

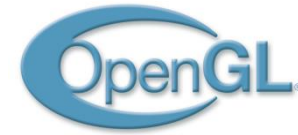
SUPSI

Computer Graphics

Mathematics for Computer Graphics – Quick summary

Achille Peternier, adjunct professor





Transformations

and their glm methods

$$\text{Translation}_{\text{glm::translate}} = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Scaling}_{\text{glm::scale}} = \begin{bmatrix} x & 0 & 0 & 0 \\ 0 & y & 0 & 0 \\ 0 & 0 & z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Vertices as
column vectors:

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \xleftarrow{w}$$

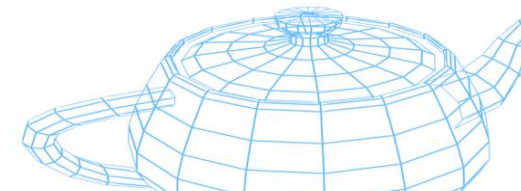
$$\text{Rotation}_x_{\text{glm::rotate}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Rotation}_y_{\text{glm::rotate}} = \begin{bmatrix} \cos \alpha & 0 & \sin \alpha & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \alpha & 0 & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Rotation}_z_{\text{glm::rotate}} = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 & 0 \\ \sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Transformation
concatenation: $\mathbf{v}_n = \mathbf{T}_3 \mathbf{T}_2 \mathbf{T}_1 \mathbf{v}_p$

(using post-multiplication and column vectors)



Projections



$$\text{Orthographic} = \text{glm::ortho} \begin{bmatrix} \frac{2}{\text{right} - \text{left}} & 0 & 0 & -\frac{\text{right} + \text{left}}{\text{right} - \text{left}} \\ 0 & \frac{2}{\text{top} - \text{bottom}} & 0 & -\frac{\text{top} + \text{bottom}}{\text{top} - \text{bottom}} \\ 0 & 0 & \frac{-2}{\text{far} - \text{near}} & -\frac{\text{far} + \text{near}}{\text{far} - \text{near}} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Perspective} = \text{glm::perspective} \begin{bmatrix} \frac{f}{\text{aspect}} & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & \frac{\text{far} + \text{near}}{\text{near} - \text{far}} & \frac{2 \times \text{far} \times \text{near}}{\text{near} - \text{far}} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

fieldOfView

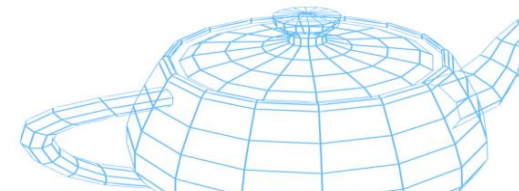
vertical (y) view angle

f

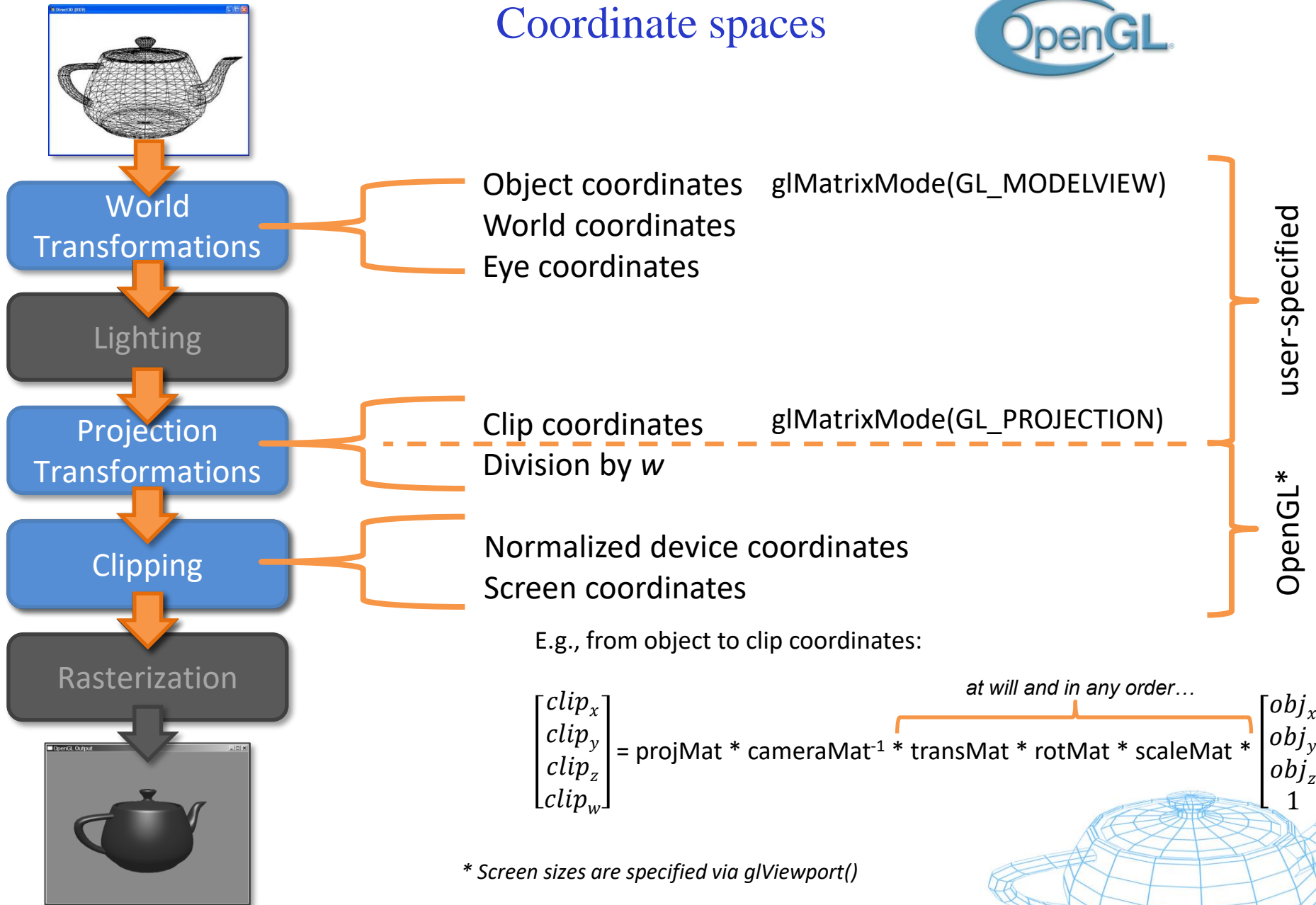
$\text{cotangent}(\text{fieldOfView}/2)$

aspect

aspect ratio (4:3, 16:9, etc.)



Coordinate spaces



GLM



```
// Main include:
#include <glm/glm.hpp>

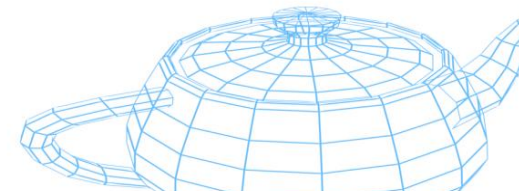
// Extension for deprecated OpenGL matrix transformations:
#include <glm/gtc/matrix_transform.hpp>

glm::scale(glm::mat4(1.0f), glm::vec3(0.5f));
glm::rotate(glm::mat4(1.0f), glm::radians(90.0f), glm::vec3(0.0f, 0.0f, 1.0f));
glm::translate(glm::mat4(1.0f), glm::vec3(10.0f, 0.0f, 0.0f));
glm::perspective(glm::radians(45.0f), 1.0f, 1.0f, 100.0f);

// Extension for printing variables via glm::to_string():
#include <glm/gtx/string_cast.hpp>

// Extension for using common math constants:
#include <glm/gtc/constants.hpp>

// Shortcuts:
glm::mat4(1.0f); // 4x4 identity matrix
glm::vec3(0.5f); // same as glm::vec3(0.5f, 0.5f, 0.5f)
```



GLM



Matrices in the documentation:

$$\begin{bmatrix} a & e & i & m \\ b & f & j & n \\ c & g & k & o \\ d & h & l & p \end{bmatrix}$$



Matrices in GLM:

```
glm::mat4 mat( a, b, c, d,
               e, f, g, h,
               i, j, k, l,
               m, n, o, p );
```

Example:

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



```
glm::mat4 translation(1, 0, 0, 0,
                      0, 1, 0, 0,
                      0, 0, 1, 0,
                      x, y, z, 1);

// or:
glm::mat4 translation =
glm::translate(glm::mat4(1.0f),
               vec3(x, y, z));
```