

5.1.1.3

5.1.1.3

Bug fixes

- #7325 procedural loading a .ass containing many nodes that reference other nodes is very slow
- #7076 Rename user data parameters that clash with shape nodes in alembic expansion
- #7286 Alembic object transform
- #7303 Python binding fixes for functions that return AtNode types
- #7341 Crash when computing the render stats for procedural
- #7419 make flatness check for quad_light more robust

5.1.1.2

Bug fixes

- #7259 linear normal subdivision: incorrect normals with multiple keys
- #7299 Regex/operators memory leak
- #6370 Procedural cache not working with custom namespaces
- #7230 Object "id" parameter overridden by procedural
- #7242 Fix alembic array attribute expansion for regular properties
- #7244 Crash when loading empty points through a procedural
- #7261 Alembic curves sets radius on curves and points to width
- #7267 linking to only some of the mesh or quad light's color components gives incorrect results
- #7289 Allow reading files marked with a UTF-8 encoding
- #7306 AiShaderGlobals() can cause crashes when called from multiple threads

5.1.1.1

Bug fixes

- #6771 Incomplete render when you restart after interrupting a render during displacement/subdivision
- #7180 `noice` crashes with separate float Z input file
- #7213 MaterialX: Displacements and mixing of generic/texture shaders in materials now supported using a shader type context
- #6926 MaterialX: Resolve environment variables in search paths
- #6993 `noice` should preserve metadata, display windows, and add `noice` args and version metadata
- #6994 `Noice` should preserve channel bit width and compression for outputs
- #7094 Procedural containing ginstances are evaluated before instanced node
- #7113 Contour lines shouldn't be drawn when `width_scale` is zero
- #7124 Memory leak when interrupting displacement
- #7127 Procedural ginstance matrices are accumulated at each render
- #7130 Render incorrectly aborted in `applyDevice` when GPU is not needed
- #7152 `noice` should exit when unsupported formats are used
- #7163 Array property fixes in alembic procedural
- #7176 Motion blur on instances of procedurals
- #7187 Fix op target selection in kick command line
- #7191 Slight error in checkerboard shader
- #7193 `noice` broken sequence handling when temporal denoising is not used
- #7198 Alembic procedural uses `inheritsXforms` property
- #7200 Output metadata: output correct metadata for single layer files

5.1.1.0

Enhancements

- **Improved stability with incompatible OptiX versions:** We have removed the hard dependency on the Nvidia OptiX library needed for GPU denoising. We can now prevent DLL conflicts when users have installed renderers from other vendors that link to a different/older version of the OptiX libraries, which was causing plugins (such as MtoA) to refuse to load. (#6959)

- **More efficient texture mapping:** EXR textures can now be read more quickly. In addition, the `texture_diffuse_blur` option is now defaulted to 0 so that sharper and more accurate diffuse reflections can be seen without texture performance being negatively impacted. This option is now deprecated and, if no problems are raised, in a future release we will remove this and the other `texture_*_blur` options. (#4692, #6605, #6890)
- **noise sequence handling and extra frames:** A new argument (`-f n` or `--frames n`) has been added to `noise` to denoise image sequences. Another new argument (`-ef n` or `--extraframes n`) specifies how many additional source frames before and after the current one should be used, for improved stability in animation sequences. (#6899) In this example, `noise` will run for 10 consecutive frames starting at frame 5, taking into account two frames before and two frames after each source frame (e.g. source frame number 5 will take into account frames 3, 4, 5, 6, and 7):

```
noise -i mysequence.0005.exr -o denoised.0005.exr -f 10 -ef 2
```

- **noise support for AOVs from multiple files:** `noise` now supports multiple input files to be denoised. These can be specified by using the `-i` (`--input`) argument as many times as needed. Multiple output files are also supported to preserve the original AOV layout. Denoised output files matching the input files can be specified with the `-o` (`--output`) argument as many times as needed. This is compatible with sequence arguments (`-f` and `-ef`) and with manually specified extra frames. As always, files from the frame to be denoised should come first. (#6826) In the following example, the input feature AOVs and the input AOVs to be denoised were rendered in separate files, and denoised results will be saved in two separate files to preserve the original AOV layout:

```
noise -i RGBA.exr -i light_AOVs.exr -i features.exr -l light_AOV1 -l light_AOV2 -o denoised_RGBA.exr -o denoised_light_AOVs.exr
```

- **noise AOV denoising:** AOVs are now fully denoised individually for increased quality at the expense of run time. Also, per-AOV variances will be used if present. Every RGB channel to be denoised will benefit from an associated variance AOV. Additionally, RGBA is no longer required to denoise AOVs from a given EXR file. (#7011, #7015)
- **noise threads:** a new command line flag (`-t` or `--threads`) has been added to specify the number of threads to use. This is only available on Windows and Linux. The flag works like the similar `kick -t <integer>`: a negative number will leave some cores unused (`-1` will leave one core unused), a positive number will use the specified number of cores, and 0 will use all cores. (#7017)
- **noise custom features:** for advanced users, the new flag (`-fe` or `--features`) specifies which AOVs will be used as features to guide denoising. For instance `-fe volume_albedo` will use `volume_albedo` to denoise instead of the default `diffuse_albedo`. (#7070)
- **More robust Alembic procedural:** We have improved the robustness of the built-in Alembic procedural with various bug fixes, including support for facevarying user data, and support for array properties. (#6909, #6937, #7025)
- **Support for Microsoft Azure Sovereign Clouds:** In addition to regular Azure cloud, Arnold batch rendering is now entitled in the Microsoft National Clouds. (#6960)

API additions

- `AiProfileGetFileName()`: To match `AiProfileSetFileName()`, we now also have the corresponding `AiProfileGetFileName()`. (#7031)

Incompatible changes

- `materialx.mtlx` renamed: The `mtlx` parameter of the `materialx` operator node, which should point to a `.mtlx` file on disk, has been renamed to `filename`. The old name `mtlx` is kept as a deprecated synonym so that existing scenes don't break. Also, if the file cannot be opened, the `materialx` node will now report an error (and abort the render) instead of a warning. (#7018)

Bug fixes

- #6844 Catch exceptions in MaterialX operator to not crash with invalid documents
- #6903 Incorrect cooker cache check prevents some upstream operators from cooking
- #6988 Noise: non ASCII characters in non-existing path cause crash on Windows
- #6999 Named MaterialX node graph outputs not correctly bound to shader inputs
- #6596 ``aov_write_float`` shader not respecting ``standard_surface.transmit_aov`` flag
- #6735 Enabling transmission affects rayDepth AOV
- #6849 Cannot render rgb masks for transmissive objects
- #6896 Object-To-World transform Uninitialized in OSL

- #6900 Noise should ignore precision errors in depth or normal buffers
- #6902 Procedural load crashing due to invalid nodes
- #6909 Match number of motion keys for normals and vertex list in alembic procedural
- #6913 Fix crash when optix denoiser DLLs are not found
- #6923 Crash when loading empty polymesh through a procedural
- #6924 Optix denoiser crashes after changing device because buffers still mapped
- #6928 Crash caused by old versions of NVML
- #6937 Add facevarying userdata support to alembic procedural
- #6938 crash due to an invalid/empty procedural within another procedural
- #6944 Crash in consecutive AiBegin sessions on machines with unsupported Nvidia drivers
- #6948 Evaluation of uninitialized debug closure when running aov shaders
- #6951 Crash when using AiNodeClone in procedurals with parallel_node_init
- #6952 Noise: +inf depth values cause issues in background
- #6970 Hang due to ginstances referencing a non-existing node
- #6977 standard_hair crash with specular tints set to 0
- #7006 kick -nokeypress doesn't work
- #7025 Array properties in alembic procedural not translated
- #7038 Typo in JSON shader stats
- #7061 Failure to write JSON stats when NaN or Inf encountered
- #7062 Using toon shader as an aov shader crashes when a background shader exists
- #7078 Noise: pixel NaNs when using parallel denoising with black RGBA
- #7083 warn when several drivers write to the same file
- #6942 AOV output string error when layer name matches scene node name

5.1.1.1

Bug fixes

- #6771 Incomplete render when you restart after interrupting a render during displacement/subdivision
- #7180 noise crashes with separate float Z input file
- #7213 MaterialX: Displacements and mixing of generic/texture shaders in materials now supported using a shader type context
- #6926 MaterialX: Resolve environment variables in search paths
- #6993 noise should preserve metadata, display windows, and add noise args and version metadata
- #6994 Noise should preserve channel bit width and compression for outputs
- #7094 Procedural containing ginstances are evaluated before instanced node
- #7113 Contour lines shouldn't be drawn when width_scale is zero
- #7124 Memory leak when interrupting displacement
- #7127 Procedural ginstance matrices are accumulated at each render
- #7130 Render incorrectly aborted in applyDevice when GPU is not needed
- #7152 noise should exit when unsupported formats are used
- #7163 Array property fixes in alembic procedural
- #7176 Motion blur on instances of procedurals
- #7187 Fix op target selection in kick command line
- #7191 Slight error in checkerboard shader
- #7193 noise broken sequence handling when temporal denoising is not used
- #7198 Alembic procedural uses inheritsXforms property
- #7200 Output metadata: output correct metadata for single layer files

5.1.1.0

Enhancements

- **Improved stability with incompatible OptiX versions:** We have removed the hard dependency on the Nvidia OptiX library needed for GPU denoising. We can now prevent DLL conflicts when users have installed renderers from other vendors that link to a different/older version of the OptiX libraries, which was causing plugins (such as MtoA) to refuse to load. (#6959)
- **More efficient texture mapping:** EXR textures can now be read more quickly. In addition, the `texture_diffuse_blur` option is now defaulted to 0 so that sharper and more accurate diffuse reflections can be seen without texture performance being negatively impacted. This option is now deprecated and, if no problems are raised, in a future release we will remove this and the other `texture_*_blur` options. (#4692, #6605, #6890)
- **noise sequence handling and extra frames:** A new argument (`-f n` or `--frames n`) has been added to `noise` to denoise image sequences. Another new argument (`-ef n` or `--extraframes n`) specifies how many additional source frames before and after the current one should be used, for improved stability in animation sequences. (#6899) In this example, `noise` will run for 10 consecutive frames starting at frame 5, taking into account two frames before and two frames after each source frame (e.g. source frame number 5 will take into account frames 3, 4, 5, 6, and 7):

```
noise -i mysequence.0005.exr -o denoised.0005.exr -f 10 -ef 2
```

- **noise support for AOVs from multiple files:** `noise` now supports multiple input files to be denoised. These can be specified by using the `-i` (`--input`) argument as many times as needed. Multiple output files are also supported to preserve the original AOV layout. Denoised output files matching the input files can be specified with the `-o` (`--output`) argument as many times as needed. This is compatible with sequence arguments (`-f` and `-ef`) and with manually specified extra frames. As always, files from the frame to be denoised should come first. (#6826) In the following example, the input feature AOVs and the input AOVs to be denoised were rendered in separate files, and denoised results will be saved in two separate files to preserve the original AOV layout:

```
noise -i RGBA.exr -i light_AOVs.exr -i features.exr -l light_AOV1 -l light_AOV2 -o denoised_RGBA.exr -o denoised_light_AOVs.exr
```

- **noise AOV denoising:** AOVs are now fully denoised individually for increased quality at the expense of run time. Also, per-AOV variances will be used if present. Every RGB channel to be denoised will benefit from an associated variance AOV. Additionally, RGBA is no longer required to denoise AOVs from a given EXR file. (#7011, #7015)
- **noise threads:** a new command line flag (`-t` or `--threads`) has been added to specify the number of threads to use. This is only available on Windows and Linux. The flag works like the similar `kick -t <integer>`: a negative number will leave some cores unused (`-1` will leave one core unused), a positive number will use the specified number of cores, and 0 will use all cores. (#7017)
- **noise custom features:** for advanced users, the new flag (`-fe` or `--features`) specifies which AOVs will be used as features to guide denoising. For instance `-fe volume_albedo` will use `volume_albedo` to denoise instead of the default `diffuse_albedo`. (#7070)
- **More robust Alembic procedural:** We have improved the robustness of the built-in Alembic procedural with various bug fixes, including support for facevarying user data, and support for array properties. (#6909, #6937, #7025)
- **Support for Microsoft Azure Sovereign Clouds:** In addition to regular Azure cloud, Arnold batch rendering is now entitled in the Microsoft National Clouds. (#6960)

API additions

- `AiProfileGetFileName()`: To match `AiProfileSetFileName()`, we now also have the corresponding `AiProfileGetFileName()`. (#7031)

Incompatible changes

- **materialx.mtlx renamed:** The `mtlx` parameter of the `materialx` operator node, which should point to a `.mtlx` file on disk, has been renamed to `filename`. The old name `mtlx` is kept as a deprecated synonym so that existing scenes don't break. Also, if the file cannot be opened, the `materialx` node will now report an error (and abort the render) instead of a warning. (#7018)

Bug fixes

- #6844 Catch exceptions in MaterialX operator to not crash with invalid documents
- #6903 Incorrect cooker cache check prevents some upstream operators from cooking
- #6988 Noise: non ASCII characters in non-existing path cause crash on Windows
- #6999 Named MaterialX node graph outputs not correctly bound to shader inputs

- #6596 `aov_write_float` shader not respecting `standard_surface.transmit_aov` flag
- #6735 Enabling transmission affects rayDepth AOV
- #6849 Cannot render rgb masks for transmissive objects
- #6896 Object-To-World transform Uninitialized in OSL
- #6900 Noise should ignore precision errors in depth or normal buffers
- #6902 Procedural load crashing due to invalid nodes
- #6909 Match number of motion keys for normals and vertex list in alembic procedural
- #6913 Fix crash when optix denoiser DLLs are not found
- #6923 Crash when loading empty polymesh through a procedural
- #6924 Optix denoiser crashes after changing device because buffers still mapped
- #6928 Crash caused by old versions of NVML
- #6937 Add facevarying userdata support to alembic procedural
- #6938 crash due to an invalid/empty procedural within another procedural
- #6944 Crash in consecutive AiBegin sessions on machines with unsupported Nvidia drivers
- #6948 Evaluation of uninitialized debug closure when running aov shaders
- #6951 Crash when using AiNodeClone in procedurals with parallel_node_init
- #6952 Noise: +inf depth values cause issues in background
- #6970 Hang due to ginstances referencing a non-existing node
- #6977 standard_hair crash with specular tints set to 0
- #7006 kick -nokeypress doesn't work
- #7025 Array properties in alembic procedural not translated
- #7038 Typo in JSON shader stats
- #7061 Failure to write JSON stats when NaN or Inf encountered
- #7062 Using toon shader as an aov shader crashes when a background shader exists
- #7078 Noise: pixel NaNs when using parallel denoising with black RGBA
- #7083 warn when several drivers write to the same file
- #6942 AOV output string error when layer name matches scene node name