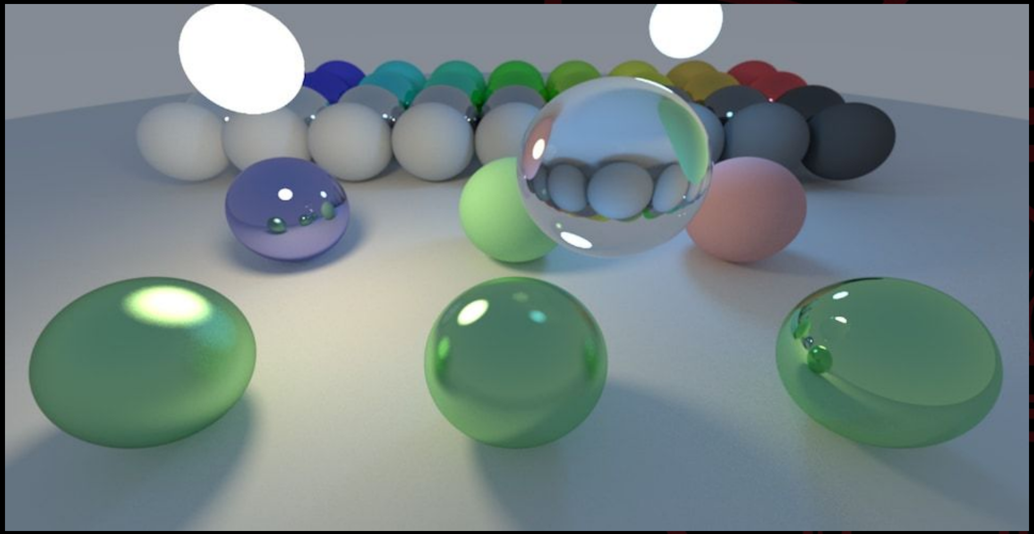
USC University of Southern California

CSCI 596

Project Presentation

3D Rendering : Path Tracer



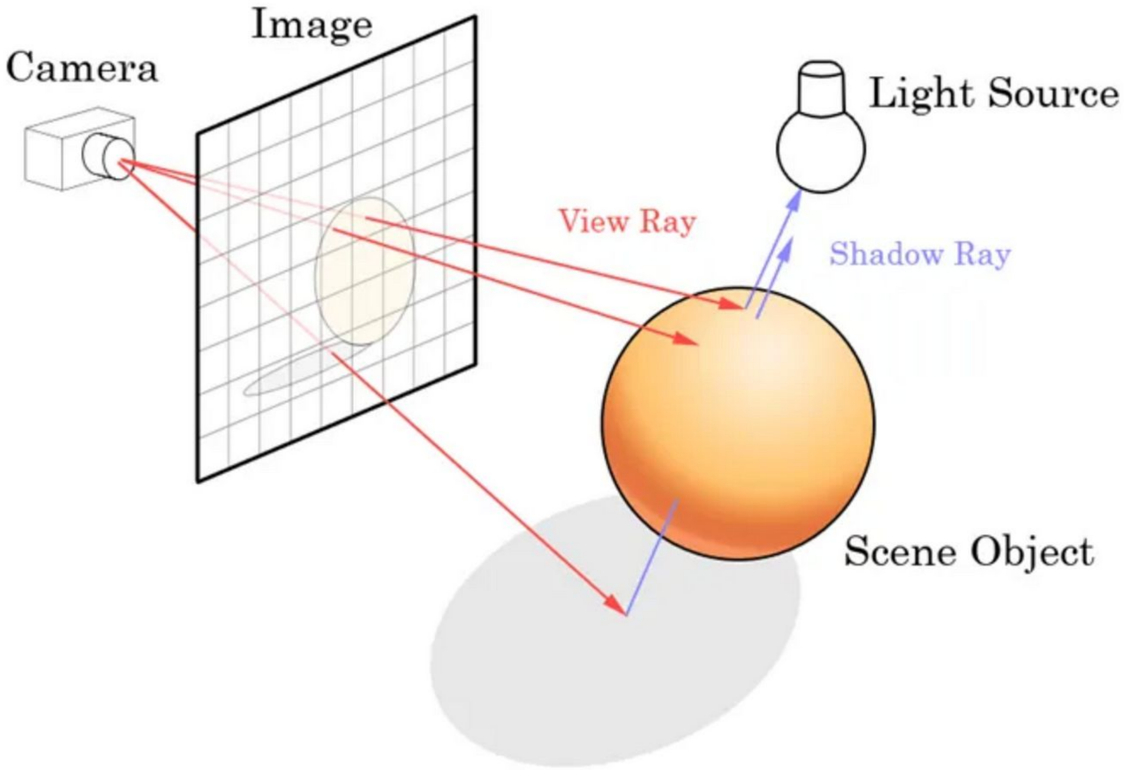
CSCI 596 : Scientific Computing and Visualization

Fall 2023

USC

Sneha Bandi

What is a Path Tracer?



- A Path Tracer is a **3D rendering technique** used in computer graphics to simulate the way light interacts with surfaces in a 3D scene.
- It is a **global illumination algorithm** that models the **realistic** behavior of light by tracing the paths of light rays as they travel through a **virtual environment.**
- The primary goal of path tracing is to produce visually **realistic** images by simulating the complex **interactions** of light with different surfaces, materials, and objects.
- Path tracers are used for creating **photorealistic images** and animations in applications such as film production, architectural visualization, and scientific visualization.

USC

Problem

- Rendering realistic 3D scenes involves **simulating the complex behavior** of light as it interacts with surfaces and materials.
- The traditional rendering equation captures this phenomenon, but it's **recursive nature** makes it challenging to solve directly, necessitating **numerical approximation methods** like Monte Carlo.
- Path tracing is particularly effective in capturing realistic lighting scenarios, including effects like soft shadows, caustics, and color bleeding. However, it can be **computationally intensive**, and require optimizations

Objective

The planned objective of this project is to develop a high-performance Path Tracer in C++ to address the challenges of rendering 3D scenes. Path Tracing, a cutting-edge technique, will be employed to simulate light interactions accurately.

The solution includes implementing robust ray tracing, advanced rendering features, and optimization for real-time rendering, catering to scientific computing and visualization needs.

Motivation

Why Scientific Computing and Visualization?

- Scientific Visualization: Accurate representation of complex scientific phenomena.
- Monte Carlo Technique: Essential in scientific computing for numerical approximation.

Why Path Tracer?

- Realism in Visualization: Path Tracing provides a realistic approach for visualizing scientific data and simulations.
- Photorealistic Rendering: Aids researchers in creating visually compelling images.

Background and Mathematical Explanation

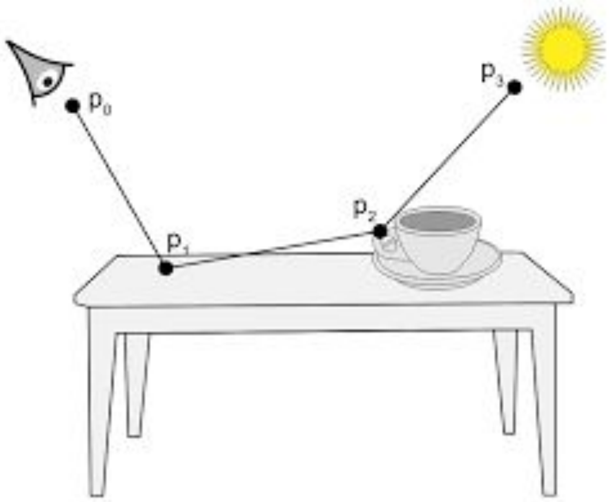
Path Tracing and the Rendering Equation

- The rendering equation mathematically represents light interactions in 3D scenes
- It shows amount of light sent out from a point in a given direction.
- Numerical approximation methods are required due to the equation's recursive nature.

$$L_o(x, w_o) = L_e(x, w_i) + \int f_r(x, w_i, w_o) L_i(x, w_i)(w_i . n) dw_i$$

Bidirectional Reflection Function (BRDF)

- The function $f_r(x_0, w_i, w_0)$ is the "Bidirectional Reflection Function".
- The Bidirectional Reflection Function (BRDF) determines how much light is sent out in a specific direction.
- It governs how much of the light received at the point x from direction w_i is sent out in direction w_0.



USC

Background and Mathematical Explanation

Monte Carlo Technique

- Monte Carlo Techniques: Key in sampling points efficiently for approximation
- Widely used for approximating complex mathematical expressions.
- Heuristics are used to select "good points" for accurate approximation.
- Balancing accuracy with computational efficiency is crucial.

Reformulated Rendering Equation

- The reformulated equation expresses light as a sum of contributions from rays of increasing length.
- Randomly generating and accumulating ray paths is essential for realistic rendering.

$$L(p_1->p_0) = L_e(p_1->p_0) + \int L_e(p_2->p_1) f(p_2->p_1->p_0) dA(p2) +$$

$$\int\int L_e(p_3->p_2) f(p_3->p_2->p_1) dA(p2) \times f(p2->p1->p0) dA(p3) dA(p2) + \ldots$$

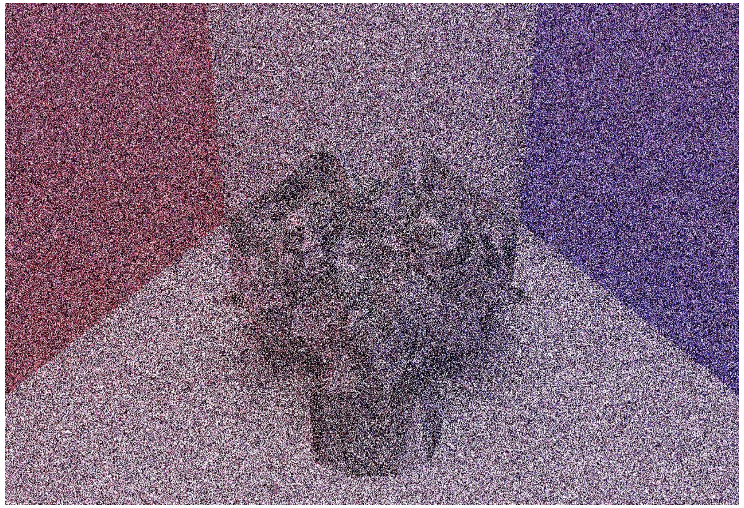
Experiment Design and Implementation

- Implementation of a C++ program considering the recursive nature of the rendering equation.
- Integration of relevant libraries for geometry representation, shading, and acceleration.
- Monte Carlo Integration: Implementation for efficient sampling.
- Detailed Experimentation with path tracing

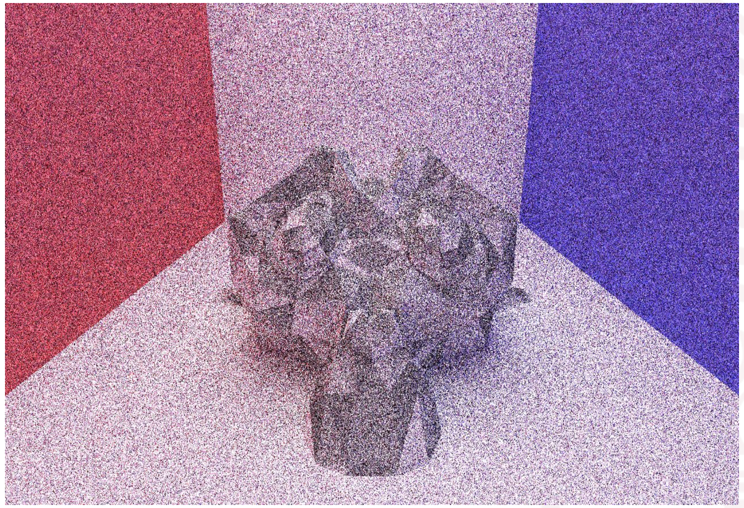
Evaluation Metrics

- Visual Quality: Comparison against reference images to assess realism and accuracy.
- Performance: Detailed analysis of rendering times for complex scenes and animations.
- Feature Completeness: Support for various materials, camera models, and light sources.
- Code Quality: Review for maintainability, scalability, and adherence to best practices.

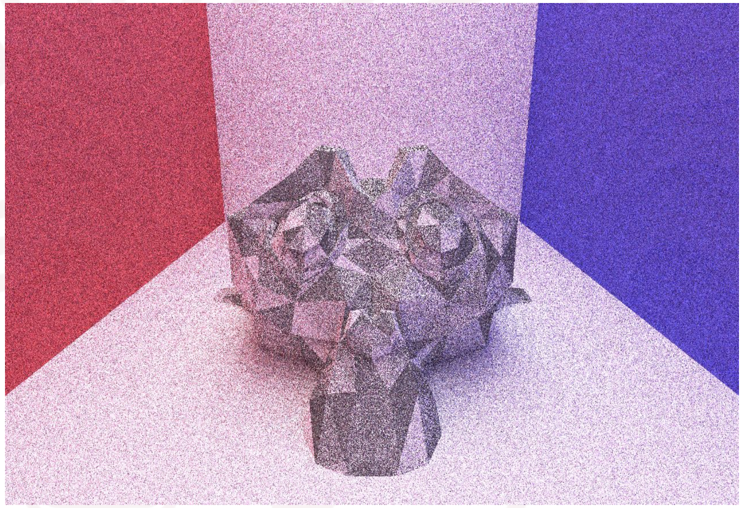
Results and Analysis



1 sample



4 samples



16 samples

Experiment to be contd. with higher sample sizes
Like 128, 256, 1024 samples

Computational Cost and Time

Samples	Seconds elapsed
1	1
4	17
16	61

Machine used:

AMD Ryzen 5700U

16 CPU cores

Threads per core: 2

Total threads: 32

Clock max: 4.3 Ghz

Clock min: 400 Mhz

Conclusion and Future Work

The projects aimed to create a high-performance Path Tracer, aligning with scientific computing and visualization requirements. By achieving the outlined objectives and meeting evaluation criteria, the resulting tool will be valuable for in fields requiring accurate and visually compelling 3D renderings.

Future work could involve further optimizations (Like : Metropolis Light Transport), additional rendering features (Like : bidirectional path tracing), and implementation and exploration of extensions like real-time interactive rendering.

Thank You

USC