3.3.3.2

Milestone 3.3.3

Enhancements

- **Improved threading for SSS**: The multi-threading performance of the lazy SSS mode has been greatly improved; some users are reporting a 1.5-2x speedup in their SSS renders. Because the lazy mode is now always much more efficient than the non-lazy mode, we have decided to make it the default and eliminate the non-lazy codepath and the sss_lazy_evaluation option. (trac#1813)
- **Adaptive maximum recursion depth for curves**: The curves node will now adaptively select the recursion depth used for its intersection test in *ribbon* mode depending on the curvature of the segment. This can provide up to a 20% performance increase depending on the shape of the curves. Straighter curves will see a greater performance increase. There is no performance increase for the *thick* mode. *Note: This change may result in small differences when comparing to images rendered with previous versions, particularly at the tips of wide ribbons.* (trac#1917)
- **Probabilistic transparency for curves**: A new render option has been added called auto_transparency_probabilistic that activates an alternate method of calculating transparency for shaders supporting it that can result in many less rays being cast from semi-transparent surfaces. This works particularly well with the curves.min_pixel_width feature as well as with opacity-mapped curves. Experiments with curves indicate a substantial performance increase using this method at the cost of some additional grain that tends to disappear at a reasonable rate as the AA setting is increased. (trac#1968)

> *A new API function has been provided for those shaders that would benefit from the alternate transparency calculation method called AiShaderGlobalsApplyOpacity(). However at the time being, of the built-in shaders only the hair shader has been adapted to use this new transparency method. Here is a code example of how this can be used in a shader:*
>
> ```
> shader_evaluate
> {
>    // Evaluate opacity
>    AtRGB opacity = AiShaderEvalParamRGB(p_opacity);
>
>    // Early exit if the apply opacity function returns true.
>    if (AiShaderGlobalsApplyOpacity(sg, opacity))
>       return;
>
>    // The rest of the shader code..
>    ...
> }
> ```

- **Optimized points primitive**: The points primitive now renders slighty faster. Additionally, non-motion blurred points submitted with motion keys are now simplified to only use a single key. (trac#1979)
- **Shading network support in volume_scattering.density**: You can now attach an arbitrary shader network into the density parameter. This can be used to "fill" 3D space with non-homogeneous fog. For example, you can use a fractal 3D shader that shades based on sg->P. (trac#1875)
- **Extended roughness range in Ward BRDF's**: The roughness of the Ward and Ward-Duer BRDF's can now be set to values as low as AI_EPSILON (1.0e-4). The previous lower limit was 0.01 (1.0e-2), which was insufficient for very sharp anisotropic reflections. (trac#1915)
- **Option to disable internal reflections in standard shader**: There is a new parameter in the standard shader called enable_internal_reflections. Disabling this option avoids the exponential explosion of reflective+refractive rays that tends to happen in glass materials, resulting in a big speedup with only relatively minor differences in the look of the glass. This is currently set to TRUE to preserve the look in existing scenes but will change to FALSE in a future release. (trac#1964)
- **Support for loading .obj geometry files in procedurals**: Geometry in .obj format can now be loaded using the procedural node. There is support for triangles, arbitrary polygons, UV's, multiple shaders per mesh (shader is directly referenced from .obj material name) and compressed .obj files (files with .obj.gz extension). (trac#1865)
- **Ray and memory stats improvements**: We now report the memory used by loading plugins (i.e. dynamic libraries containing 3rd party shaders) and by shader message passing. In addition, we now report the number of lighting calculations: lightloops (cached, direct, indirect), indirect diffuse, indirect glossy. Finally, in the ray stats, the volume shadows are now reported separately from regular shadows. (trac#1991)
- **Consolidated frequent warnings**: There are a number of debug warnings that keep being ignored (because they are very frequent). Examples include passing identical deformation motion blur data to polymesh and curves nodes, or passing identity matrices. These warnings have been aggregated and we now emit "performance warnings" instead, so that they don't fill up the log files. (trac#2007)

API additions

- **AiShaderGlobalsApplyOpacity()**: This is used for probabilistic transparency. See above for more details including a usage example. (trac#1968)

Incompatible changes

- **sss_lazy_evaluation on by default**: This feature is now always enabled so the option has been removed. (trac#1996, #1813)
- **Removed GI_quickshade**: This feature is unused and has been deprecated for a while. All related parameters and options have been removed: (trac#2009)
    - polymesh.GI_quickshade
    - options.GI_quickshade_dist
    - options.GI_quickshade_raymask

Bug fixes

#1971	Memory leak when tracing rays in "free" render mode
#1978	crash in AI_RENDER_MODE_CAMERA mode using uninitialized shaderglobals struct
#2076	Bad intersection test on disk points
#2074	Setting light.filters array to NULL causes a crash
#2070	Remove several memory leaks found in testsuite
#2055	node_update is not called in the nodes of a procedural when load_at_init is set to false
#2049	Crash when a hair material has diffuse cache on and a SSS cache flush is done after a render
#1935	kick overwrites the same image file with every frame (with -turn <n> option)
#922	kick doesn't override the output file with -o
#2050	custom plugin filters crash with negative AA samples
#2047	Linking a shader component (such as .r) discards the static value for the other components
#2045	Modified textures are being permanently invalidated
#2042	Standard shader produces bright dots at grazing angles
#2034	Problem loading .obj files with Windows-style EOLs
#2033	kick -info crashes with parameter names > 32 chars
#2032	Plugins in '.' are not loaded when "-l" option an/or plugin searchpath is specified
#2019	Writing instance parameters to .ass is broken
#2006	memory leak in persp_camera
#2005	Potential leak/crash when changing bucket_scanning interactively
#2004	curves should never return a 0 length normal
#2002	Statistics can use the wrong number of threads
#1998	avoid false sharing in light LSD structs
#1995	increase texture_max_memory_MB to 512
#1994	Crash when using AiRadiance or AiIrradiance
#1992	memory pool calls realloc() with an invalid address
#1989	make it impossible to write AOVs for shadow rays
#1988	lower threshold when calculating surface derivatives
#1987	bug in regular patch classification during adaptive subdivision
#1981	Empty arrays not properly written to .ass files
#1977	ERROR | Inside box.c -- could not find box normal
#1976	valgrind warnings in patch subdivision
#1975	standard shader produces black/white dots in glossy reflections
#1974	motion blur artifacts in low primitive-count scenes
#1973	Crash on AiAOVSetRGB
#1972	darkening when MIS samples are increased
#1967	corrupt rays from AiCookTorranceIntegrate
#1944	output drivers don't allow extension different than image file format