CG Assignment 5

Q) How can you detect hidden surface using z buffer approach? What is the limitation of the z-buffer? which algorithm removes this drawback & how?

Detecting Hidden surfaces using z-buffer (Depth Buffer) Approach:

→ The z-Buffer algorithm is a well-known method for hidden surface removal in 3D rendering.

→ It involves using a seperate buffer (known as the z-buffer or depth buffer) to keep track of the depth (z-coordinate) of each pixel on the screen.

→ The z-buffer is initialized with the maximum depth value for all pixels before rendering begins.

→ As each object or polygon is rasterized onto the screen, the algorithm checks if a new pixel has a similar z-coordinate (closer to the camera) than the value stored in the z-buffer.

→ If it does, the pixel's depth value is updated in the z-buffer, the pixel's color value from the corresponding object or polygon is stored in the frame buffer.

→ This process ensures that only the closest visible surface ~~conditions~~ contributes to the final image, effectively detecting & removing hidden surfaces.

Limitation of z-Buffer Approach

1 → High Memory requirement.
→ With increasing screen resolutions & complex scenes, the memory req'd for the z-buffer can become quite large, especially in real-time rendering applications.

2 → May suffer from aliasing effects.

Alg. that removes this Drawback.

a) Tile-Based Deferred Rendering (TBDR).

→ TBDR is a defered rendering technique that optimizes the rendering process by dividing the screen into smaller tiles or clusters.

→ Instead of storing a seperate depth value for each pixel, TBDR stores depth information for each tile, reducing memory usage significantly.

→ The process involves several steps:

1) Tile Division
2) Geometry Pass
3) Lighting Pass

Q) Diff bet Area subdivision Method & Depth Buffer Approach for detecting visible surfaces in 3D?

1. Area subdivision Method
2. It is a class of hidden surface removal algorithms that divide the 3D scene into smaller regions or areas and perform visibility tests to determine which surfaces are visible in each region.
3. These methods involve analyzing the scene geometry & ~~determining~~ visibility at the object level.

Characteristics of OSM
1. Object-level Visibility
2. Pre-processing
3. Visibility Determination
4. Object sorting
5. order-dependent

6. These methods are also known as scan-line algorithms because they work by scanning the screen or viewport horizontally or vertically to identify visible surfaces.

characteristics
1. scan-line-based
2. Visibility test
3. Memory efficient
4. order independent

Depth-Buffer Approach (z-buffer)

characteristics

1) z-buffer implementation
2) Memory requirements
3) Pixel-Based
4) Efficient Hidden Surface Removal
5) Order-Independent

Comparision

Area subdivision	Depth Buffer
1) uses diff. techniques to solve the hidden surface removal in 3D rendering	
2) work by dividing the screen.	2) uses seperate buffer to keep track of the depth of each pixel

Q3) How does OSM approach work differently than the ISM approaches for detecting visible surfaces in 3D?

OSM

→ Object-Space Methods

→ In OSM, visibility tests are performed in the object space before the actual rasterization & rendering process takes place.

→ These methods involve analyzing the scene geometry & determining visibility at the object level.

characteristics

ISM

→ Image-Space Methods

→ In ISM, visibility tests are performed at the pixel level during or after the rasterization process.

→ These methods involving analyzing the rendered image or framebuffer to identify visible surfaces.

Characteristics

1) Pixel-level visibility
2) Z-Buffer Algorithm
3) Pixel sorting
4) Memory requirement
5) order - Independent

Q) Diff bet Depth buffer & Depth sorting Approach for detecting visible surfaces in 3D.

- Z-buffer

Key characteristics

1) Pixel-Based

Depth sorting Approach

→ It is an object-level hidden surface removal technique that involves sorting the objects in the scene based on their distance from the camera.

→ By rendering objects from the closest to the farthest, the method ensures that only the visible parts of each object contribute to the final image.
→ This techniques is also known as the Painter's Algorithm.

key characteristics:
1) Object-level sorting
2) Visibility Determination
3) Rendering order Matters
4) order-dependent
5) No Additional buffer

Comparision:
1) The main diff ~~approaches~~ lies in their fundamental Principles & the level at which visibility is determined.
2) The DB approach is pixel-based & determines visibility at the pixel level using a z-buffer to compare depth values.
3) The Depth sorting approach is object-based & determines visibility at the object level by sorting objects based on their distance to the camera.
4) The DB requires additional memory for the z-buffer but not on DS

Q) Explain the working mechanism of Back face detection algorithm.

Steps

1) Surface Normal Calculation
2) View vector calculation.
3) Dot product calculation
4) check Dot product Sign
5) Discard Back Faces
6) Render front faces

Q) Why is it required to remove hidden surfaces in 3D viewing?

→ Because:

1. Realism
2. Visibility
3. Efficiency
4. Depth perception
5. Lighting & shading
6. Culling & optimization

Q) How does hidden surface removal algorithm bring visual realism in graphical scenes?

1) Accurate Occlusion
2) Depth perception
3) Realistic Lighting & shading
4) Reduced clutter
5) Efficient Rendering
6) Realistic Overlapping & Intersections

(Q8) What are list priority algorithms? Explain the working mechanism of Depth sorting approach

→ list priority alg. are a class of hidden surface removal alg.'s used in computer graphics to determine the visible portions of 3D objects or surfaces in a scene.

→ These algorithms organize the objects or polygons in a priority list based on their depth or distance from the camera.

→ The Depth sorting approach is one of the list priority algs used for hidden surface removal.

Working

1) Scene Geometry & Camera Setup
2) Distance Calculation
3) Object sorting
4) Rendering order
5) Depth buffer usage
6) Order Dependency

Q) How is the scanline approach diff from z-buffer approach for hidden surface removal?

Scan-line approach

→ It is a method for hidden surface removal that involves scanning the screen or viewport horizontally (or vertically) & analyzing the intersections of scan lines with the edges of polygons in the scene.

→ It operates on a polygon-by-polygon basis & identifies visible surfaces by considering the intersections of the scan lines with the polygon edges

key charac

1. scan line Processing
2. Polygon sorting
3. Edge Table (ET) & Active Edge Table (AET)
4. Pixel-Based
5. Interior-Filling

Scan-line	Z-buffer
1) Used for 2D rendering	1) Used for 3D rendering
2) Used for interior polygon filling	2) focuses on hidden surface removal.
3) does not require an additional buffer.	3) requires.