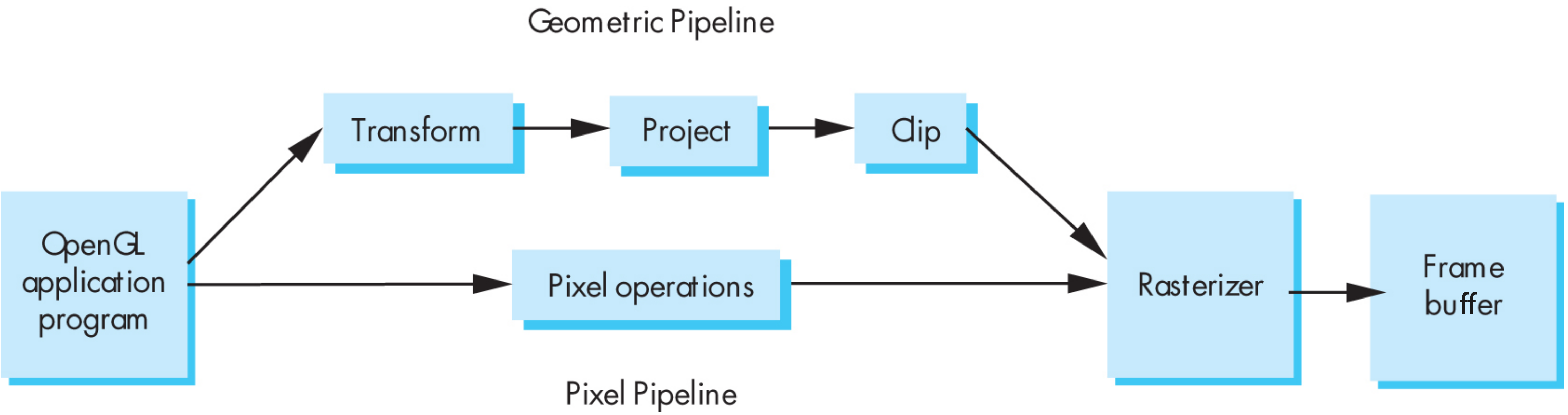
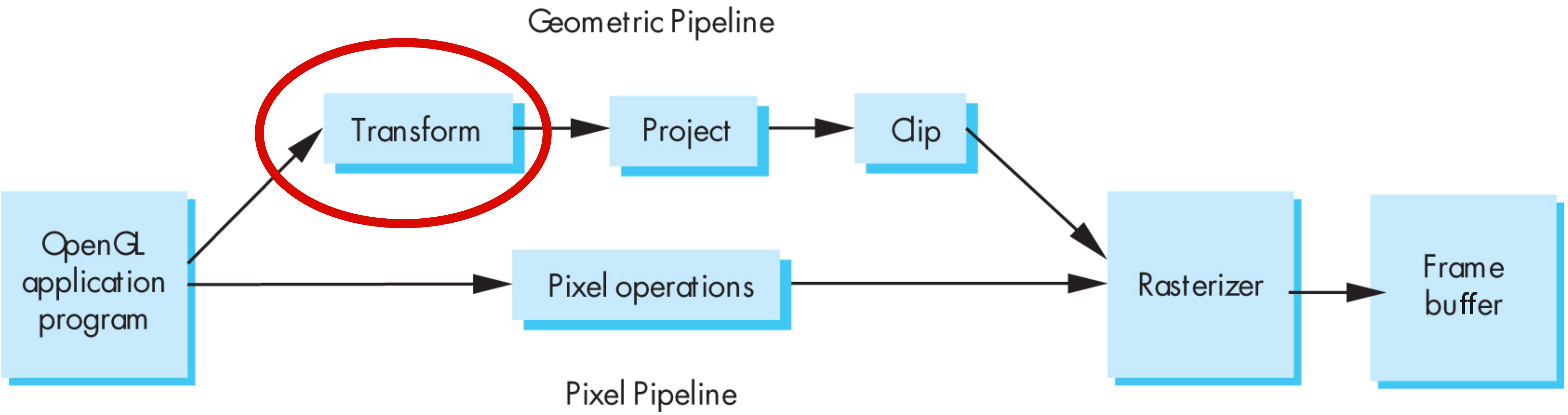
Graphics Pipeline (cont.)

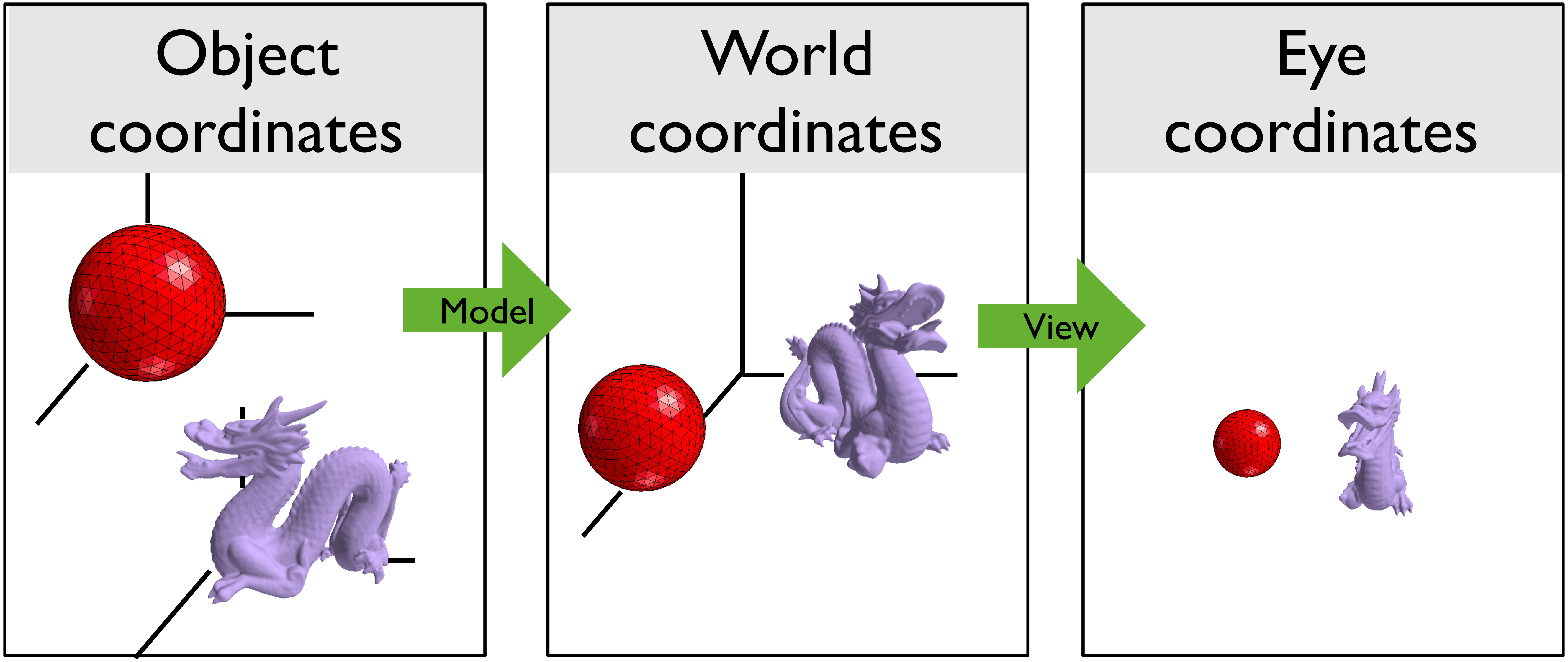
Graphics Pipeline



Transform


Geometric Pipeline
Transform
Project
Clip
OpenGL application program
Pixel operations
Rasterizer
Frame buffer
Pixel Pipeline

“Modelview” Transformation



Object coordinates

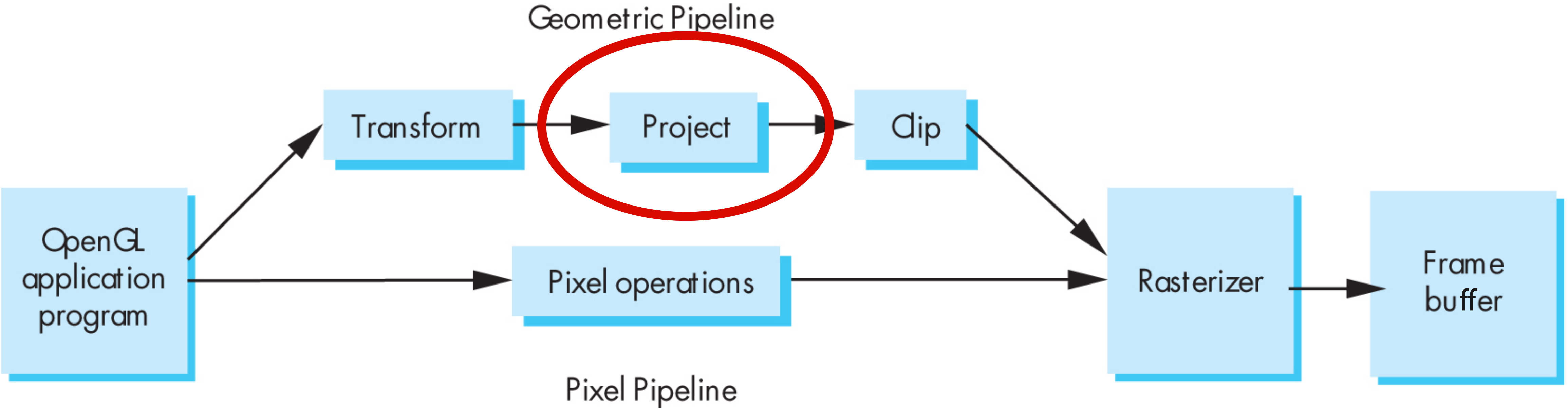
Model

World coordinates

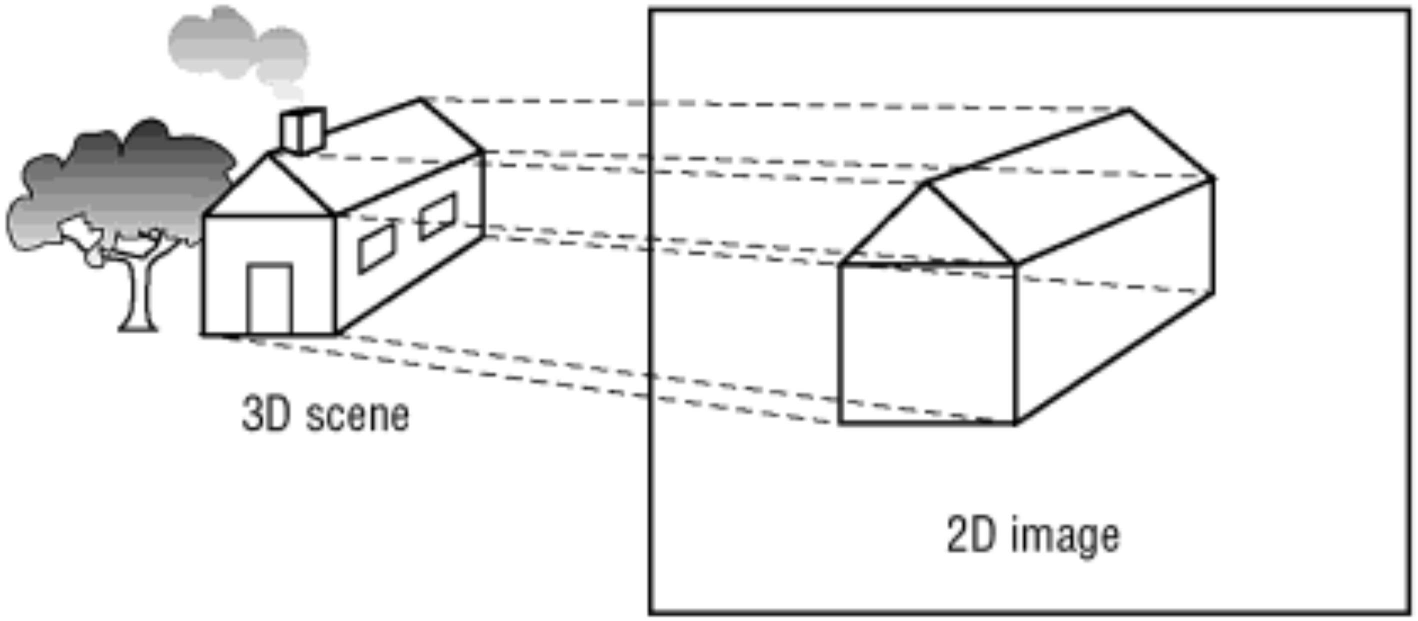
View

Eye coordinates

Project

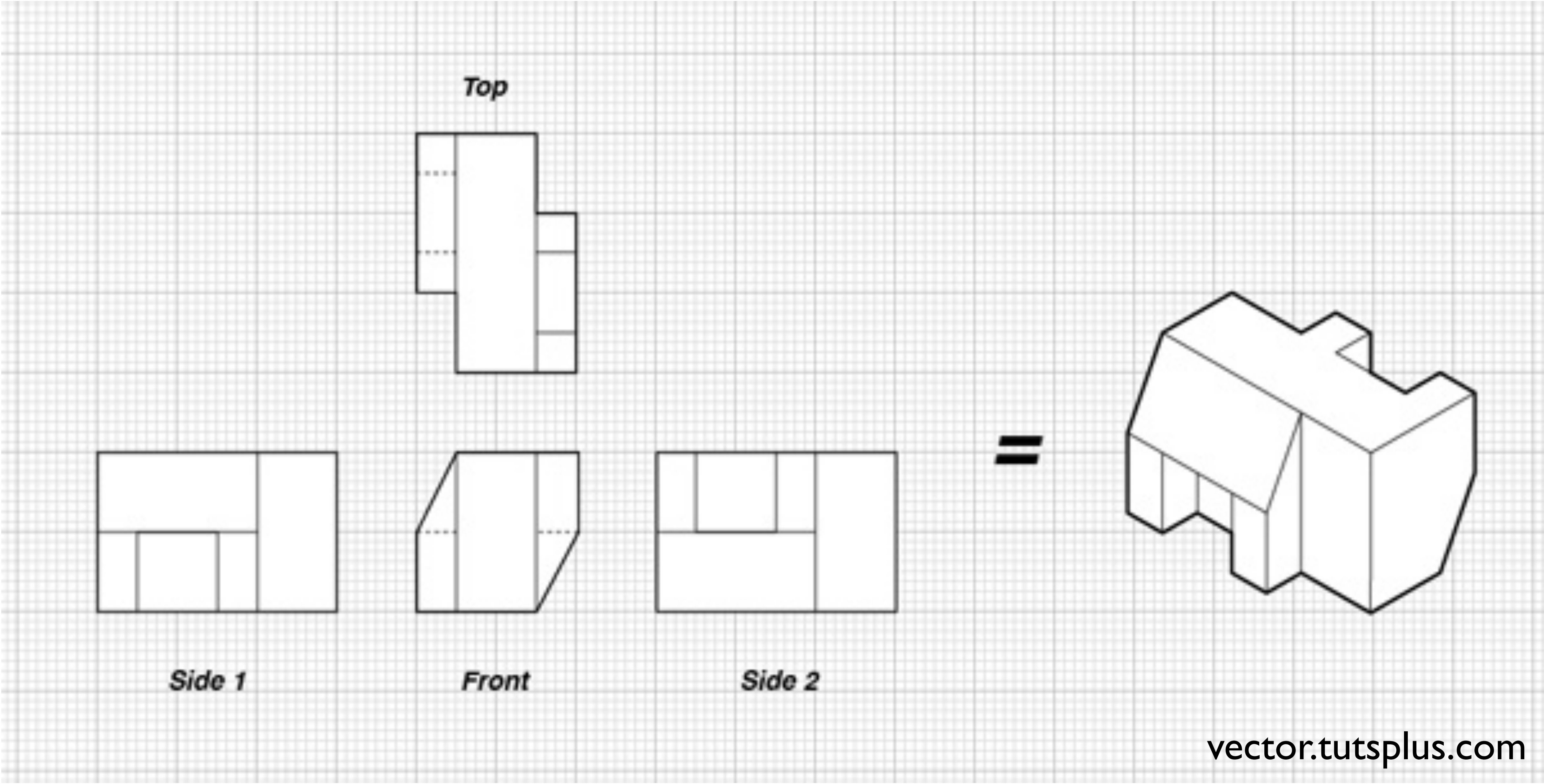

Geometric Pipeline
Transform
Project
Clip
OpenGL application program
Pixel operations
Rasterizer
Frame buffer
Pixel Pipeline

Projection: map 3D scene to 2D image



OpenGL Super Bible, 5th Ed.

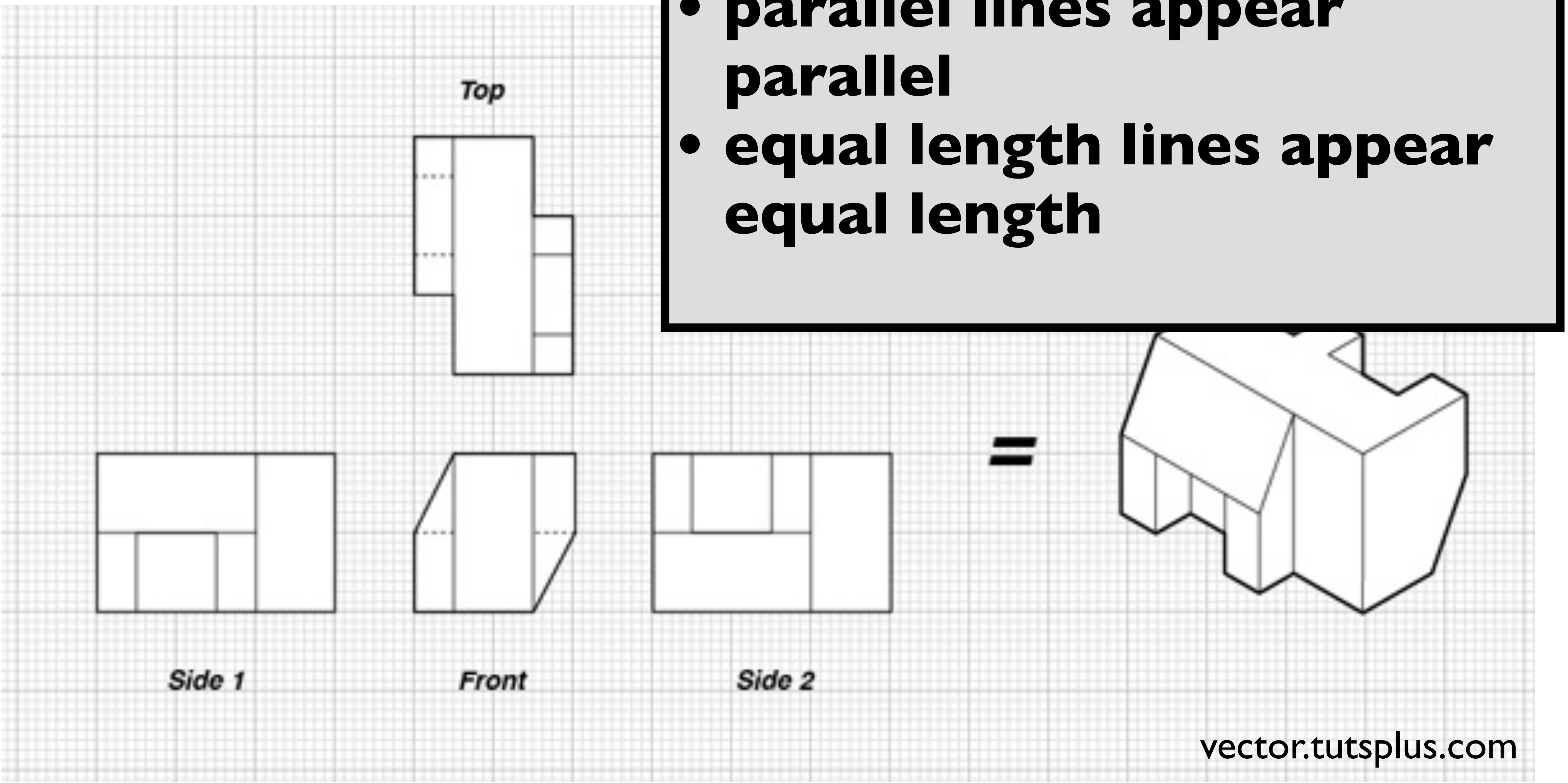
Orthographic projection



vector.tutsplus.com

Orthographic projection

- **parallel lines appear parallel**
- **equal length lines appear equal length**

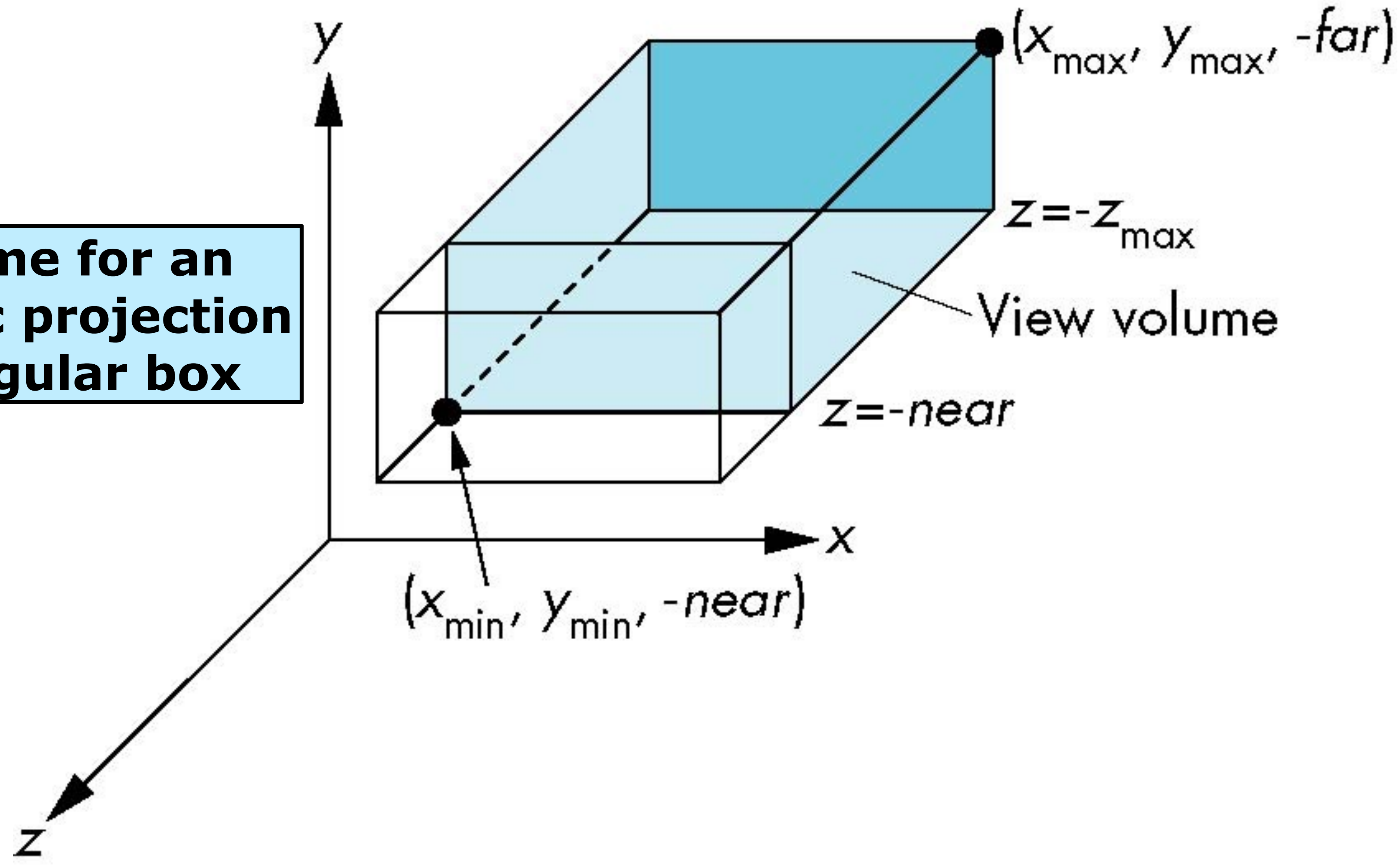


vector.tutsplus.com

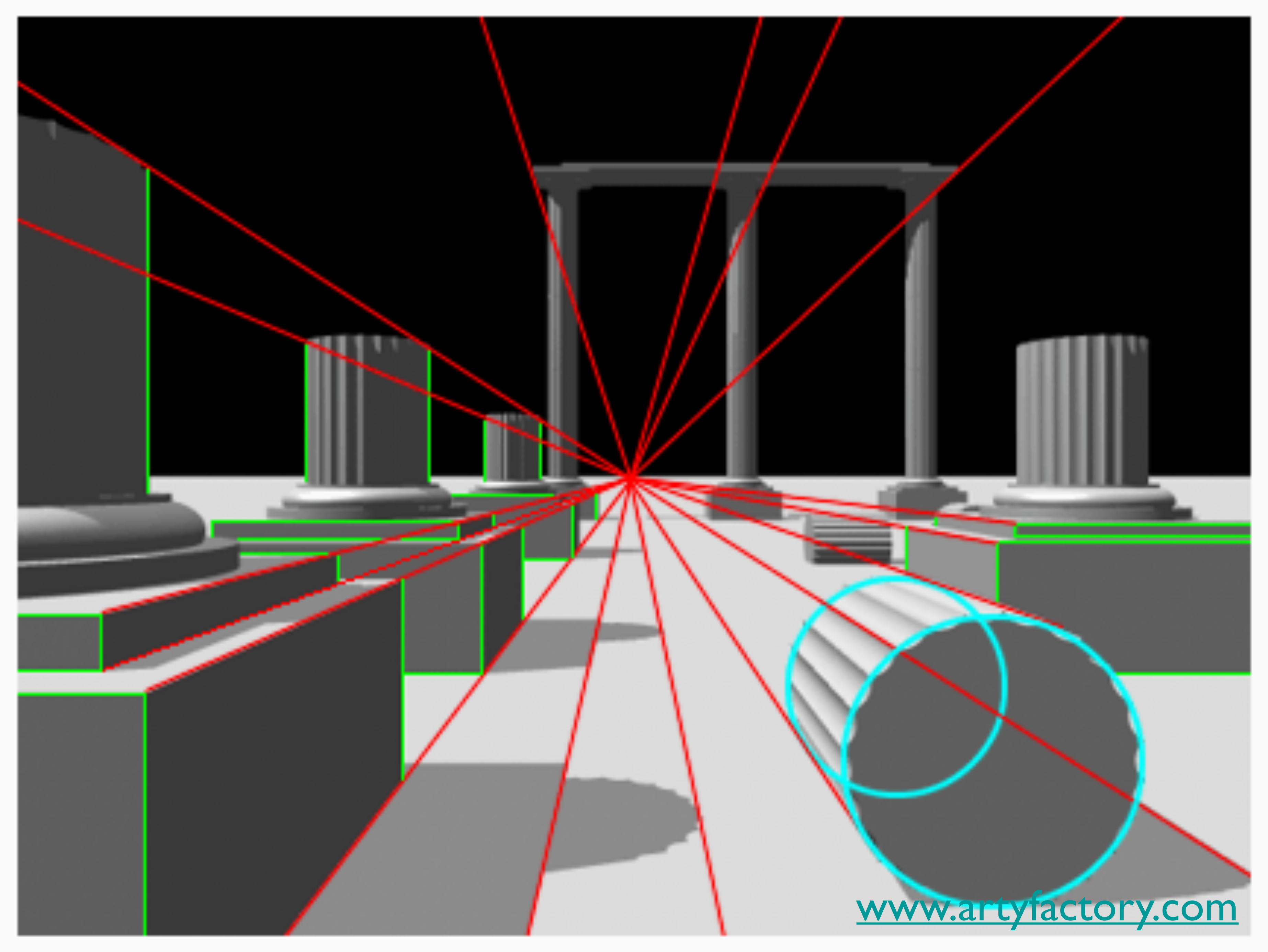
OpenGL Orthogonal Viewing

```
glOrtho(left,right,bottom,top,near,far)
```

View volume for an orthographic projection is a rectangular box



Perspective projection

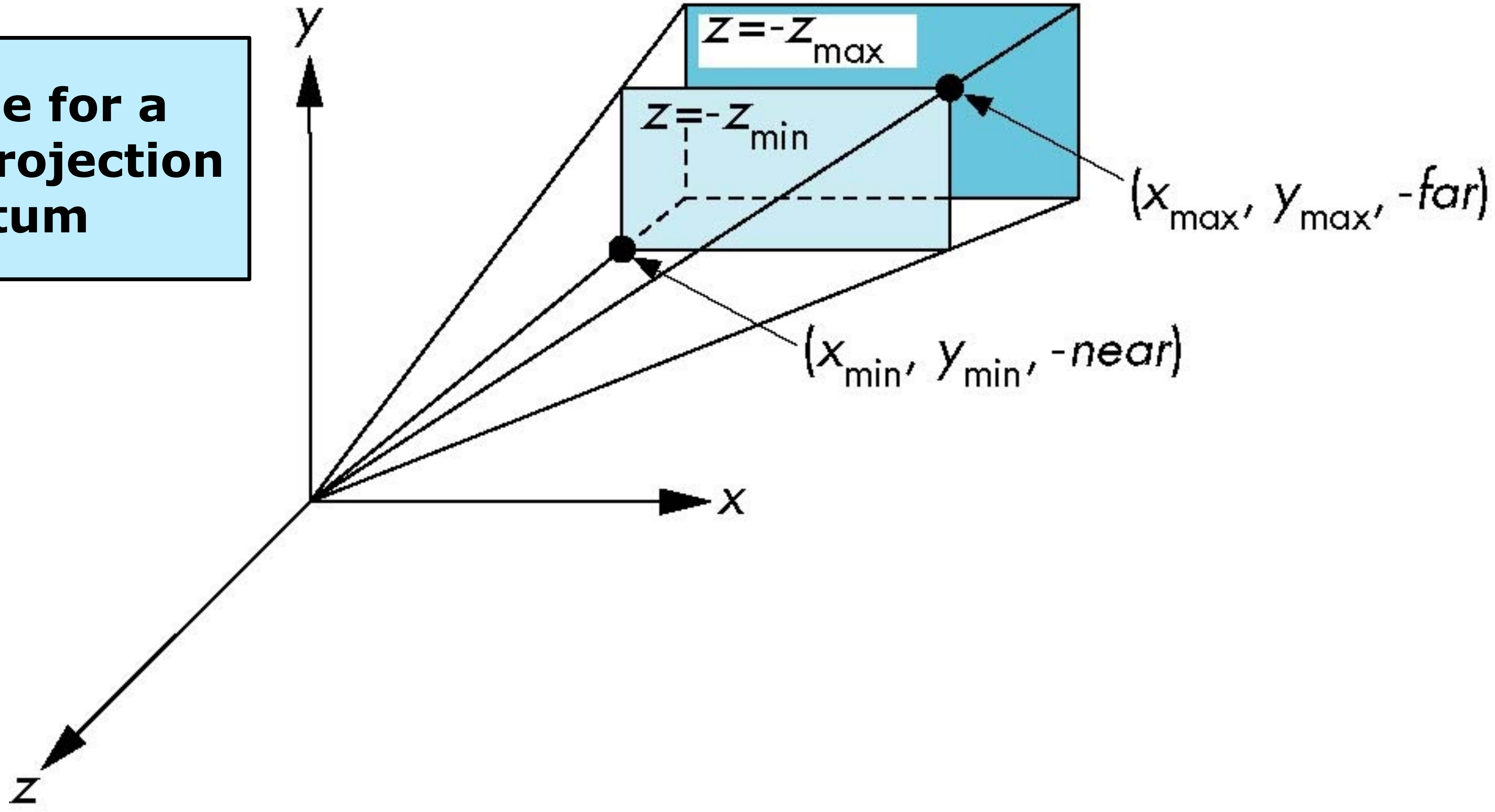


www.artyfactory.com

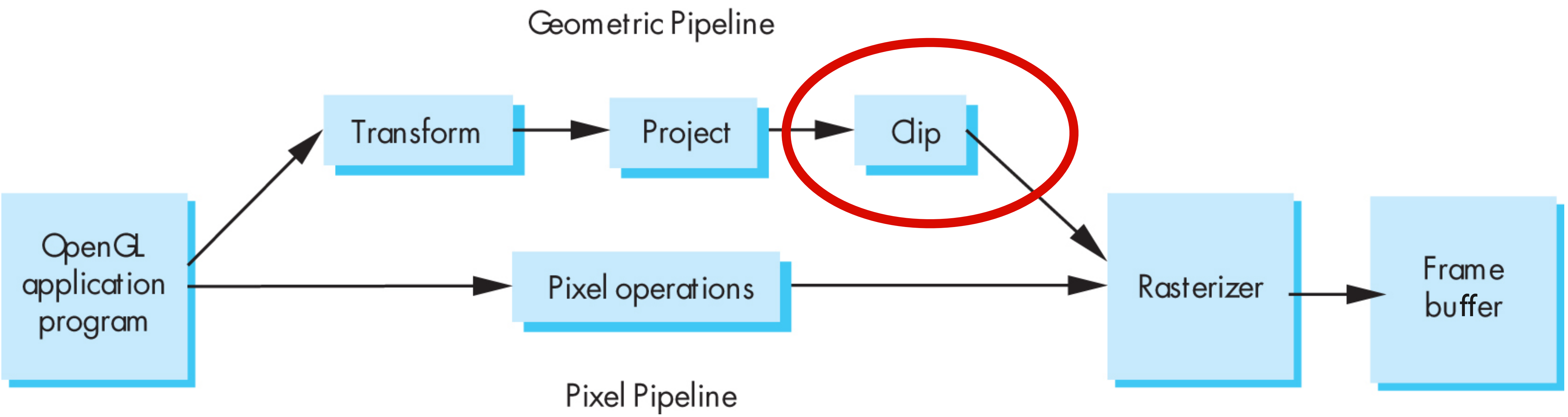
OpenGL Perspective Viewing

```
glFrustum(xmin,xmax,ymin,ymax,near,far)
```

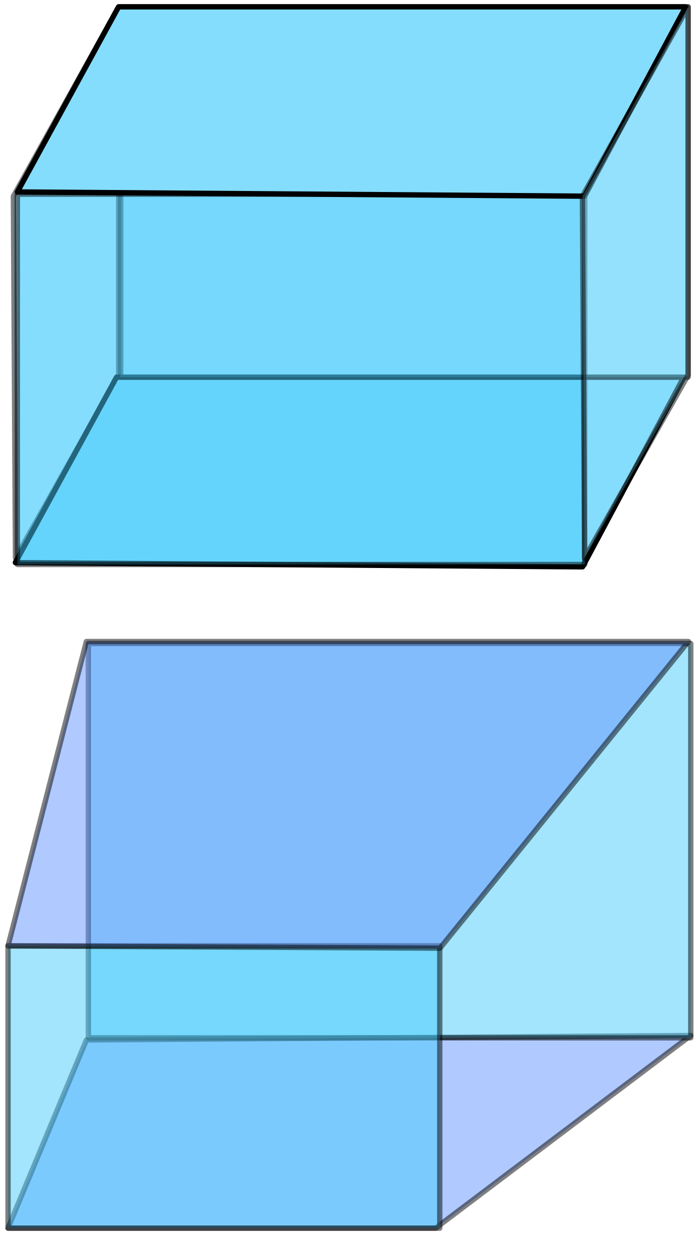
View volume for a perspective projection is a frustum

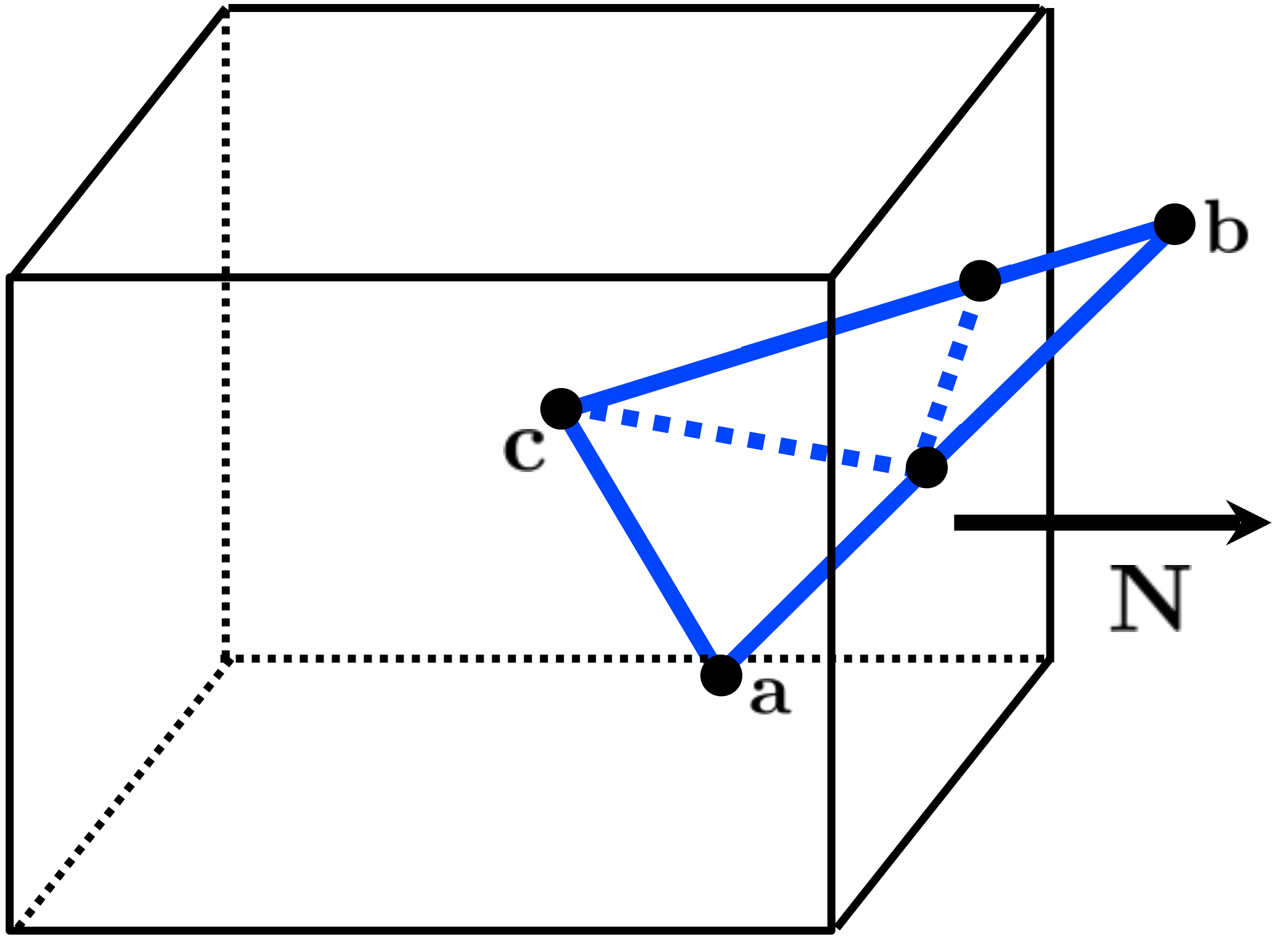


Clip

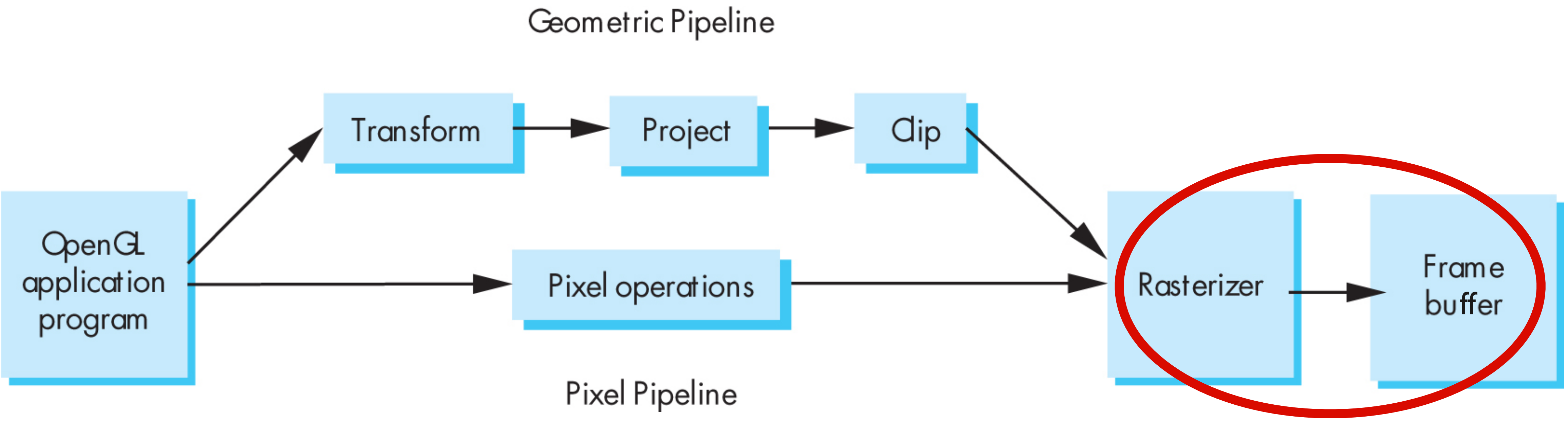

Geometric Pipeline
Transform
Project
Clip
OpenGL application program
Pixel operations
Rasterizer
Frame buffer
Pixel Pipeline

Clip against view volume

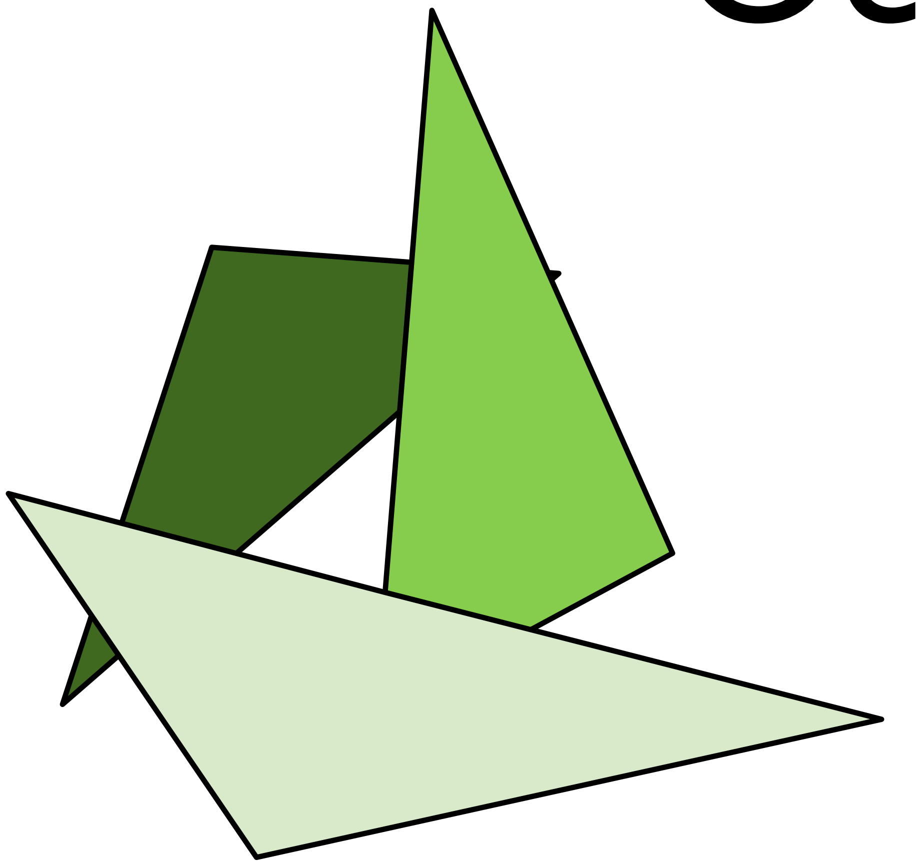




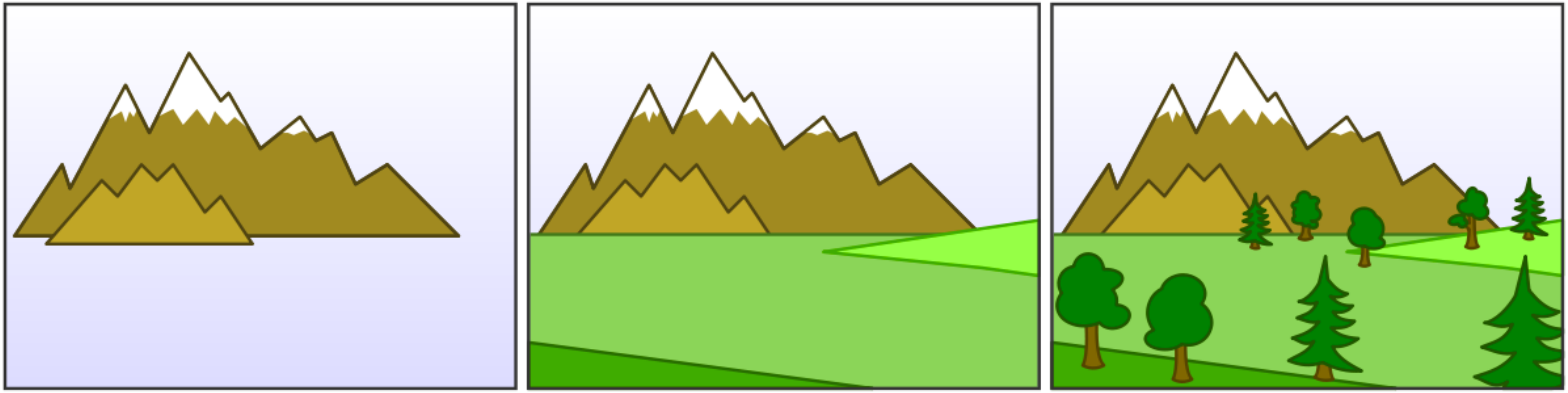
Hidden Surface Removal



Occlusion

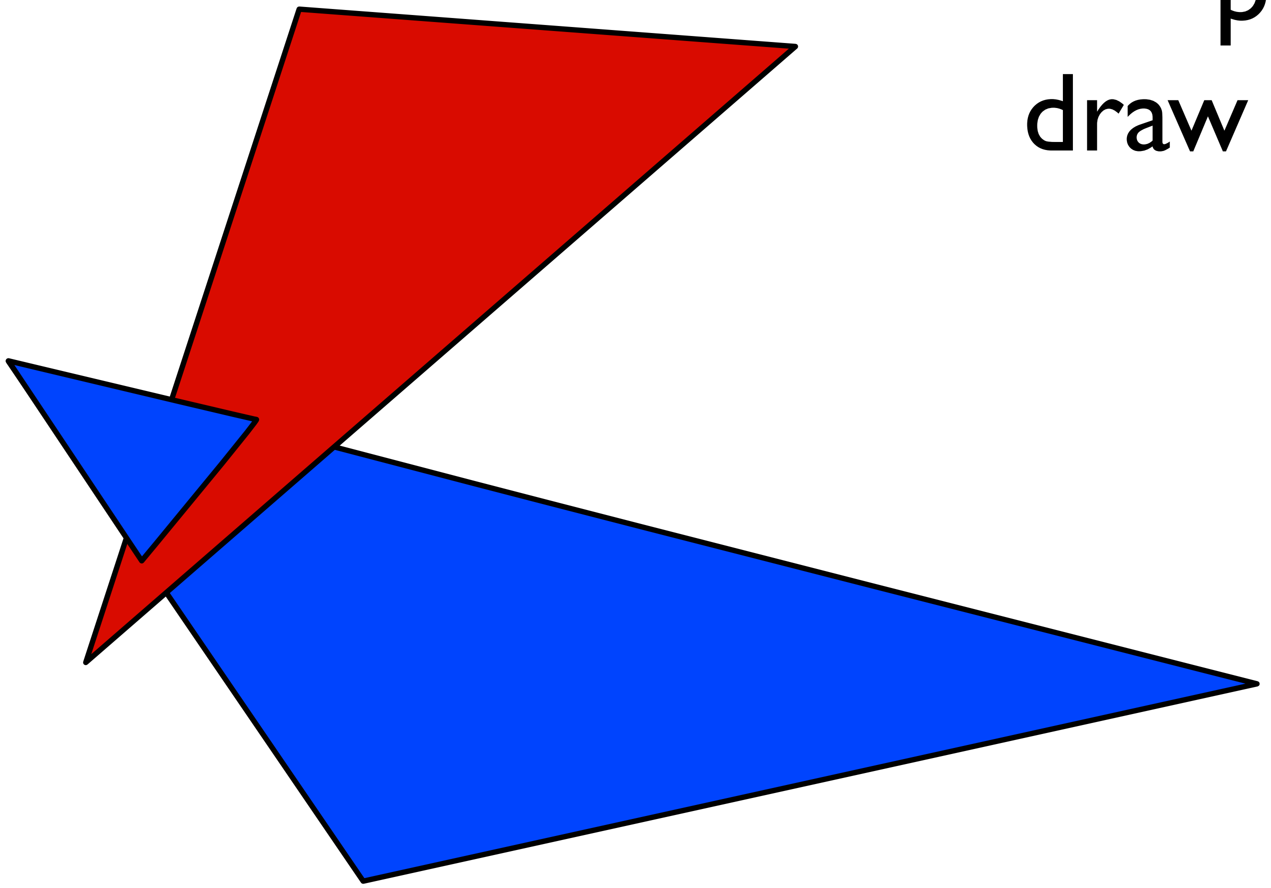


“painter’s algorithm”
draw primitives in
back-to-front order



[Wikimedia Commons]

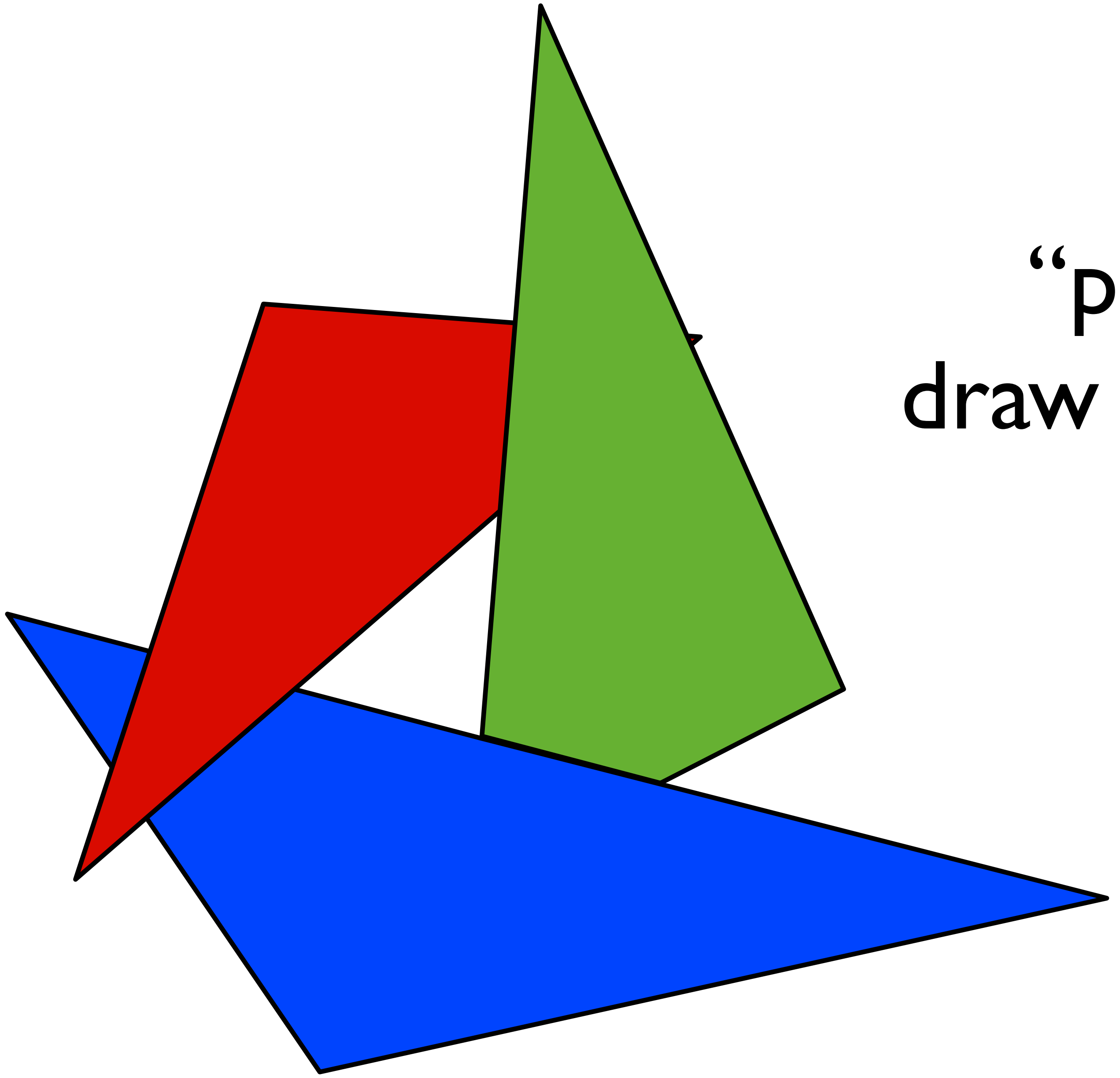
Occlusion



“painter’s algorithm”
draw primitives in back-to-front order

problem:
triangle intersection

Occlusion



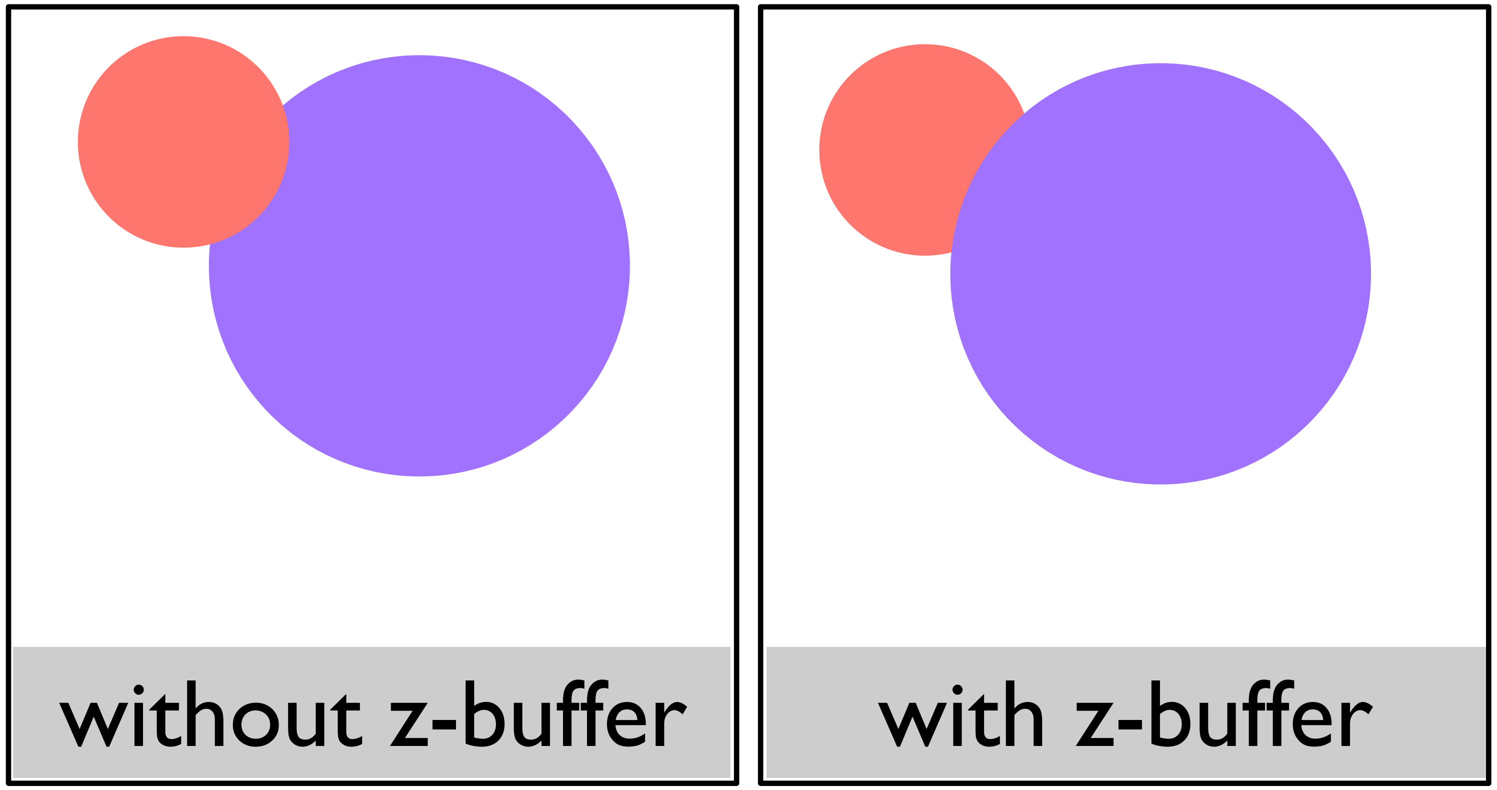
“painter’s algorithm” draw primitives in back-to-front order

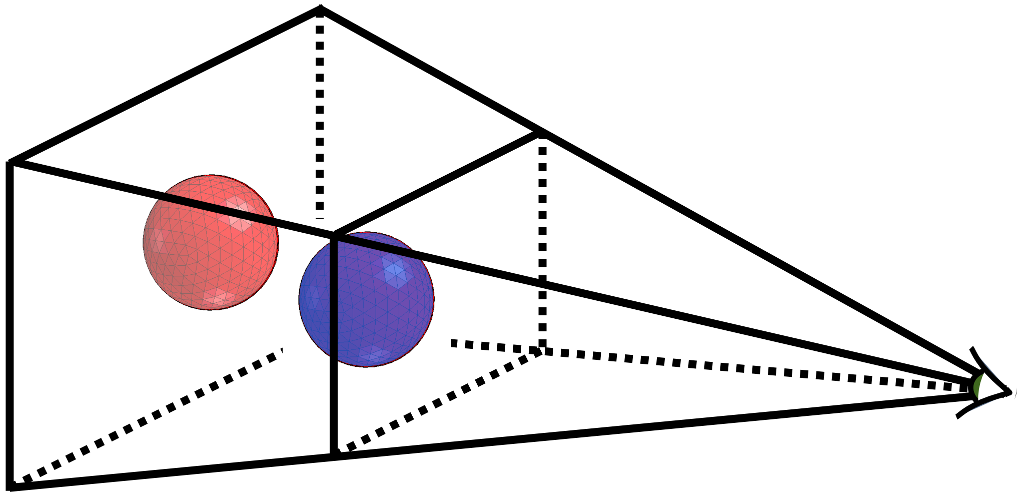
problem: occlusion cycle

Use a *z-buffer* for hidden surface removal

test depth on a pixel by pixel basis

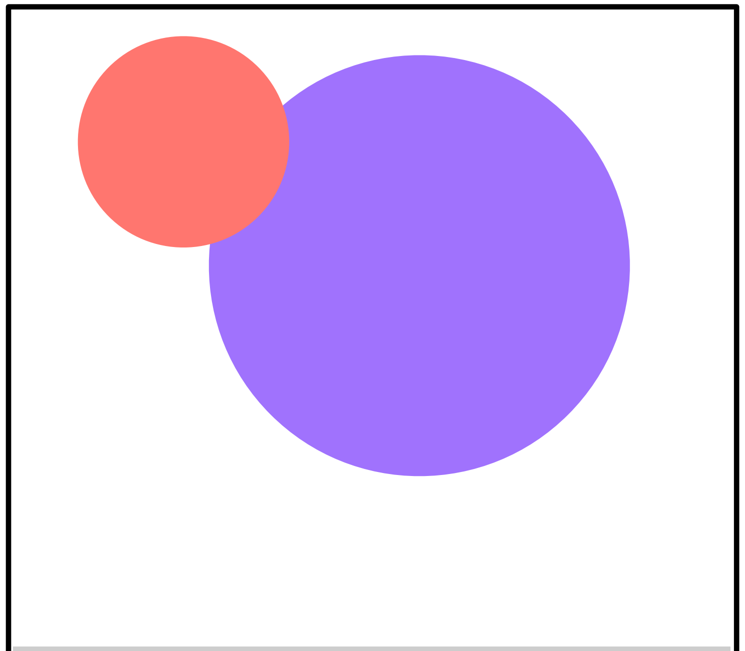
red drawn last



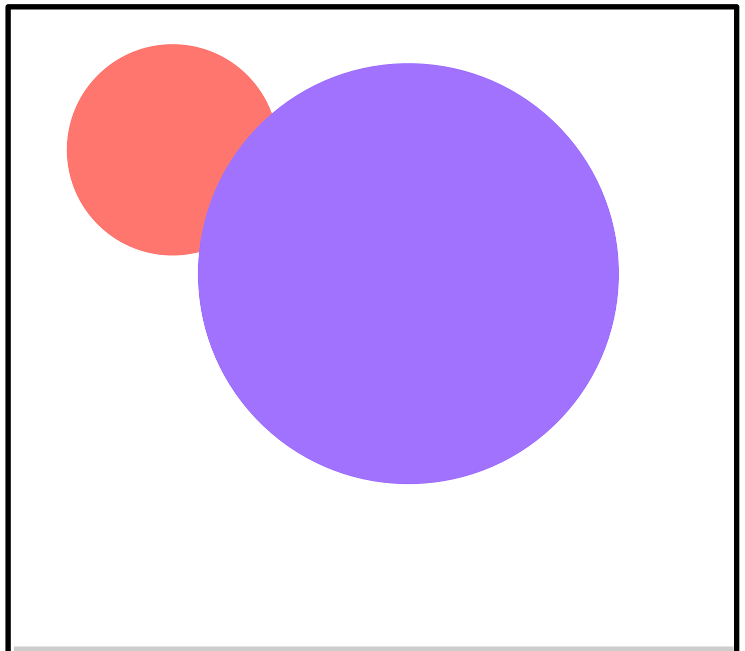


Use a *z-buffer* for hidden surface removal

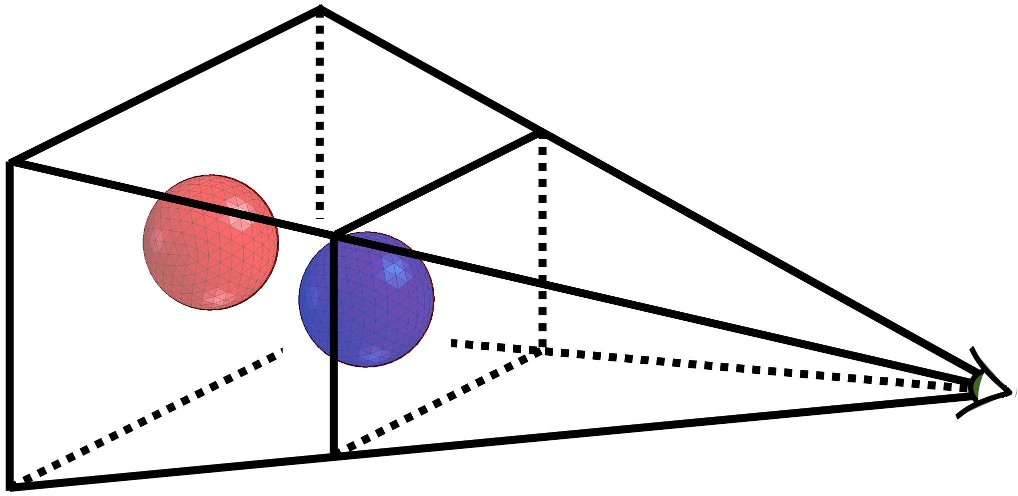
at each pixel, record distance to the closest object that has been drawn in a *depth* buffer



without z-buffer



with z-buffer



Use a *z-buffer* for hidden surface removal

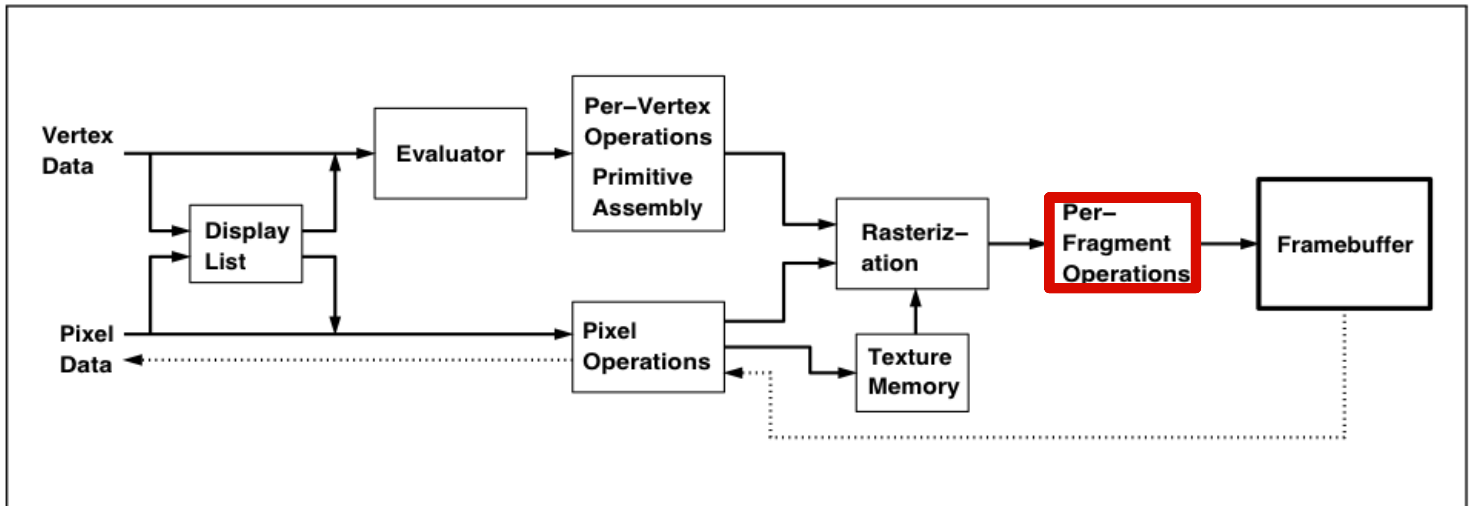
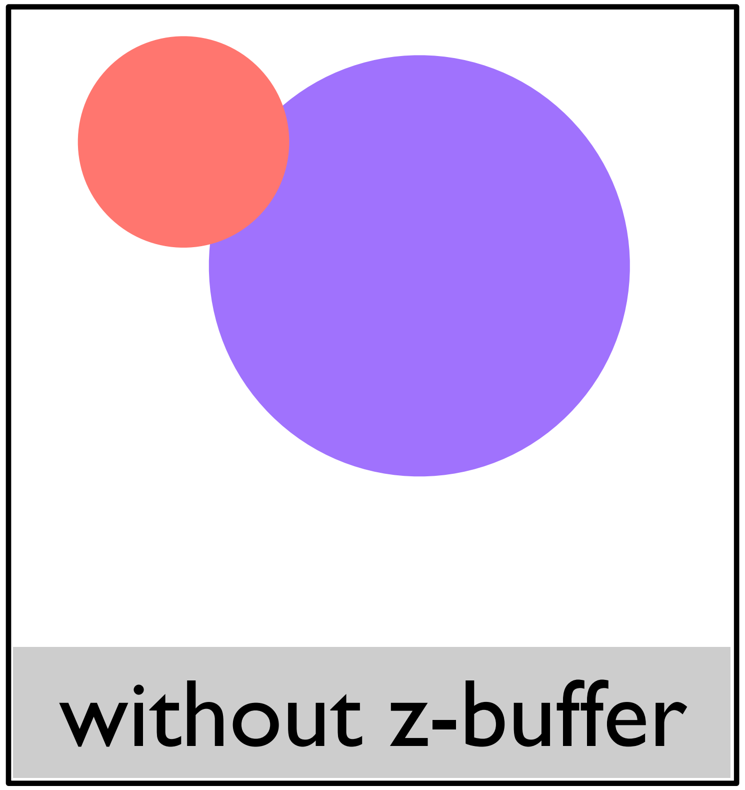
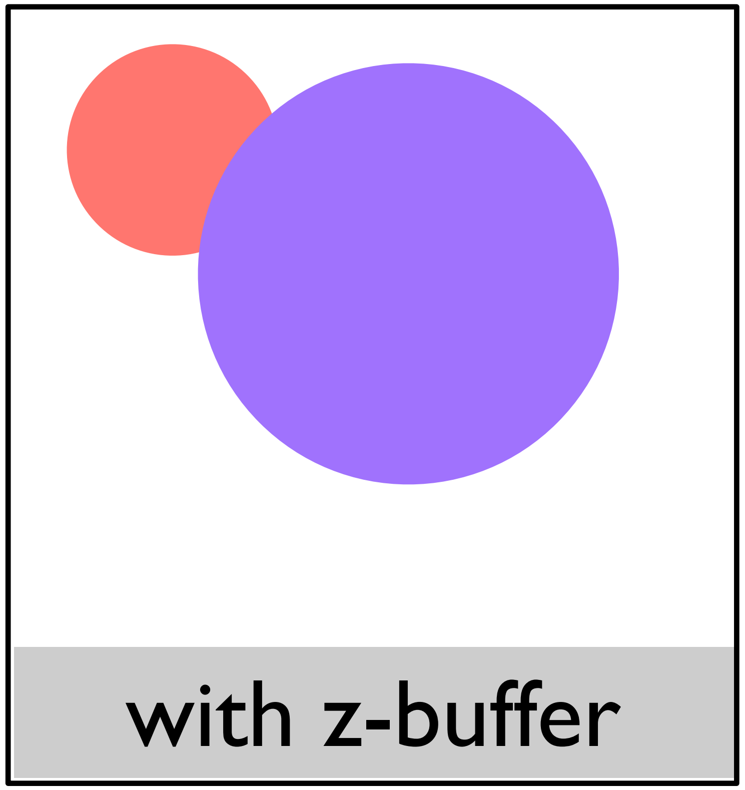


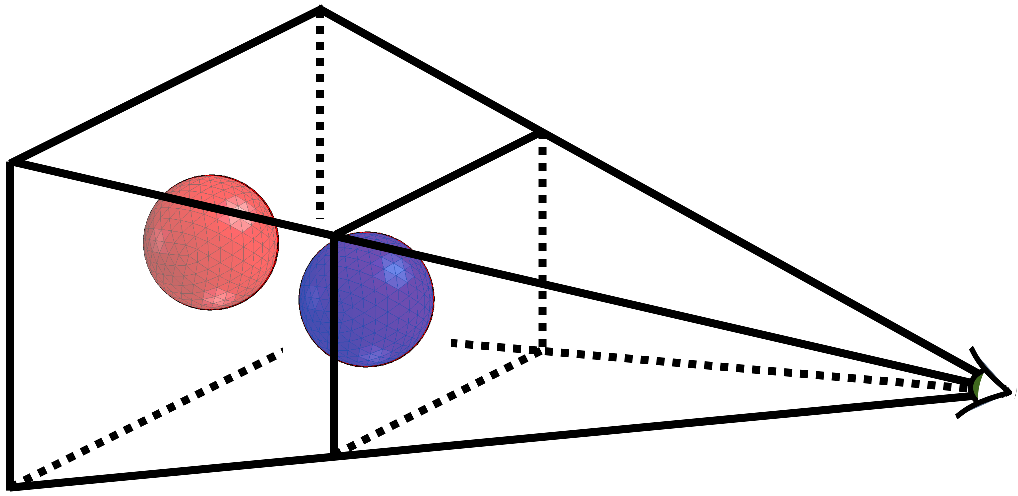
Figure 1. Block diagram of OpenGL.



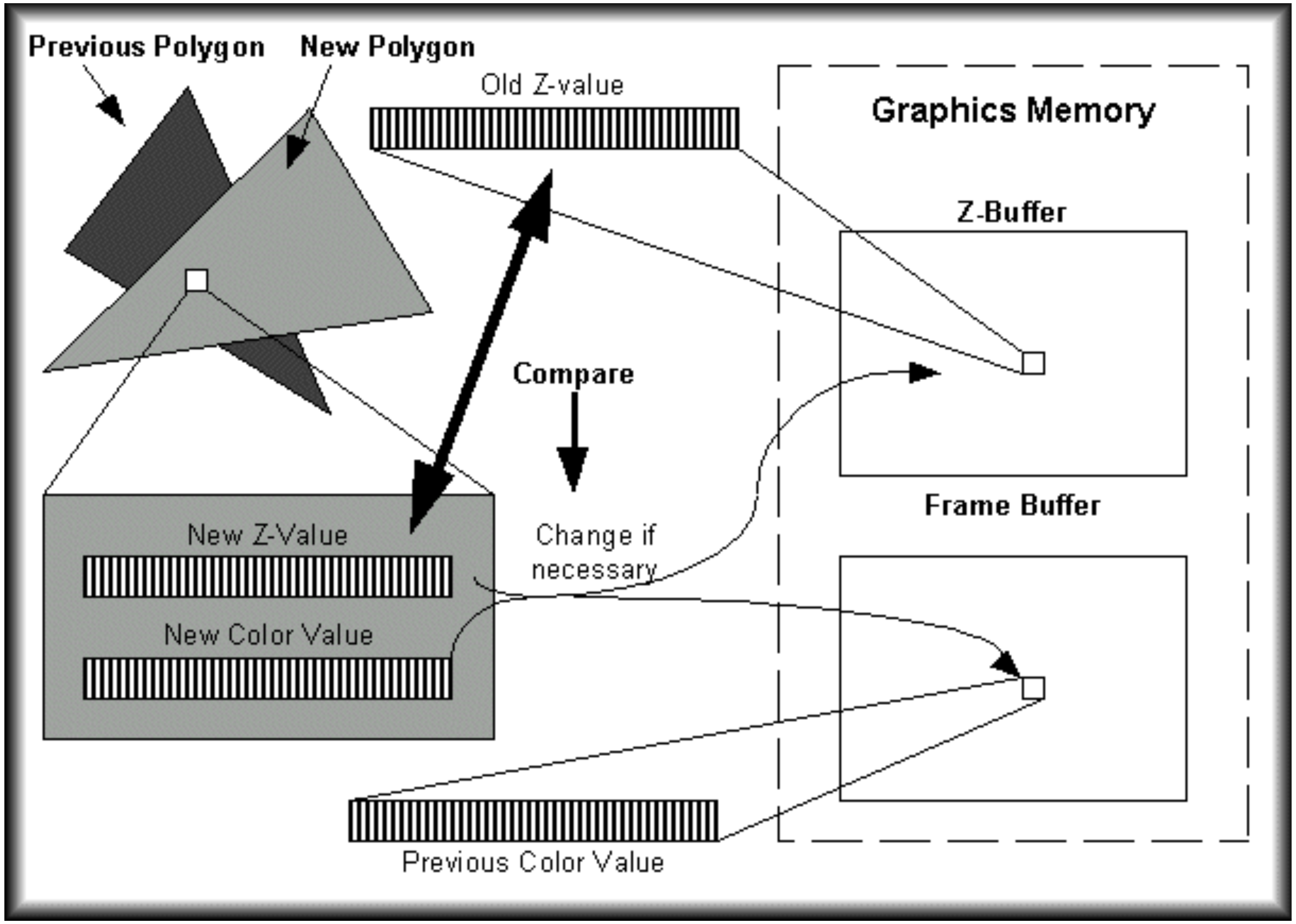
without z-buffer



with z-buffer

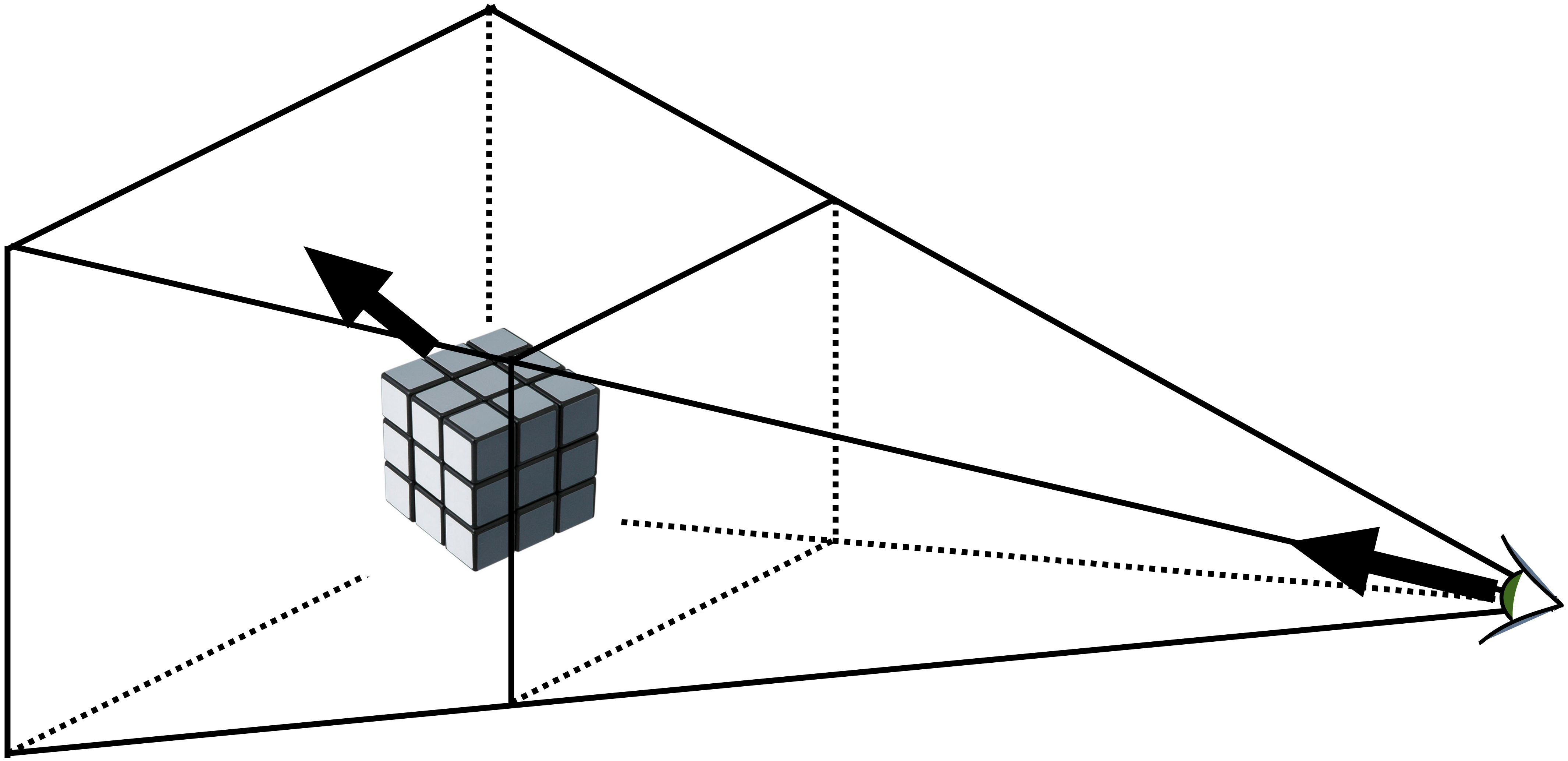


Use a *z-buffer* for hidden surface removal



http://www.beyond3d.com/content/articles/41/

Backface culling: another way to eliminate hidden geometry



Hidden Surface Removal in OpenGL

```
glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);

glEnable(GL_DEPTH_TEST);

glEnable(GL_CULL_FACE);
```

For a perspective transformation, there is more precision in the depth buffer for z-values closer to the near plane