

4.0.1.2

Milestone 4.0.1

Enhancements

- **Added `enable_threaded_procedurals`:** This global option allows users to enable or disable the parallel loading of procedural nodes. This is normally enabled by default, but it can be disabled as a convenience to avoid crashes with procedural code that is not yet thread-safe (e.g. Katana). Rather than disabling this option, we urge developers to make their code thread-safe to take full advantage of the capabilities of modern hardware. (#2637)
- **Reduced peak memory usage for deformation motion blurred ray accels:** The peak memory usage while building the ray accel data structure for deformation motion blurred objects has been significantly reduced, often by half. This can help when rendering large deformation motion blurred objects such as multi-million polygon/hair objects. Note that this memory peak is usually not reported by the Arnold statistics, but the effect of this optimization can still be noticed by faster build times when previous versions ran out of physical memory. (#2631)
- **Faster deformation motion blurred ray accels:** Objects with extreme deformation motion blur should now render faster. In one particular production scene with exploding, very fast moving particles, this resulted in a 200x speedup. Scenes with average deformation won't see much or any speedup. This does however result in about a 25% slower build time. (#2635)
- **Faster ray accels:** Fixed a bug in ray accel creation that was generating inefficient (but still valid and correct) ray accels. Render time should be several percent faster; we've seen 7% in our tests for architectural scenes. (#1526)
- **Faster auto-transparency AOV blending:** The AOV composition code has been optimized, greatly reducing the overhead incurred when the `enable_aov_composition` render option is set. Scenes with shaders that export many different AOVs yet only actually write a few to disk will see the most benefit. (#2634,#2639)
- **Faster gamma correction in handle-based texture lookups:** The inverse-gamma correction for handle-based lookups of LDR texture maps has been optimized resulting in up to 1.2x speedups in simple scenes. The older filename-based texture lookups are unaffected by this change, as are scenes that have no inverse-gamma correction, i.e. HDR maps or LDR maps with `options->texture_gamma == 1.0`. (#2628)
- **Faster approximate `expf()`:** The internal `expf()` function has been approximated and optimized, resulting in ~3% speedups in simple scenes which heavily use that function (gaussian filters, fog shaders, etc). (#2629)

API additions

- **Arithmetic operators for `AtPoint2`:** `AtPoint2` was missing the overloaded operators (`+`, `*`, `=`, etc...) present in `AtPoint`. (#2621)

```
AtPoint2 a, b;
...
AtPoint2 c = a + b;
```

- **`rgb()` accessor in `AtRGBA`:** It is now possible to get and set the RGB part in an `AtRGBA` with the `rgb()` accessor, and use it for common `AtRGB` operations. This can be useful in code that needs to mix RGB and RGBA colors: (#2633)

```
AtRGBA rgba1, rgba2;
...
AtColor c = rgba1.rgb() + rgba2.rgb();
rgba2.rgb() = rgba1.rgb();
```

Incompatible changes

- **Deprecated legacy utility functions:** The following scalar utility functions from `ai_types.h` have been deprecated: `MAP01()`, `MAP01F()`, `INVERSE()`, `ACOSF()`, `AiBerpScalar()`. As far as we can tell, they are unused, or in the case of `ACOSF()` superseded by a templated version that works on both floats and doubles. (#2622)
- **Templatized `BILERP()` and `BIHERP()`:** Following all the other utilities in `ai_types.h`, these two scalar utilities have been templated so that they can work on both floats and doubles. In addition, they now return a value instead of void. The old non-returning versions have been kept but deprecated. (#2622)

Bug fixes

Ticket	Summary	Component	Owner	Priority	Version	Created
#1526	SAH equation in BVH builder is wrong	arnold	thiago	major	3.0	2 years
#2620	kick -licensecheck has stopped giving license details	kick	oscar	major	3.3	5 weeks
#2622	ai_types.h is not exposed in the Doxygen html	arnold	marcos	major	3.3	5 weeks
#2632	Symbols are not correctly stripped in libai.dylib	arnold	oscar	major	3.3	3 weeks
#2642	wrong self-hit treatment in cone/cylinder	arnold	oscar	major	3.3	2 weeks
#2644	'box' count is not reported in the initial log messages	arnold	oscar	trivial	3.3	2 weeks

Ticket	Summary	Keywords	Component	Owner	Priority	Milestone
#2656	always_linear metadata doesn't work when writing to .ass	4.0.1.1	arnold	angel	critical	4.0.2
#2646	intersection precision problems in cone primitive	4.0.1.1	arnold	oscar	major	4.0.2
#2652	Handle ABS(int)	4.0.1.1	arnold	thiago	major	4.0.2
#2655	Linking to a component in an array element not working	4.0.1.1	arnold	angel	major	4.0.2

Ticket	Summary	Keywords	Component	Owner	Priority	Milestone
#2662	Halo artifacts with 'importance_sampling' in 'volume_scattering'	framestore, 4.0.1.2	arnold	oscar	critical	4.0.2
#2665	Bug when writing to .ass shaders with component links	4.0.1.2	arnold	angel	critical	4.0.2
#2641	Threading Issue when generating SSS pointclouds	framestore, 4.0.1.2	arnold	oscar	major	4.0.2
#2661	missing Doxygen docs in AiSSSEvaluateIrradiance()	4.0.1.2	arnold	thiago	major	4.0.2