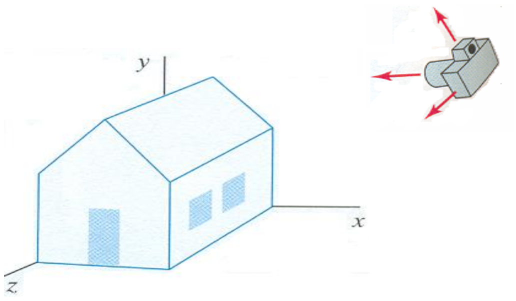
Universidade de Aveiro
Departamento de Electrónica,
Telecomunicações e Informática

3D Visualization



Beatriz Sousa Santos, J. Madeira

Overview

- 3D Viewing
- Planar Projections
- Matricial Representation
- Application Example
- Projections in OpenGL / WebGL

3D VIEWING

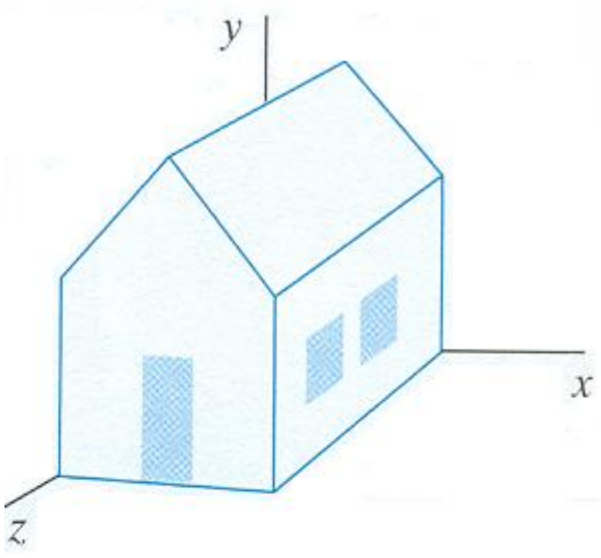
3D Visualization

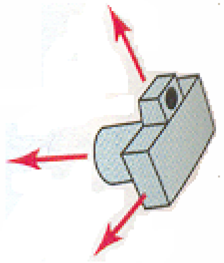
- The process of obtaining a 2D image representing a 3D scene is analogous to photographing

- Some visualization / viewing parameters have to be set:

  - position
    (analogous to camera position, depending on the required view)

  - orientation
    (analogous to camera orientation)





In CG there are more degrees-of-freedom than in traditional photography (e.g., choosing the projection type, the location of the projection plane, etc.)

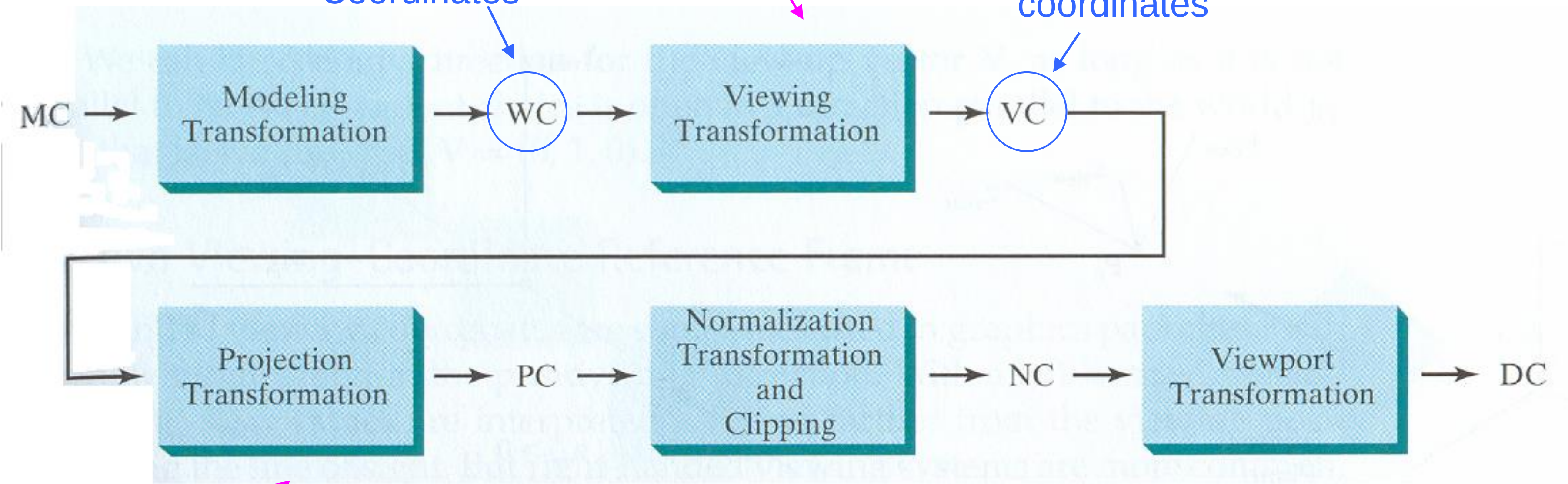
Viewing Pipeline – Coordinate transformations

From scene coordinates to device coordinates:

1st – define the viewing transformation

world Coordinates

viewing coordinates



2nd – define the projection transformation

Need also to identify the visible surfaces and carry out the *surface rendering*

- Some of the stages of the 3D viewing pipeline are similar to those of the 2D viewing pipeline:

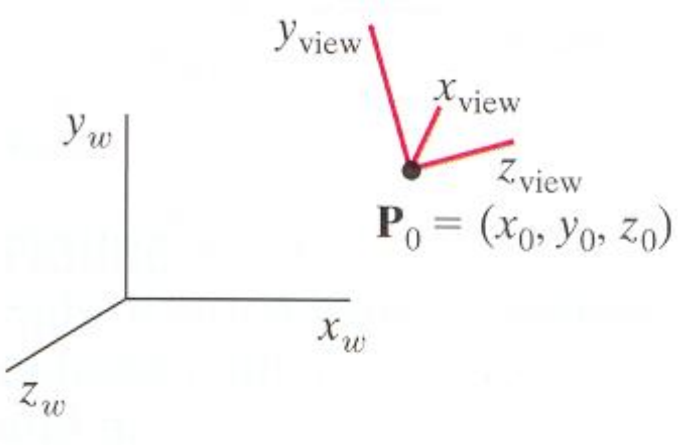
  - A 2D viewport – on the output device – is used to show a projection of the 3D scene

  - The clipping window is defined on the viewing plane

  - BUT the 3D clipping is carried out regarding a volume defined by a set of clipping planes

- The viewpoint, the viewing plane, the clipping window and the clipping planes are defined on the

  viewing coordinates system



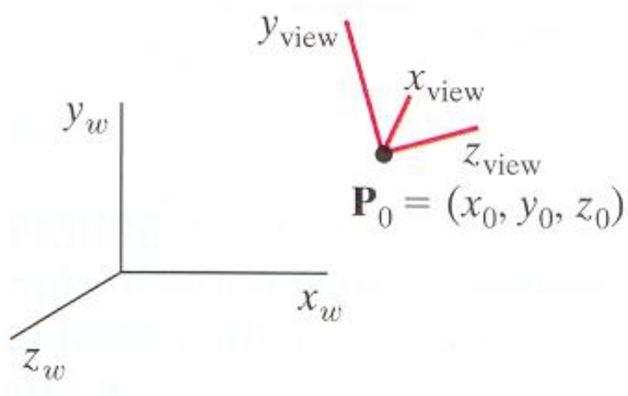
3D Viewing Coordinates

- Setting a 3D viewing coordinates system is analogous to setting a 2D viewing coordinates system:

  1- Choose a point ***Po (xo, yo, zo)*** as origin:
  viewing position or viewpoint

  2- Choose a view-up vector which defines the y_{view} direction

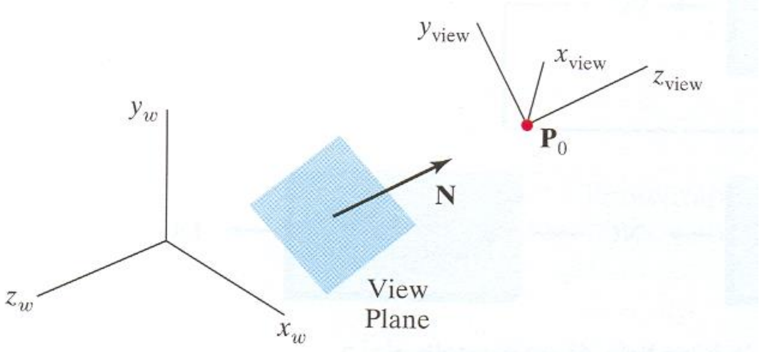
  3- Choose a direction for one of the other axis: z_{view}

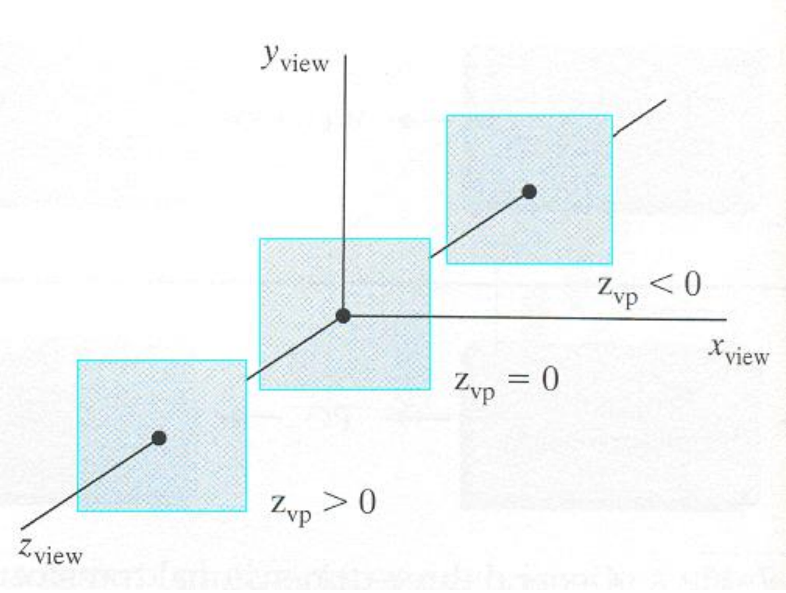


- In general, the viewing plane (i.e., projection plane) is defined as orthogonal to z_{view}

- The orientation of the viewing plane (and the positive direction for z_{view}) is defined by a normal vector $\mathbf{N}$

- An additional parameter defines the position of the viewing plane z_{vp} on the z_{view} axis

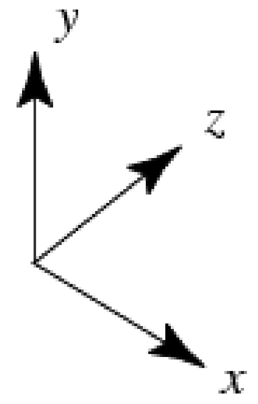




Possible positions for the viewing plane

- In general, the viewing coordinates system is defined as "right-handed"

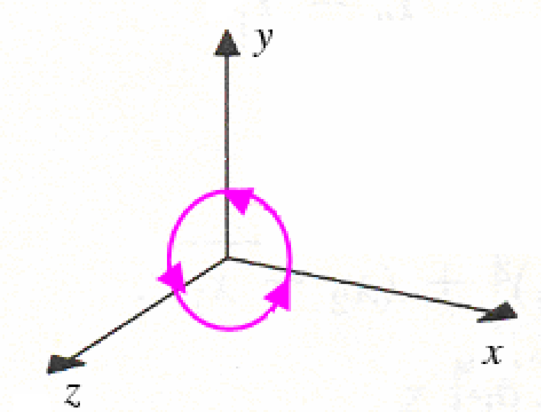
- But some graphics APIs use a "left-handed" coordinates system



"Left-handed" system (z is larger behind the plane)

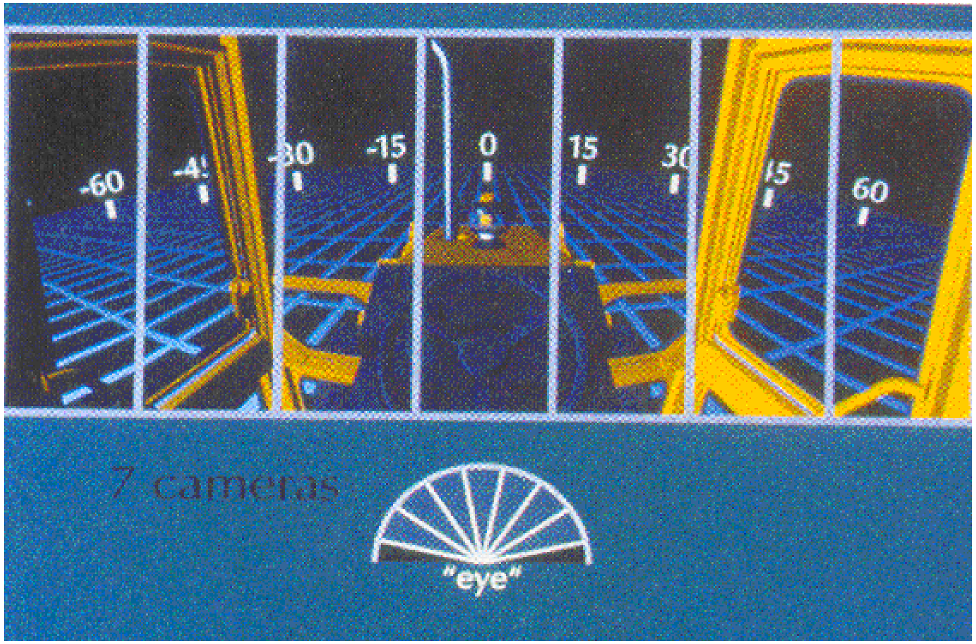
- In a "right-handed" coordinates system, and when looking at the origin from a point on one of the positive semi-axis, CCW 90º positive rotation angles transform a positive semi-axis into another positive semi-axis

rotation axis	direction of positive rotation
x	y to z
y	z to x
z	x to y

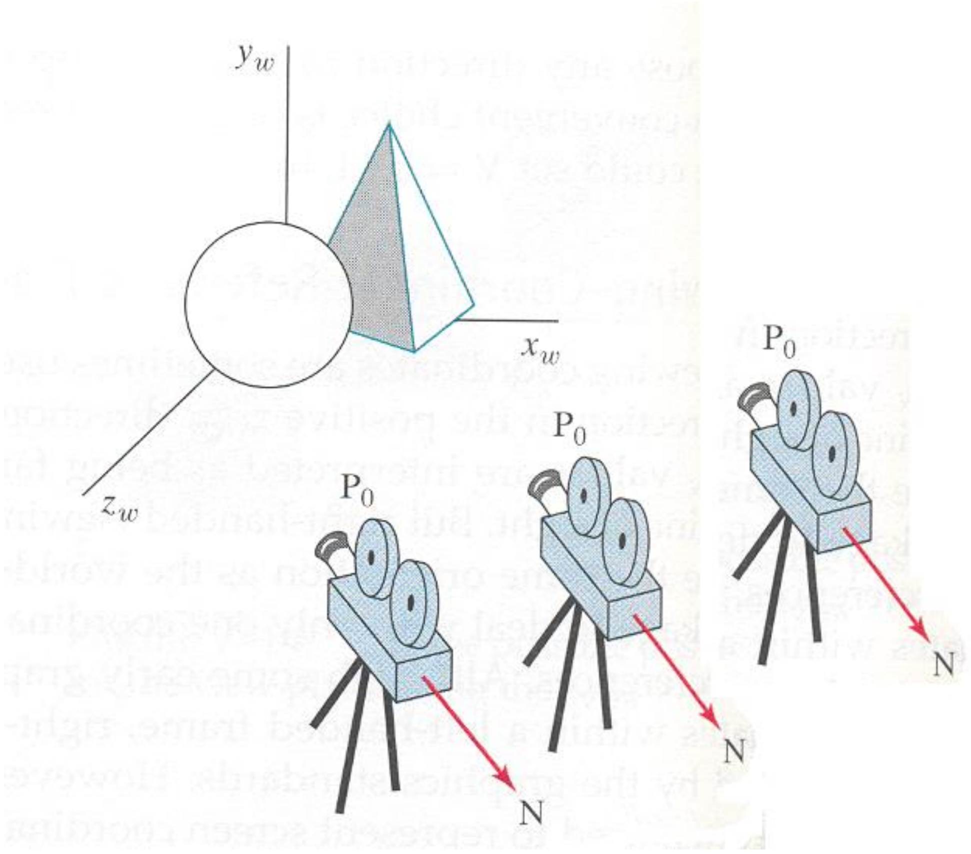


3D Viewing Effects

- Changing some viewing parameters, we can obtain different viewing effects (e.g., different side views, panning, etc.)

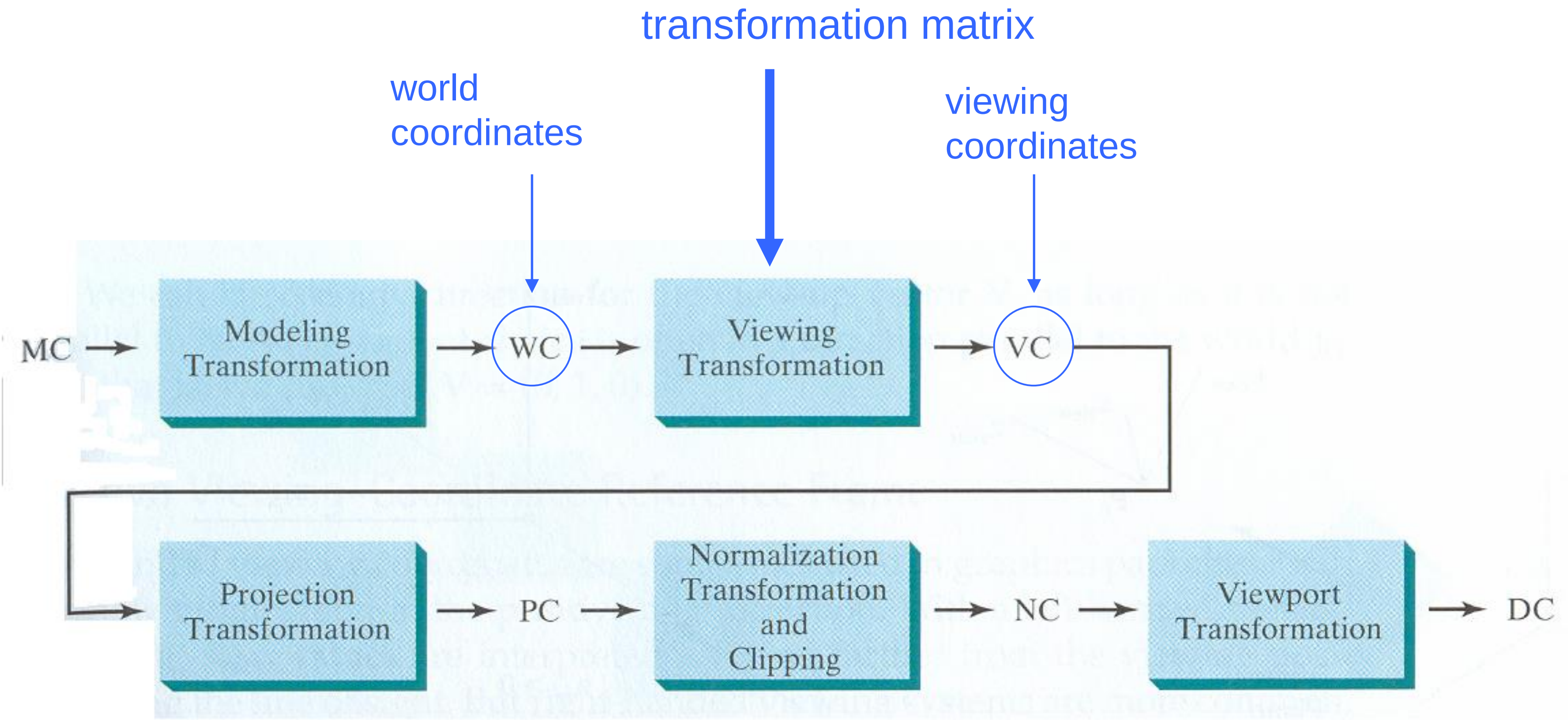


Maintaining the viewpoint and varying the direction of **N** we can show models positioned around the viewpoint and making up the scene

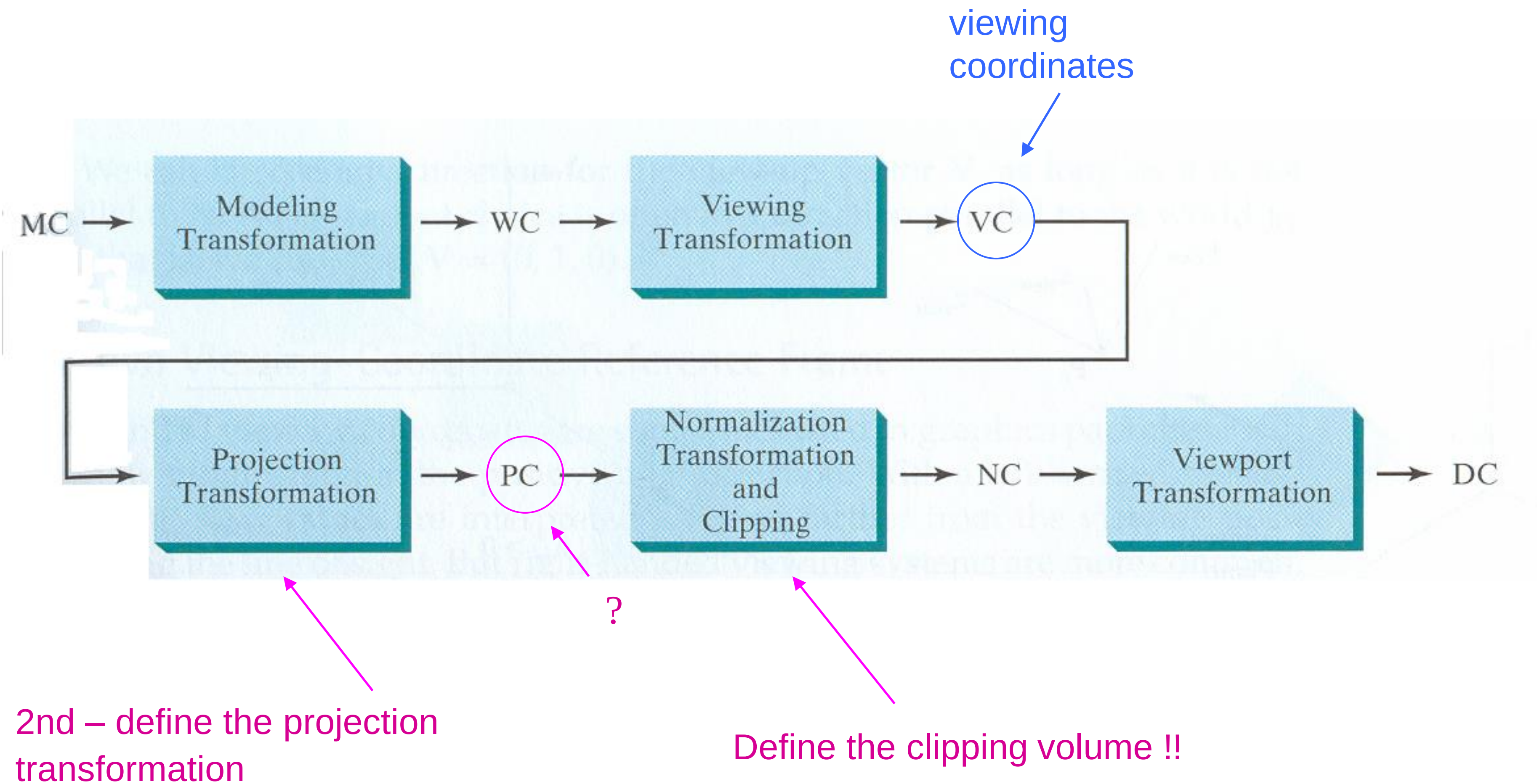


Keeping the direction of N and displacing the viewpoint we obtain a *panning* effect

• When the viewing parameters are known, it is easy to determine the transformation matrix that maps world coordinates (WC) into viewing coordinates (VC)



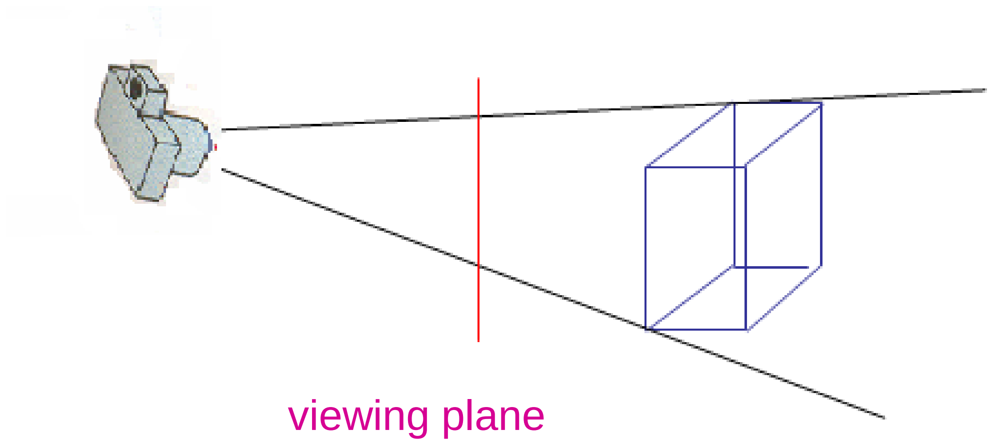
3D Viewing Pipeline



PLANAR GEOMETRIC PROJECTIONS

Projections

- The 3D scene is projected onto the 2D viewing plane
  (information will be lost !!)



There already are 3D display devices but, in most cases, 2D display devices are used

Projections

- Planar geometric projections are obtained using

  projecting straight lines and planar surfaces

- There are other projection types…
- The main classes of planar geometric projections:

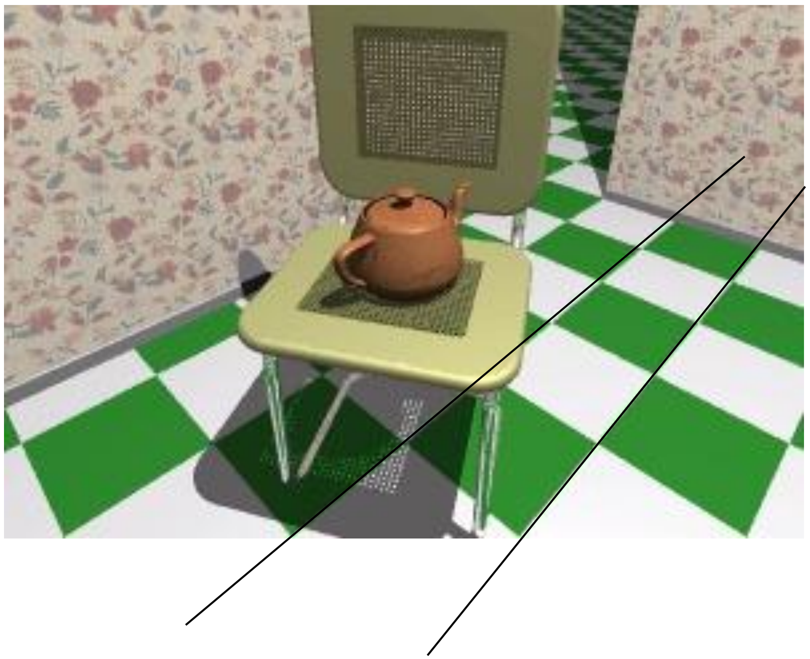
  - Parallel projections

  - Perspective projections

- Perspective projections generate more realistic images
- But imply more calculations and are not always the best option

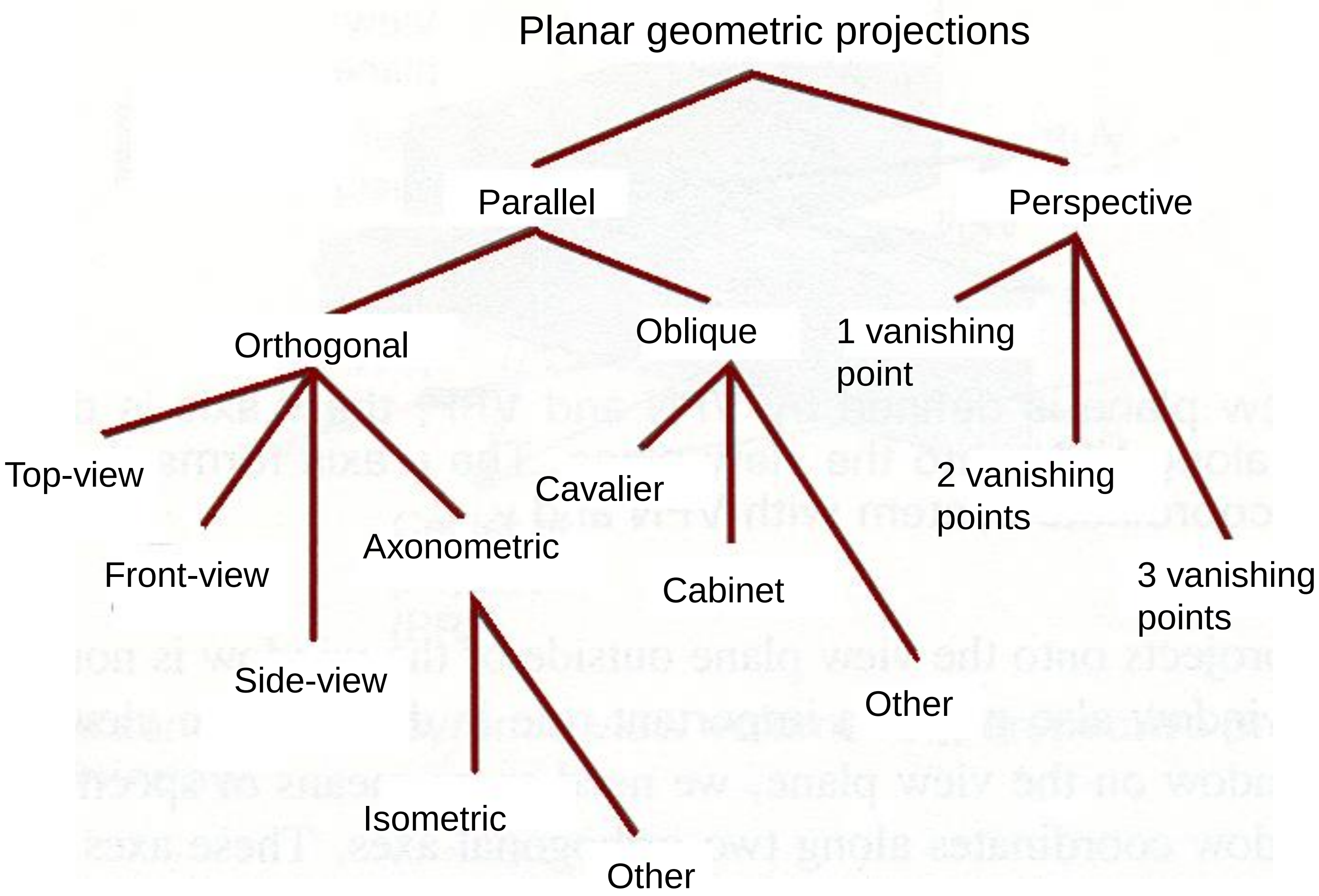
Parallel and perspective projections

parallel projection

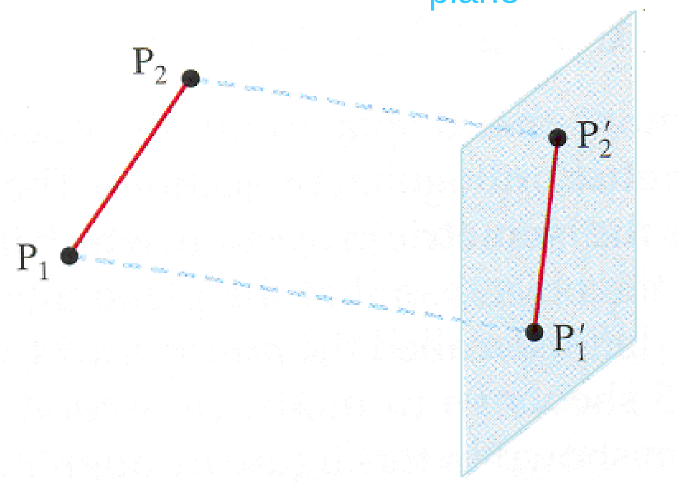


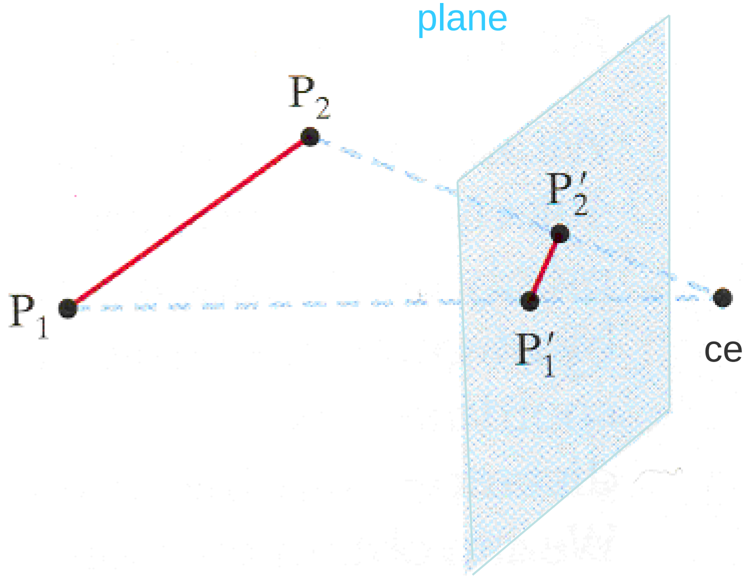
perspective projection

Projections



Parallel projection vs Perspective projection





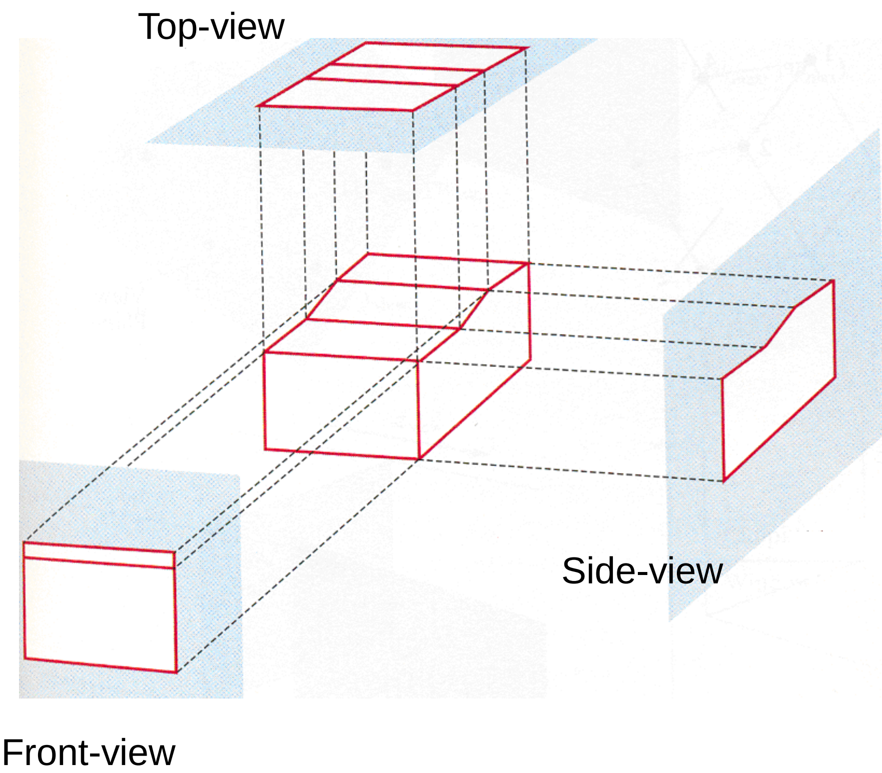
For parallel projections, the projector straight-lines are parallel, i.e., converge at an indefinite distance

For perspective projections, the projector straight-lines converge at the projection center

ORTHOGONAL PARALLEL PROJECTIONS

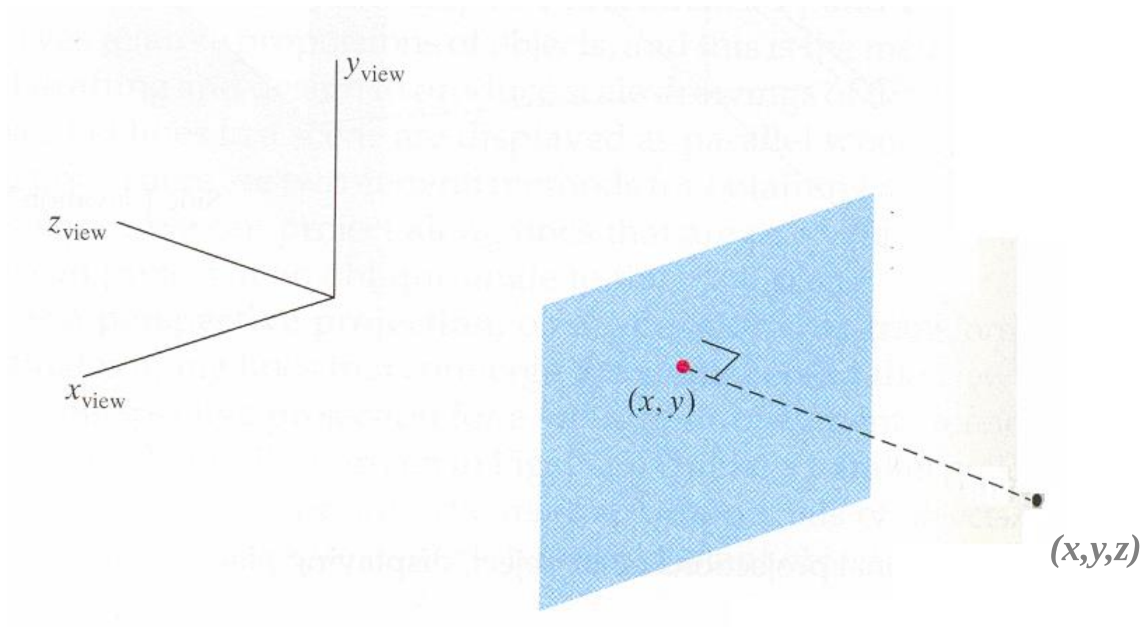
Orthogonal Parallel Projections (Orthographics)

- The projectors are perpendicular to the projection plane
- The projection plane is parallel to a set of the object's faces
- Some angles, lengths and areas can be directly measured
- The views might not convey the 3D structure / shape of the objects
- Frequently used in Engineering and Architecture

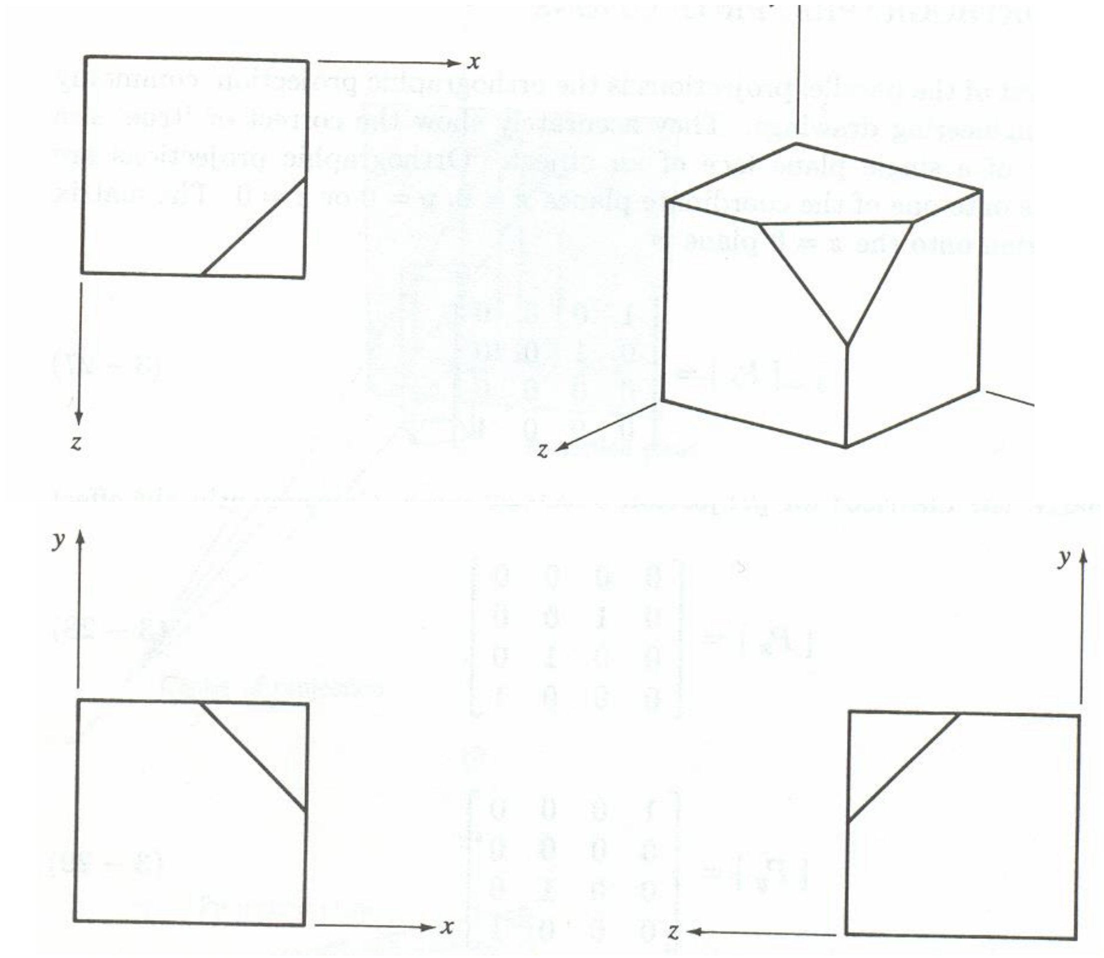


Orthogonal projection coordinates

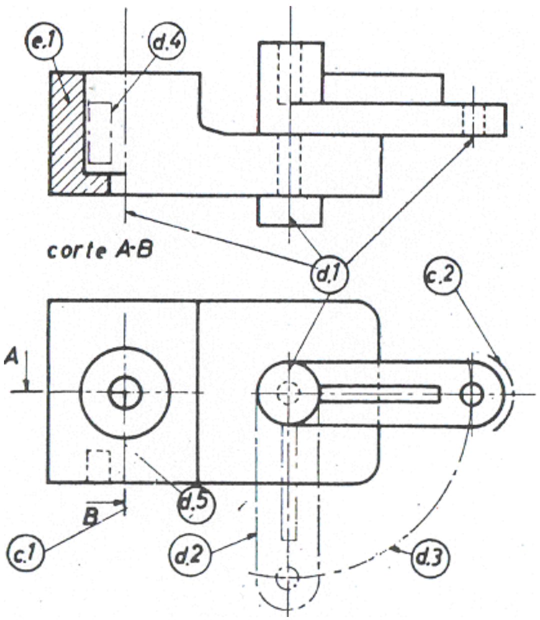
- If the direction projection is parallel to the ZZ' axis, what are the coordinates of the projected point ?

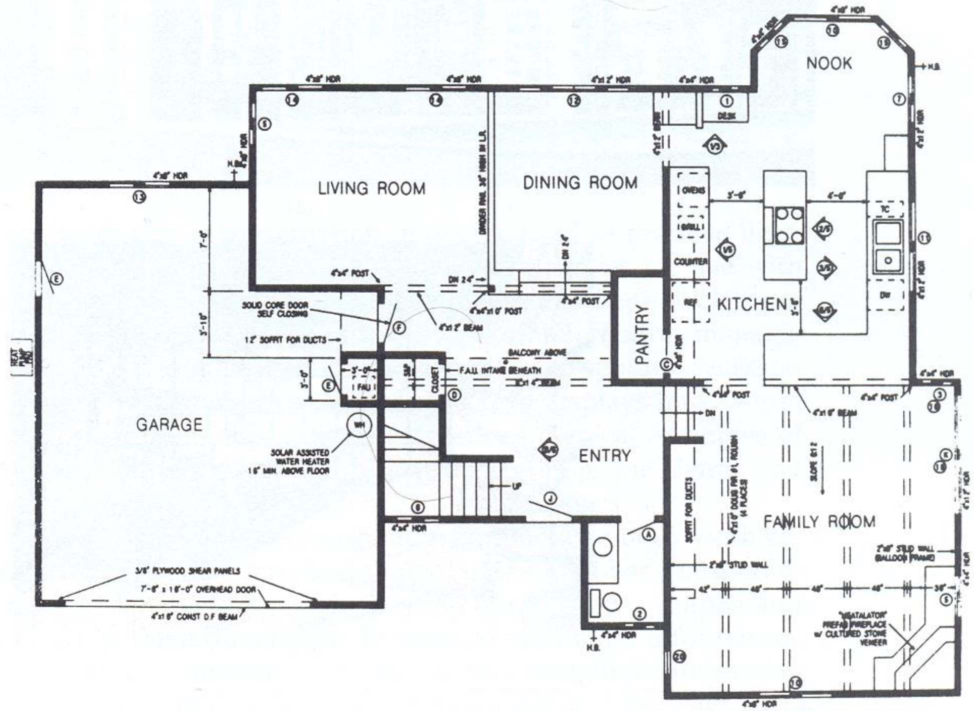


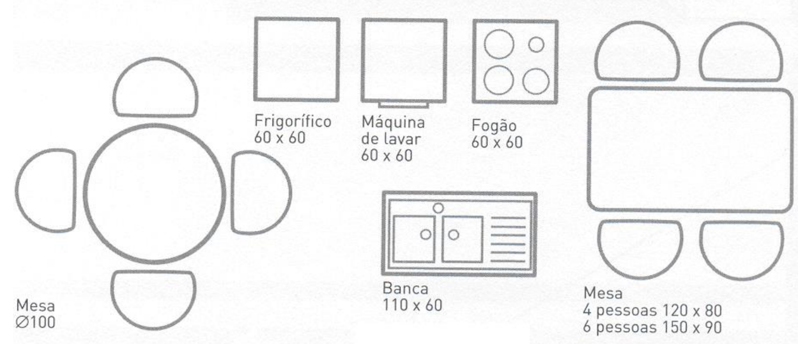
Orthogonal Parallel Projections



Top-views and Side-views

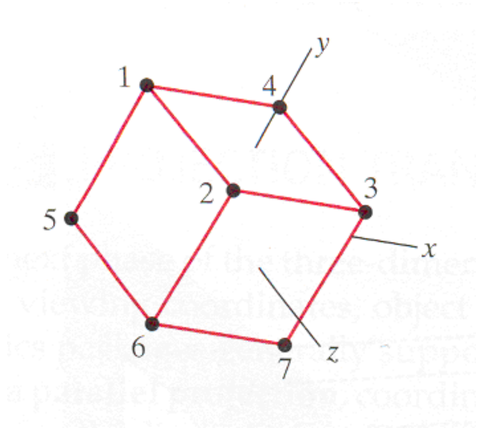


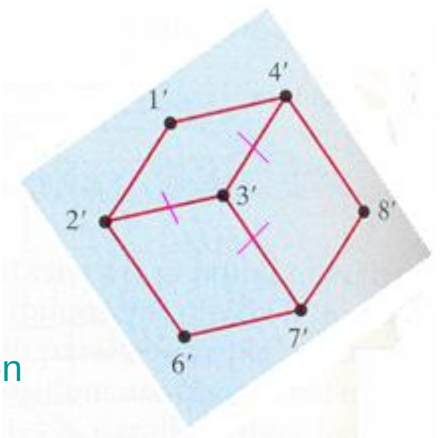




Axonometric Projections

- Orthogonal parallel projections, where the projection plane in not parallel to a set of the object's faces
- Give a better idea of the object's 3D structure / shape
- 3 classes
    - Isometric
    - Dimetric
    - Trimetric

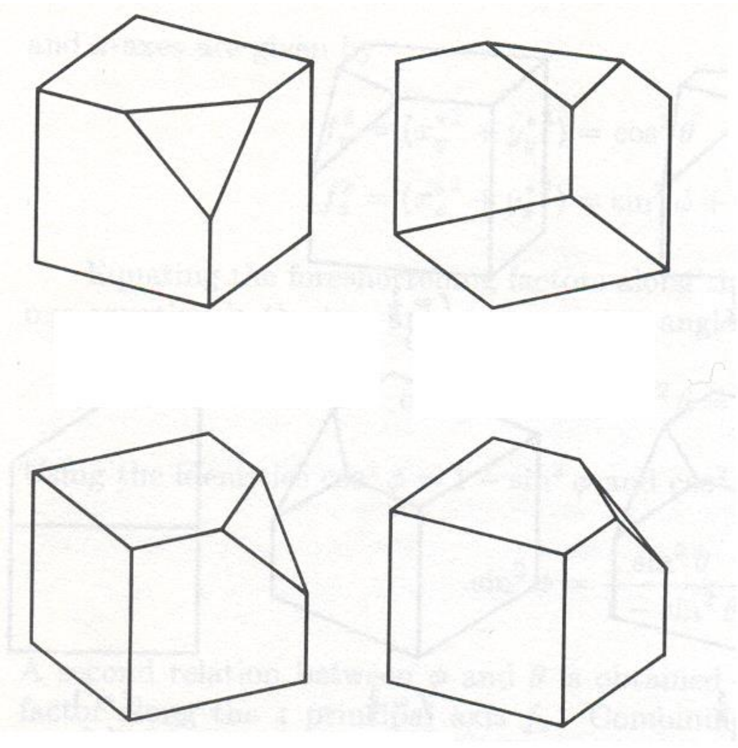




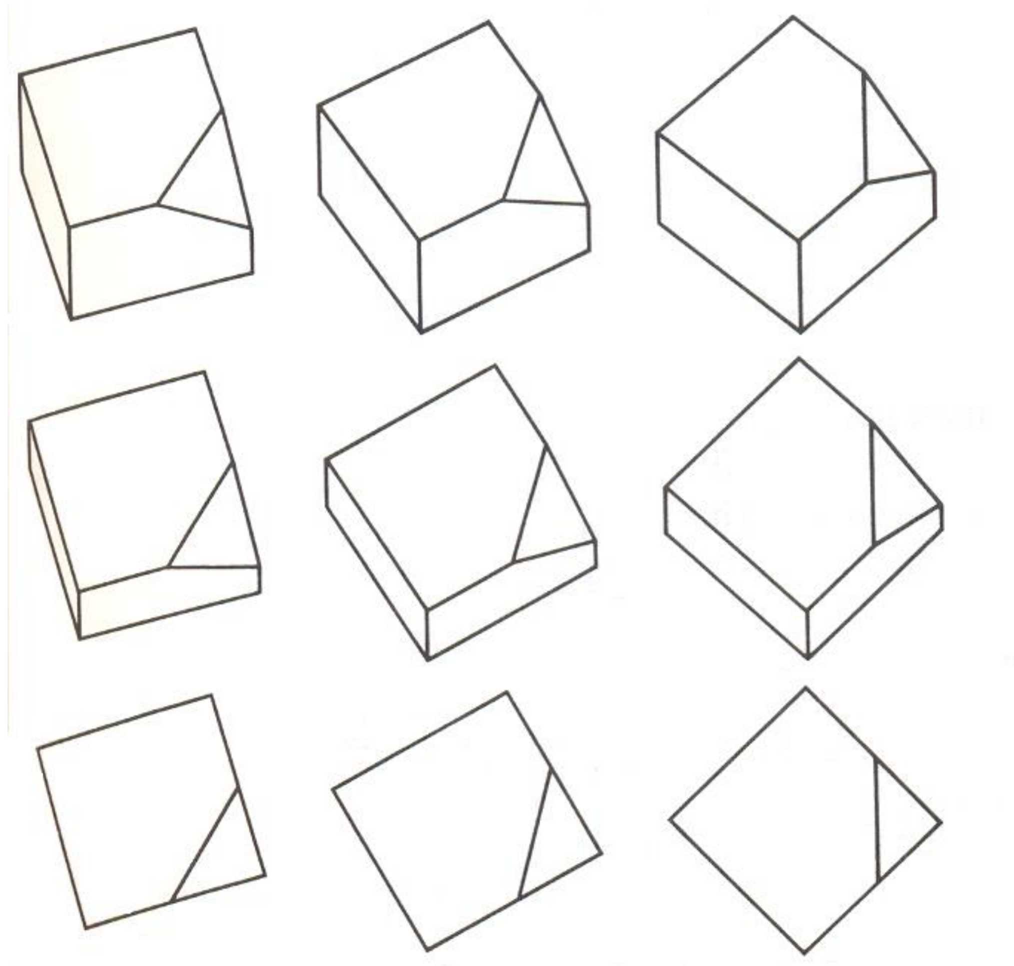
Projection plane

Isometric projection of a cube:
3 faces are shown and all edges have the same length

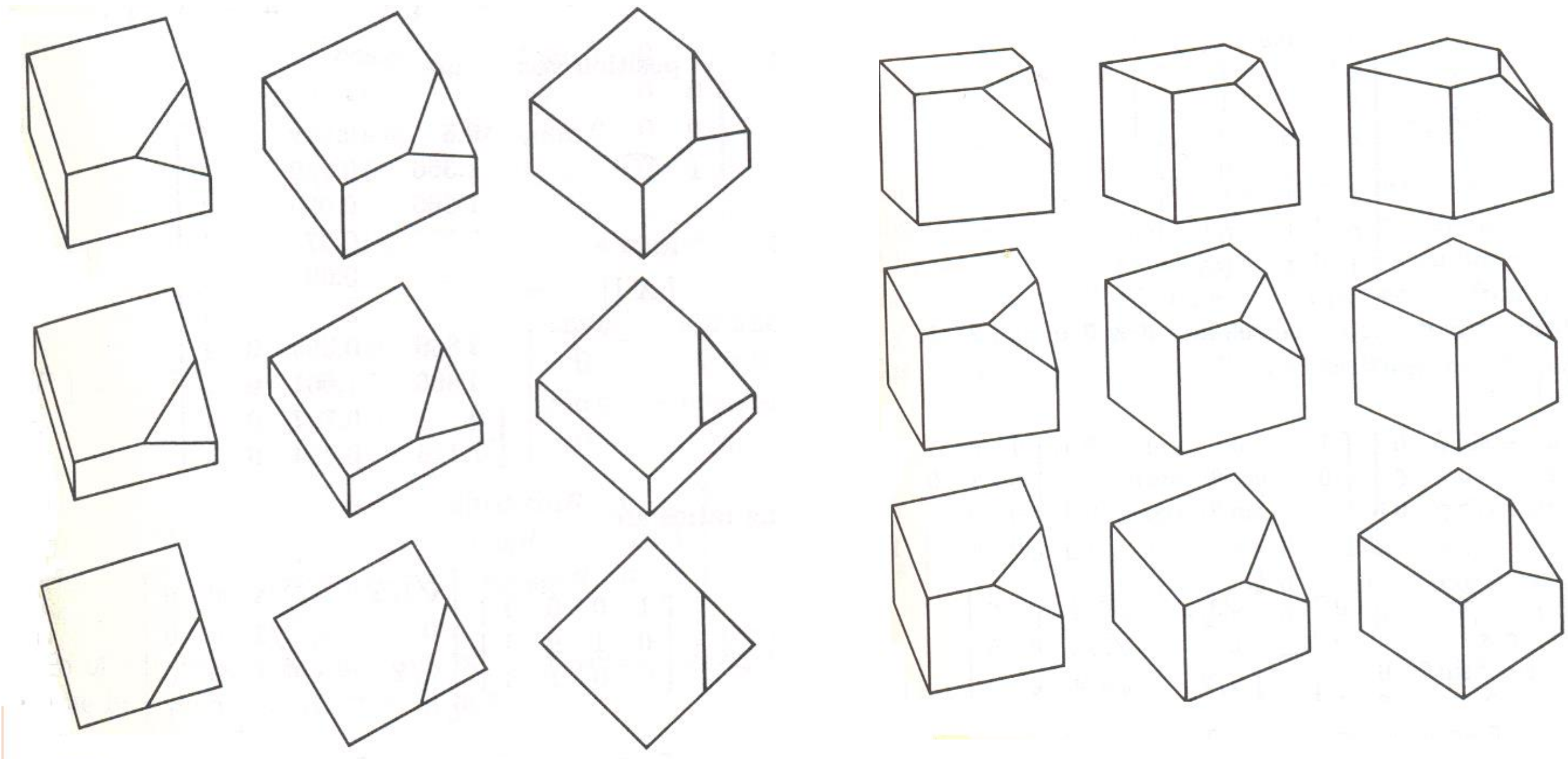
Isometric Projections

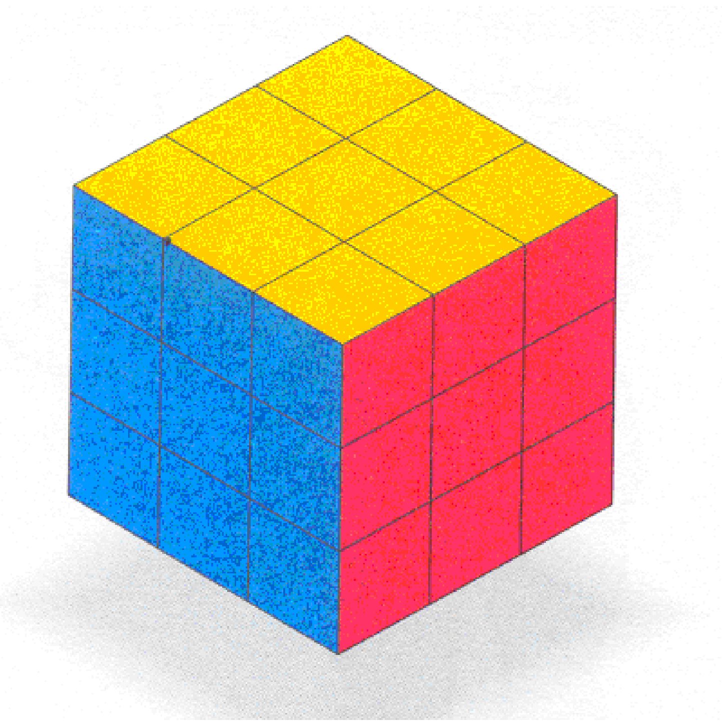


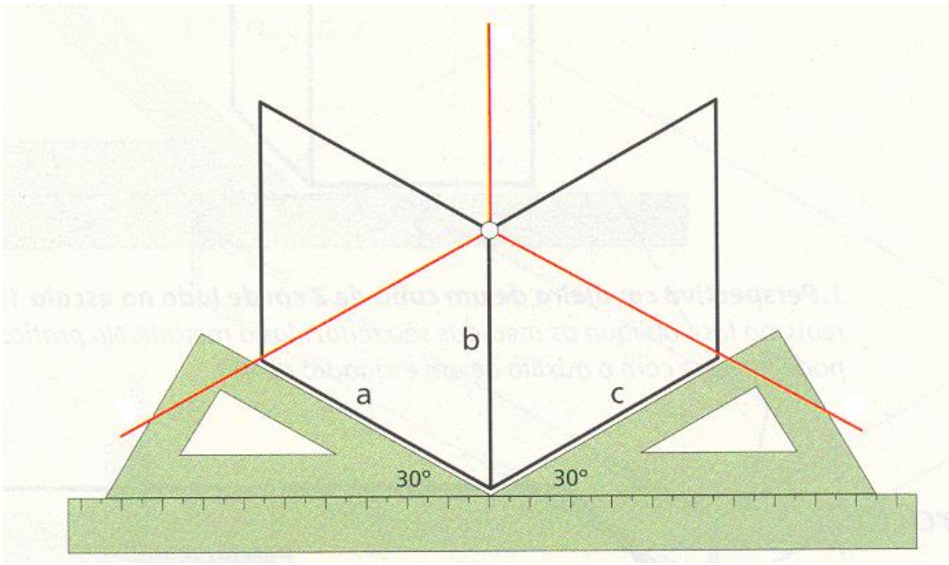
Dimetric Projections

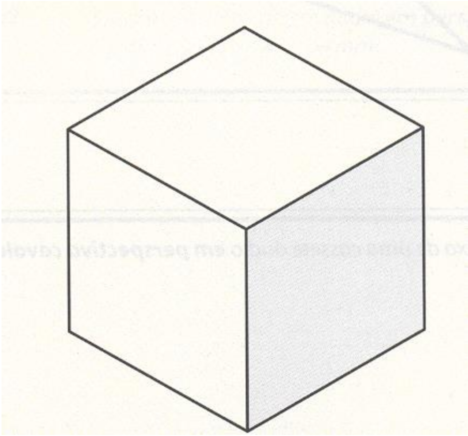


Trimetric Projections



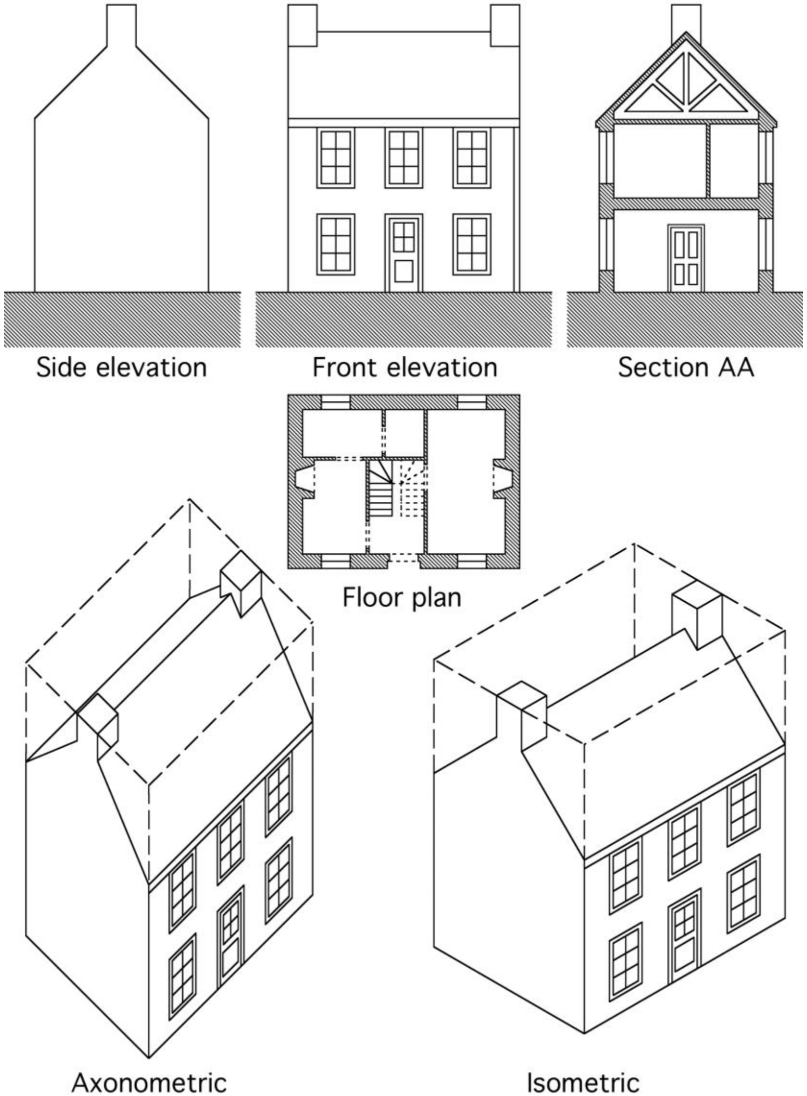






Drawing an isometric projection

Orthographic projections



[van Dam]

Orthographic projections

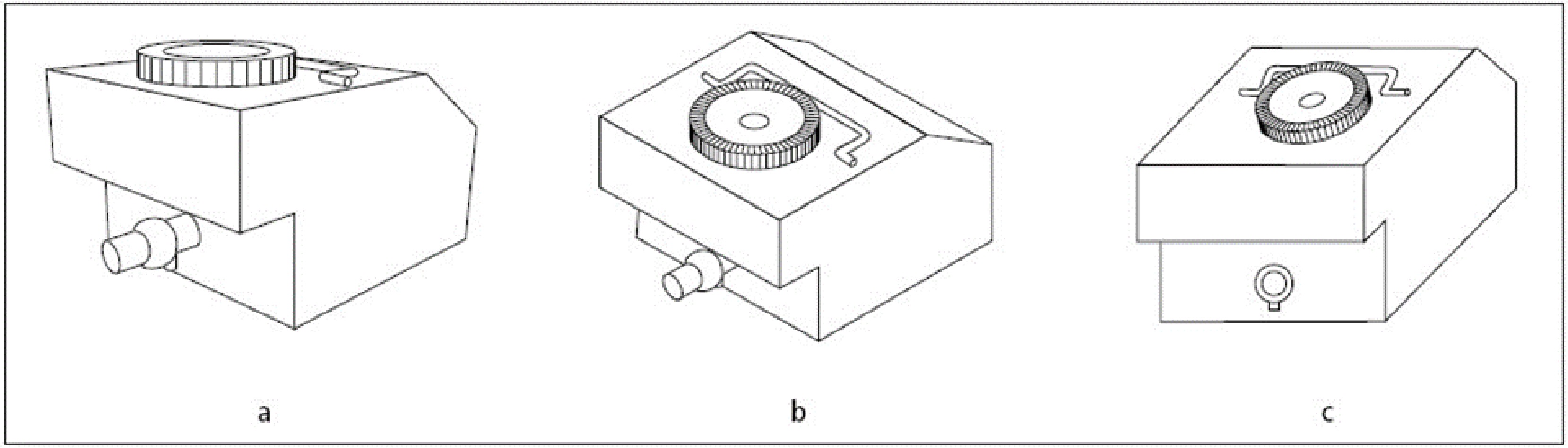
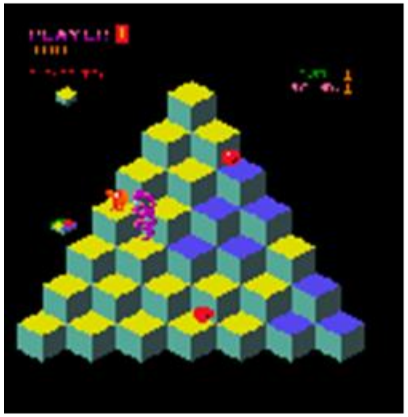
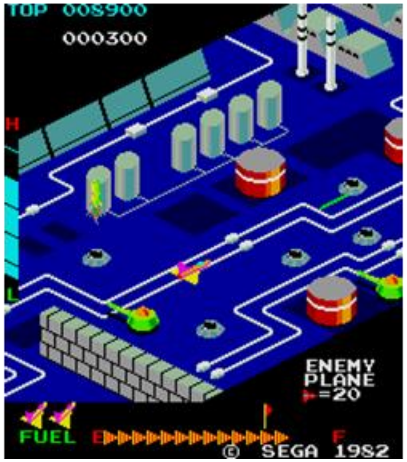


Figure 2-17. (a) Perspective, (b) isometric, and (c) oblique drawings.

[van Dam]

Axonometric Projection in Games

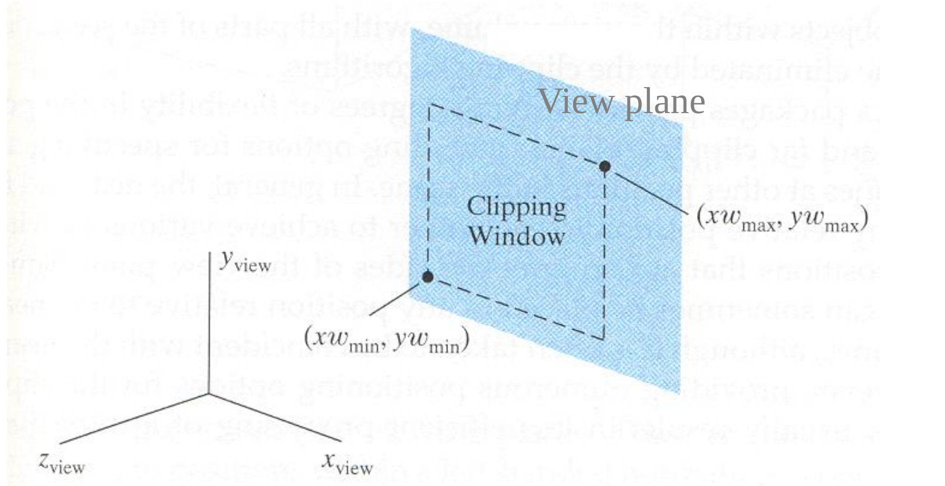


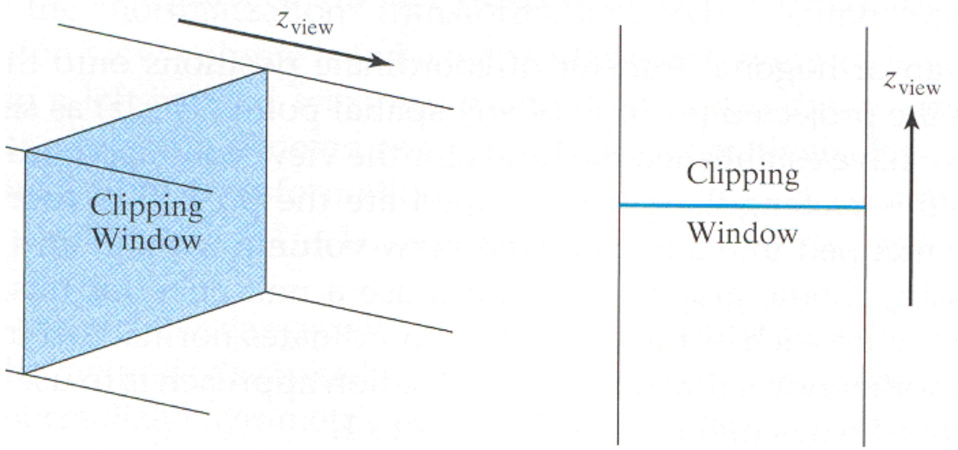


[van Dam]

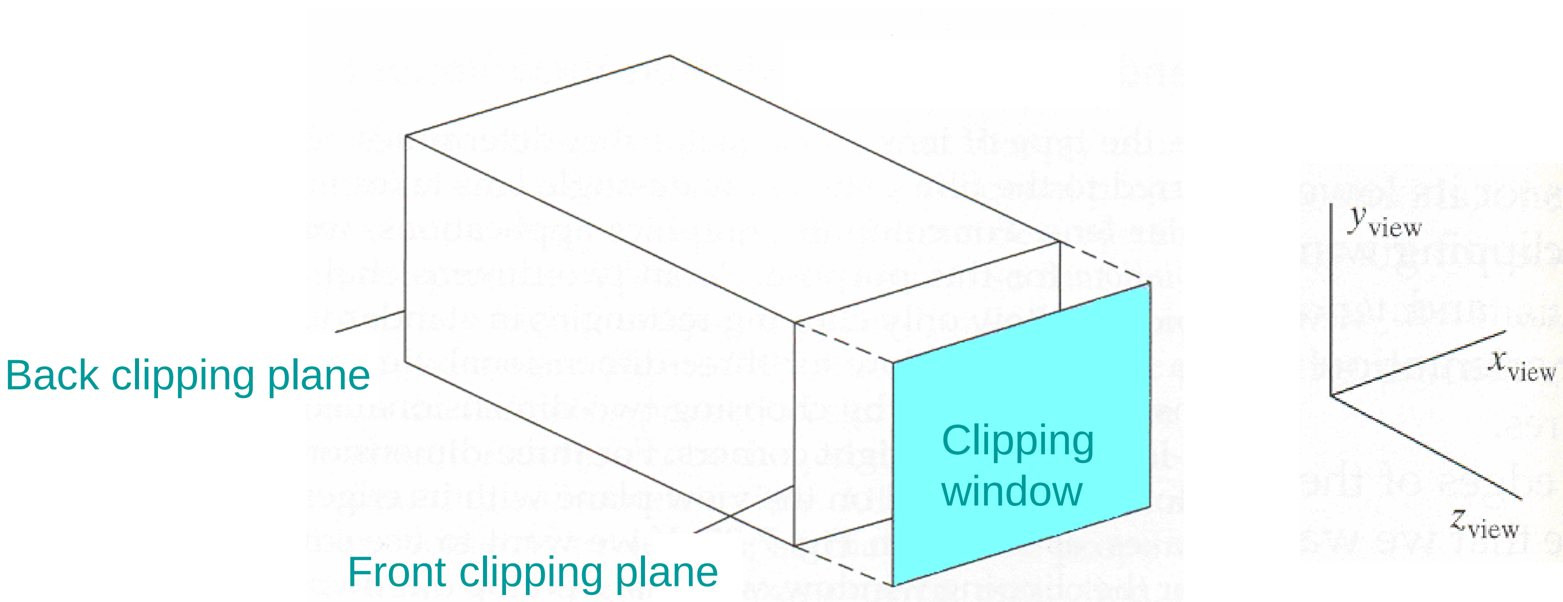
Clipping window and orthogonal projection

- In photography, the lens determines which “amount” of the scene is transferred to the final image
- In CG, it is the clipping window defined on the view plane
- In general, they are rectangular and parallel to the XX’ and YY’ axes
- The clipping window is associated with the view volume



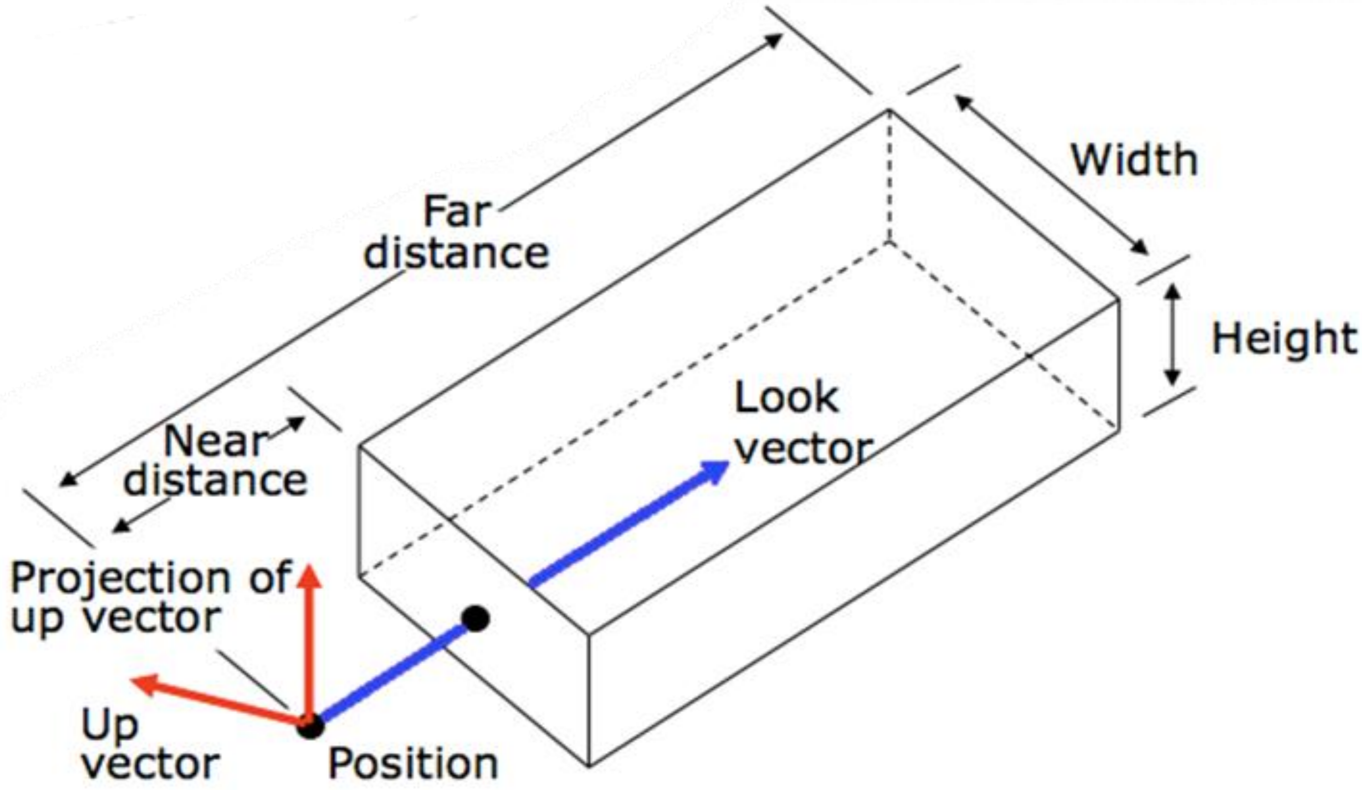


Orthogonal projection view volume



Finite view volume for a orthogonal parallel projection, with front and back clipping planes

The Parallel View Volume

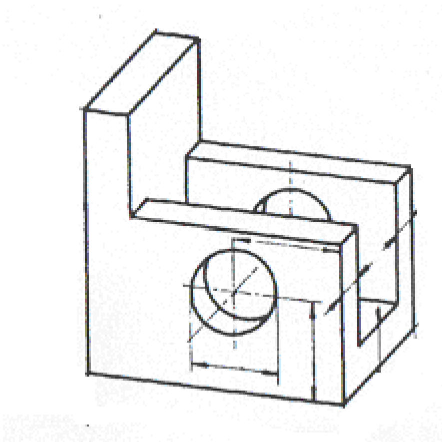


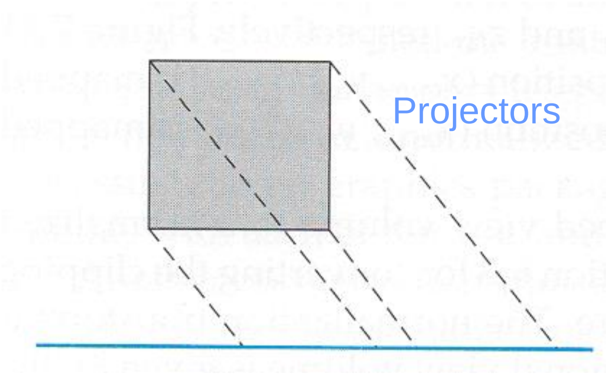
[van Dam]

OBLIQUE PARALLEL PROJECTIONS

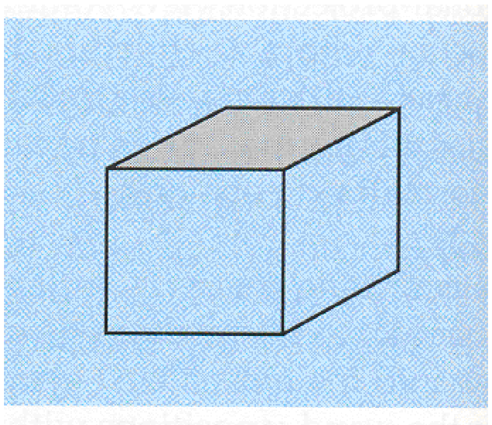
Oblique Parallel Projections

- The projectors are oblique regarding the projection plane
- Often used in Engineering:
  - easy to draw
  - convey a good idea of shape / structure





Projection plane

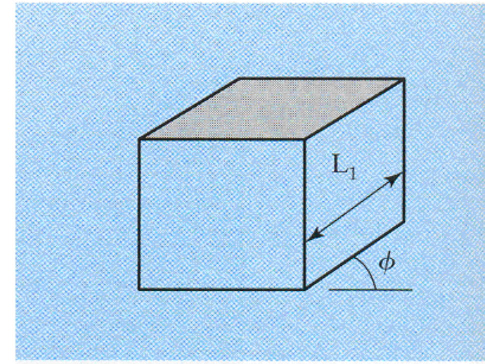


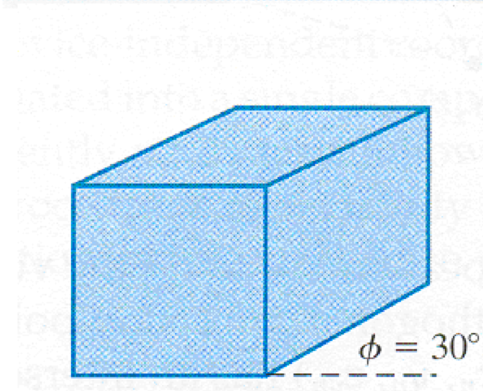
Projection plane

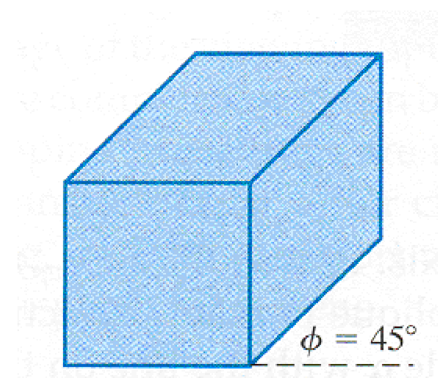
Oblique projection of a cube:
3 faces are shown

Cavalier Projection

- Length (L_1) of the cube's edges is preserved
- Does not look realistic
- The angle Φ is usually:
  - Φ = 30º
  - Φ = 45º





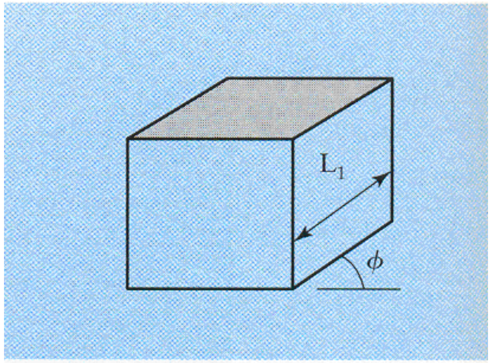


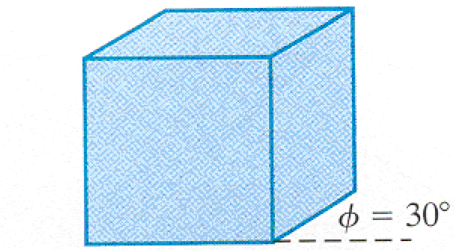
Cabinet Projection

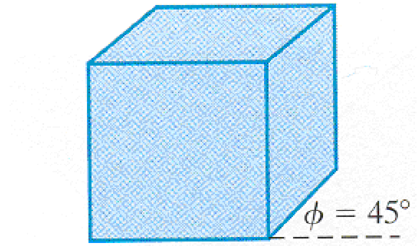
- Depth of the cube (L_1) is represented with a 0.5 scale factor

- Looks more realistic

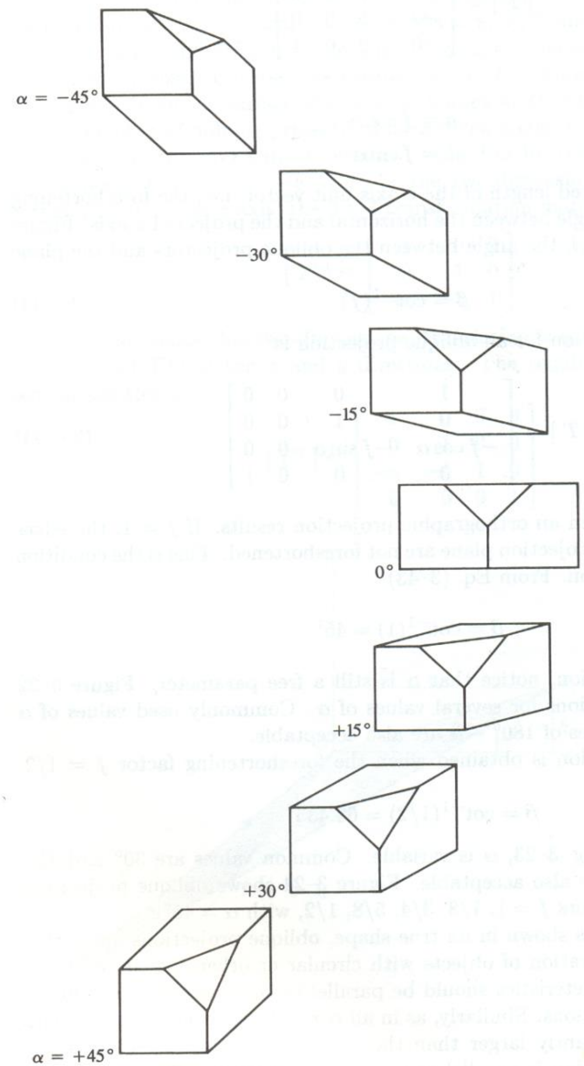
- The angle Φ is usually :
  - Φ = 30º
  - Φ = 45º

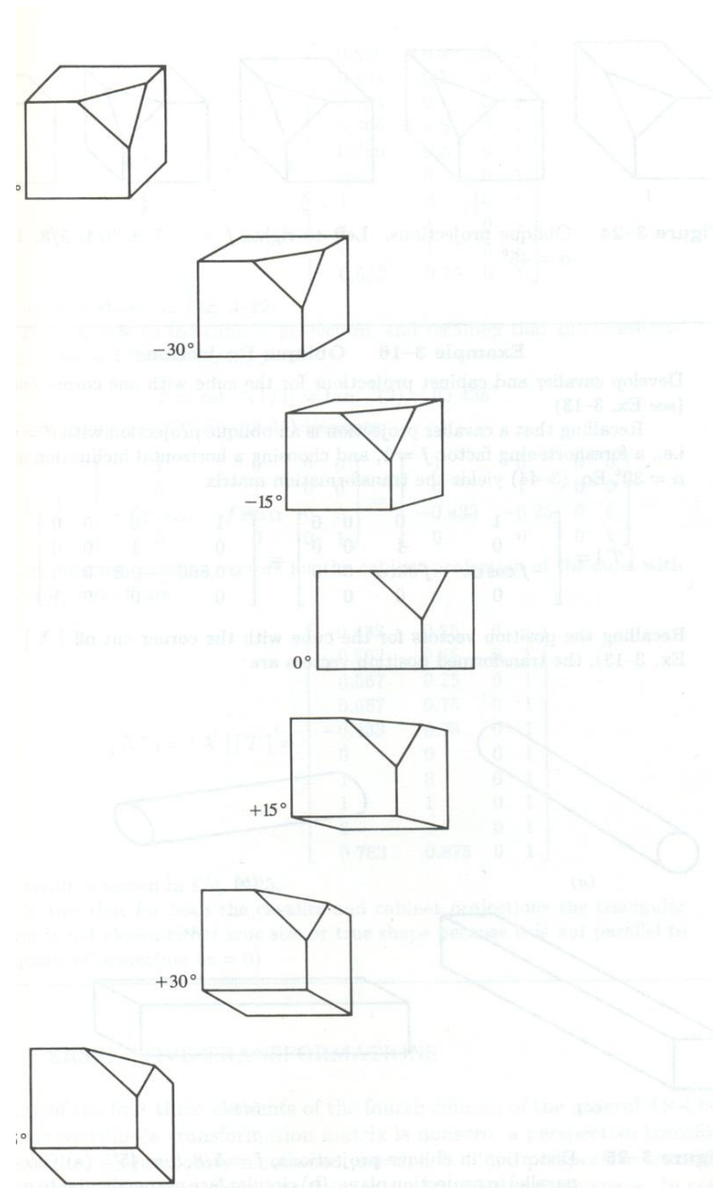




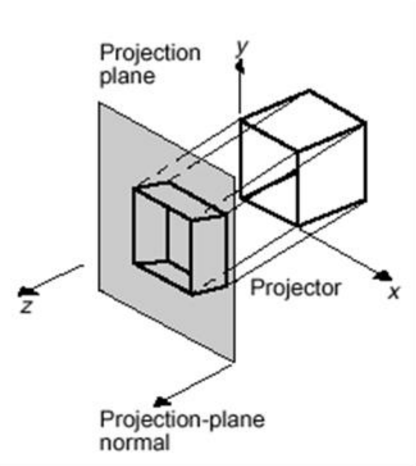


Cavalier and Cabinet Projections

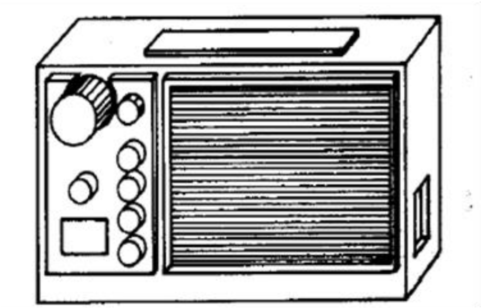




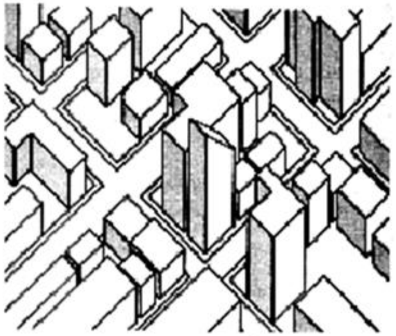
Examples



Construction of
oblique parallel projection



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



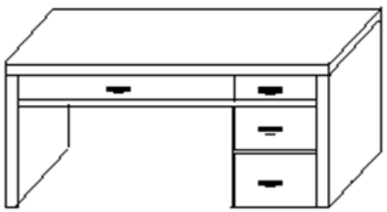
(Carlbom Fig. 2-6)

Plan oblique projection of city

[van Dam]

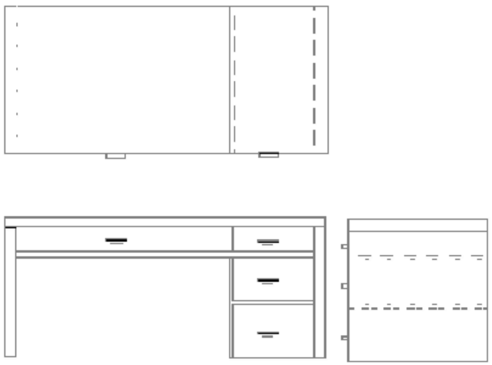
Examples

Cavalier



Cabinet

Carlbom Fig. 3-2



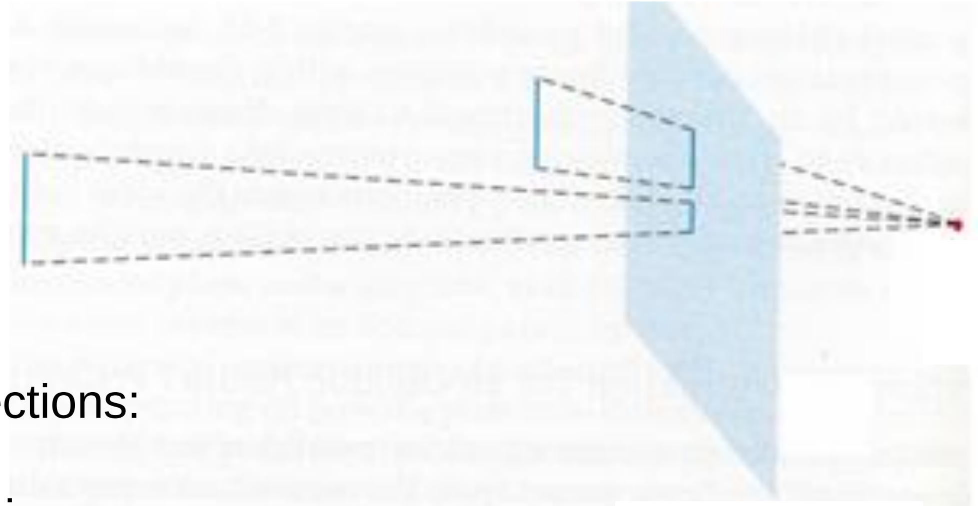
Multiview Orthographic

[van Dam]

PERSPECTIVE PROJECTIONS

Perspective Projection

- The projections of straight-line segments with the same length, but located at different distances from the projection plane, are projected with different lengths



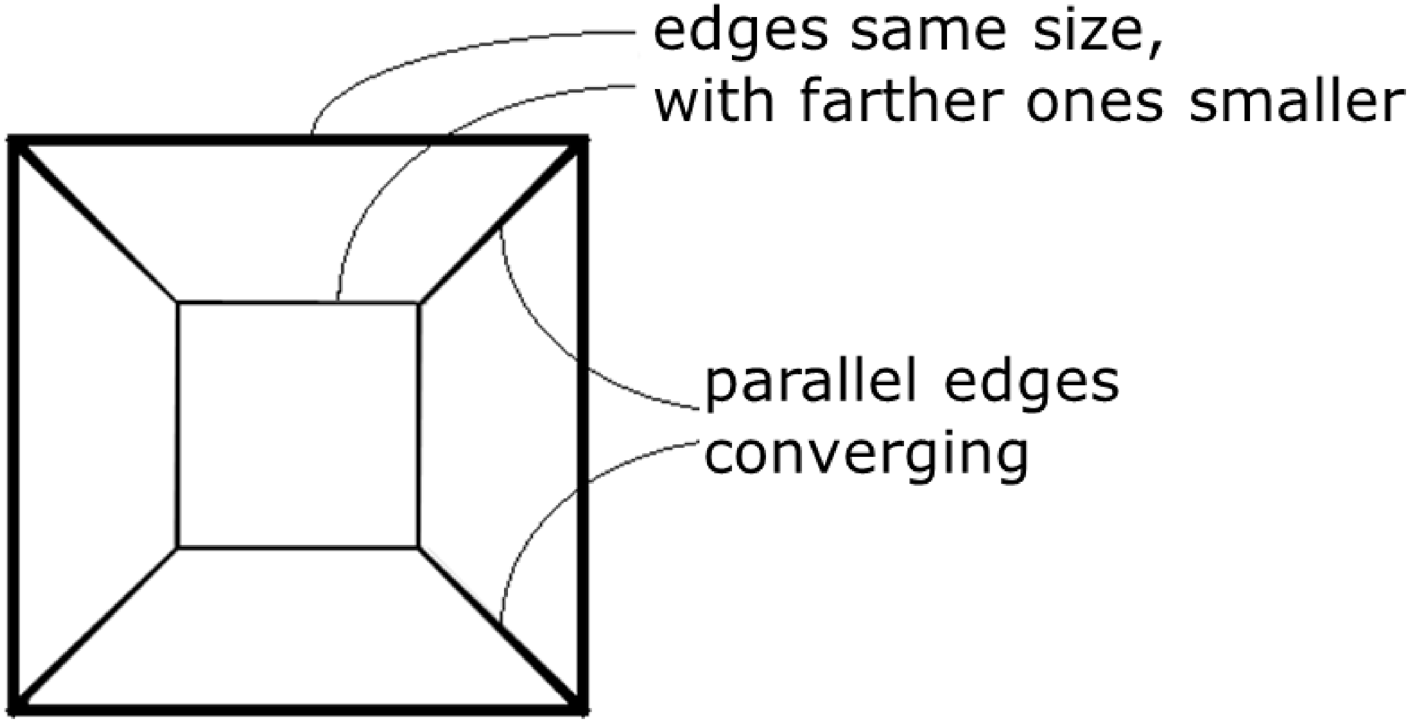
- Regarding the parallel projections:

  - It generates more realistic images

  - But it does not preserve relative sizes of objects

  - It requires more calculations

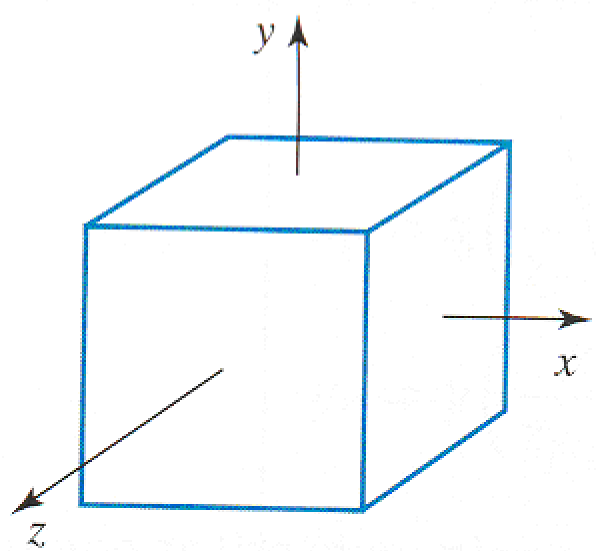
Perspective projection

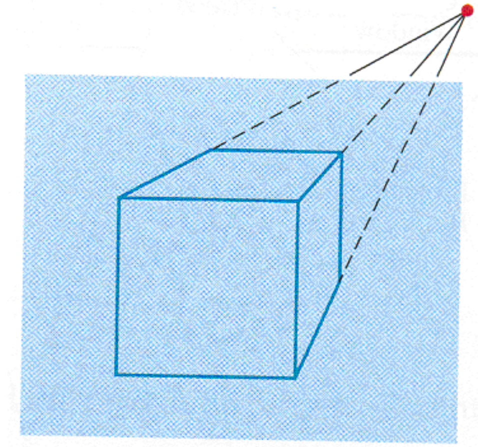


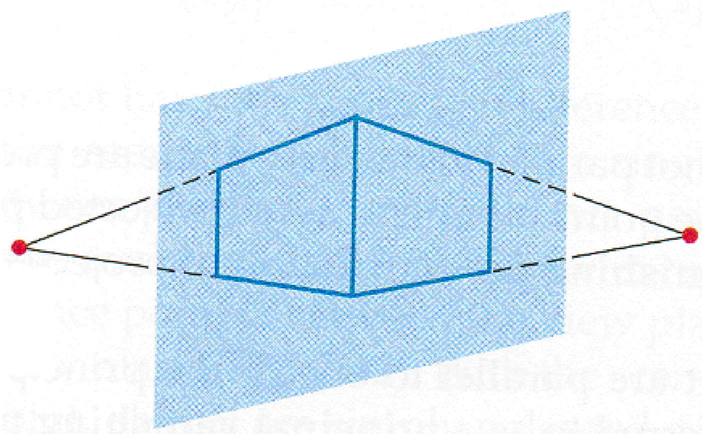
[van Dam]

Perspective projections with 1, 2 or 3 vanishing points

- Straight-lines – parallel to a coordinate axis that intersects the projection plane –, converge to that axis' vanishing point



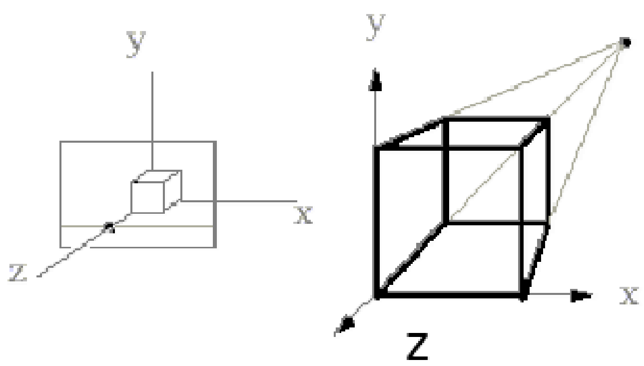




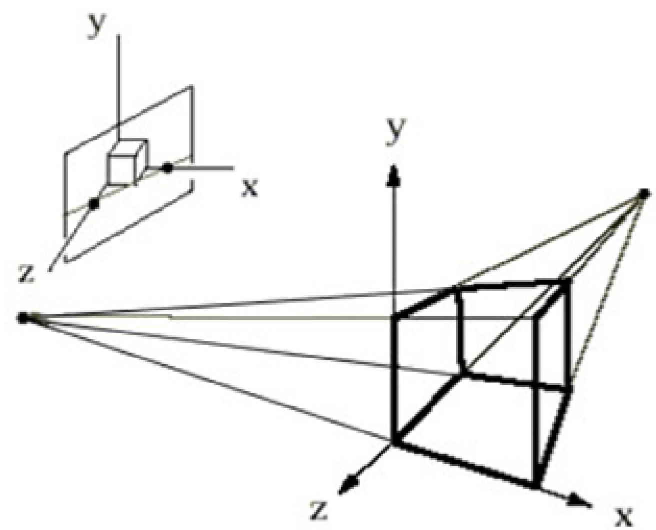
Number of vanishing points:

number of coordinate axes intersecting the projection plane

Vanishing points



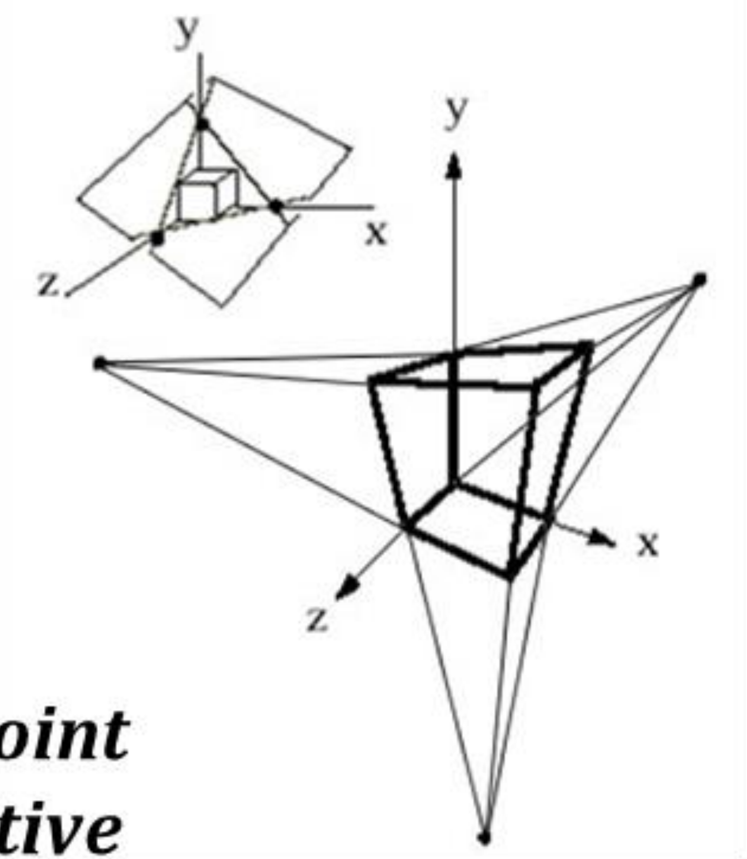
One Point Perspective
(z-axis vanishing point)



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

[van Dam]

Vanishing points

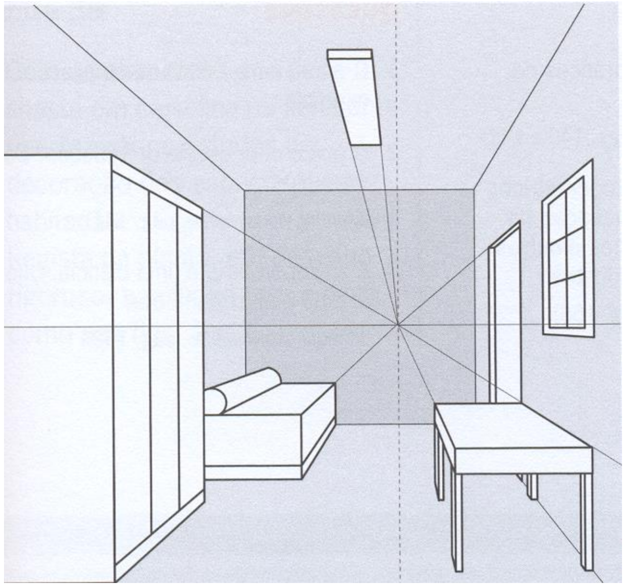


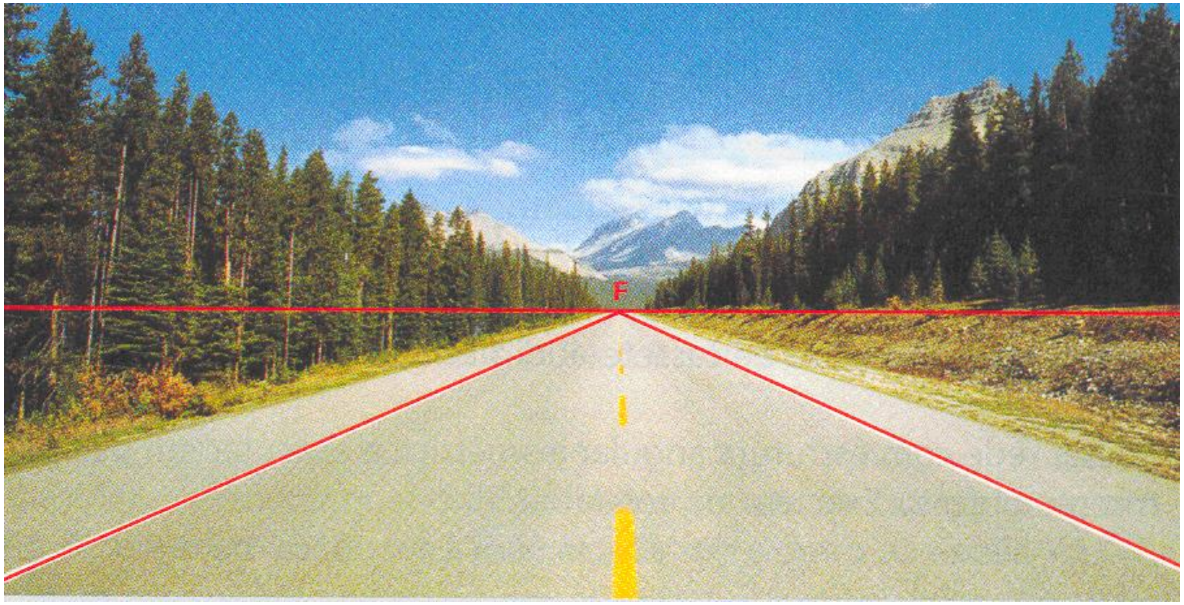
Three Point Perspective

(z, x, and y-axis vanishing points)

[van Dam]

Perspective with 1 vanishing point





Perspectives with 1 and 2 vanishing points (frontal and angular)

Frontal perspective

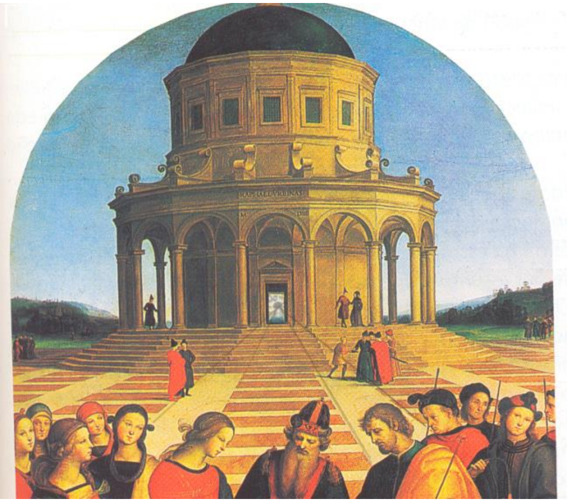
Angular perspective

Perspective in Art

The Trinity and the Virgin
Mastaccio, 1427

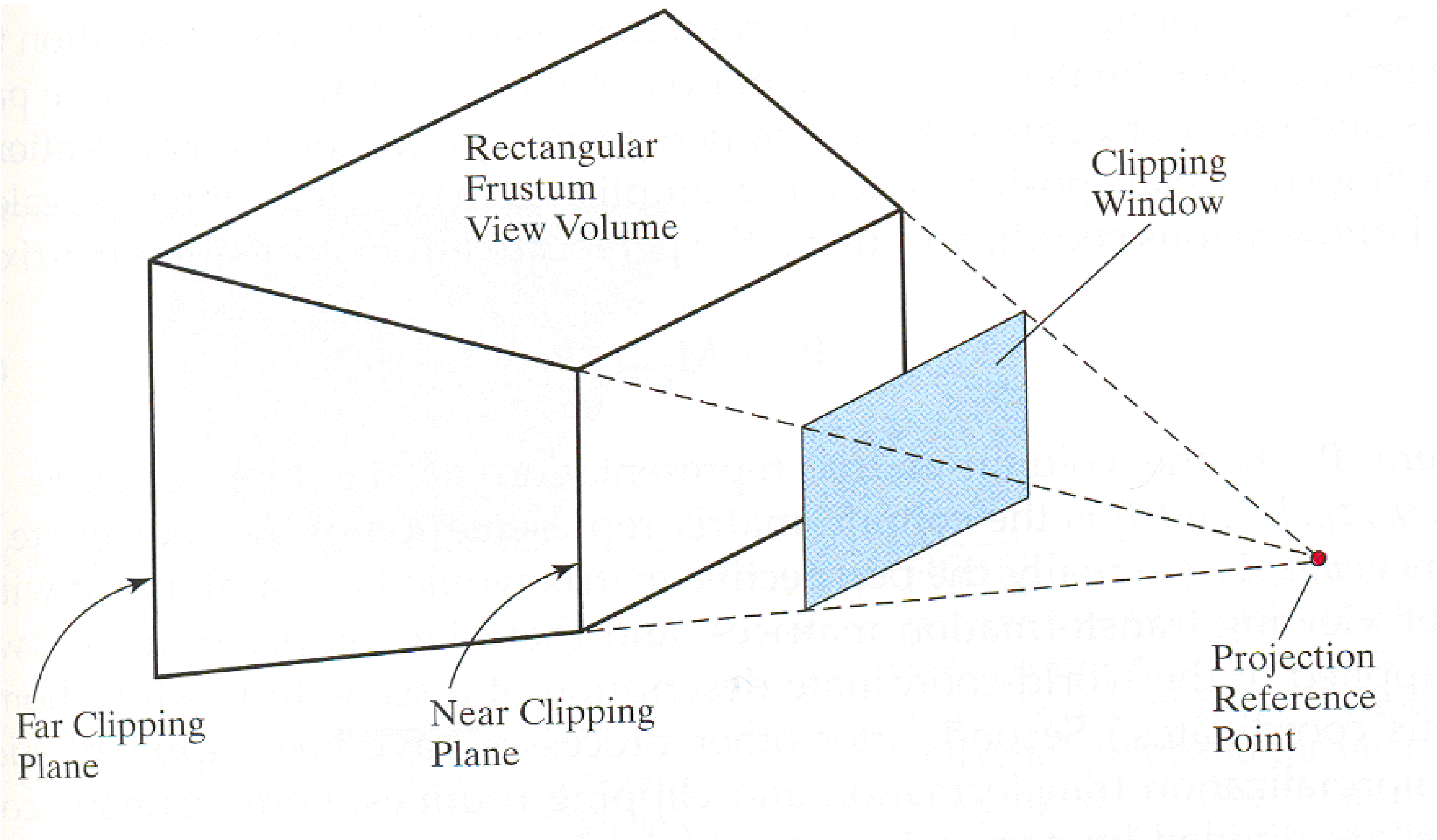
Considered the first
painting with perspective

Perspective with 1 vanishing point

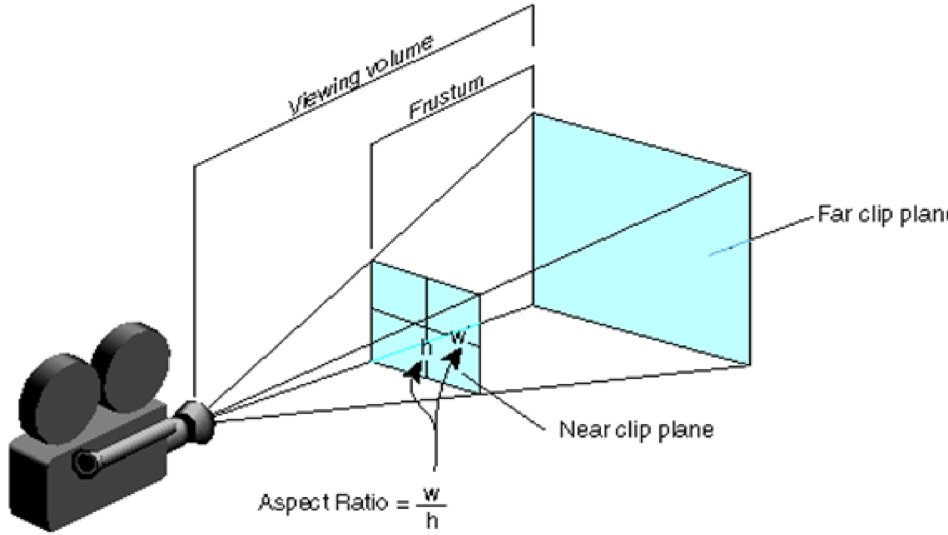


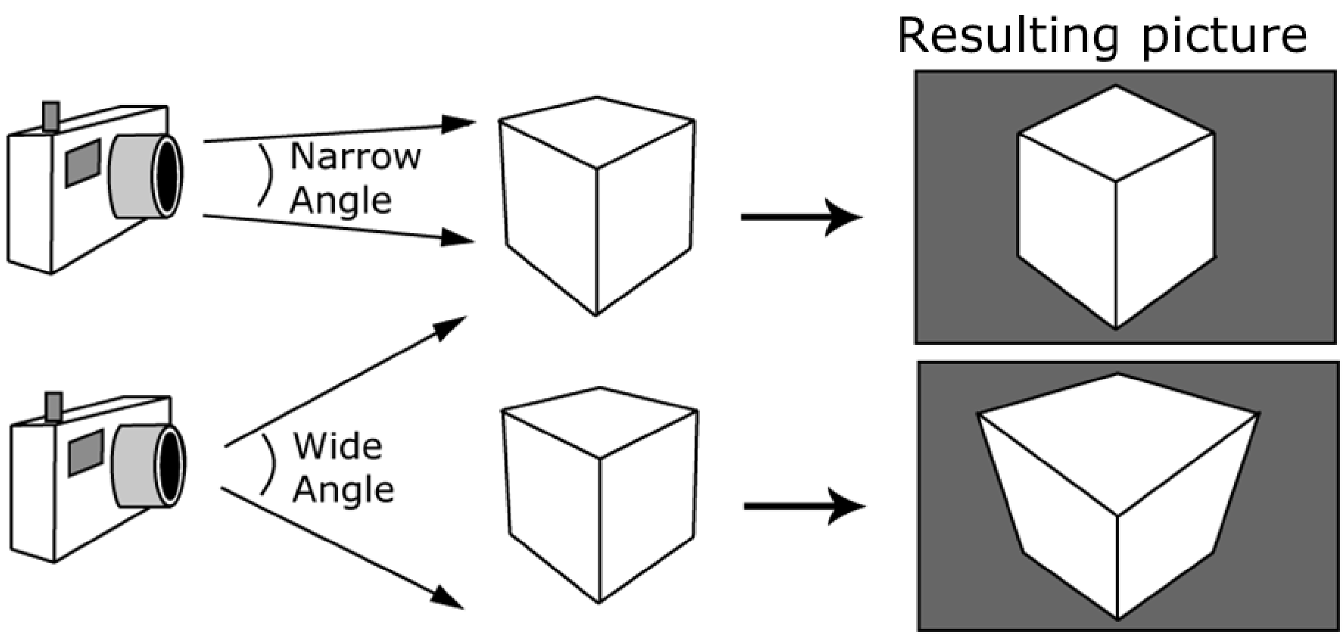
From http://www.sanford-artedventures.com

View Volume and Clipping Window for Perspective Projection



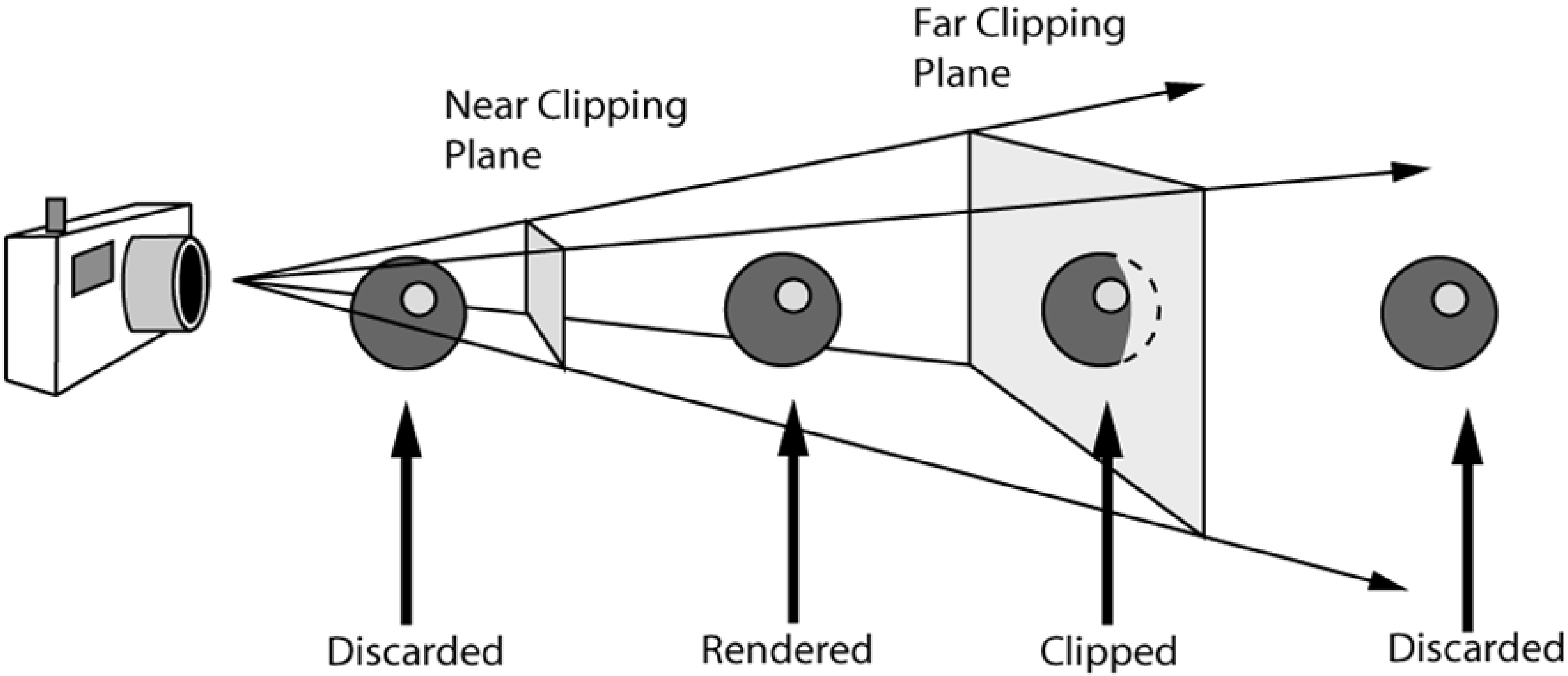
View Angle





[van Dam]

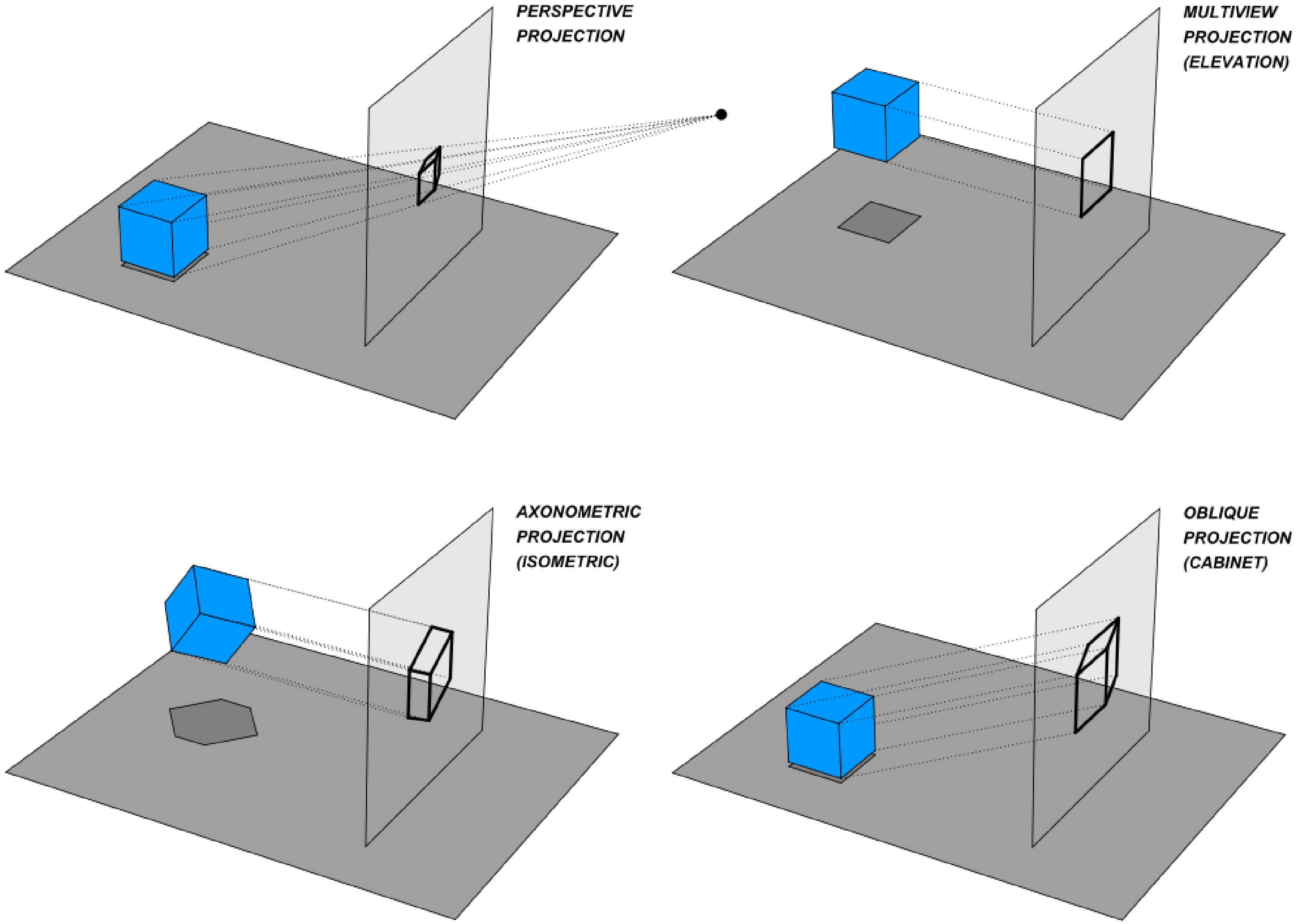
Clipping planes



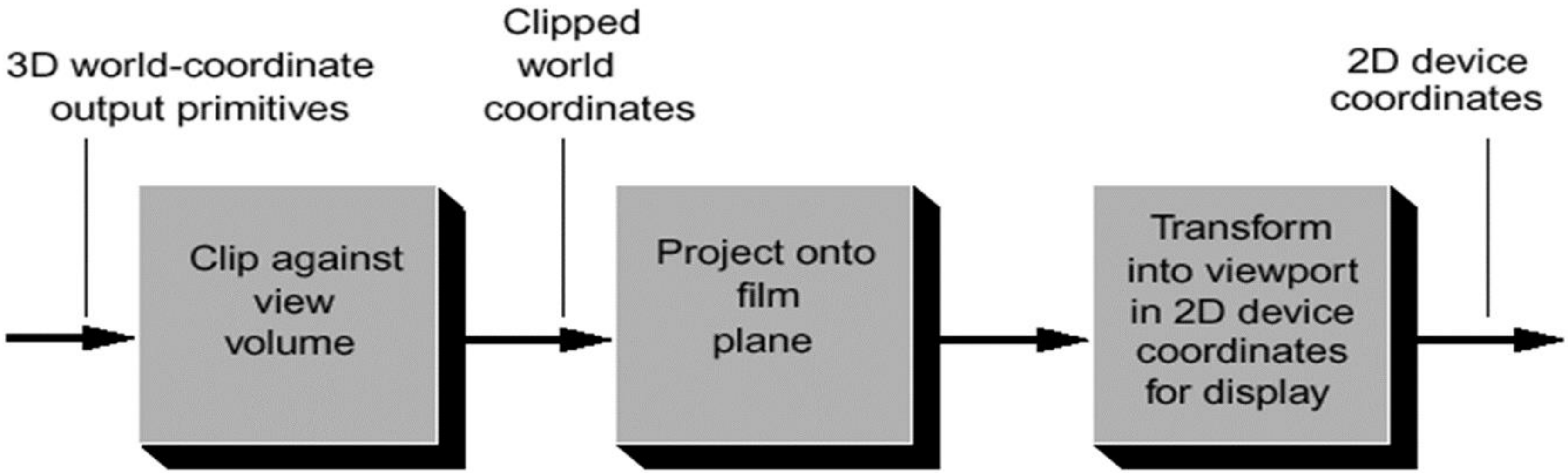
[van Dam]

A 3D scene generated using perspective projection

Cube – Various projections



How to project ?



[van Dam]

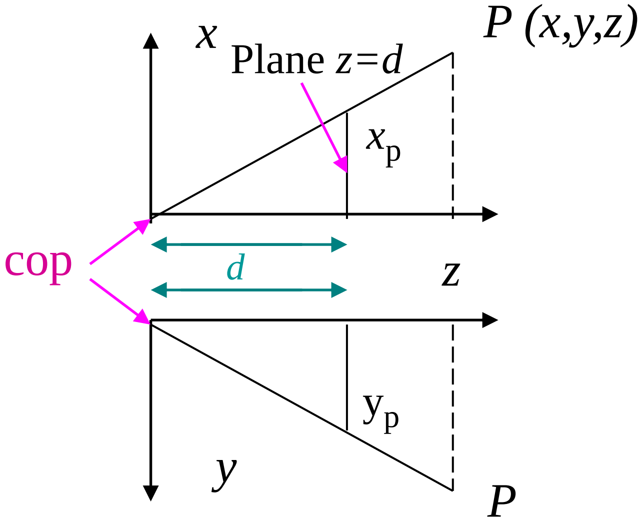
MATRICIAL REPRESENTATION

The Mathematics of Planar Projections

- A projection can be achieved through matrix multiplication, using a (4 x 4) projection matrix in homogeneous coordinates
- The projection matrix can be concatenated with the model-view matrix to carry out any modeling transformations before the actual projection
  - Animations
  - More complex projections are decomposed into a sequence of simpler transformations
- Let's consider the simplest cases, when the projection plane is XOY or a plane parallel to XOY

Perspective projection with projection plane at z = d and center of projection at (0, 0, 0)

$P\ (x, y, z)$ – original point
$P_p\ (x_p, y_p, z_p)$ – projected point



Distance ratios:

$x_p / d = x / z$ $\qquad$ $y_p / d = y / z$

Multiplying by d:

$$x_p = \frac{d \cdot x}{z} = \frac{x}{z / d}$$

$$y_p = \frac{d \cdot y}{z} = \frac{y}{z / d}$$

Dividing by z implies that objects further away appear smaller

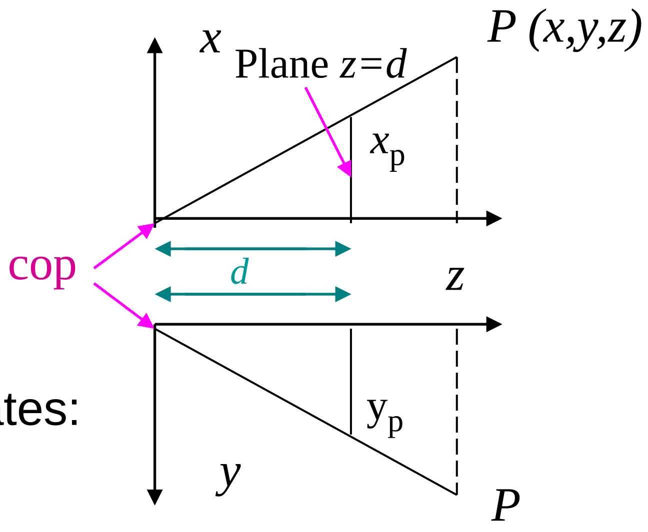
Perspective projection with projection plane at z = d and center of projection at (0, 0, 0)

$P\ (x, y, z)$ – original point

$P_p\ (x_p, y_p, z_p)$ – projected point

All z values are possible except $z=0$

The projection matrix in homogeneous coordinates:

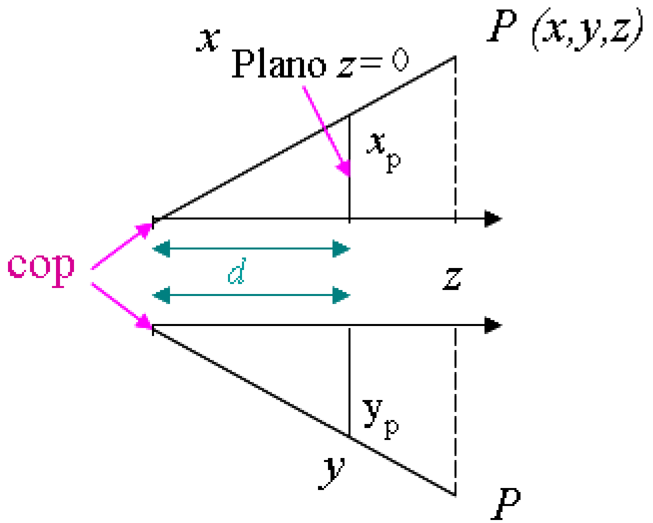


$$M_{pers} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix} \longrightarrow P_p = M_{pers} \cdot P$$

Perspective projection with projection plane at z = 0 and center of projection at (0, 0, -d)

$P\ (x, y, z)$ – original point

$P_p\ (x_p, y_p, z_p)$ – projected point



Distance rations:

$x_p / d = x / (z + d) \qquad y_p / d = y / (z + d)$

Multiplying by d:

$$x_p = \frac{d \cdot x}{z + d} = \frac{x}{z / d + 1}$$

$$y_p = \frac{d \cdot y}{z + d} = \frac{y}{z / d + 1}$$

$$M'_{pers} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/d & 1 \end{bmatrix}$$

This matricial representation allows to replace d with ∞, and we obtain the matrix for the orthogonal, parallel projection on the projection plane $z = 0$:

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

What are the coordinates of a projected point ?
Is that the expected result ?

TASK

Application problem (see PDF)

4- Consider the parallelepiped defined by the vertices:

V_1 (0, 0, 1)	V_2 (1, 0, 0)	V_3 (2, 0, 1)	V_4 (1, 0, 2)
V_5 (0, 1, 1)	V_6 (1, 1, 0)	V_7 (2, 1, 1)	V_8 (1, 1, 2)

We want to represent it using a *Perspective Projection*: the projection plane is the plane $z = 0$ and the center of projection is point (0, 0, 4).

a) Using *Homogeneous Coordinates*, determine the matrix that represents the corresponding projection transformation. Explain the steps carried out.

b) Compute the coordinates of the projected vertices.

c) Draw the projected parallelepiped. Identify the projected vertices and the visible edges.

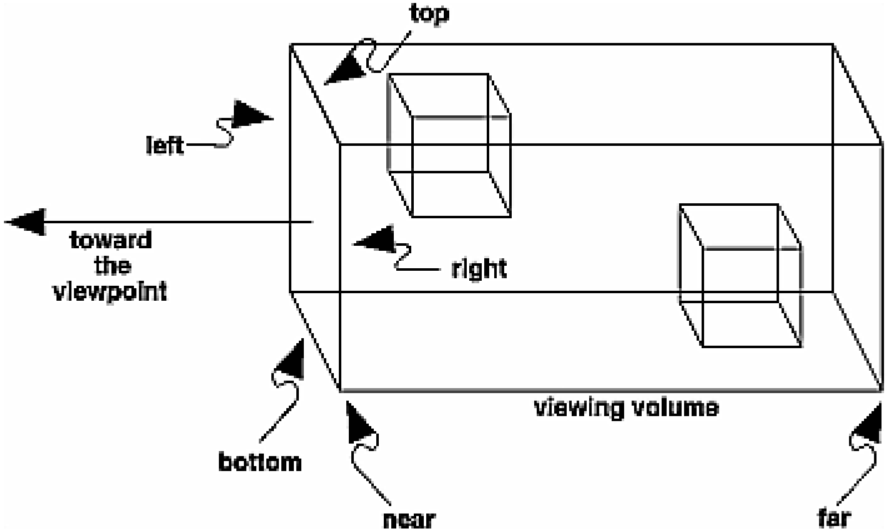
d) Given the obtained projection, classify it. Justify your answer.

PROJECTIONS IN OPENGL / WEBGL

OpenGL (Pre-3.1) – Orthogonal Parallel Projection

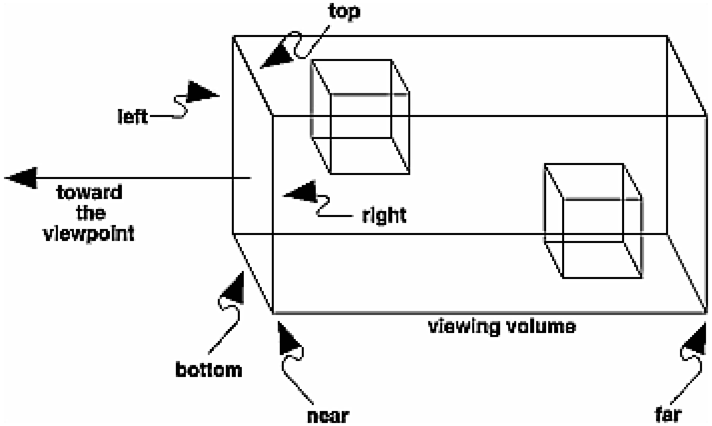
- The direction of projection is defined by vector (0, 0, -1) and is parallel to the ZZ’ axis
- The projection plane is XOY (z = 0)
- The view volume (i.e., the faces of the parallelepiped) is defined by

```
glOrtho( left, right, bottom, top, near, far );
```



[OpenGL - The Red Book]

OpenGL (Pre-3.1) – Orthogonal Parallel Projection



[OpenGL - The Red Book]

- Signed distances relative to (0, 0, 0) :

  right > left, top > bottom, and far > near (!!)

- The clipping planes z = -near and z = -far might have different signs.
- Lower-left corner of the window defined on the front clipping plane:

  (left, bottom, -near)

- Lower-left corner of the window defined on the front clipping plane:

  (right, top, -near)

OpenGL (Pre-3.1) – Example

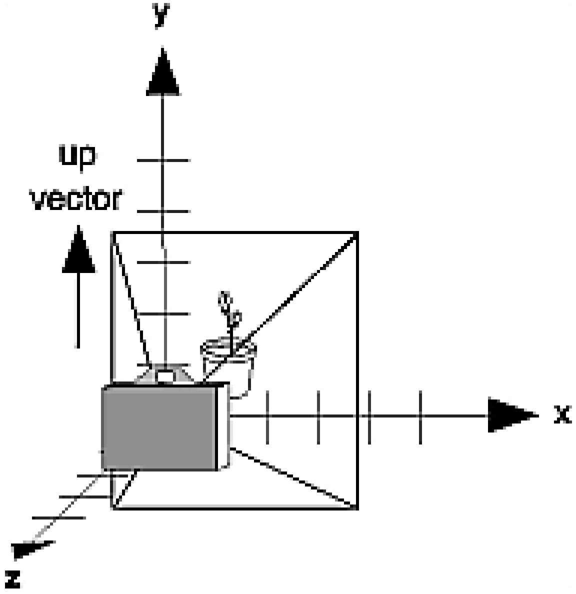
- Cubic view volume with edge length 2

```
glMatrixMode( GL_PROJECTION );
glLoadIdentity( );
glOrtho(-1.0, 1.0, -1.0, 1.0, -1.0, 1.0);
```

- Default values ?

OpenGL (Pre-3.1) – Perspective Projection

- Viewer is at (0, 0, 0)
- Looking at the negative ZZ' semi-axis

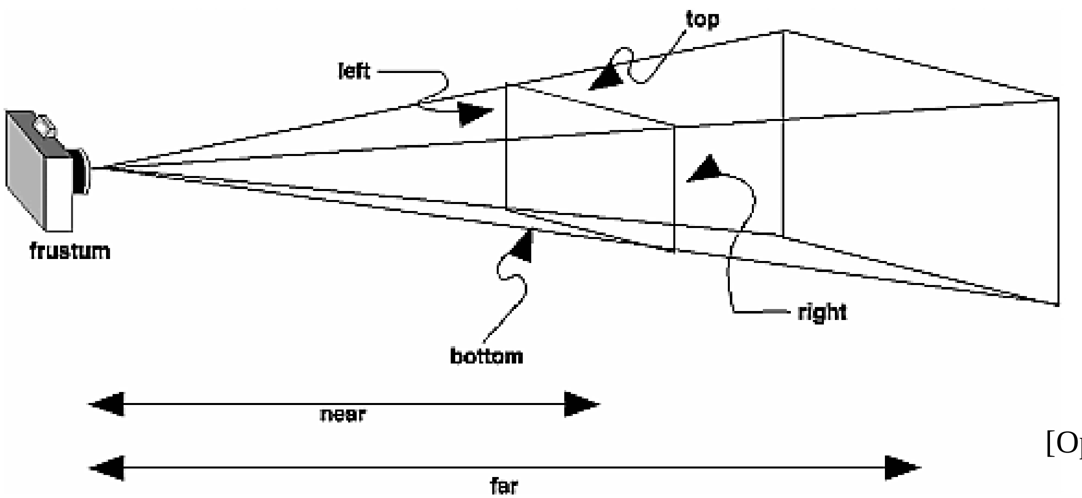


[OpenGL - The Red Book]

OpenGL (Pre-3.1) – Perspective Projection

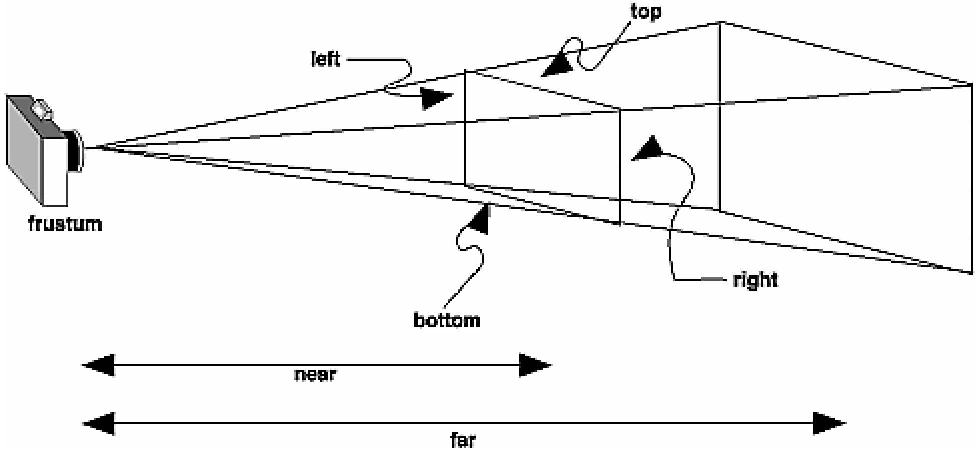
- Viewer is at (0, 0, 0)
- Looking at the negative ZZ' semi-axis
- View volume (i.e., the faces of a truncated pyramid) is defined by

```
glFrustum( left, right, bottom, top, near, far );
```



[OpenGL - The Red Book]

OpenGL (Pre-3.1) – Perspective Projection



[OpenGL - The Red Book]

- The clipping plane z = - near (front) and z = - far (back) satisfy

far > near > 0

- Lower-left corner of the window defined on the front clipping plane

(left, bottom, -near)

- Upper-right corner of the window defined on the front clipping plane

(right, top, -near)

OpenGL (Pre-3.1) – Perspective Projection

- Defining a view volume

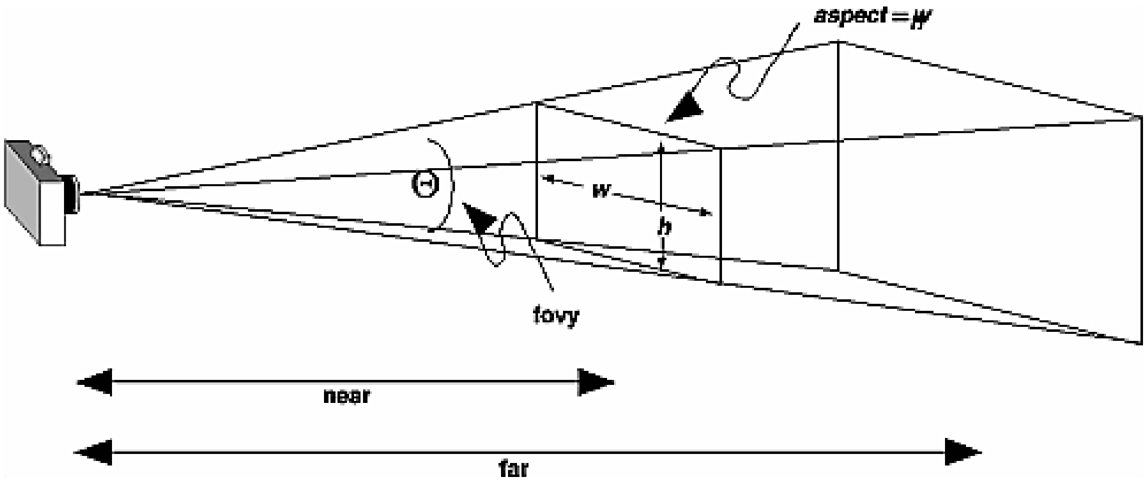
```
glMatrixMode( GL_PROJECTION );
glLoadIdentity( );
glFrustum( -1.0, 1.0, -1.0, 1.0, 1.0, 5.0 );
```

- Default values ?

- The viewer (i.e., the projection center) cannot be located inside the view volume.

OpenGL (Pre-3.1) – Auxiliary Functions

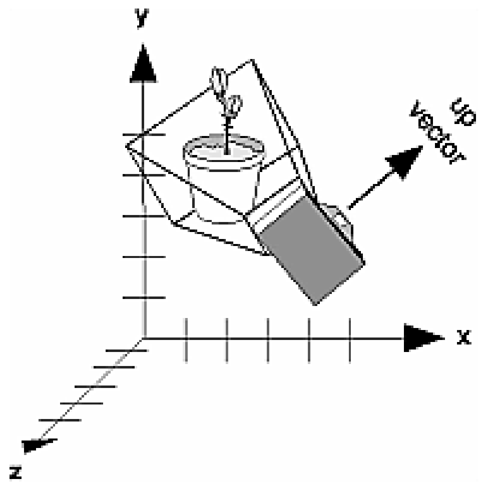
[OpenGL - The Red Book]




```
gluPerspective( fov, aspect, near, far );
```

- Not easy to use…

OpenGL (Pre-3.1) – Auxiliary Functions



[OpenGL - The Red Book]

```
gluLookAt( eyex, eyey,eyez,
           atx, aty, atz,
           upx, upy, upz);
```

- Not easy to use…

OpenGL / WebGL

- The auxiliary functions of the previous versions no longer exist !!

- Need to:

  – Position the viewer

    - Model-View Matrix

  – Select the projection type

    - Projection Matrix

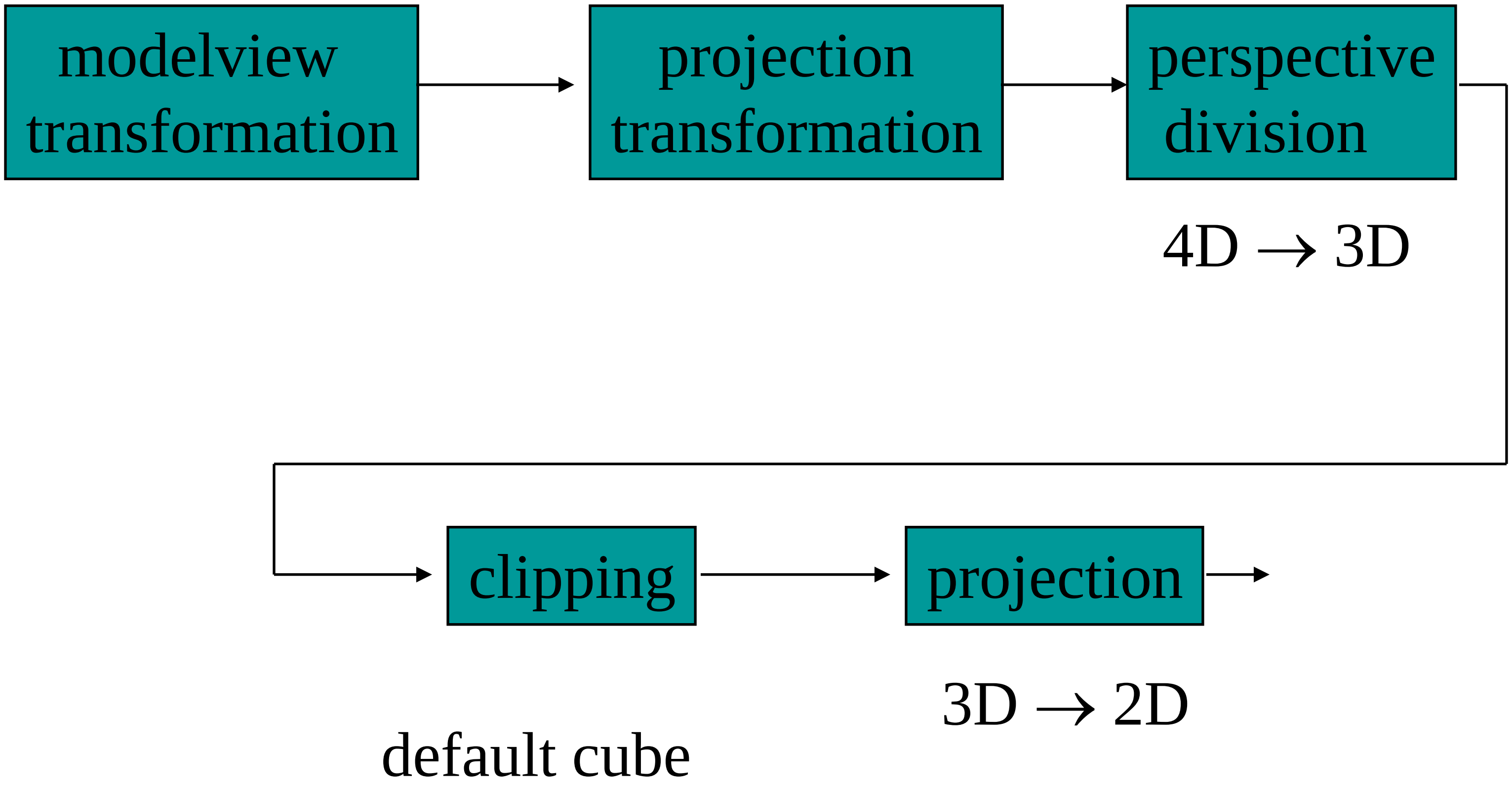
  – Set the view volume according to the chosen projection

    - View-Volume

- Auxiliary function !!

- What is set by default ?

OpenGL / WebGL

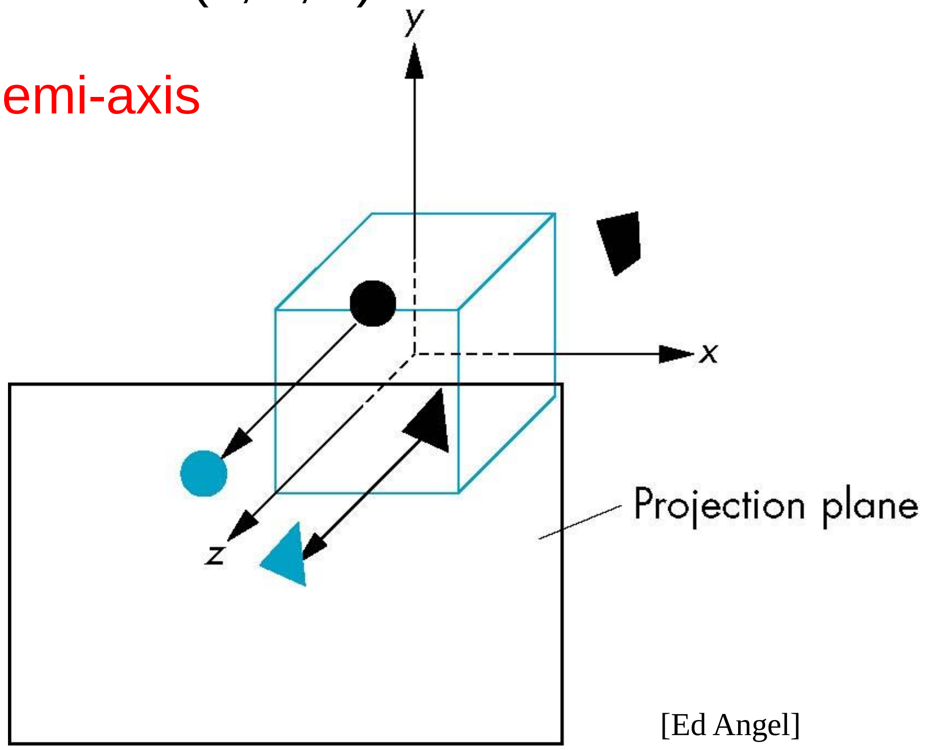


[Ed Angel]

OpenGL / WebGL – Default

- Orthogonal, Parallel Projection / Orthographic Projection
  - Viewer at an indefinite distance from (0, 0, 0)
  - Looking at the negative ZZ' semi-axis

- View Volume
  - Cube centered at (0, 0, 0)
  - Edge length 2

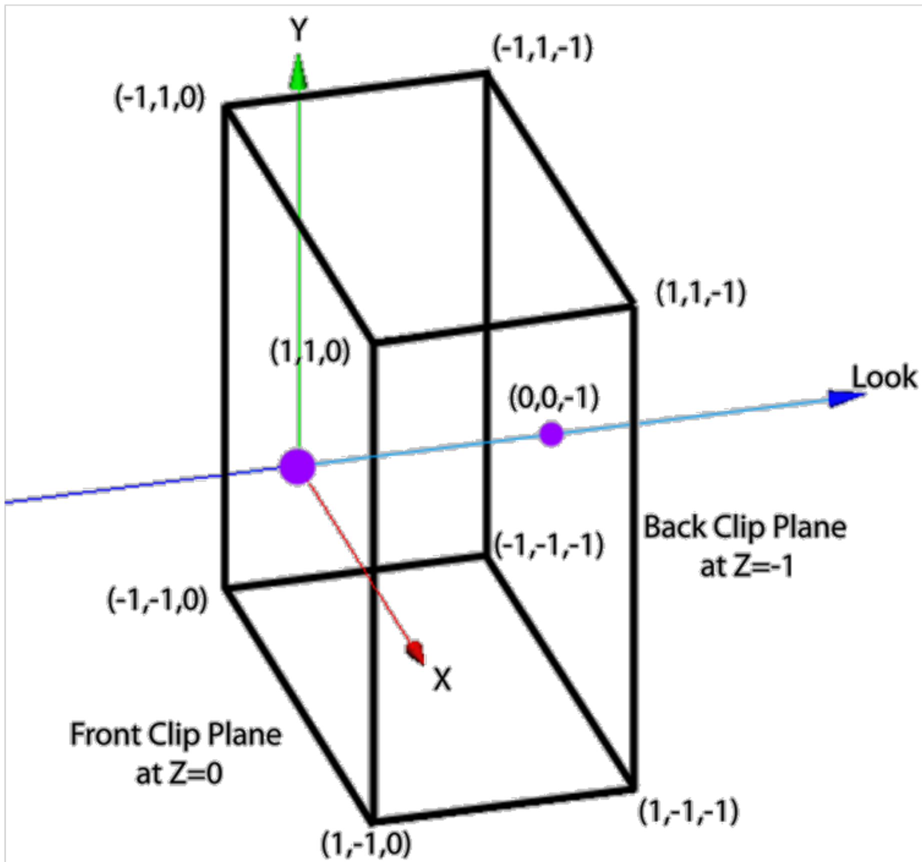


[Ed Angel]

OpenGL / WebGL

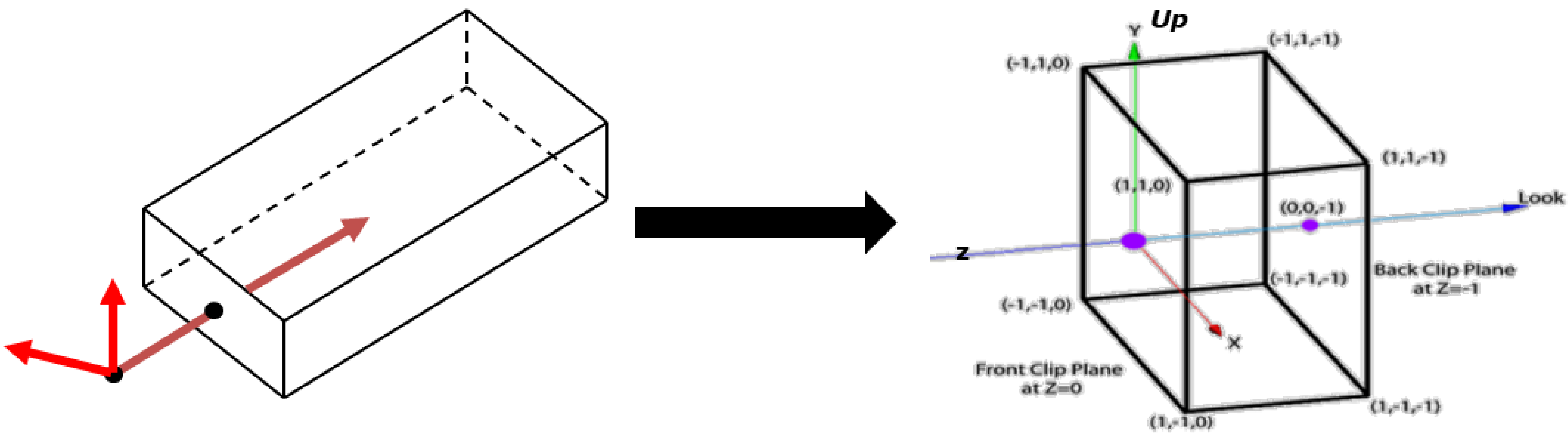
- How to visualize models placed outside the view volume ?
  - Apply translation transformations
- What if we want to look at a side-face of a model ?
  - Apply rotation transformations
- The Model-View Matrix will be changed !!
  - Matrix multiplication order
  - Auxiliary functions to set and multiply transformation matrices

Another Canonical Parallel View Volume



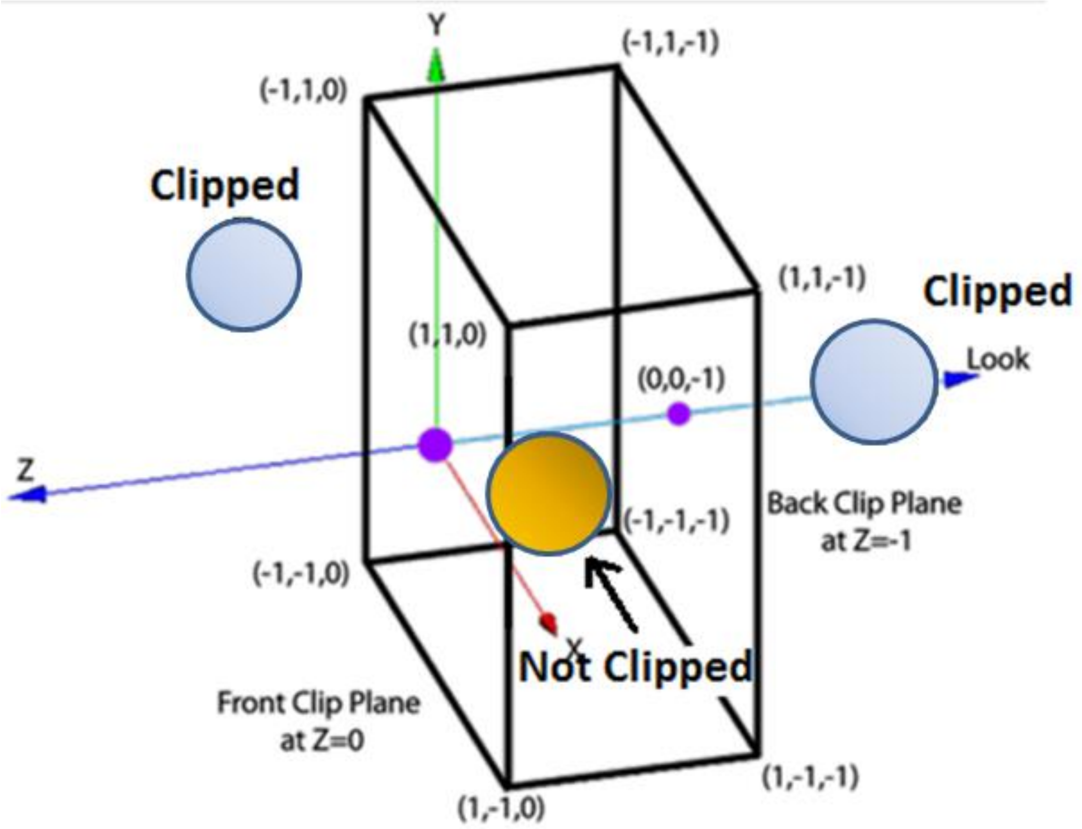
[van Dam]

The Normalizing Transformation



[van Dam]

Clipping against the View Volume

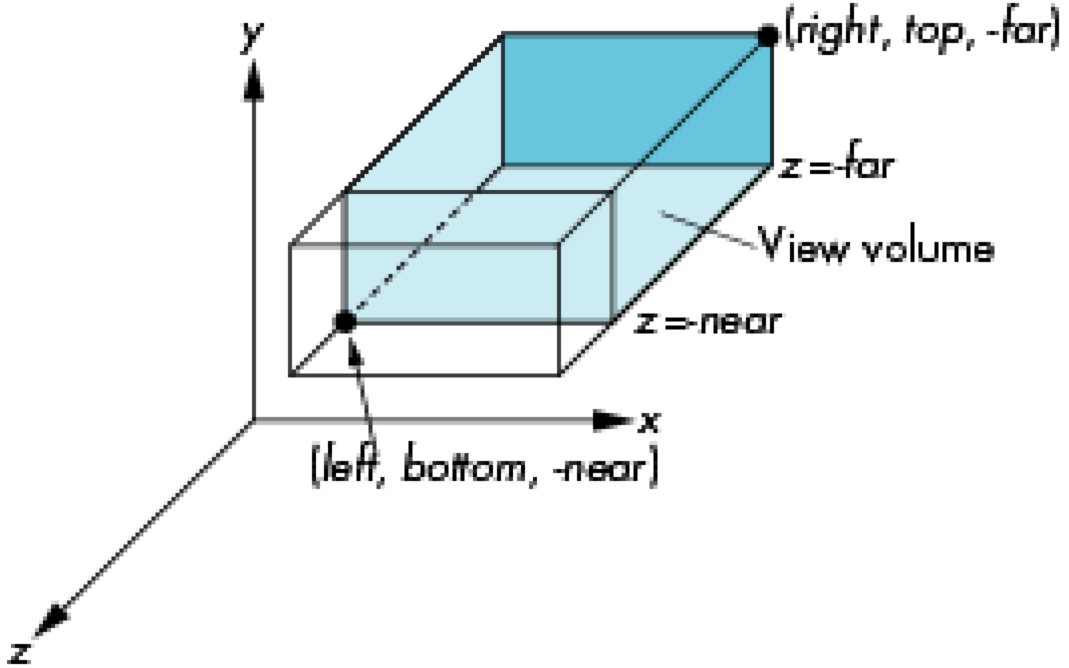


[van Dam]

OpenGL / WebGL – Orthogonal Parallel Projection

- View volume for the orthographic projection

`ortho(left,right,bottom,top,near,far)`

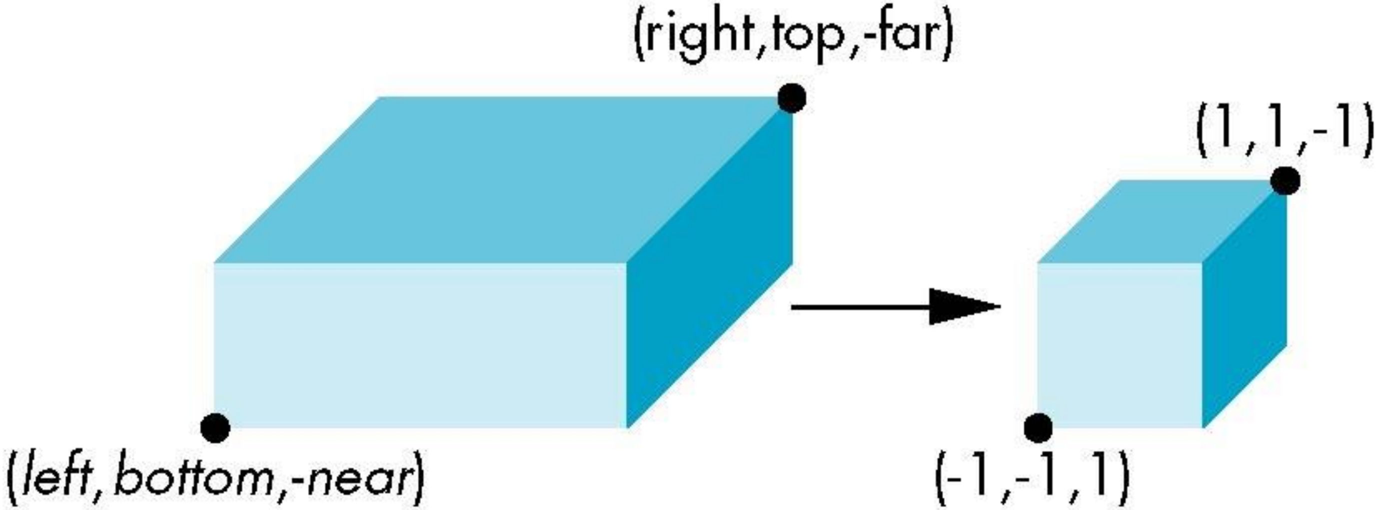


[Ed Angel]

OpenGL / WebGL – Orthogonal Parallel Projection

- View volume for the orthographic projection

```
ortho(left,right,bottom,top,near,far)
```



[Ed Angel]

OpenGL / WebGL – Orthogonal Parallel Projection

- 2 steps:

  T(-(left+right)/2, -(bottom+top)/2,(near+far)/2))

  S(2/(left-right),2/(top-bottom),2/(near-far))

$$\mathbf{P} = \mathbf{ST} = \begin{bmatrix} \frac{2}{right-left} & 0 & 0 & -\frac{right-left}{right-left} \\ 0 & \frac{2}{top-bottom} & 0 & -\frac{top+bottom}{top-bottom} \\ 0 & 0 & \frac{2}{near-far} & \frac{far+near}{far-near} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Projection matrix:

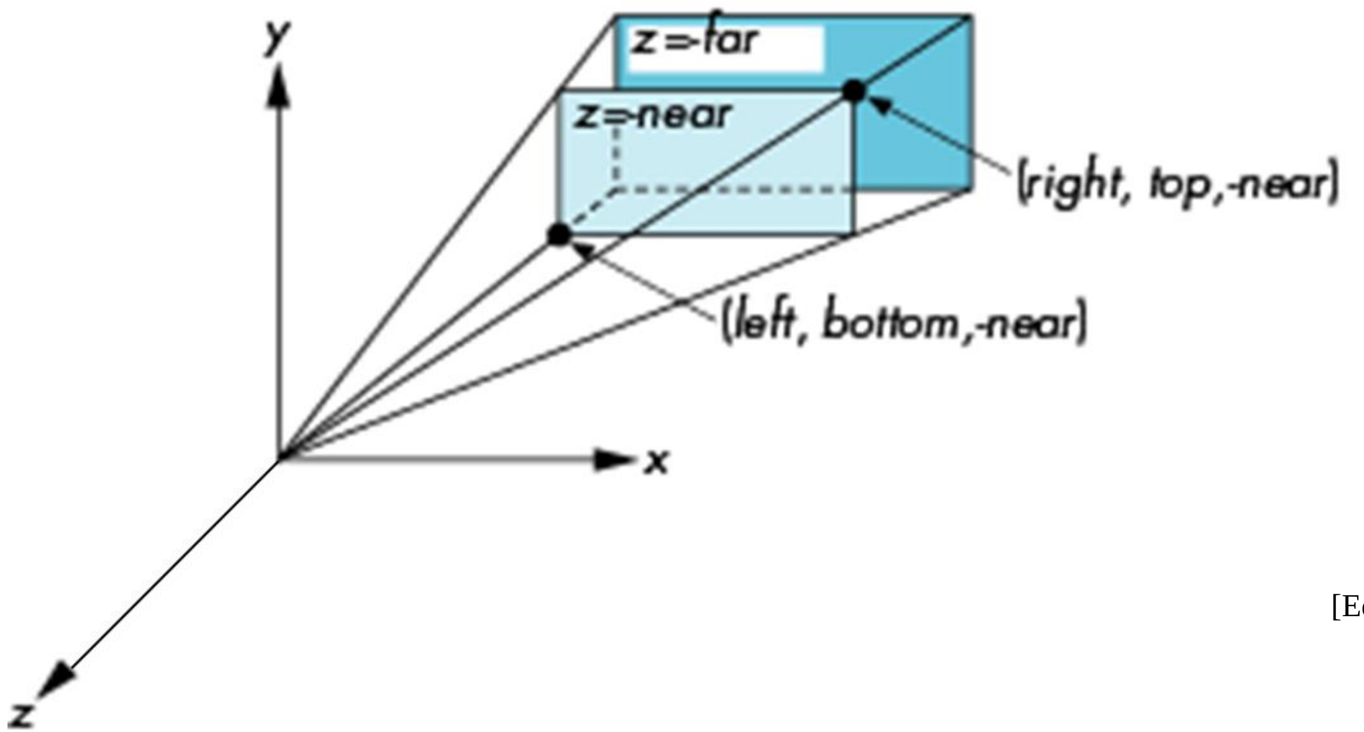
$$\mathbf{P} = \mathbf{M}_{\mathrm{orth}}\mathbf{ST}$$

[Ed Angel]

OpenGL / WebGL – Perspective Projection

- Viewer at (0, 0, 0)
- Looking at the negative ZZ' semi-axis

```
frustum(left,right,bottom,top,near,far);
```

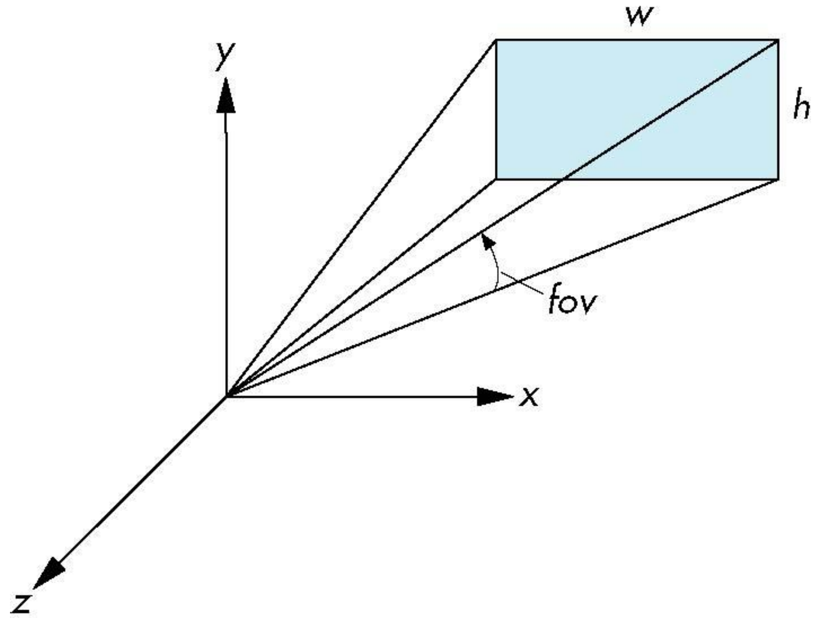


[Ed Angel]

OpenGL / WebGL – Perspective Projection

- Viewer at (0, 0, 0)
- Looking at the negative ZZ' semi-axis

```
perspective(fovy,aspect,near,far);
```

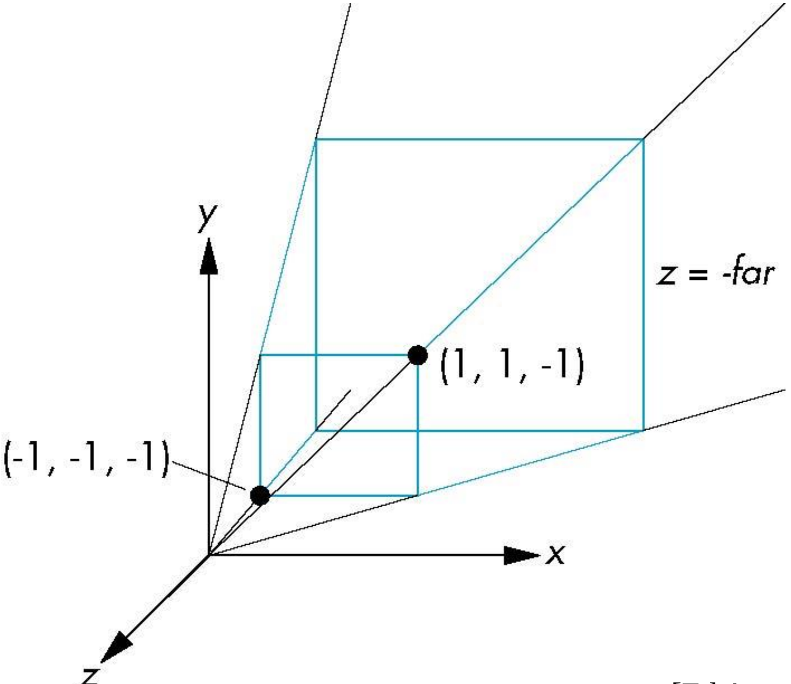


[Ed Angel]

OpenGL / WebGL – Perspective Projection

- Viewer at (0, 0, 0)
- Clipping planes at z = -1 and z = -far
- FOV = 90º
  - x = ± z and y = ± z

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

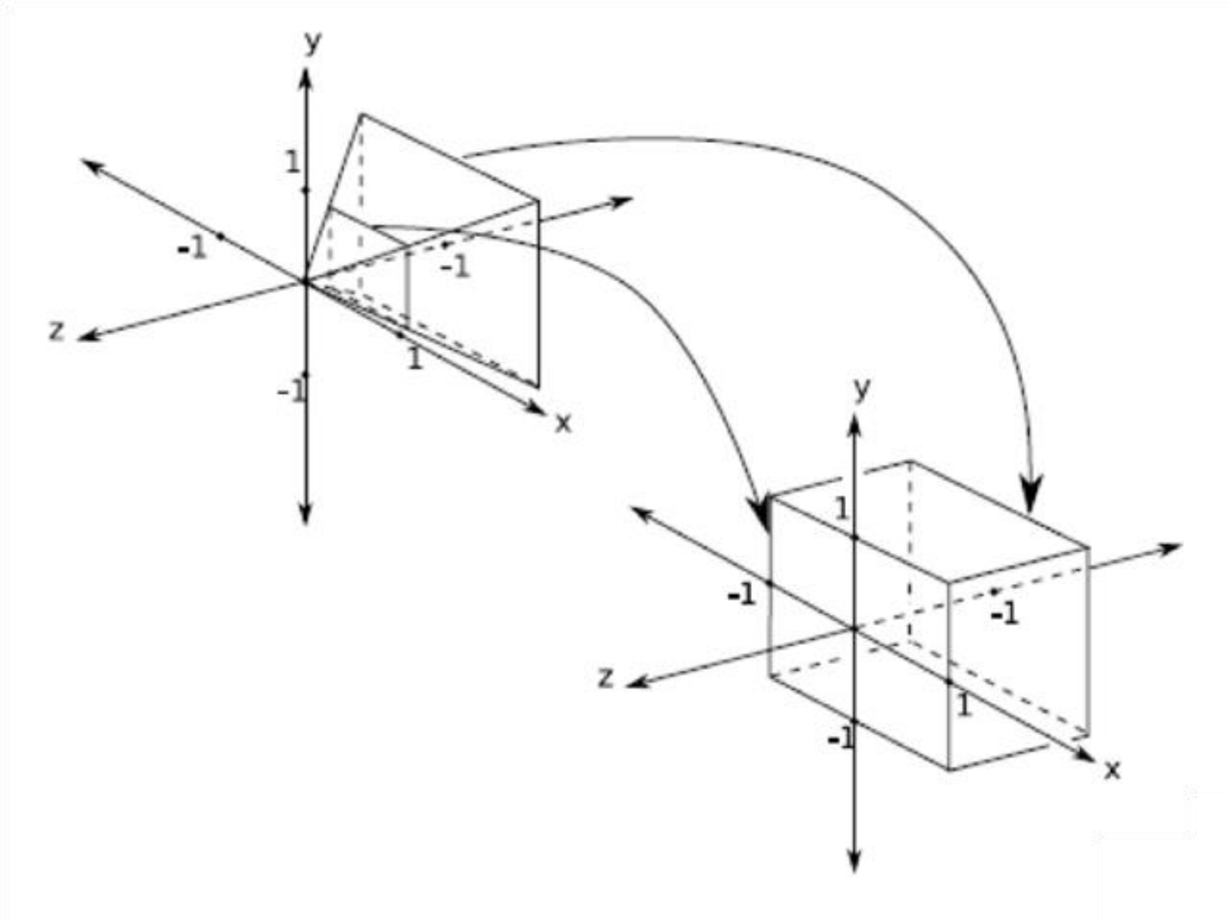


[Ed Angel]

OpenGL / WebGL

- What if we want a perspective projection ?
- Convert into a orthogonal, parallel projection !!
  - Apply the required transformation to all the models in the scene
  - And to the perspective view volume
- Just carry out matrix products and get the global transformation matrix
  - CPU or GPU

Perspective to Parallel Transformation



[van Dam]

OpenGL / WebGL – Generalization

$$\mathbf{N} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \alpha & \beta \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

[Ed Angel]

In Euclidean coordinates, the point $(x, y, z, 1)$ corresponds to

$$x'' = x/\text{-}z$$
$$y'' = y/\text{-}z$$
$$z'' = -(\alpha+\beta/z)$$

whose orthogonal projection is $(x'', y'', 0)$, as wanted

OpenGL / WebGL – Generalization

Selecting

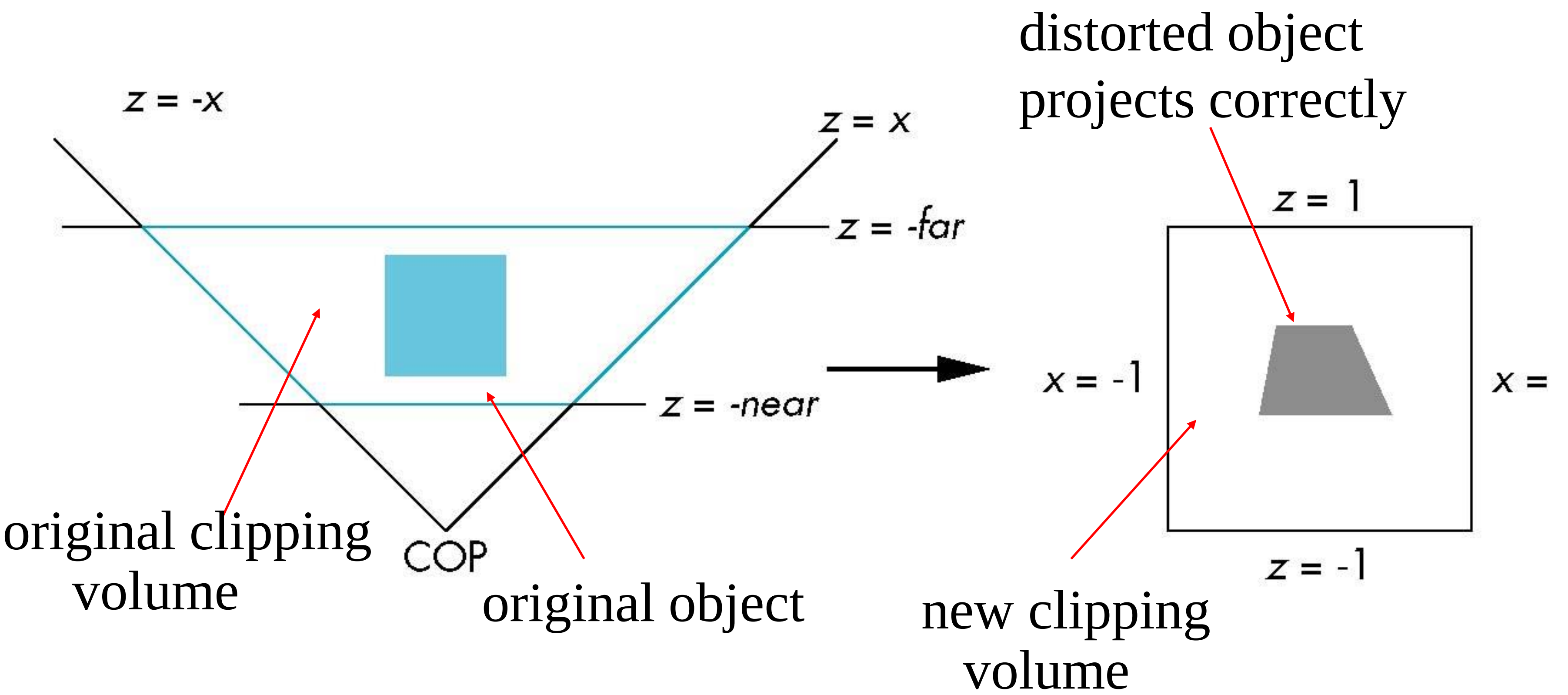
$$\alpha = \frac{\text{near} + \text{far}}{\text{far} - \text{near}}$$

$$\beta = \frac{2\text{near} * \text{far}}{\text{near} - \text{far}}$$

the near plane is mapped onto $z = -1$
the far plane is mapped onto $z = 1$
and the side faces are mapped onto $x = \pm 1$, $y = \pm 1$

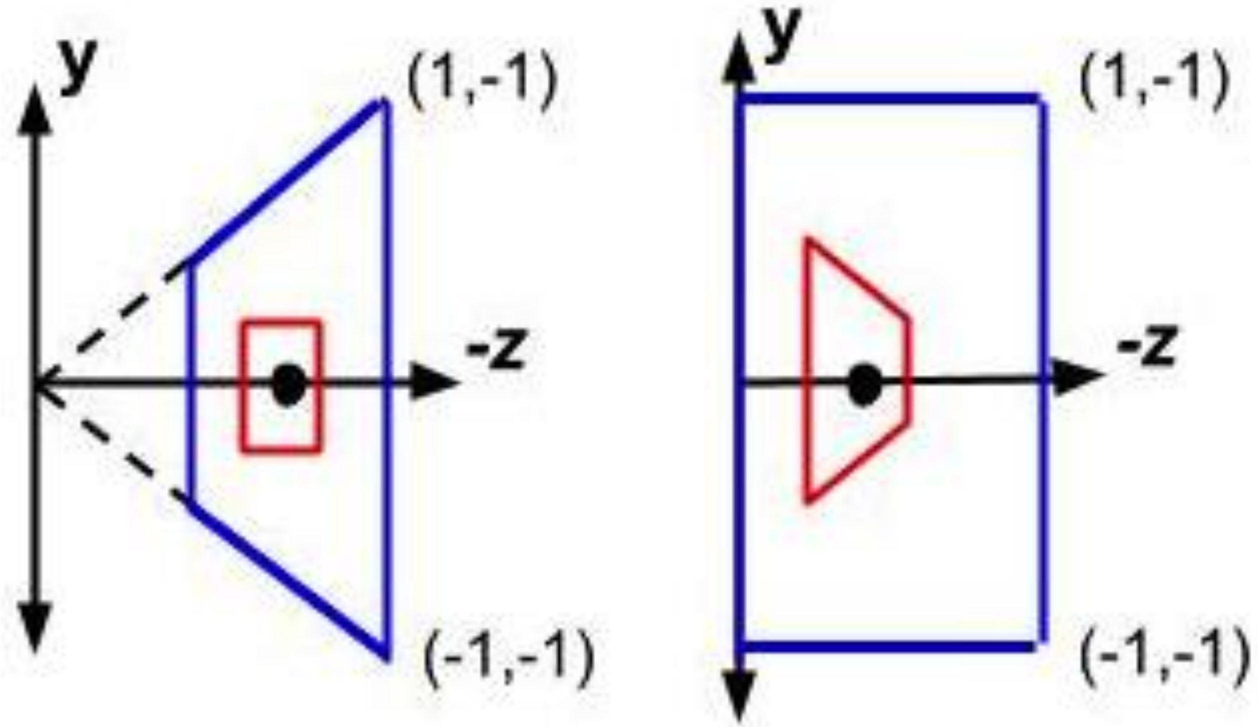
We get the default view volume !!

OpenGL / WebGL – Generalization



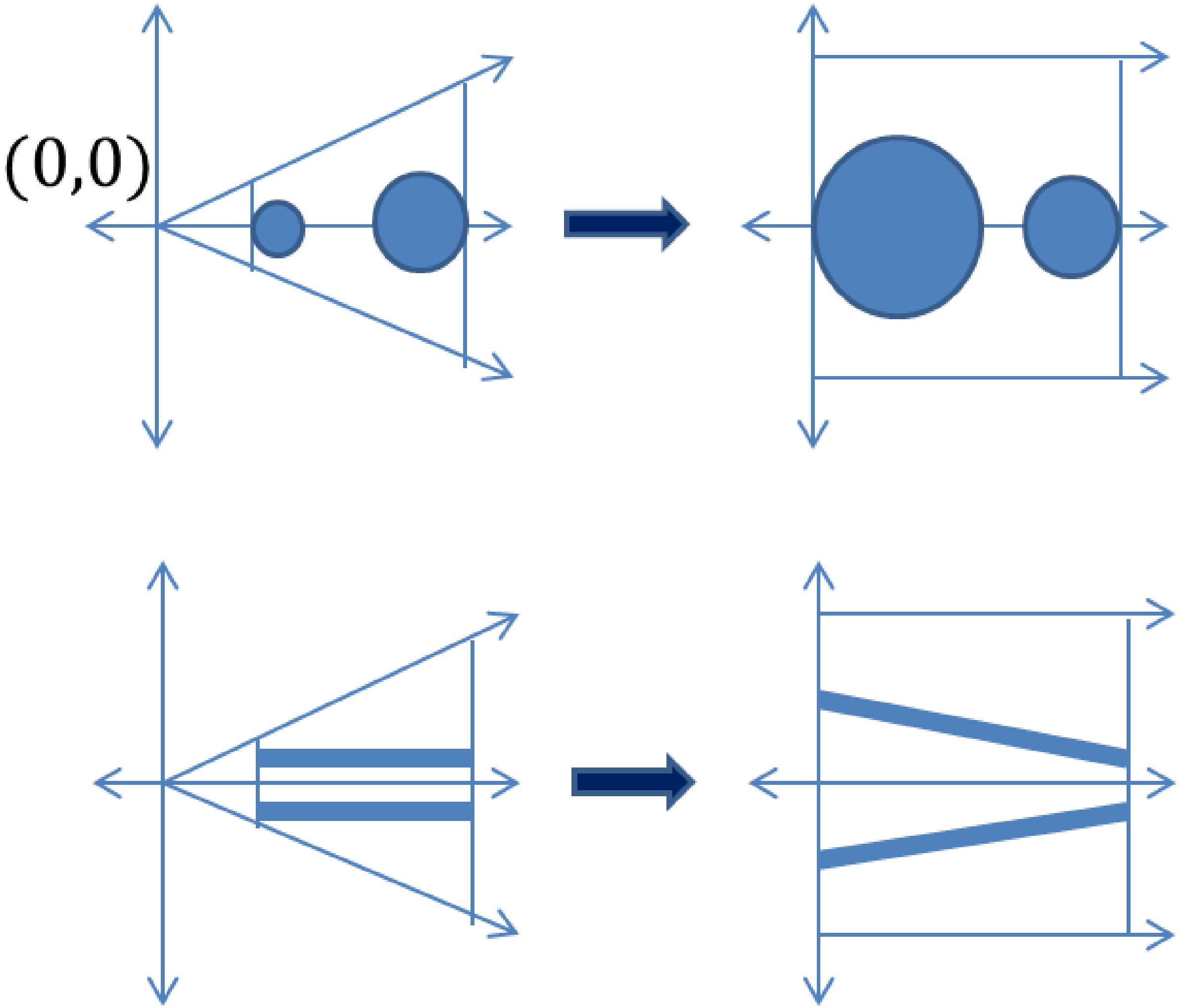
[Ed Angel]

Example – Deforming the view volume



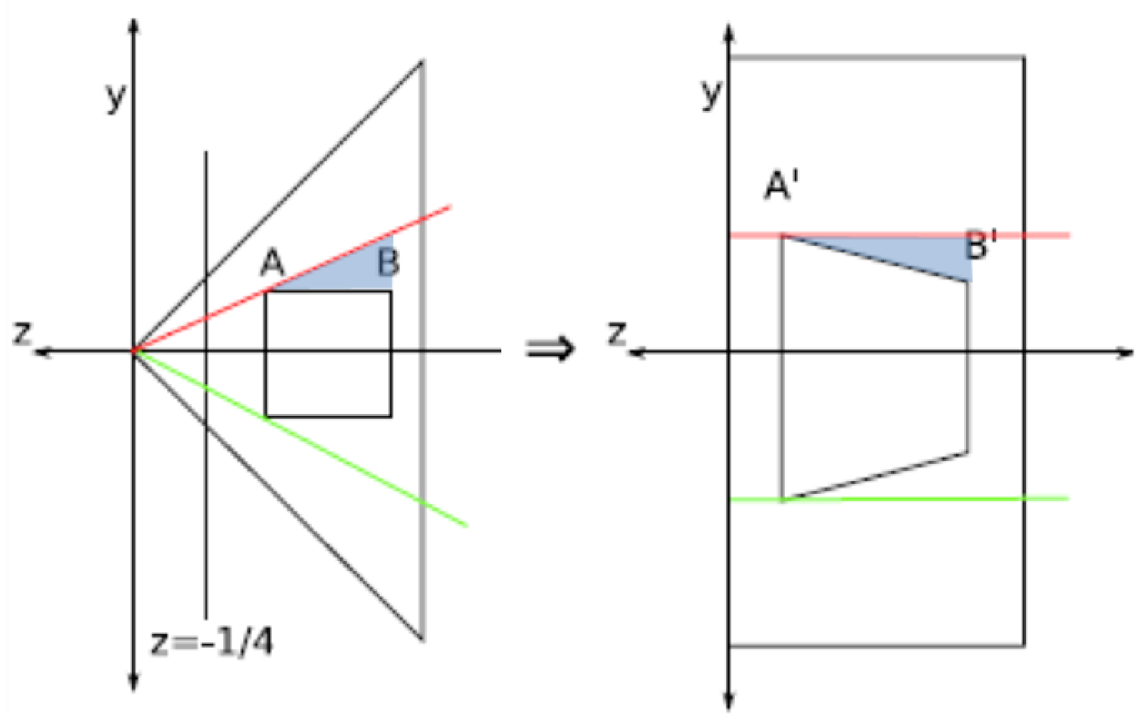
[Andy van Dam]

Examples – Deforming the view volume



[Andy van Dam]

Example – Deforming the view volume



[Andy van Dam]

REFERENCES

References

- D. Hearn and M. P. Baker, *Computer Graphics with OpenGL*, 3rd Ed., Addison-Wesley, 2004
- E. Angel and D. Shreiner, *Introduction to Computer Graphics,* 6th Ed., Pearson Education, 2012
- J. Foley et al., *Introduction to Computer Graphics*, Addison-Wesley, 1993
- D. Rogers and J. Adams, *Mathematical Elements for Computer Graphics*, 2nd Ed., McGraw-Hill, 1990