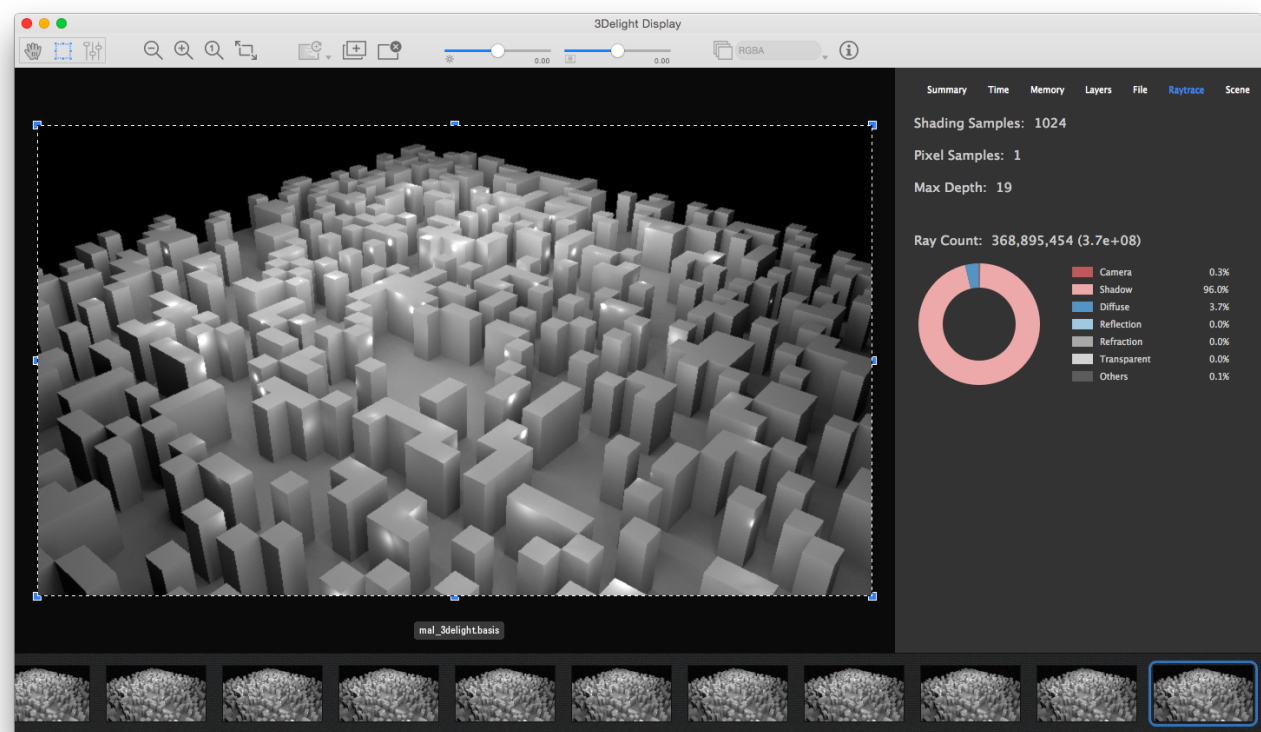
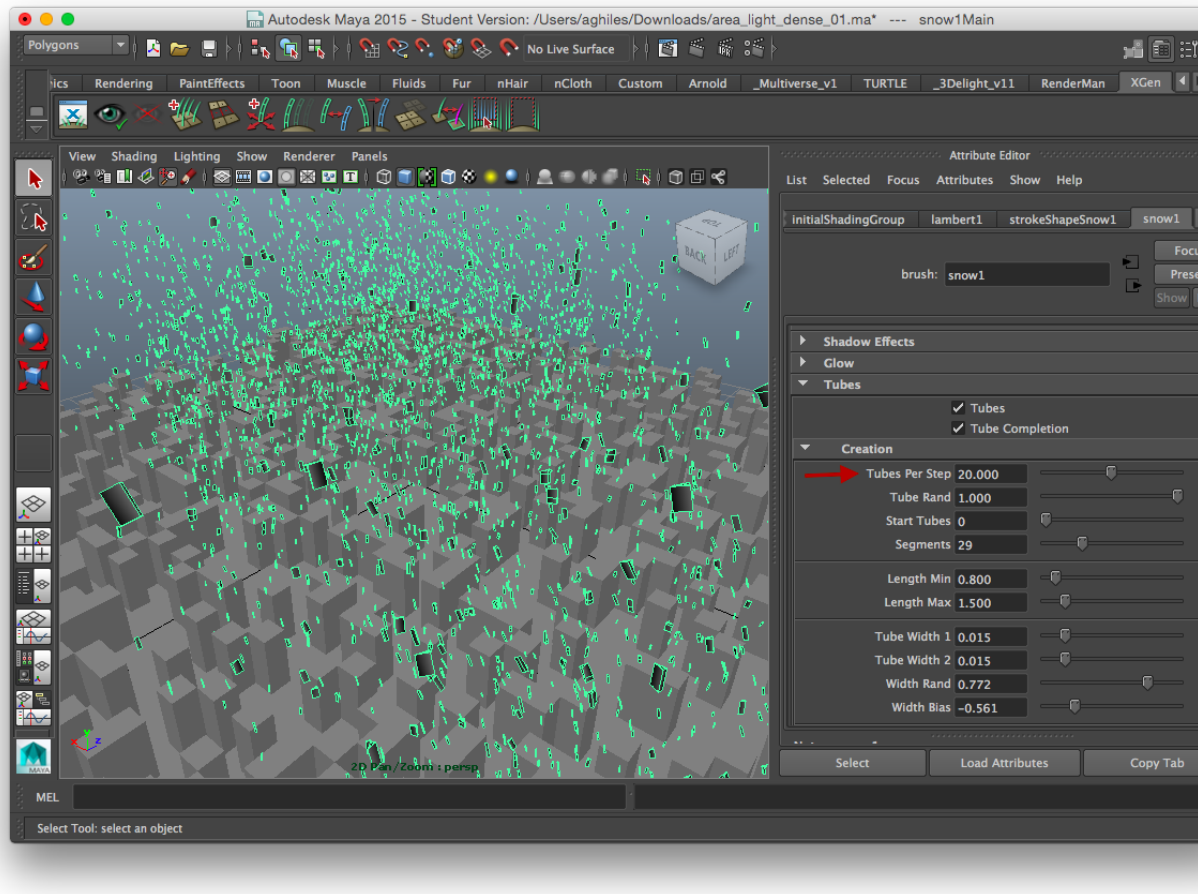


Geo Light Sampling Algorithms Comparison



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



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".

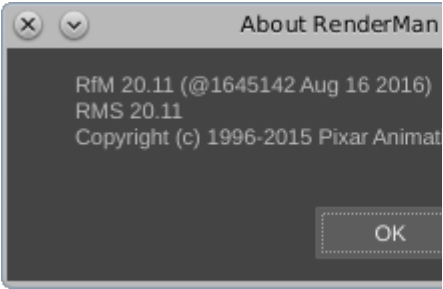
- Only diffuse reflectors.
- We will disable any adaptive sampling so to make sure we have meaningful ray-counts.
- We will use only direct lighting to estimate the geometric area light contribution. In the statistics files for each renderer, one can see that we have only one path length.
- The light 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/R
--	--------	-------------

Version		
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.

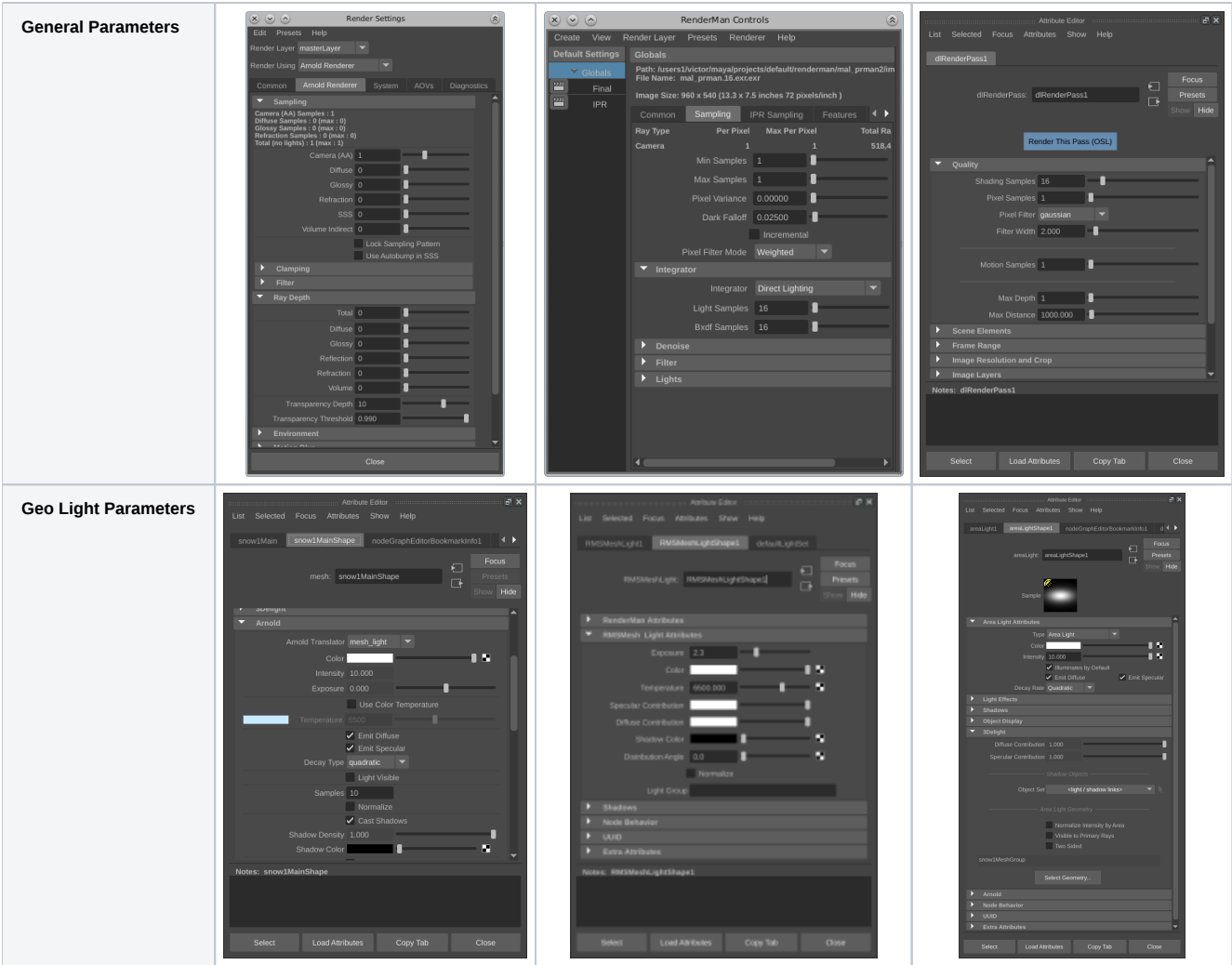
The Setup

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
--	--------	---------------	--------------



Notes

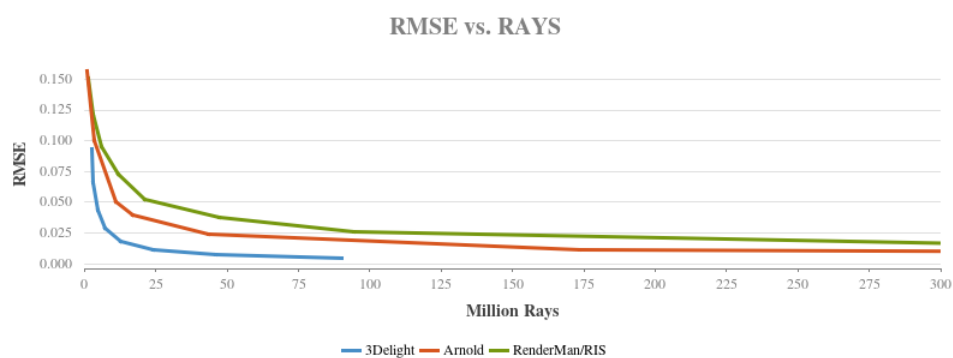
Arnold — For light samples, *Arnold* uses effective sample counts that are proportional – within a constant – to the *square* of the user specified value. As we will see, this makes sense from a UI standpoint since the variance follows the inverse of the same rule in the case of *Arnold*. This makes 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.

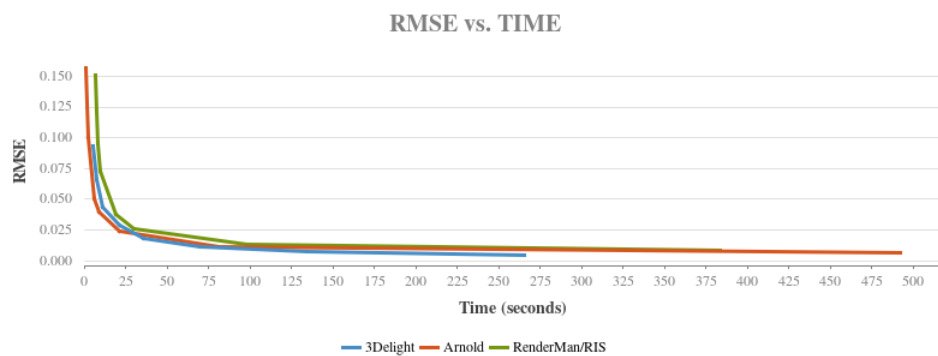
Results

The following graph gives a good idea on the convergence rate of the difference in light's samples.



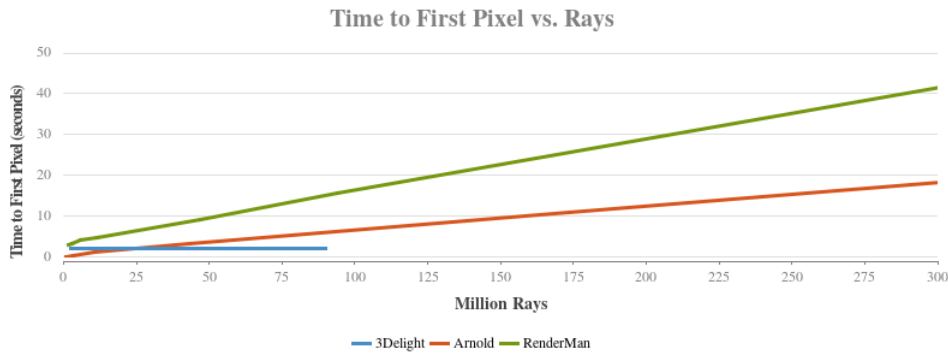
The follow

ing graph showing the time required to achieve a certain quality. For users' perspective, this is an important qua



n
tit
y.

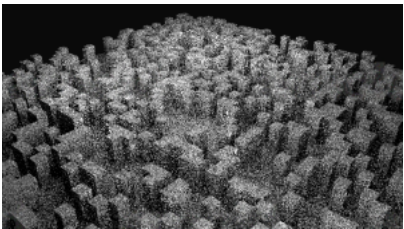
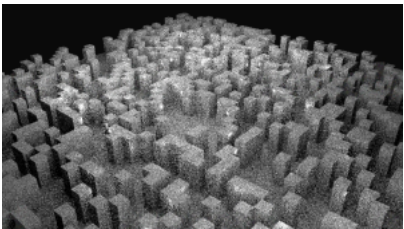
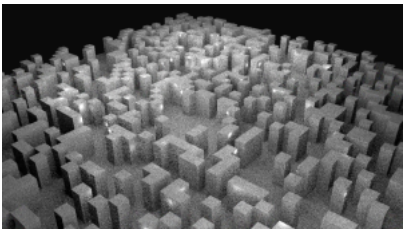
T
h
e
f
o
l
l
o
w
i
n
g
g
r
a
p
h
s
h
o
w
s
h
o
w
m
u
c
h
t
i
m
e
i
t
t
a
k
e
s
t
o
b
u
i
l
d
t
h
e
a
c
c
e
l
e
r
a
t
i
o
n
d
a
t
a
s
t
r
u
c
t
u
r
e
d
e
p
e
n
d
i
n
g
o
n
s



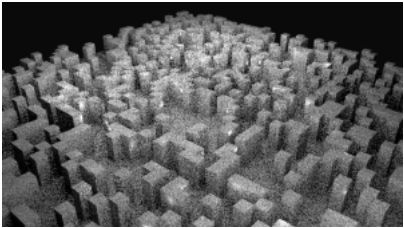
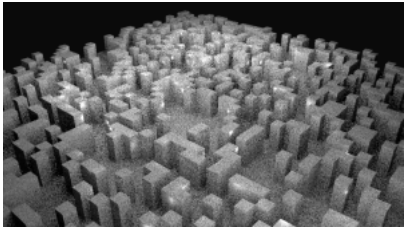
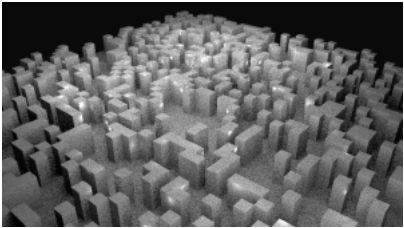
a m p l e /r a y c o u n t.	
---	--

Arnold

*TTFP = Time to first pixel.

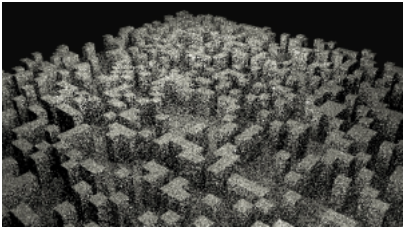
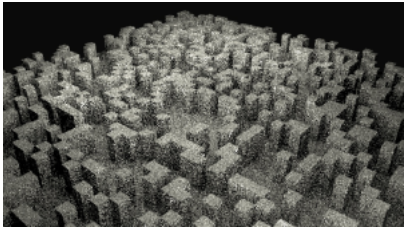
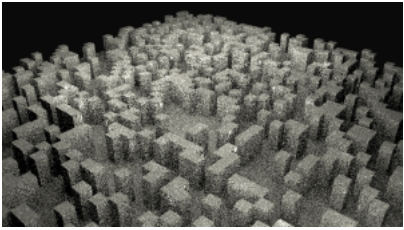
Samples (effective)	2 (1.23)	4 (4.91)	8 (19.64)	
Image				
Time	1 sec.	2 sec.	6 sec.	9
TTFP*	0 sec.	0.35 sec.	1.2 sec.	2
Shadow Rays	0.678 M	3.26 M	10.8 M	1
RMSE	0.15699	0.100115	0.0501787	0

3Delight

Samples	2	4	8	:
Image				
Time	5.24 sec.	7.18 sec.	11.21 sec.	20
TTFP	2 sec.	2 sec.	2 sec.	2 :
Shadow Rays	2.45 M	3.13 M	4.51 M	7.:
RMSE	0.0933142	0.0658266	0.0441248	0.0

RenderMan/RIS

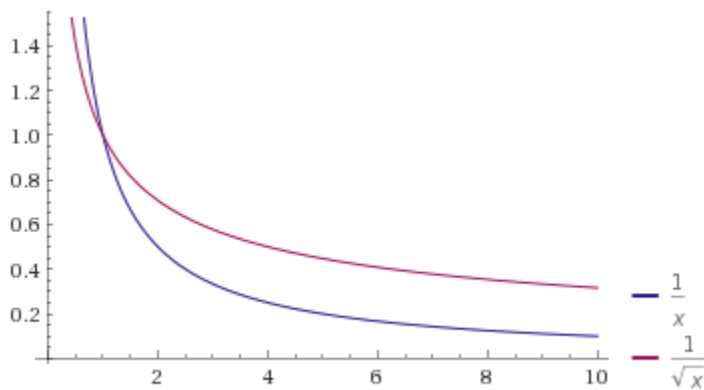
Samples	1	2	4	8
---------	---	---	---	---

Image				
Time	6.74 sec.	7.23 sec.	7.99 sec.	9.4 sec.
TTFP	3.1 sec.	3.2 sec.	4 sec.	4.7 sec.
Rays	1.47 M	2.94 M	5.88 M	11.76 M
RMSE	0.151125	0.121487	0.0953649	0.0714286

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 $\sim 1/x$
 - b. *Arnold* Variance $\sim 1/\sqrt{x}$
 - c. *RenderMan/RIS* Variance $\sim 1/\sqrt{x}$ (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 <i>3Delight</i> and <i>Arnold</i>	