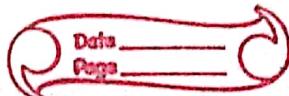


Ch Assignment 5



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 separate 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, and 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 reqd 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 deferred rendering technique that optimizes the rendering process by dividing the screen into smaller tiles or clusters.

→ Instead of storing a per-pixel 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 b/w Area subdivision Method & Depth Buffer Approach for detecting visible surfaces in 3D?

• Area subdivision Method

- 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.
- These methods involve analysing the scene geometry & determining visibility at the object level.

Characteristics

1. Object-level Visibility
2. Pre-processing
3. Visibility Determination
4. Object rendering
5. Order-dependent

of OSM

• 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
Compositing

A disadvantage of Depth Buffer

- 1) Uses diff. techniques to solve the hidden surface removal in 3D rendering.
- 2) Work by dividing the screen.

No 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 Method
- 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 Method

- In ISM, visibility tests are performed at the pixel level during or after the rasterization process.
- These methods involve analyzing the generated 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

(a) Difference between Depth buffer & Depth sorting Approach (or 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 furthest, the method ensures that only the visible parts of each object contribute to the final image.
- This technique 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

Comparison:

- 1) The main diff approach lies in their fundamental Principles & the level at which visibility is determined.
- 2) The DSS 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 DS requires additional memory for the Z-buffer but not in DSS.

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 & Interactions

(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?

Scanline approach

- It is a method for hidden surface removal that involves scanning the screen or warps 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

- 1) used for 2D rendering
- 2) used for interior polygon filling
- 3) does not require an additional buffer.

Z-buffer

- 1) used for 3D rendering
- 2) focuses on hidden surface removal.
- 3) requires.