IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

IRP

Hybrid Relightable 3D Gaussian Rendering (sdmay25-40)

Jackson Vanderheyden, Brian Xicon, Luke Broglio, Ethan Gasner, Kyle Kohl
Advisor: Simanta Mitra

IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

Project Overview

- Turn videos into realistic 3D models.
- Render composite scenes containing 3D Gaussian and polygon models.
- Packaged into Unity, a popular game engine, for easy adoption.

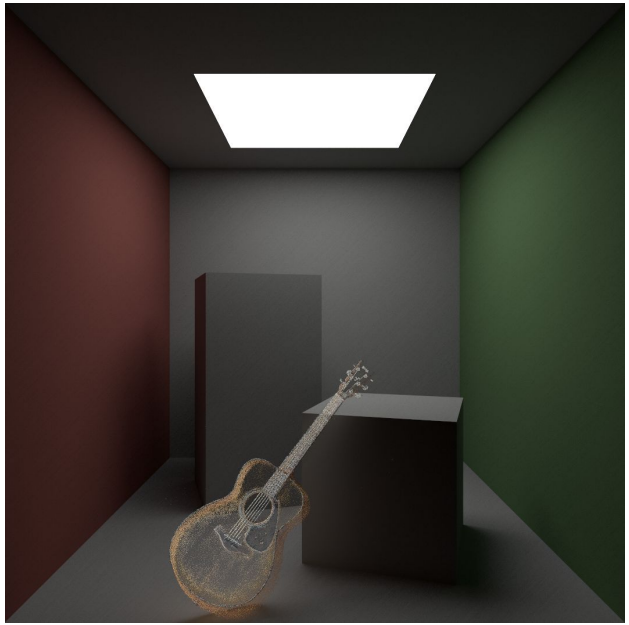


Figure 1: Hybrid Scene consisting of a Cornell box and a Gaussian model of a guitar.

IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

Intended Users

Users

- 3D Artist
- Neural Radiance Fields (NeRF) Users
- Non-Technical User

Industries

- Entertainment
- Marketing
- Real Estate

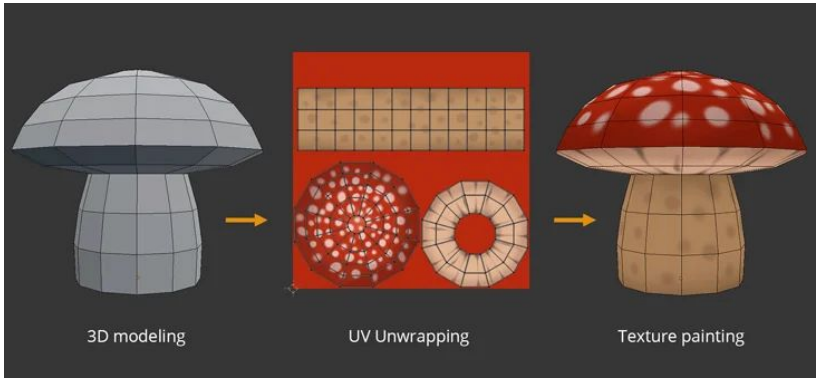


Figure 2: 3D modeling workflow for creating a mushroom
Source: https://game-ace.com/blog/3d-modeling-in-unity/

IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

Solving Engineering Problems

- Two major engineering problems were solved when choosing the system's techniques.
- The first was choosing a novel view synthesis method.
  - The system uses 3D Gaussian Splatting because it is the most modern practice and produces the highest quality models.
- The second was choosing a rendering approach (path tracing or rasterization).
  - The system uses path tracing to increase visual fidelity with realistic light propagation.

IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

Professional Responsibility

- The design fit in and provides value to society because it mitigates the challenges of manually re-creating real-world objects.
- Automatic reconstruction of 3D models reduces the reliance on manual modeling, raising concerns about job displacement; however, the system wouldn’t replace workers, as it lacks fine-grained creative control a 3D artist could achieve.

IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

Functional Requirements

- Create point clouds using Structure from Motion (SfM) techniques from videos or video frames
- Input SfM point clouds for Gaussian model optimization.
- Provide Gaussian model with position, opacity, covariance, and spherical harmonics for 3D Gaussian parsing.
- Input parsed 3D Gaussian model for Bounding Volume Hierarchy (BVH) generation.
- Provide Unity mesh objects for BVH generation.
- Input 3D Gaussian models, triangle meshes, mesh material data, and camera parameters for hybrid path tracer.
- Present the rendered texture from pipeline to Unity's camera.

IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

Non-Functional Requirements

- The system should be downloadable via Unity's package manager.
- The system should run in real-time at 30 FPS with dynamic updates.
- The system should generate an optimized Gaussian model from video in under an hour.
- The system should be CUDA compatible NVIDIA graphics card
- The system should work with python and these python libraries: plyfil, ipython, torch, opencv-python, torchvision, and tkinter

IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

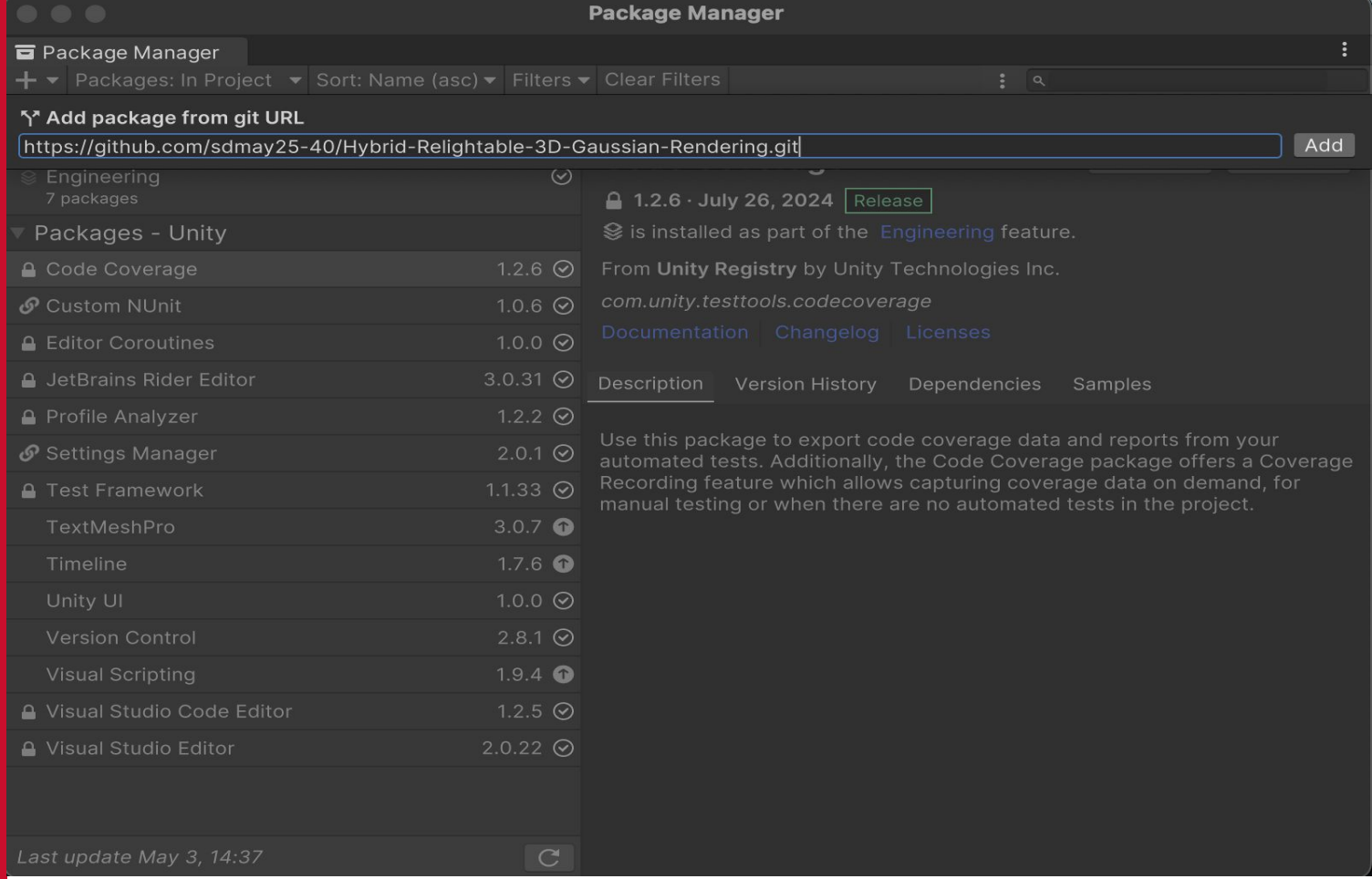


Figure 3: Unity package manager

IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

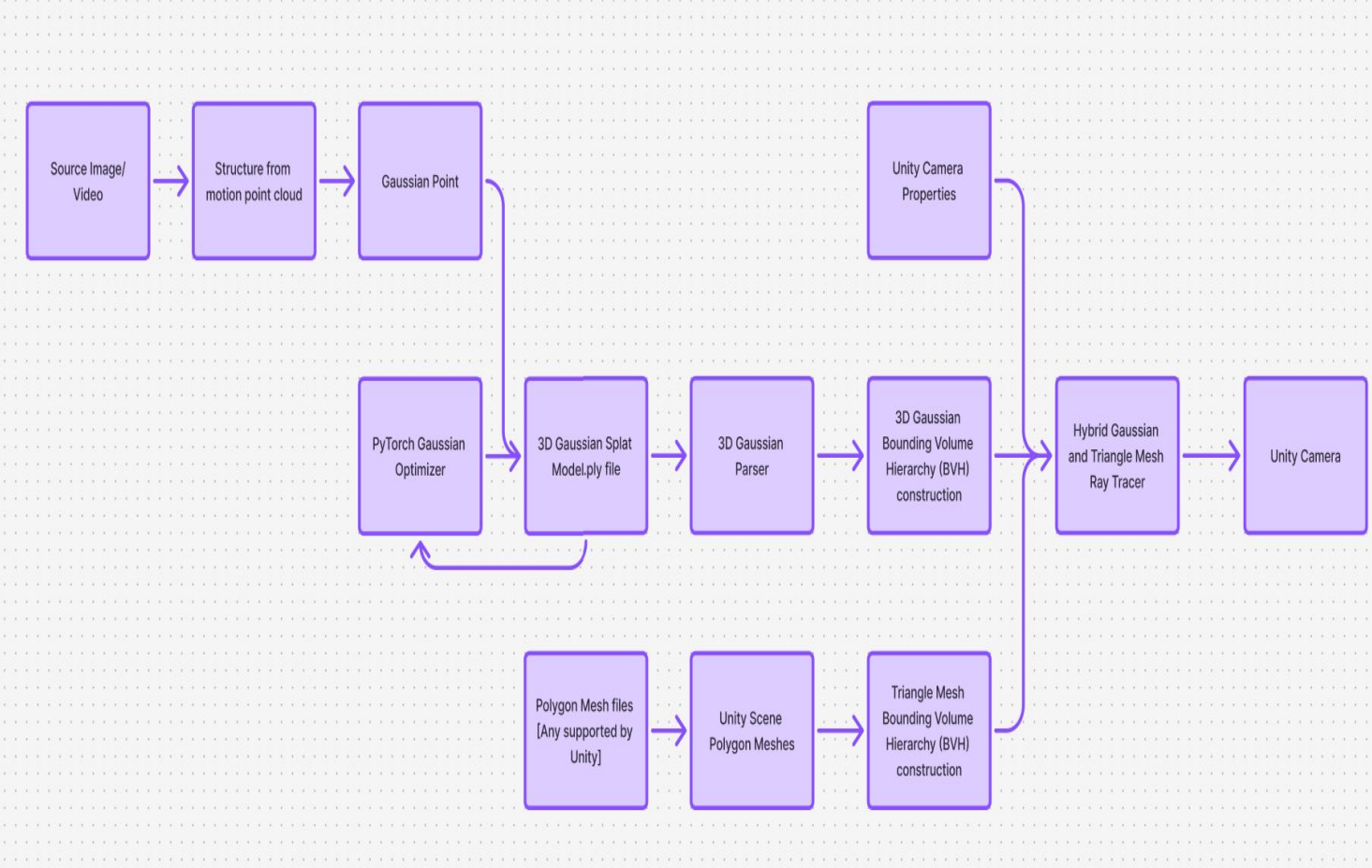


Figure 4: Flow diagram of our project pipeline

IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

3D Gaussian Splatting

- To be able to create models we use 3D Gaussians.
- Think of a 3D Gaussian as a blurry dot in space.
- By combining and reshaping multiple 3D Gaussians we can build 3D objects.
- 3D Gaussians can have multiple different variables including position, spherical harmonics, opacity, and covariance.

Figure 5: The Gaussian point optimizer splits to better fit the intended shape.
Source: https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/

IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

Gaussian Showcase

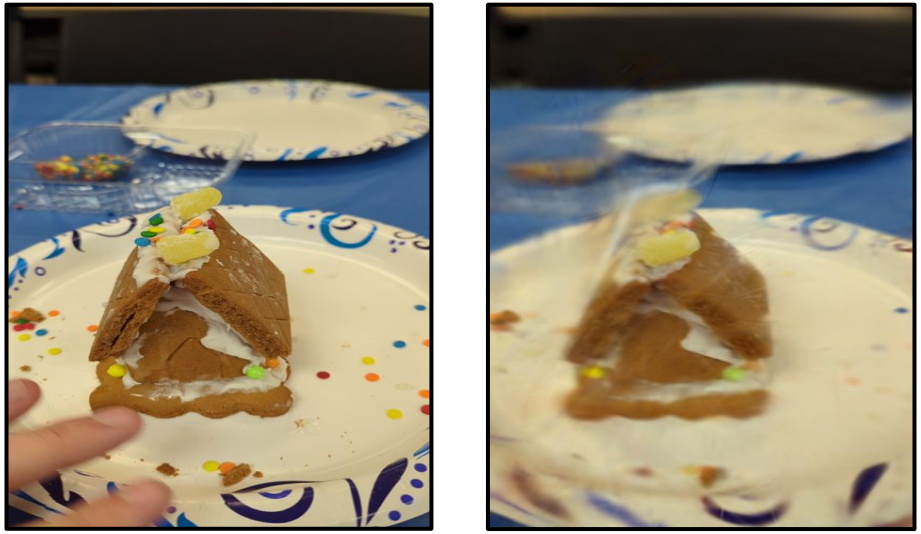


Figure 6: The image on the left showcases the original image and the image on the right showcases the model made up of Gaussians.

IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

Bounding Volume Hierarchy (BVH)

- In order for our raytracer to run in real time we need to make the process of finding intersections as efficient as possible
- A BVH speeds the process up by allowing the raytracer to efficiently check if the ray intersects the region of space a triangle is in.
- The BVH is made of a BST of Axis Aligned Bounding boxes.
- We implemented a BVH for both triangle and gaussian models

IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

Bounding Volume Hierarchy (BVH)

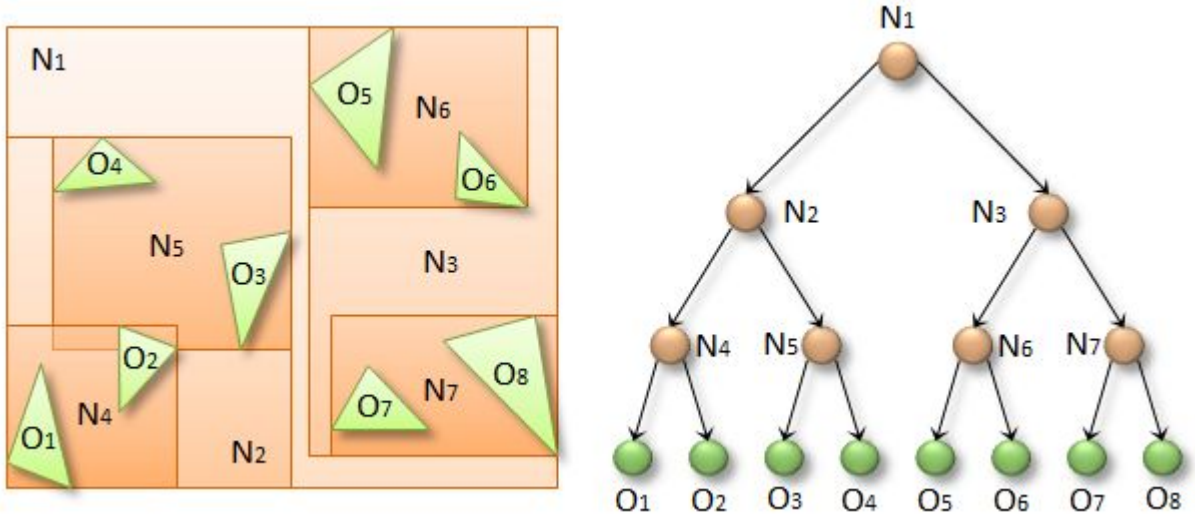


Figure 7: Diagram showing how an example BVH works
Source: https://developer.nvidia.com/blog/thinking-parallel-part-ii-tree-traversal-gpu/

IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

Hybrid Path Tracer

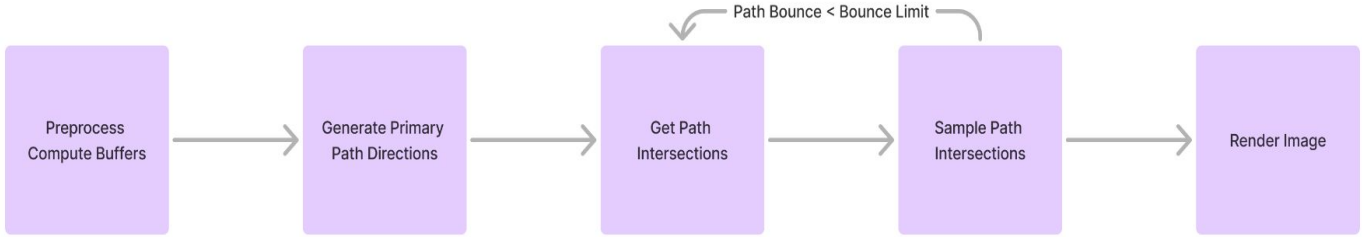
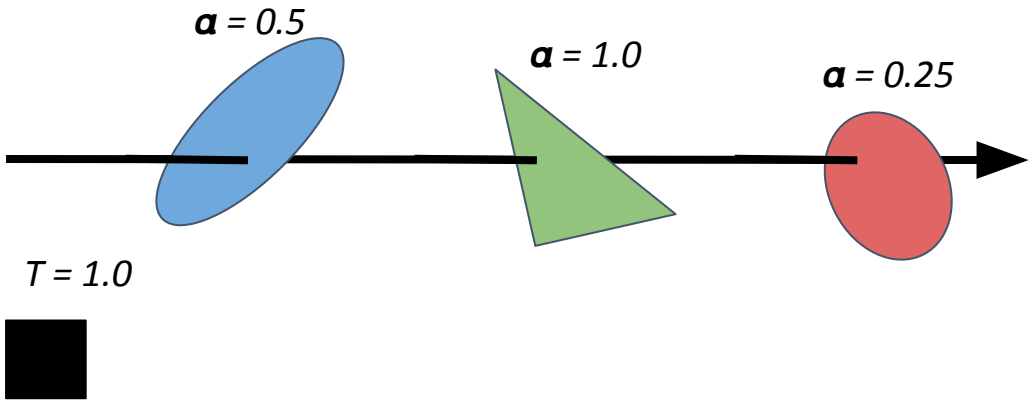


Figure 8: High-level overview of rendering pipeline

IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

Determining Intersections in Composite (Hybrid) Scenes

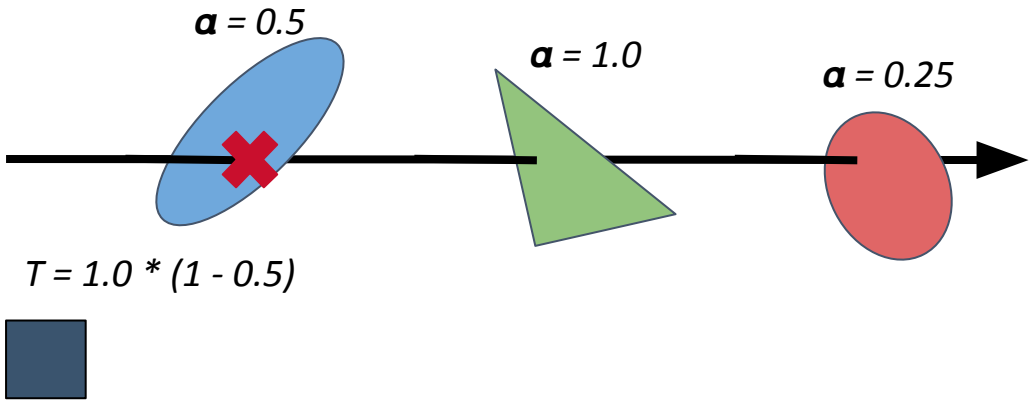
- The system traces the scene until the path's accumulated transmittance reaches a minimum T value.
- Intersection contributions are weighted by opacity and transmittance.



IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

Determining Intersections in Composite (Hybrid) Scenes

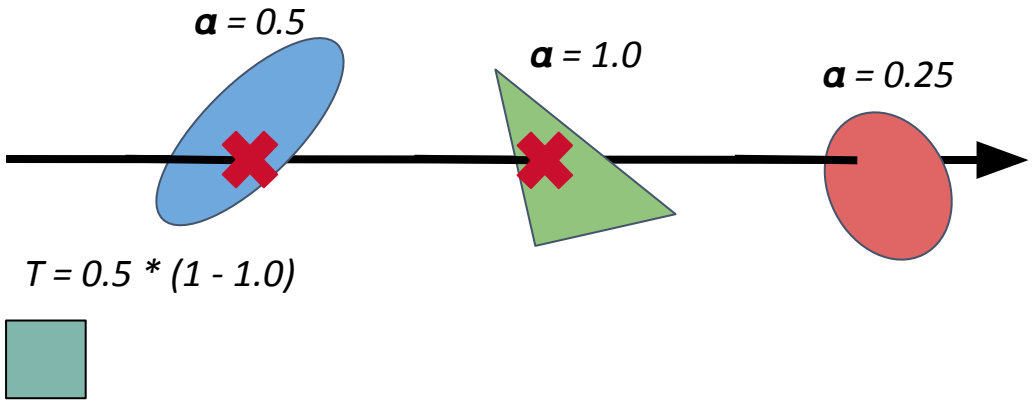
- The system traces the scene until the path's accumulated transmittance reaches a minimum T value.
- Intersection contributions are weighted by opacity and transmittance.



IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

Determining Intersections in Composite (Hybrid) Scenes

- The system traces the scene until the path's accumulated transmittance reaches a minimum T value.
- Intersection contributions are weighted by opacity and transmittance.



IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

The Rendering Equation

- Used to calculate color at intersection points.

$$L_o(x, V) = L_e(x, V) + \int_\Omega f_r(x, L, V) L_i(x, L)(L \cdot N) dL$$

Outgoing Light Emissive Light Reflected Incoming Light

IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

The Rendering Equation

- Used to calculate color at intersection points.

$$L_o(x, V) = L_e(x, V) + \frac{1}{n} \sum_{i=1}^{n} \frac{f_r(x, L, V) L_i(x, L)(L \cdot N)}{pdf(L)}$$

Outgoing Light Emissive Light Reflected Incoming Light

IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

Risk Identification and Mitigation

- Integrating the machine learning and ray tracing components.
  - Mitigated by communicating between teams and making modifications when necessary.
- Bad data sets being used for training
  - To mitigate this we used well known, high quality datasets.
- Less than real time performance.
  - Mitigated by the use and rigorous testing of BVHs.

IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

Testing

- Unit Testing:
  - pixel-level image comparisons with visual tools to validate ML outputs
  - Render single Gaussians to complex scenes for accuracy.
- Interface Testing: Validate PyTorch data handoff to rendering within Unity
- Integration testing: Validate added functionality improves performance
- System Testing: Render complex scenes derived from original video input.

Machine Learning Testing

ML Applications

- Structure from Motion
  - 3D structures from 2D images
- Gaussian Point Optimizer
  - Position, opacity, covariance, spherical harmonics (color)

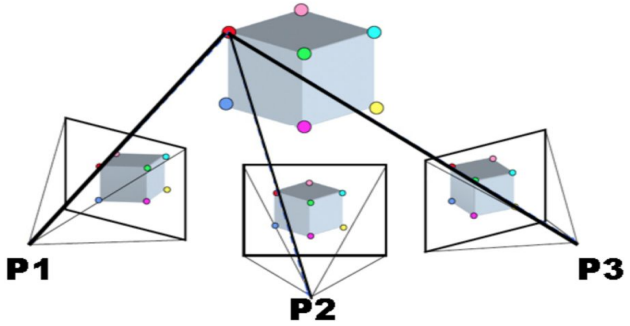


Figure 9: Structure for Motion (SfM) generates an approximate 3D shape using images from different perspectives.

IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

Structure From Motion

- **COLMAP** is a 3D reconstruction library for SfM, supporting feature extraction, matching, and point cloud generation.

Figure 10: Point cloud of a horse statue

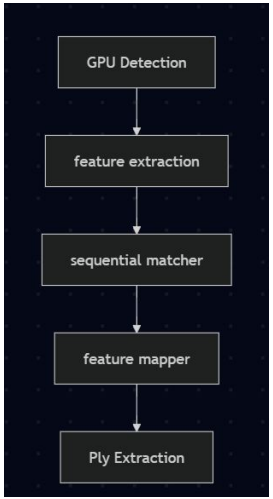


Figure 11: SfM pipeline

IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

Gaussian Point Optimizer

- Optimizes parameters like the position and covariance of our Gaussians.
- Retrieves the sparse point cloud from SfM.
- Compares the pixels of the 2D images from the sparse point cloud to ground truth images.
- Configures the parameters of the Gaussians to match the ground truth images.

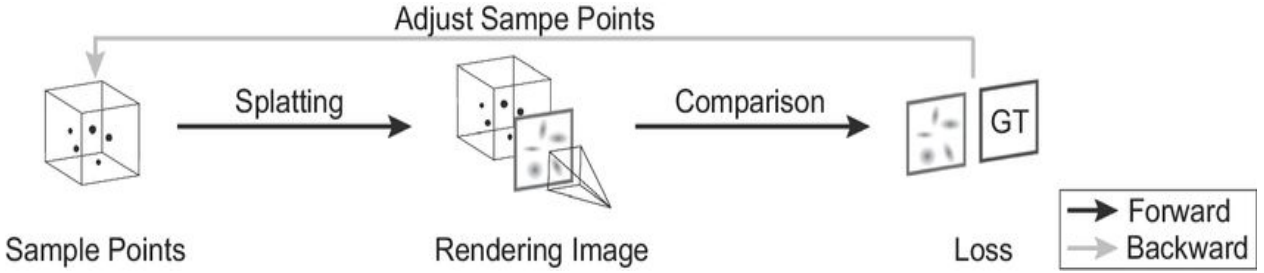


Figure 12: Flow chart of Gaussian point optimizer loop
Source: https://www.researchgate.net/figure/Pipeline-of-Gaussian-Splatting-Pipeline-of-Gaussian-Splatting-First-initial-sample_fig4_378904778

IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

Gaussian Point Optimizer

Figure 13: Truth image compared to model of a horse statue during training

IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

Gaussian Point Optimizer

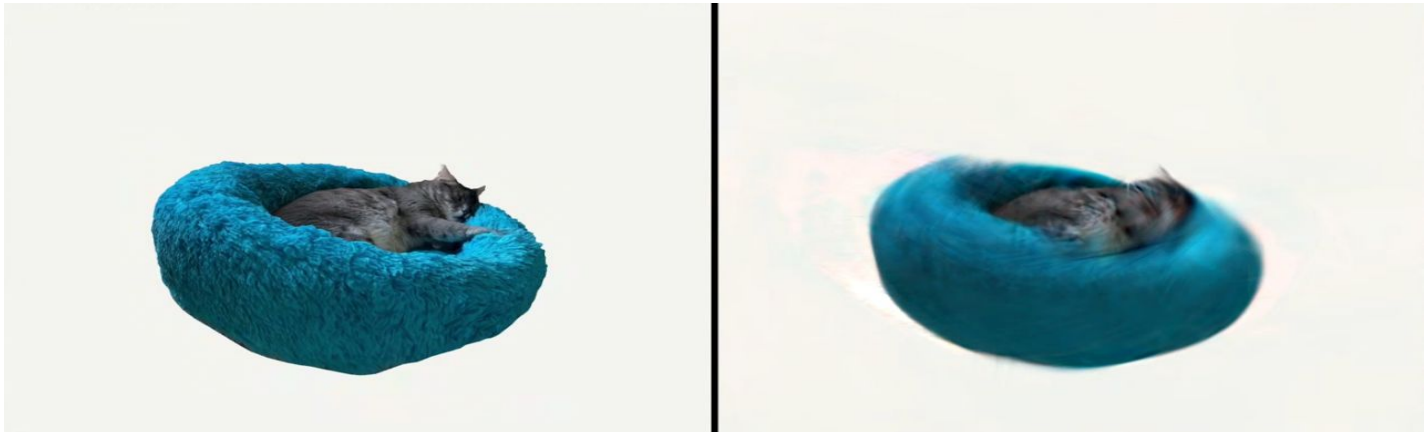


Figure 14: Truth image compared to model of a cat during training

IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

Testing the 3D Gaussian Parser

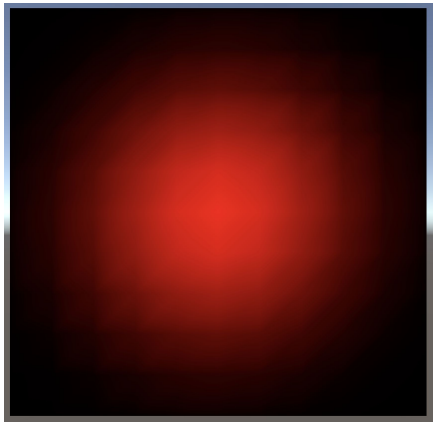


Figure 15: A Gaussian with its center in the middle a plane

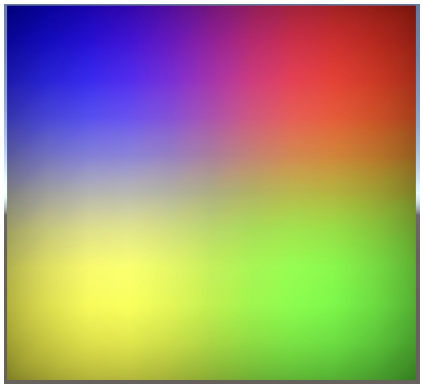


Figure 16: Four different colored Gaussians placed on a plane such that they overlap at the edges.

Figure 17: A Gaussian centered on a plane stretched (scaled in the y direction) and rotated.

IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

Bounding Volume Hierarchy (BVH)

- Our triangle BVH is implemented in C# and constructs an individual BVH for every Unity mesh in the scene.
- Our gaussian BVH is also implemented in C# and constructs a BVH bottom up by first creating an AABB for every Gaussian.

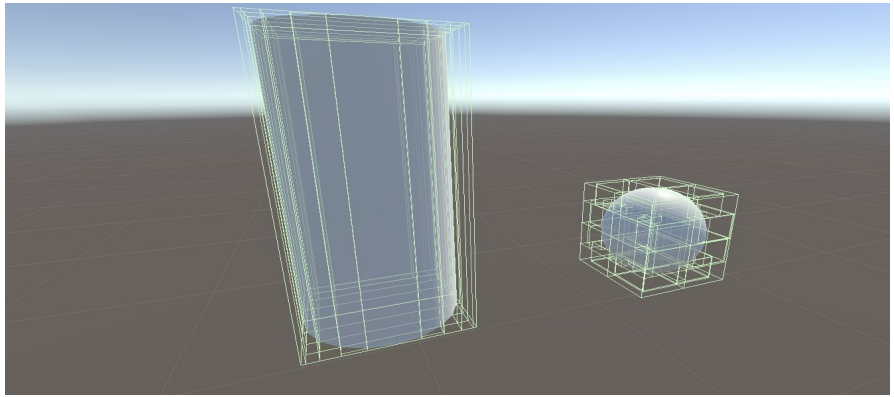


Figure 18: The bounding boxes created by our implementation for a cylinder and a square

IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

Testing the BVH

- To test the BVH the performance of the same scene was measured when rendered with and without a BVH.
- There was a major performance increase when using the BVH.

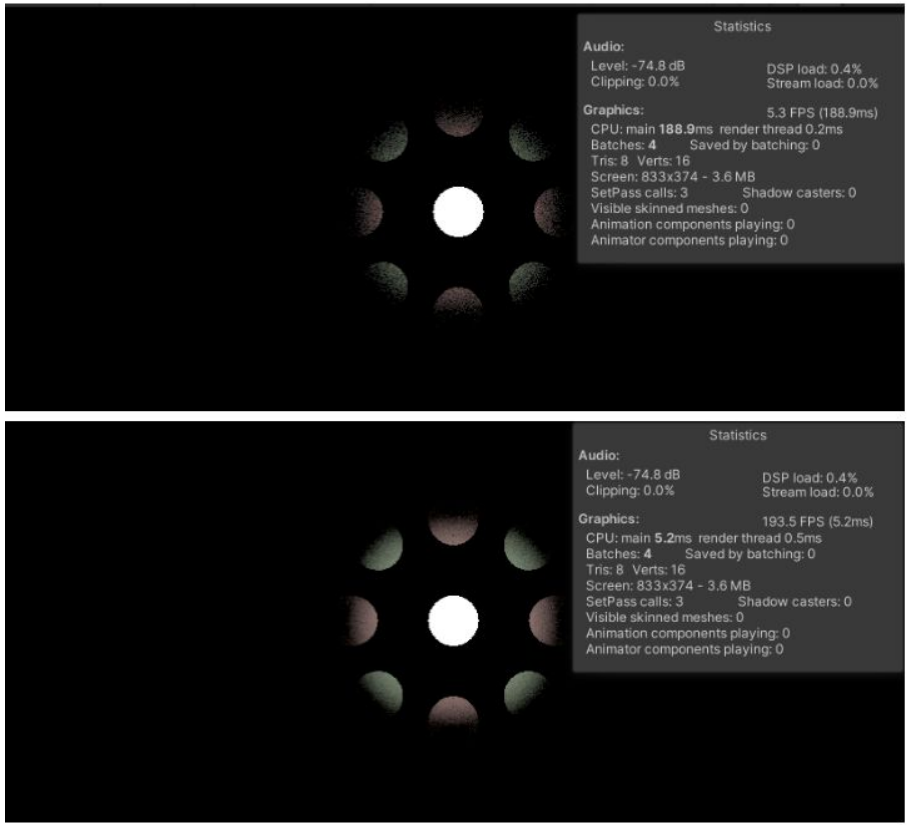


Figure 19: Performance comparison, in FPS, between a scene without a BVH (top) and one with a BVH (bottom)

IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

Triangle Mesh Ray Tracer Prototype

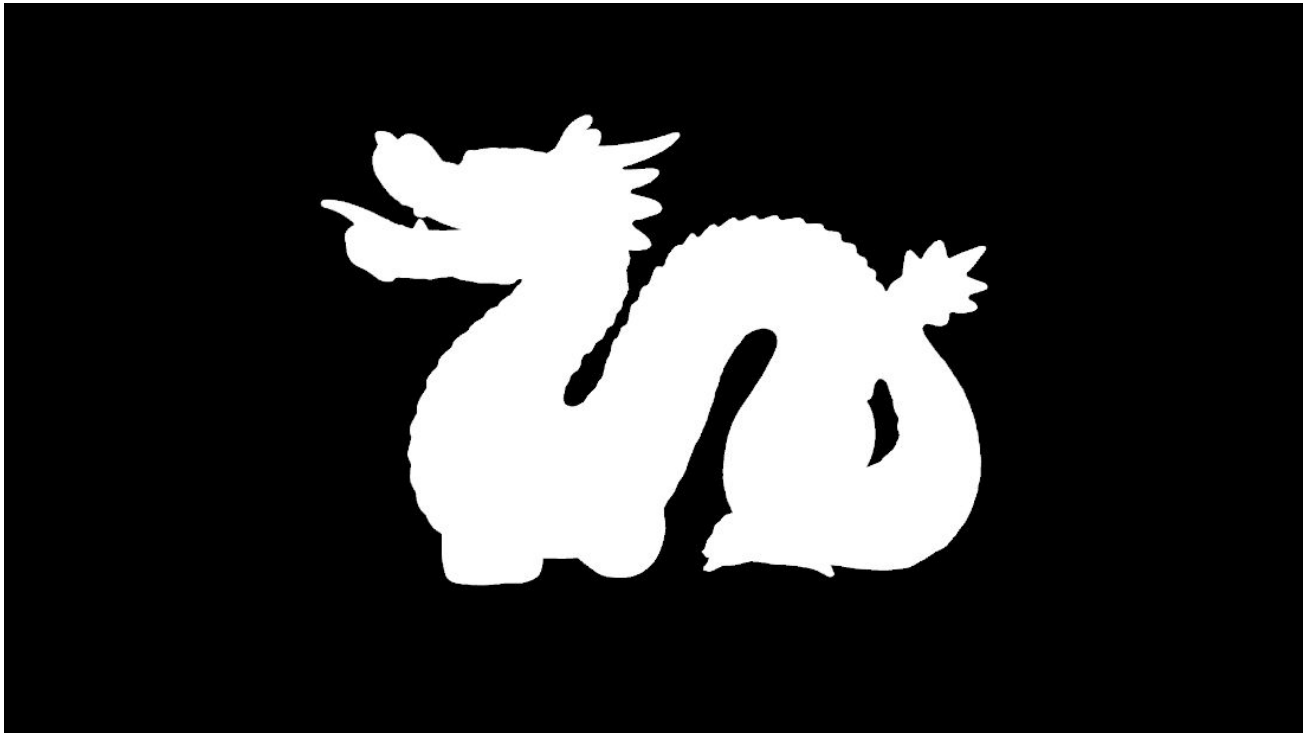


Figure 20: Rendered image of Stanford Dragon demonstrating ray-triangle intersections and material albedo.

IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

Testing Physically Based Rendering

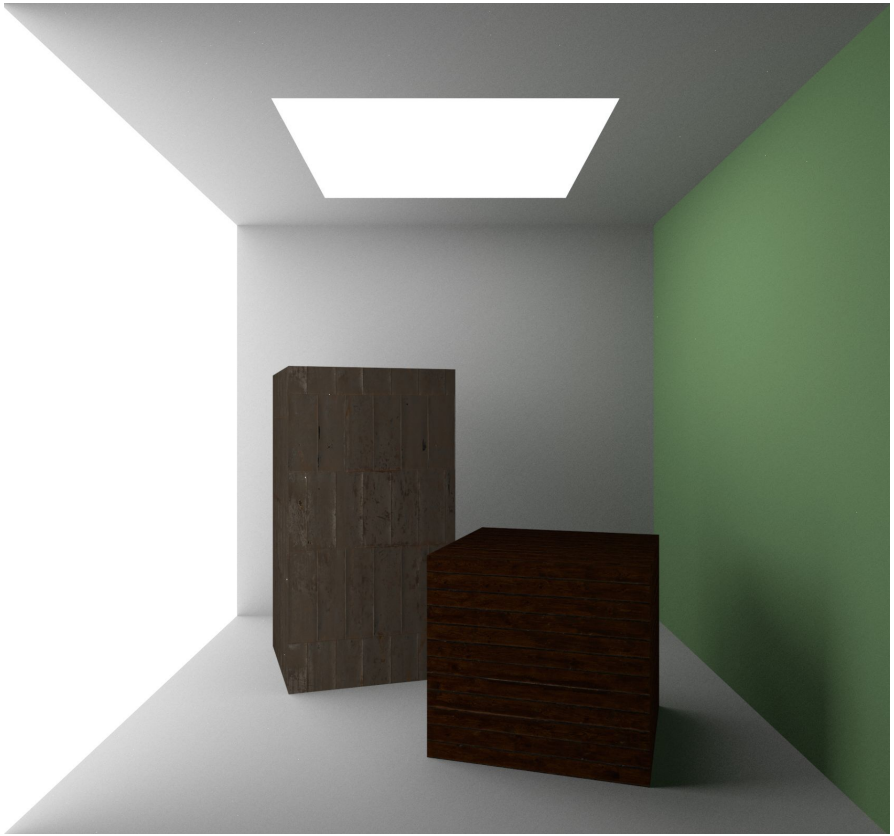


Figure 21: Modified Cornell Box with PBR materials

IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

Implications and Next Steps

Implications

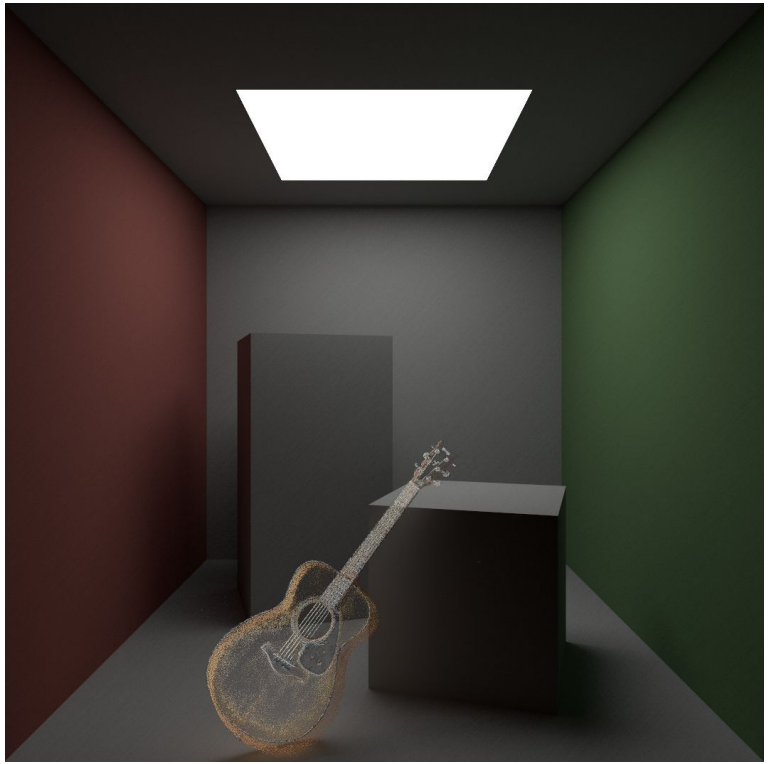
- The program significantly speeds up the time needed to generate real-world 3D model.

Next Steps

- Improve the training speed
- Increase the quality of the Gaussian models
- Implement the material predictor

IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

Questions?



IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

THANK YOU