Shadows, etc.

cheesy shadows

picture & demo

cheesy shadows

for each light {
  for each polygon in the scene {
    for each vertex of the polygon {
      project the vertex onto the floor with respect to the light
    }
    draw the projected polygon
  }
}

projection

suppose we have a point light at the origin

\[ s = \alpha v \text{ for some } \alpha \]

(0,0,0)

assume the equation of the floor plane is \( Ax + By + Cz + D = 0 \)

\[ s = \alpha v \text{ for some } \alpha \]

so \[ \alpha = -D / (Av_x + Bv_y + Cv_z) \]

(0,0,0)
using the pipeline

\[ s = \frac{-Dv}{(Av_x + Bv_y + Cv_z)} \]

\[
\begin{pmatrix}
D & 0 & 0 & 0 \\
0 & D & 0 & 0 \\
0 & 0 & D & 0 \\
-A & -B & -C & 0 \\
\end{pmatrix}
\begin{pmatrix}
v_x \\
v_y \\
v_z \\
1 \\
\end{pmatrix}
\]

\( M_{\text{shadow}} \rightarrow \) perspective division \( \rightarrow \) add p

for point light at point p

\( M(v-p) \rightarrow \) perspective division \( \rightarrow \) add p

final shadow matrix for point light

\[ p \cdot q \mathbf{I} - p^T q \]

where \( p = p_x, p_y, p_z, 1 \), \( q = A, B, C, D \)

Exercise:
Use this equation to compute the shadow matrix when the light is at 0. How does your answer compare with our derivation for this special case?

directional light

shadow matrix

for a light with direction \( (p_x, p_y, p_z) \)

\[ p \cdot q \mathbf{I} - p^T q \]

where \( p = p_x, p_y, p_z, 0 \), \( q = A, B, C, D \)

practical problems

drawing two polygon in same place
polygon offset

when enabled, a small offset is added to the depth values during scan conversion

see Woo p. 250

practical problems

drawing beyond the floor

stencil buffer

initialize stencil buffer to all 0's
draw floor:
whenever pixel is drawn, set corresponding position in stencil buffer to 1
draw shadow:
draw shadow pixels provided the corresponding stencil values is 1

practical problems

what about ambient light?

alpha blending