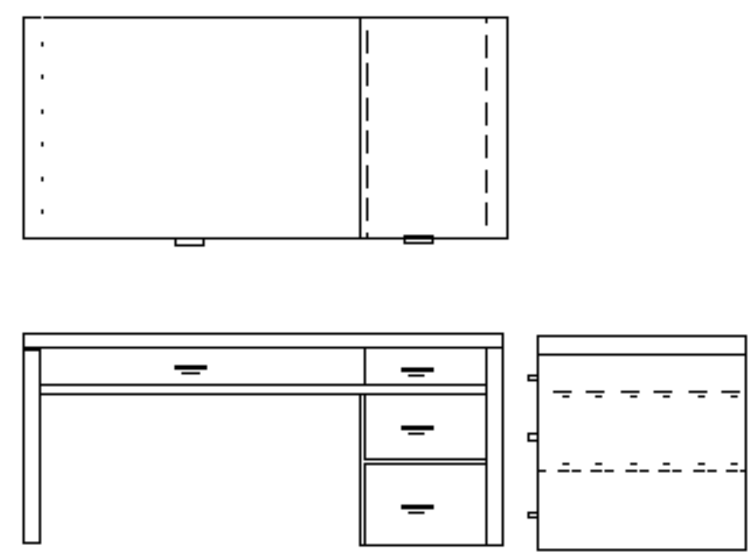



**Parallel Oblique Cavalier**



**Parallel Oblique Cabinet**

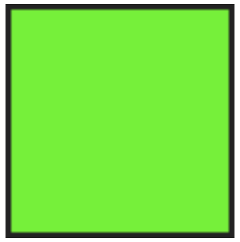
Carlbom Fig. 3-2

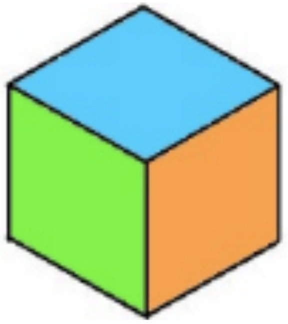


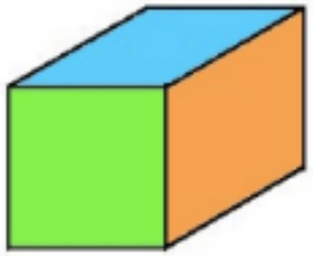
**Parallel Orthographic Multiview**

# Summary

- Three main types of parallel projections:
  - **Orthographic (multiview)**: projectors orthogonal to projection plane, single face shown
  - **Axonometric**: projection plane rotated relative to principle axes, reveals multiple faces
  - **Oblique**: projectors intersect projection plane at oblique angle, revealing multiple faces, often more skewed representation, with a plane of interest undistorted

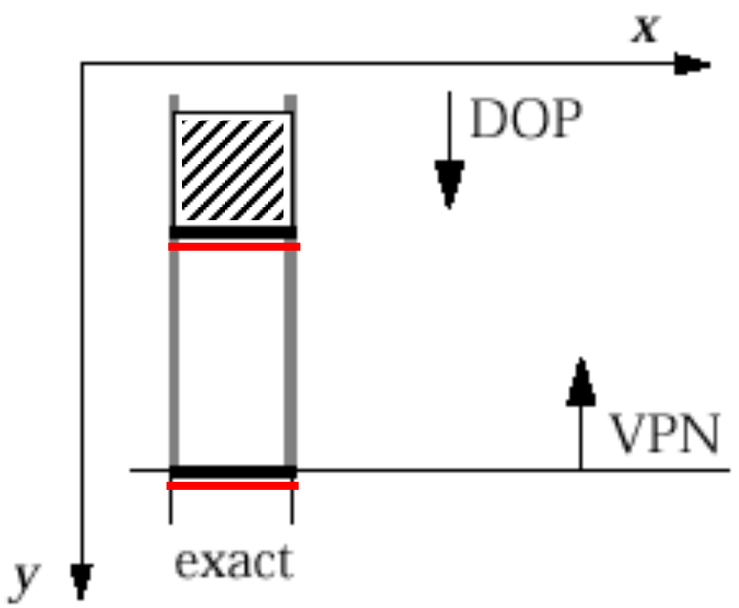






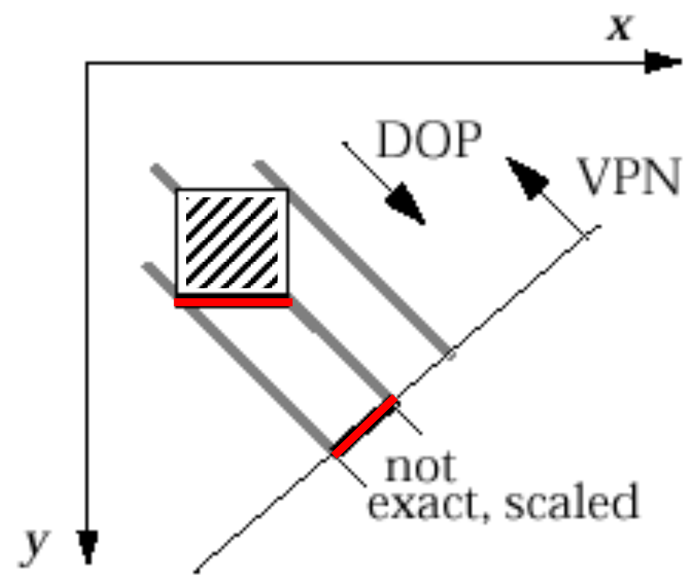
# Parallel Projections

## Summary



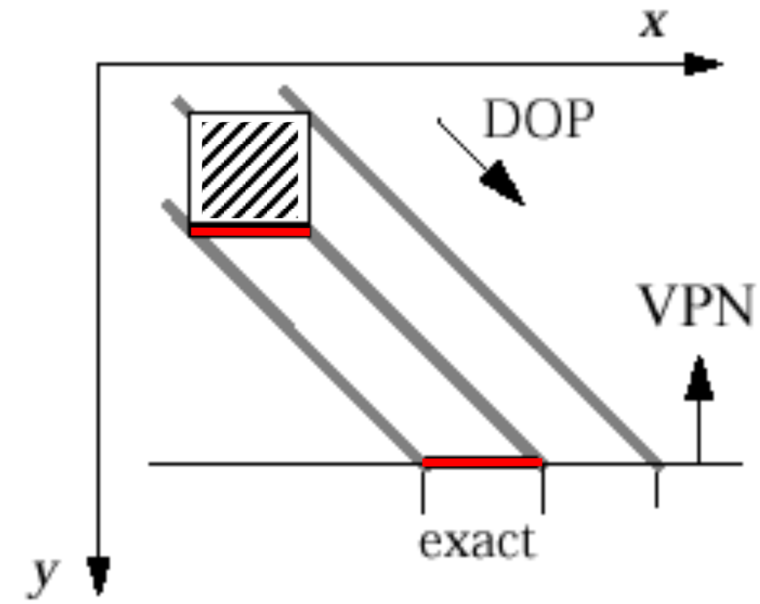

Parallel
Orthographic
Multi-view

VPN || axis
DOP || VPN
single face
exact measurements




Parallel
Orthographic
Axonometric

VPN not || axis
DOP || VPN
several faces (adjacent)
none exact
foreshortening uniform




Parallel
Oblique

VPN || axis
DOP not || VPN
several faces (adjacent)
one exact
others uniformly
foreshortened

# Specification of an Arbitrary 3D View

# Projection Plane

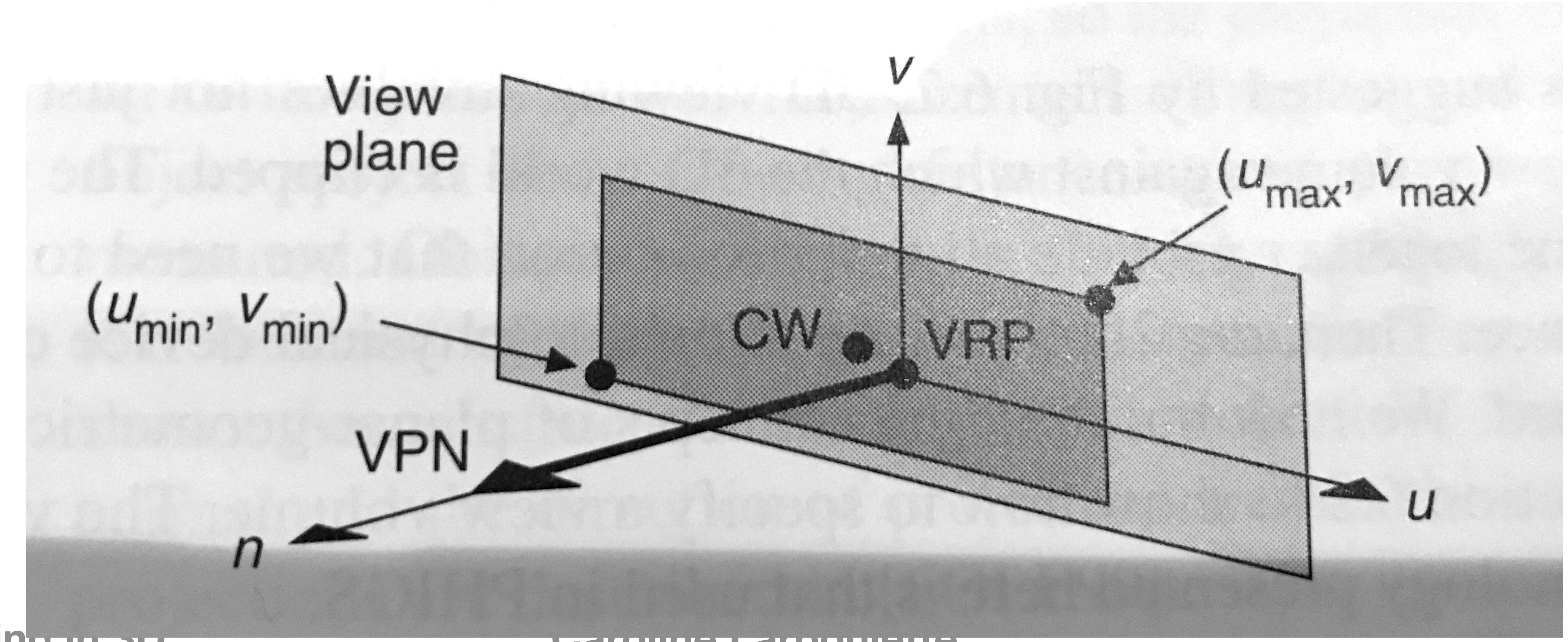
- Also called view plane
- Defined by
  - a point on the plane: VRP (View Reference Point)
  - a normal to the plane: VPN (View Plane Normal)
- Maybe anywhere in the scene
  - in front / behind the objects
  - cut through the objects

# Viewing window

- We need a window on the view plane
  - Contents are then mapped to the viewport
  - Objects that project outside the window are not displayed
- Defined with
  - Minimum and maximum coordinates
  - Two orthogonal axes

# Viewing-Reference Coordinate (VRC) System

- Origin is VRP (View Reference Point)
- n-axis, perpendicular, to plane is VPN
- v-axis, projection of Up-Vector (VUP) parallel to VPN
- u-axis is $\overrightarrow{v} \times \overrightarrow{n}$
- CW is center of the window

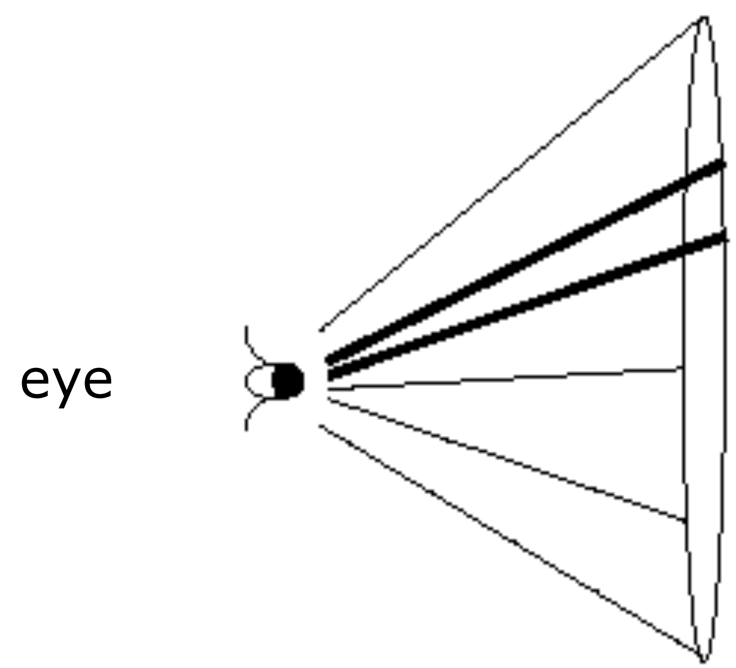


# Projection

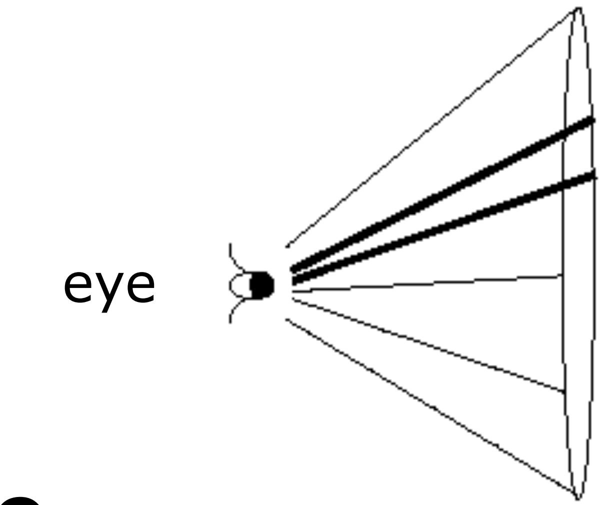
- Center of projection and direction of projection (DOP) are defined by a projection reference point PRP
  - PRP is the center of projection if perspective
  - DOP is the vector from PRP to CW if parallel
- PRP is specified in the VRC system (if camera moves, projection characteristic do not change)

# View Volume

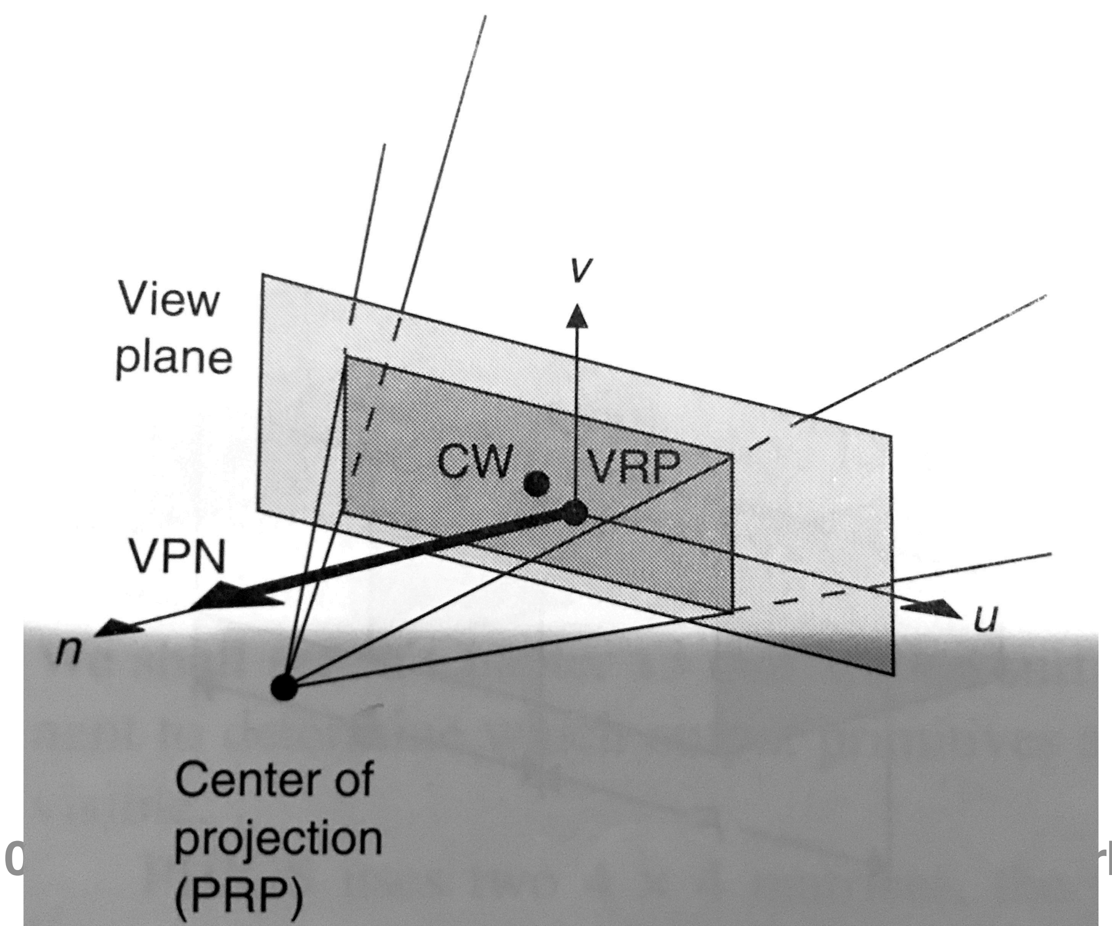
- Reality: eye sees a cone-like volume

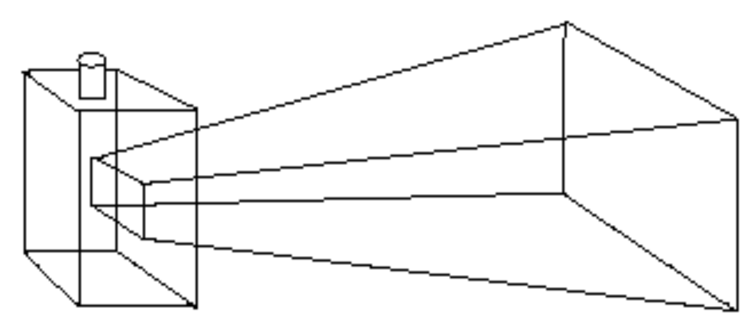


# View Volume

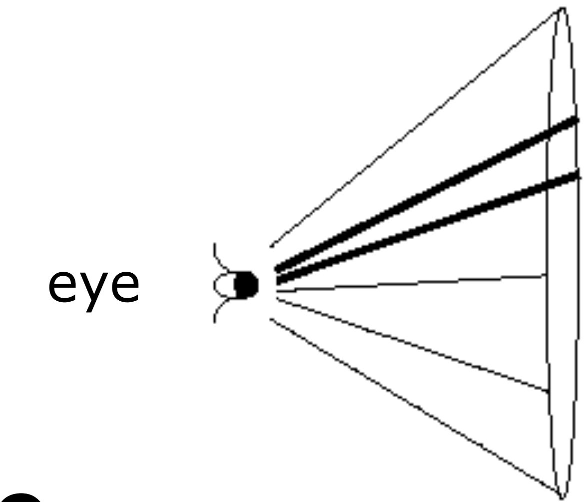



- Reality: eye sees a cone-like volume
- Perspective: semi-infinite pyramid with apex at PRP (center of projection) and the 4 edges passing through corners of the window: frustum

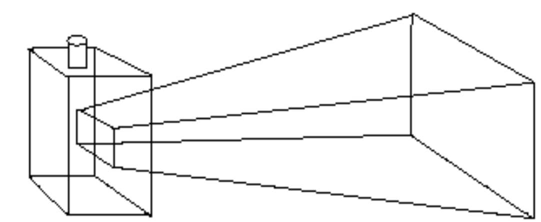


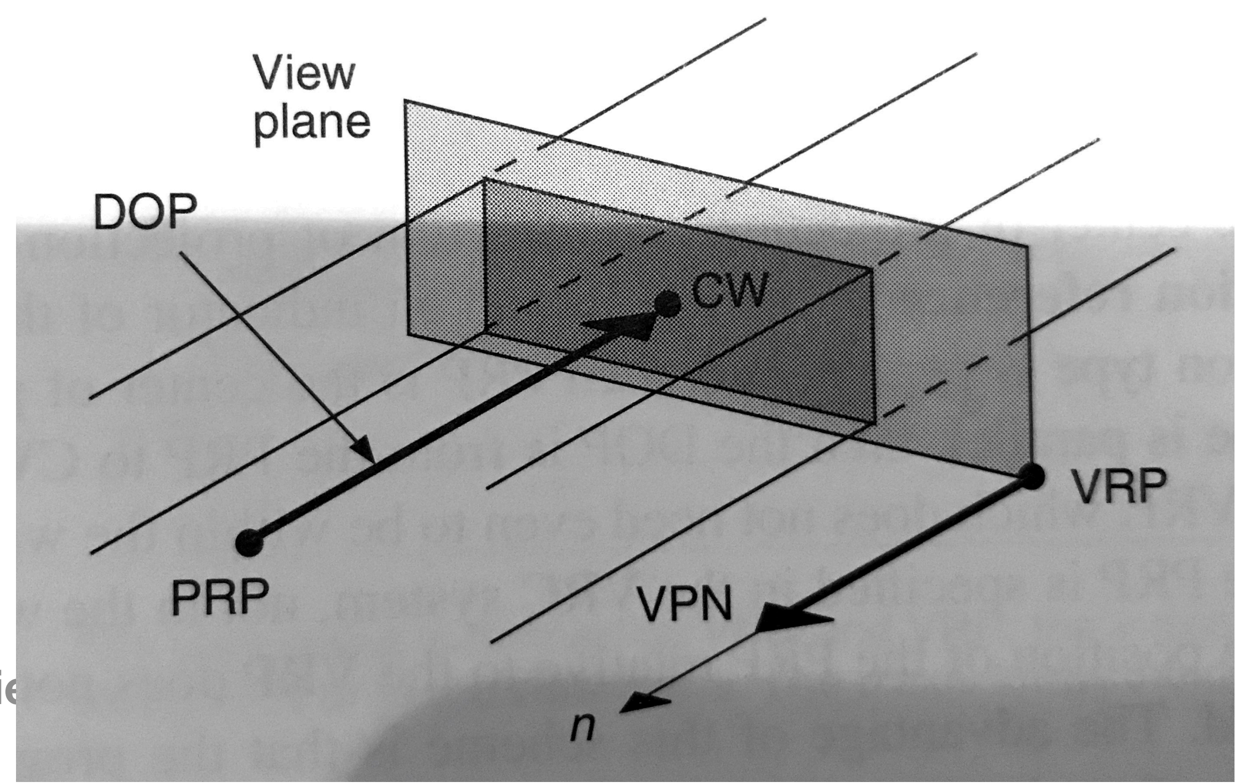
# View Volume

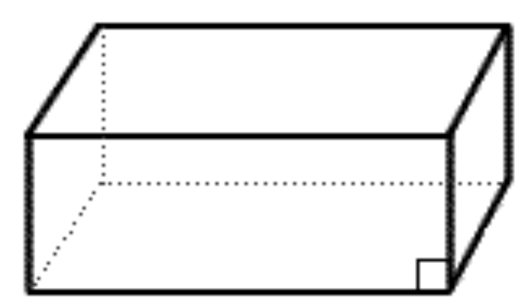



- Reality: eye sees a cone-like volume



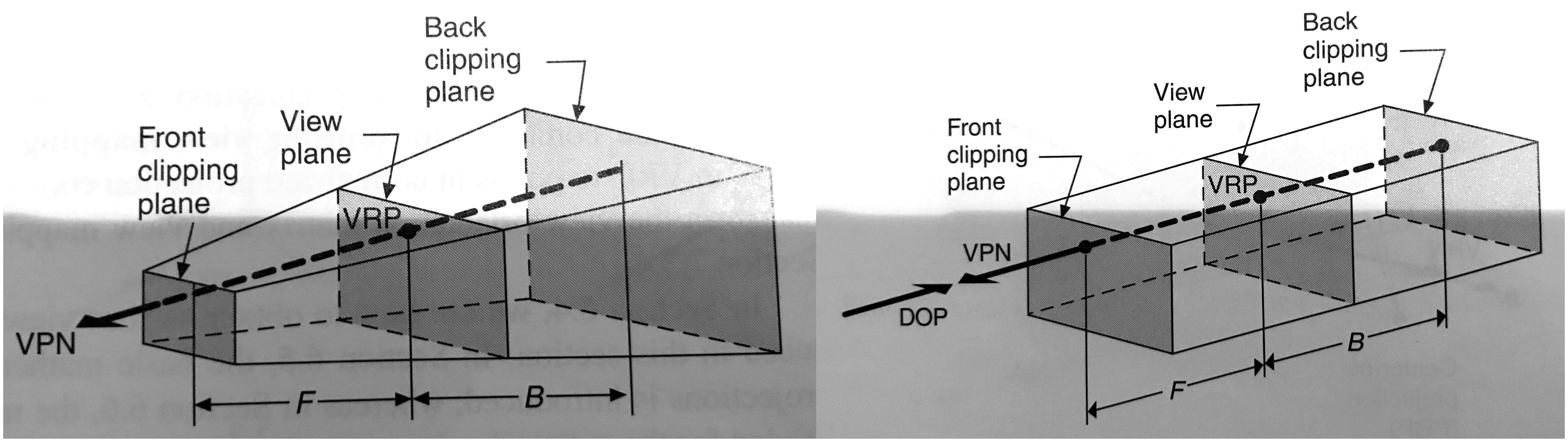
- Perspective: semi-infinite pyramid
- Parallel: infinite parallelepiped with sides parallel to the direction of projection (from PRP to center of window CW)



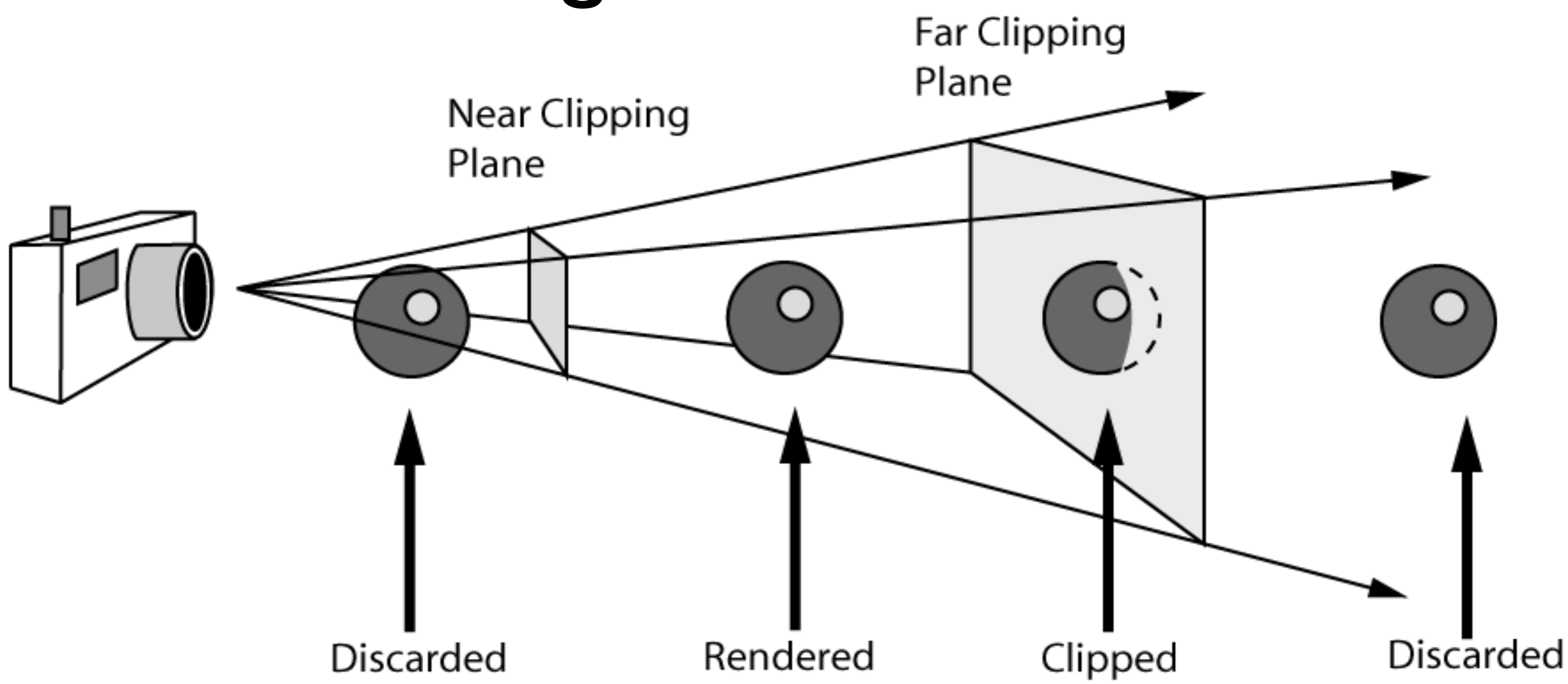
# Finite View Volume

- Use of a front clipping plane and back clipping plane
- Parallel to the view plane (normal is VPN)
- Defined by front and back distance relative to VRP in VPN direction
- Front distance must be shorter than back distance (else volume empty)



# Near and Far Clipping Planes

- The view volume or frustum defines what we can see in the scene
- Objects outside are discarded
- Objects inside are rendered
- Objects intersecting faces of the volume are clipped



# Near and Far Clipping Planes

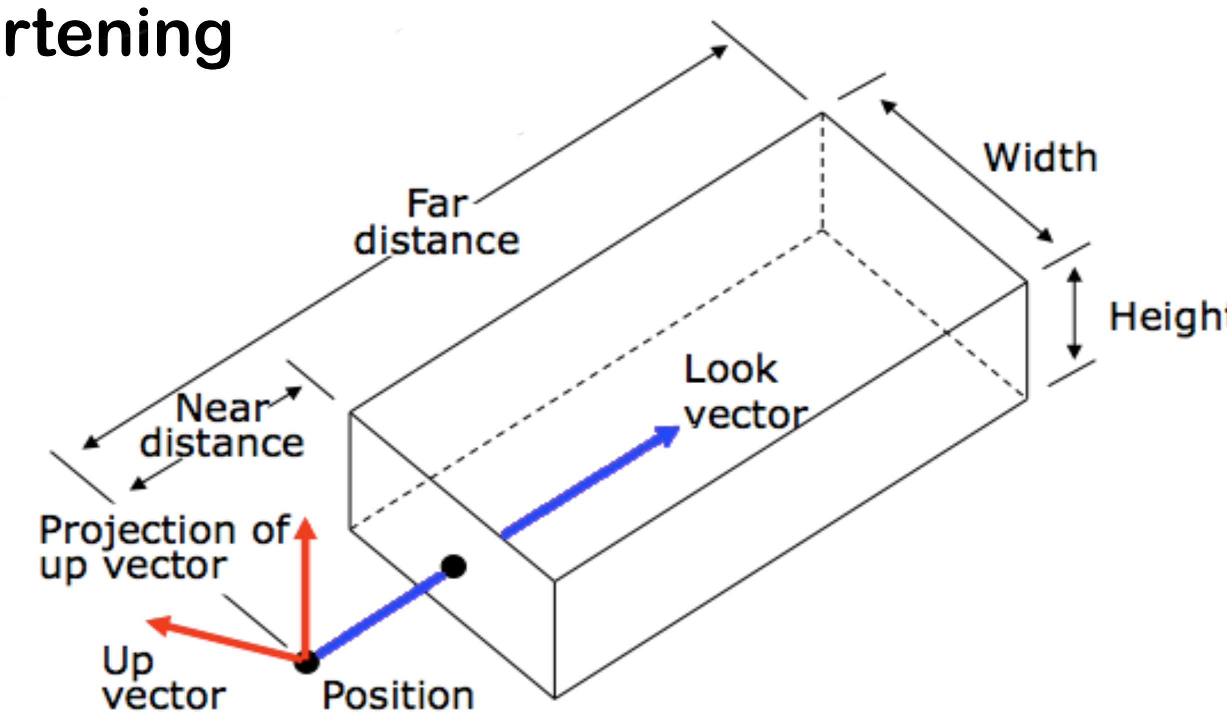
- Reasons for front (near) clipping plane :
  - Usually don't want to draw things too close to camera
    - Would block view of rest of scene
    - Objects would be quite distorted (perspective)
  - Don't want to draw things behind camera
    - Wouldn't expect to see things behind camera
    - In the case of perspective camera, if we were to draw things behind camera, they would appear upside-down and inside-out because of perspective transformation

# Near and Far Clipping Planes

- Reasons for back (far) clipping plane :
  - Don’t want to draw objects too far away from camera
    - Distant objects may appear too small to be visually significant, but still take a long time to render; different parts of an object may map onto same pixel (sampling error)
    - By discarding them we lose a small amount of detail but reclaim a lot of rendering time
    - Helps to declutter a scene
- These planes need to be properly placed, not too close to the camera, not too far

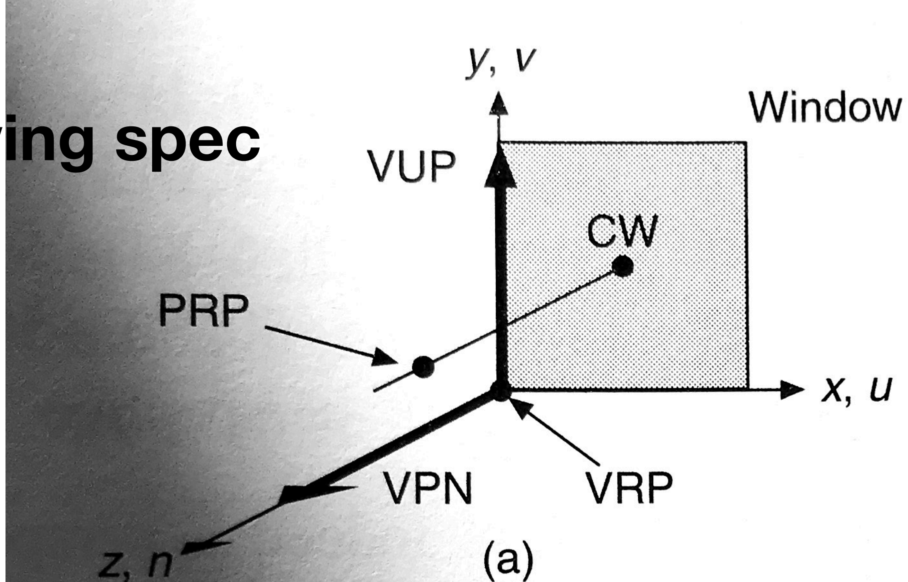
# Parallel View Volume

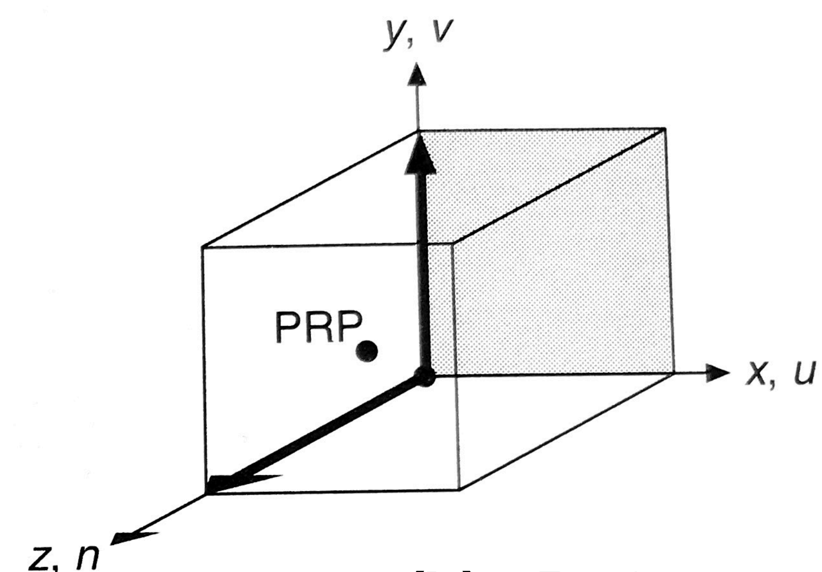
- Objects appear same size no matter how far away since projectors are all parallel: uniform foreshortening based on angle of film plane to projectors, not depth-dependent perspective (non-uniform) foreshortening
- Benefits of parallel view volume
  - Easier clipping because of simpler plane equations
  - Easier depth comparison for Visible Surface Determination (solving the obscuration problem)
  - Really easy to project a 3D scene to 2D projection/film plane because no perspective foreshortening



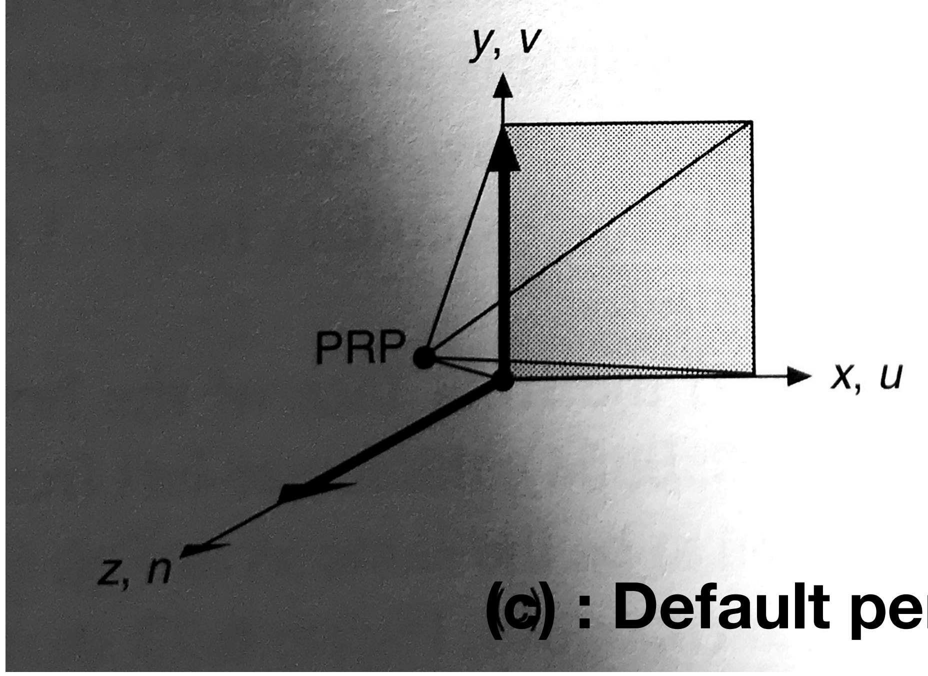
# Viewing Reference (VRC) and World (WC) coordinates

Default viewing spec




**(b) : Default parallel projection view volume**

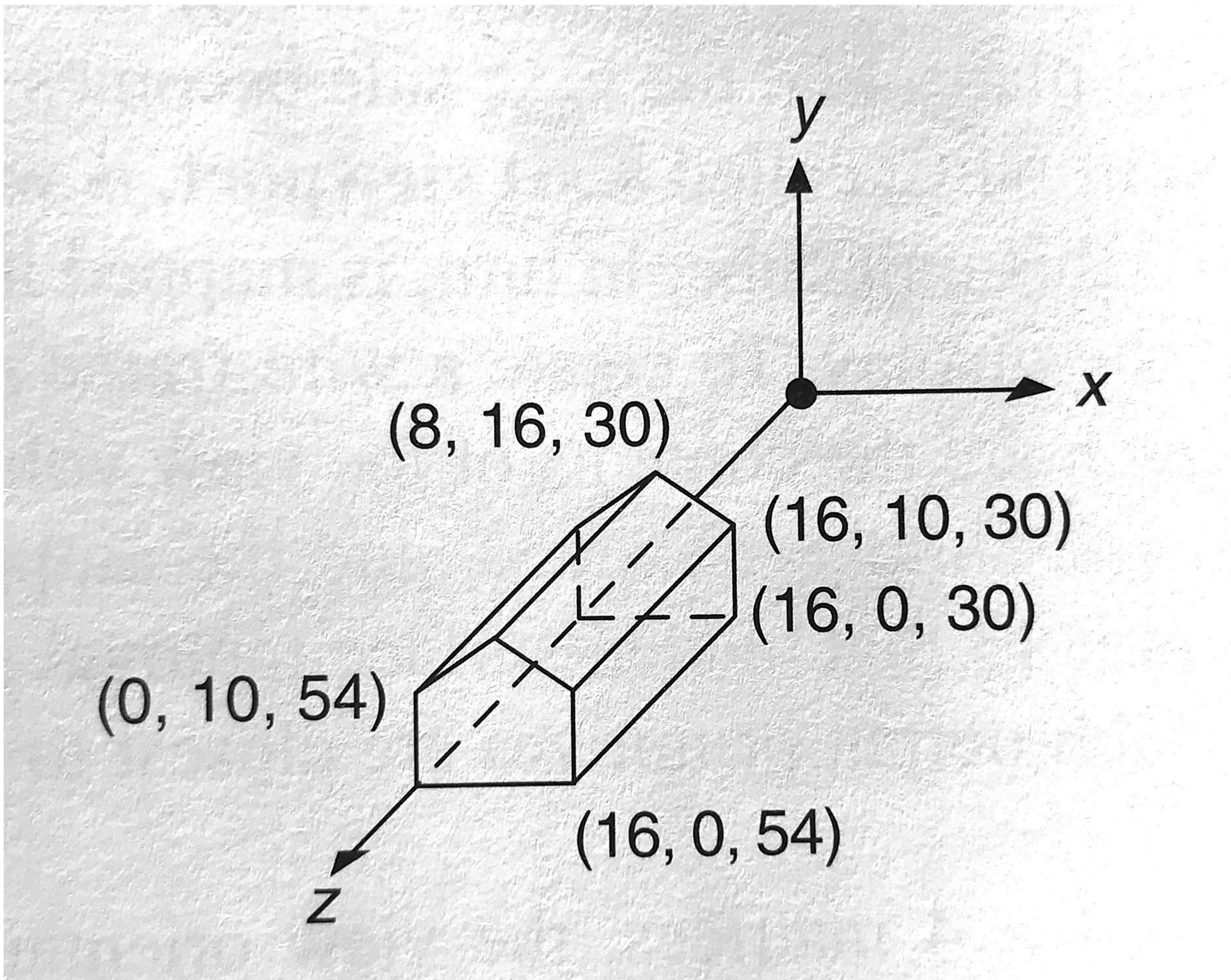



**VRP (WC) = (0,0,0) = origin**
**VPN (WC) = (0,0,1) = z-axis**
**VUP (WC) = (0,1,0) = y-axis**
**PRP (VRC) = (0.5, 0.5, 1.0)**
**Window (VRC) = (0,1,0,1)**

**(c) : Default perspective projection view volume**

# Examples of 3D Viewing

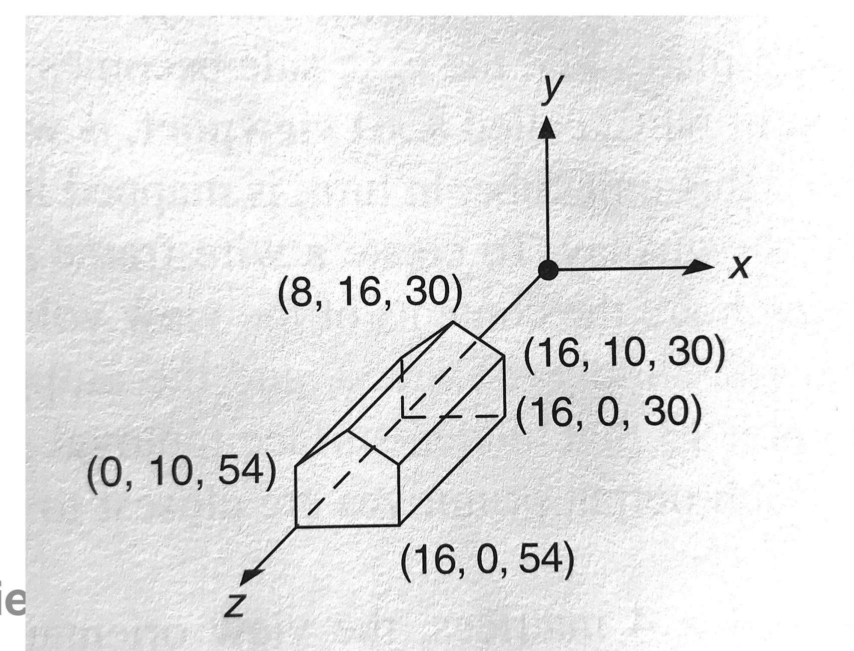
# 3D Model of House

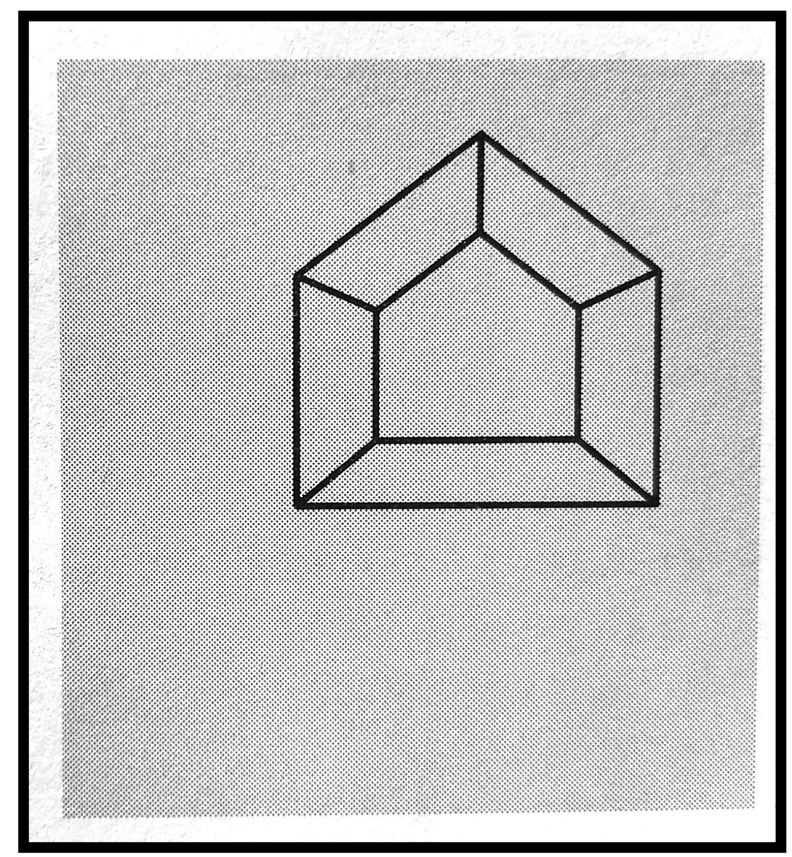


# Example 1

## One-point perspective projection

- Center of projection = viewer position = PRP at x=8, y=6, z=84
- x is at horizontal center of the house
- y is at approximate eye level standing on the (z,x) plane
- z is 30 units before the house

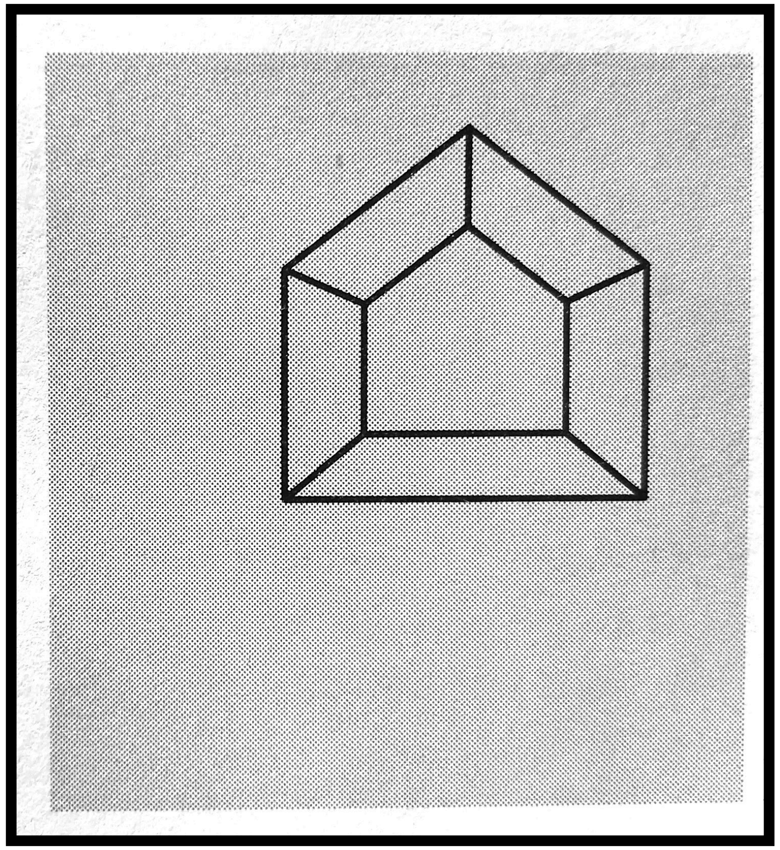


# Example 1

## One-point perspective projection

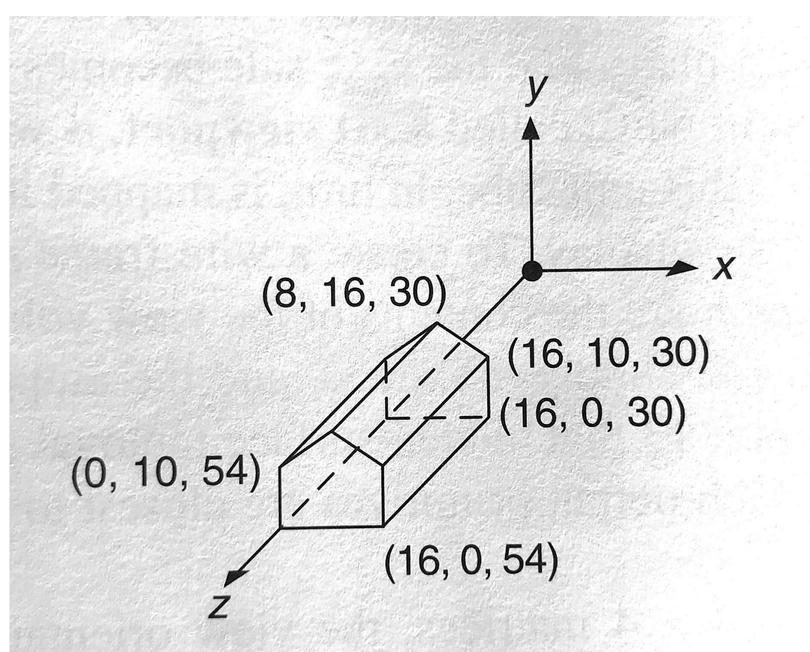
- VRP (WC) = (0,0,0)
- VPN (WC) = (0,0,1)
- VUP (WC) = (0,1,0)
- PRP (VRC) = PRP (WC) = (8,6,84)
- Window (VRC) = (-50, 50, -50, 50)

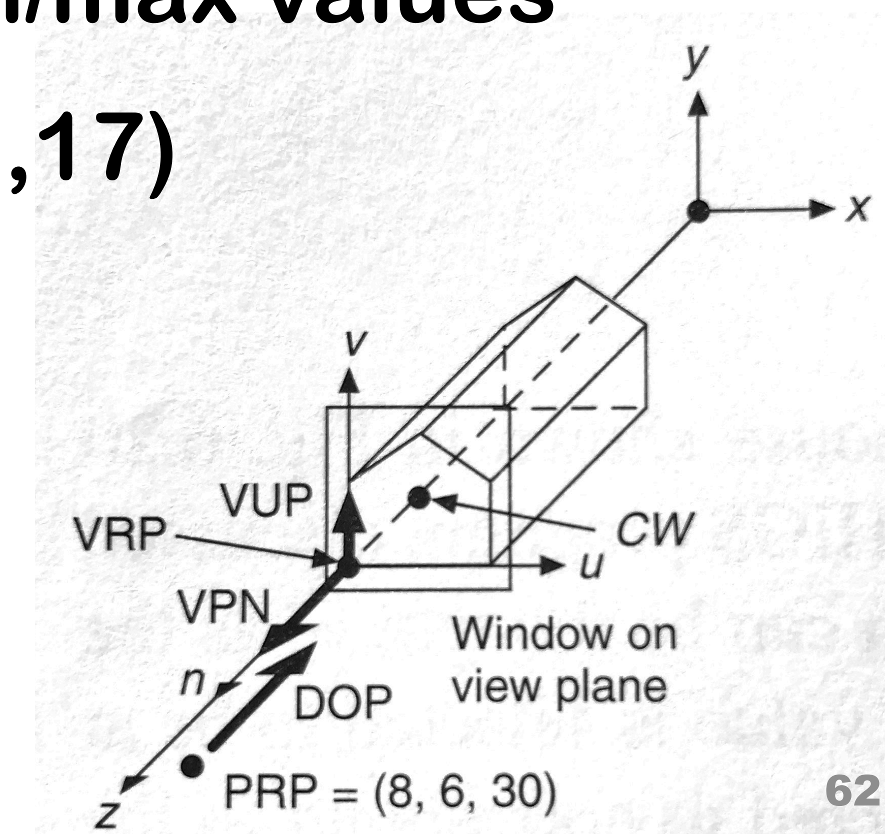


# Example 1

## One-point perspective projection

- To center the house, we can make view plane coincide with the front plane of the house
- VRP (WC) becomes (0,0,54)
- To make the house occupy the entire window, we can change u_min/max and v_min/max values
- Window (VRC) becomes (-1,17,-1,17)

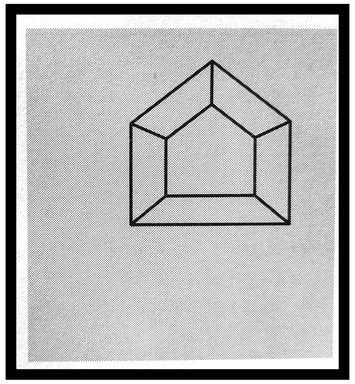


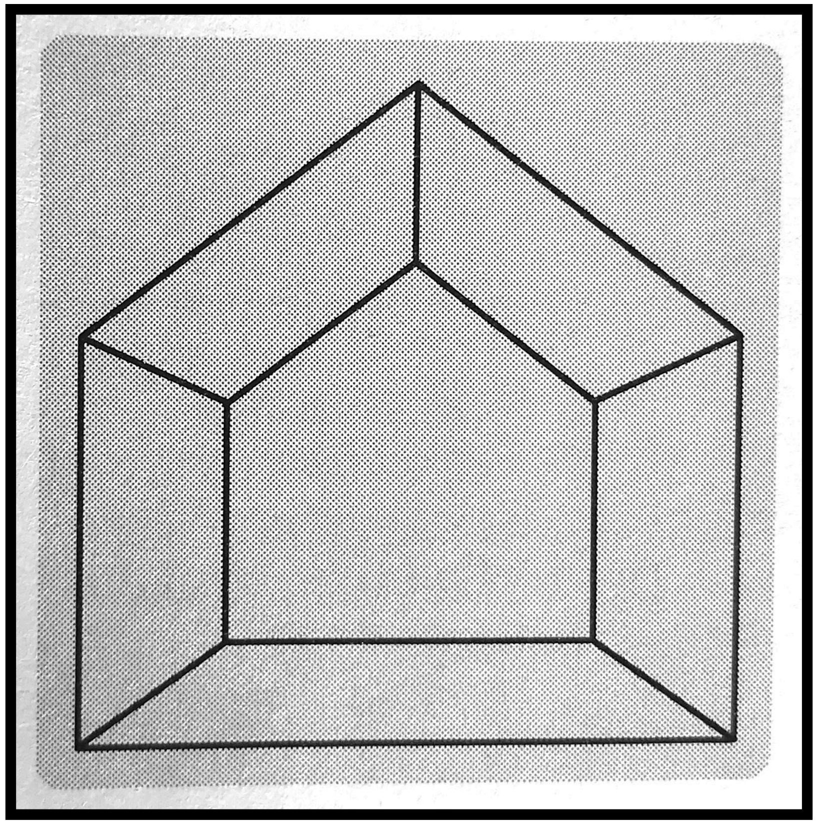



# Example 1

## One-point perspective projection

- VRP (WC) = (0,0,0)
- VPN (WC) = (0,0,1)
- VUP (WC) = (0,1,0)
- PRP (VRC) = PRP (WC) = (8,6,84)
- Window (VRC) = (-50, 50, -50, 50)



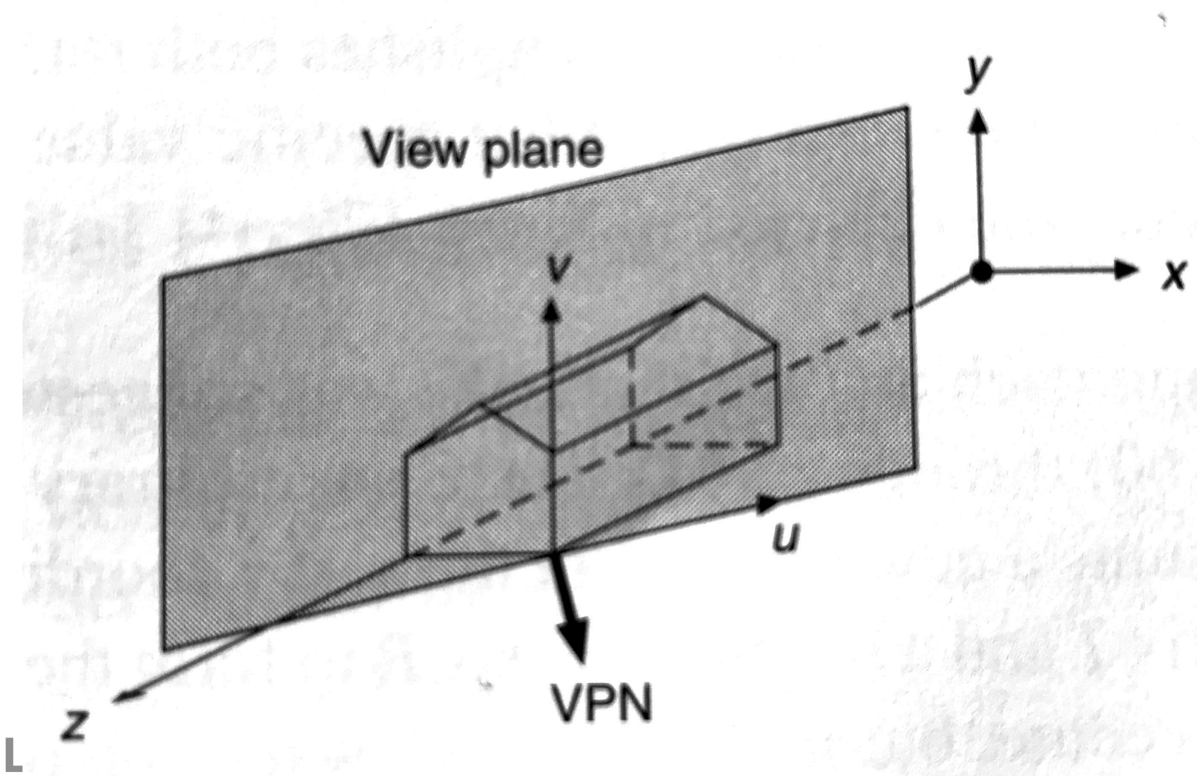


- VRP (WC) = (0,0,54)
- VPN (WC) = (0,0,1)
- VUP (WC) = (0,1,0)
- PRP (VRC) = PRP (WC) - VRP (WC) = (8,6,30)
- Window (VRC) = (-1, 17, -1, 17)

# Example 2

## Two-point perspective projection

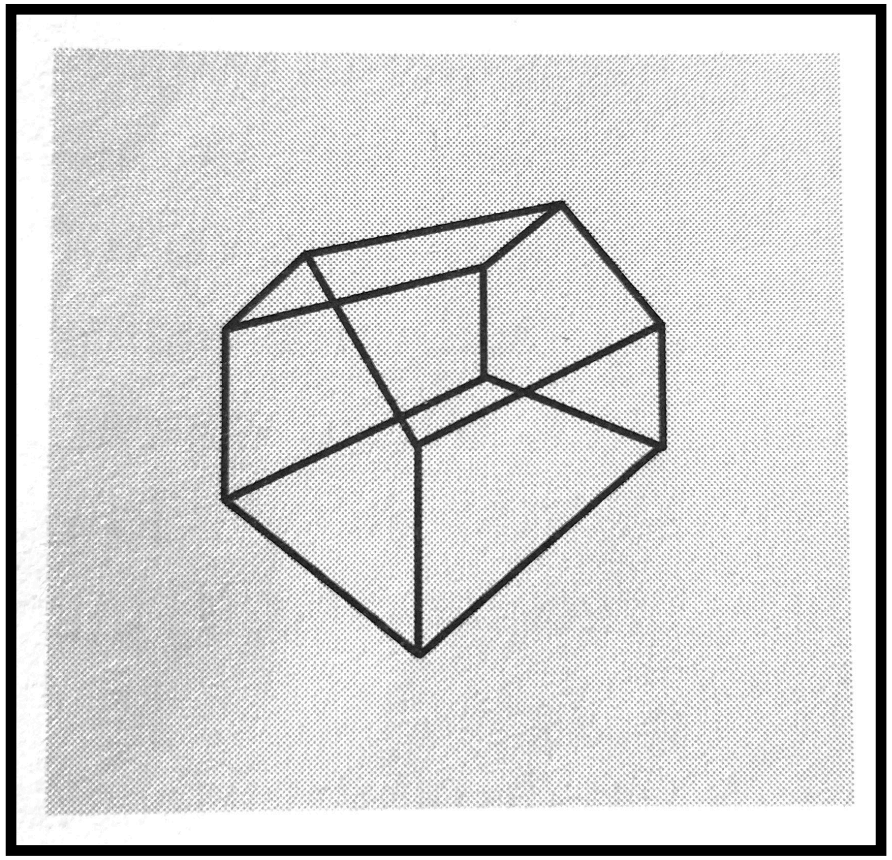
- To obtain a two-point perspective projection, the projection plane need to be reoriented such that it cuts two axes, x-axis and z-axis for example
- VPN (WC) becomes (1,0,1)



# Example 2

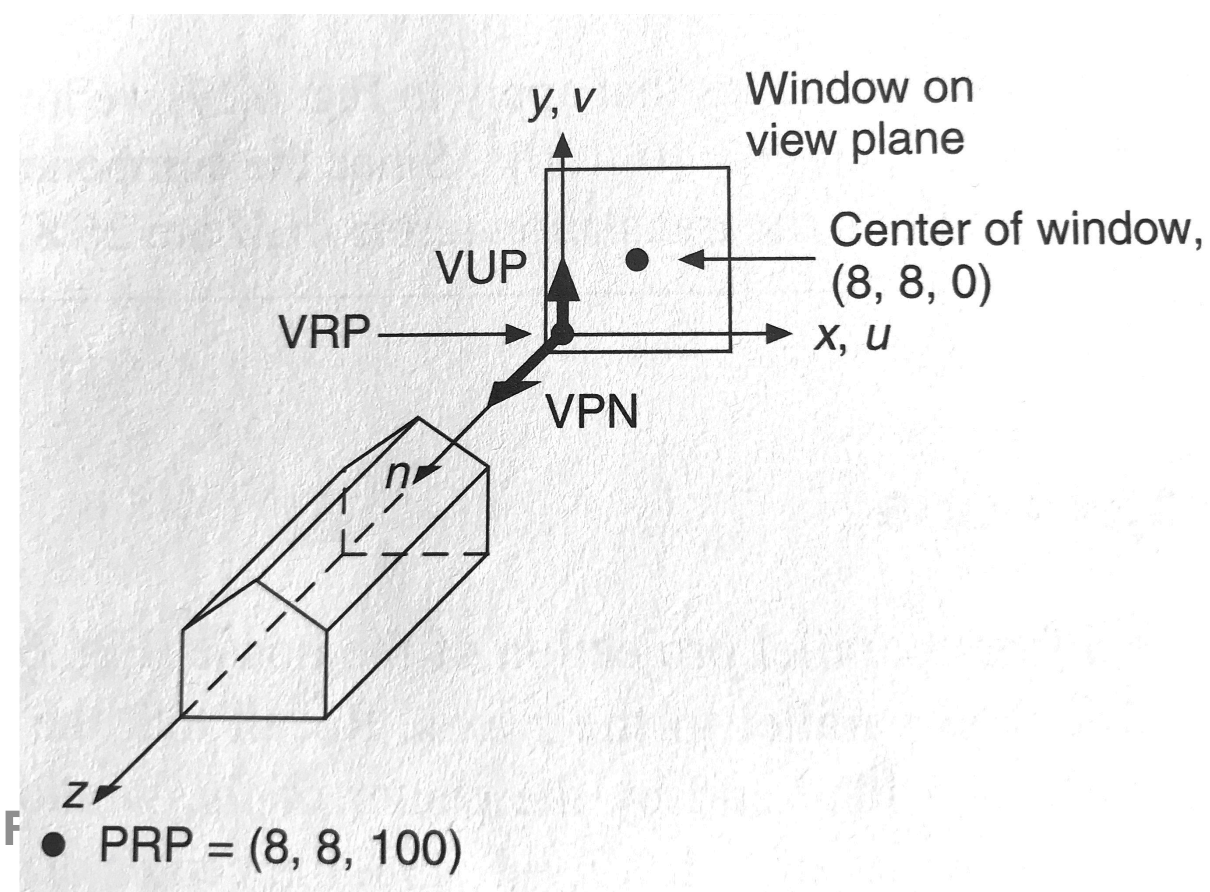
## Two-point perspective projection

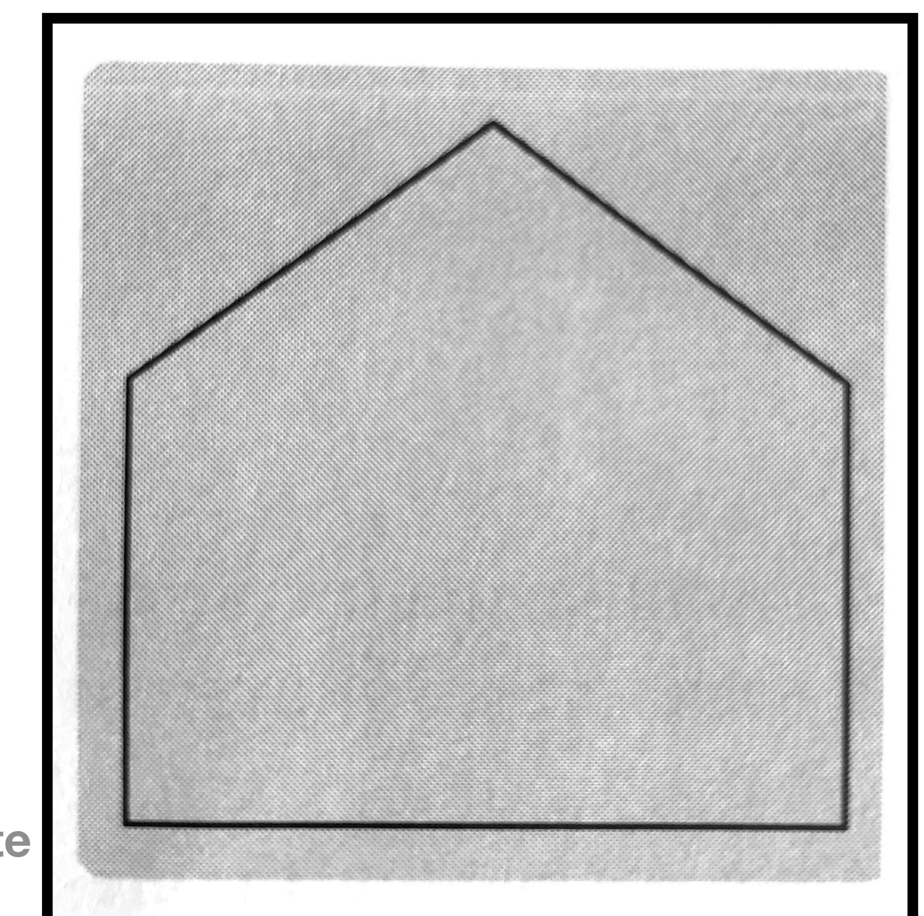
- VRP (WC) = (16,0,54)
- VPN (WC) = (1,0,1)
- VUP (WC) = (0,1,0)
- PRP (VRC) = (0,25,28.3)
- Window (VRC) = (-20, 20, -5, 35)



# Example 3
## Parallel Projection

- Front parallel projection: DOP // z-axis
- If window is (-1,17,-1,17), center is at (8,8,0)
- PRP at (8,8,100) gives a projection along z-axis

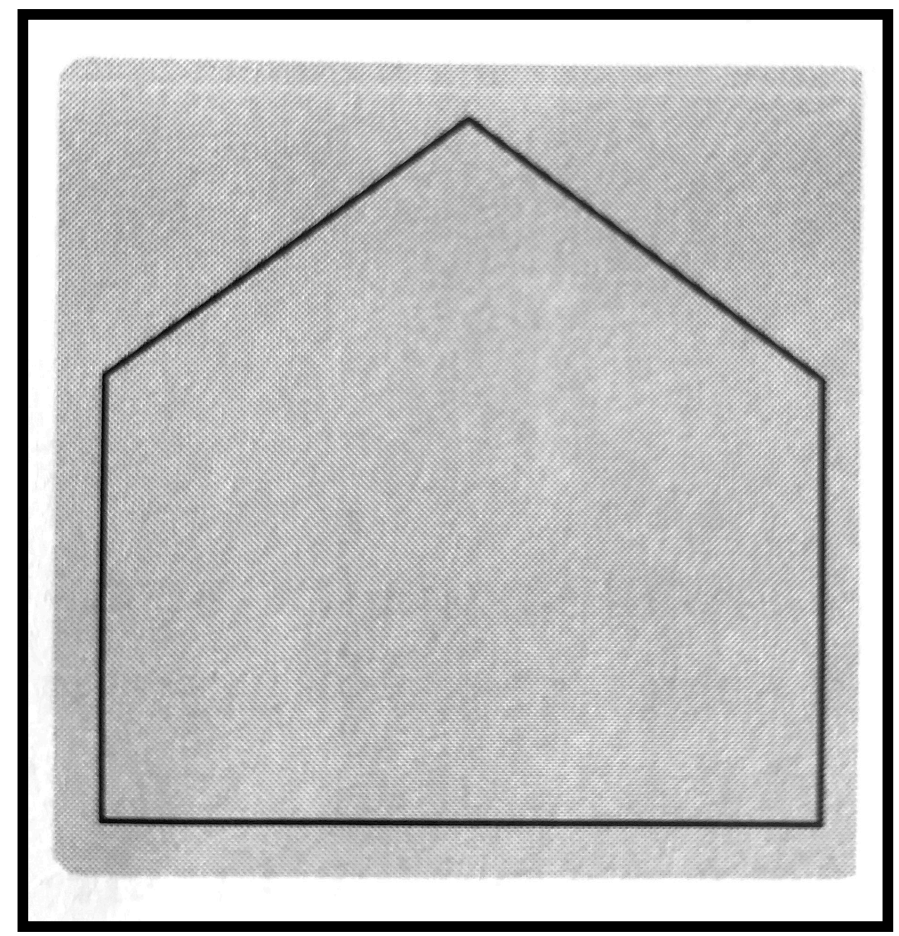


# Example 3

## Parallel Projection

- VRP (WC) = (0,0,0)
- VPN (WC) = (0,0,1)
- VUP (WC) = (0,1,0)
- PRP (VRC) = (8,8,100)
- Window (VRC) = (-1, 17, -1, 17)



# Planar Geometric Projections

# Projection Matrix

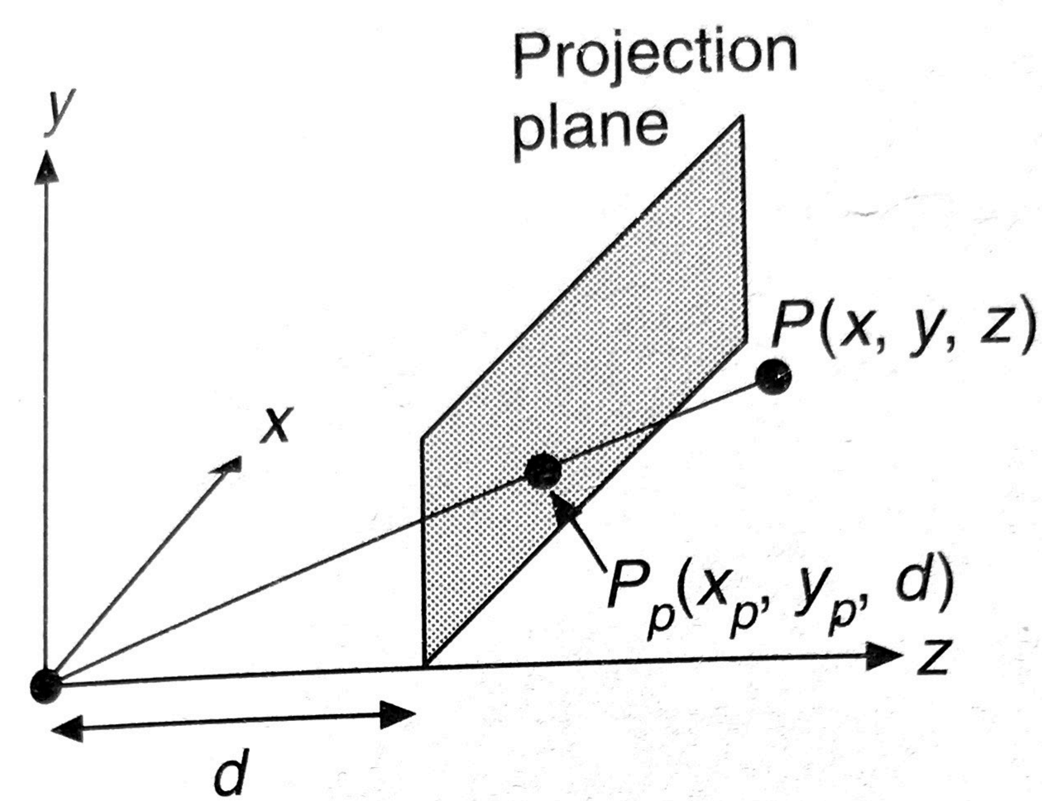
- Projection Matrix is 4x4
- Can be composed with Modelview Matrix
- First transform, then project

# Perspective Projection

## One-point perspective

- Let's assume the projection plane is normal to z-axis at z=d
- Let P(x,y,z) be the point to be projected
- What is the projection matrix to obtain

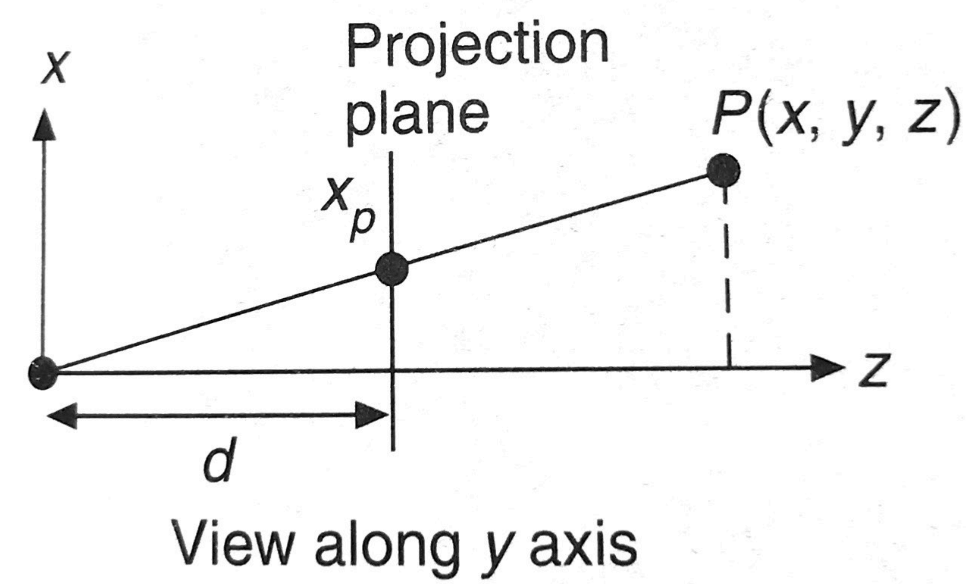
$$P_p(x_p, y_p, z_p) = P_p(x_p, y_p, d) \quad ?$$



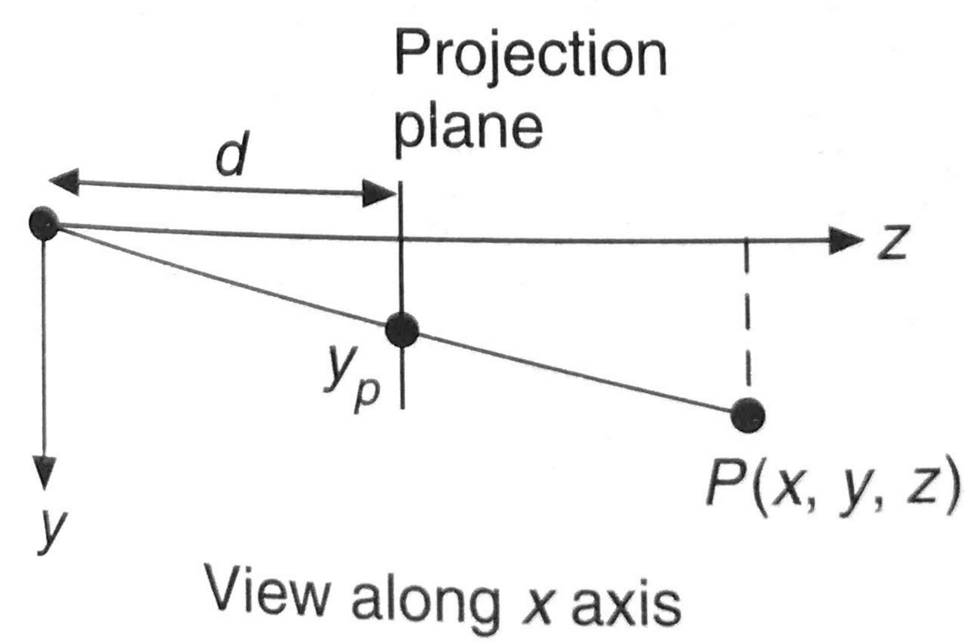
# Perspective Projection

## One-point perspective

$$\frac{x_p}{d} = \frac{x}{z} \qquad x_p = \frac{d\,.\,x}{z} = \frac{x}{z\,/d}$$




$$\frac{y_p}{d} = \frac{y}{z} \qquad y_p = \frac{d\,.\,y}{z} = \frac{y}{z\,/d}$$



# Perspective Projection

## One-point perspective

$$x_p = \frac{d \,.\, x}{z} = \frac{x}{z/d} \qquad\qquad y_p = \frac{d \,.\, y}{z} = \frac{y}{z/d}$$

$$M_{per} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix}$$

# Perspective Projection

## Two-point perspective

- Add a rotation about y-axis
- Projection plane cuts both x and z-axis
- Mper is multiplied by rotation matrix

$$M_{per2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix} \cdot \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ -\sin(\theta)/d & 0 & \cos(\theta)/d & 0 \end{bmatrix}$$

# Parallel Projection

- Let's assume the projection plane is normal to z-axis at z=0

$$x_p = x, y_p = y, z_p = 0$$

$$M_{ort} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

# Canonical Volumes

- Before projecting the points, they need to be clipped against the 3D volume
- To simplify clipping, points are first transformed from view volume to canonical view volume
- It's the normalising transformation

**Perspective**

x=z

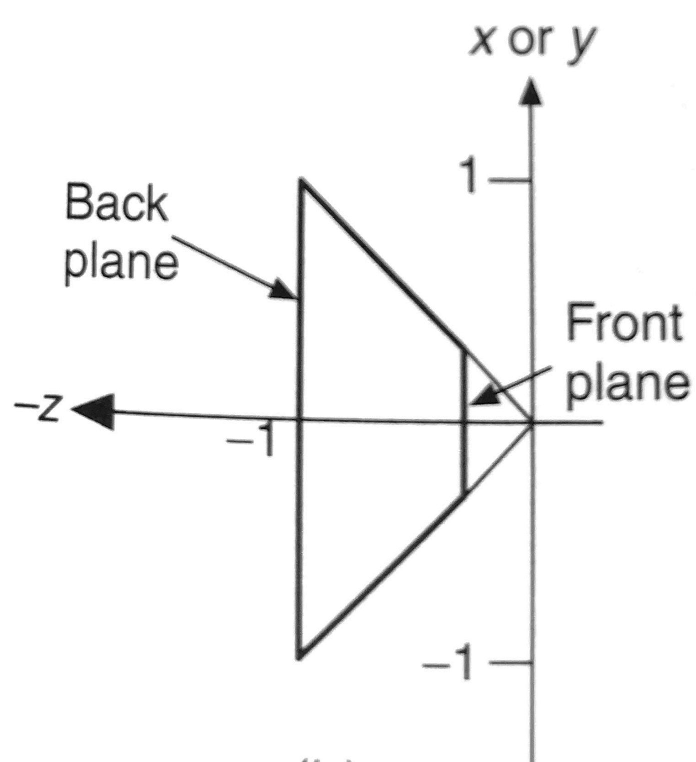
x=-z

y=z

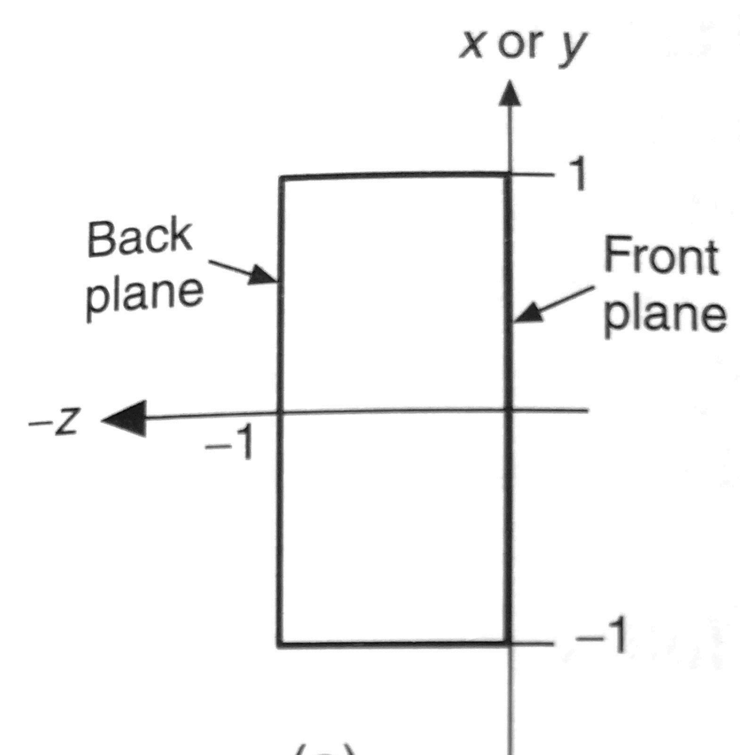
y=-z

z=-zmin

z=-1




(b)




(a)

**Parallel**

x=-1

x=1

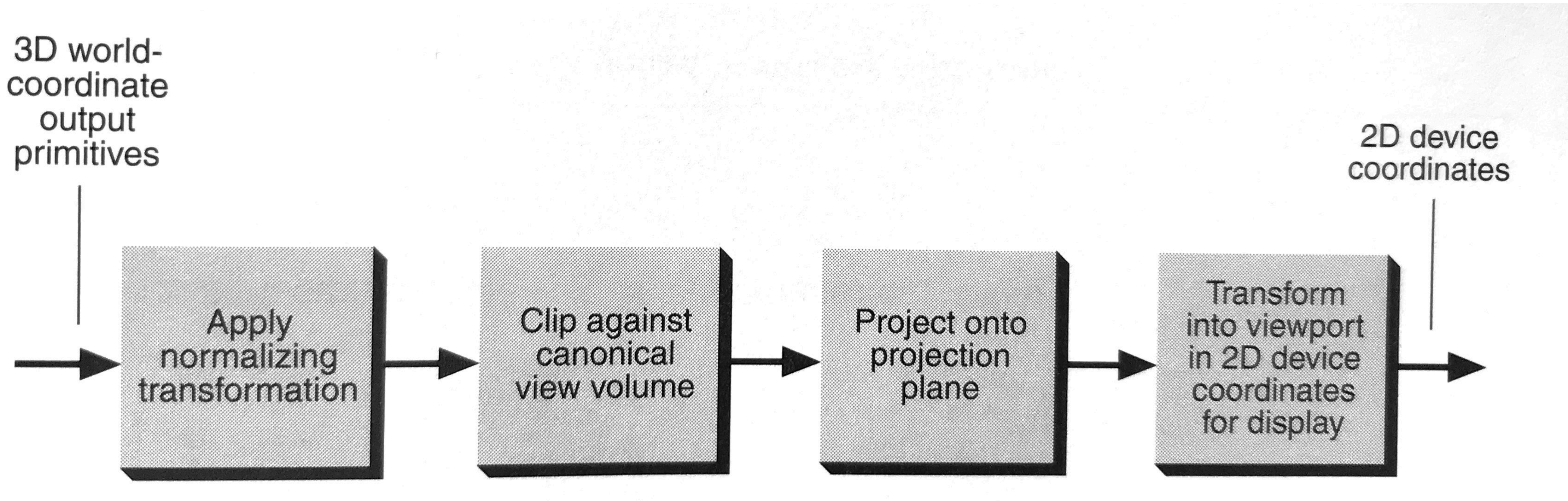
y=-1

y=1

z=0

z=-1

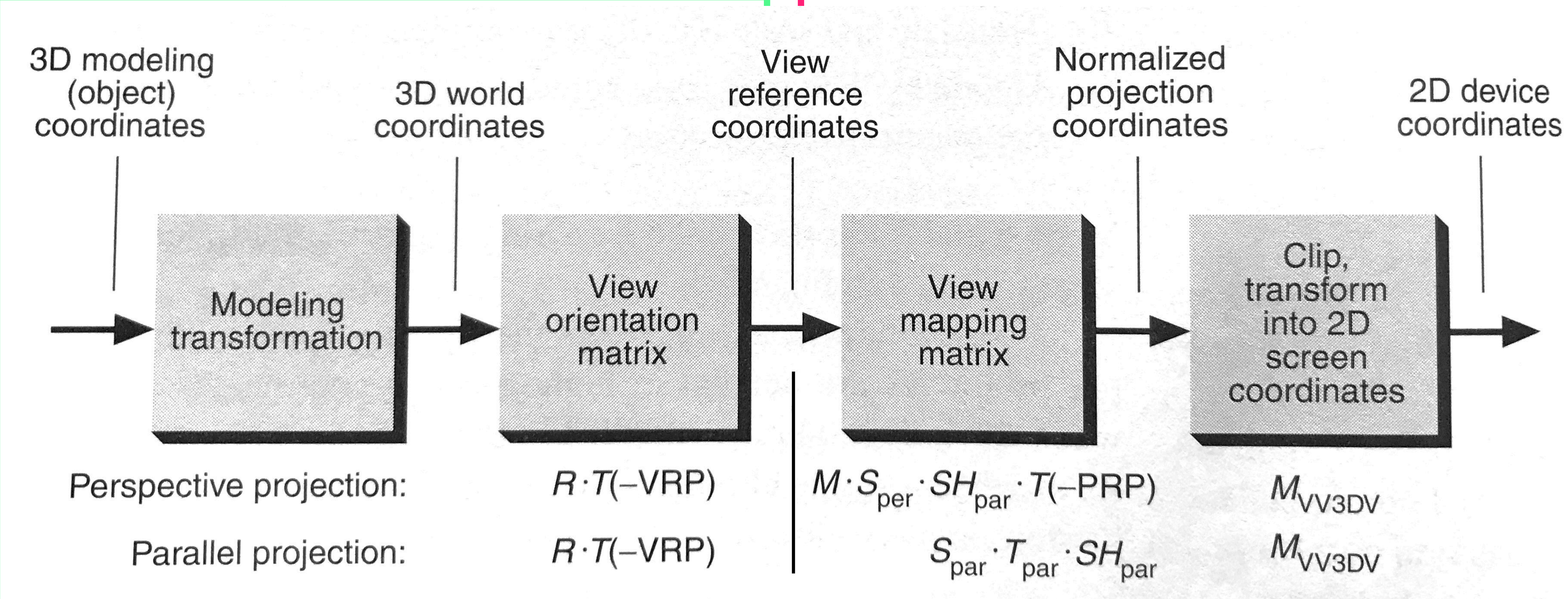
# Projection Pipeline



# Coordinate Systems

# 3D Viewing

## Summary




**Local Coordinates** | **World Coordinates** | **Eye Coordinates** | **3D Screen Coordinates** | **2D Device Coordinates**

**MODELVIEW Matrix**

**PROJECTION Matrix**