CSC 240 Computer Graphics
Video 17: Hidden Surface Removal
& Blender Basics

Nick Howe
Smith College

Some slides & content courtesy Sara Mathieson
Hidden Surface Removal

Given some scene geometry and camera viewpoint, in most cases many scene polygons will not be visible.

- Why?
  - Outside view frustum (behind camera, offscreen, far away)
  - Facing away from camera
  - Occluded by other objects

https://www.artstation.com/artwork/9Kx5R
Hidden Surface Removal

Occlusion culling = rough but drastic occlusion handling

- Rule out occluded objects; don’t even try rendering

https://docs.unity3d.com/560/Documentation/Manual/OcclusionCulling.html
Hidden Surface Removal

View frustum culling uses geometry to weed out polygons

- Too near or far: compare z value to clipping planes
  Q. Why have both near and far clipping planes?

- Top/bottom/left/right: compute projected coordinates, then run 2D clipping algorithm
Hidden Surface Removal

Backface culling skips polygons that face away from view

- How to quickly check?
  - Sign of dot product: surface normal with camera normal $\hat{p} \cdot \hat{c} < 0$
  - **Winding order**: points are counter-clockwise in projection
Hidden Surface Removal

Occlusion handling is critical to correct rendering

- Surfaces that are behind others should not be visible!

- Two simple ways to achieve:
  - **Painter’s Algorithm**
    Renders polygons by distance from camera (farthest first)
  - **Z-Buffering**
    Records z coordinate for all pixels; only overwrites if closer

Pros & cons?

- Faster
- Fine-grained

https://www.videoblocks.com/video/3d-wireframe-buildings-over-black-background-p9qls7t
Hidden Surface Removal

Z-Buffering uses an extra image plane holding $z$ values

- Each pixel draw checks its $z$ value against the buffer
  - Only draws if $z$ is greater (closer to camera)
  - Updates buffer with new value during draw (in hardware)

"Z buffer" https://commons.wikimedia.org/wiki/
Hidden Surface Removal

What will be the rendered image and z-buffer?
(Assume orthographic projection of this 2D scene.)

- Render in this order: red, yellow, green, blue

Render: yellow, red, green, black
Z-buffer: -7, -3, -1, -1, -99
Z Fighting

Overlapping polygons at the same z value can interfere with each other.

- Z buffer cannot distinguish which is in front \(\rightarrow\) unpredictable result!

- Two possible causes:
  - Polygons actually at same position (solve via model change)
  - Polygons separated by less than Z buffer resolution

- Possible solutions:
  - Increase Z resolution – less separation between near and far clip
  - Post-projection offset

```javascript
var material = new THREE.MeshLambertMaterial({
  polygonOffset: true,
  polygonOffsetFactor: -1.0,
  polygonOffsetUnits: -4.0
});
```
Questions

1. Why is z-buffering generally preferred to the painter’s algorithm?
   *It runs faster and handles partial overlap*

2. Give the z-buffer at right, list the shapes from nearest to farthest.
   *Pentagon, triangle, star, diamond, circle*

3. Suppose that the camera is pointing in the –z direction. What has to be true of the normal vectors of polygons eliminated via backface culling?
   *They must have positive z components: \( \mathbf{\hat{c}} = (0,0,-1), \mathbf{\hat{p}} = (x, y, z) \Rightarrow \mathbf{\hat{p}} \cdot \mathbf{\hat{c}} = 0x + 0y - 1z = -z \)
   *This is less than zero only if z is positive.*
Why Blender?

Blender is a digital art tool, not a programming tool. So why study it in a computer science class?

- It offers a platform for experimenting with geometry, texture, ray tracing, and other topics we have studied.
- It facilitates the creation of 3D models for use in programmed environments.

Our use of Blender will focus mostly on these topics. We will not study many advanced features.

*Consider taking an ARS digital art course to learn more!*
Blender Orientation

Blender is a tool for developing high-quality 3D models.
Blender Basic Controls

Before you can interact with objects, you need to be able to view them.

**Pan**
- Translate the point of view
  - **Shift+Drag MMB / Shift-Alt-Drag**

**Orbit**
- Rotate the point of view around center of scene
  - **Drag MMB / Alt-Drag**

**Zoom**
- Move the point of view in and out
  - **Ctrl+Drag MMB / Ctrl-Alt-Drag**
Editing in Blender

Many details; some basic principles

- Many changes (scale, translate, rotate, etc.) controlled by mouse
- Type \( x \), \( y \), or \( z \) during edit operation to restrict to one axis
- Type \( X \), \( Y \), or \( Z \) during edit operation to restrict to two axes axis
- Type a number during edit operation to choose a precise value

- Other tricks we will learn during labs
Rendering in Blender

- Process of turning 3D scene into 2D image
- For animated films, it can take hours to render a single frame
- Important steps:
  - Camera placement
  - Lighting
  - Materials
  - Render with low quality settings, repeat above steps
  - Render with high quality settings

https://www.amazon.com/Disney-Pixar-Talking-Amazon-Exclusive/dp/B01B7OY2M4
Exporting from Blender

Objects and scenes built in Blender can be saved in a file for use elsewhere

- Many file formats out there
- Support growing for GLTF (*Graphics Language Transmission Format*)
- Online archives of free 3D models
Questions

1. Have you downloaded and installed Blender yet?
   *If not, you should do so before Monday’s class.*

2. Outline how you could scale an object in Blender to three times its size along the x and y axes, but hold it steady in z.
   *Select the scale tool and begin a mouse drag. Type Z then 3.*

3. Find the URL of an online source of free 3D models
   *Examples include free3D.com, archive3D.net, artist-3d.com, etc.*
   *Some government agencies provide public domain 3D models related to their operations, including the [Smithsonian](https://www.smithsonian.gov), [NASA](https://www.nasa.gov), and more.*
Review

After watching this video, you should be able to…

- List three major strategies for hidden surface removal
- Identify polygons outside the view or that face away from a camera
- Understand the z buffering algorithm and its limitations
- Apply offsets in Three.js to mitigate z fighting
- Perform basic operations in Blender
- Work with models from 3D archives

Music: [https://www.bensound.com](https://www.bensound.com)