CSC418 Computer Graphics

- Polygon normals
- Back Faces
- Visibility Algorithms

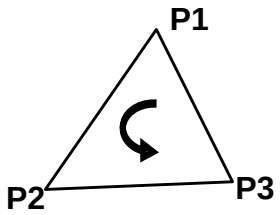
Polygon Normal

- Triangle normal calculation
- Polygon normal calculation
- Polymeshes (vertex normals)

Polygon Normal

- **Triangle normal calculation**

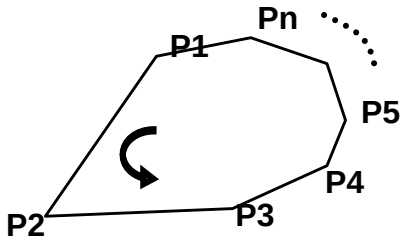
N=(P2-P1)X(P3-P2)



Polygon Normal

- **Polygon normal calculation**

N=(P2-P1)X(P3-P2)



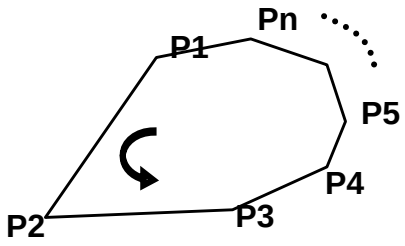
Polygon Normal

- **Polygon normal calculation**

N=(P2-P1)X(P3-P2)

Problems

- **Sliver triangles**
- **Non-planar polygons**

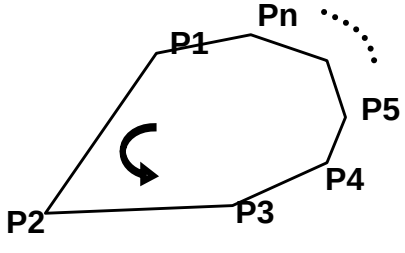


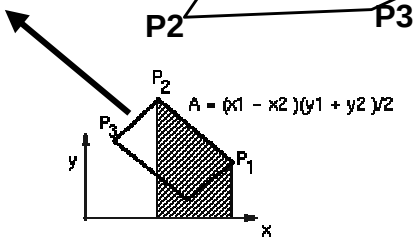
Polygon Normal

- **Polygon normal calculation**

$Nx = \Sigma\ (yj - yi)(zj + zi)$
$Ny = \Sigma\ (zj - zi)(xj + xi)$
$Nz = \Sigma\ (xj - xi)(yj + yi)$

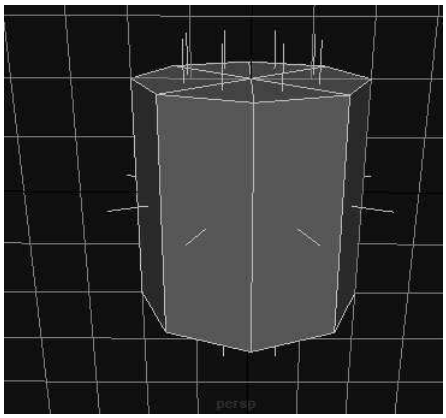
$j = (i+1) \bmod n$





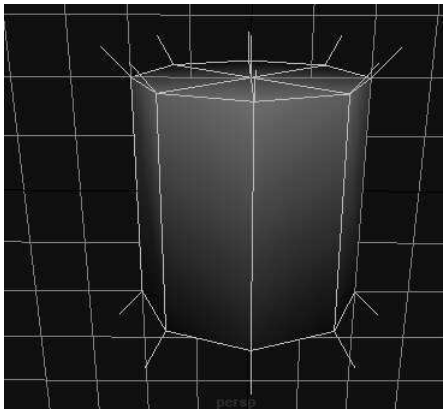
Polymesh Normal

- **Face normals**



Polymesh Normal

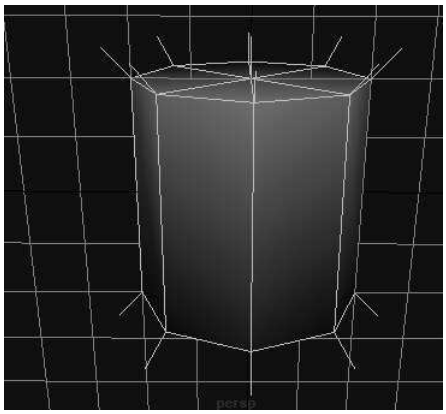
- **Vertex normals**



Polymesh Normal

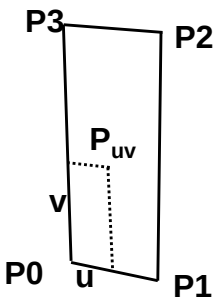
- **Vertex normals**

average the face normals



Bilinear Interpolation

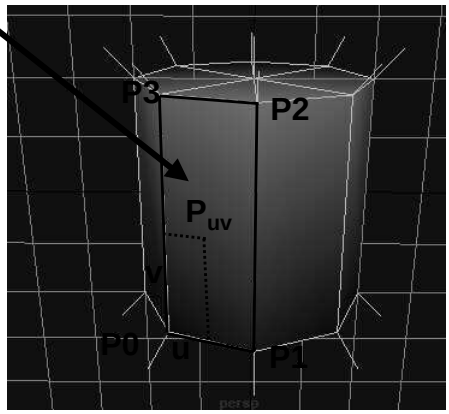
P_{uv} = (1-u)(1-v)P0 +
(u)(1-v)P1 +
(u)(v)P2 +
(1-u)(v)P3



Bilinear Interpolation

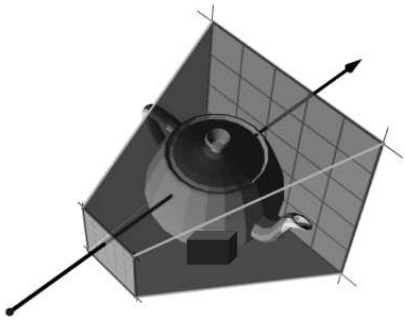
Gourard shading (color interpolation)

$P_{uv} = (1-u)(1-v)P0 +$
$(\ u\ )(1-v)P1 +$
$(\ u\ )(\ v\ )P2 +$
$(1-u)(\ v\ )P3$



Visibility Problem

- **What is NOT visible?**



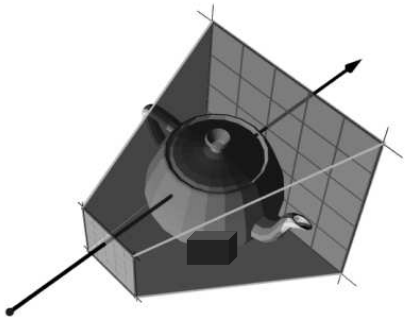
Visibility Problem

- **What is NOT visible?**

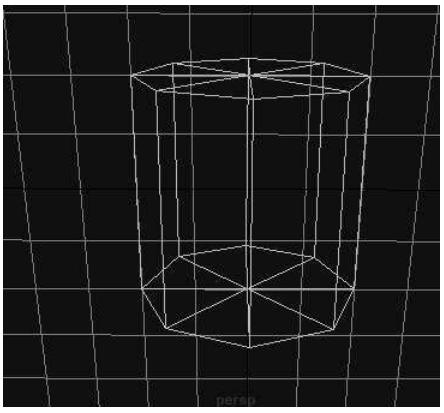
primitives outside of the field of view

back-facing primitives

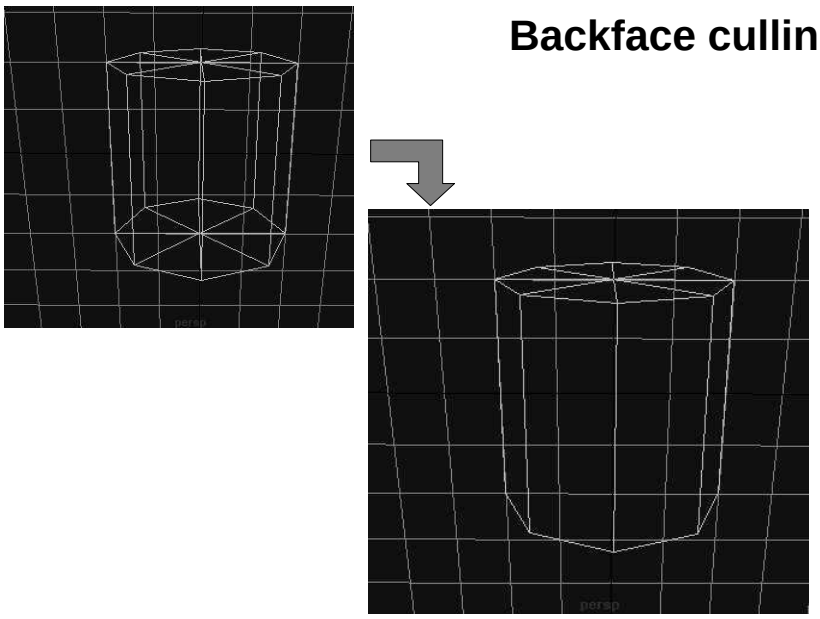
primitives occluded by other objects closer to the camera



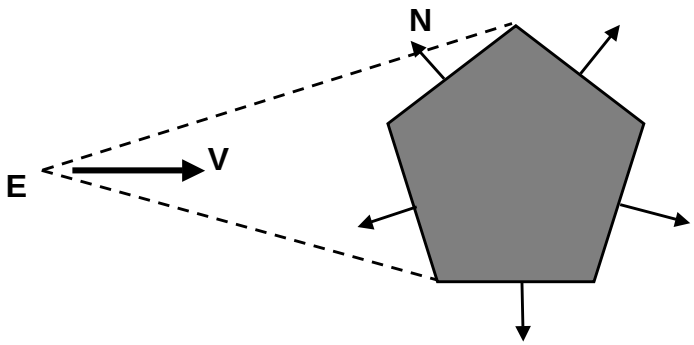
Backface culling



Backface culling

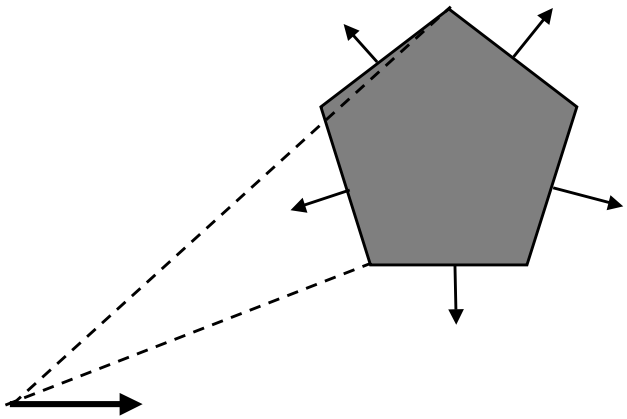


Backface culling



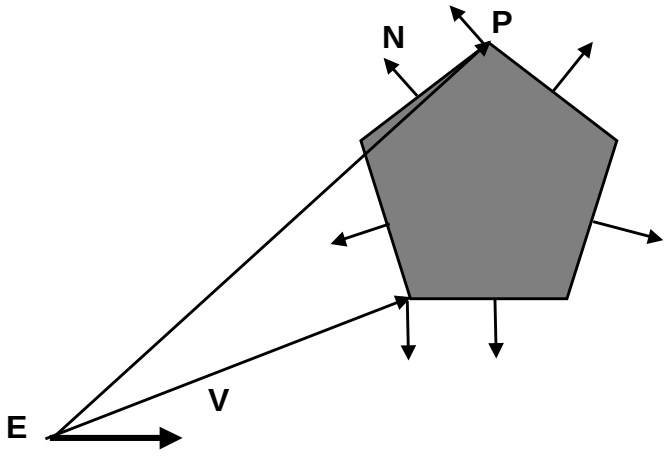
Backface culling

- N.V > 0 is a back face?



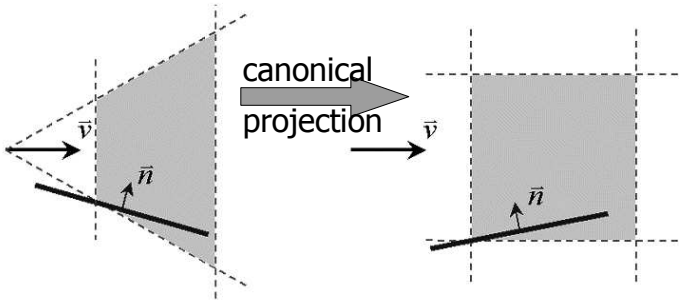
Backface culling

- N.(P-E) >0



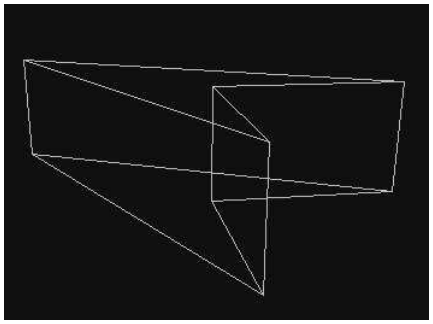
Backface culling

Where in the graphics pipeline can we do backface culling?



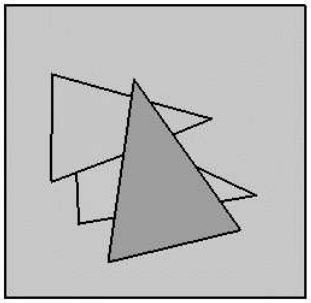
Occluded faces

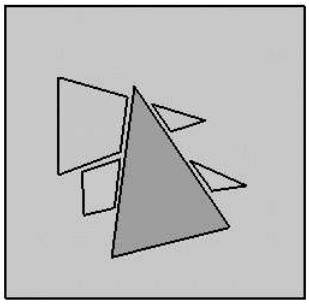
Does backface culling always determine visibility completely for a single object?



Occluded faces

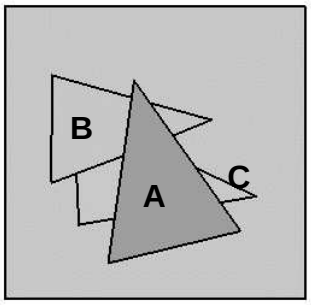
- **In typical scenes** some polygons will overlap, we must determine which portion of each polygon is visible to eye!

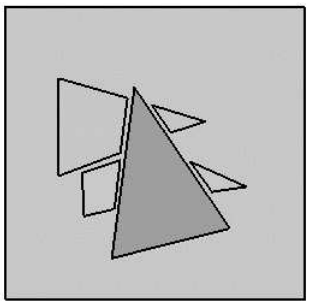




Painters Algorithm

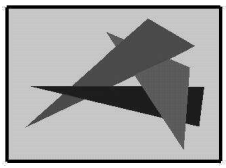
- **Sort primitives in Z.**
- **Draw primitives back to front (CBA).**

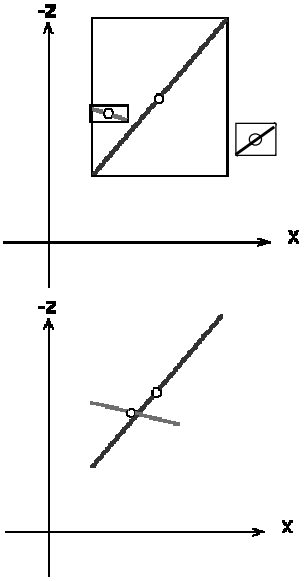




Painters Algorithm

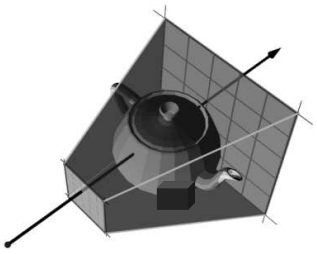
- Problems
  - Large faces
  - Intersecting faces
  - Cycles





Visibility Problem

- Image space algorithms
  - Operate in display terms pixels, scanlines
  - Visibility resolved to display resolution
  - Examples: Z-buffer, ray-tracing
  - O(n*resolution)
- Object Space algorithms
  - Analytically compute visible fragments
  - Examples: painters algorithm, BSP
  - $O(n^2)$



CSC418 Computer Graphics

- **Next Lecture**
  - **BSP trees**
  - **Depth sorting**
  - **Z-buffer A-buffer**