3D computer graphics

Francesc Sebé

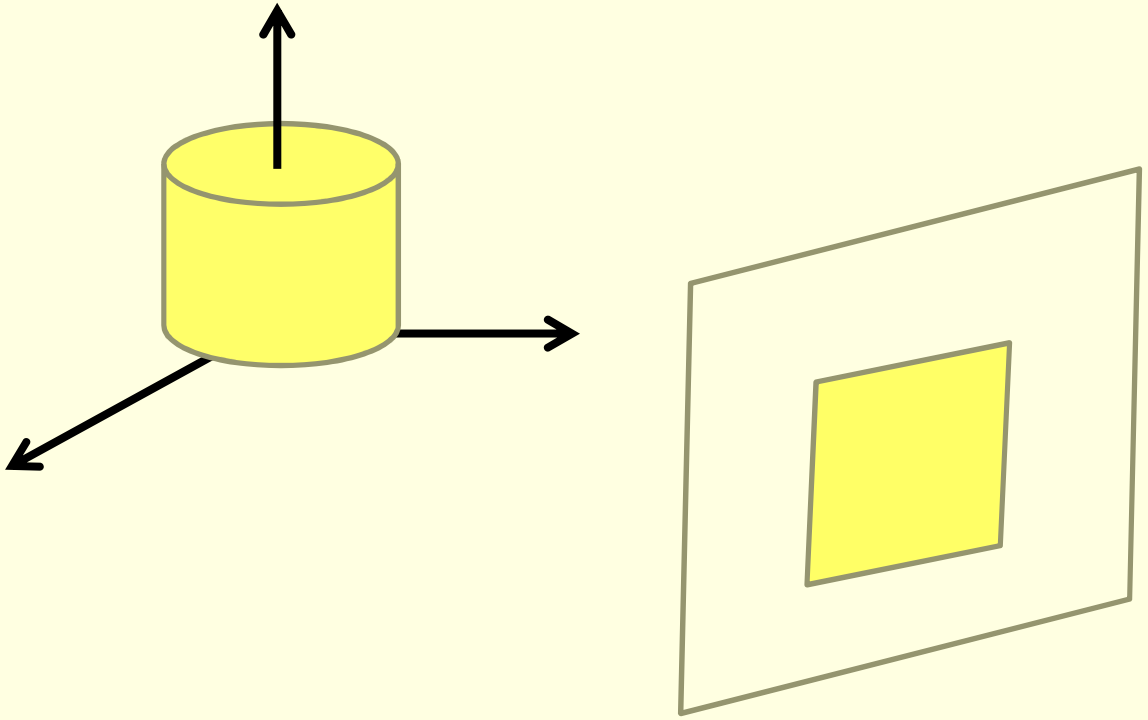
‘Computació gràfica i multimèdia’

Escola Politècnica Superior

Universitat de Lleida

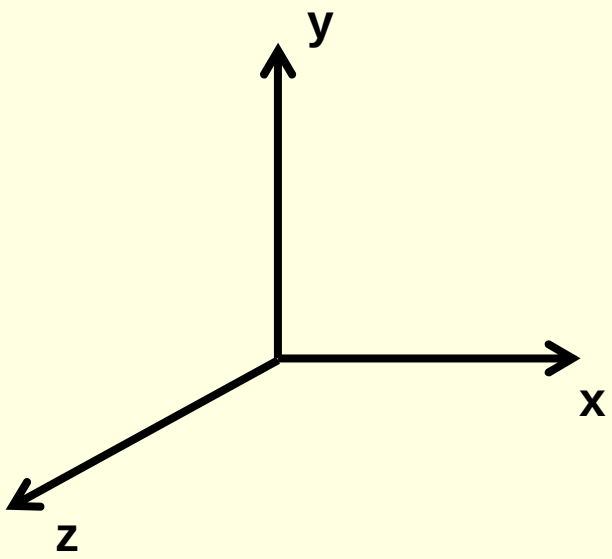
Introduction

- A three-dimensional scene has to be viewed on a two-dimensional screen.



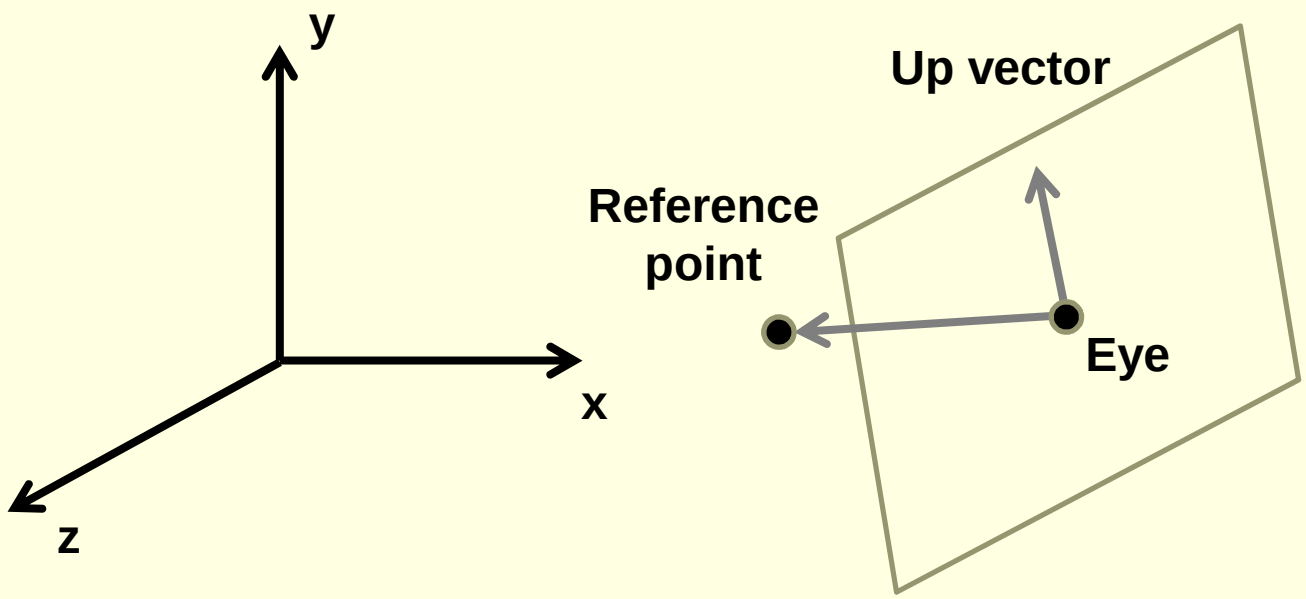
World coordinates

- The objects composing the scene are specified with respect to some refence system
  - Employing (x,y,z)-coordinates



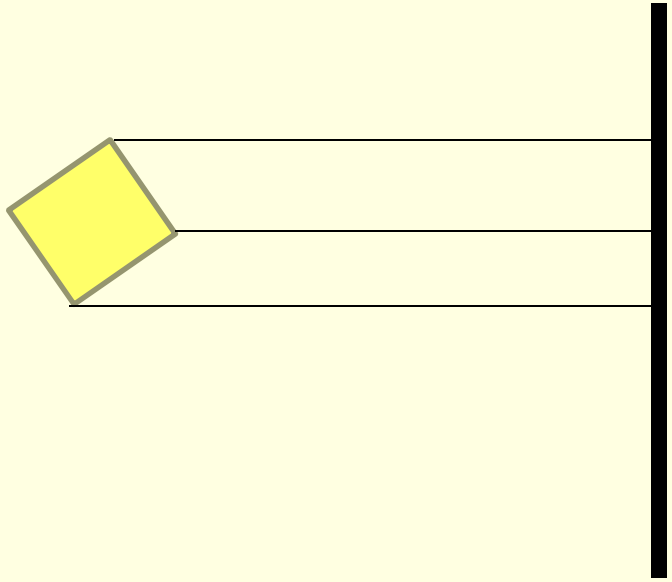
View-plane

- We have to specify the location and orientation of the plane on which the scene will be projected



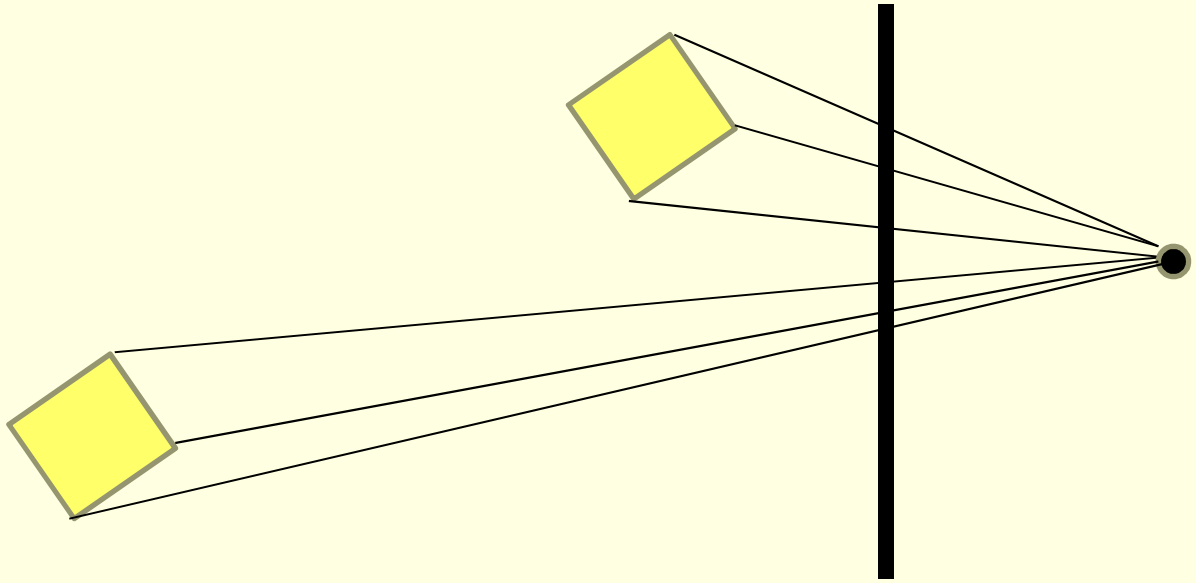
Projections

- **Parallel projection**: coordinate positions are transferred to the view plane along parallel lines



Projections

- **Perspective projection**: projection lines converge to a point behind the view plane

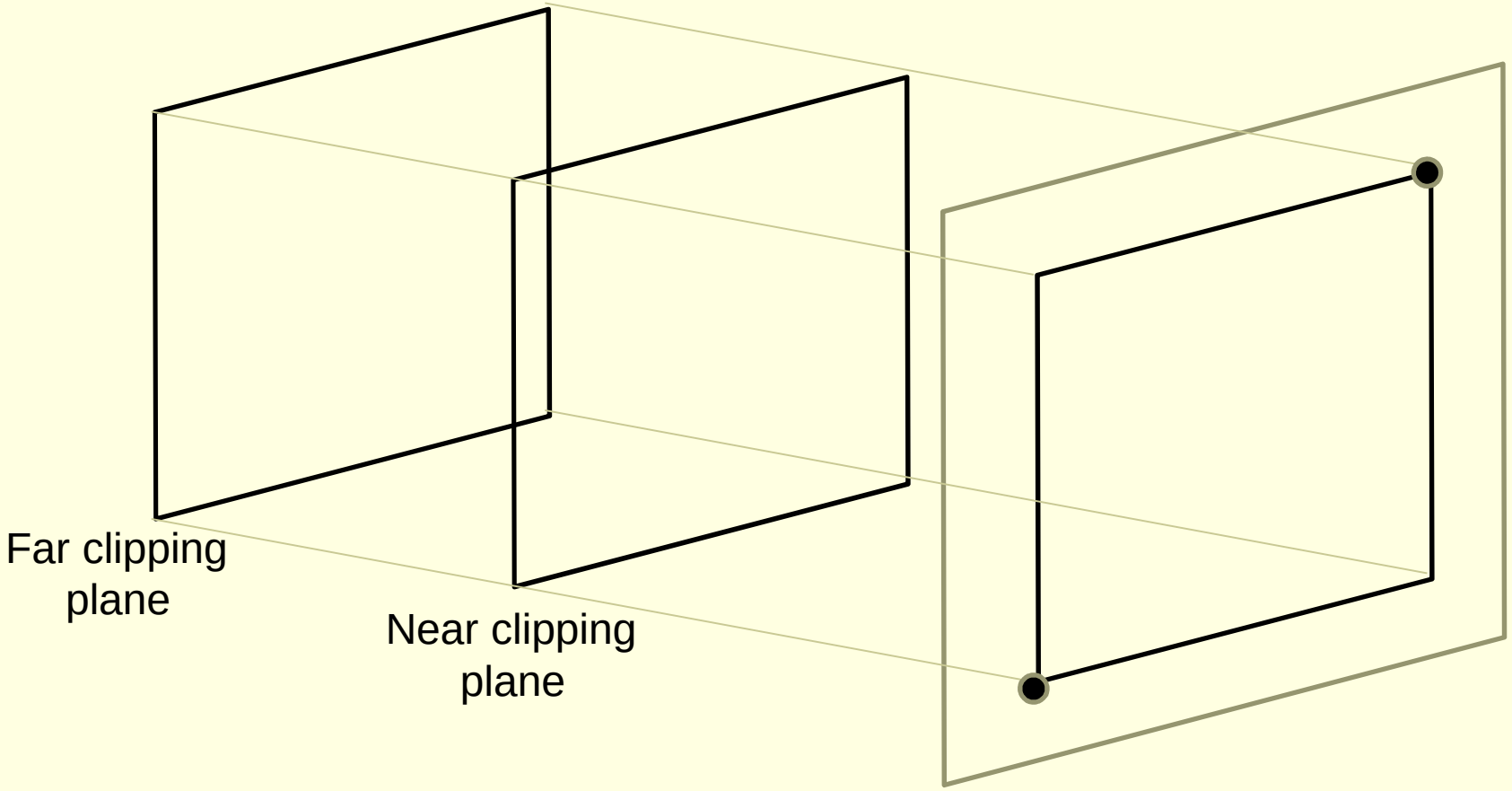


Projections

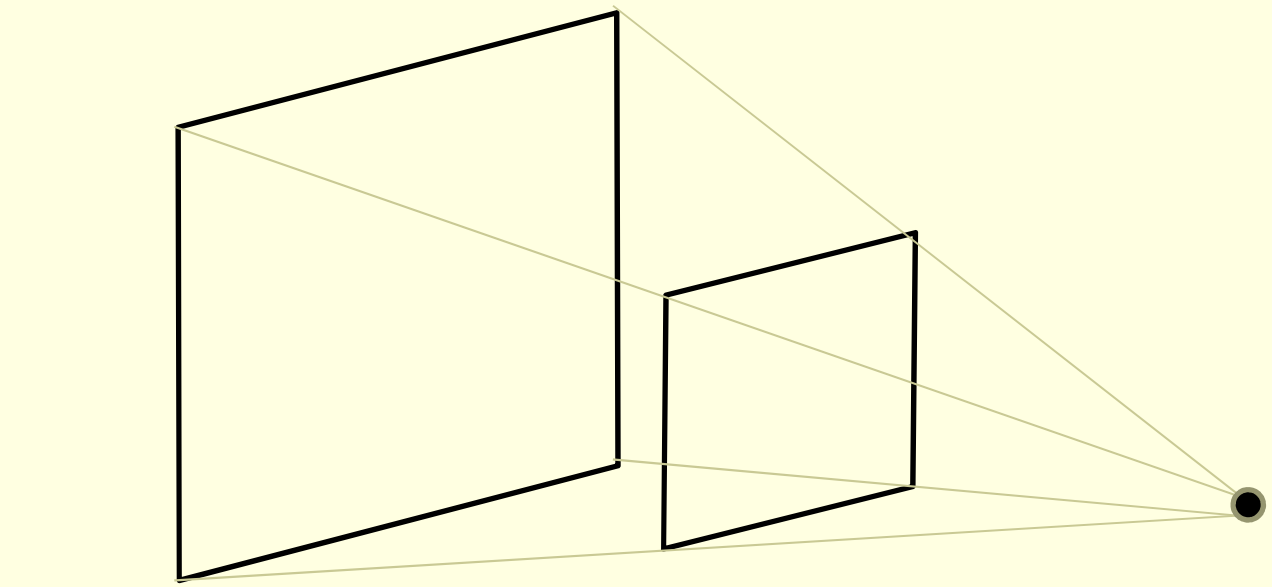
- **Perspective projection:**
  - Does not preserve relative proportions of objects
  - Distant objects become smaller in size
  - It produces more realistic images

Parallel projection: view volume

- Contains the part of the world that will be viewed



Perspective projection: view volume

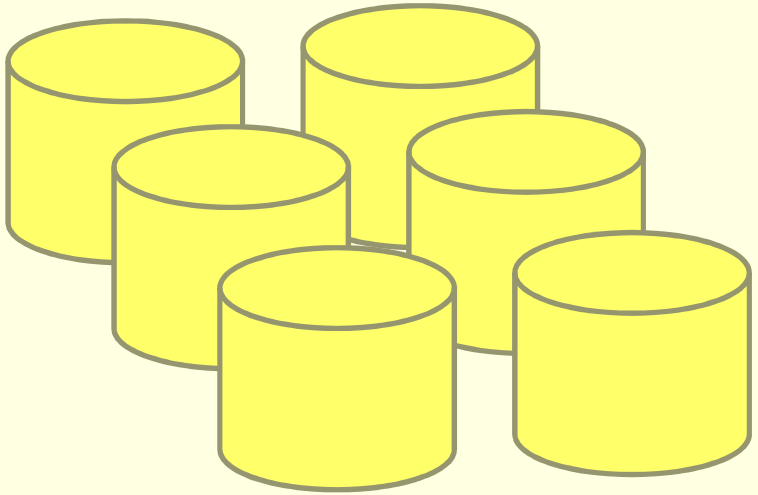


Far clipping plane

Near clipping plane

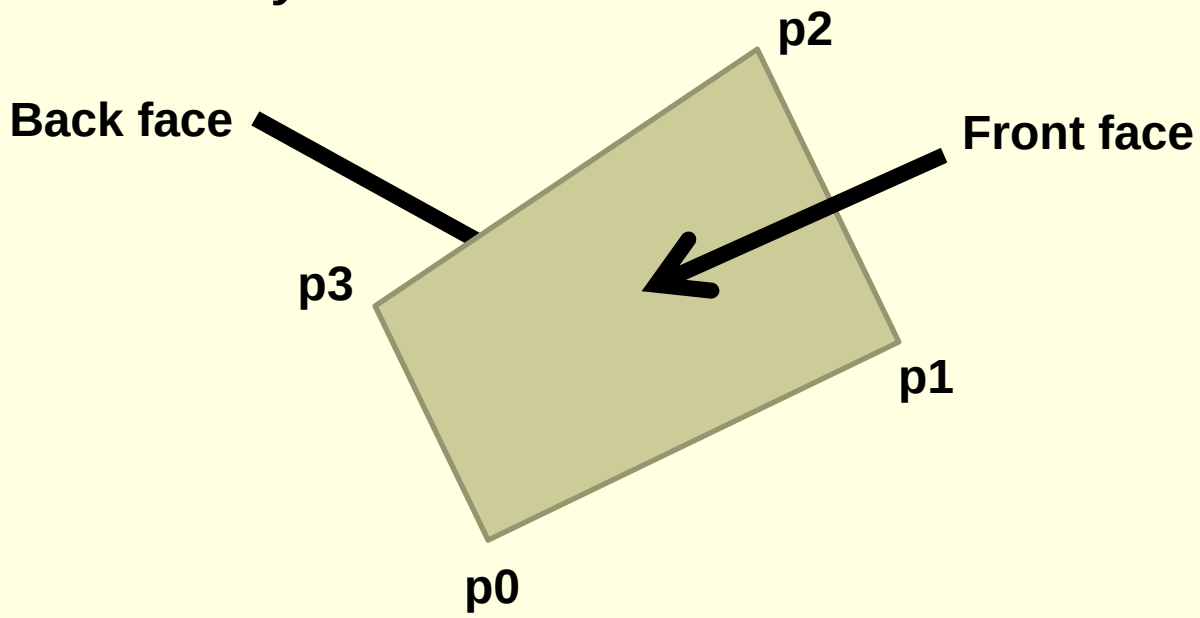
Visible surface detection

- Projection algorithms have to take into account that not all the objects/or parts of them of the view volume are visible.



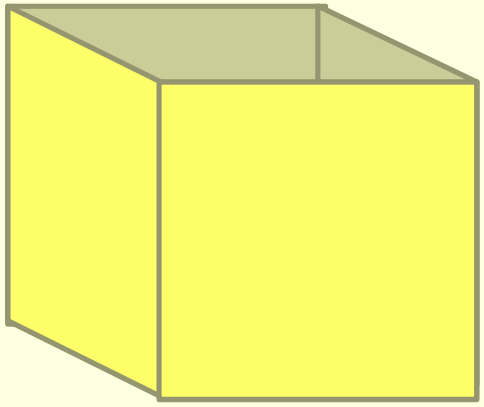
Polygons in the space

- In the space, polygons have front and back-face.
  - Vertices are usually provided counter-clockwisely



Polygons in the space

- In the space, polygons have front and back-face.



3D GRAPHICS IN OPENGL

Enable depth test

- glutInit(&argc, argv);
- glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | **GLUT_DEPTH**);
- glutInitWindowPosition(…);
- glutInitWindowSize(…);
- glutCreateWindow(…);
- **glEnable(GL_DEPTH_TEST)**;
- (…)
- glutMainLoop();

Display procedure

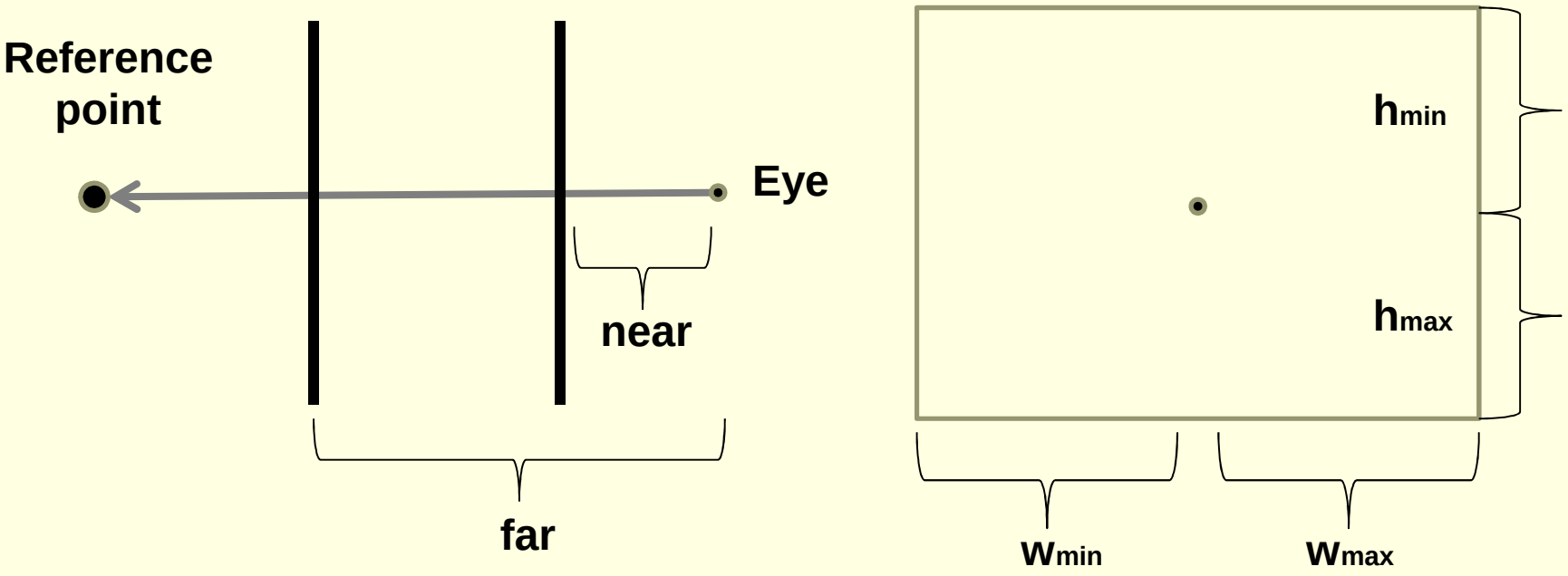
- First, clear the screen:
  - glClearColor(1.0,1.0,1.0,0.0);
  - glClear(GL_COLOR_BUFFER_BIT|**GL_DEPTH_BUFFER_BIT**);
- Next, specify the observer position.

Display procedure

- Observer positioning:
  - glMatrixMode(GL_MODELVIEW);
  - glLoadIdentity();
  - gluLookAt(x,y,z, ref_x, ref_y, ref_z, up_x,up_y,up_z);
  - (x,y,z) → Eye position
  - (ref_x, ref_y, ref_z) → Point the viewer is looking at
  - (up_x,up_y,up_z) → Up vector (unitary)

Display procedure

- Next, set projection parameters:
  - glMatrixMode(GL_PROJECTION);
  - glLoadIdentity();
  - glOrtho(w_{min},w_{max},h_{min},h_{max},near,far);



Display procedures

- Vertices are provided with (x,y,z) coordinates
  - glVertex3i (x,y,z);