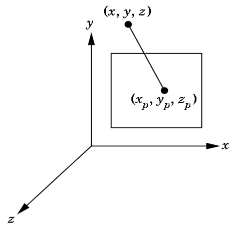
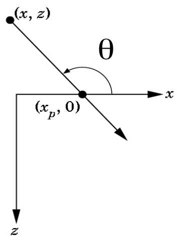
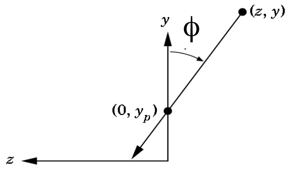
Projection Normalization for Oblique Parallel Projections



- Orthogonal parallel projection can be seen as just a special case of an oblique parallel projection.
- An oblique projection can be characterized by the angle of the projectors with the VP.



(a)



(b)

- Top and side views (see left) of a projector and VP z=0.
- (θ, ϕ) characterize the degree of obliqueness.

Considering the top view (a), x_p can be found by

$$\tan\theta = \frac{z}{x - x_p} \Leftrightarrow x_p = x - z\cot\theta$$

and likewise following (b): $y_p = z - z\cot\phi$

For VP z=0 this results to P:

$$M = \begin{bmatrix} 1 & 0 & -\cot\theta & 0 \\ 0 & 1 & -\cot\phi & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

After extracting the orthogonal projection from *M* we derive an additional shear:

$$M = M_{orth} H(\theta, \phi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -\cot\theta & 0 \\ 0 & 1 & -\cot\phi & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Projection Normalization for Oblique Parallel Projections

$$M = M_{orth} H(\theta, \phi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -\cot\theta & 0 \\ 0 & 1 & -\cot\phi & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

M is not in canonical form! It is a simple shear followed by an orthographic projection.

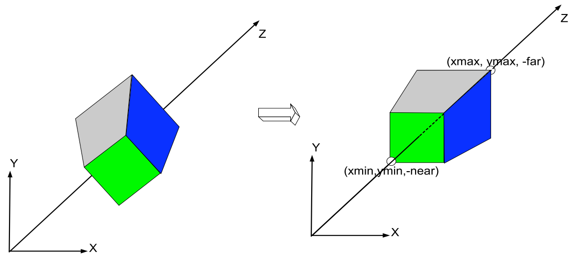
The same translation and scaling used for the orthographic case has to be inserted between the shear and the projection:

$$M = M_{orth} STH(\theta, \phi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{2}{x_{\max} - x_{\min}} & 0 & 0 & -\frac{x_{\max} + x_{\min}}{x_{\max} - x_{\min}} \\ 0 & \frac{2}{y_{\max} - y_{\min}} & 0 & -\frac{y_{\max} + y_{\min}}{y_{\max} - y_{\min}} \\ 0 & 0 & \frac{-2}{far - near} & \frac{far + near}{far - near} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -\cot\theta & 0 \\ 0 & 1 & -\cot\phi & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

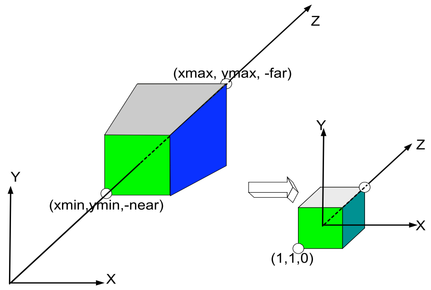
$$\Rightarrow P = \begin{bmatrix} \frac{2}{x_{\max} - x_{\min}} & 0 & \frac{-2\cot\theta}{x_{\max} - x_{\min}} & -\frac{x_{\max} + x_{\min}}{x_{\max} - x_{\min}} \\ 0 & \frac{2}{y_{\max} - y_{\min}} & \frac{-2\cot\phi}{y_{\max} - y_{\min}} & -\frac{y_{\max} + y_{\min}}{y_{\max} - y_{\min}} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Camera Transformation (M_{cam}) and Projection Normalization (M_{pers}) for Perspective Views

- Camera Transformation for Perspective Views (M_{cam})
  1. Convert World to Camera Coordinates
     - Camera (COP) at origin, looking in the –z direction
     - Display plane center along the z axis
  2. Combinations of translate, scale, and rotate transformations
     - Can be accomplished through camera location specification



- Projection Normalization for Perspective Views (M_{pers})
  1. Convert viewing box to right frustum (on axis)
     - This is because many APIs including OpenGL allow non-right viewing volumes
  2. Scale the right frustum into canonical form
  3. Convert viewing box (right frustum) to a right parallelpiped
     - "Shrinking" objects that are further away



Projection Normalization for Perspective Projections

- Again, find a transformation that distorts the vertices in a way that we can use a simple canonical projection:
  perspective-normalization transformation

Given 1) a simple perspective projection with VP z=-1 and COP at origin:

$$M_{persp} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

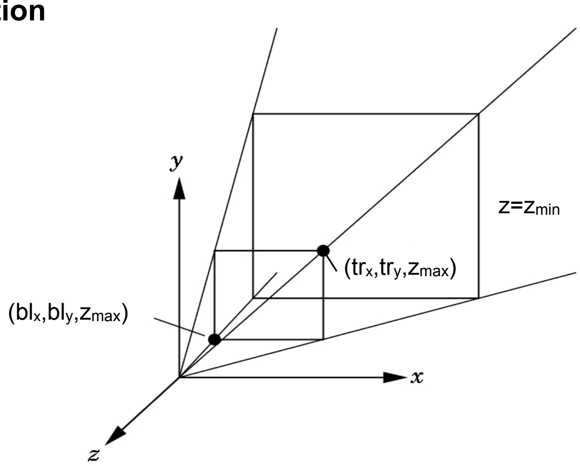
Given 2) a perspective view volume with the angle of view being 90° => frustum sides intersect VP at 45° angle. View volume is a semi infinite view pyramid with:

x = +/- z, y = +/- z

View volume is finite by specifying the

near plane $z = z_{max}$ and the

far plane $z = z_{min}$ with $z_{max} > z_{min}$



Projection Normalization for Perspective Projections

Let N be a nonsingular matrix similar to M_{persp} with: $\alpha \neq 0, \beta \neq 0$

$$N = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \alpha & \beta \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Applying N to a homogeneous-coordinate point:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \alpha & \beta \\ 0 & 0 & -1 & 0 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ \alpha z + \beta \\ -z \end{bmatrix} \xrightarrow{\text{perspective division}} \begin{bmatrix} -\frac{x}{z} \\ -\frac{y}{z} \\ -\left(a + \frac{\beta}{z}\right) \\ 1 \end{bmatrix} = \begin{bmatrix} x'' \\ y'' \\ z'' \\ 1 \end{bmatrix}$$

Applying an orthographic projection along the z-axis to N:

$$M_{orth}N = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \alpha & \beta \\ 0 & 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Applying the result to an arbitrary point p':

$$M_{orth}Np' = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 0 \\ -z \end{bmatrix} \xrightarrow{\text{perspective division}} \begin{bmatrix} -\frac{x}{z} \\ -\frac{y}{z} \\ 0 \\ 1 \end{bmatrix}$$

If we apply N followed by an orthogonal projection to a point we achieve the same result for x and y as applying a perspective projection to the same point!

Projection Normalization for Perspective Projections

Nonsingular matrix N transforms the original viewing volume into a new volume.

Now choosing α, β such that the new volume is the canonical view (clipping) volume.

Given the sides

$x = +/- z$	transformed by	x''	results to	$x'' = +/- 1$	and
$y = +/- z$	transformed by	y''	results to	$y'' = +/- 1$	and

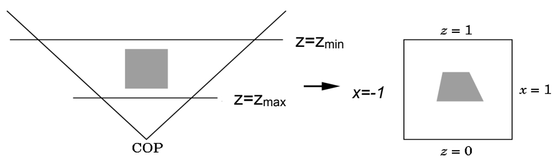
The front of the view volume $z=z_{max}$ is transformed to: $z'' = -\left(a + \frac{\beta}{z_{max}}\right)$

The back of the view volume $z=z_{min}$ is transformed to: $z'' = -\left(a + \frac{\beta}{z_{min}}\right)$

How to choose values α, β ?

We choose: $\alpha = \frac{z_{max} + z_{min}}{z_{max} - z_{min}}$ $\beta = -\frac{2 z_{max} z_{min}}{z_{max} - z_{min}}$

Then the plane $z=z_{min}$ is mapped to the plane $z''=-1$ and the plane $z=z_{max}$ is mapped to the plane $z''=1$, hence we achieve the canonical volume:



➡ *N* transforms the viewing volume to a right parallelepiped, a following orthographic projection is the same as a perspective transformation. *N* is called the **perspective normalization matrix**.

Projection Normalization for Perspective Projections

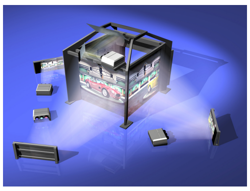
$z'' = -\left(a + \frac{\beta}{z}\right)$ is nonlinear but preserves depth-ordering, hence $z_1 > z_2 \Rightarrow z''_1 > z''_2$

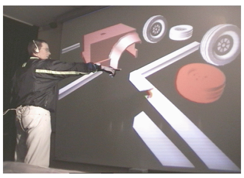
Notes:

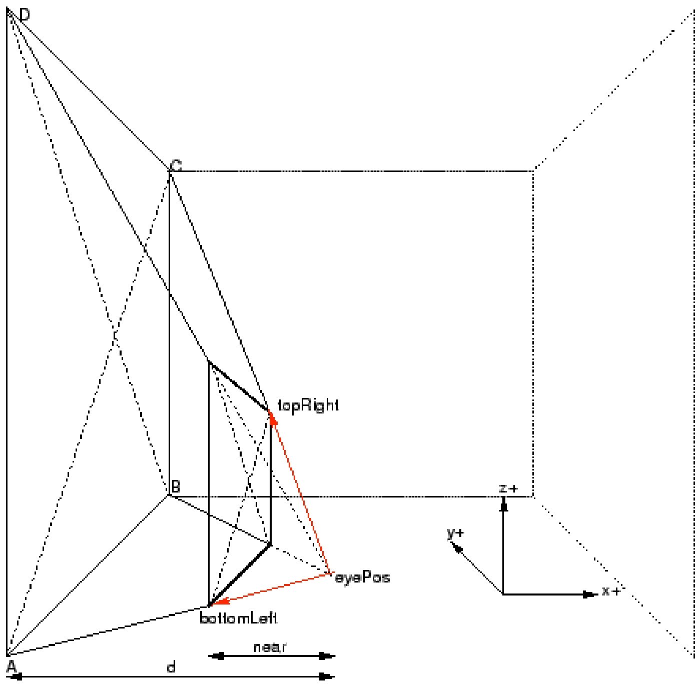
- Hidden surface removal works in the normalized volume.
- Nonlinearity can cause numerical problems due to limited resolution in the depth buffer.
- Only one viewing pipeline is required by carefully choosing a projection matrix to insert into the pipeline.
- **Perspective-Normalization Matrix (N_{per})** converts frustum view volume into canonical orthogonal view volume:

$$N_{per} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{far + near}{far - near} & -\frac{2\, far \times near}{far - near} \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Projection Normalization for non-right Perspective Projections





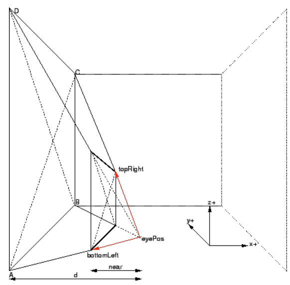


Projection Normalization for non-right Perspective Projections

- The right (symmetric) perspective projection is a special case for an arbitrary perspective projection like the orthographic projection was for the parallel oblique case.
- An arbitrary perspective projection is required, e.g., for driving several large-screen projection-based VR display types which
  1. use head tracking and
  2. fix the VPs w.r.t. the moving COP

  and which hence require dynamic frustum calculation (responsive workbenches, Holoscreens, CAVEs,…)

  ➢ This type of projection is a.k.a. **off-axis projection**!



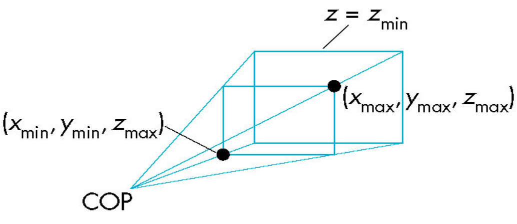
Do derive the projection matrix for off-axis set-ups we follow the same path as we did for the parallel projection case:

➢ Insertion of a shear transformation into the projection pipeline.

Projection Normalization for non-right Perspective Projections

Find *H* which satisfies:

$$H\begin{bmatrix} \frac{(x_{\min}+x_{\max})}{2} \\ \frac{(y_{\min}+y_{\max})}{2} \\ z_{\min} \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ z_{\min} \\ 1 \end{bmatrix}$$



$$H(\theta,\phi) = H\left(\cot^{-1}\left(\frac{x_{\min}+x_{\max}}{2z_{\max}}\right), \cot^{-1}\left(\frac{y_{\max}+y_{\min}}{2z_{\max}}\right)\right)$$

The resulting frustum is described by the planes:

$$x = \pm\frac{(x_{\max}-x_{\min})}{2z_{\max}}, y = \pm\frac{(y_{\max}-y_{\min})}{2z_{\max}}, z = z_{\min}, z = z_{\max}$$

Projection Normalization for non-right Perspective Projections

Now scale the sides to achieve $x = \pm z, y = \pm z$

without changing near/far planes .

$$S = S\left(\frac{2z_{max}}{x_{max} - x_{min}}, \frac{2z_{max}}{y_{max} - y_{min}}, 1\right)$$

S is without reference to z since it is uniquely determined by its results on four points, here the intersection points of the near plane and the sides. Now

$$N = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \alpha & \beta \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

gets the far plane to -1 and the near plane to 1 with the already chosen

$$\alpha = \frac{z_{max} + z_{min}}{z_{max} - z_{min}} \quad \beta = -\frac{2z_{max}z_{min}}{z_{max} - z_{min}}$$

Projection Normalization for Perspective Views

- This results to the projection matrix:

$$P_{pers} = N_{per}SH = \begin{bmatrix} \frac{2(-near)}{x_{max} - x_{min}} & 0 & \frac{x_{max} + x_{min}}{x_{max} - x_{min}} & 0 \\ 0 & \frac{2(-near)}{y_{max} - y_{min}} & \frac{y_{max} + y_{min}}{y_{max} - y_{min}} & 0 \\ 0 & 0 & -\frac{far + near}{far - near} & -\frac{2\, far \times near}{far - near} \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

- Where H converts a non-right frustum to a right frustum
- Where S scales the frustum into a canonical perspective view volume
- Where N is the Perspective-Normalization Matrix

Project onto Projection Plane

- Since normalization changed all projections into an orthogonal projection:
  - Just ignore the z value!
  - In effect, a non-event!
- In reality, we retain the z-value for hidden-surface removal and shading effects.
- Viewable world now in **Normalized Device Coordinates (NDC)**.

3D Viewing Summary