General Projection
- image plane need not be perpendicular to view plane

Perspective Projection
- our camera must model perspective

Perspective Projections
- classified by vanishing points

Review: Camera Motion
- rotate/translate/scale difficult to control
- arbitrary viewing position
- eye point, gaze/lookat direction, up vector

Review: Moving Camera or World?
- two equivalent operations
- move camera one way vs. move world other way
- example
  - initial OpenGL camera: at origin, looking along -z axis
  - create a unit square parallel to camera at z = -10
  - translate in z by 3 possible in two ways
    - camera moves to z = -3
  - Note OpenGL models viewing in left-hand coordinates
  - camera stays put, but world moves to -7
  - resulting image same either way
  - possible difference: are lights specified in world or view coordinates?

Review: World to View Coordinates
- translate eye to origin
- rotate view vector (lookat – eye) to w axis
- rotate around w to bring up into vw-plane

Pinhole Camera
- theoretical perfect pinhole
- light shining through tiny hole into dark space yields upside-down picture
- one ray of projection

Perspective Projections
- classified by vanishing points
- one-point perspective
- two-point perspective
- three-point perspective
Perspective to Orthographic
• transformation of space
• center of projection moves to infinity
• view volume transformed
  • from frustum (truncated pyramid) to parallelepiped (box)

Simple Perspective Projection Matrix
\[
\begin{bmatrix}
  x \\
  y \\
  z \\
  d
\end{bmatrix}
= \begin{bmatrix}
  1 & 0 & 0 & 0 \\
  0 & 1 & 0 & 0 \\
  0 & 0 & 1 & 0 \\
  0 & 0 & 1/d & 1
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  z \\
  d
\end{bmatrix}
\]

Simple Perspective Projection Matrix
• expressible with 4x4 homogeneous matrix
• use previously untouched bottom row
• perspective projection is irreversible
  • many 3D points can be mapped to same
  (x, y, d) on the projection plane
  • no way to retrieve the unique z values

Orthographic Camera Projection
• camera's back plane parallel to lens
• infinite focal length
• no perspective convergence
  • just throw away z values

Why Canonical View Volumes?
• permits standardization
• clipping
  • easier to determine if an arbitrary point is enclosed in volume with canonical view volume vs. clipping to six arbitrary planes
• rendering
  • projection and rasterization algorithms can be reused

Perspective Projection
• expressible with 4x4 homogeneous matrix
• use previously untouched bottom row
• perspective projection is irreversible
  • many 3D points can be mapped to same (x, y, d) on the projection plane
  • no way to retrieve the unique z values

Moving COP to Infinity
• as COP moves away, lines approach parallel
  • when COP at infinity, orthographic view

Canonical View Volumes
• standardized viewing volume representation
  • perspective
  • orthogonal
  • parallel

View Volumes
• specifies field-of-view, used for clipping
• restricts domain of z stored for visibility test

Projective Transformations
• planar geometric projections
• planar: onto a plane
• geometric: using straight lines
• projections: 3D -> 2D
• aka projective mappings
• counterexamples?
Understanding Z

- Why near and far plane?
  - Near plane:
    - Avoid singularity (division by zero, or very small numbers)
  - Far plane:
    - Store depth in fixed-point representation (integer), thus have to have fixed range of values (0…1)
    - Avoid/reduce numerical precision artifacts for distant objects

Orthographic Derivation

- Scale, Translate, Reflect for new coord sys

Orthographic OpenGL

```c
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
glOrtho(left,right,bot,top,near,far);
```