Ray Tracing Stress Test

We will compare 3Delight's OSL rendering core with two other renderers (*Arnold* and *RenderMan/RIS*) using C++ for shading and also to our legacy RenderMan compliant renderer (optimised RSL SIMD just-in-time compiler based). Since all renderers are quite different, we will try to get to a most common denominator, in terms of setup, to be able to compare *apples to apples.* We want to know: can we go as fast, in OSL, as highly optimized C++ ray-tracer? How does OSL compare to our optimised RSL SIMD just-in-time compiler?

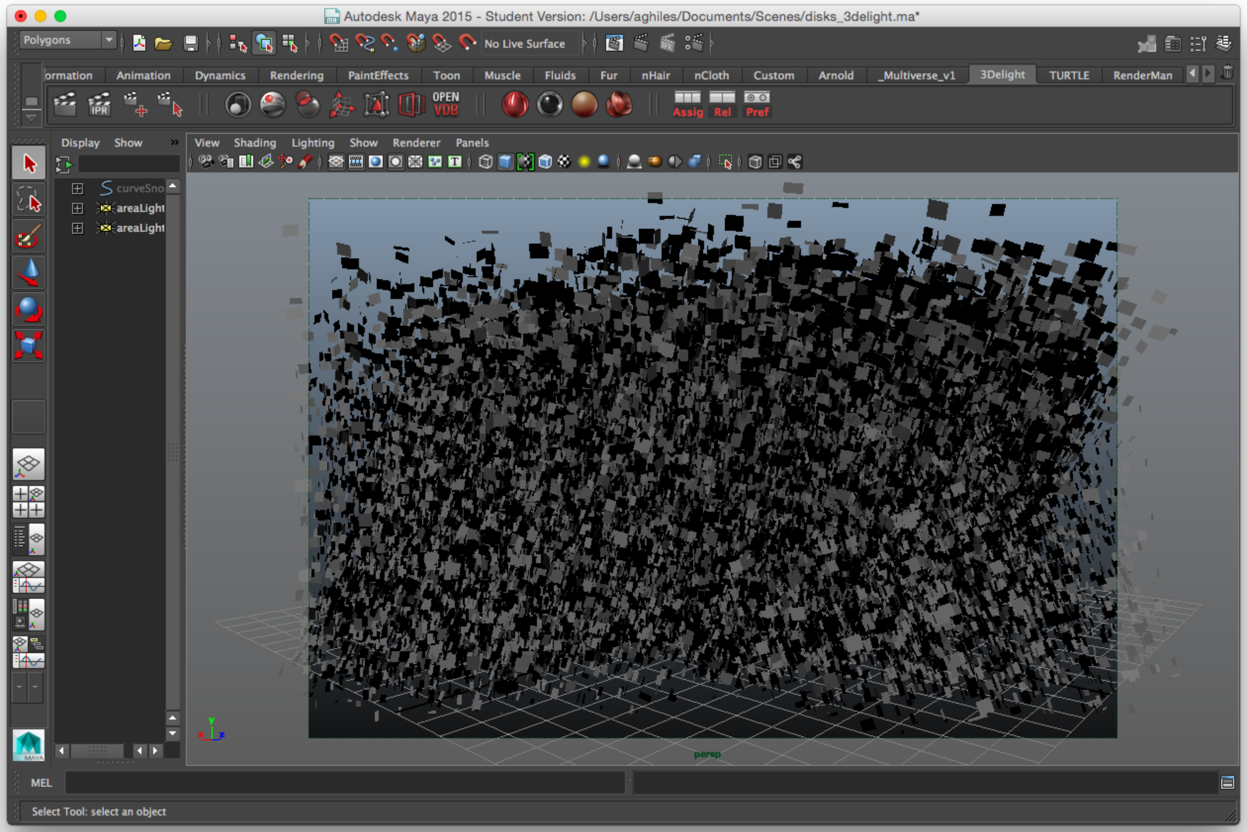
In this particular case, we will be constructing an artificial test scene that *simulates* foliage. Rendering foliage is a good stress test for many reasons:

- Many shader evaluations, including for transmission rays. No "opaque" object tricks are possible because leaves are usually modeled using cutouts. Shader evaluation efficiency is important.
- Many short ray probes necessitate a good integration and coordination between shading system and space partitioner.
- Any bottlenecks in the ray tracer tend to be amplified and adversely affect performance.

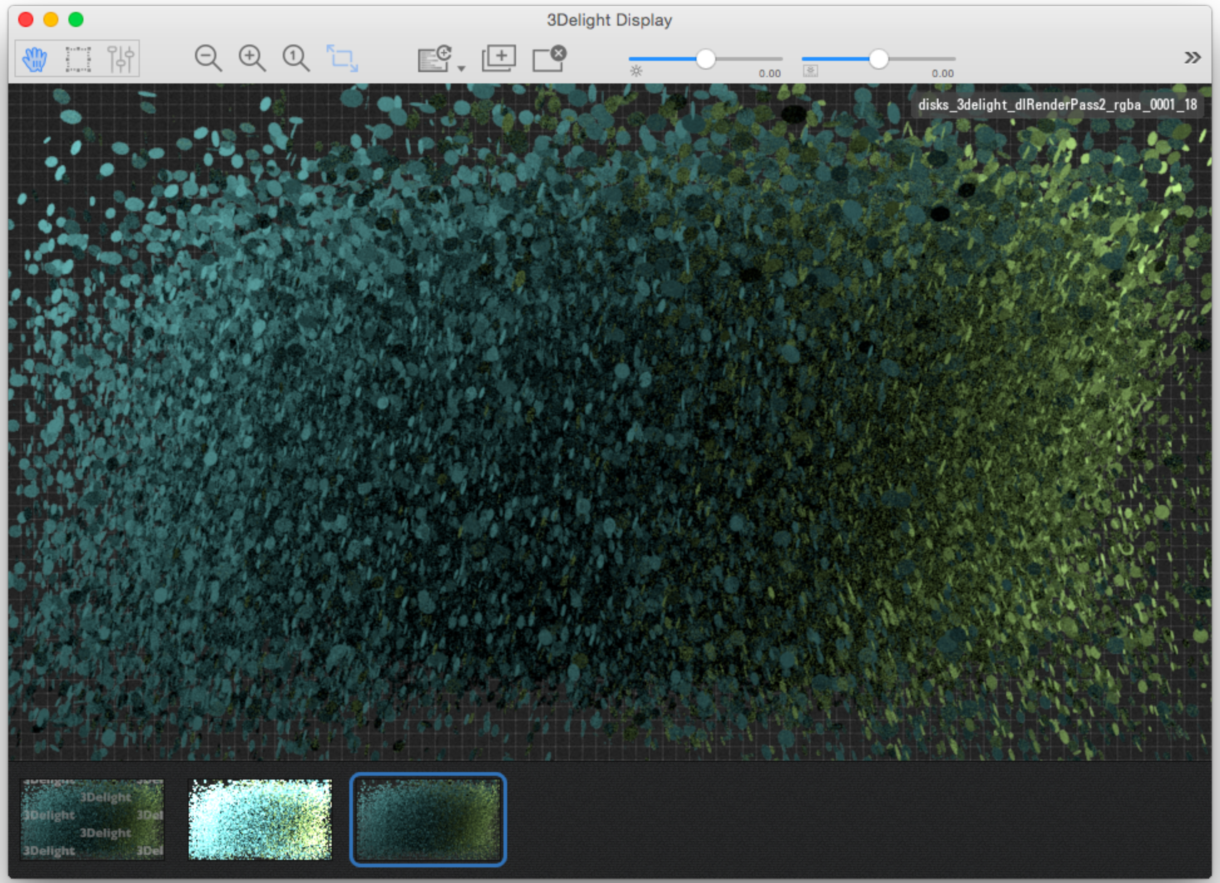
The Scene

The scene is designed in such a way as to put all the contenders in an even playing field and get meaningful performance numbers. For example:

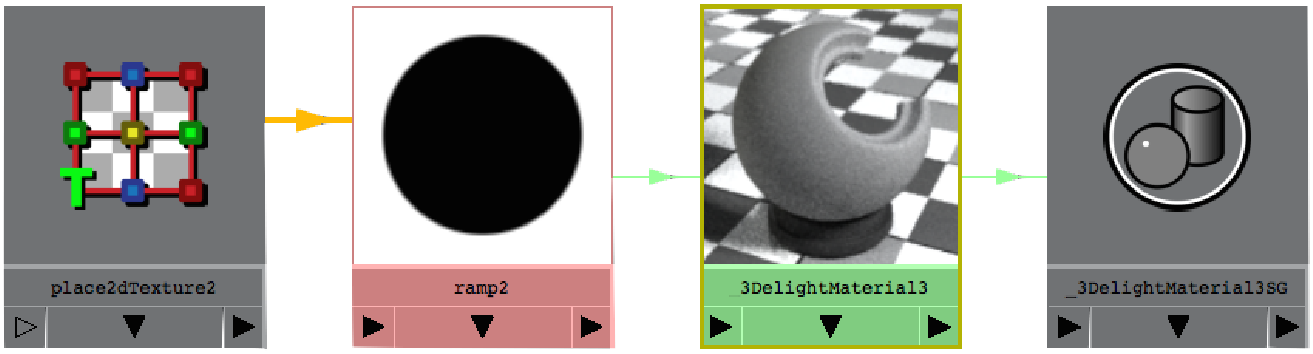
- We will just use diffuse reflectors.
- We will disable any adaptive sampling so to make sure we have very close ray-counts.
- We will use only one diffuse bounce. This will circumvent some complex variables from the equation (Russian roulette and other tricks of the trade). When looking at the statistics, all renders have a *path length* of 3.
- The scene is easy to load, setup and render. It is available for all tested renderers.
- All renders are done in Maya 2015 Extension 1.



The scene looks quite ugly in the viewport: just a large collection of rectangles generated using PaintEffects.



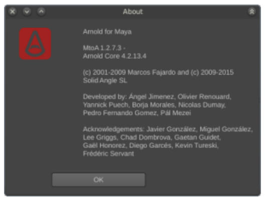
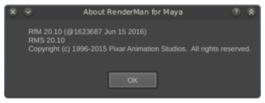
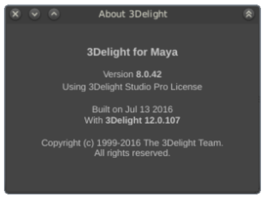
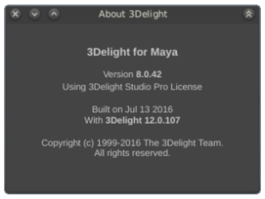
The render looks slightly better. Squares are transformed into circles using a cutout.
The scene is lit by two large area lights from each side.



The cutout is done using a Maya Ramp.
In Arnold it is connected to "opacity", in RenderMan to "presence" and in 3Delight to "transparency" !
(3Delight shown here)

The Renderers

	Arnold	RenderMan/RIS	3Delight OSL	3Delight RSL

Version	 About Arnold for Maya MtoA 1.2.7.3 - Arnold Core 4.2.13.4 (c) 2001-2009 Marcos Fajardo and (c) 2009-2015 Solid Angle SL Developed by: Ángel Jimenez, Olivier Renouard, Yannick Puech, Borja Morales, Nicolas Dumay, Pedro Fernando Gomez, Pál Mezei Acknowledgements: Javier González, Miguel González, Lee Griggs, Chad Dombrova, Gaetan Guidet, Gaël Honorez, Diego Garcés, Kevin Tureski, Frédéric Servant OK	 About RenderMan for Maya RfM 20.10 (@1623687 Jun 15 2016) RMS 20.10 Copyright (c) 1996-2015 Pixar Animation Studios. All rights reserved. OK	 About 3Delight **3Delight for Maya** Version **8.0.42** Using 3Delight Studio Pro License Built on Jul 13 2016 With **3Delight 12.0.107** Copyright (c) 1999-2016 The 3Delight Team. All rights reserved.	 About 3Delight **3Delight for Maya** Version **8.0.42** Using 3Delight Studio Pro License Built on Jul 13 2016 With **3Delight 12.0.107** Copyright (c) 1999-2016 The 3Delight Team. All rights reserved.
Technology	Unidirectional path tracer.	Using unidirectional path tracer. Other options are available but not useful for this test.	Unidirectional path tracer.	Undirectional path tracer
Shaders	C++	C++	*OSL*	*RSL*

The Methodology

We will be testing the wall clock time for all renderers. Every renderer will be run 3 times in succession and the average time will be taken. We make sure that nothing is running on the machine to slow down the poor renderers. The render is launched from **Maya 2015 Extension 1**. We are using a mid level Linux machine for our tests:

% lscpu

```
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                16
On-line CPU(s) list:   0-15
Thread(s) per core:    2
Core(s) per socket:    4
Socket(s):             2
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 26
Stepping:              5
CPU MHz:               2395.000
BogoMIPS:              4787.24
Virtualization:        VT-x
L1d cache:             32K
L1i cache:             32K
L2 cache:              256K
L3 cache:              8192K
```

% free -m

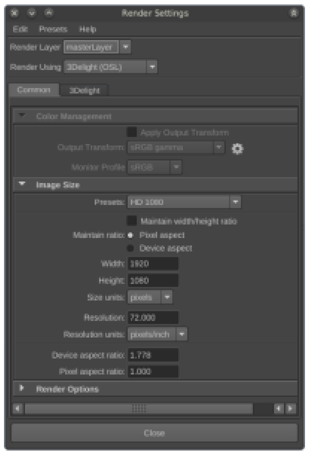
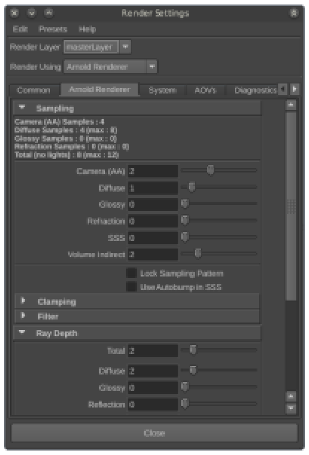
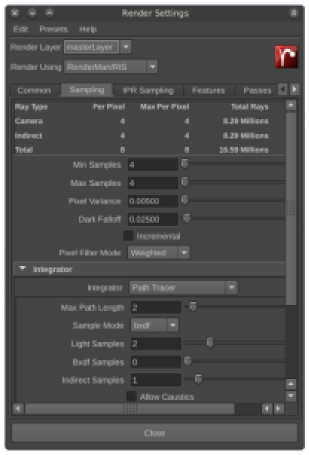
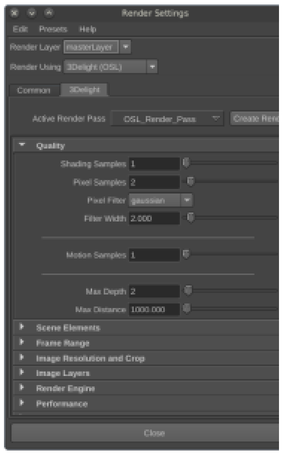
```
             total       used       free     shared    buffers     cached
Mem:         11984       8139       3844          0        151       4728
-/+ buffers/cache:       3259       8725
Swap:            0          0          0
```

The Setup

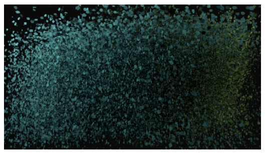
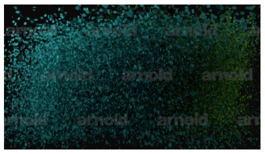
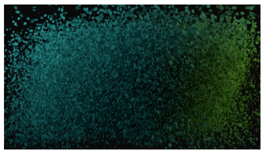
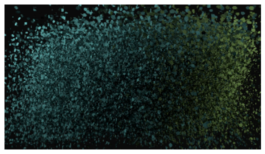
Each renderer uses a different sampling strategy and how rays are distributed between each sampled component. We selected a setup that produces nearly the same number of rays in all renderers. Look wise, it was easy to obtain very similar renders between Arnold and 3Delight. We had to boost area light contributions in order to get a similar lighting in RenderMan/RIS.

More details:

- We use non-normalized area lights. This is because *3Delight OSL* doesn't support normalized lights yet.
- There is a slight difference in tint between Arnold and 3Delight. We think it's due to differences in color management between the two renderers.
- *RenderMan/RIS* area lights sources had to be set with higher intensity. We are unsure why.
- *RenderMan/RIS* has very different diffuse lighting component. Again, we are unsure why. One hint might be how the shading normals are treated.

	Common	Arnold	RenderMan/RIS	3Delight OSL
Version	 Render Settings Render Layer: masterLayer Render Using: 3Delight (OSL) Common / 3Delight Color Management Apply Output Transform Output Transform: sRGB gamma Monitor Profile: sRGB Image Size Presets: HD 1080 Maintain width/height ratio Maintain ratio: ● Pixel aspect ○ Device aspect Width: 1920 Height: 1080 Size units: pixels Resolution: 72.000 Resolution units: pixels/inch Device aspect ratio: 1.778 Pixel aspect ratio: 1.000 Render Options	 Render Settings Render Layer: masterLayer Render Using: Arnold Renderer Common / Arnold Renderer / System / AOVs / Diagnostics Sampling Camera (AA) Samples : 4 Diffuse Samples : 4 (max : 8) Glossy Samples : 0 (max : 0) Refraction Samples : 0 (max : 0) Total (no lights) : 8 (max : 12) Camera (AA) 2 Diffuse 1 Glossy 0 Refraction 0 SSS 0 Volume Indirect 2 Lock Sampling Pattern Use Autobump in SSS Clamping Filter Ray Depth Total 2 Diffuse 2 Glossy 0 Reflection 0	 Render Settings Render Layer: masterLayer Render Using: RenderMan/RIS Common / Sampling / IPR Sampling / Features / Passes Ray Type / Per Pixel / Max Per Pixel / Total Rays Camera 4 4 8.29 Millions Indirect 4 4 8.29 Millions Total 8 8 16.59 Millions Min Samples 4 Max Samples 4 Pixel Variance 0.00500 Dark Falloff 0.02500 Incremental Pixel Filter Mode: Weighted Integrator Integrator: Path Tracer Max Path Length 2 Sample Mode: bxdf Light Samples 2 Bxdf Samples 0 Indirect Samples 1 Allow Caustics	 Render Settings Render Layer: masterLayer Render Using: 3Delight (OSL) Common / 3Delight Active Render Pass: OSL_Render_Pass Quality Shading Samples 1 Pixel Samples 2 Pixel Filter: gaussian Filter Width 2.000 Motion Samples 1 Max Depth 2 Max Distance 1000.000 Scene Elements Frame Range Image Resolution and Crop Image Layers Render Engine Performance
Remarks		Area lights have 1 sample each.	Min Samples = Max Samples. Light Samples = 2, seems to give closer results to Arnold in terms of ray counts.	Only shading samples to set for sampling BRDFs. *3Delight OSL* tal s care of the rest.

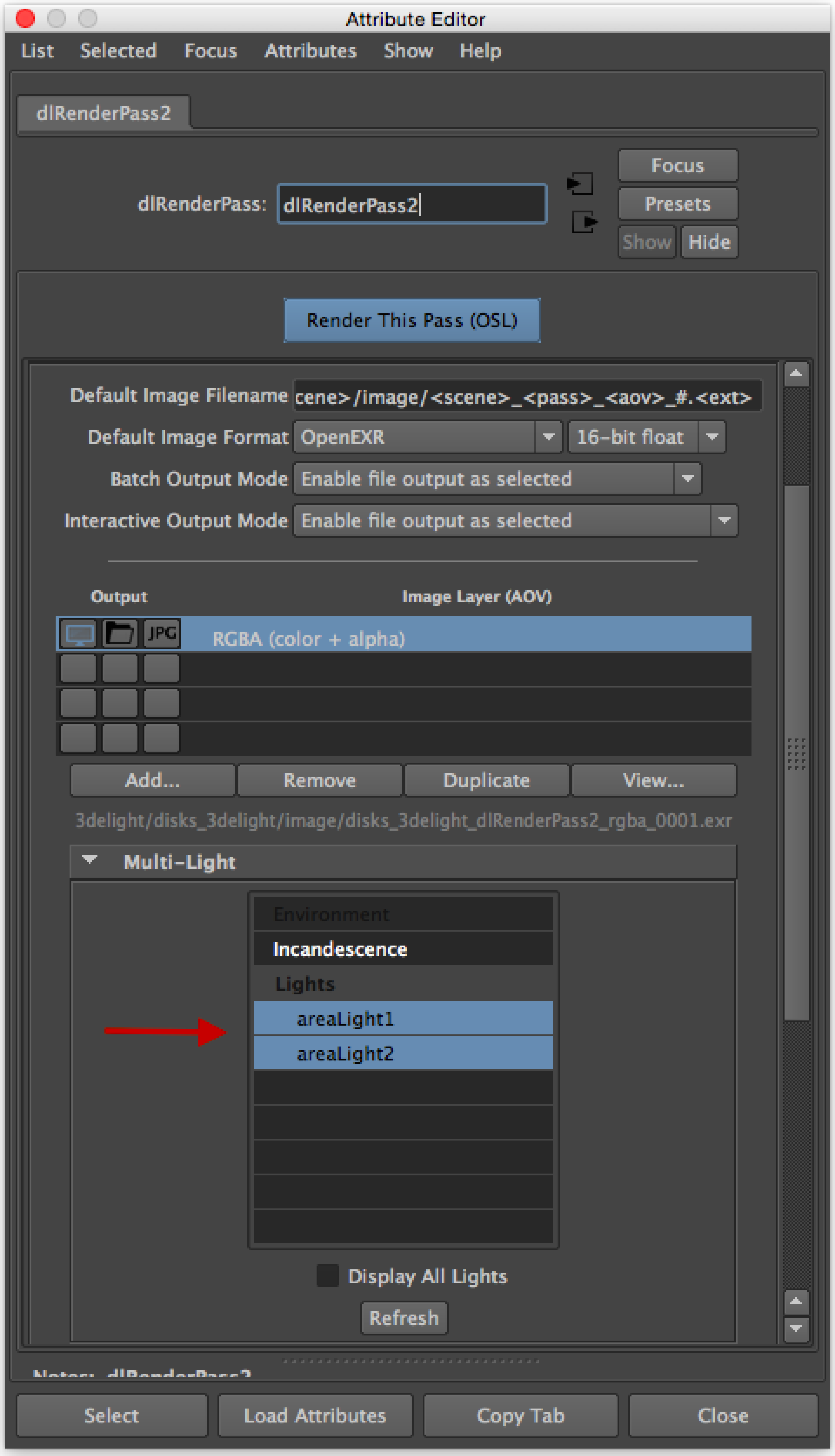
The Results

	3Delight OSL	**Arnold**	**RenderMan/RIS**	**3Delight RSL**
Image				
Total number of rays	28.5 Millions	31.6 Millions	30.9 Millions	37.4 Millions
Time on Linux	**33 seconds**	50 seconds	74 seconds	179 seconds
Compared to 3Delight OSL *	1	1.36 times slower	2.07 times slower	4.16 times slower

** Taking into account the total number of traced rays*

Multi-Light Rendering Performance Test using *3Delight OSL*

3Delight's multi-light feature allows you separate lighting components per light or per sets of lights. It is a subset of the LPE feature, in OSL parlance. Multi-light rendering is enabled in the *Image Layers* section of the OSL render pass; just select the RGBA layer and select the two area lights. This test evaluates the performance degradation when using this feature.



dlRenderPass: dlRenderPass2

Default Image Filename cene>/image/<scene>_<pass>_<aov>_#.<ext>
Default Image Format OpenEXR 16-bit float
Batch Output Mode Enable file output as selected
Interactive Output Mode Enable file output as selected

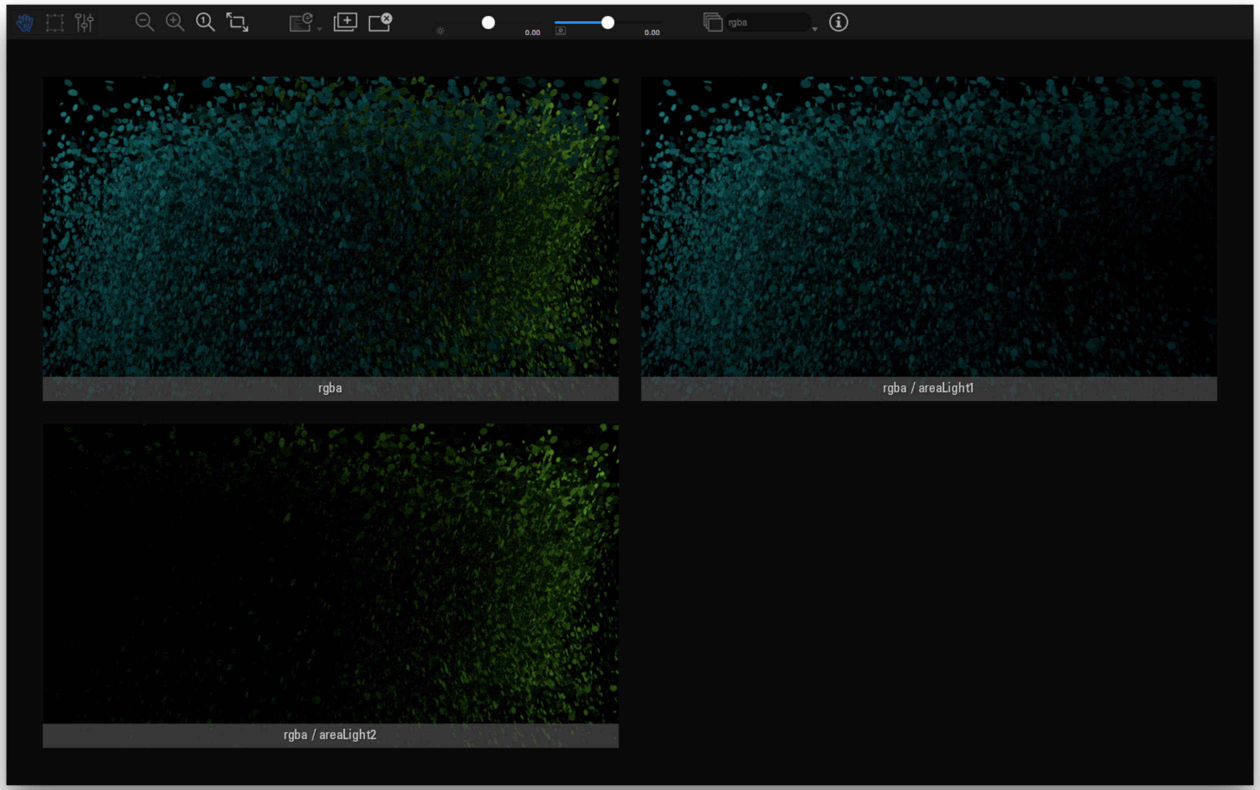
RGBA (color + alpha)

3delight/disks_3delight/image/disks_3delight_dlRenderPass2_rgba_0001.exr

Environment
Incandescence
Lights
areaLight1
areaLight2

Select the two area lights to split the RGBA into two additional outputs (one for each area light)

In the snapshot of *3Delight Display* below, you can see the resulting beauty (top left) and the two area lights:



Render time is almost identical; 34 seconds for the entire render.

Conclusion

- 3Delight's OSL-based path-tracer provides outstanding performance results. Building a rendering core around the OSL shading system, as opposed to just "integrating" OSL in the renderer, seems to have given *3Delight* a good performance advantage.
- Multi Light in *3Delight* doesn't have a perceptible adverse effect in this test (although from experience, rendering much more Multi Light AOVs, one can expect a 1-3% penalty).
- This particular test shows the limitations of RSL shaders and the limitations of the JIT architecture they use.

Resources

Here are the links to reproduce the tests on your system.

	Arnold	**3Delight OSL**	**3Delight RSL**	**RenderMan/RIS**
Maya Scene (ASCII)	disks_arnold.ma	disks_3delight.ma	disks_3delight.ma (same)	disks_prman.ma
Statistics	stats_arnold.1.txt	stats_3delight.txt	stats_3delight_RSL.txt	stats_prman.txt, stats_prman.xml
EXR images (linear)	disks_arnold.exr	disks_3delight.exr	disks_3delight_RSL.exr	disks_prman.exr

3Delight packages used for testing:

http://www.3delight.com/packages/testing/WPOIOF/3delight-12.0.107-setup-x64.exe
http://www.3delight.com/packages/testing/WPOIOF/3delight-12.0.107-Linux-x86_64.tar.xz
http://www.3delight.com/packages/testing/WPOIOF/3delight-12.0.107-Darwin-Universal.dmg