HOCHSCHULE FURTWANGEN UNIVERSITY HFU

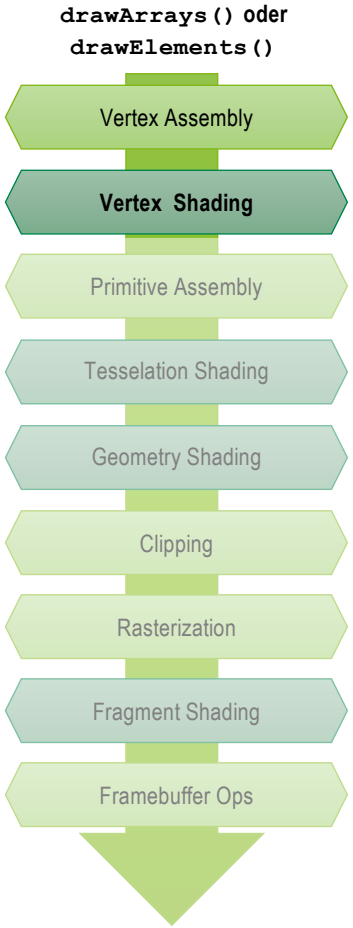
CODE3

SoSe 2025

Prof. Dr.-Ing. Uwe Hahne

Appendix on transformations

Vertex Assembly and Vertex Shader



Vertex Assembly

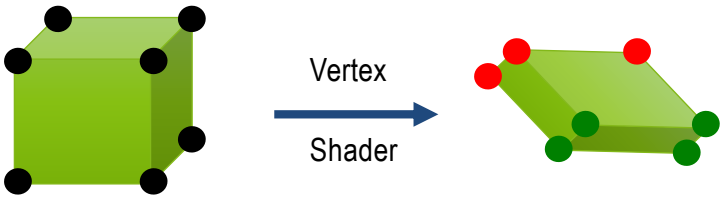
Search for all attributes of a single vertex from potentially multiple vertex buffers

Vertex Shader

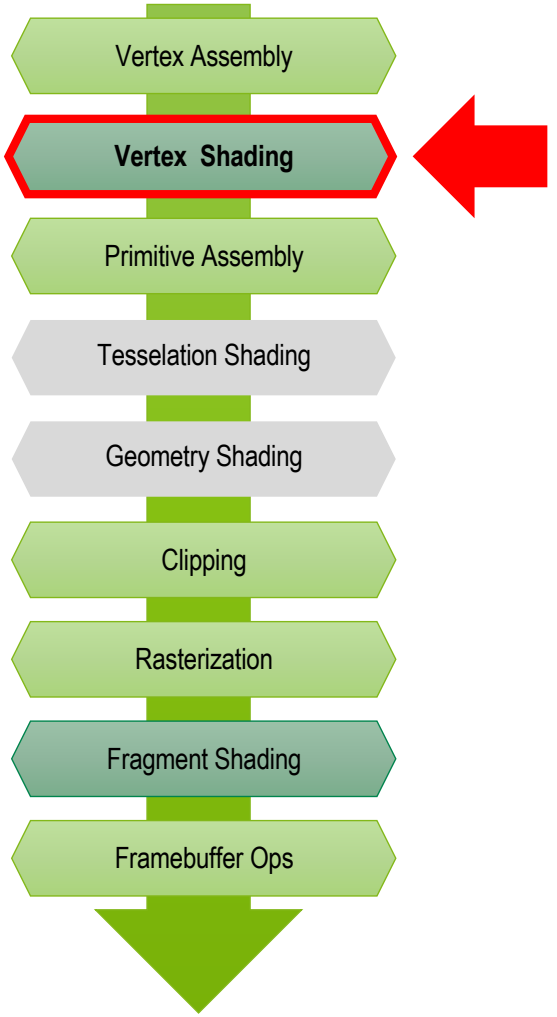
Transformation of the vertex into clip coordinates (e.g. depending on the current camera / scene transformation)

Generation or transformation of further vertex attributes (e.g. normals, texture coordinates, colors)

Lighting calculation per vertex (unusual nowadays)



Configuration of the pipeline: install vertex shader




```
attribute vec4 aPosition;
attribute vec4 aVertexColor;
uniform mat4 uModelViewMatrix;
uniform mat4 uProjectionMatrix;
varying vec4 vColor;

void main() {
    gl_Position = uProjectionMatrix
*
                  uModelViewMatrix *
                  aPosition;
    vColor = aVertexColor;
}
```

What global data
does the shader expect?

Transformation in vertex shader

`attribute:` Vertex attribute, receives data from an attribute buffer (VBO)
The position is potentially different for each vertex.

Two matrices are passed as global variables (**`uniform`**); they are constant for all vertices during a **`gl.draw*()`** command.

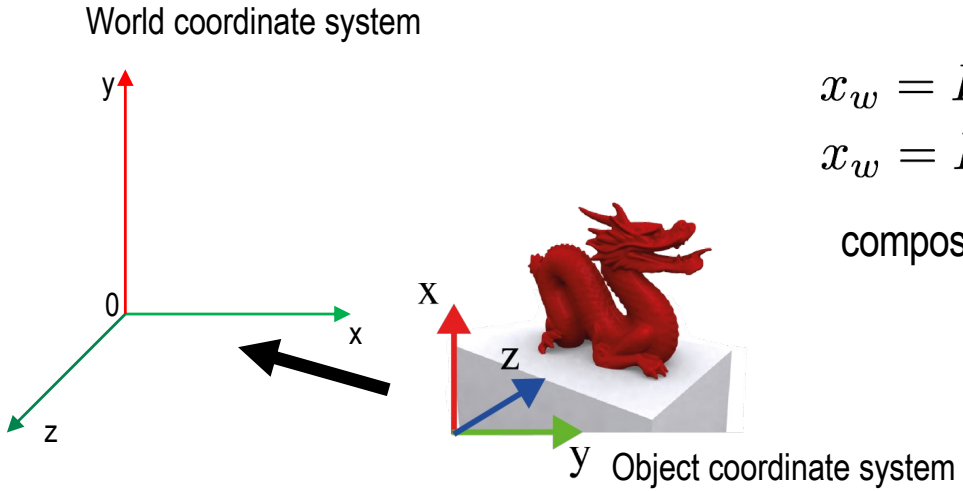
```
attribute vec4 aPosition;
uniform mat4 uModelViewMatrix;
uniform mat4 uProjectionMatrix;

void main() {
    gl_Position = uProjectionMatrix * uModelViewMatrix *
                  aPosition;

}
```

Vertex Shader (GLSL)

Goal: **`gl_Position`** must contain the vertex position in clip coordinates after the shader has been executed.

Matrix multiplications are read from right to left: first apply `uModelViewMatrix`, then `uProjectionMatrix`.

Modeling transformation



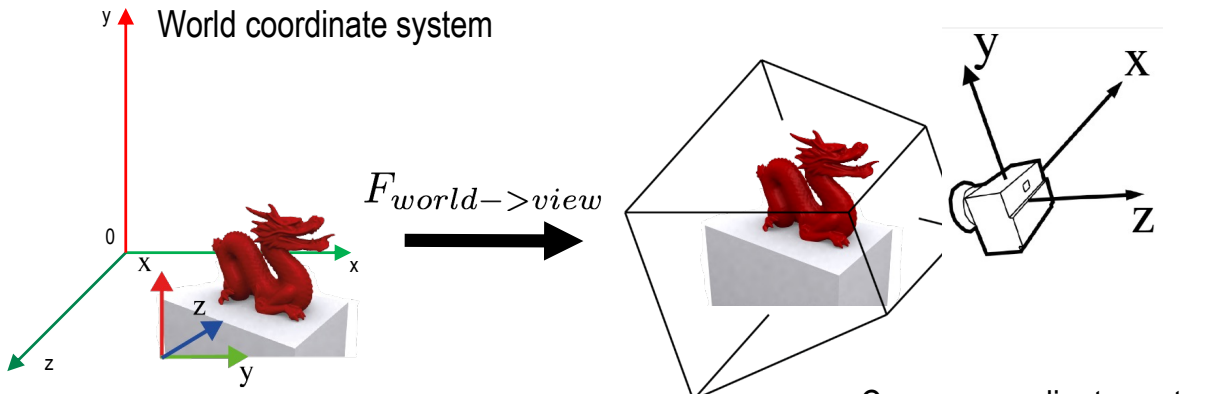
$$x_w = F \cdot x_o$$

$$x_w = F_3 \cdot F_2 \cdot F_1 \cdot x_o$$

composite transformations

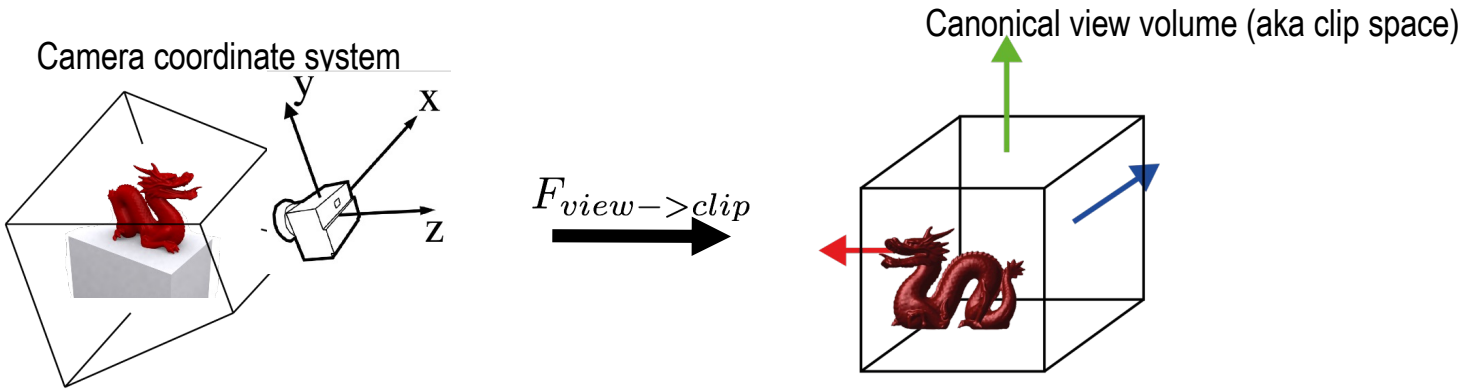
$$F(x,y,z,w) = \begin{bmatrix} a_{11} & a_{12} & a_{13} & t_x \\ a_{21} & a_{22} & a_{23} & t_y \\ a_{31} & a_{32} & a_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

View transformation

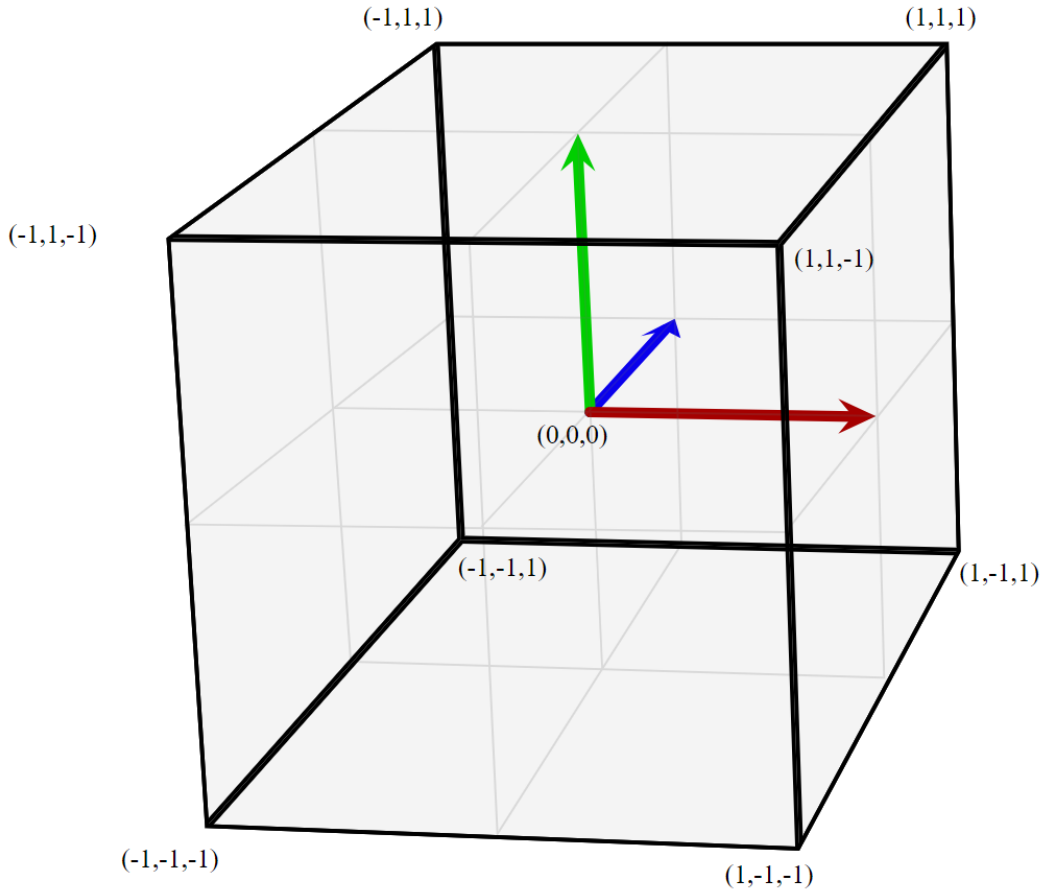


$$F_{modelView} = F_{world->view} F_{object->world}$$

Projection transformation

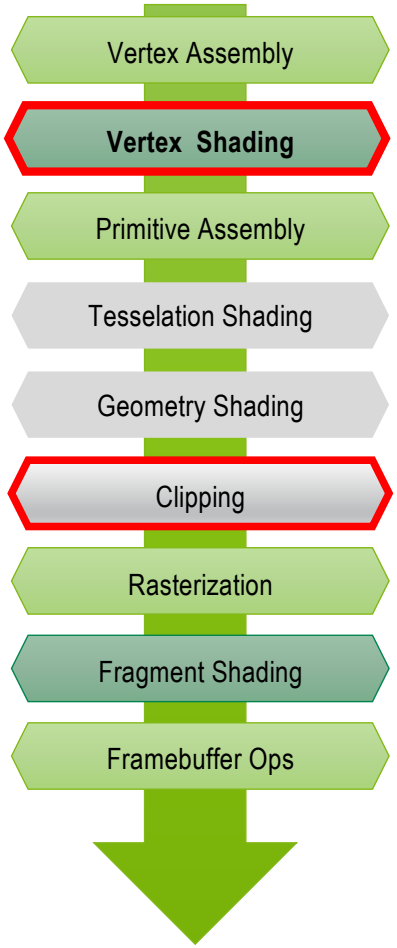


Clip space

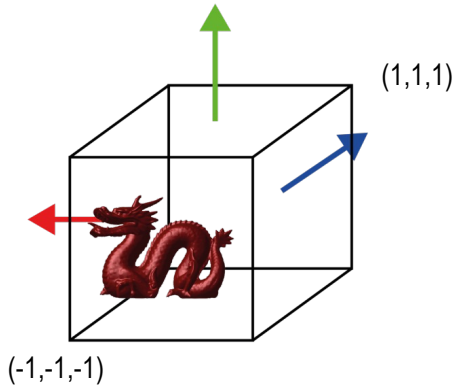


Clipspace

Viewport - Transformation



- Vertex shader returns position in canonical volume before perspective division

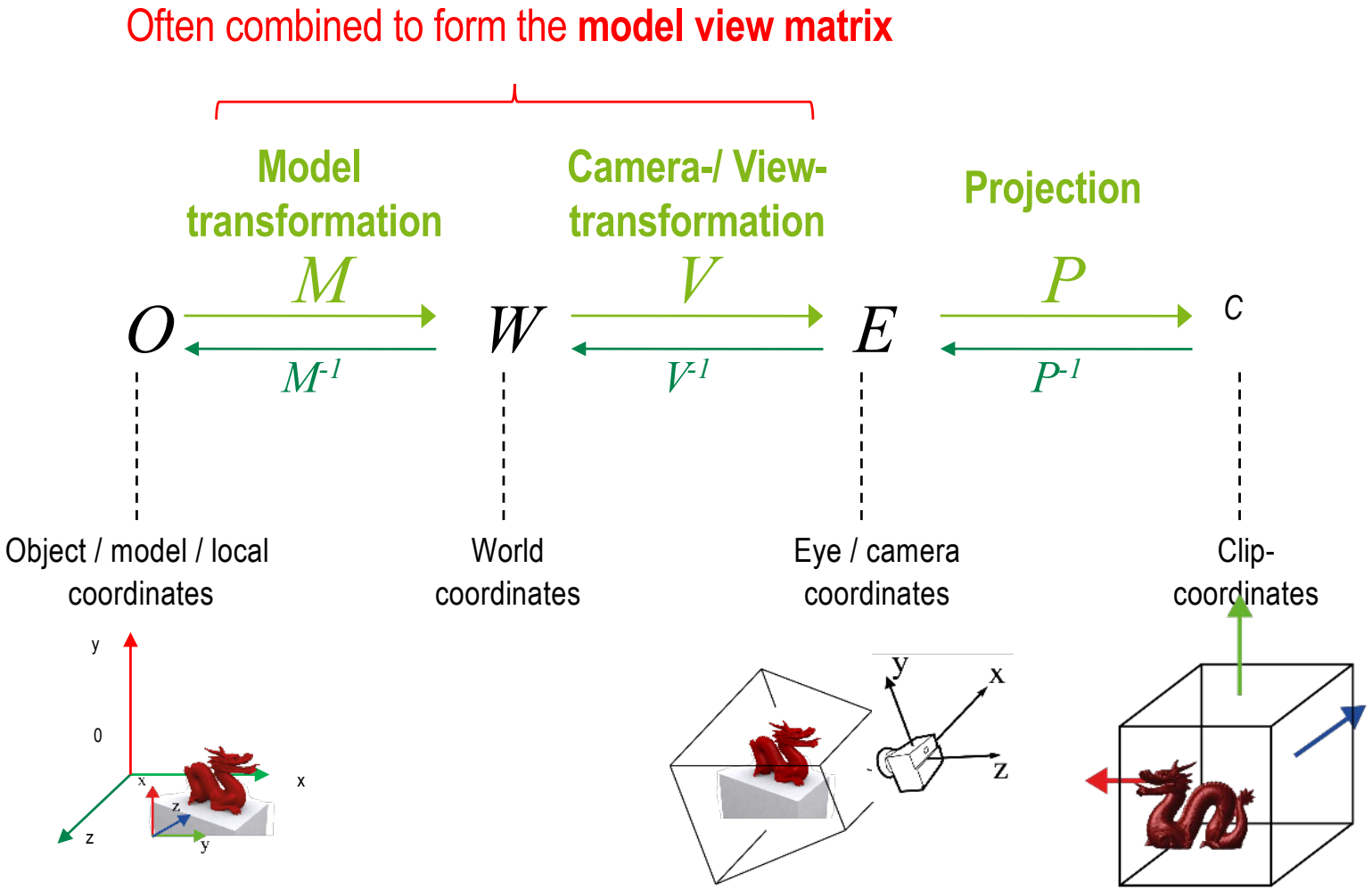


- Clipping
- Perspective division

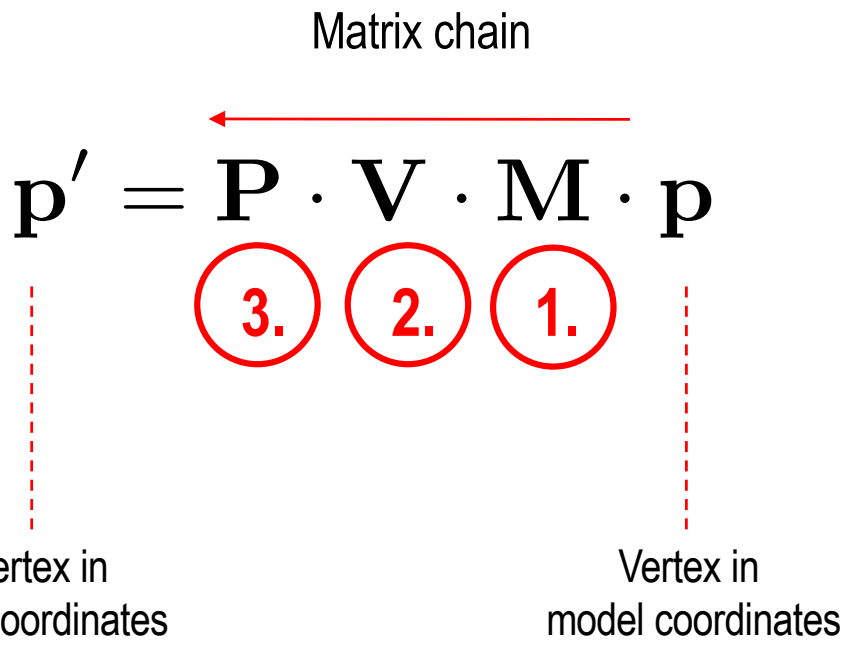
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \frac{1}{w_c} \begin{bmatrix} x_c \\ y_c \\ z_c \\ w_c \end{bmatrix}$$

- Viewport transformation (window coordinates)

Transformation chain from model to clip coordinates



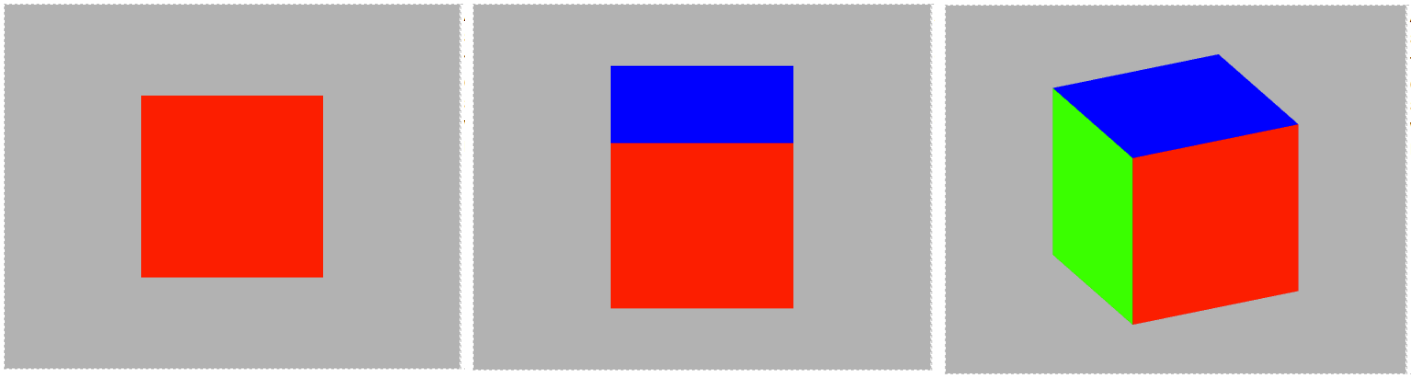
Die Transformationskette von Modell- in Clipkoordinaten



Attention order!

- The vertex to be transformed is on the right
- The transformations are multiplied in sequence from the left
- Read from right to left

Effect of the model-view-transformation matrix

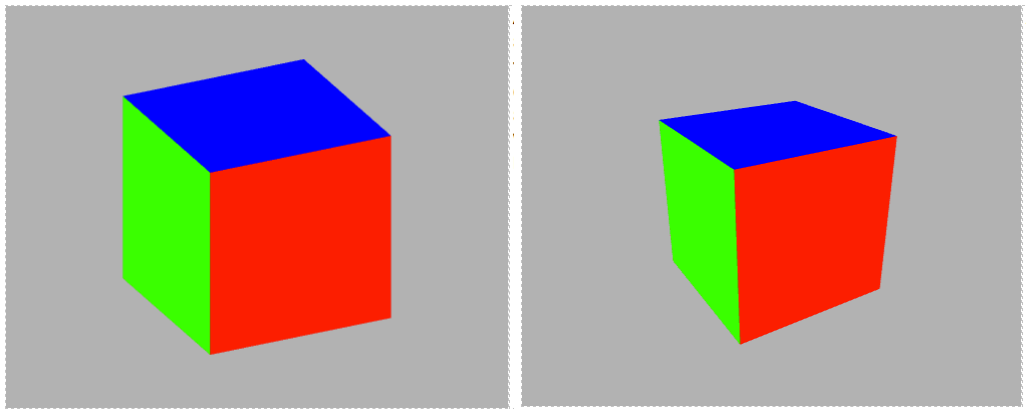


Identity | Rotation around X-axis | Rotation around X- and Y-axis

Who is actually rotating here, the camera or the scene?

- The model view matrix is a relative transformation between two coordinate systems
- Rotation of the scene corresponds to opposite rotation of the camera, one transformation is the inverse of the other

Effect of the projection matrix



Orthographic projection | Perspective projection

Orthographic projection (“telephoto lens”)

- Parallel lines remain parallel

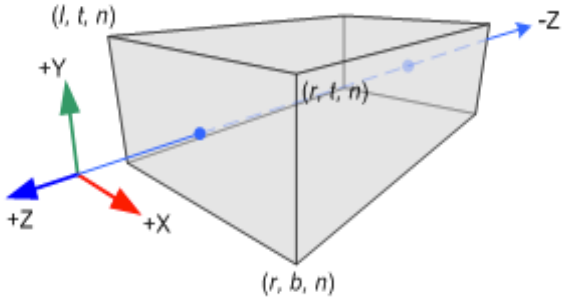
Perspective projection (“wide-angle lens”)

- “Oblique” projection with a selectable aperture angle
- Falling lines, distant objects appear smaller

Orthographic projection

Illustration by Song Ho Ahn, www.songho.ca

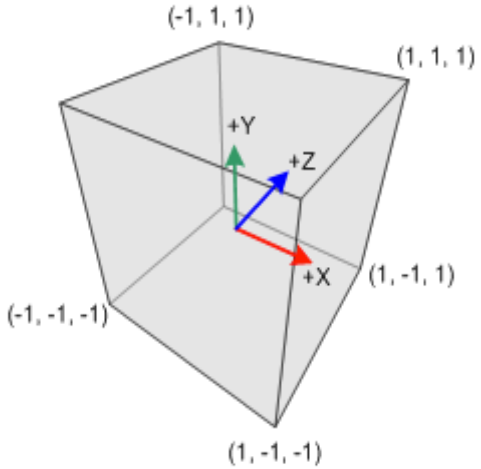
Camera coordinates



$$\begin{pmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & \frac{-2}{f-n} & -\frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

projection matrix

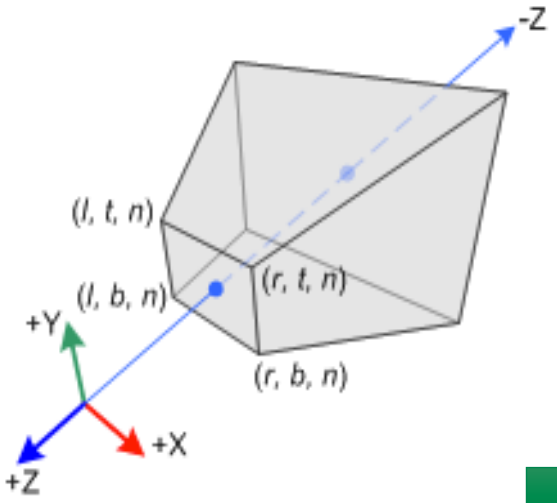
Clip coordinates



Perspective projection

Illustration by Song Ho Ahn, www.songho.ca

camera coordinates



$$\begin{pmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & \frac{-(f+n)}{f-n} & \frac{-2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

projection matrix

clip coordinates