Transforming Normals

- normal
- direction specifying orientation of polygon
- vs. w=1 for points/vectors of object vertices
- used for lighting
- must be normalized to unit length
- can compute if not supplied with object

Transforming Geometric Objects

- lines, polygons made up of vertices
- transform the vertices
- interpolate between
- does this work for everything? no!
- normals are trickier

Modularization

- advantages
  - define object once, instantiate multiple copies
  - transformation parameters often good control knobs
  - maintain structural constraints if well-designed
- limitations
  - expressivity: not always the best controls
  - can't do closed kinematic chains
  - keep hand on hip
- avoid incremental changes to coordinate systems
- accumulation of numerical errors

Computing Normals

- normal
- direction specifying orientation of polygon
- w=0 means direction with object
- can compute if not supplied with object

Matrix Stacks

- challenge of avoiding unnecessary computation
- using inverse to return to origin
- computing incremental $T_{1} \rightarrow T_{2}$

Hierarchical Modelling

- advantages
  - no need to compute inverse matrices all the time
  - modularizes changes to pipeline state
- disadvantages
  - not built in to WebGL
  - but easy to implement with Array.pop/push
  - see also

Translation Hierarchies

- drawing a scaled square
  - push/pop ensures no coord system change

scaling and Rotating

- nonuniform scaling does not work
- x-y=0 plane
  - line $x=y$
  - normal: $[1,-1,0]$
  - direction of line $x=-y$
  - (ignore normalization for now)
Planes and Normals

- Plane is all points perpendicular to normal
  - \( N \cdot P = 0 \) (with dot product)
- \( N^T \cdot P = 0 \) (matrix multiply requires transpose)

Explicit form: plane = \( ax + by + cz + d \)

Finding Correct Normal Transform

- Transform a plane
  
  \[
  \begin{pmatrix}
  P \\
  N
  \end{pmatrix} \rightarrow \begin{pmatrix}
  MP \\
  QN
  \end{pmatrix} \quad \text{given } M,
  \\
  \text{what should } Q \text{ be?}
  
  \begin{align*}
  N^T P &= 0 \\
  (QN)^T (MP) &= 0 \\
  N^T Q^T MP &= 0 \\
  Q^T M &= I
  \end{align*}
  
  Thus, the normal to any surface can be transformed by the inverse transpose of the modeling transformation.

Thus the normal to any surface can be transformed by the inverse transpose of the modeling transformation.