Transforming the normal vector for reflection

- Reflection of a triangle in the xy plane through y-axis.
  - Not to be confused with a rotation around y-axis.
- Geometric argument suggests normal should remain the same because what is in front of the triangle remains in front.
- \((B-A) \times (C-A)\) is out of page by right hand rule.
- But \((B'-A') \times (C'-A')\) is into page by right hand rule.
- Resolve this by noting that a reflection maps a right-hand coordinate system into a left-hand one. So, we need to use the “left-hand” rule.

\[
\begin{align*}
M &= \begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{bmatrix} \\
\end{align*}
\]

Transforming the Normal Vector

Derived transformation for the normal: \(N = (M^{-1})^T\)

Note that \(N\) is not necessarily a unit vector.

The formula proves that orthogonal transformations also transform the normal because orthogonal transformations satisfy:

\[
M = (M^{-1})^T
\]

(The inverse of an orthogonal matrix is its transpose)

Transforming the Normal Vector

\(M\) is a transformation in ordinary coordinates (not homogenous, for simplicity).

Let \(t\) be a vector tangent to the polygon, and \(n\) the normal.

\(Mt\) is on the transformed polygon.

Want a transformation \(N\) so that \(N\) is perpendicular to all \(Mt\).

\(n \cdot t = 0\) and so (trick!)

\[
(n \cdot M^{-1})(Mt) = 0
\]

So, \(n \cdot M^{-1}\) is the row vector version of the transformed normal.

\(So, \ N = (n \cdot M^{-1})^T = (M^{-1})^T n\)

And \(N = (M^{-1})^T\)

3D Graphics Concepts

(H&B ch. 7, Watt ch. 5, Foley et al ch. 6)

- Modeling: For now, objects will be collections of polygons in 3D. Complex shapes will be many small polygons.
- Issues:
  - Which polygons can be seen? (some polygons hide others, and some are outside the relevant volume of space and need to be clipped)
  - Where do they go in the 2D image? (key abstraction is a virtual camera)
  - How bright should they be? (for example, to make it look as if we are looking at a real surface)
Objects (world coordinates)

Transform to standard camera coordinates

Clip in 3D (against canonical view frustum)

Project to 2D with standard camera model

Render 2D polygons

Determine what is in front*

Lighting information (was in world coordinates, but now transformed)

Camera information (world coordinates)

Back face culling

* different parts have different positions in the pipeline, depending on strategy

Today

Pinhole cameras

- Abstract camera model--box with a small hole in it
- Pinhole cameras work for deriving algorithms--a real camera needs a lens

Pinhole

image plane

virtual image

Parallel lines meet*

Common to draw film plane in front of the focal point (virtual image)

*Exceptions?
Vanishing points

• Each set of parallel lines (=direction) meets at a different point
  – The vanishing point for this direction
• Sets of parallel lines on the same plane lead to collinear vanishing points.
  – The line is called the horizon for that plane
  – Standard horizon is the horizon of the ground plane.
• One way to spot fake images