**Abstract:**

The purpose of this project is to introduce ray tracing as technique to render 3D images and models. It deals with the algorithm and method involved in creating a ray tracer. The advantages of ray tracing over scan line rendering and LEE’s rasterization and the downsides of using a simple ray tracer and how Kd trees can be used to overcome the shortcomings.
Image Rendering:
An image is generated from a given model in rendering. The positioning and arrangement of the model affect the generation of an image from it. The point of view of the model affects the amount of light falling on the face of the model. The appearance of surface of the image is highly dependent on that of the actual texture of the model, as the texture of any surface plays a pivotal role in the amount of light reflected, refracted or diffused by the object.

Purpose of rendering images:
Rendering is done for the purpose of simulating the naturally occurring effects, like the interaction of light with various forms of matter. Particle system uses a large number of graphical objects to simulate funny phenomena which are otherwise very difficult to render. Particle systems include natural phenomena like smoke, fire, water, explosions. Volumetric sampling like fog, dust. In optics, 3D rendering is used in a caustic network to check the simulated light rays being reflected or refracted by a curved surface or object, like ripples seen in a swimming pool.

Rendering techniques:
- Raseterization: This considers the object in the scene and then projects them to form an image. There is no means to generate a point-of-view perspective effect.
- Ray Casting: This considers the scene as observed from a specific point-of-view, calculating the observed image based on only geometry and basic optical laws of reflection to on intensity.
- Radiosity: It simulates the way in which reflected light, instead of just reflecting to another surface, also illuminates the area around it. It produces more realistic shading and captures the ambience of an indoor scene.
- Ray Tracing: Similar to ray casting, but employs the advanced optical simulation, to obtain more realistic results, but the trade off comes in the form of speed which is often large magnitudes of that in rasterization.

Advantages:
- Ray Casting: Real time Simulations.
- Radiosity: Real time rendering method, used from beginning to end to create animated 3D films.
- Ray Tracing: Extension of ray casting and scan line. Beneficial where complex and accurate rendering of shadows, reflection and refraction of light. High level of detailing is observed.

Disadvantages:
- Rasterization: Quality of image is low.
- Ray Casting: Ignorance to detail.
- Radiosity: Complex objects are slow of simulate due to repeated recursion. Higher level of realism is achieved through ray tracing.
- Ray Tracing: As it is a brute force method, it is too slow for real time.

How does ray tracing overcome the shortcomings of other rendering techniques:
Techniques like ray casting and scan line rendering while quick, ignore details. They are incapable of rendering complex images like shadows, reflection and refraction of light. Ray tracing on the other hand, uses techniques that ensures that maximum level of accuracy is achieved.
**Ray Tracing:**

This technique extends scan line and ray casting. It handles complicated objects and is also known as pixel-pixel-rendering. The objects can be described mathematically. It is based on a number of randomly generated samples form a model.

The samples are rays of light intersecting with objects that are in between the pixel and the light source and calculating the effect of the light on the pixel. It is beneficial where complex and accurate rendering of shadows, refraction or reflection is concerned. Multiple rays are generally shot for each pixel and traced, not just into the first object of intersection, but rather through a number of sequential bounces using the law of optics. The incident and reflection angles are equal. Once the ray encounters a light source and once a set of limiting number of bounces has been evaluated, surface illumination at that final point is evaluated. Changes along the way through the various bounces evaluated to estimate a value is observed at the point of view. This is repeated for each sample, for each pixel.

At each point of intersection, multiple rays can be spawned. But this method is too slow for real time rendering. As brute force is employed, the amount of time taken render the image is often very high and can go up to even hours. Efforts have been put in order to reduce the rendering time taken by the ray tracer by eliminating calculations on areas that do not require high levels of calculation.

Efforts at optimizing to reduce the number of calculations needed in portions of a work where detail is not high or does not depend on Ray Tracing features, has lead to a realistic possibility of wider use of Ray Tracing.

**Method:-**

- Shoot ray from camera through each pixel in image plane.
- Camera is at (0,0,0) and points in the negative direction.
- Determine the image coordinates in 3D.

**Determining Phong Color:-**

- Use phong model.
- Use shadow rays.
- Compute specular reflection and specular transmission by recursion.
Angle of reflection: incoming angle = outgoing angle.

**Algorithm:-**

1. For each pixel(x,y), fire a ray from COP through (x,y).
2. For each ray and object, calculate closest intersection.
3. For closest intersection point p.
   - Calculate surface normal.
   - For each light source, fire shadow ray.
   - For each unblocked shadow ray, evaluate local Phong model for that light and add color to that pixel.

- Ray surface intersections and illumination calculations are to be evaluated recursively.

**Mathematics of ray tracing:-**

Phong shading equation: Color $C = \sum_{\text{lights}} \text{specular} + \text{diffuse} + \text{ambient}$ components

$$C = (K_s \sum l \cdot (R \cdot E)) + (K_d \sum l \cdot (N \cdot L)) + (K_a I_a)$$
Ray tracer output:-

![Ray tracer output image]

Rendering time:-

2:47 hours

Simulation Conditions:-

Disadvantages:-

Simple ray tracer takes a really long time to render as it all renderer objects (triangles) are checked for to see whether they interfere with the pixel under consideration. It cannot be used for real-time applications.

How is it overcome:-

This can be overcome by using Kd trees. As a Kd tree splits the image according to what lies exactly in front of the pixel, less time is wasted on checking irrelevant renderer objects (triangles).

Conclusion:-

The simple ray tracer used for rendering here evaluates using a pixel-by-pixel rendering. All other objects in the scene are checked to see whether or not they affect the pixel.