Computer Graphics
CS 543 – Lecture 5 (Part 3)
Viewing

Prof Emmanuel Agu

Computer Science Dept.

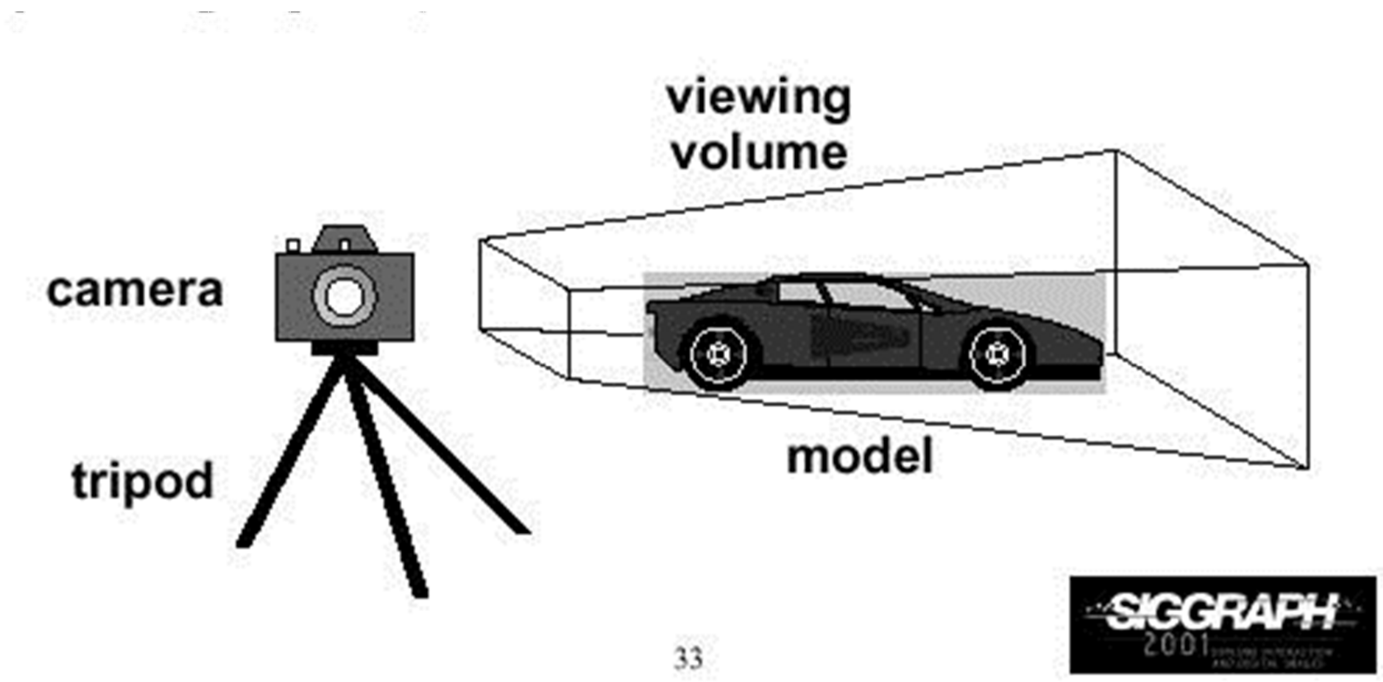
Worcester Polytechnic Institute (WPI)

Objectives

- Introduce viewing functions
- Look at alternate camera controls

3D Viewing?

- **Note:** View volume may have different shapes



Computer Viewing

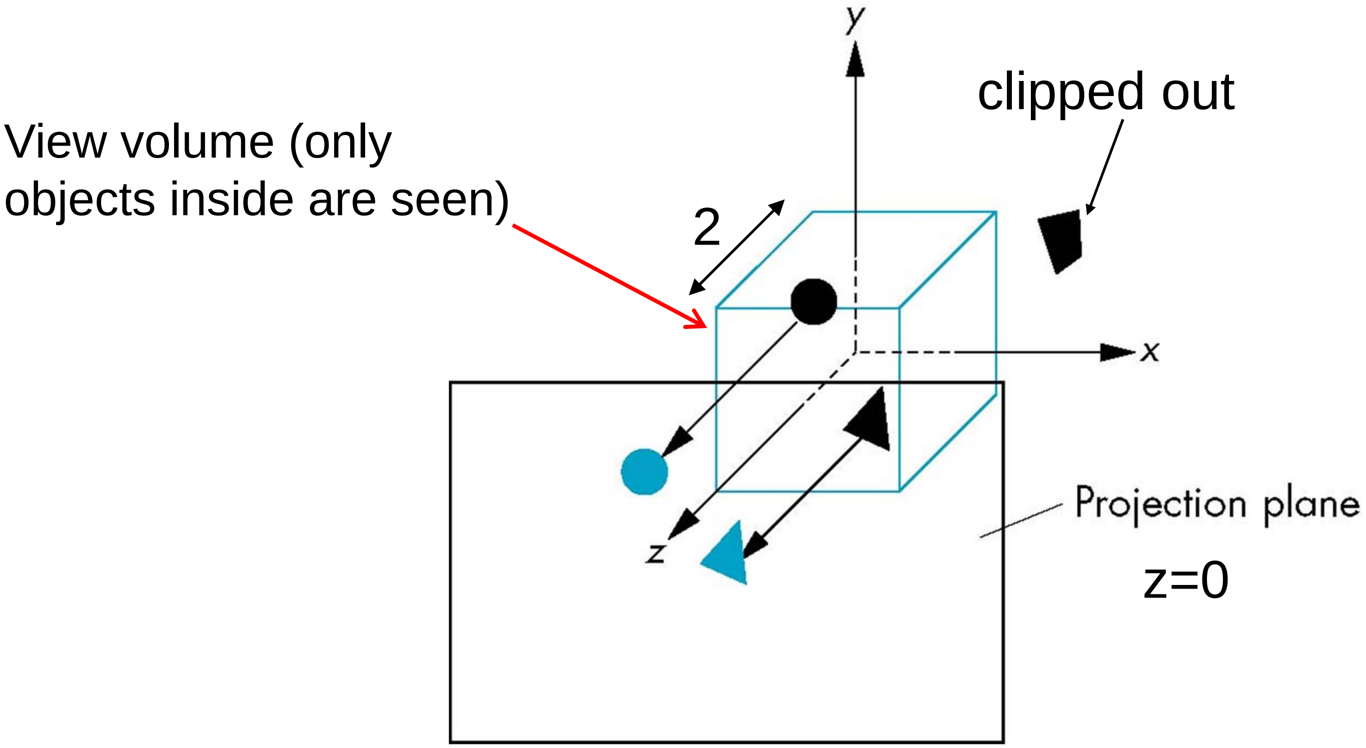
- There are three aspects of viewing process, all of which are implemented in the pipeline,
  - Positioning the camera
    - Setting the model-view matrix
  - Selecting a lens
    - Setting the projection matrix
  - Clipping
    - Setting the view volume

The OpenGL Camera

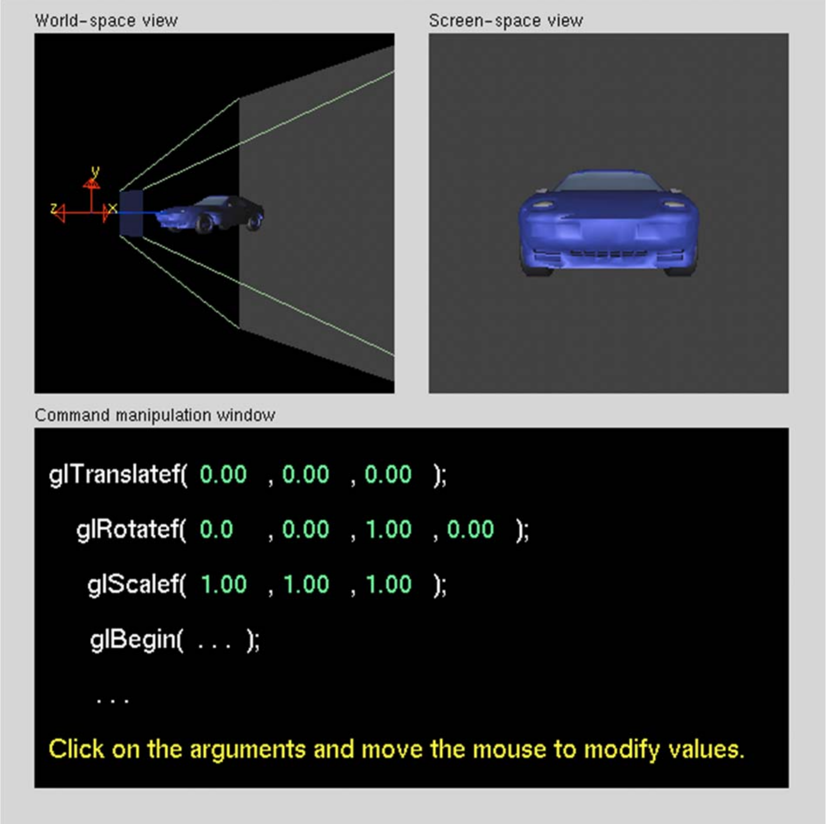
- In OpenGL, initially object and camera frames are the same
  - Default model-view matrix is an identity
- Camera located at origin and points in negative z direction
- OpenGL also specifies a default view volume that is a cube with sides of length 2 centered at origin
  - Default projection matrix is an identity

Default Projection

Default projection is orthogonal

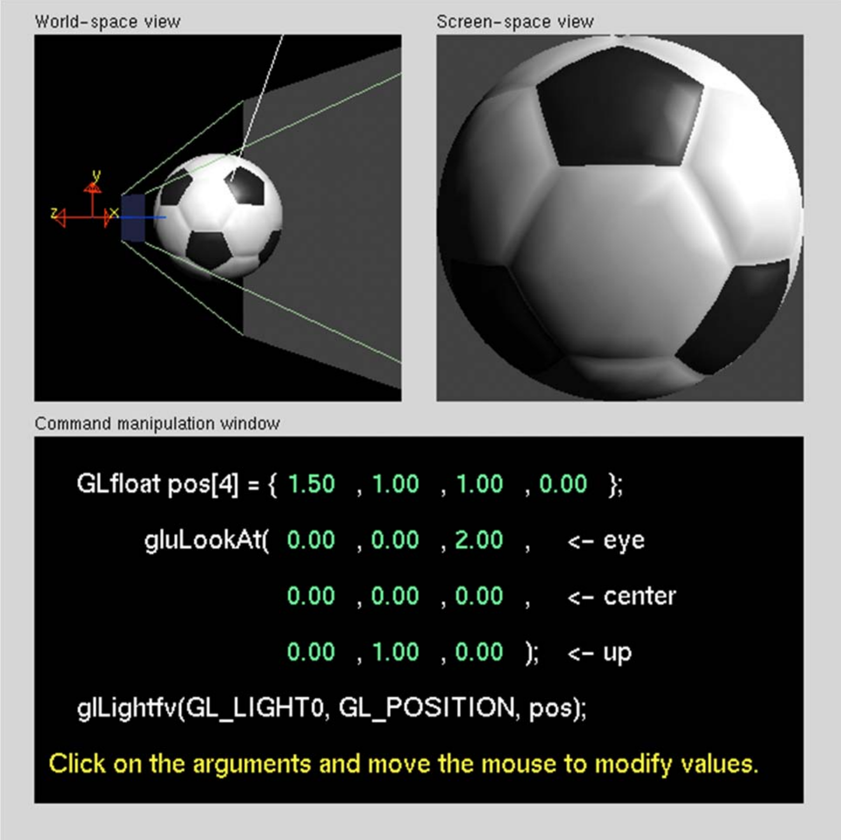


Nate Robbins LookAt Demo



```
glTranslatef( 0.00  , 0.00  , 0.00  );

   glRotatef( 0.0   , 0.00  , 1.00  , 0.00  );

    glScalef( 1.00  , 1.00  , 1.00  );

    glBegin( . . . );

     . . .
```

Click on the arguments and move the mouse to modify values.



```
GLfloat pos[4] = { 1.50  , 1.00  , 1.00  , 0.00  };

      gluLookAt(  0.00  , 0.00  , 2.00  ,    <- eye

                  0.00  , 0.00  , 0.00  ,    <- center

                  0.00  , 1.00  , 0.00  );   <- up

glLightfv(GL_LIGHT0, GL_POSITION, pos);
```

Click on the arguments and move the mouse to modify values.

Moving the Camera Frame

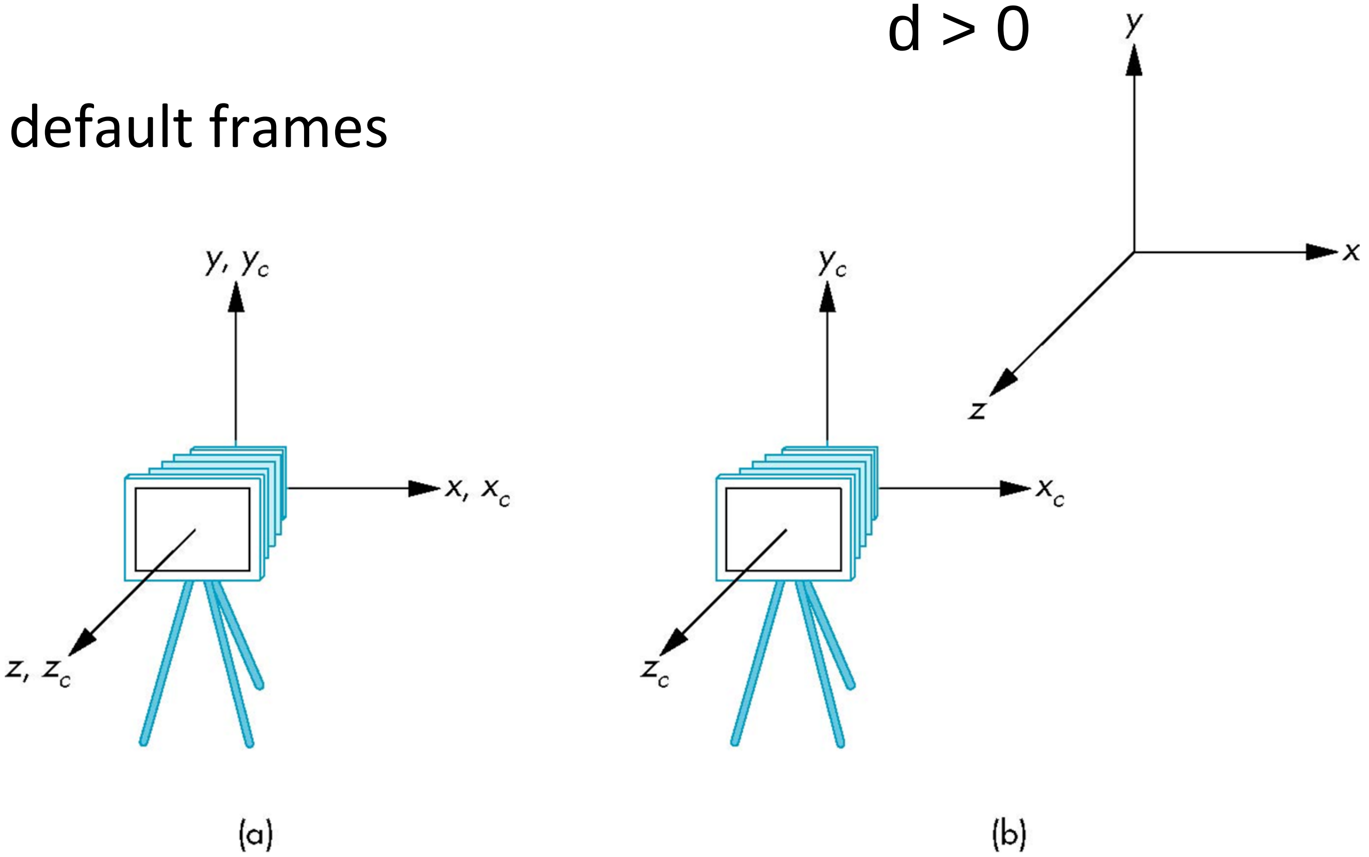
- If we want to visualize object with both positive and negative z values we can either
  - Move the camera in the positive z direction
    - Translate the camera frame
  - Move the objects in the negative z direction
    - Translate the world frame
- Both of these views are equivalent and are determined by the model-view matrix
  - Want a translation (`Translate(0.0,0.0,-d);`)
  - `d > 0`

Moving Camera back from Origin

frames after translation by –d

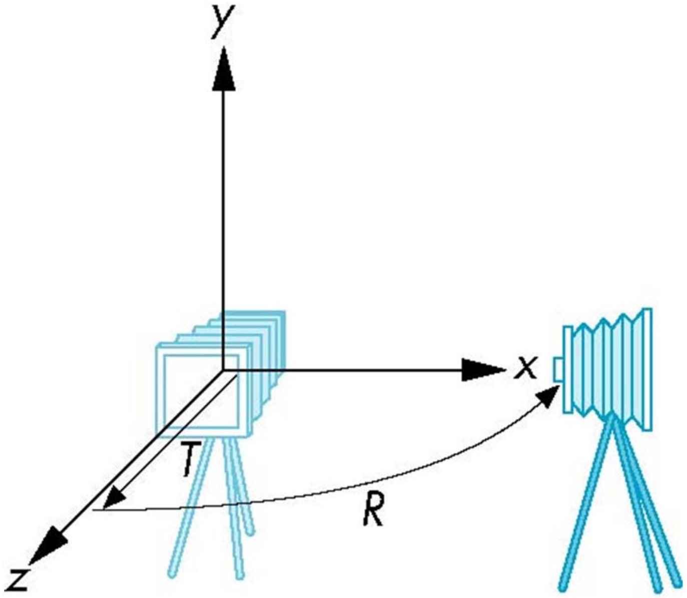
d > 0

default frames



Moving the Camera

- We can move the camera to any desired position by a sequence of rotations and translations
- Example: side view
  - Rotate the camera
  - Move it away from origin
  - Model-view matrix C = TR



OpenGL code

- Remember that last transformation specified is first to be applied

```
// Using mat.h

mat4 t = Translate (0.0, 0.0, -d);
mat4 ry = RotateY(90.0);
mat4 m = t*ry;
```

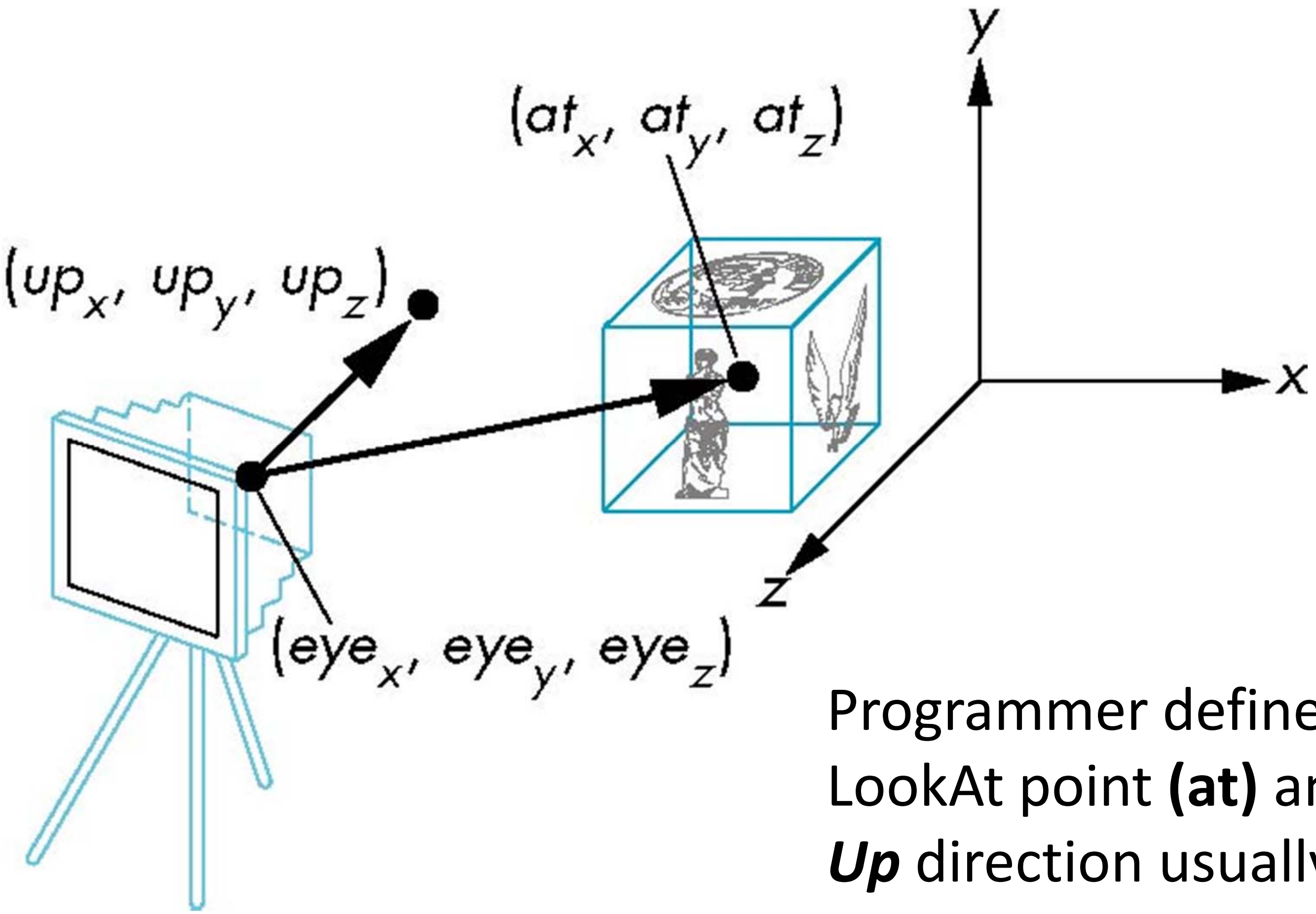
The LookAt Function

- The GLU library contained function gluLookAt to form required modelview matrix
- Note the need for setting an up direction
- Replaced by LookAt() in mat.h
  - Can concatenate with modeling transformations
- Example: isometric view of cube aligned with axes

```
void display( ){
    ………

    mat4 mv = LookAt(vec4 eye, vec4 at, vec4 up);
    ……..
}
```

gluLookAt

```
LookAt(eye, at, up)
```



Programmer defines: **eye** position
LookAt point **(at)** and **Up** vector
Up direction usually set to (0,1,0)

References

- Angel and Shreiner
- Hill and Kelley, appendix 4