Screen-Space Global Illumination on Deep G-Buffers

Jonas H. Nielsen

January 4, 2016

Overview

Deferred Rendering

- Deep G-buffer
- Scene Reconstruction

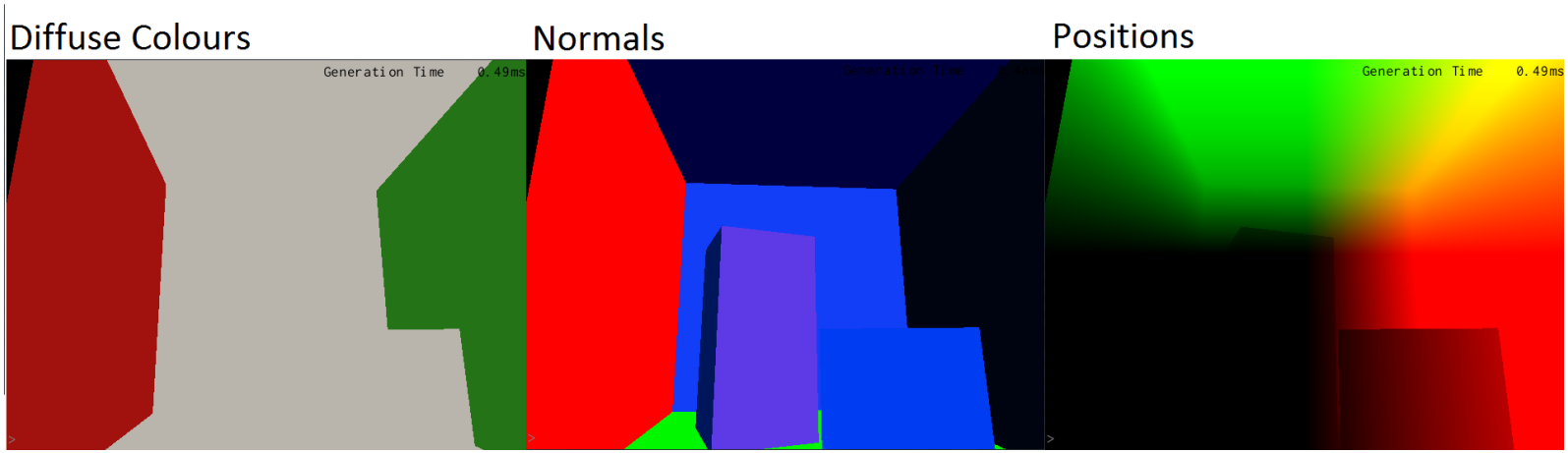
Omni-directional Shadowmapping

Filters

- Radiosity
- SSAO
- Gaussian Filter

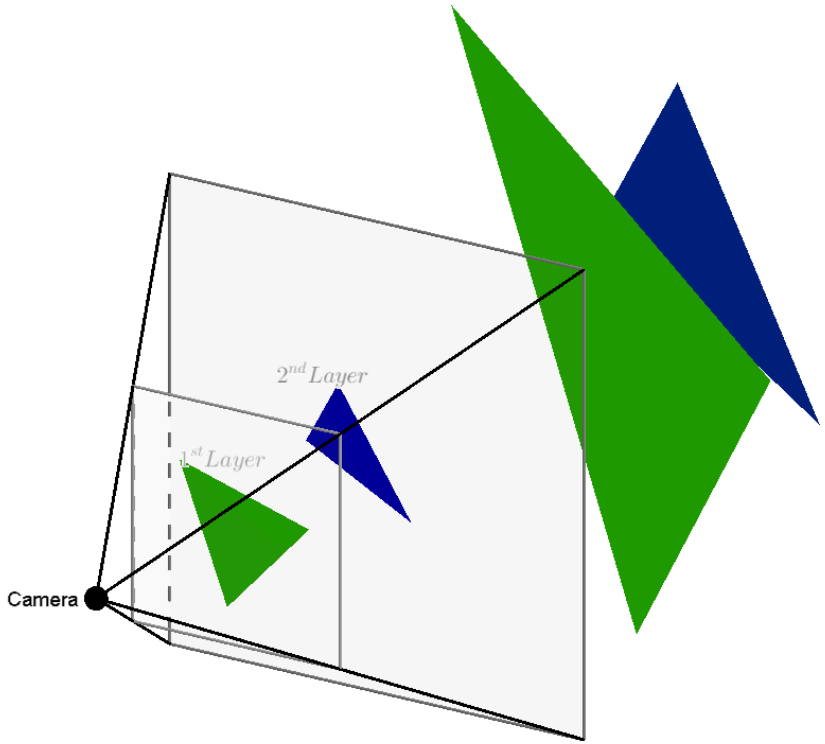
Deferred Rendering Overview

- Draw geometric information to off-screen textures.
- Use off-screen textures to do shading at later stage.
- Abstracts shading calculations from geometric complexity.



Deep G-buffer

- Store multiple layers of geometry in off-screen targets.
- Determine lower layer contents by contents of upper layers.
- Use geometry shader to do in a single pass
- Use previous frame's depth buffer to predict contents.



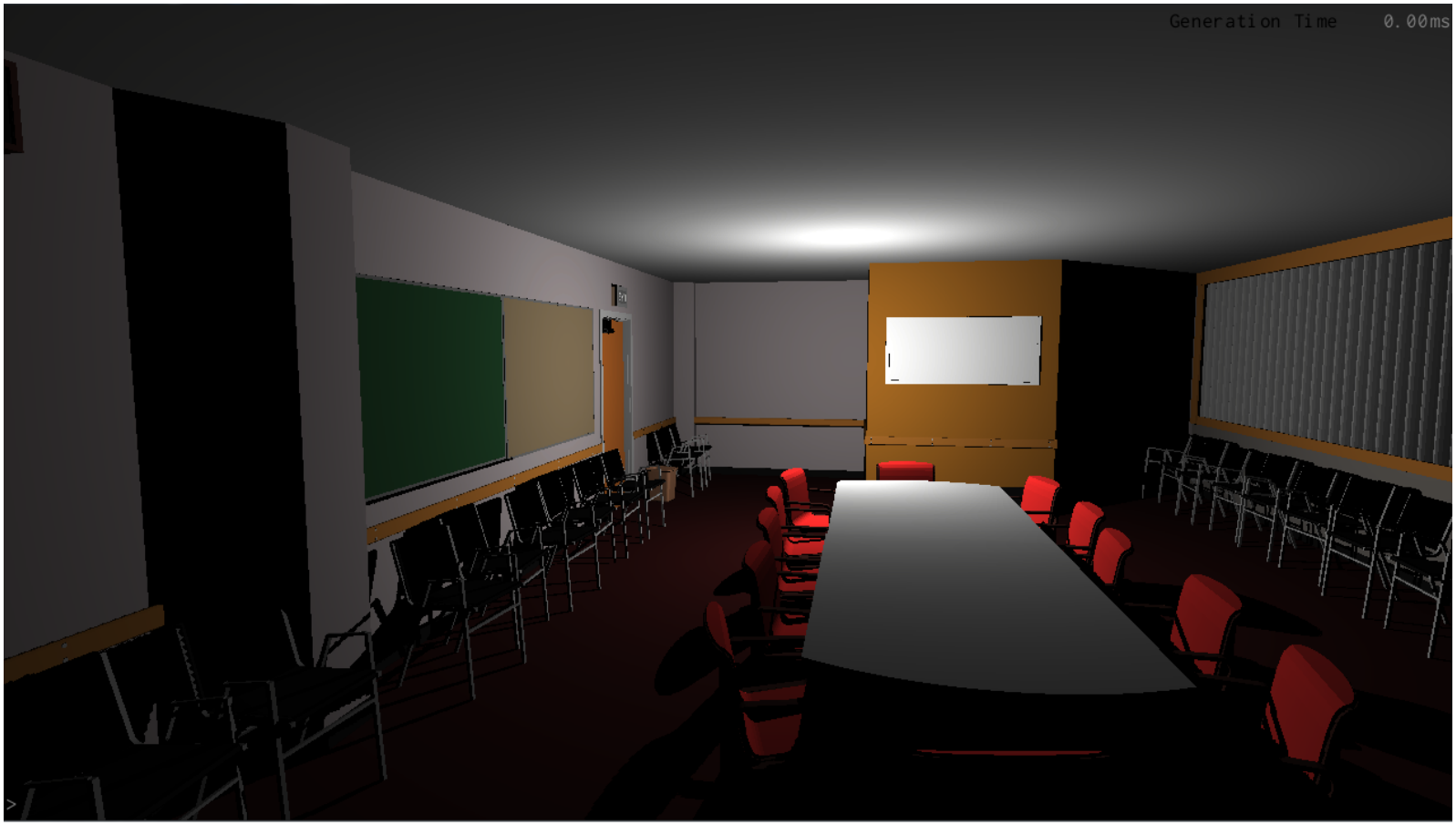
Scene Reconstruction

- Reconstruct geometric information.
- Allows to buffer less data, by reconstruction.
- Use inverse projection matrix to get view-space coords.
- Less VRAM writes and reads vs reconstruction cost.

Cube Shadow Mapping

- Render vertex-depth unto a cube map.
- Use vertex view-length as depth.
- Center view-matrices around light position.
- Adjust up-, and at-vectors according to sides of cube.
- Rasterise in single pass with geometry shader.

Shadow Results



Sampling Strategy

- Sample in screen-space along a spiral curve.
- Use parameters to impact form of curve.

$$\sigma = t + \frac{\psi}{\tau}$$

$$\theta = 2\pi\sigma\tau + \phi$$

$$\overline{u} = \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix}$$

$$h = R\sigma$$

$$\mathbf{v} = h\overline{u}$$

$$0 < t < 1, 0 < \phi < 2\pi, 0 < \psi < 1$$

Radiosity Equation

- Rendering equation for radiosity:

$$B(\mathbf{X}) = \int_{\Omega} \frac{\rho_d}{\pi} B(\mathbf{Y}) \cos(\theta) V() d\omega_i$$

- Monte-Carlo Approximation:

$$B(\mathbf{X}) = \sum_{i=0}^{n} \frac{\rho_d}{\pi} B(\mathbf{Y}(\omega_i)) \cos(\theta) V() \Delta\omega_i$$

Solving $\Delta\omega_i$

- Assume all samples represent equal solid angles:

$$B(\mathbf{X}) = \frac{2\pi}{n}\frac{\rho_d}{\pi}\sum_{i=0}^{n} B(\mathbf{Y}[\omega_i])\cos(\theta)V()$$

- Weight based on surface differential:

$$\Delta\omega_i = 2\pi\frac{\frac{\cos(\phi)}{\pi r^2}}{M}$$

$$M = \sum_{i=0}^{n}\frac{\cos(\phi)}{\pi r^2}$$

$$B(\mathbf{X}) = \frac{2\pi}{M}\frac{\rho_d}{\pi}\sum_{i=0}^{n} B(\mathbf{Y}[\omega_i])\cos(\theta)V()\frac{\cos(\phi)}{r^2}$$

Simple method result

Alternate Method Result

Timings for different $d\omega$s

	$2\pi\frac{\cos(\phi)}{r^2 M}$	$\frac{2\pi}{M}$
Samples	*Time (ms)*	*Time (ms)*
10	19,8	13,3
15	27,6	17,9
20	35,6	22,5
25	43,6	27,1
30	51,5	31,5
40	67,2	40,7
Per Sample	1,582	0,912
Overhead	3,95	4,22
Advantage	0,00%	42,35%

Timings for different formats

	R11G11B10	**RGB16F**	**RGB8**
Samples	Time (ms)	Time (ms)	Time (ms)
10	15,9	18,6	15,7
15	21,3	24,7	21,1
20	26,7	30,9	26,5
25	32,2	37,1	32
30	37,8	43,3	37,4
35	43,3	49,5	42,9
40	48,7	55,8	48,2
Per sample	1,096	1,24	1,086
Overhead	4,861	6,129	4,829
Improvement	0,00%	-13,14%	0,91%

Timings with layer-loop

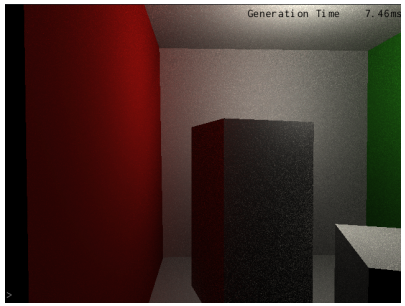
	Without loop	**With loop**
Samples	Time (ms)	Time(ms)
10	15,9	15,4
15	21,3	20,6
20	26,7	25,9
25	32,2	31,2
30	37,8	36,5
35	43,3	41,8
40	48,7	47
Per sample	1,096	1,056
Overhead	4,861	4,807
Improvement	0,00%	3,65%

Alchemy SSAO

$$AO(\mathbf{X}) \approx max\left(0, 1 - \frac{2\sigma}{s}\sum_{i=1}^{s}\frac{max(0, \mathbf{v_i}\cdot\overline{n} + \mathbf{X_z}\beta)}{\mathbf{v}_i\cdot\mathbf{v}_i + \epsilon}\right)^K$$

Filtering Overview

- Raw radiosity and SSAO are visibly noisy.



- Apply 2-pass gaussian filter:

$$BI_{1st}(P_c) = \frac{1}{W} \sum_{i=-R}^{R} \gamma(i) B(P(i,0))$$

$$BI(P_c) = \frac{1}{W} \sum_{i=-R}^{R} \gamma(i) BI_{1st}(P(0,i))$$

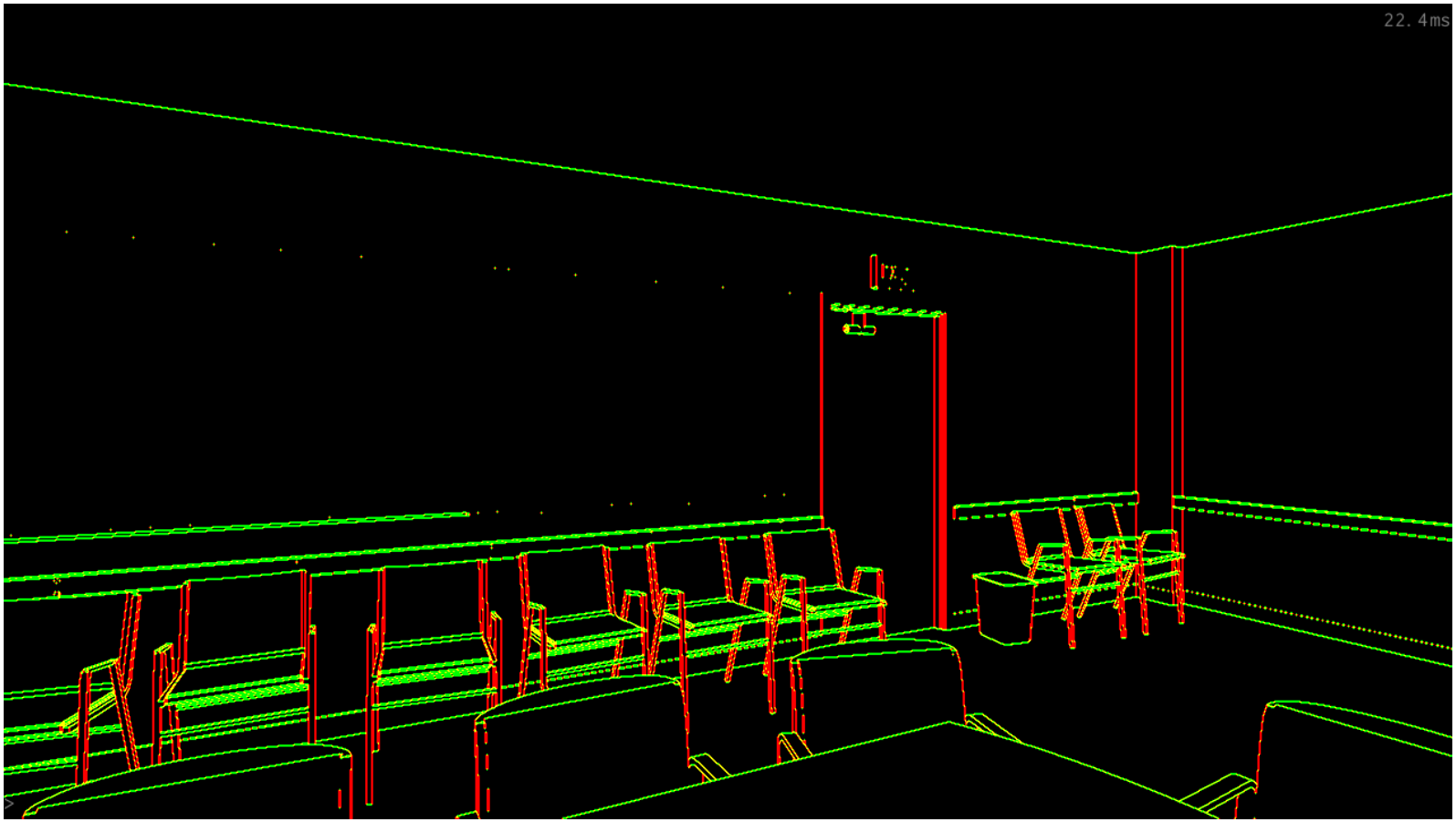
$$W = \sum_{i=-R}^{R} \gamma(i)$$

Pure Gaussian Result

- Simple Gaussian filter becomes blurry

Edge Detection

- To fix, first run edge detection algorithm:



Modified Filter

- Stop the gaussian filter loop if it hits an edge