CS 4204 Computer Graphics

GLUT (Continue)

More Interactions

Yong Cao

Virginia Tech

References:

Interactive Computer Graphics, Fourth Edition, Ed Angle

Orthographical Projection

Synthetic camera model

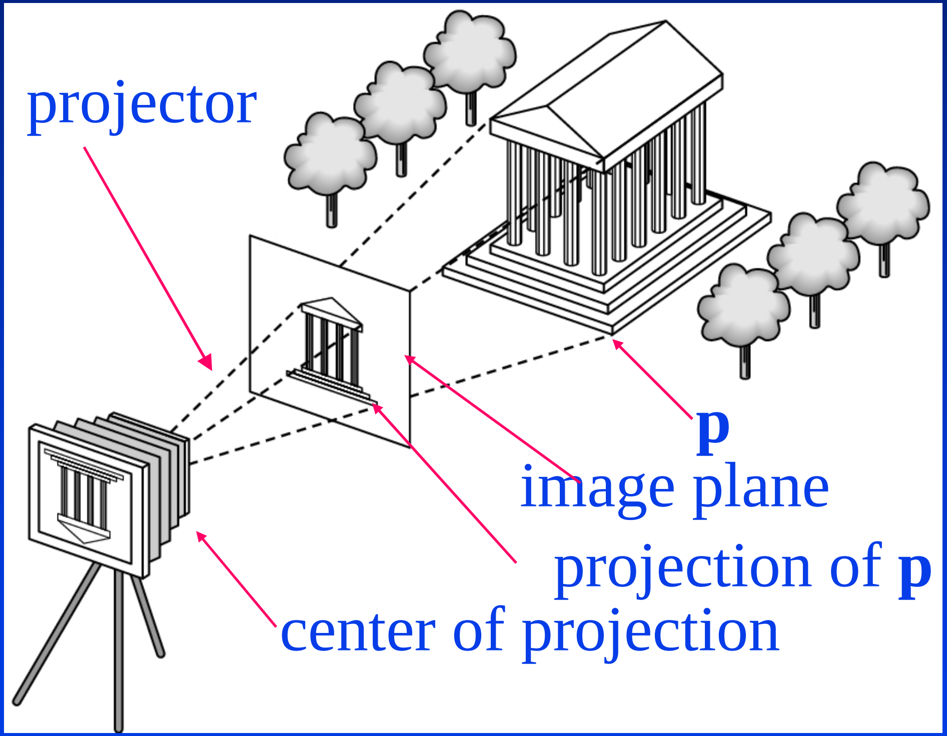
View angle and view volume

Clipping

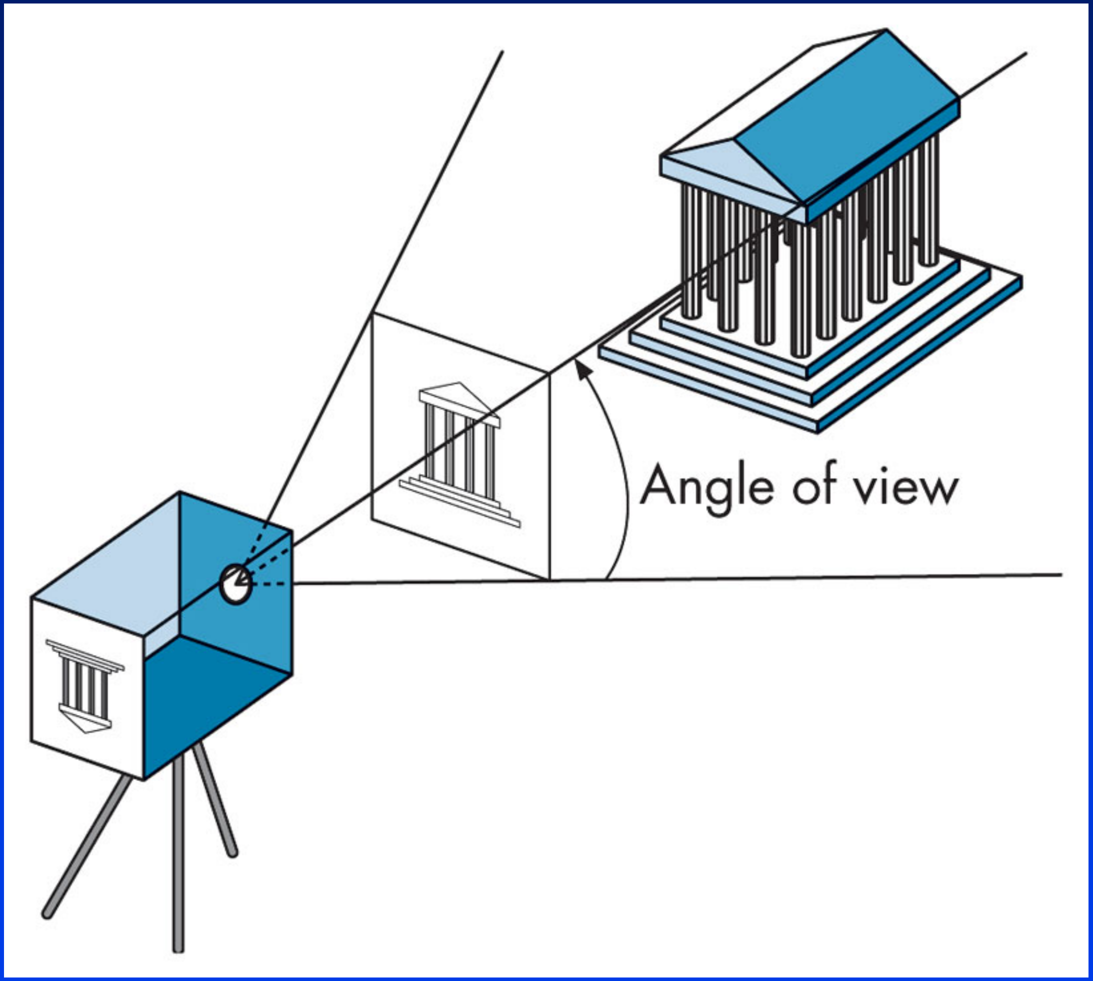
Aspect ratio

Viewport mapping

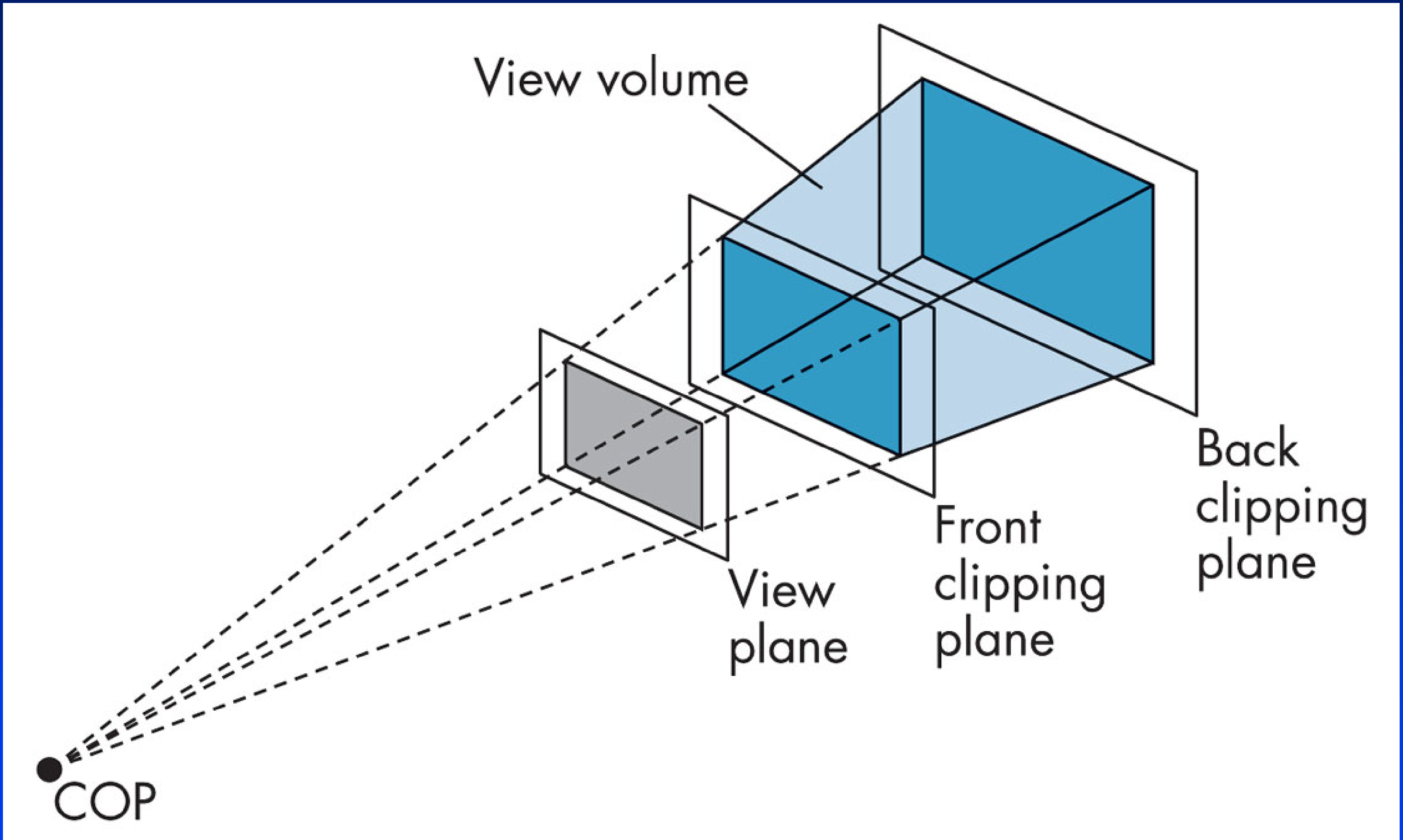
Synthetic Camera Model



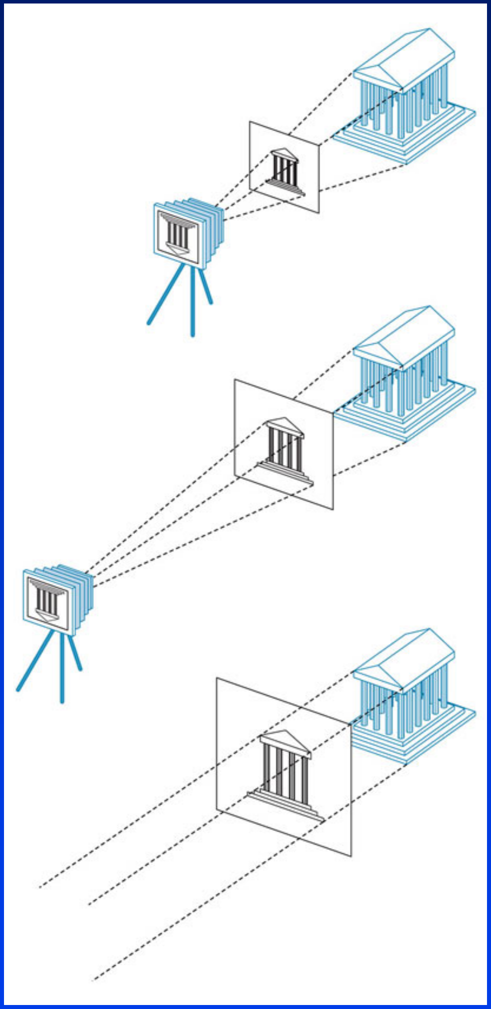
View angle



View volume

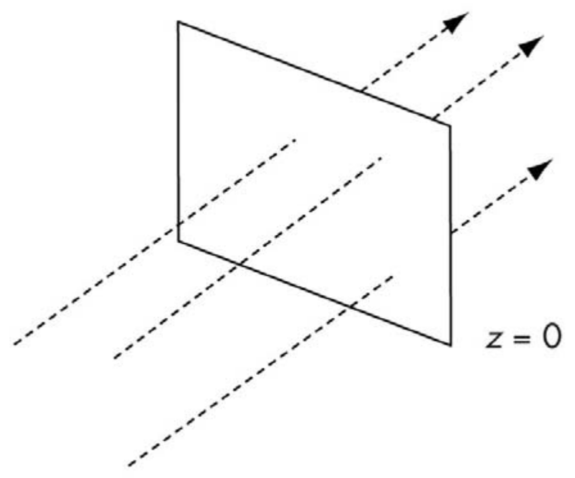

View volume
Back clipping plane
Front clipping plane
View plane
COP

Orthographical projection

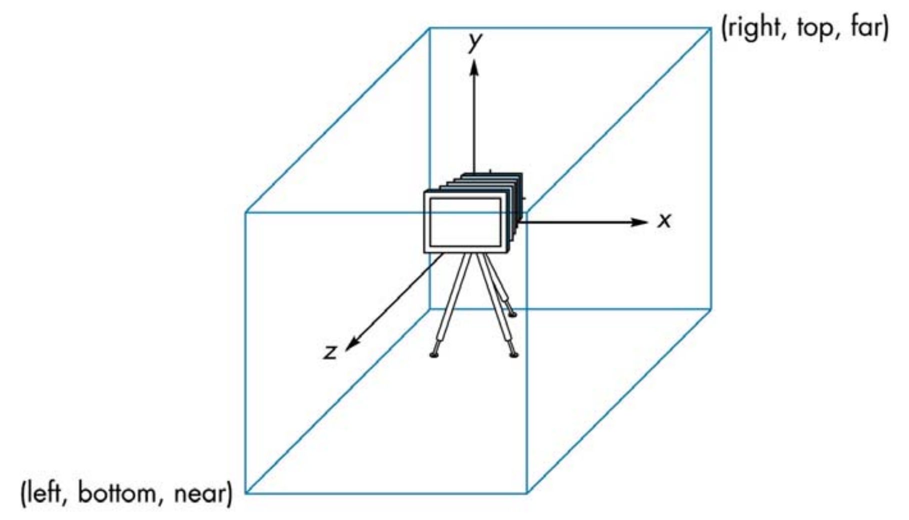


Creating an orthographic view by moving the camera away from the projection plane.

Orthographical projection

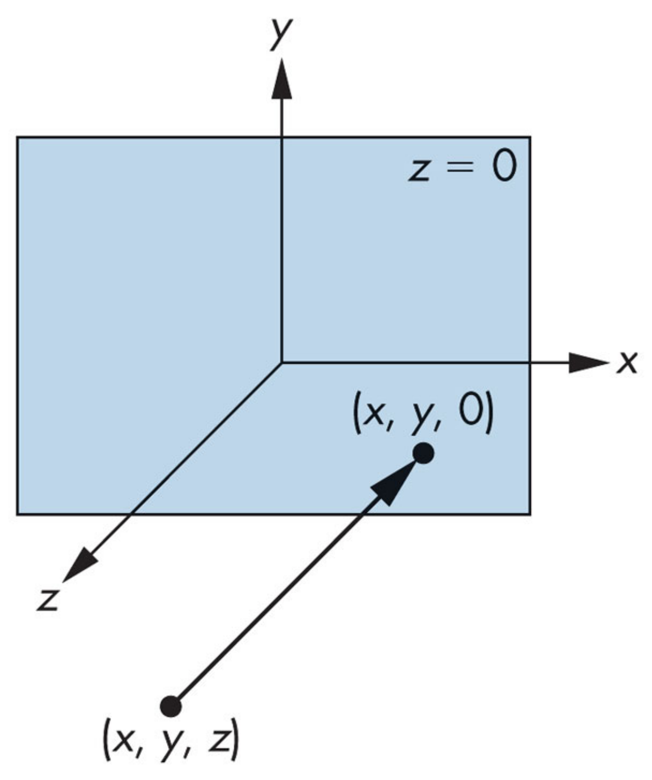


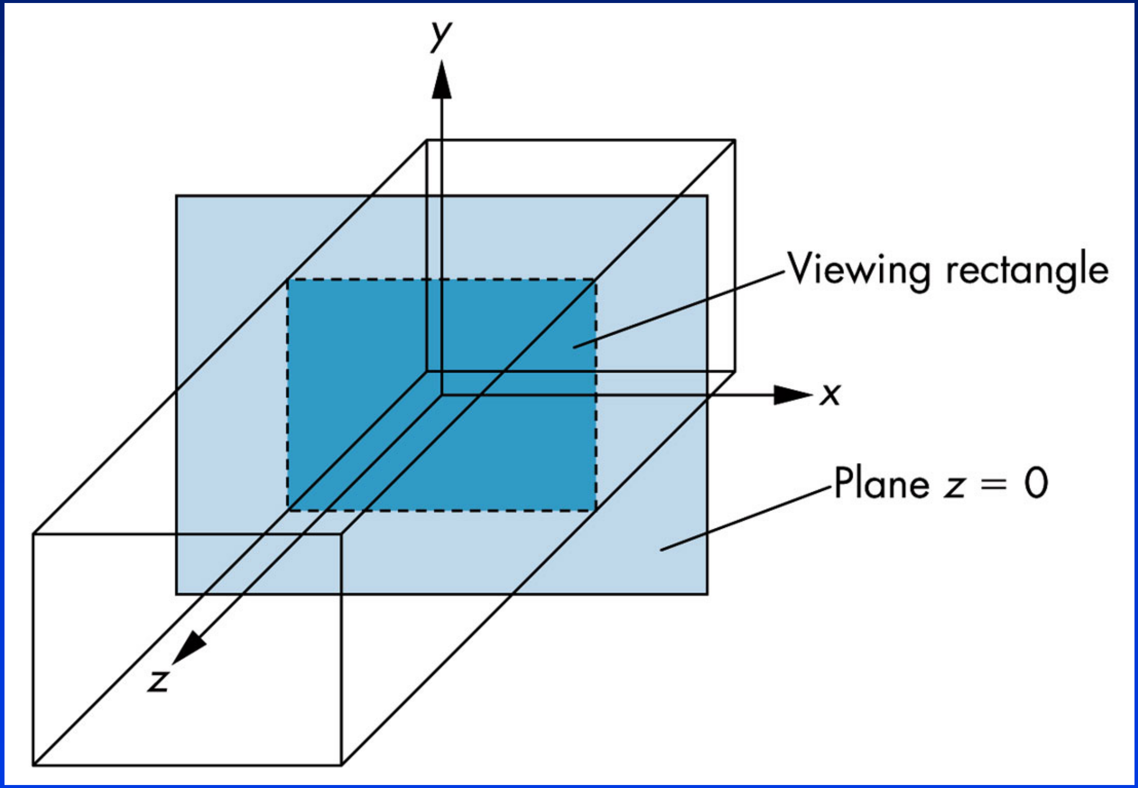
Orthographic projectors with projection plane z = 0.



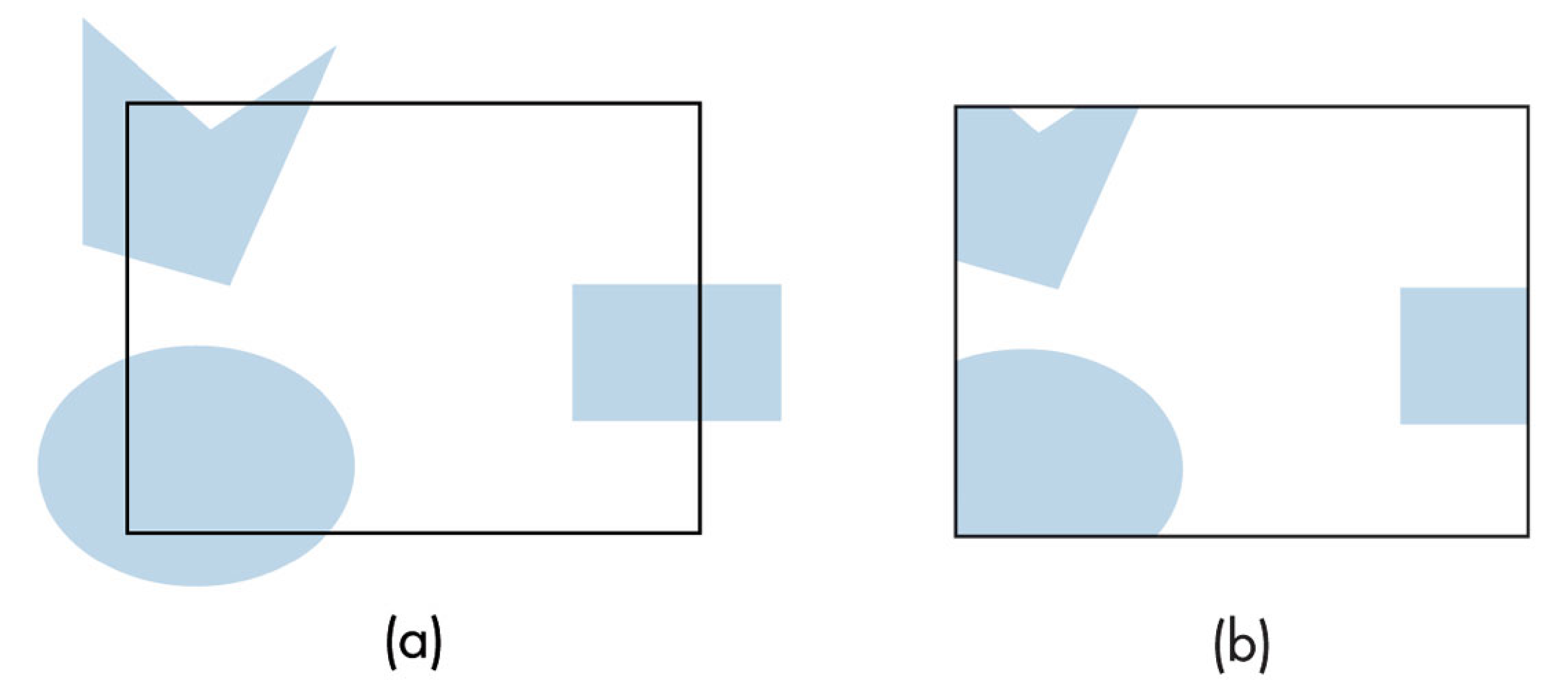
The default camera and an orthographic view volume.

Orthographical projection





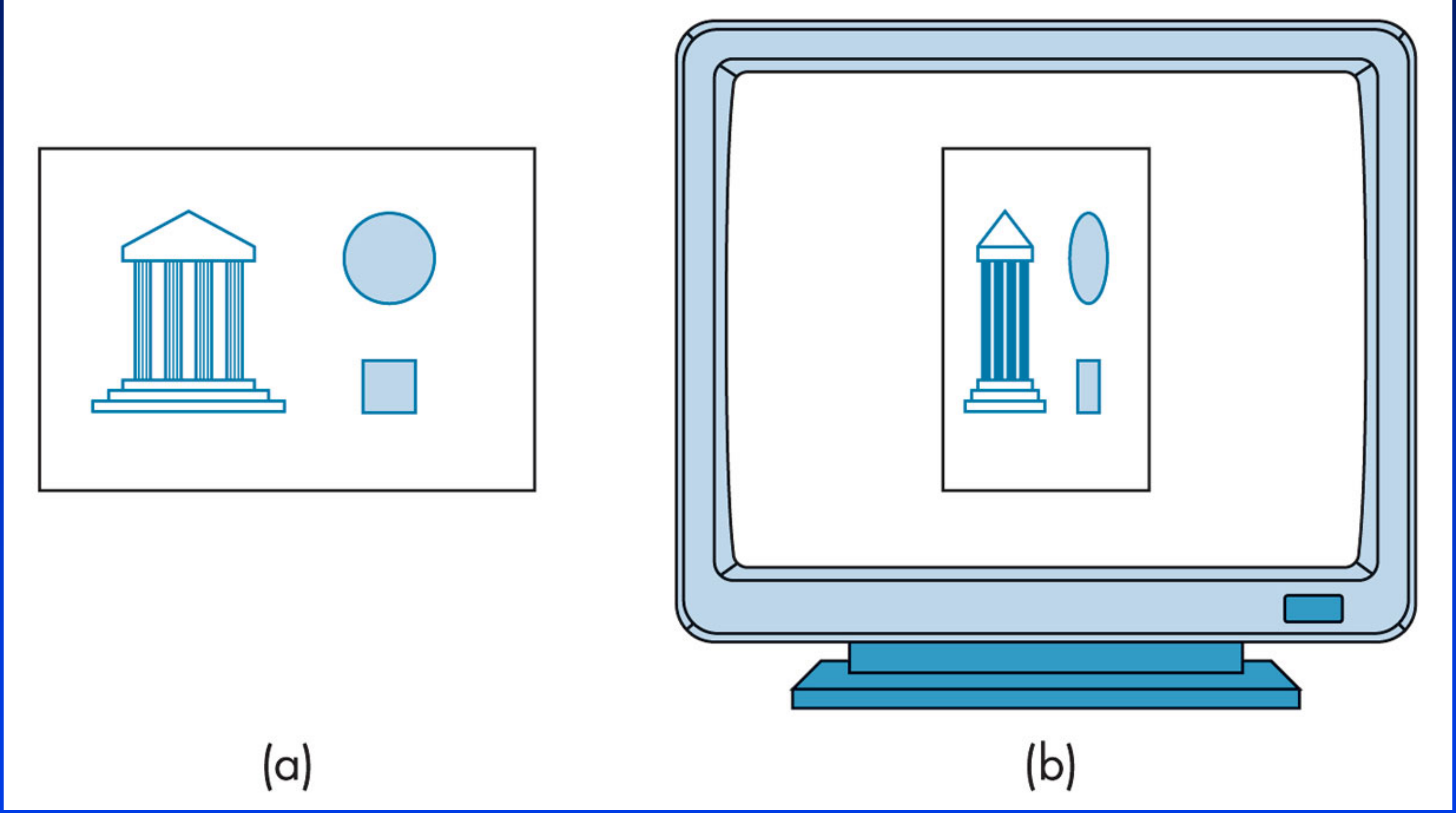
Clipping



(a) Objects before clipping.

(b) Image after clipping.

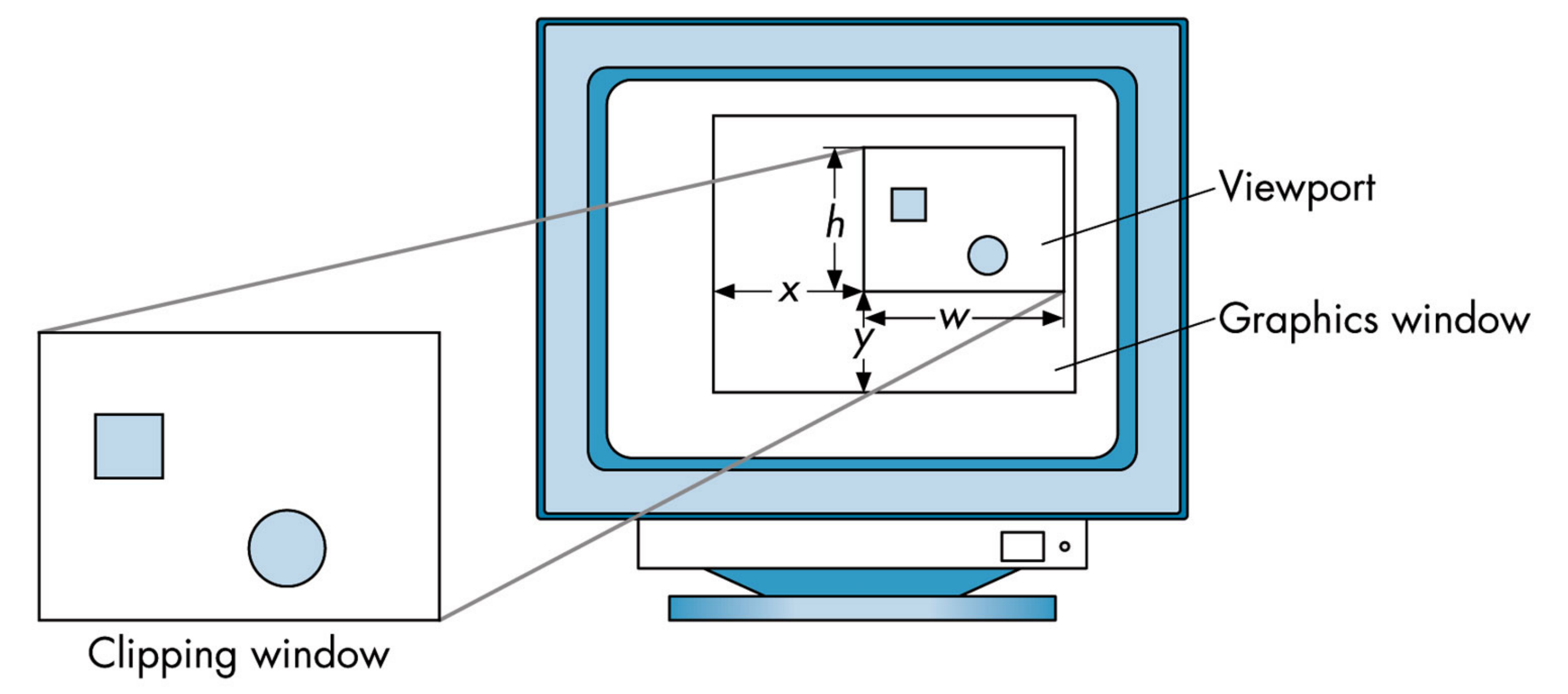
Aspect-ratio



(a) Viewing rectangle.

(b) Display window.

A mapping to the viewport.



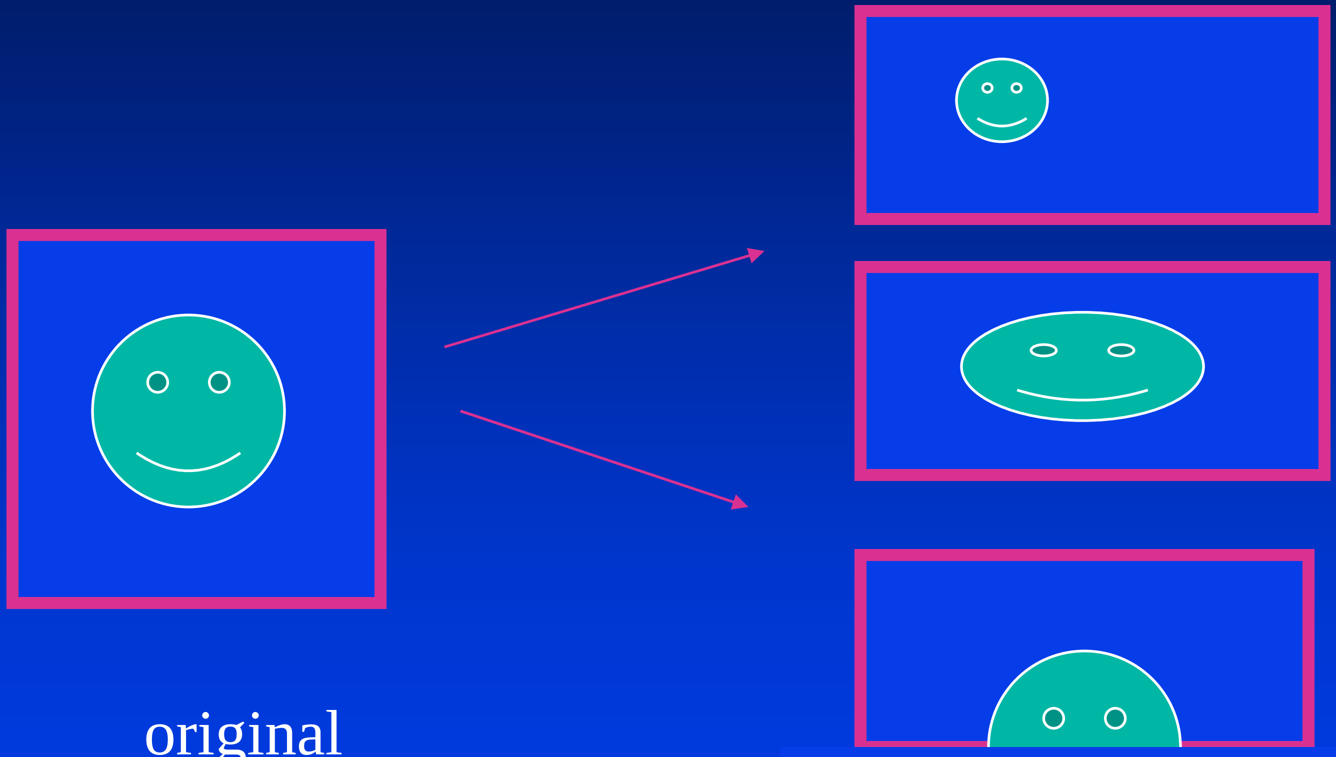
Reshaping the window

We can reshape and resize the OpenGL display window by pulling the corner of the window

What happens to the display?

- Must redraw from application
- Two possibilities
  - *Display part of world*
  - *Display whole world but force to fit in new window*
    - **Can alter aspect ratio**

Reshape possibilities



original

reshaped

The Reshape callback

```
glutReshapeFunc(myreshape)
void myreshape( int w, int h)
```

- Returns width and height of new window (in pixels)
- A redisplay is posted automatically at end of execution of the callback
- GLUT has a default reshape callback but you probably want to define your own

The reshape callback is good place to put viewing functions because it is invoked when the window is first opened

Example Reshape (Demo)

This reshape preserves shapes by making the viewport and world window have the same aspect ratio

```
void myReshape(int w, int h)
{
    glViewport(0, 0, w, h);
    glMatrixMode(GL_PROJECTION); /* switch matrix mode */
    glLoadIdentity();
    if (w <= h)
        gluOrtho2D(-2.0, 2.0, -2.0 * (GLfloat) h / (GLfloat) w,
            2.0 * (GLfloat) h / (GLfloat) w);
    else  gluOrtho2D(-2.0 * (GLfloat) w / (GLfloat) h, 2.0 *
            (GLfloat) w / (GLfloat) h, -2.0, 2.0);
    glMatrixMode(GL_MODELVIEW); /* return to modelview mode */
}
```

Toolkits and Widgets

Most window systems provide a toolkit or library of functions for building user interfaces that use special types of windows called widgets

Widget sets include tools such as

- Menus
- Slidebars
- Dials
- Input boxes

But toolkits tend to be platform dependent

GLUT provides a few widgets including menus

Menus

GLUT supports pop-up menus

- A menu can have submenus

Three steps

- Define entries for the menu
- Define action for each menu item
  - *Action carried out if entry selected*
- Attach menu to a mouse button

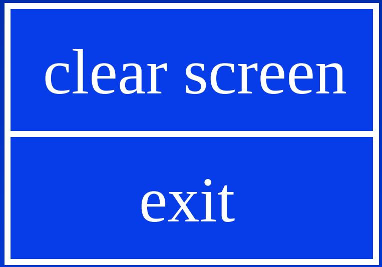
Defining a simple menu

In `main.c`

```
menu_id = glutCreateMenu(mymenu);
glutAddmenuEntry(“clear Screen”, 1);

gluAddMenuEntry(“exit”, 2);

glutAttachMenu(GLUT_RIGHT_BUTTON);
```



entries that appear when
right button depressed

identifiers

Menu actions

- Menu callback

```
void mymenu(int id)
{
        if(id == 1) glClear();
        if(id == 2) exit(0);
}
```

- Note each menu has an id that is returned when it is created
- Add submenus by

entry in parent menu

```
glutAddSubMenu(char *submenu_name, submenu id)
```

Other functions in GLUT

Dynamic Windows

- Create and destroy during execution

Multiple Windows (Demo)

Changing callbacks during execution

Timers

```
void glutTimerFunc( unsigned int msecs, void (*func)(int
value), value);
```

Portable fonts

- `glutBitmapCharacter` (See demo)
- `glutStrokeCharacter` (See demo)