

Wow! RTX Your Next Big Idea.

Hook (Manuscript)

This, is a raytracing engine.

By the end of this video,
not only will you be able to create this in either OpenGL, Metal, or Vulkan.

But also you will understand the theory behind it,
such that you can implement raytracing in any context.

Intro (Manuscript)

No less, Raytracing has revolutionized how we simulate graphics.

Years ago, I was like you, unsure how to simulate light with math or how graphics cards could accelerate this process.

I'm Sol Midnight, and this is Blockglow.

Join me on my journey to enlighten you about raytracing.

open chime

Theory (Extemporaneous)

- Core Concepts
 - Raytracing casts multiple rays from camera → scene
 - Raycasting uses just one ray
 - Each ray determines part of final color through recursive bounces
 - More rays = better geometry precision, but higher computational cost
 - Ray types include primary (eye), shadow, reflection, and refraction rays
 - Rays follow physics-based rules for surface interaction
 - BxDF functions (Bidirectional x Distribution Function):
 - BRDF: Bidirectional Reflectance Distribution Function

- Describes how light reflects off surfaces
 - Examples: matte (diffuse), glossy, mirror surfaces
 - Takes incoming light direction, outputs reflection direction
 - Defines material's reflective properties
- BTDF: Bidirectional Transmittance Distribution Function
 - Describes how light passes through materials
 - Examples: glass, water, translucent plastics
 - Handles refraction based on material density
 - Defines material's transparent properties
- Combined as BSDF (Bidirectional Scattering Distribution Function)
 - Unified model for both reflection and transmission
 - Critical for realistic material rendering
 - Used in physically based rendering (PBR)
- Geometry Approaches
 - Hardware Geometry:
 - Uses traditional triangles/polygons
 - Native GPU support, faster for simple scenes
 - Memory efficient for detailed models
 - Harder to modify geometry dynamically
 - Accelerated by BVH (Bounding Volume Hierarchy)
 - Hardware RT cores optimize triangle intersection tests
 - Meshlets enable efficient geometry processing
 - Voxel-Based:
 - Divides space into 3D grid of cubes
 - Great for destructible environments
 - Easier to modify geometry in real-time
 - Higher memory usage for high detail
 - Popular in games like Minecraft
 - DDA (Digital Differential Analyzer) algorithm for traversal
 - RLE (Run-Length Encoding) for compression
 - Sparse Voxel Octrees (SVO):
 - Hierarchical voxel structure
 - More memory efficient than raw voxels
 - Better for large, detailed scenes
 - Used in some modern raytracers
 - Adaptive subdivision based on detail level
 - Efficient empty space skipping

- Supports dynamic LOD (Level of Detail)
- Light Behavior
 - Scene brightness doesn't affect required ray count
 - Water creates additional rays through reflection/refraction
 - Randomly scattered rays approximate real light behavior
 - Caustics occur when curved surfaces focus light rays
 - Monte Carlo integration for light sampling
 - Importance sampling reduces noise
 - Multiple Importance Sampling (MIS) combines strategies
 - Global Illumination captures indirect lighting

History (Extemporaneous)

- 1500s: Albert Durer - Early 3D scene projection
- 1968: Arthur Appel creates raycasting at IBM
- 1980: Turner Whitted introduces recursive raytracing
- 1984: Cook introduces distributed raytracing
- 1986: Kajiya presents the rendering equation
- 1995: Henrik Jensen introduces photon mapping
- 2010s: Minecraft raytracing
- 2018: NVIDIA RTX brings real-time raytracing
- 2022-23: Octane and Hexane raytracers released

Modern Applications (Extemporaneous)

- Film Industry examples: Monster House, Cloudy with Meatballs
- Hybrid Solutions: Mix rasterization + raytracing
- Denoising reduces required rays while maintaining quality
- Path tracing simulates realistic light transport
- Real-time GI in games using hardware RT
- Architectural visualization with instant feedback
 - Real-time material/lighting changes for client reviews
 - Sun studies and daylight analysis
 - Integration with BIM for accurate visualization
- Scientific visualization for accurate light simulation
 - Optical systems design and testing
 - Medical imaging simulation
 - Solar and atmospheric modeling

- Virtual production with LED walls
 - Real-time environment rendering on LED displays
 - Camera tracking for perspective correction
 - Interactive lighting between real and virtual elements

Advanced Techniques (Extemporaneous, not covered by demo)

- Bidirectional path tracing: Traces from both light source AND camera
 - Quality improves with more samples
 - Perfect for offline rendering when time isn't critical
 - Metropolis Light Transport (MLT) for difficult light paths
 - bidirectional path tracing with statistical randomness
 - Volumetric rendering for participating media
 - Subsurface scattering for translucent materials
 - Adaptive sampling focuses rays where needed
 - ReSTIR for efficient light sampling
-

API (Impromptu)

OpenGL

Vulkan

Metal