Johns Hopkins
Engineering for Professionals

605.767 Applied Computer Graphics

Brian Russin

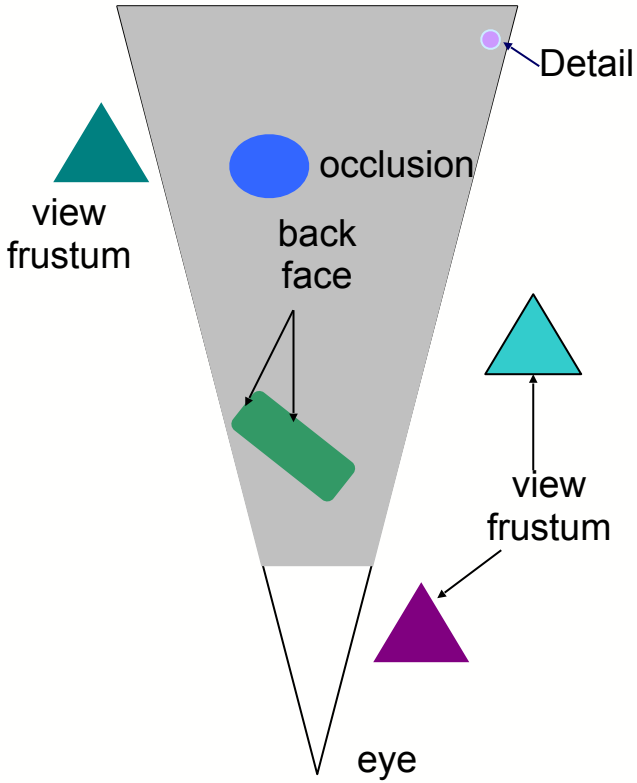
Module 5C
Culling Techniques

Culling Techniques Overview

- Culling
  - Backface culling
  - Hierarchical view-frustum culling
  - Portal culling
  - Detail culling
  - Occlusion culling
- Levels of Detail

Culling Techniques

- Cull means “remove from a flock”
  - In graphics context - do not process data that will not contribute to the final image
- Types of culling
  - Backface culling
  - Hierarchical view-frustum culling
  - Portal culling
  - Detail culling
  - Occlusion culling
- **Exact Visible Set** (EVS)
  - All primitives that are partially or fully visible
  - Often prohibitively expensive to determine
- **Potentially Visible Set** (PVS)
  - Fully includes the EVS but less expensive to determine
    - Only discards invisible geometry
      - Conservative



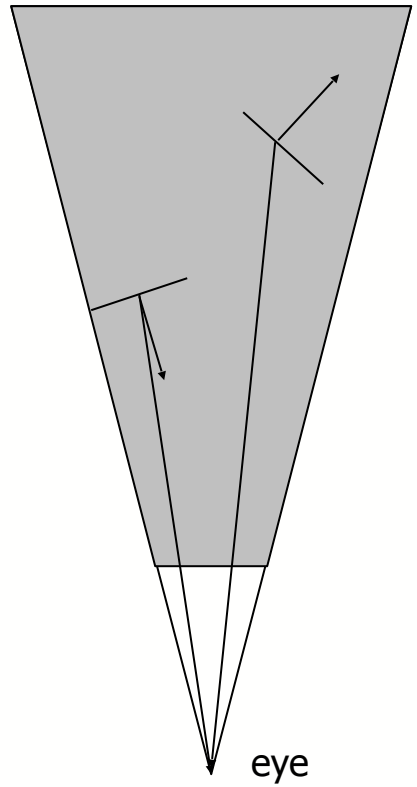
Backface Culling

- Technique to discard polygons that face away from the viewer
- Can be used for closed surfaces
  - Can generally remove half the polygons
- Also where backfaces never should be seen
  - Example: walls in a room
- Test can be done in world or view space
- Also can be done in screen space
  - Computing the normal of the projected polygon
  - OpenGL uses this method

Cube:
3 faces are visible-they face the viewer
3 are hidden or blocked by the other faces-they face away from the viewer

Application Back-face Polygon Culling

- Simple geometric test
  - Construct a vector from an arbitrary point on the plane (a polygon vertex) to the view position
  - Compute dot product of this vector with the polygon normal
    - Vectors do not need to be unit length
  - Front faces: sign of dot product > 0
  - Back faces: sign of dot product < 0
  - Test is equivalent to finding to which side of the plane the view position lies
- Can be done in world space or view space
- Orthographic projection
  - Do not need to construct a vector
  - Use the negative view direction
  - Constant for the scene



OpenGL Back Face Culling

- OpenGL back face culling requires consistent vertex ordering
  - Counter-clockwise or clockwise
  - Use **glFrontFace(GL_CCW)** or **glFrontFace(GL_CW)** to specify orientation
  - Specify which face to cull: **glCullFace(GL_BACK)** (or **GL_FRONT)**
  - Enable or disable: **glEnable(GL_CULL_FACE)**
- OpenGL computes the polygon area in projected, window coordinates

$$a = \frac{1}{2}\sum_{i=0}^{n-1} x_i y_{i+1} - x_{i+1} y_i$$

  - Sign of the area determines orientation of the polygon (cw vs. ccw)
  - If polygon specified as CCW: if a > 0 the polygon is front-facing
  - If polygon specified as CW: if a < 0 the polygon is front-facing
- OpenGL uses this method since normals to the polygon plane may not be available
- Impact on OpenGL pipeline
  - Decreases load on rasterization stages but increases load in geometry stages

Clustered Backface Culling

- Clustered Backface Culling methods test to decide if a set of polygons should be sent to the rendering pipeline
  - Testing is faster if done on a set of primitives
  - https://cgvr.cs.uni-bremen.de/teaching/cg_literatur/clustered_backface_culling.pdf
- Use a normal cone
  - Contains points and normal directions for all polygons in the set
  - Defined by normal n, half angle , anchor point a, and distances to truncate the cone
  - See Figure 19.11 (14.11 in 3rd Edition)

$$n \cdot \left(\frac{e - f}{\| e - f \|} \right) \geq \sin(\alpha)$$ Polygons in the cone are frontfacing

$$-n \cdot \left(\frac{e - b}{\| e - b \|} \right) \geq \sin(\alpha)$$ Polygons in the cone are backfacing

e is location of viewer
f is the apex of the front facing cone
b is the apex of the back facing cone