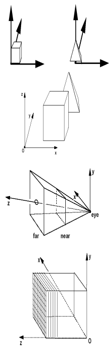
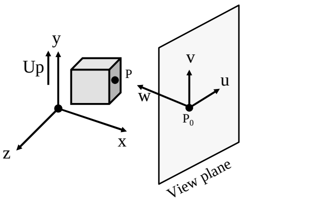
Common Coordinate Systems

- Object space
  - local to each object
- World space
  - common to all objects
- Eye space / Camera space
  - derived from view frustum
- Screen space
  - indexed according to hardware attributes



Viewing in 3D

(Chapt. 6 in FVD, Chapt. 12 in Hearn & Baker)

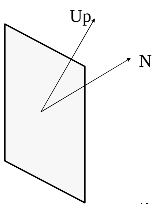


- $P_0=(x_0,y_0,z_0)$ is a point where a camera is located.
- P is a point to look-at.
- $N=(P_0-P)/|P_0-P|$ is the view-plane normal vector.
- Up is the view up vector, whose projection onto the view-plane is directed up.

Specifying the Viewing Coordinates

- *Viewing Coordinates system*, [u, v, w], describes 3D objects with respect to a viewer.
- A *viewing plane* (*projection plane*) is set up perpendicular to w and aligned with (u,v).
- To set a view plane we have to specify a *view-plane normal vector*, N, and a *view-up vector*, Up, (both, in world coordinates):

How to form Viewing coordinate system



$$w = N / \| N \|$$

First, normalize the look-at vector to form the w-axis

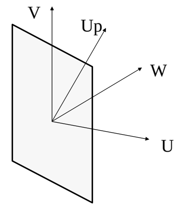
- How to form Viewing coordinate system :

$$w=\frac{N}{|N|} \quad ; \quad u=\frac{Up\times N}{|Up\times N|} \quad ; \quad v=w\times u$$

- The transformation, M, from world-coordinate into viewing-coordinates is:

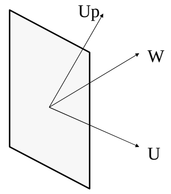
$$M=\begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ w_x & w_y & w_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Create V perpendicular to U and W



$$V = W \times U$$

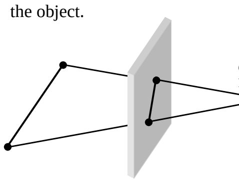
Create U perpendicular to Up and W



$$U = \frac{Up \times W}{|Up \times W|}$$

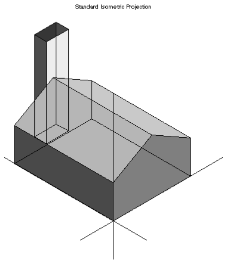
Projections

- Viewing 3D objects on a 2D display requires a mapping from 3D to 2D.
- A projection is formed by the intersection of certain lines (*projectors*) with the view plane.
- Projectors are lines from the *center of projection* through each point in the object.

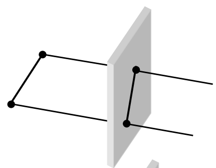


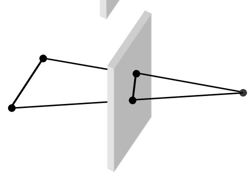
Parallel Projection

A parallel projection preserves relative proportions of objects, but does not give realistic appearance (commonly used in engineering drawing).



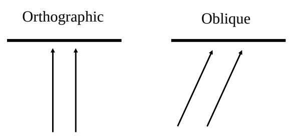
- Center of projection at infinity results with a parallel projection.
- A finite center of projection results with a perspective projection.





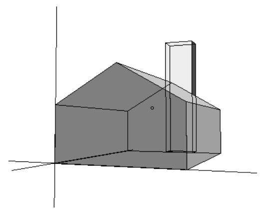
Parallel Projection

- Projectors are all parallel.
- Orthographic: Projectors are perpendicular to the projection plane.
- Oblique: Projectors are not necessarily perpendicular to the projection plane.

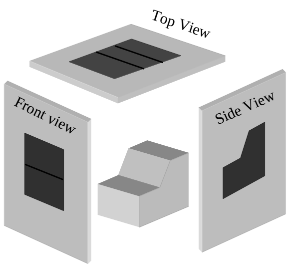


Perspective Projection

A perspective projection produces realistic appearance, but does not preserve relative proportions.



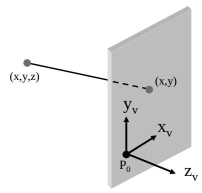
- Lengths and angles of faces parallel to the viewing planes are preserved.
- **Problem**: 3D nature of projected objects is difficult to deduce.



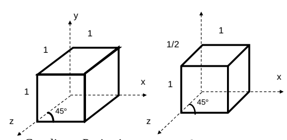
Orthographic Projection

- Since the viewing plane is aligned with (x_v,y_v), orthographic projection is performed by:

$$\begin{bmatrix} x_p \\ y_p \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} x_v \\ y_v \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_v \\ y_v \\ z_v \\ 1 \end{bmatrix}$$



- Cavalinear projection :
  - Preserves lengths of lines perpendicular to the viewing plane.
  - 3D nature can be captured but shape seems distorted.
- Cabinet projection:
  - lines perpendicular to the viewing plane project at 1/2 of their length.
  - A more realistic view than the Cavalinear projection.

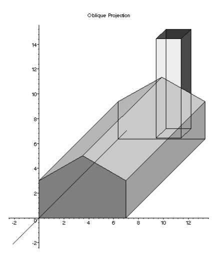


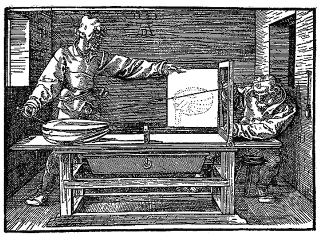
Cavalinear Projection

Cabinet Projection

Oblique Projection

- Projectors are ***not*** perpendicular to the viewing plane.
- Angles and lengths are preserved for faces parallel to the plane of projection.
- Somewhat preserves 3D nature of an object.

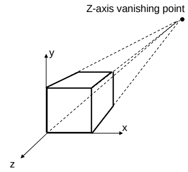


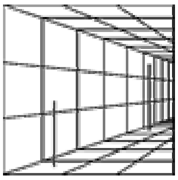


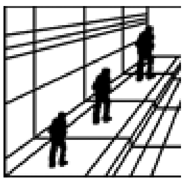
Perspective Projection

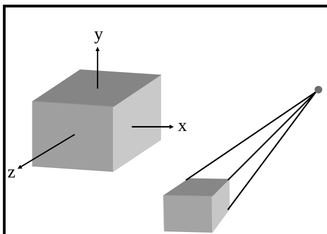
- In a perspective projection, the center of projection is at a finite distance from the viewing plane.
- Parallel lines that are not parallel to the viewing plane, converge to a *vanishing poin*t.

  ⇨ A vanishing point is the projection of a point at infinity.

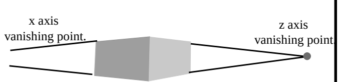








One point (z axis)
perspective projection



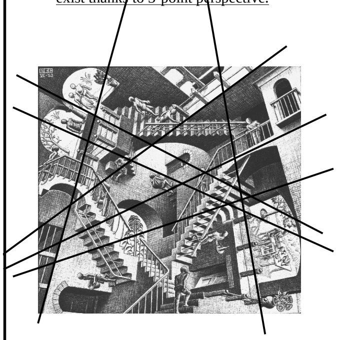
Two points
perspective projection

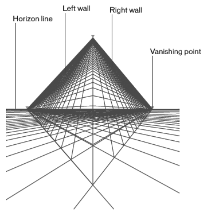
Vanishing Points

- There are infinitely many general vanishing points.
- There can be up to three *axis vanishing points* (principal vanishing points).
- Perspective projections are categorized by the number of principal vanishing points, equal to the number of principal axes intersected by the viewing plane.
- Most commonly used: one-point and two-points perspective.

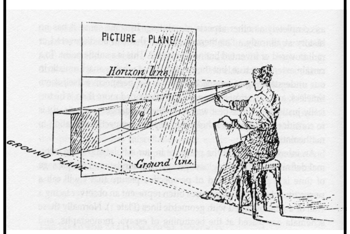
3-point Perspective

M.C.Escher's "*Relativity*" where 3 worlds co-exist thanks to 3-point perspective.

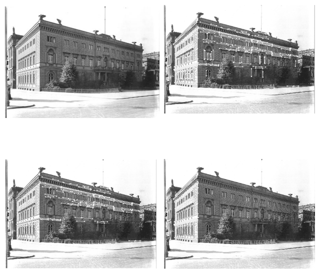


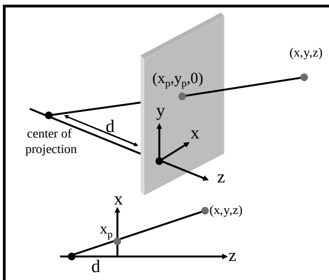


Perspective Projection



3-point Perspective



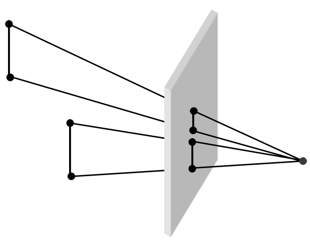


- Using similar triangles it follows:

$$\frac{x_p}{d} = \frac{x}{z+d} \quad ; \quad \frac{y_p}{d} = \frac{y}{z+d}$$

$$\Rightarrow \quad x_p = \frac{d \cdot x}{z+d} \quad ; \quad y_p = \frac{d \cdot y}{z+d} \quad ; \quad z_p = 0$$

A perspective projection produces realistic appearance, but does not preserve relative proportions.



Observations

- M_{per} is singular ($|M_{per}|=0$), thus M_{per} is a many to one mapping
- Points on the viewing plane (z=0) do not change.
- The vanishing point of parallel lines directed to (U_x,U_y,U_z) is at $[dU_x/U_z, dU_y/U_z]$.
- When $d \rightarrow \infty$, $M_{per} \rightarrow M_{ort}$

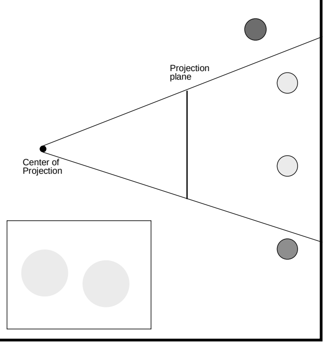
- Thus, a perspective projection matrix is defined:

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

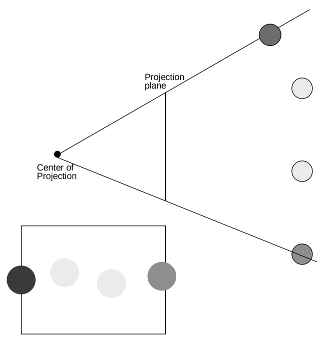
$$M_{per} P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{d} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 0 \\ \frac{z+d}{d} \end{bmatrix}$$

$$\Rightarrow \quad x_p = \frac{d \cdot x}{z+d} \quad ; \quad y_p = \frac{d \cdot y}{z+d} \quad ; \quad z_p = 0$$

Zoom-in



Zoom-in

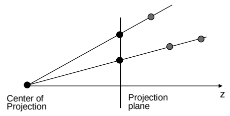


Summary

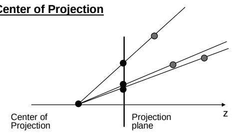
Planar geometric projections
- Parallel
  - Oblique
    - Cavalinear
    - Cabinet
    - Other
  - Orthographic
    - Top
    - Side
    - Front
    - Other
- Perspective
  - One point
  - Two point
  - Three point

What is the difference between moving the center of projection and moving the projection plane?

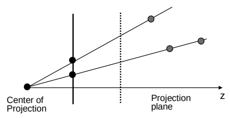
Original



Moving the Center of Projection

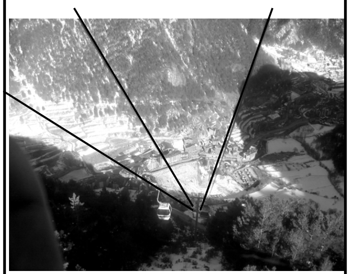


Moving the Projection Plane

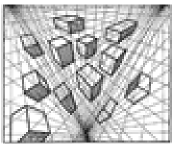


Another view in Perspective

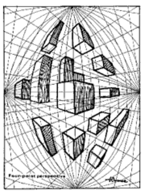
Another view in Perspective



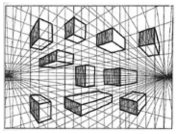
Three-point perspective



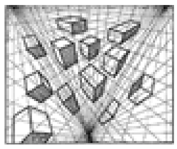
Four-point perspective



Two-point perspective

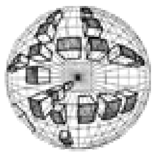


Three-point perspective

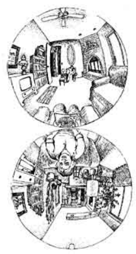


Fisheye views of the Hagia Sophia
(Istanbul)
(also known as Aya Sofya)

Five-point perspective ?



Six-point perspective ?



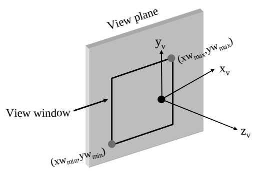
Vertical lines



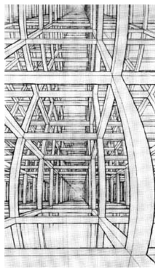

Fisheye view

View Window

- After objects were projected onto the viewing plane, an image is taken from a *View Window*.
- A view window can be placed *anywhere* on the view plane.
- In general the view window is aligned with the viewing coordinates and is defined by its extreme points: (xw_{min},yw_{min}) and (xw_{max},yw_{max})

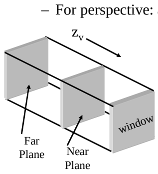


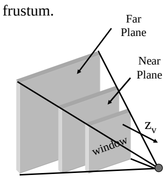
House of Stairs

M.C. Escher's House of Stairs

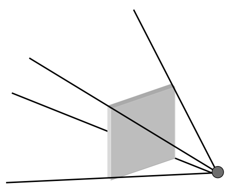
- In order to limit the infinite view volume we define two additional planes: *Near Plane* and *Far Plane*.
- Only objects in the bounded view volume can appear.
- The near and far planes are parallel to the view plane and specified by z_{near} and z_{far}.
- A limited view volume is defined:
  - For orthographic: a rectangular parallelpiped.
  - For oblique: an oblique parallelpiped.
  - For perspective: a frustum.



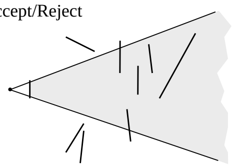


View Volume

- Given the specification of the *view window*, we can set up a *View Volume*.
- Only objects inside the view volume might appear in the display, the rest are clipped.



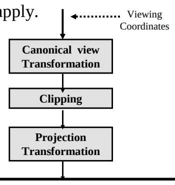
Trivial Accept/Reject



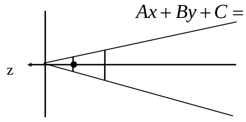
Red polygons are rejected
Black are accepted
And
Blue are tagged for clipping

Canonical View Volumes

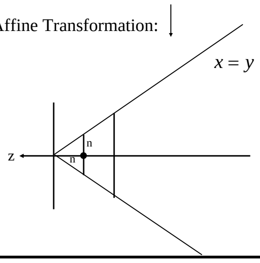
- In order to determine the objects that are seen in the view window we have to clip objects against six planes forming the view volume.
- Clipping against arbitrary 3D plane requires considerable computations.
- For fast clipping we transform the general view volume to a *canonical view volume* against which clipping is easy to apply.

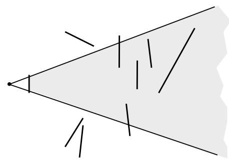


$Ax + By + C = 0$



Affine Transformation:





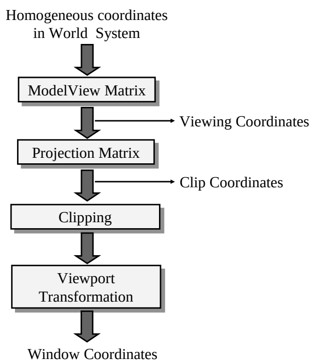
Classify the vertices of the polygons against each side Si of the frustum in turn:

If all the vertices are on the outside of some Si, then cull the polygon, otherwise,

Polygons with all its vertices inside all Sj are accepted.

All others are tagged.

OpenGL Transformation Pipe-Line



Perspective transformation:

$$\begin{bmatrix} x''_c \\ y''_c \\ z''_c \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -2\frac{fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x'_c \\ y'_c \\ z'_c \\ 1 \end{bmatrix}$$

$$\frac{x''}{1} = \frac{x'}{-z'} \Rightarrow x'' = \frac{x'}{-z'} \qquad \frac{y''}{1} = \frac{y'}{-z'} \Rightarrow y'' = \frac{y'}{-z'}$$

$$[0 \quad 0 \quad -n \quad 1] \rightarrow [0 \quad 0 \quad -1 \quad 1]$$

$$[0 \quad 0 \quad -f \quad 1] \rightarrow [0 \quad 0 \quad 1 \quad 1]$$