

An Unbiased Hybrid Rendering Approach to Path Guiding

Jia-Wun Jhang and Chun-Fa Chang
 National Taiwan Normal University, Taiwan

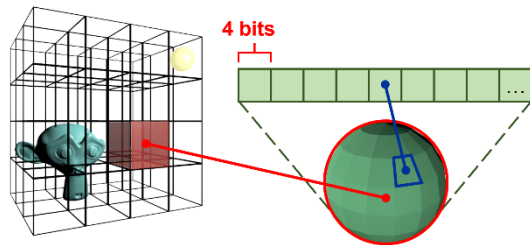
Introduction

We present a hybrid rendering approach that combines ray tracing with rasterization. Typically a hybrid renderer produces the initial frame with rasterization and GPU shaders, then adds the missing global illumination effects (such as the shadow, reflection, and refraction) to some objects by ray tracing selected regions of the initial frame. However the other regions that are not enhanced by ray tracing may still miss the subtle global illumination effects such as indirect lighting, especially when the results are compared to the ground truth images of an unbiased Monte Carlo path tracer.

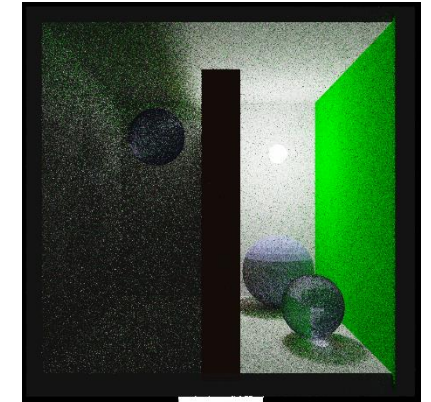
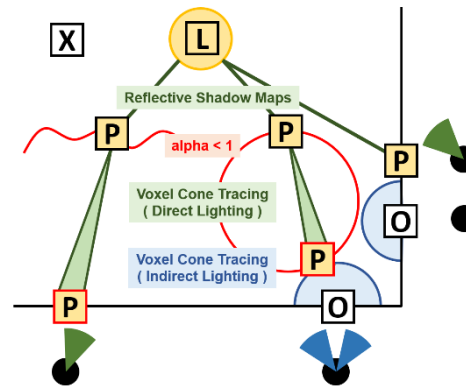
We take a different approach to combine ray tracing and rasterization, in which our final output images are still produced with a path tracer. We leverage the GPU rasterization to build the necessary data that are required for path guiding [MGN17] [DHD20], thus improves the convergence of our path tracer. We borrow the ideas of Voxel Cone Tracing [CNS11] and implement it in GPU shaders. However we do not use Voxel Cone Tracing for rendering the output frame. Instead we use it to build the path guiding data. The advantage of our proposed hybrid approach is that it maintains the unbiased results of a Monte Carlo path tracer while incurring relatively small performance hit in path guiding.

Unbiased Hybrid Rendering

Our unbiased hybrid renderer consists of a Monte Carlo path tracer which samples the reflection directions based on path guiding. The path guiding improves the convergence of the path tracer by providing the information about the surrounding lighting conditions at the path vertices. We use a rather simple irradiance record for path tracing, which divides the scenes into uniform-sized voxels and stores the lighting information from all incoming directions in each voxel. The simplicity of our irradiance record allows us to build its content with GPU rasterization methods such as the voxel cone tracing. We first divide the scene space into uniform sized voxels. Each voxel contains an irradiance record that stores the lighting information from all incoming directions. Before the path tracer starts, we fill in the irradiance records in all voxels using Voxel Cone Tracing.



We leverage the GPU rasterization in our hybrid renderer to speed up the processing of irradiance records that are used for path guiding. First we divide the scene space into uniform sized voxels. Each voxel contains a tag that is first initialized to either X or O: X for empty voxels, O for voxels that contain objects. We also set the tag to L for voxels that contain a light source, then use the reflective shadow map [DS05] to find the voxels that could be directly illuminated and set their tags to P. Finally we process the indirect lighting by finding the voxels with the O tag and perform voxel cone tracing to build the irradiance records that are used for path guiding.



Results

We show the results of our hybrid renderer using two scenes as shown here. The first is a Cornell box with a center wall to emphasize the indirect lighting in the left side. The second is a underwater scene to emphasize the caustics. All figures are rendered at the size of 1024x1024 and show the convergence after accumulating 30 frames. They are rendered on a notebook PC with an NVIDIA GeForce GTX 1070 GPU.

