UNIT – 4

Ray Casting

Ray casting is a fundamental technique in computer graphics, particularly in rendering 3D scenes. It involves tracing rays from a viewer's perspective (or a camera) into the scene to determine the colors, shading, and visibility of objects in the scene. Ray casting is used for generating 2D images of 3D scenes, and it forms the foundation of more complex rendering techniques like ray tracing.

1. Basic Concept of Ray Casting

The main idea in ray casting is to determine what objects in a scene are visible from a certain viewpoint by shooting rays through each pixel on the screen. These rays are projected from the viewpoint and continue into the scene, interacting with objects (such as spheres, planes, or complex 3D models) to compute visibility, color, and shading for each pixel.

- **Process**:
  - A ray is cast from the camera (or eye) through each pixel on the screen into the 3D scene.
  - The ray is tested for intersections with objects in the scene.
  - If an intersection is found, the closest intersecting object determines the pixel's color, based on lighting and material properties.

2. Steps in Ray Casting

The ray casting process involves several steps, each contributing to generating the final image.

a. Ray Generation

- Rays are generated from the camera or eye through each pixel in the view plane. Each pixel on the screen represents a direction from the camera's viewpoint, forming a viewing frustum.

Ray Casting

b. Ray-Object Intersection

- Once rays are generated, each ray is checked for intersections with objects in the scene.
- Mathematical techniques are used to test intersections for different shapes. For example:
  - **Sphere Intersection**: A quadratic equation is solved to find intersection points.
  - **Plane Intersection**: A simple linear equation is solved to see where the ray intersects the plane.
  - **Complex Shapes**: Ray intersections with triangles, polygons, and meshes often involve more advanced calculations.

c. Shading and Lighting Calculation

- After determining the closest intersection point, the color of the pixel is calculated by considering lighting in the scene.
- **Shading Models**:
  - **Flat Shading**: A single color is applied based on the angle between the light source and the surface normal.
  - **Phong or Gouraud Shading**: Provides smooth shading by interpolating colors and considering light direction, material properties, and surface normals.
- **Lighting Calculation**: Determines how much light each pixel receives based on light sources, ambient light, and shading techniques.

d. Depth Calculation

Ray Casting

- For each pixel, the depth (distance from the camera) of the closest intersecting object is stored to ensure that only the nearest object is visible at that pixel, handling cases where multiple objects overlap from the viewpoint.

3. Mathematical Foundations of Ray Casting

Ray casting relies on linear algebra and geometry for intersection tests and calculating rays.

- **Ray Equation**:
  - A ray RRR can be defined with an origin OOO and a direction DDD as: R(t)=O+tDR(t) = O + tDR(t)=O+tD
  - Here, ttt is a parameter that determines how far along the direction vector DDD the ray travels.
- **Intersection with Spheres**:
  - For a sphere with center CCC and radius rrr, the intersection with a ray R(t)R(t)R(t) involves solving: ||R(t)−C||2=r2||R(t) - C||^2 = r^2||R(t)−C||2=r2
  - Expanding this equation yields a quadratic equation in ttt, which can be solved to find the intersection points.
- **Intersection with Planes**:
  - For a plane with a normal NNN and a point P0P_0P0 on the plane, the intersection with a ray R(t)R(t)R(t) is found by solving: (P0−O)·N=t(D·N)(P_0 - O) \cdot N = t(D \cdot N)(P0−O)·N=t(D·N)
  - If ttt is positive, the intersection lies in front of the camera.

4. Advantages of Ray Casting

Ray Casting

Ray casting has several benefits that make it a popular technique for certain types of rendering and applications:

- **Simplicity**: It's straightforward to implement and understand. It primarily involves ray-object intersection calculations and is thus conceptually simpler than full ray tracing.
- **Visibility Determination**: Ray casting directly determines which objects are visible in a scene, making it ideal for rendering scenes with complex depth information.
- **Flexibility**: Ray casting can handle different shapes and geometries by implementing appropriate intersection tests.

5. Limitations of Ray Casting

Despite its simplicity, ray casting has limitations:

- **Lack of Realistic Effects**: Unlike ray tracing, ray casting does not handle reflections, refractions, shadows, or global illumination. The technique is primarily focused on visibility and basic shading.
- **Computational Complexity**: For each pixel, ray casting involves checking intersections with every object in the scene, which can become computationally expensive for large scenes.
- **Limited Shading**: Basic ray casting supports only flat or simple shading models, which may result in less realistic images.

6. Applications of Ray Casting

Ray casting is used in a variety of applications where quick visibility determination and basic shading are sufficient. Some of these applications include:

- **Rendering Engines**: Early rendering engines, particularly in real-time applications, utilized ray casting for efficiency.

Ray Casting

- **Hidden Surface Removal**: Ray casting is often used to determine visible surfaces in a 3D scene, helping eliminate hidden surfaces from view.
- **CAD Software**: In computer-aided design (CAD), ray casting can be used to display hidden line views or shaded models with basic lighting.
- **Volume Rendering**: Ray casting is used in volume rendering to visualize 3D data, such as medical imaging, where rays pass through a 3D volume to sample data points.
- **Simple Simulations**: Ray casting is also used in applications where simplicity is crucial, such as certain mobile games or simulation environments that do not require complex visual effects.

7. Comparison with Ray Tracing

Ray casting and ray tracing are often mentioned together, but they serve different purposes:

Feature	Ray Casting	Ray Tracing
Primary Purpose	Visibility and basic shading	Realistic rendering with effects
Reflections	Not supported	Supported
Refractions	Not supported	Supported
Shadows	Not supported	Supported
Performance	Generally faster	Slower, due to additional ray tracing per pixel