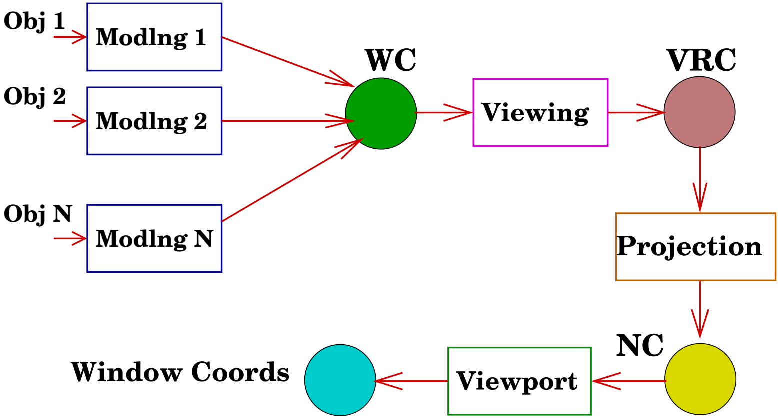
CSE251
Basics of Computer Graphics
Module: Graphics Pipeline

Avinash Sharma

Spring 2018

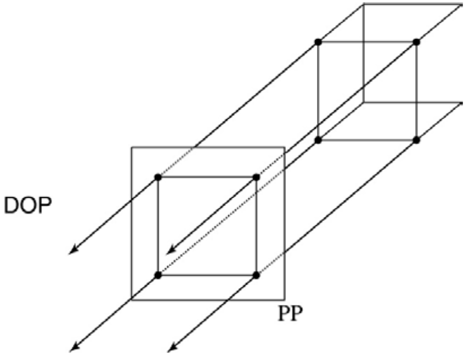
3D Graphics: Block Diagram



Projections

- Projection involves *projectors* starting from 3D points and hitting the 2D *projection plane*, forming the *image* of the point.
- Two types of projections.
- **Parallel projection**: Projectors are parallel to each other, all have the same *direction of projection* (**DOP**).

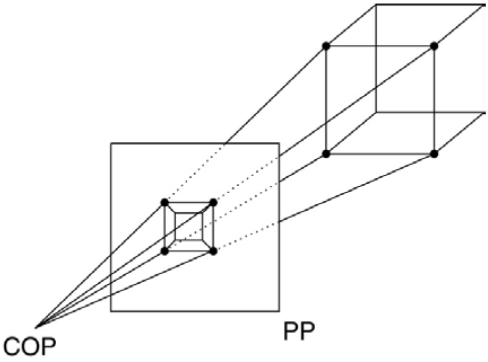
Projections (cont.)



Projections

- Projection involves *projectors* starting from 3D points and hitting the 2D *projection plane*, forming the *image* of the point.
- Two types of projections.
- **Parallel projection**: Projectors are parallel to each other, all have the same *direction of projection* (**DOP**).
- **Perspective projection**: All projects pass through a point in space called the *centre of projection* (**COP**).

Projections (cont.)



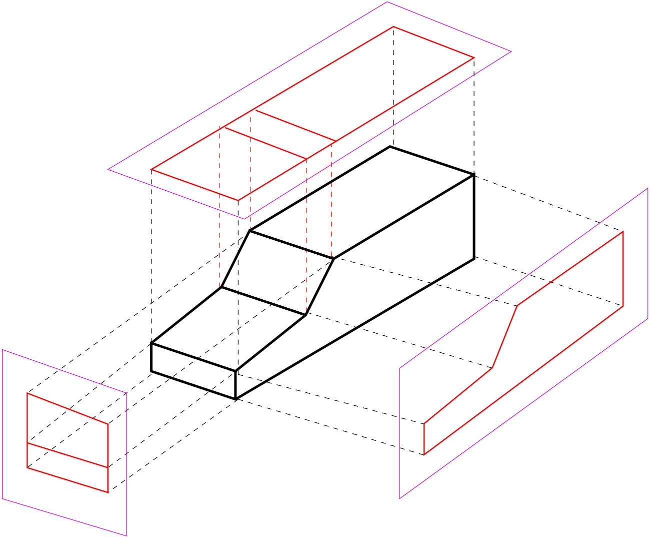
Parallel Projections

- **Orthographic:** Projection plane is perpendicular to the direction of projection.
  - If direction of projection parallel to the axes: *plan, elevation, side elevation*.
  - If DoP $(\pm 1, \pm 1, \pm 1)$: *isometric* projection.

Orthographic Projections

- Lengths parallel to the projection plane are preserved.
- Only direction of projection matters; distance from the point to the projection plane doesn't.
- Good approximation for a camera with a long focal length. (Orthographic with uniform scaling).
- Plan, elevation, side views etc.

Orthographic Projections (cont.)



Orthographic Projection Equation

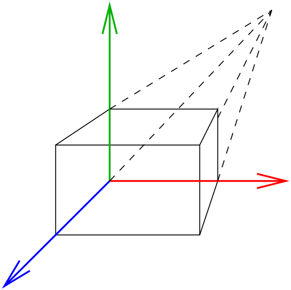
- Can be expressed as a matrix equation:

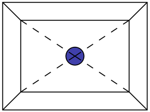
$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

- If uniform scaling is involved, the top two 1's should be the scale factor.

Perspective Projections

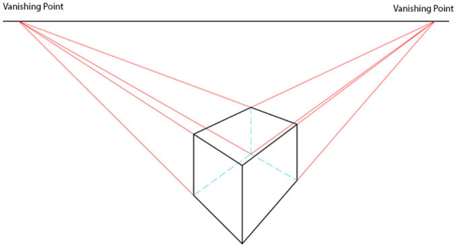
- Can be characterized by the number of **vanishing points**. (projections of points at infinity).
- Depends on the number of axes the projection plane intersects.
- 1-point, 2-point, and 3-point perspective projections.





Perspective Projections

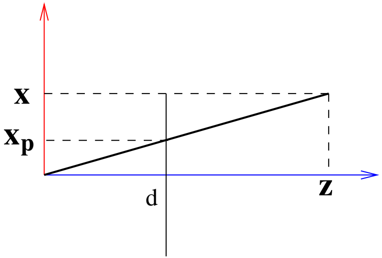
- Can be characterized by the number of **vanishing points**. (projections of points at infinity
- Depends on the number of axe the projection plane intersects
- 1-point, 2-point, and 3-point perspective projections.



Geometry of Perspective Projection

- What is x_p, y_p, z_p?
- We know x, z, and d.
- Remember similar triangles?

$$\begin{bmatrix} x_p \\ y_p \\ z_p \\ w \end{bmatrix} = \begin{bmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

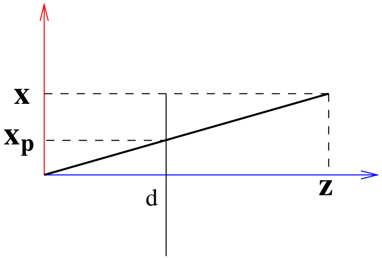


Geometry of Perspective Projection

- $\frac{x_p}{d} = \frac{x}{z}, \quad \frac{y_p}{d} = \frac{y}{z}, \quad z = d.$
- In matrix form,

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{d} & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

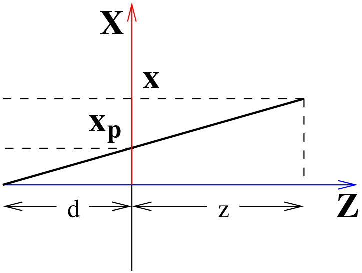
- Coordinates scaled down proportional to the depth or z values.



Another View

- Shift origin to lie on the projection plane, CoP at $(0, 0, -d)$, we get the matrix:

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

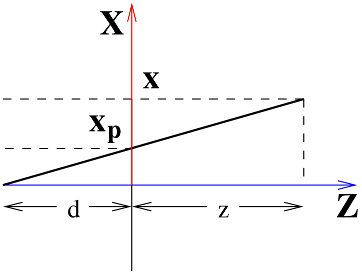


Another View

- Shift origin to lie on the projection plane, CoP at $(0, 0, -d)$, we get the matrix:

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{d} & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

- Orthographic Projection matrix is a special case when $d \to \infty$.



$$x_p = \frac{xd}{z+d} = \frac{x}{1+\frac{z}{d}}, \quad z_p = 0$$

Projections: Summary

Perspective

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{d} & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$d < \infty$$

Orthographic

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{d} & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$d = \infty$$

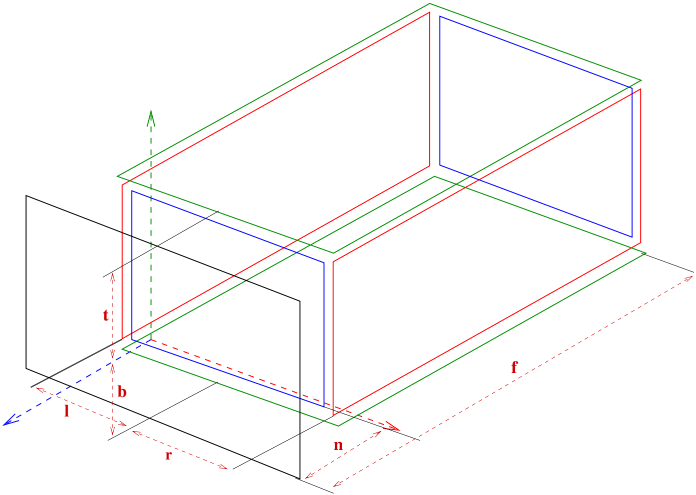
Volume of Visibility

- Cameras have finite fields of view in horizontal and vertical directions.
- What is the shape of its visible space?
- A cylinder for orthographic projections and a cone starting from the CoP for perspective may seem natural.
- Mathematics is difficult for cones; rectangular structures are easier!
- **View Volume:** The volume of potentially visible space.

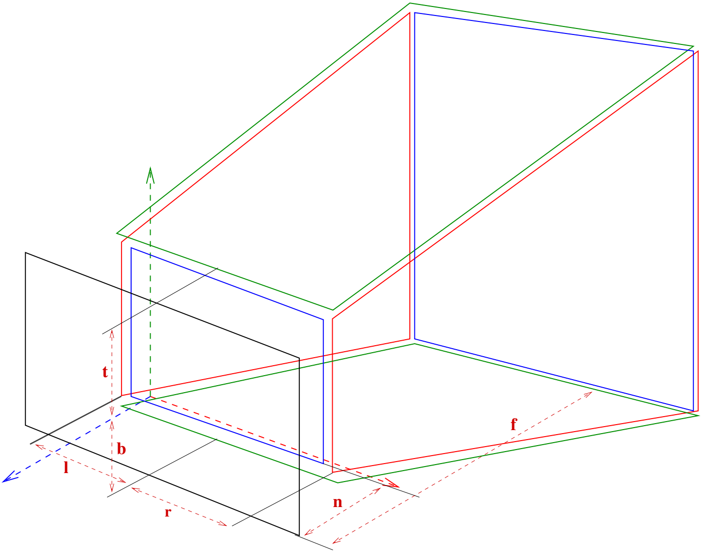
View Volume

- View volume is a cube for orthographic cameras and a (truncated) pyramid for perspective projections.
- 4 planes (left, right, top, bottom) define the view volume.
- Graphics cameras use 2 additional planes to limit visibility: **near & far!**
- Planes are specfied in VRC; they move with the camera

Orthographic View Volume

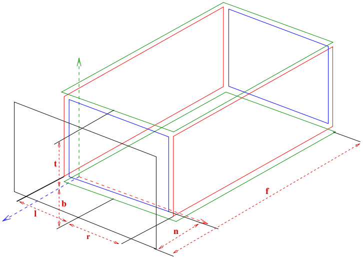


Perspective View Volume

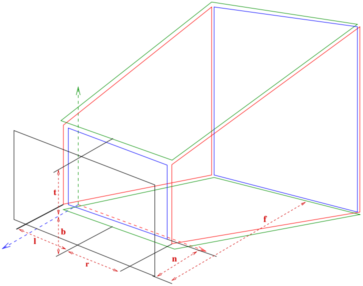


Orthographic & Perspective

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



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



- View volume: Top, Bottom, Left, Right, Near, Far!

What About the Focal Length?

- An ideal pin-hole camera has the whole world in focus.
- Finite focal-length lenses introduce the effect of focus in real cameras.
- Even for them, the depth of field (region in focus) increases as the f-stop increases or the aperture gets smaller.
- Computer Graphics simulates ideal pin-hole cameras.
- Depth of field can be simulated by intentional blurring.

View Volume Specification

- View volume is specified by 6 planes:
  left, right, top, bottom, near, far. All values in VRC

  `glm::frustum()` or `glm::ortho()`

- left, right, top, bottom: signed distances.
  near, far: positive distances to planes.

- Needn't be symmetric!

Alternate Specification

- Symmetric view volumes: horizontal and vertical fields of view $\theta_{\mathbf{h}}, \quad \theta_{\mathbf{v}}$
- For symmetric perspective view volumes:

  $\tan \frac{\theta_h}{2} = \ldots?$

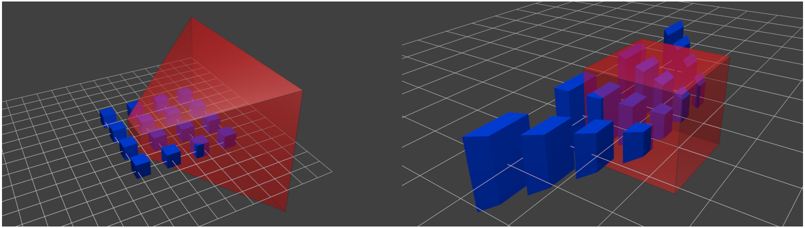
  $\tan \frac{\theta_v}{2} = \ldots?$
- Aspect ratio: top / right.

  Using `glm::perspective()`

Canonical View Volume

- Projection is not performed right away; instead, map the view volume to a cube of fixed dimensions, called the **canonical view volume** or a standard view volume
- A **normalizing matrix** performs this transformation.
- Why?
  - Easier to eliminate objects outside the view volume.
  - Orthographic & perspective aren't different.
  - The z-coordinates not thrown away. (Used later!)

Canonical View Volume: Visualization

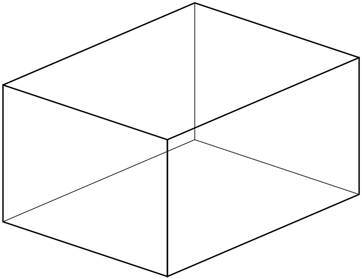


http://www.opengl-tutorial.org/beginners-tutorials/tutorial-3-matrices/

Canonical View Volume: Dimensions

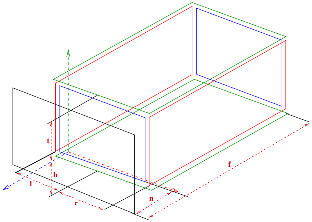
- OpenGL:

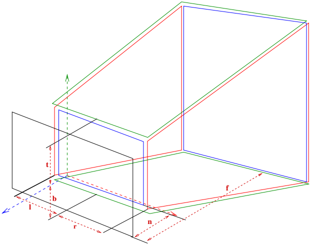
  $$-1 \leq x \leq 1$$
  $$-1 \leq y \leq 1$$
  $$-1 \leq z \leq 1$$



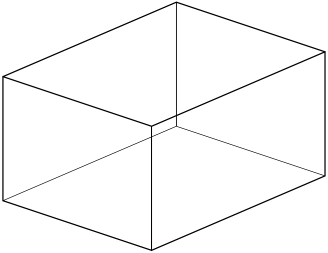
- Canonical view volume is the Orthographic View Volume, with appropriate scaling.

Orthographic & Perspective





To target view volume:



Orthographic Normalizing Matrix

- What are the side lengths on start?
- What are the side lengths at end?
- Whats the scale factor?
- Where is the origin at strat? At end?
- How do we achieve that?

Orthographic Normalizing Matrix

- Lengths (right - left), (top - bottom) and (far - near) scaled to 2.
- Shift origin so as to range from -1 to +1.
- Matrix??

Orthographic Normalizing Matrix (cont.)

- Matrix:

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

- $(0,0,-near)$ maps to $+1$ and $(0,0,-far)$ maps to -1.
- Drop z and use the (x,y) coordinates as the (normalized) window coordinates.