Ray Tracing

Overview of Ray Tracing Techniques:

Light Tracing: This method starts at the light source and shoots rays to find where they intersect the scene. The problem with light tracing is that it attempts to solve a global problem of how light disperses in a scene, which can be inefficient as many rays may not hit the camera.

Path Tracing: The more efficient method, path tracing, starts by shooting rays out of the sensor (camera) and traces them until they reach a light source. This process is more aligned with the goal of determining how light enters the camera.

Ray Tracing Pipeline:

The pipeline involves three main operations: ray generation, intersection, and shading.

Ray Generation: Rays are shot out of the camera.

Intersection: This step finds where the ray intersects scene geometry. It is the only global interaction with the scene.

Shading: Involves determining the light absorbed or reflected at the point of intersection.

Ray Generation and Intersection:

Rays are represented as a geometric construct with an origin and direction vector.

The intersection process is a crucial part of ray tracing. It reduces the complexity of handling geometry to a single query: finding where a ray intersects with the scene.

Camera Types in Ray Tracing:

Perspective Camera: Mimics a pinhole camera, shooting out a cone of rays through a plane.

Orthographic Camera: Uses parallel rays hitting a plane. This type of camera is useful for rendering images without perspective distortion.

Primitive Intersections:

Intersecting rays with primitives like spheres, planes, and triangles is foundational in ray tracing.

For example, intersecting a ray with a sphere involves solving a quadratic equation, while intersection with a plane is simpler.

Barycentric Coordinates in Triangle Intersections:

Barycentric coordinates are used to parametrize points inside a triangle, based on the triangle's vertices.

They are essential for determining if a point lies within a triangle during the ray-triangle intersection process.

Recursive Nature of Ray Tracing:

The process of ray tracing can be recursive. After determining the light incident at a point, new rays may be generated for further intersections and shading, repeating the process to simulate multiple reflections and transmissions.

Explicit Representation:

This is a form of representation where there is a parameter that can be varied to generate points on a 3D object. It's also referred to as a parametric representation.

For example, Ray(t)=Origin+t×Direction in the case of a ray, the 't' parameter in its equation is used to generate different points along the ray.

Implicit Representation:

In an implicit representation, someone provides a 3D point, and there is a condition to check whether that point is on the object or not.

Unlike explicit representation, there is no parameter to vary in order to generate points on the object. The focus is on whether a given point satisfies a certain condition to be part of the object.