3.3.1.0

Milestone 3.3.1

Enhancements

- **Subdiv optimizations**: The performance and temporary memory usage of subdivision surfaces has been improved by rewriting some internal data structures. Subdivision time is up to 4x faster on some scenes, while using up to 50% less peak memory. (trac#1888)
- **Faster writing of .ass.gz files**: The zlib compression settings are now optimized for speed instead of file size. The new .ass.gz files are slightly bigger (~5-10%) but are generated 3-4x faster. (trac#1889)
- **Faster Oren-Nayar BRDF evaluation**: Optimized the implementation of AiOrenNayarBRDF() by reducing the amount of expensive trigonometric function calls. (trac#1898)
- **Support for writing .ass files after render**: There is a new parameter in the global options called preserve_scene_data, disabled by default. If enabled, it will keep the original geometry data (normally removed after node initialization to save memory) and allow you to save the scene to .ass even after all the multiple re-renders and modifications that happen in an interactive session. This option is only intended for interactive modifications of scenes (e.g. light/shader tweaking), and not for final or batch rendering. It has a tradeoff, as the memory occupied by the scene will be higher when set. (trac#1892)
- **kick waiting for keypress**: There is a new -nokeypress command-line flag that makes kick return immediately after rendering instead of sitting there waiting for a keypress. (trac#1891)
- **Upgraded OIIO to 0.8.3**: This release of OpenImageIO includes many fixes like improved TIFF error messages, --checknan option for maketx, timer improvements and a clamping/wrapping fix for .png output files when alpha > color. (trac#1902)

API additions

- **New microfacet BRDF's**: Added two new physically-based BRDF's for shader-writers to use. These are modern microfacet-based BRDF's that both exhibit an off-specular peak and reduce the amount of edge darkening apparent in older, Phong-based BRDF's (like the stretched-Phong BRDF used in the standard shader).
    - Cook-Torrance microfacet BRDF (trac#1862): This is an implementation of the Cook-Torrance BRDF based on the Beckmann microfacet distribution. The new API is:
      ```
      AI_API AtFloat AiCookTorranceBRDF(const AtVector *L, const AtVector *V, const AtVector *N, const
      AtVector *u, const AtVector *v, AtFloat rx, AtFloat ry);
      AI_API AtColor AiCookTorranceIntegrate(const AtVector *N, AtShaderGlobals *sg, const AtVector
      *u, const AtVector *v, AtFloat rx, AtFloat ry);
      ```

      Notes: This implementation omits the Fresnel term from the classic BRDF's equation, so it's up to the shader writer to add Fresnel effects to the shaders that use this BRDF. Also, although parametrized for anisotropy, this BRDF is currently isotropic so the u, v and ry parameters are ignored for the time being.
    - Improved Ward-DÃ¼r microfacet anisotropic BRDF (trac#1893): This is an implementation of the Ward-Geisler-Moroder-DÃ¼r BRDF, which is an improvement upon the Ward-DÃ¼r BRDF, which is itself an improvement upon the classic Ward BRDF. The new API is:
      ```
      AI_API AtFloat AiWardDuerBRDF(const AtVector *L, const AtVector *V, const AtVector *N, const
      AtVector *u, const AtVector *v, AtFloat rx, AtFloat ry);
      AI_API AtColor AiWardDuerIntegrate(const AtVector *N, AtShaderGlobals *sg, const AtVector *u,
      const AtVector *v, AtFloat rx, AtFloat ry);
      ```
- **Metadata iterator**: The new AtMetaDataIterator can be used to traverse all metadata attached to a node or one of its parameters. The API is similar to other existing iterators, as can be seen in the following usage example: (trac#1887)

  ```
  AtMetaDataIterator *iter = AiNodeEntryGetMetaDataIterator(node_entry, "some_parameter");
  while (!AiMetaDataIteratorFinished(iter))
  {
     AtMetaDataEntry *entry = AiMetaDataIteratorGetNext(iter);
     printf("%s\n", entry.name);
  }
  AiMetaDataIteratorDestroy(iter);
  ```

  We can choose whether we want to iterate over the metadata attached to a specific parameter (as in the previous example) or pass NULL as the parameter string to get metadata attached to the node itself:

  ```
  AtMetaDataIterator *iter = AiNodeEntryGetMetaDataIterator(node_entry);  // optional second argument
  defaults to NULL
  while (!AiMetaDataIteratorFinished(iter))
     ...
  ```
- **AOV read API**: We have implemented a set of API functions to read data from an AOV, similar to the existing API to set data. The new API is AiAOVGet*(). (trac#1907)
- **AI_LOG_COLOR**: Added flag AI_LOG_COLOR in ai_msg.h to enable or disable the use of different colors for log messages based on severity. (trac#1877)

Incompatible changes

- **Oren-Nayar roughness**: The roughness parameter in AiOrenNayarBRDF() and AiOrenNayarIntegrate() has been normalized to the 0-1 range, which makes it easier to tweak and texture-map. Previously the range was 0-Pi. (trac#1911)

- **Walter fixes for Ward BRDF**: As described in Bruce Walter's widely-known technical report, the sampling equations from Ward's original paper were incorrectly weighted uniformly in all directions, making the results from AiWardIntegrate() overly bright and not physically plausible. We have applied Walter's fixes, but note that these corrected equations will severely darken the results from AiWardIntegrate() (especially at grazing angles) and it is recommended that shader writers use the newly introduced Improved Ward-DÃ¼r BRDF instead. (trac#1893)

Bug fixes

#1899	crash in catclark code when face_visibility is false on all faces
#1814	Add signal handling in Linux to do scene cleanup on termination
#1903	memory usage stats overflowing on Windows x64
#1913	PNG driver adds noise on fully saturated colors
#1912	flickering in SSS in static geometry
#1906	Missing operator function implementation in AtVector
#1904	problems piping commands to kick on Linux
#1901	Light nodes causing a crash when scene is aborted after initialization
#1900	print full backtrace when crashing in multiple threads
#1896	Bug in spot lights MIS
#1894	AiNodeSetRGBA() is not honoring the always_linear metadata
#1890	crash in AiNodeSetRGB/RGBA when passed a NULL node
#1884	reject attemps to declare varying user-data on instances
#1883	Bad ray intersection in quad light MIS
#1881	light sample cache should flush when traceset changes
#1880	light sample cache should check sg->skip_shadow
#1879	autobump should operate in the Displacement context
#1878	potential flickering in SSS at certain sample settings
#1874	abort and display message bank contents when full
#1873	crash with a fatal message when nans are encountered in displacement
#1872	off-by-one error in hair caching code
#1870	hermite basis matrix in the curves primitive is incorrect