Culling

Computer Graphics
CSE 167
Lecture 12
CSE 167: Computer graphics

• Culling
  – Definition: selecting from a large quantity
  – In computer graphics: selecting primitives (or batches of primitives) that are visible

• If culling is performed early in the graphics pipeline, then rejected invisible objects are not fetched, transformed, rasterized, or shaded
Types of culling

• View frustum culling
• Backface culling
• Contribution (or small object) culling
• Degenerate culling
• Occlusion culling
View frustum culling

- Triangles outside of view frustum are off-screen

Images: SGI OpenGL Optimizer Programmer's Guide
Bounding volumes

• How to cull objects consisting of many polygons?
• Intersect bounding volume with view frustum instead of each primitive
• Simple shape that completely encloses an object
Bounding volumes

• Commonly, a cuboid or sphere
  – Easier to calculate tight fits for cuboids (boxes)
  – Easier to calculate culling for spheres
• Cull bounding box
  – Box is smallest box containing the entire object
  – Simple approach: rectangular box, axis-aligned to object space coordinate system
    • May not be tightest fit
View frustum culling

• Frustum is defined by 6 planes
• Each plane divides space into outside/inside
• Check each object against each plane
  – Outside, inside, intersecting
• If outside all planes
  – Outside the frustum
• If inside all planes
  – Inside the frustum
• Else, partially inside frustum
  – Intersecting the frustum
Frustum with oriented planes

• Normal of each plane points outside of frustum
  – Outside is positive distance
  – Inside is negative distance
Distance to plane

• A plane is described by a point $\mathbf{p}$ on the plane and a unit normal $\mathbf{n}$

• Find the (perpendicular) distance from point $\mathbf{x}$ to the plane
Distance to plane

- The distance is the length of the projection of $(x-p)$ onto $n$

$$\text{dist} = (x - p) \cdot \vec{n}$$
Distance to plane

• The distance has a sign (oriented plane)
  – Positive on the side of the plane the normal points to
  – Negative on the opposite side
  – Zero exactly on the plane

• Divides 3D space into two infinite half-spaces

\[
\text{dist}(x) = (x - p) \cdot \hat{n}
\]
Distance to plane

• Simplification

\[
\text{dist}(x) = (x - p) \cdot n \\
= x \cdot n - p \cdot n \\
\text{dist}(x) = x \cdot n - d, \quad d = pn
\]

• Where \( d \) is distance from the origin to the plane
• \( d \) is independent of \( x \)
• We can represent a plane with just \( d \) and \( n \)
Sphere-plane test

• For sphere with radius \( r \) and origin \( x \), test the distance to the origin, and see if it is beyond the radius

• Three cases:

  \( \text{dist}(x) > r \)
  - Completely above

  \( \text{dist}(x) < -r \)
  - Completely below

  \(-r < \text{dist}(x) < r \)
  - Intersects
View frustum culling using spheres

• Pre-compute the normal \( \mathbf{n} \) and value \( d \) for each of the six planes.
• Given a sphere with center \( \mathbf{x} \) and radius \( r \)
• For each of the six clipping planes
  – If \( \text{dist}(\mathbf{x}) > r \), then sphere is outside (terminate loop)
  – Else if \( \text{dist}(\mathbf{x}) < -r \), then add 1 to count
    • (Alternatively, set a flag if \( \text{dist}(\mathbf{x}) \geq -r \))
• If we did not terminate the loop early, check the count
  – If the count is 6 (or flag was not set), then the sphere is completely inside
  – Otherwise, the sphere intersects the frustum
View frustum culling using spheres

- Math for Game Developers - Frustum Culling
  - https://www.youtube.com/watch?v=4p-E_31XOPM
View frustum culling groups of objects

- Able to cull a whole group quickly
- But, if the group is partly in and partly out, able to cull individual objects
View frustum culling using hierarchical bounding volumes

- Given hierarchy of objects
- Bounding volume of each node encloses the bounding volumes of all its children
- Start by testing the outermost bounding volume
  - If it is entirely outside, do not draw the group at all
  - If it is entirely inside, draw the whole group
View frustum culling using hierarchical bounding volumes

• If the bounding volume is partly inside and partly outside
  – Test each child’s bounding volume individually
  – If the child is in, then draw it; if it is out, then cull it; if it is partly in and partly out, then recurse
  – If recursion reaches a leaf node, then draw it normally
View frustum culling

• Rendering Optimizations - Frustum Culling
  – https://www.youtube.com/watch?v=kvVHp9wMAO8

• View Frustum Culling Demo
  – https://www.youtube.com/watch?v=bJrYTBGpwic
Backface culling

• Consider triangles as “one-sided” (oriented triangle) and only visible from the “front”
• Closed objects
  – If the “back” of the triangle is facing away from the camera, it is not visible
  – Gain efficiency by not drawing it (culling)
  – Roughly 50% of triangles in a scene are back facing
Backface culling

• Convention: triangle is front facing if vertices are ordered counterclockwise
Backface culling

- Compute triangle normal after projection (homogeneous division)
  \[ \mathbf{n} = (\mathbf{p}_1 - \mathbf{p}_0) \times (\mathbf{p}_2 - \mathbf{p}_0) \]

- If the third component of \( \mathbf{n} \) negative, then front-facing; otherwise, back-facing
  - Remember: projection matrix is such that homogeneous division flips sign of third component
Backface culling

Without backface culling  With backface culling
Backface culling

- Allow one- or two-sided triangles

In OpenGL

```c
glDisable(GL_CULL_FACE);
```

Two-sided triangles (no backface culling)

```c
glEnable(GL_CULL_FACE);
glCullFace(GL_BACK);
```

One-sided triangles (backface culling)
Contribution (or small object) culling

• Object projects to less than a specified size
  – Cull objects whose screen-space bounding box is less than a threshold number of pixels, as these objects do not contribute significantly to the final image
Degenerate culling

- Projected triangle is degenerate
  - Normal $n = 0$
    - Plane at infinity
      - Not really degenerate
  - All vertices in a straight line
    - Colinear
  - All vertices in the same place
    - Coincident

Occlusion culling

- Geometry hidden behind occluder cannot be seen

Images: SGI OpenGL Optimizer Programmer's Guide
Occlusion culling

• Umbra 3 Occlusion Culling explained
  https://www.youtube.com/watch?v=5h4QgDBwQhc