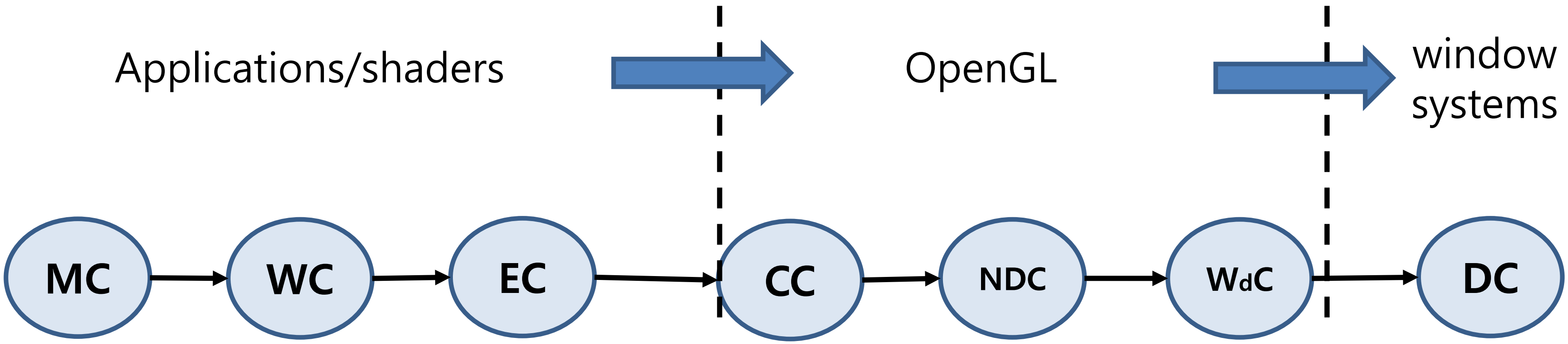
Projections, Viewing Pipeline

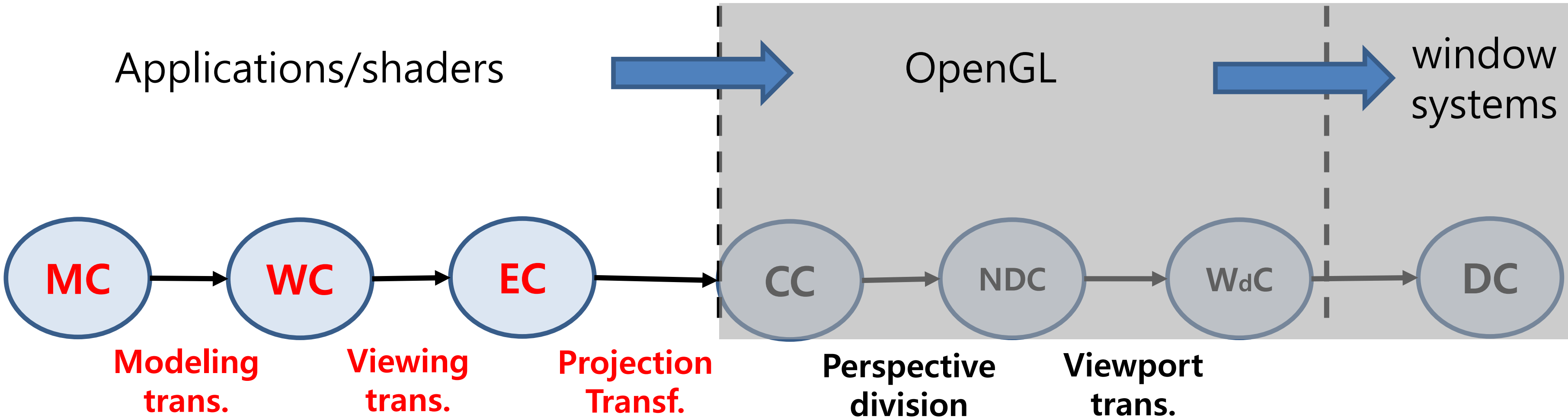
Sung Soo Hwang

OpenGL 3D Viewing Pipeline



MC: Modeling Coordinates
WC: World Coordinates
EC: Eye Coordinates
CC: Clip Coordinates
NDC: Normalized Device Coordinates
WdC: Window Coordinates
DC: Device Coordinates

OpenGL 3D Viewing Pipeline

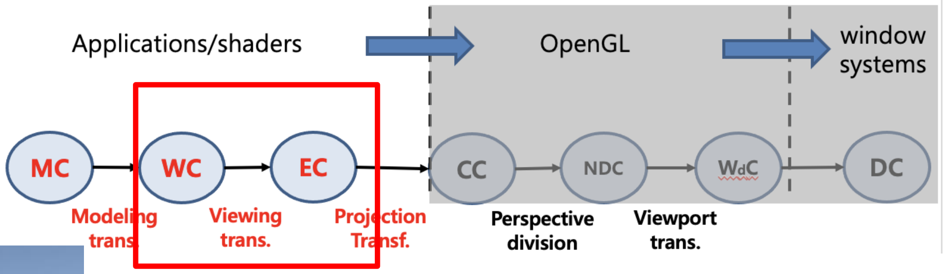


MC: Modeling Coordinates
WC: World Coordinates
EC: Eye Coordinates
CC: Clip Coordinates
NDC: Normalized Device Coordinates
WdC: Window Coordinates
DC: Device Coordinates

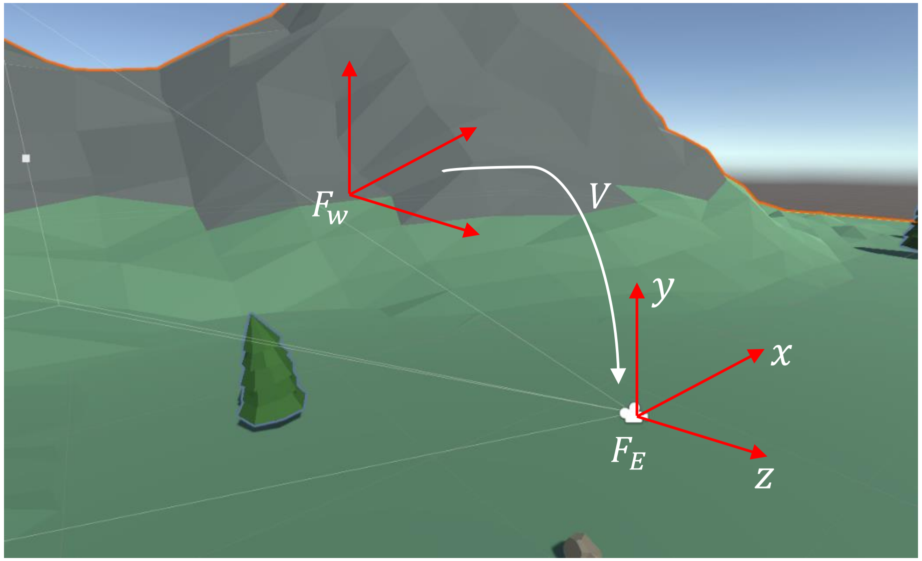
```
#version 430

in vec4 vPosition;
in vec4 vColor;
out vec4 fColor;
layout(location=1) uniform mat4 M;
layout(location=2) uniform mat4 V;
layout(location=3) uniform mat4 P;


void main()
{
    gl_Position = P * V * M * vPosition;
    fColor = vColor;
}
```

shader.vert

Viewing Transformation



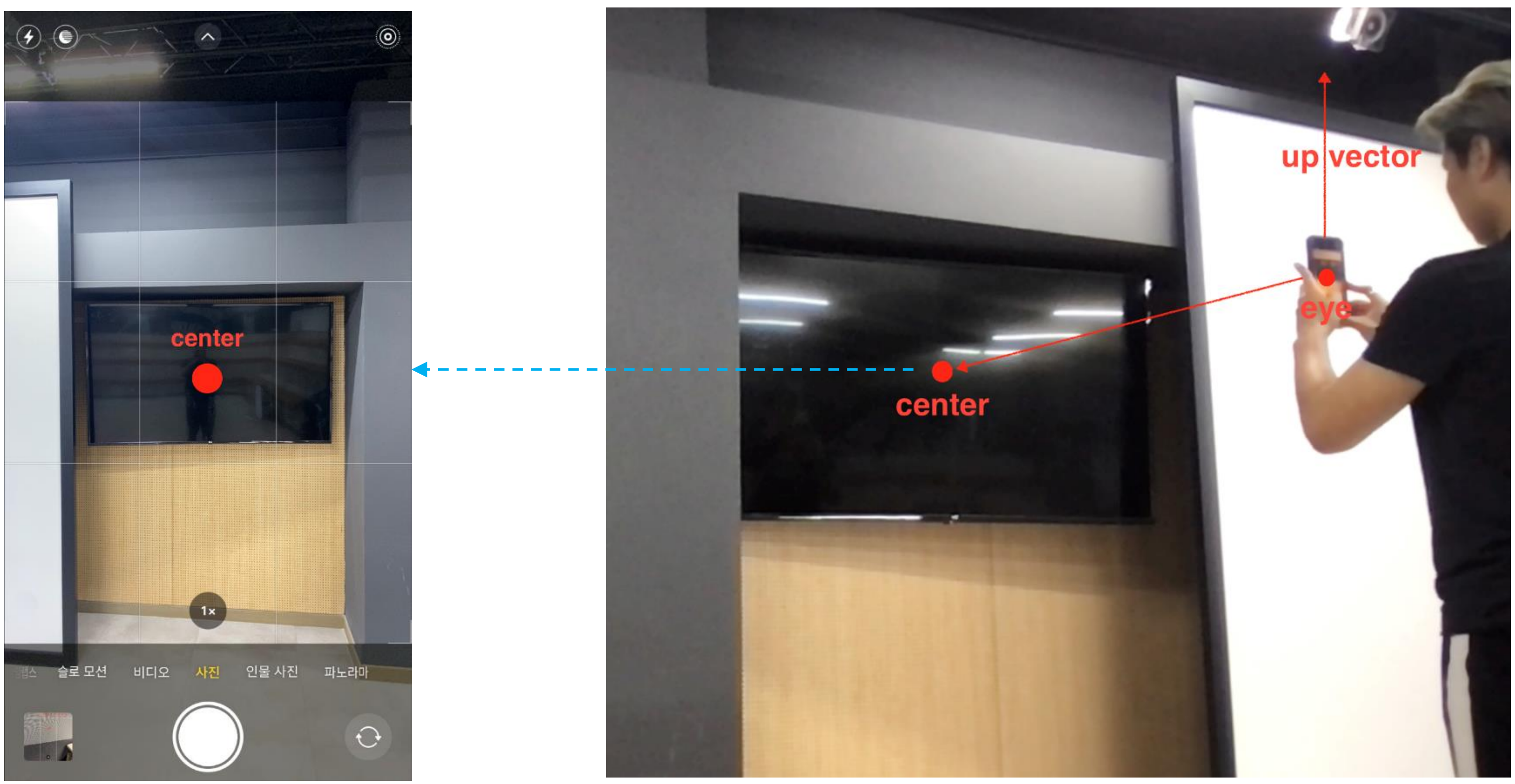
Viewing
Transformation (V)



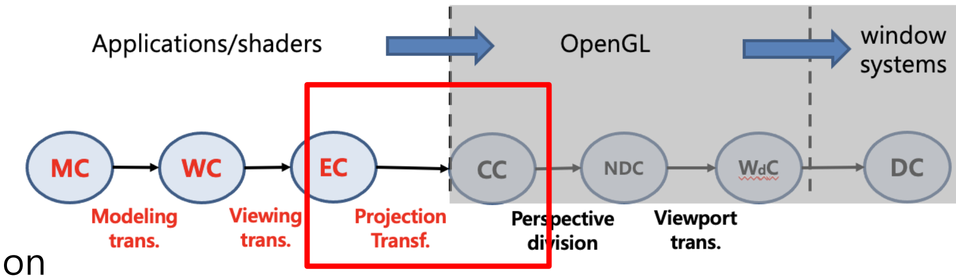
Eye (Camera) coordinate system (F_E):
The eye coordinate system is a coordinate system defined by the camera position and orientation.

Viewing Transformation

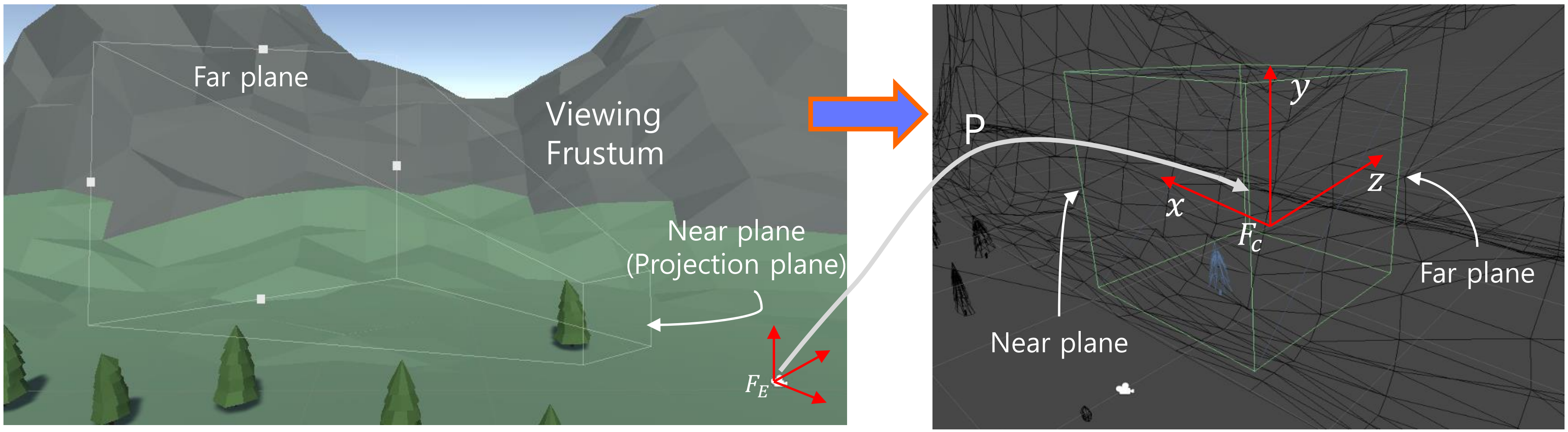
- Camera position and orientation in world coordinates
  - center: Center of projection, projection reference point.
  - eye: The camera position is a vector in world space that points to the camera's position.
  - up vector: view up vector



Projection Transformation

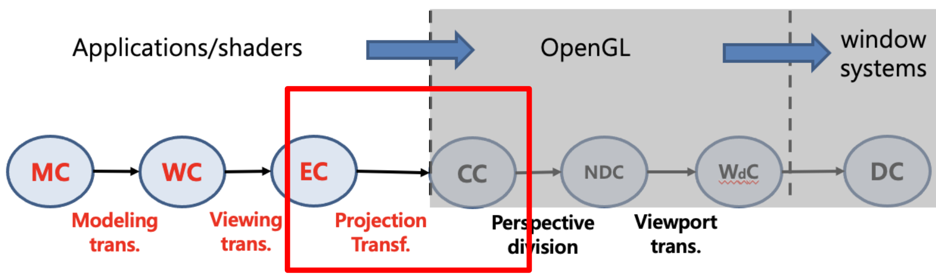


Projection Transformation (P)

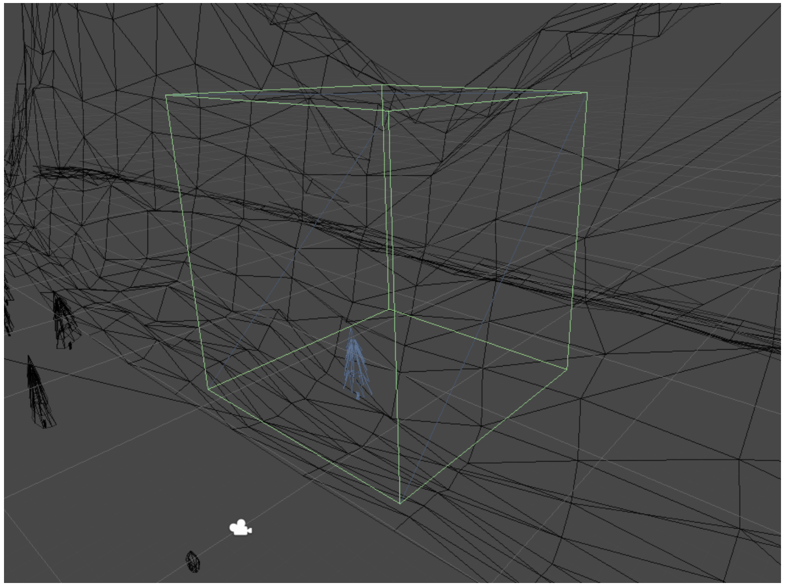


Clip coordinate system (F_C):
All coordinate values are converted into clip coordinates by normalizing the viewing frustum into a cube

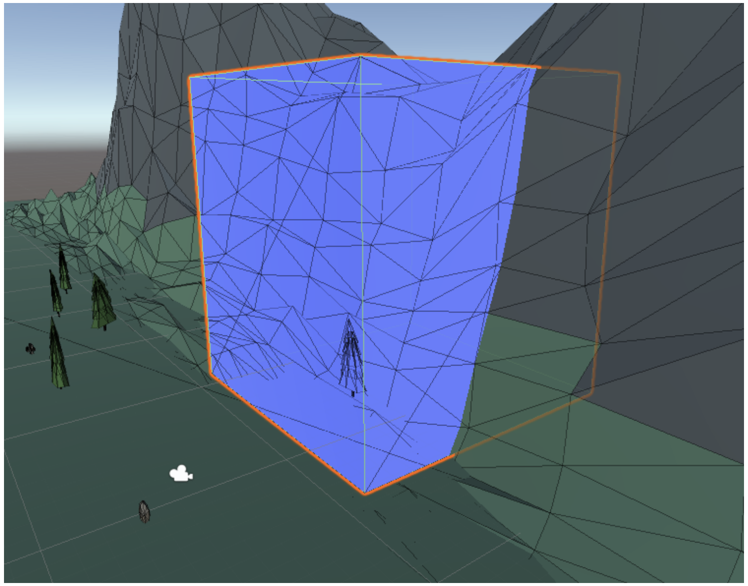
Projection Transformation



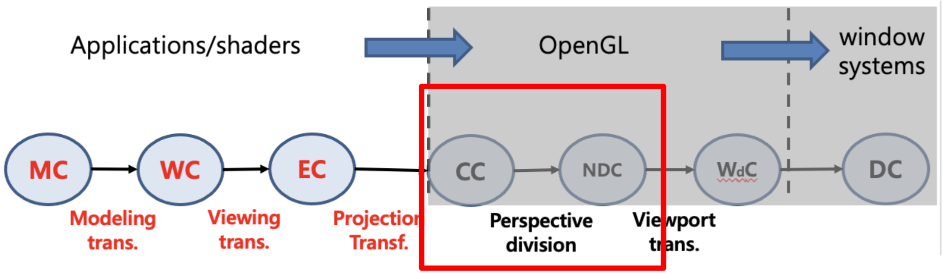
Renderable space

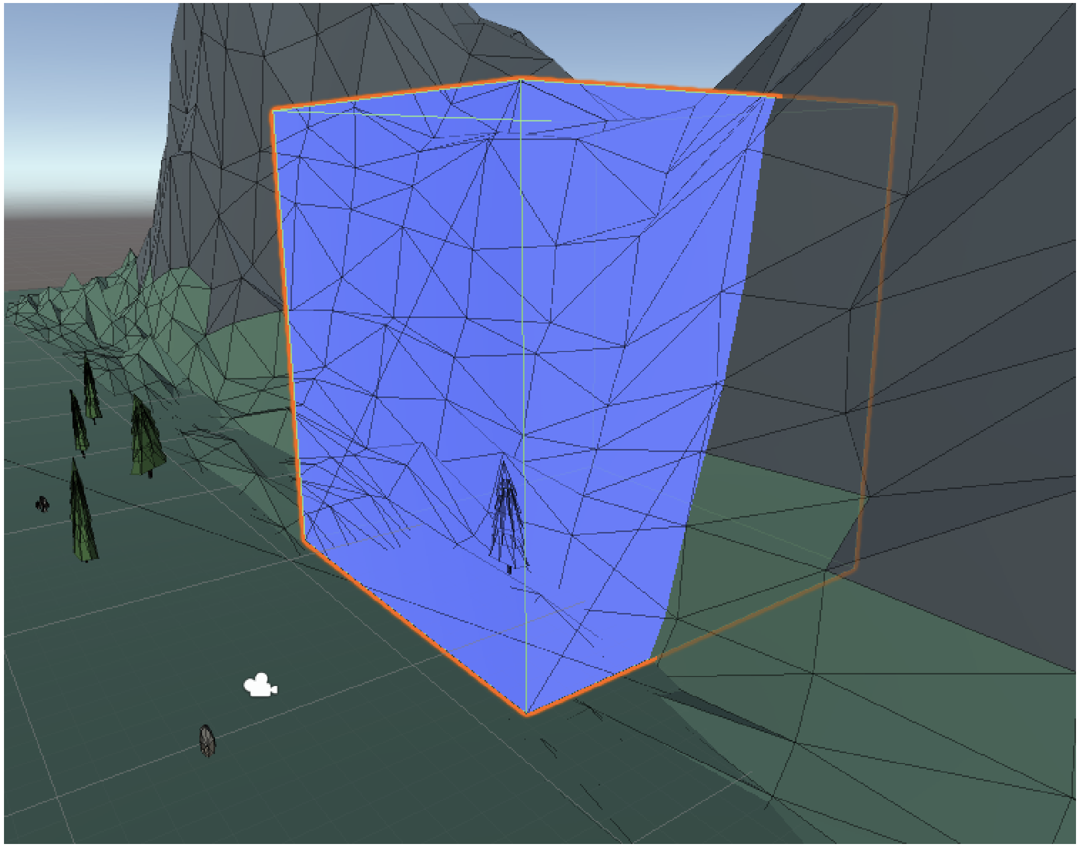


3D Clipping



Viewing Transformation

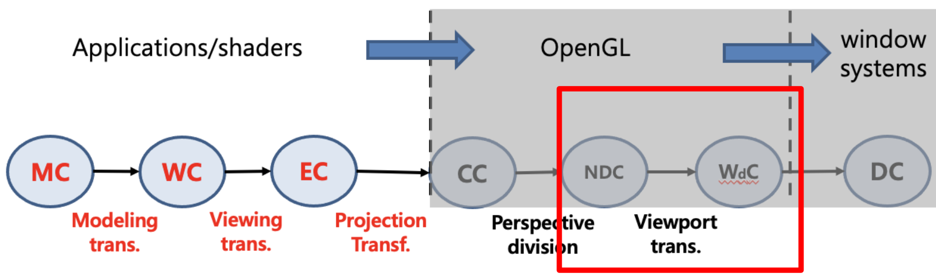




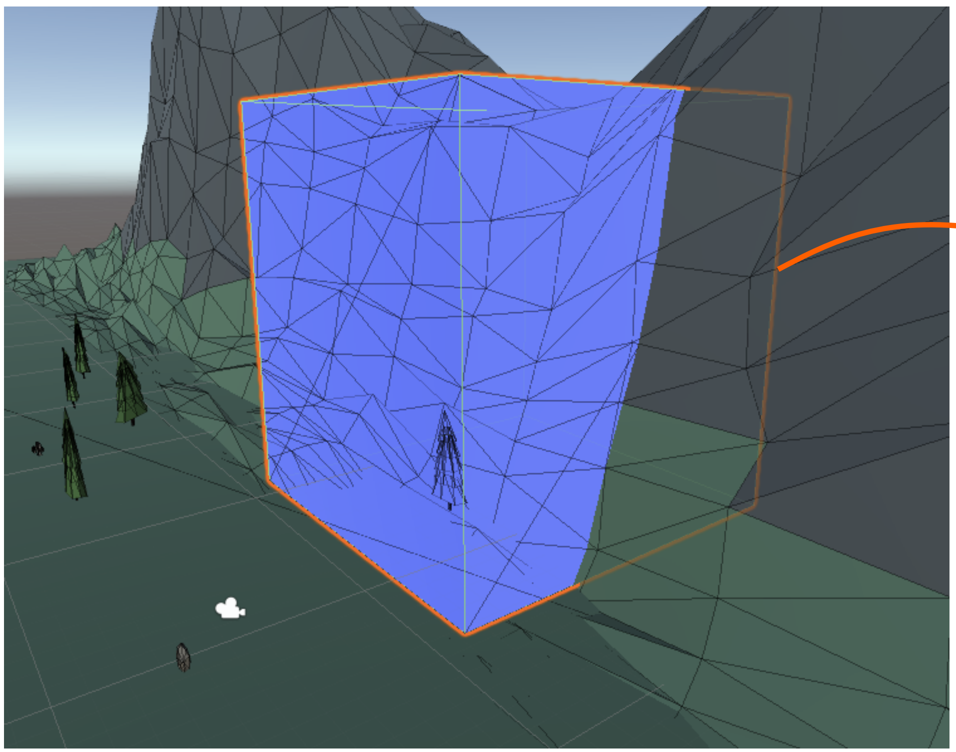
$$p' = \frac{1}{w} p$$

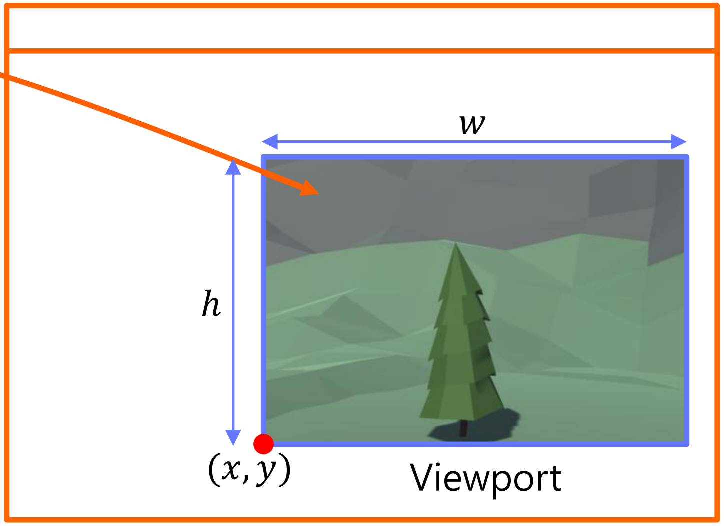
Perspective division:
Convert all the clip coordinates in the cube by mapping them to the corresponding Euclidean point.

Viewing Transformation



Viewport transformation



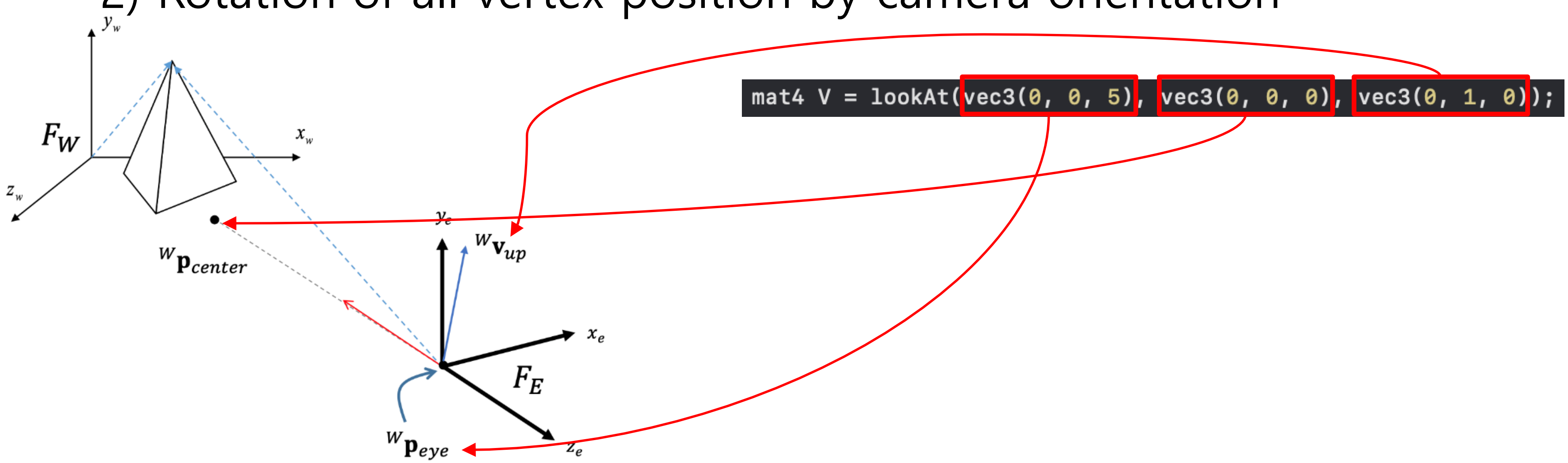


Window coordinate system

Viewing Transformation

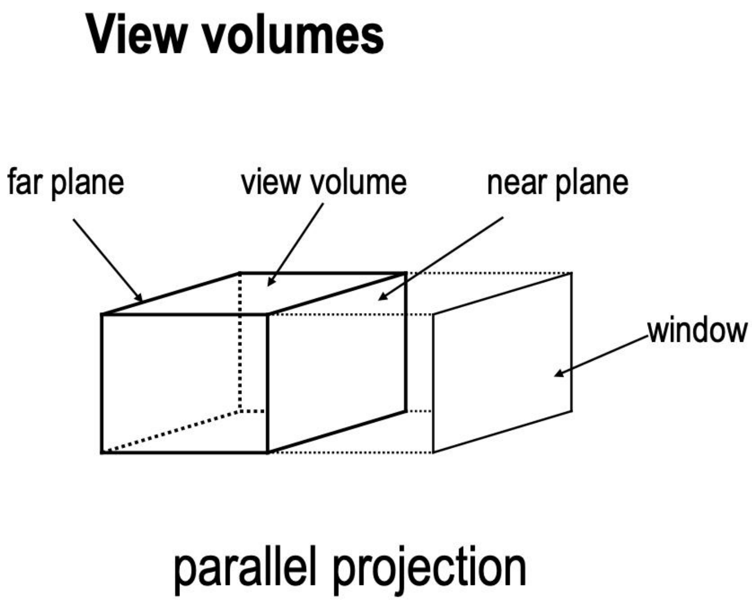
- Camera position and orientation in world coordinates
  - Center of projection, projection reference point (${}^{W}p_{center}$)
  - The camera position is a vector in world space that points to the camera's position (${}^{W}p_{eye}$)
  - View up vector(${}^{W}v_{up}$)

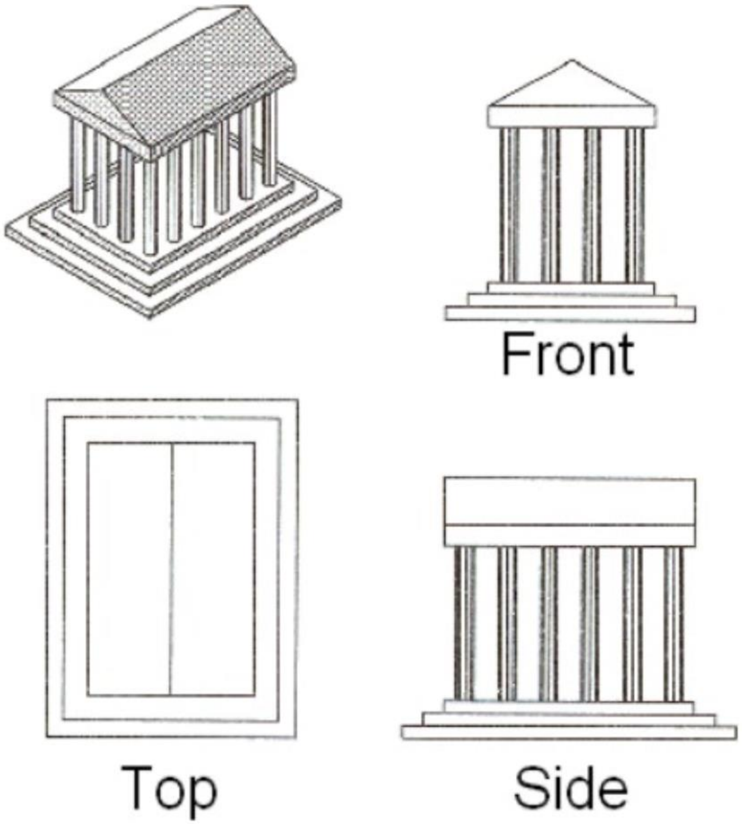
- Transformation
  - 1) Translation of all vertex positions by projection center
  - 2) Rotation of all vertex position by camera orientation



Projection Transformation

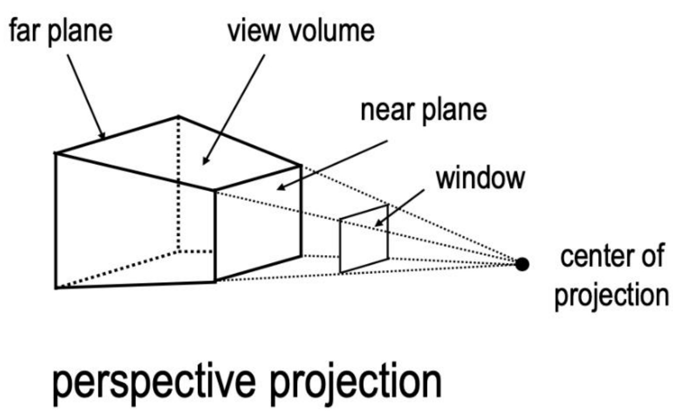
- Parallel (Orthographic) projection
  - Rays or projection lines in parallel projection are **parallel to each other**.

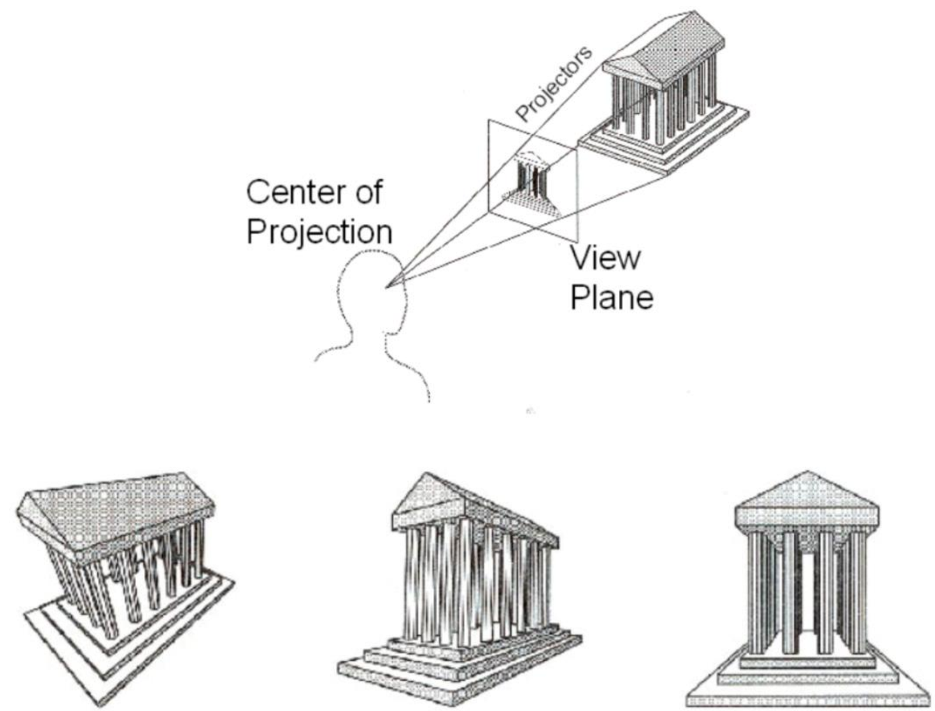




Projection Transformation

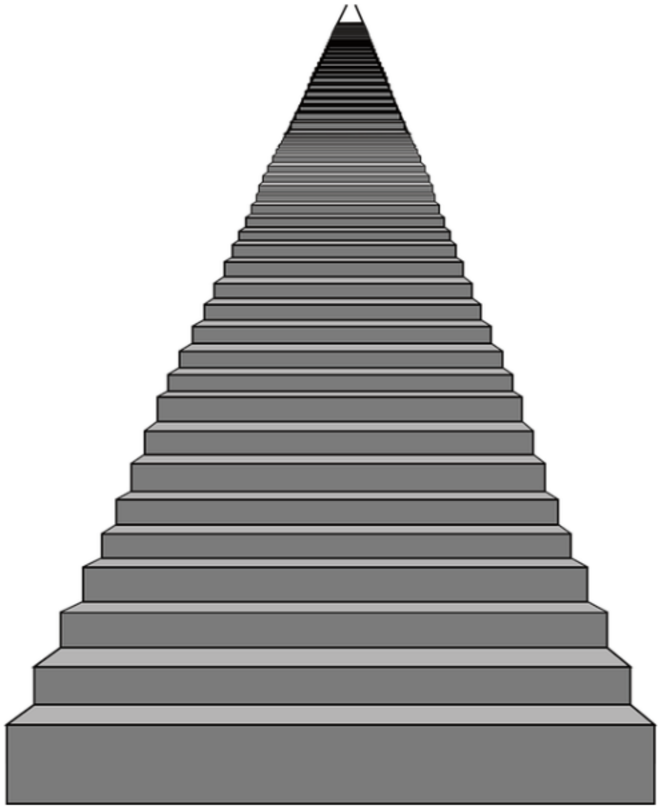
- Perspective projection
  - In this, all the parallel lines in the object which is not parallel to the view plane are converged and **the point of converging is known as the vanishing point**.





Projection Transformation

- Perspective projection
  - In this, all the parallel lines in the object which is not parallel to the view plane are converged and **the point of converging is known as the vanishing point**.



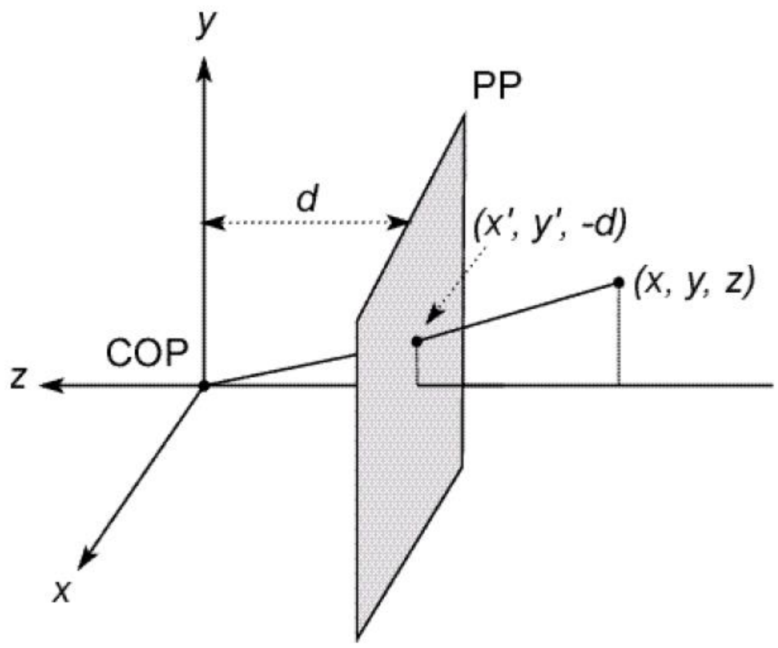
Projections

- Orthographic projection(parallel projection)
  - We specify a direction of projection(DOP) instead of a COP

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Projections

- Perspective projection
  - Consider the projection of a point onto the projection plane



  - By similar triangles, we can compute x′ and y′

$$\frac{y'}{d} = \frac{y}{z} \Rightarrow y' = \frac{d}{z}y,\ x' = \frac{d}{z}x$$

Projections

- Perspective projection

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \left(\frac{d}{z}\right) x \\ \left(\frac{d}{z}\right) y \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z/d \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$