The final step: window to viewport transformation

- The normalizing transformation maps the view volume (frustum for perspective projection and parallelepiped for parallel projection) to a cube centered at (0,0,0) (OpenGL case) with sides of length 2.

- Next, clipping is performed; the result is primitives which lie within the cube (*normalized device coordinates*):

$$-1 \leq x \leq 1, -1 \leq y \leq 1, -1 \leq z \leq 1$$

- Next, orthographic projection is performed:

$$x_p = x,\ y_p = y,\ z_p = 0$$

- Note that the z coordinates are not discarded but are used later for visible surface determination.

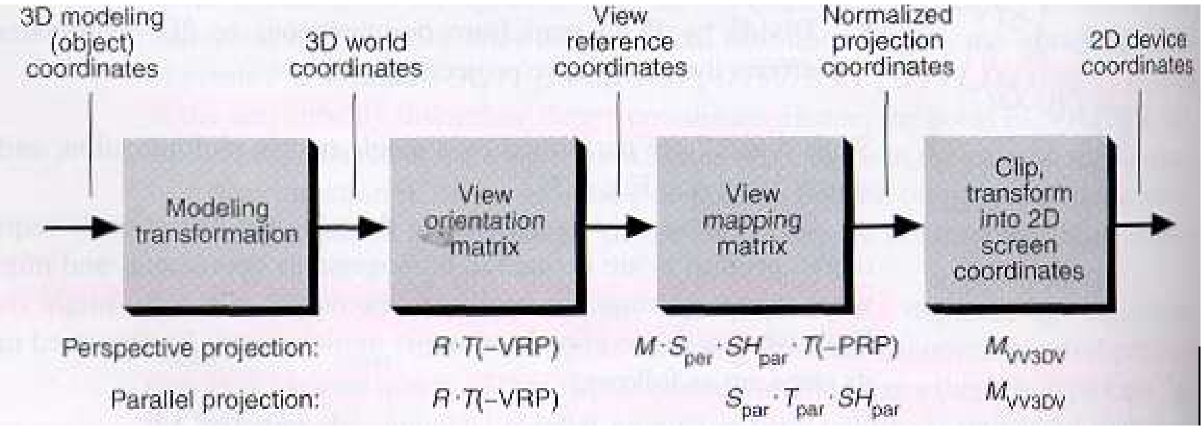
- Next, the window-to-viewport transformation is carried out to obtain the screen coordinates.

- Note that the the normalizing transformation has lined up the sides of the viewing window (z=0) with the viewport (i.e., only translation and scaling are required to map the window to the viewport).

Overview of the 3D viewing process

- **Approach 1**

1. Extend 3D coordinates to homogeneous coordinates
2. Apply normalizing transformation N_{par} or N_{per}
3. Homogenize (i.e., divide by W)
4. Clip in 3D against the parallel or perspective view volume
5. Extend 3D coordinates to homogeneous coordinates
6. Perform parallel (M_{ort}) or perspective (M_{per}) projection
7. Apply the canonical view volume to 3D viewport transformation
8. Homogenize (i.e., divide by W) to obtain 2D coordinates



- **Approach 2**

1. Extend 3D coordinates to homogeneous coordinates
2. Apply normalizing transformation N_{par} or N'_{per}
3. Clip in homogeneous coordinates
4. Perform parallel (M_{ort}) projection
5. Apply the canonical view volume to 3D viewport transformation
6. Homogenize (i.e., divide by W) to obtain 2D coordinates