Real-time Global Illumination Independent Study Project Report

Che Sun

1 Introduction

Real-time Global Illumination (RTGI) is an active research area in which many rendering techniques have been developed to achieve the goal of solving light transport equation in real-time by utilizing modern GPU's powerful programmable pipeline.

In this independent study project I focus on reading papers on several popular RTGI methods. They are: Screen-space based techniques, VPLs (Many-lights) based techniques, Rasterization based techniques and Voxelization based techniques. In some papers, these techniques are combined together to develop more complex RTGI methods.

2 Sub-techniques implemented

Many important sub-techniques are introduced by the papers I read. Following are all implemented in my programming projects, some of them are very new techniques.

Geometry Buffer

By using multiple render targets, scene geometric properties are stored in a bunch of 2D textures which are used later in techniques such as deferred lighting and SSDO. In my implementation, diffuse reflectivity (albedo), world position, world normal, linear depth are stored.

Spherical Harmonics

SH uses a set of function basis to approximate incoming radiance in a spatial position.

Deferred Lighting

Modern graphics systems such as game engines employ this technique to support many-lights based rendering.

Random sampling textures

Monte Carlo integration based techniques employ randomness to sample incoming radiance. I use C runtime random number generator and Halton sequence to generate random sampling textures.

Geometry-aware filtering

Position and normal information can be used to detect huge differences between adjacent pixels to achieve better image filtering. I learned this technique from SSDO paper.

Simple HDR-tone mapping

Now I know all light energy transfer can be modeled with HDR floating-point data. Before outputting to frame buffer, an HDR-tone mapping filter must be applied to the final image data.

Reflective Shadow Maps

RSM is the fundamental tool describing VPL direct lighting distribution. The basic idea is to generate a per-light geometry buffer. Instead of storing albedo, surface outgoing radiance is stored for later indirect lighting calculation.

GPU-based concurrent link lists

Shader Model 5 introduces advanced GPU programming memory access pattern. Under OpenGL GLSL 420 standard, we could read from and write to arbitrary buffer and texture locations. GPU concurrent link list stores per-pixel lists in a generic buffer. This technique is very useful in that many difficult-to-solve rendering problems are now easy to resolve, such as order-independent-transparence and ray-bundle intersection data set generation.

Shader Atomic Operations

It is common that multiple shader execution threads access a same memory location in Shader Model 5 shader programs. We must deal with this carefully to avoid unpredictable runtime errors. Shader Atomic Operations are a bunch of built-in shader functions helping us with this.

GPU generic buffers.

Shader Programming models and languages such as GLSL, Cg and HLSL are still evolving. Generic buffers are introduced in Shader Model 5 and allow us to use generic C-like structs and store them in GPU memory. Now we can implement complex data structures such as linked list on GPU.

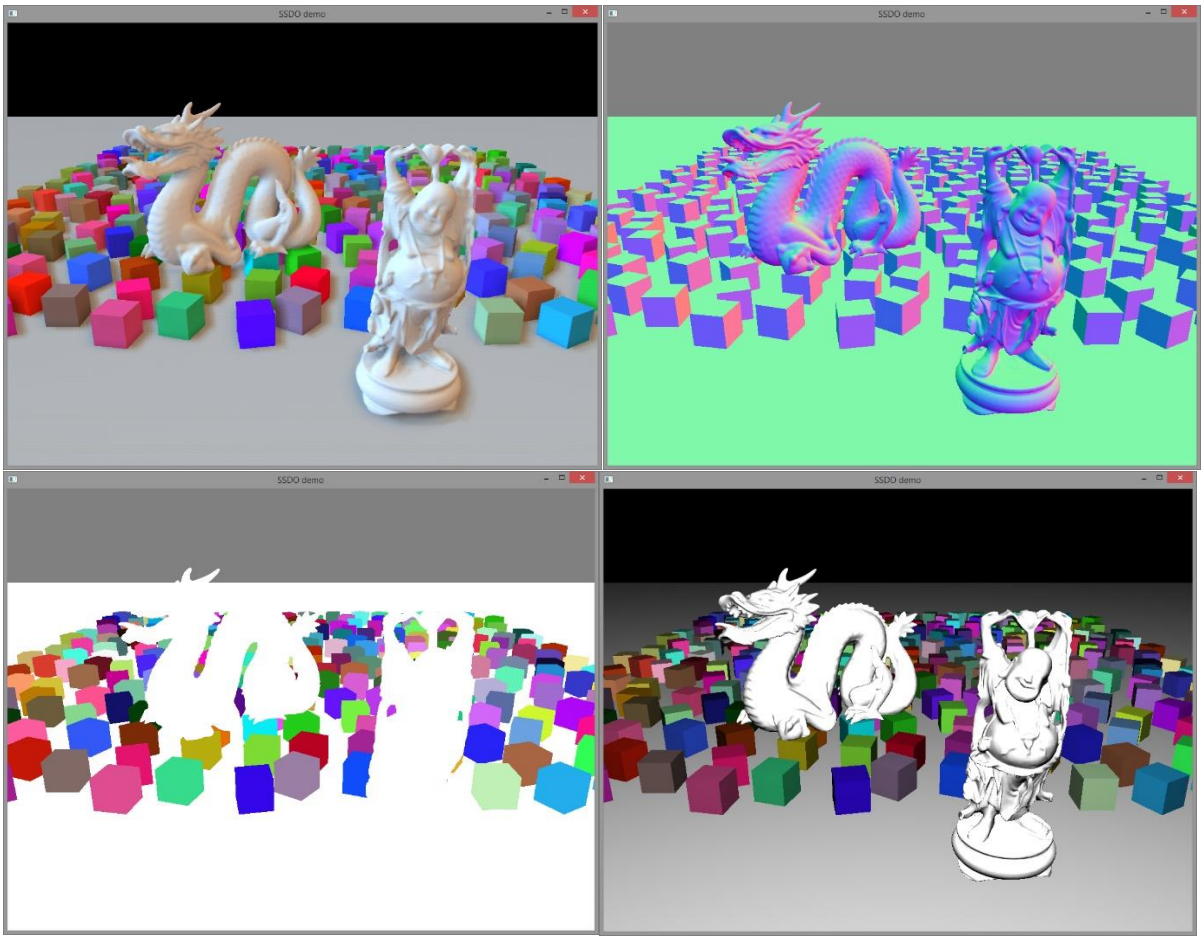
Scene Voxelization

Voxelization is a big topic and a promising technique used by some RTGI methods in the last few years. The famous LPV (CryEngine3) and SVO (Unreal Engine 4) both use scene voxelization to accelerate ray-intersection and light propagation calculation. Now I am using this technique to implement my ray-bundles radiance accumulation buffer.

3 Programming Projects

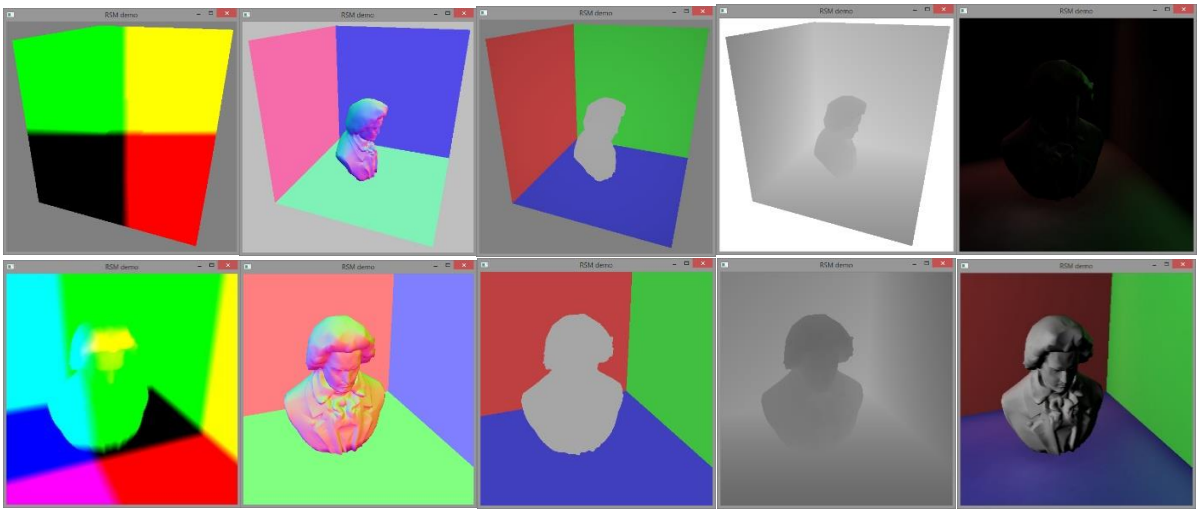
SSDO

Finished. See screenshots below:



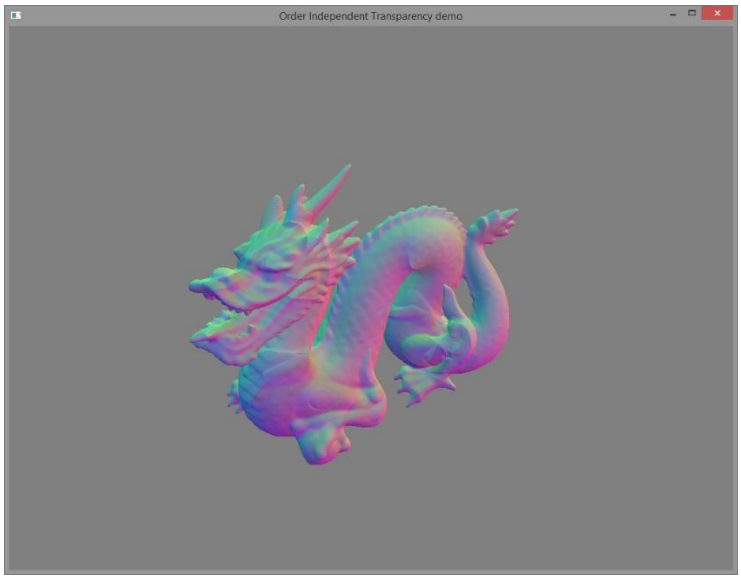
Reflective Shadow Maps

Finished. See screenshots below:



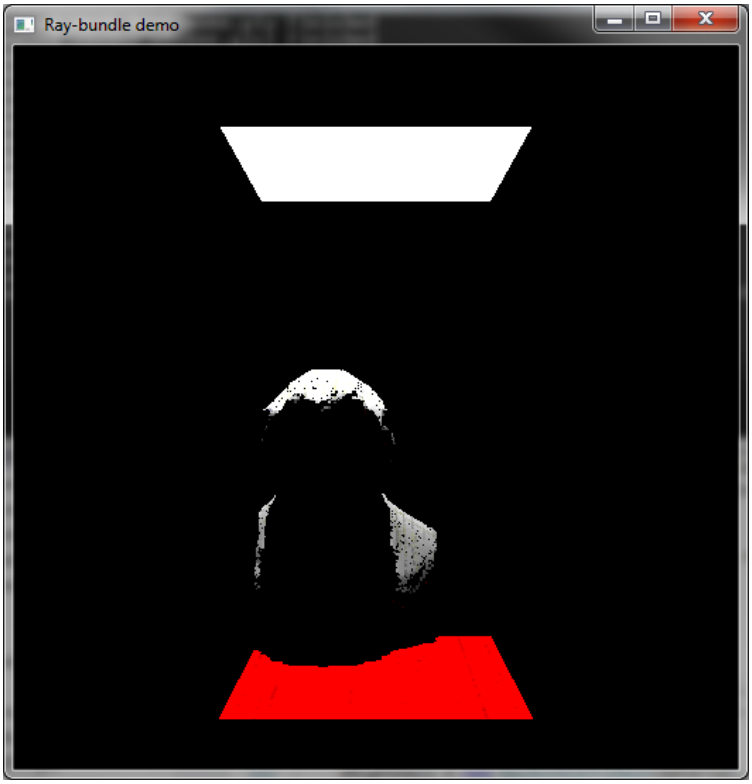
Order Independent Transparence

Finished. This demo makes use of GPU-based concurrent linked lists. See screenshot below:



Ray-bundles and Rasterization based GI

Still working on this. Debugging and trying to figure out a better way of implementing ray-bundles accumulation buffer.



4 Referrences

Approximating dynamic global illumination in image space

Tobias Ritschel, Thorsten Grosch, Hans-Peter

I3D '09 Proceedings of the 2009 symposium on Interactive 3D graphics and games

Cascaded light propagation volumes for real-time indirect illumination

Anton Kaplanyan, Carsten Dachsbacher

I3D '10 Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games

Global Illumination using Parallel Global Ray-Bundles

HERMES, J., HENRICH, N., GROSCH, T., AND MUELLER, S. 2010. Vision, Modeling and Visualization.

Real-Time Bidirectional Path Tracing via Rasterization

Yusuke Tokuyoshi, Shinji Ogaki

I3D '12 Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games

Real-Time Concurrent Linked List Construction on the GPU

Jason C. Yang, Justin Hensley, Holger Grün, Nicolas Thibieroz

EGSR'10 Proceedings of the 21st Eurographics conference on Rendering

Reflective Shadow Maps

Carsten Dachsbacher, Marc Stamminger

I3D '05 Proceedings of the 2005 symposium on Interactive 3D graphics and games

Scalable Realistic Rendering with Many-Light Methods

Carsten Dachsbacher, Jaroslav Krivánek, Milos Hasan, Adam Arbree, Bruce Walter, and Jan Novák

EG 2013

Voxel-based Global Illumination

Sinje Thiedemann, Niklas Henrich, Thorsten Grosch, Stefan Müller

I3D '11 Symposium on Interactive 3D Graphics and Games

The State of the Art in Interactive Global Illumination

Tobias Ritschel, Carsten Dachsbacher, Thorsten Grosch, Jan Kautz

Computer Graphics Forum archive

Volume 31 Issue 1, February 2012

Interactive Indirect Illumination Using Voxel Cone Tracing

Cyril Crassin, Fabrice Neyret, Miguel Sainz, Simon Green, Elmar Eisemann

I3D '11 Symposium on Interactive 3D Graphics and Games

Extending OGRE with Light Propagation Volumes

J Elvek

2012 Master Thesis

Instant Radiosity

Alexander Keller

SIGGRAPH '97 Proceedings of the 24th annual conference on Computer graphics and interactive techniques

Books:

GPU Pro 1

Wolfgang Engel (Editor) A K Peters/CRC Press (June 14, 2010)

GPU Pro 2

Wolfgang Engel (Editor) A K Peters/CRC Press; 1 edition (February 14, 2011)

GPU Pro 3

Wolfgang Engel (Editor) A K Peters/CRC Press; 1 edition (February 21, 2012)

GPU Pro 4

Wolfgang Engel (Editor) A K Peters/CRC Press (April 26, 2013)

Physically Based Rendering, Second Edition

Matt Pharr, Greg Humphreys

Morgan Kaufmann; 2 edition (July 12, 2010)

OpenGL Programming Guide 8th Edition

Dave Shreiner, Graham Sellers, John M. Kessenich, Bill M. Licea-Kane

Addison-Wesley Professional; 8 edition (March 30, 2013)