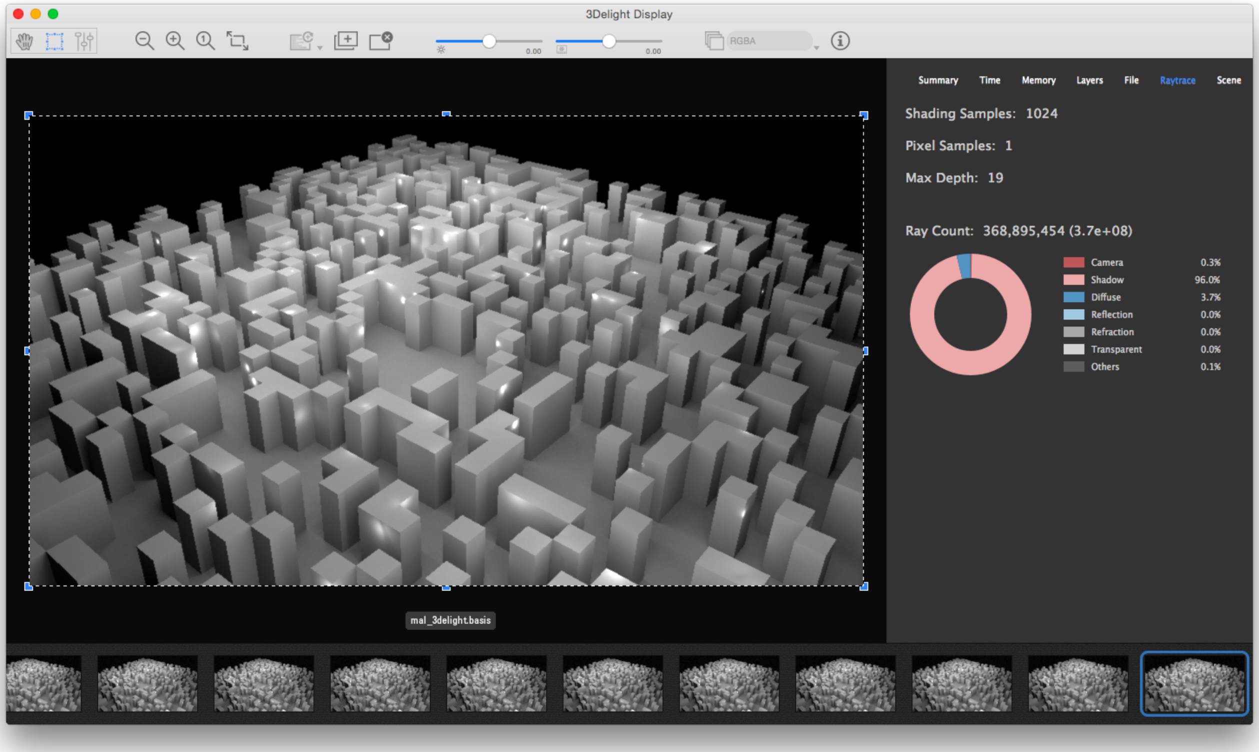
Geo Light Sampling Algorithms Comparison



Shading Samples: 1024

Pixel Samples: 1

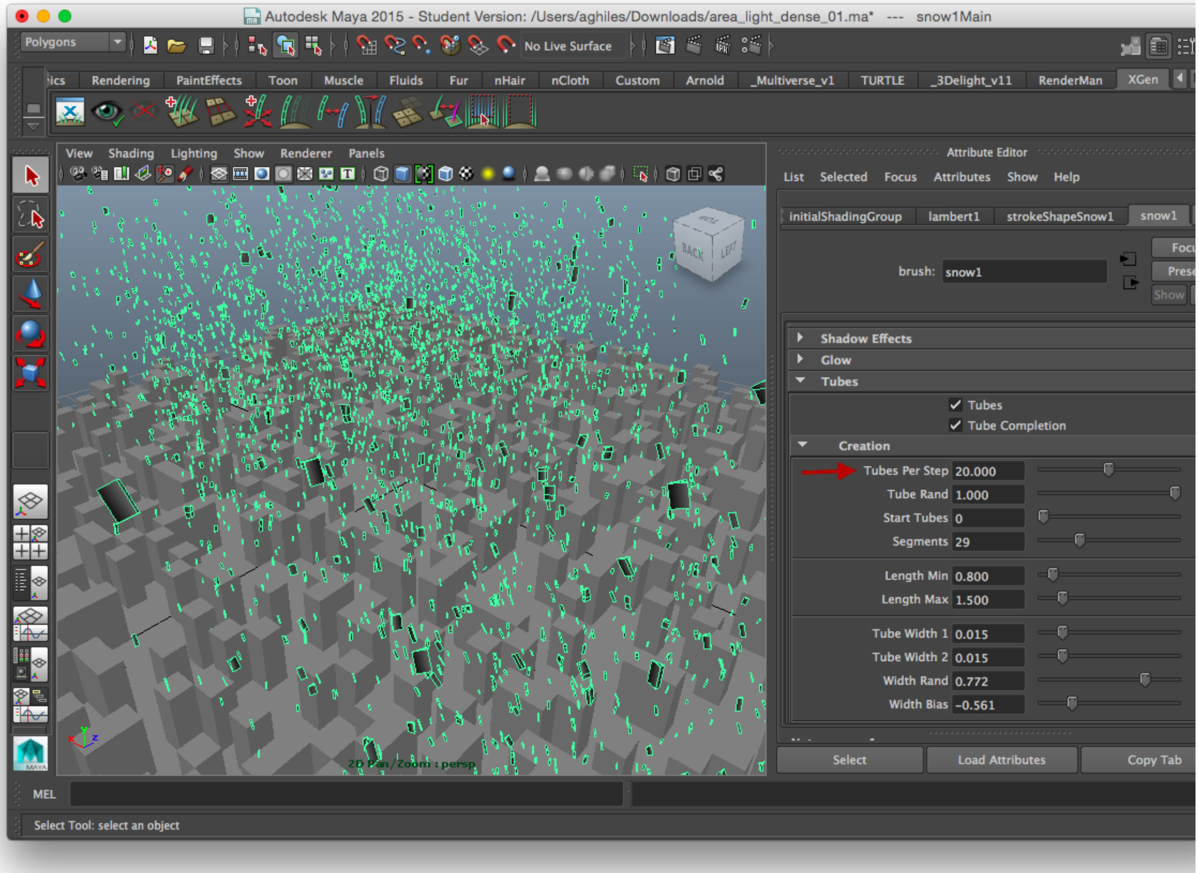
Max Depth: 19

Ray Count: 368,895,454 (3.7e+08)

Camera	0.3%
Shadow	96.0%
Diffuse	3.7%
Reflection	0.0%
Refraction	0.0%
Transparent	0.0%
Others	0.1%

It is not uncommon for production scenes to rely heavily on geo area lights. In this test we will compare geo light **sampling technology** in *3Delight OSL*, *Arnold* and *RenderMan/RIS*. We will use a test with a geometric area light that is composed of a relatively large number of faces. The test, although limited, should allow us to find the convergence rate of each algorithm as well as general performance.

The Scene



Autodesk Maya 2015 - Student Version: /Users/aghiles/Downloads/area_light_dense_01.ma* --- snow1Main

brush: snow1

☑ Tubes
☑ Tube Completion

Tubes Per Step	20.000
Tube Rand	1.000
Start Tubes	0
Segments	29
Length Min	0.800
Length Max	1.500
Tube Width 1	0.015
Tube Width 2	0.015
Width Rand	0.772
Width Bias	-0.561

Select Tool: select an object

The test scene allows for procedural generation of floating area lights using the "Tubes Per Step" PaintEffects attribute.

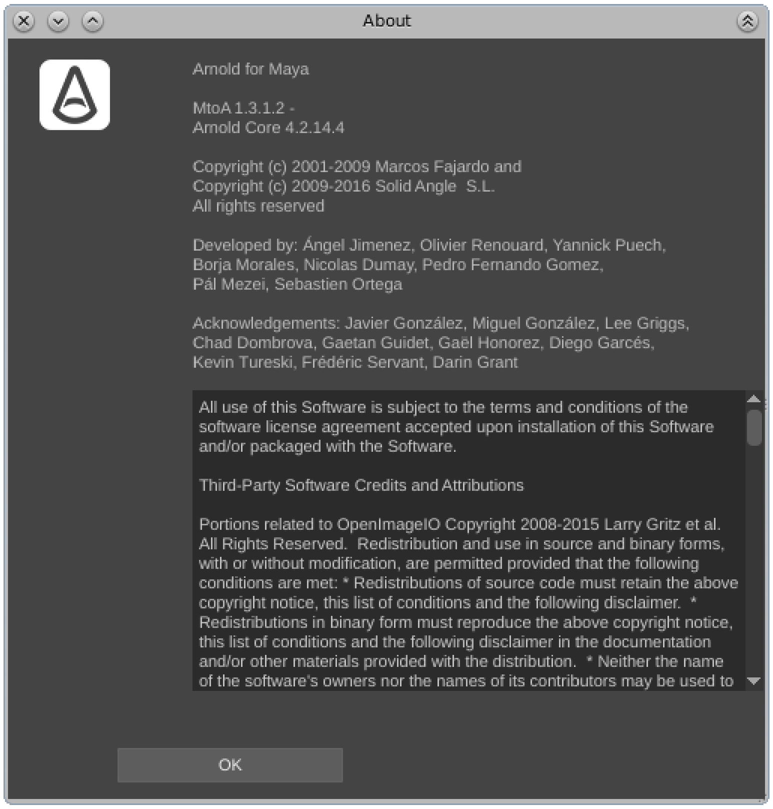
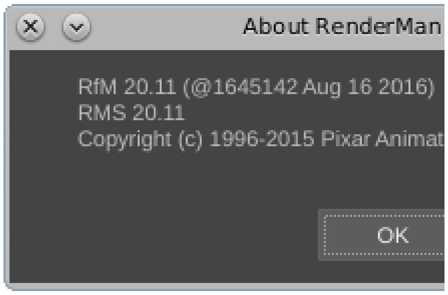
The scene is designed to procedurally generate a number of area lights floating on top of a "city".

- Only diffuse reflectors.
- We will disable any adaptive sampling so to make sure we have meaningful ray-counts.
- We will use only direct lighting to estimate the geometric area light contribution. In the statistics files for each renderer, one can see that we have only one path length.
- The light sources contain about 80K triangles.
- All Renders are done in *Maya 2016.*

The Renderers

ⓘ *RenderMan/RIS 21* is about to be released with new sampling technology. We will complete the test as soon as the renderer is available.

	Arnold	RenderMan/RI

Version	 About Arnold for Maya MtoA 1.3.1.2 - Arnold Core 4.2.14.4 Copyright (c) 2001-2009 Marcos Fajardo and Copyright (c) 2009-2016 Solid Angle S.L. All rights reserved Developed by: Ángel Jimenez, Olivier Renouard, Yannick Puech, Borja Morales, Nicolas Dumay, Pedro Fernando Gomez, Pál Mezei, Sebastien Ortega Acknowledgements: Javier González, Miguel González, Lee Griggs, Chad Dombrova, Gaetan Guidet, Gaël Honorez, Diego Garcés, Kevin Tureski, Frédéric Servant, Darin Grant All use of this Software is subject to the terms and conditions of the software license agreement accepted upon installation of this Software and/or packaged with the Software. Third-Party Software Credits and Attributions Portions related to OpenImageIO Copyright 2008-2015 Larry Gritz et al. All Rights Reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. * Neither the name of the software's owners nor the names of its contributors may be used to 	 About RenderMan RfM 20.11 (@1645142 Aug 16 2016) RMS 20.11 Copyright (c) 1996-2015 Pixar Animat
Technology	Unidirectional path tracer.	Using unidirectional path tracer. Other options are avail
Shaders	C++	C++

Methodology

We are interested to find out about:

- The quality of the **geo light** samples drawn by each renderer. How fast do we converge to ground truth by increasing the number of samples ?
- Performance. How fast is the sampler to produce an image of a given quality ?

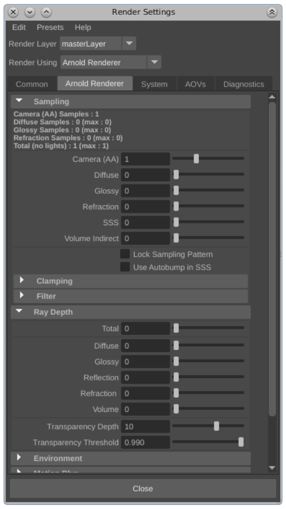
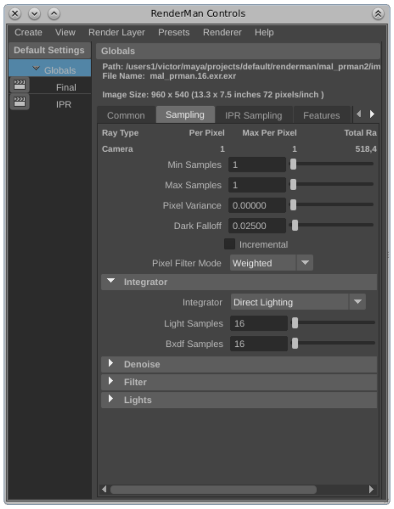
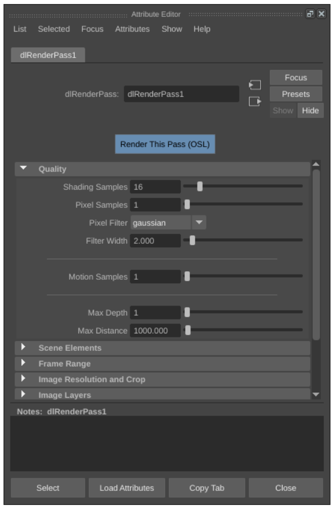
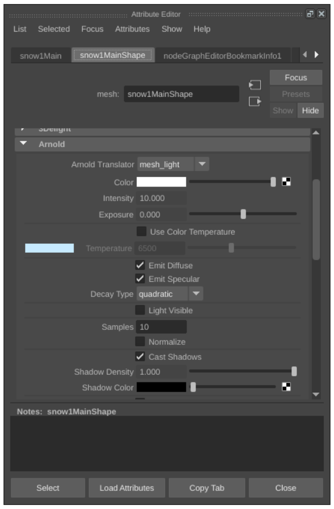
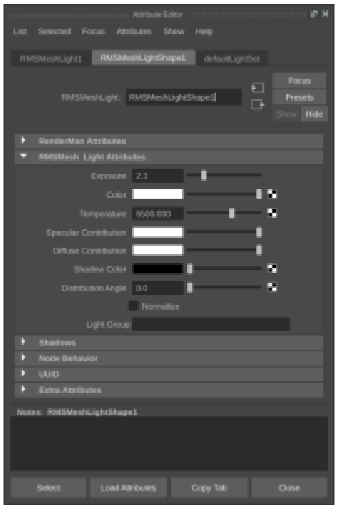
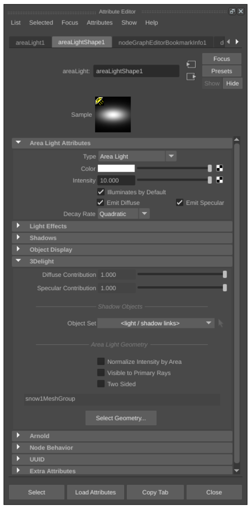
We will start by creating a "ground truth" image for each renderer; an image generated by using a very large amount of samples so there is no more apparent noise. Although it is normal for images generated by each renderer to vary slightly, encouragingly, images produced by both *3Delight* and *Arnold* are almost exactly the same. We will then render several images with varying amount of samples and measure the RMSE between these images and the ground truth for each renderer. Timings and statistics will be collected at each render. Having this data will allow us to draw a conclusion about the convergence rate and general performance.

The Setup

We use a 1x1 pixel sample in all renderers. Adaptivity is disabled as well as all additional bounces. In *Arnold*, the light samples are attributes of the geometry. In *RenderMan/RIS* the light samples are attributes of a custom shape. In *3Delight*, the light attributes are on *Maya's* area light.

> Admittedly, those are not standard settings for any useful render. The goal here is to *isolate* one algorithm in order to understand its behaviour. Understanding an algorithm can shed light on its strengths and weaknesses and allows us to draw interesting conclusions for the more general case.

	Arnold	**RenderMan/RIS**	**3Delight OSL**

General Parameters	 Render Settings Render Layer: masterLayer Render Using: Arnold Renderer Arnold Renderer — Sampling Camera (AA) Samples : 1 Diffuse Samples : 0 (max : 0) Glossy Samples : 0 (max : 0) Refraction Samples : 0 (max : 0) Total (no lights) : 1 (max : 1) Camera (AA) 1 Diffuse 0 Glossy 0 Refraction 0 SSS 0 Volume Indirect 0 □ Lock Sampling Pattern □ Use Autobump in SSS Ray Depth Total 0 Diffuse 0 Glossy 0 Reflection 0 Refraction 0 Volume 0 Transparency Depth 10 Transparency Threshold 0.990	 RenderMan Controls Globals Path: /users1/victor/maya/projects/default/renderman/mal_prman2/im File Name: mal_prman.16.exr.exr Image Size: 960 x 540 (13.3 x 7.5 inches 72 pixels/inch) Sampling Ray Type: Camera — Per Pixel 1 — Max Per Pixel 1 — Total Ra 518,4 Min Samples 1 Max Samples 1 Pixel Variance 0.00000 Dark Falloff 0.02500 □ Incremental Pixel Filter Mode Weighted Integrator Integrator Direct Lighting Light Samples 16 Bxdf Samples 16	 Attribute Editor dlRenderPass: dlRenderPass1 Render This Pass (OSL) Quality Shading Samples 16 Pixel Samples 1 Pixel Filter gaussian Filter Width 2.000 Motion Samples 1 Max Depth 1 Max Distance 1000.000 Notes: dlRenderPass1
Geo Light Parameters	 Attribute Editor mesh: snow1MainShape Arnold Arnold Translator mesh_light Color Intensity 10.000 Exposure 0.000 □ Use Color Temperature Temperature 6500 ☑ Emit Diffuse ☑ Emit Specular Decay Type quadratic □ Light Visible Samples 10 □ Normalize ☑ Cast Shadows Shadow Density 1.000 Shadow Color Notes: snow1MainShape	 Attribute Editor RMSMeshLight: RMSMeshLightShape1 RMSMesh Light Attributes Exposure 2.3 Color Temperature 6500.000 Specular Contribution Diffuse Contribution Shadow Color Distribution Angle 0.0 □ Normalize Light Group Notes: RMSMeshLightShape1	 Attribute Editor areaLight: areaLightShape1 Area Light Attributes Type Area Light Color Intensity 10.000 ☑ Illuminates by Default ☑ Emit Diffuse ☑ Emit Specular Decay Rate Quadratic 3Delight Diffuse Contribution 1.000 Specular Contribution 1.000 Shadow Objects Object Set <light / shadow links> Area Light Geometry □ Normalize Intensity by Area □ Visible to Primary Rays □ Two Sided snow1MeshGroup

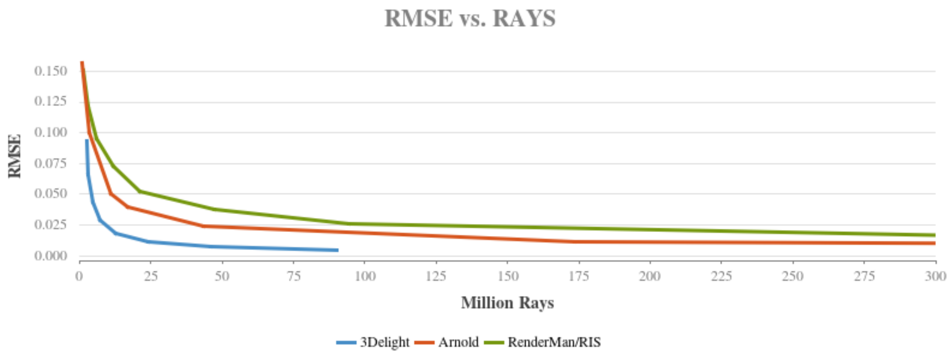
Notes

Arnold — For light samples, *Arnold* uses effective sample counts that are proportional – within a constant – to the *square* of the user specified value. As we will see, this makes sense from a UI standpoint since the variance follows the inverse of the same rule in the case of *Arnold*. This makes the light samples slider *linear* in term of perceived noise. In the *Arnold* tables below, we will specify the effective samples per pixel along with the user samples. Those effective samples are gathered from *Arnold*'s diagnostics files.

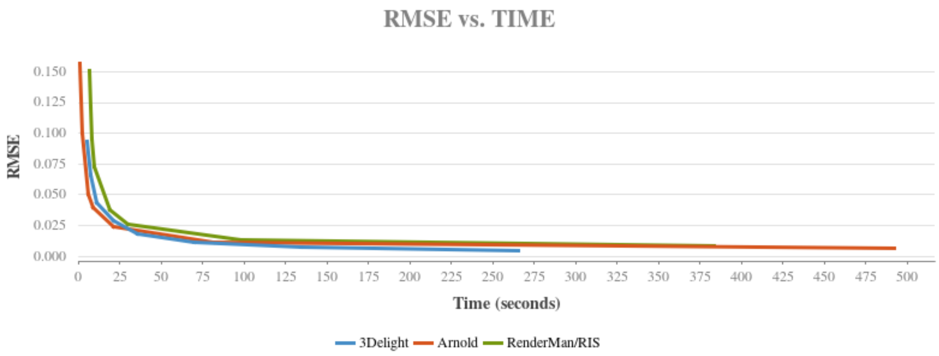
RenderMan/RIS – While in *Arnold* the *light samples* effectively control image quality when only direct lighting is needed, in *RenderMan/RIS*, we had to match light samples count with BxDF samples count to achieve acceptable quality and satisfactory convergence rates. Using light samples only, or BxDF samples only, produced noisy renders. In the *RenderMan/RIS* results below, "N samples" means N samples for *both* light and BxDF.   We did all the tests with the "advanced (mode 4)" light sampler since it produced the best results. The samples used by the renderer are the ones entered in the UI and are not squared as in *Arnold*. Note that we used the path tracer with one bounce instead of the "direct lighting" algorithm for one of the images because of a crash (quality and speed did not seem to suffer although there were some minor differences in the render).

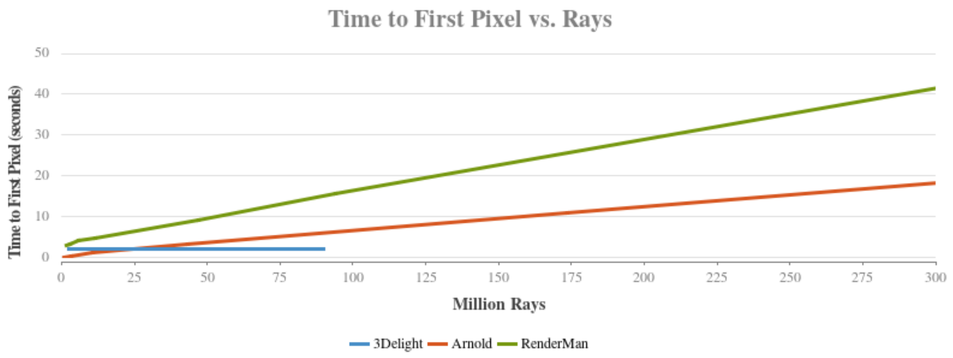
3Delight – We have only one control for the general quality of the render. In the case of direct lighting, *3Delight* "understands" that samples are best used for light sampling and that's what it does. As the tests will show, those samples have a *linear* impact on perceived noise levels.

Results

The following graph gives a good idea on the convergence rate of the different light samplers.	
The follow	

ing graph shows the time required to achieve a certain quality. From user's perspective, this is an important qua

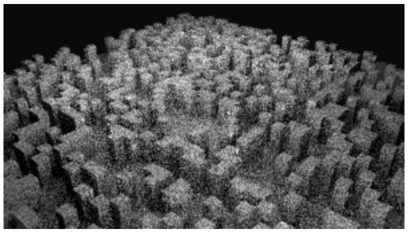
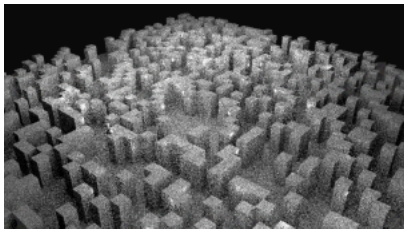
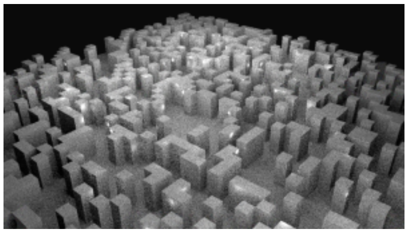


ntity.	
The following graph shows how much time it takes to build the acceleration data structure depending on s	

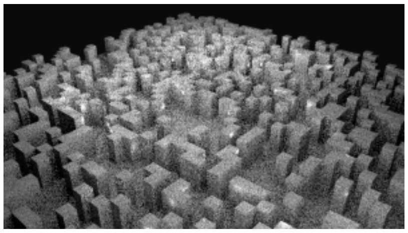
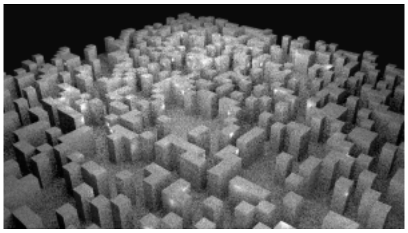
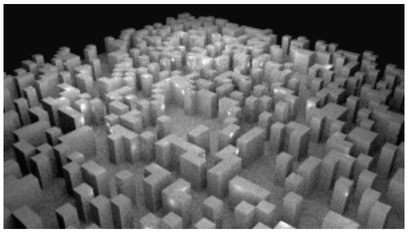
ample/raycount.	

Arnold

*TTFP = Time to first pixel.

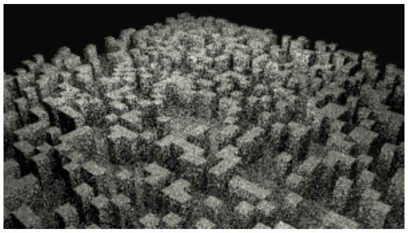
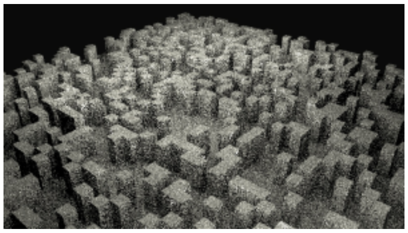
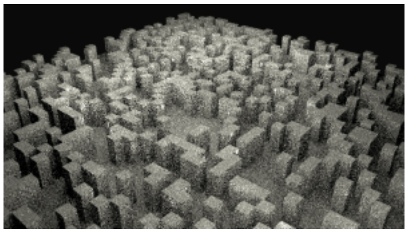
Samples (effective)	2 (1.23)	4 (4.91)	8 (19.64)	
Image				
Time	1 sec.	2 sec.	6 sec.	9
TTFP*	0 sec.	0.35 sec.	1.2 sec.	2
Shadow Rays	0.678 M	3.26 M	10.8 M	1
RMSE	0.15699	0.100115	0.0501787	0

3Delight

Samples	2	4	8	1
Image				
Time	5.24 sec.	7.18 sec.	11.21 sec.	20
TTFP	2 sec.	2 sec.	2 sec.	2 s
Shadow Rays	2.45 M	3.13 M	4.51 M	7.2
RMSE	0.0933142	0.0658266	0.0441248	0.0

RenderMan/RIS

Samples	1	2	4	8

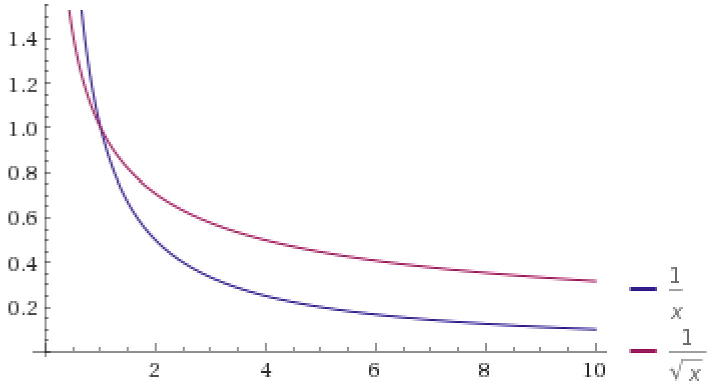
Image				
Time	6.74 sec.	7.23 sec.	7.99 sec.	9.4
TTFP	3.1 sec.	3.2 sec.	4 sec.	4.7
Rays	1.47 M	2.94 M	5.88 M	11
RMSE	0.151125	0.121487	0.0953649	0.0

Conclusions

- *3Delight* generates light samples that are *algorithmically* better (in term of variance) than both *Arnold* and *RenderMan*. In short, for x effective samples:

  a. *3Delight* Variance ~ 1/x
  b. *Arnold* Variance ~ 1/sqrt(x)
  c. *RenderMan/RIS* Variance ~ 1/sqrt(x) (possibly slightly worse but could be within a constant)

  Plot:



- *3Delight* is slower to generate these samples. For draft renders (high variance), *Arnold* is fastest. For final renders (low variance) *3Delight* is fastest and becomes increasingly faster with increasing samples.
- *Arnold* and *RenderMan/RIS* draw samples at about the same speed, but the quality of *Arnold* samples is better.
- Both *Arnold* and *RenderMan/RIS* produce biased images at low sample counts. More specifically: images are darker. *3Delight* manages to keep the same energy overall independently of sample counts.
- *Arnold, 3Delight* and *RenderMan/RIS* rely on acceleration data structures to sample the geometric area lights. In *Arnold* and *RenderMan/RIS*, the algorithmic complexity to build those data structures is tied linearly to the number of samples (as well as the complexity of the light). In *3Delight*, only the complexity of the light matters (time to first pixel for *3Delight* was 2-3 seconds no matter how many samples there were).

Resources

	RenderMan/RIS	Arnold	3Delight
Images and Stats	renderman.tar.gz	arnold.tar.gz	3delight.tar.gz
Maya Scene	mal_prman.ma	mal_3delight.ma	
Remarks		The same scene works with both *3Delight* and *Arnold*	