3.3.4.2

Milestone 3.3.4

This release is the result of a lot of hard work over the last few months and it represents our best Arnold ever. There are many important fixes including OSX support, the flatness subdiv mode works incredibly well now, UV smoothing matches Softimage, physically_based is now enabled (most BRDF's were Pi times brighter than they should). Note that, although this is a binary-compatible release, the look of the images will change; please read these notes carefully and think twice before upgrading your shows.

Enhancements

- **Mac OSX support**: Beginning with this release, we will provide official binary packages for the Mac OSX 64-bit architecture (i.e. darwin_x86_64). This new platform is fully supported, including the license usage. Existing licenses can be used for Mac OSX clients. (trac#1923)
- **UV smoothing controls in Catmull-Clark subdivision**: Added a subdiv_uv_smoothing parameter to the polymesh object that allows the user to choose between the classic pin_corners UV subdivision mode and the new pin_borders mode that should prove a better match to the way some other software packages like Softimage handle UV coordinate subdivision. (trac#2102)
- **Improved pixel error heuristic in Catmull-Clark subdivision**: The estimation of pixel edge lengths was numerically unstable, which could result in "popping" from adaptive subdivision with certain camera moves. We have switched to a more robust estimate that reduces this problem. (trac#2121)
- **Improved adaptive Catmull-Clark subdivision rate estimates**: The estimate used to guess the iteration count for each patch of a subdivision mesh has been greatly improved, making the use of adaptive subdivision (i.e. subdiv_pixel_error > 0) more appealing than ever. In particular, theflatness heuristic is now much more accurate at detecting flat regions. This results in smoother results using fewer triangles in most cases. The new algorithm can use less *peak* memory, but most importantly it generates vastly better results, and so can easily generate less *retained* memory too, about 5x for the same quality. (trac#2124)
- **Importance sampling of textured quad lights**: When a shader is connected to a quad_light, we automatically construct importance sampling tables similar to those used in the skydome_light. This permits efficient sampling according to the luminance of the texture, which can greatly reduce sampling noise, specially when using HDRI textures. Like the skydome_light, the resolution of the table is controlled by the resolution parameter with a default value of 512. (trac#2026)
- **Corrected BRDFs for energy conservation**: A new physically_based render option has been added that causes the BRDF functions to preserve instead of generate energy under direct lighting. This resolves many of the divide-by-Pi issues that were previously occuring. Note that the default value for this parameter has been set to true which **WILL** change the lighting of scenes currently using the following affected BRDFs: (trac#2010)

```
AiOrenNayarBRDF()
AiLommelSeeligerBRDF()
AiModifiedPhongBRDF()
AiStretchedPhongBRDF()
AiOrenNayarIntegrate()
AiLommelSeeligerIntegrate()
AiDirectDiffuse()
```

- **User-selectable BRDFs in the standard shader**: We have added a specular_brdf parameter to the standard shader. You can use it to select the BRDF used to compute the specular highlight and glossy reflections. Available BRDFs are stretched_phong, cook_torrance andward_duer. All these BRDFs support MIS for efficient sampling. (trac#2012,trac#2013)
- **standard shader's specular controls more easily mappable**: There is a new specular_roughness parameter in the standard shader which is capable of the full range specular effects with an input parameter in the [0,1] range. This is much easier to drive through a texture map than the [0,+inf] parameter range used by the Phong_exponent parameter. (trac#2015)
- **Anisotropy controls in the standard shader**: Two new controls called specular_anisotropy and specular_rotation have been added to the standard shader which permit anisotropic effects in the BRDFs that support them. (trac#2091)
- **Improved BRDF sampling at grazing angles**: The annoying bright dots at grazing angles in the stretched-Phong BRDF used by the standard shader have been eliminated. This was mostly evident in low-poly meshes, not so much in highly tessellated meshes. In addition, the Ward-Duer BRDF now has less edge darkening in low-poly meshes. (trac#2042, trac#2107)
- **Beer's Law in the standard shader**: Added a transmittance parameter to the standard shader that, following Beer's Law, allows the user to specify how well light is transmitted (and tinted) through the object's volume. This will affect not only refraction but also shadow rays (if the object is not marked opaque). (trac#2017)
- **Atmospherics affect shadow rays**: Paving the way for volumetric shadows and self-shadows, attenuation/absorption effects from atmospheric shaders will now affect shadow rays which may result in differences in the rendered image. (trac#2029)
- **Physically based volume scattering**: The built-in volume_scattering shader has received a number of corrections to make it more physically plausible. The phase function has been normalized (dividing it by 4Pi if physically_based is enabled), and the equation for forward /backward scattering has been corrected. Consequently, we have removed the phase_function enum as it added little value. The scattering behavior is now completely controlled by the eccentricity parameter which has a [-1,+1] range. The default of 0 indicates isotropic scattering, positive values correspond to forward scattering, and negative values to backward scattering. (trac#2023)
- **Improved volume sampling**: The volumetric importance sampling code is now both simpler and more accurate, which drastically reduces volumetric noise near light sources. (trac#2101)
- **Added invert_normals parameter to ambient_occlusion shader**: When this mode is enabled, ambient occlusion rays are fired *inwards* int o the object. This can be used to simulate dirt and rust in corners. Of course, for this trick to work, you need a far_clip distance that is smaller than the object width, or at least some distance falloff (otherwise all the rays are blocked and the result is pure black). (trac#2054)
- **Added ambocc shading mode in utility shader**: A new ambient occlusion mode has been added to the utility shader, which is the default Arnold shader. The maximum distance is hard-coded to 100, while the sampling control is taken from GI_diffuse_samples. (trac#2083)
- **Added u and v color modes in utility shader**: In addition to the uv mode, which shows both U and V coordinates at the same time, it is sometimes useful to visualize either the U or the V coordinate alone. (trac#2071)

- **UV coordinates in the sphere primitive**: The sphere primitive now has support for U and V coordinates, using the standard polar parametrization, with the poles in the Y axis. This allows it to be textured, which can be useful for shaderball previews, etc. Tangents are also defined in sg->dPdU and sg->dPdV. (trac#2077)
- **Sprite/billboard support in the points primitive**: A new quad mode has been added to the points.mode parameter which is perfect for rendering large amounts of sprites/billboards. Particles in quad mode have a natural UV parameterization which allows them to be texture-mapped. (trac#1956)
    - For quad mode, the size of the particles is controlled with points.radius, which represents the half-length (radius) in the U direction of the sprite particle.
    - There are two new parameters, which are only available when rendering in quad mode:
        - points.aspect: the aspect ratio of the sprite. length_V = length_U / aspect. The default value is 1, i.e. square sprites.
        - points.rotation: the counter-clockwise rotation angle, in degrees. It can be negative and greater than 360. The default value is 0.
- **Motion blur optimization**: Top-level acceleration structures now take into account deformation in certain cases, leading to large speedups for scenes with fast moving, deforming characters. (trac#2109)
- **Reduced memory usage for wide ray trees**: Very "wide" ray trees caused by aggressive sampling settings, or large amounts of internal reflections and refractions no longer cause memory spikes due to a restructuring of memory allocations inside the renderer. The memory usage is now proportional to the *depth* of the tree, not the number of rays per camera ray. (trac#2093)
- **Blackman-Harris pixel filter**: The new blackman_harris_filter node produces similar results to a gaussian filter but has smoother tails which reduces aliasing. The recommended width is 3. (trac#2094)
- **Log detailed information when a signal is caught**: When a signal is caught, a warning/error message is displayed besides a signal description. (trac#1921)
    - Signals reported as WARNING: SIGINT and SIGTERM.
    - Signals reported as ERROR: SIGHUP, SIGSEGV, SIGBUS, SIGFPE, SIGILL, SIGABRT and SIGQUIT.
    - The rest of signals are processed by the default signal handler.
- **Progress messages displayed in green**: The render-completed percentages are now displayed in green to make them easier to find in large log files. (trac#2110)
- **Added skip_license_check option**: The new global option skip_license_check disables license checking, assuming that the license is not available, which will make the renderer produce watermarked images. This can be handy when some client machines don't need a license, for example if a modeler or animator is using a lighter's workstation temporarily. Note that this could also be done by tinkering with the environment variables (ARNOLD_LICENSE_HOST, ARNOLD_LICENSE_PORT), but, apart from being more difficult, this could lead to a delay while the license is being checked and denied. For convenience, we also added a -sl command line option in kick. (trac#2095)
- **Upgraded OIIO to 0.8.8**: This OpenImageIO release includes several important bug fixes as well as minor improvements, most notably: (trac#2044, #2045)
    - ImageCache: check for max open files when re-opening a closed file, not just when opening for the first time, or we may exceed max_open_files, and thus possibly hit fundamental OS limits.
    - ImageCache: fix bug wherein an invalidated and modified file would continue to flush in subsequent invalidations, even if the file was not modified again.
    - Improved PNG write speed by 4x.

API additions

- **New Ashikhmin-Shirley BRDF**: (trac#2011)

```
AtFloat AiAshikhminShirleyBRDF(const AtVector *L, const AtVector *V, const AtVector *N, const AtVector
*u, const AtVector *v, AtFloat nx, AtFloat ny);
AtColor AiAshikhminShirleyIntegrate(const AtVector *N, AtShaderGlobals *sg, const AtVector *u, const
AtVector *v, AtFloat nx, AtFloat ny);
```

- **MIS sampling methods for all BRDF's in the standard shader**: All of the specular BRDFs used by the standard shader have been extended to use MIS. This functionality is also available for shader writers to use via the following newly exported API functions: (trac#2013)

```
AiAshikhminShirleyMISBRDF()
AiAshikhminShirleyMISPDF()
AiAshikhminShirleyMISSample()
AiAshikhminShirleyMISCreateData()
AiCookTorranceMISBRDF()
AiCookTorranceMISPDF()
AiCookTorranceMISSample()
AiCookTorranceMISCreateData()
AiStretchedPhongMISBRDF()
AiStretchedPhongMISPDF()
AiStretchedPhongMISSample()
AiStretchedPhongMISCreateData()
AiWardDuerMISBRDF()
AiWardDuerMISPDF()
AiWardDuerMISSample()
AiWardDuerMISCreateData()
```

- **Derivative computation for skydome mapping functions**: The following versions of the AiMapping*() functions have been added that include derivative/tangent calculations: (trac#2028)

```
AiMappingMirroredBallderivs()
AiMappingAngularMapDerivs()
AiMappingLatLongDerivs()
AiMappingCubicMapDerivs()
```

Incompatible changes

- **BRDF and volume scattering normalization**: As mentioned above, several BRDF's have been corrected since they were Pi times brighter than intended. The volumetric scattering phase function has also been corrected since it was 4Pi times brighter than intended. For backwards-compatibility, we have added an option to "roll back" these corrections; if you need to preserve the look in existing scenes that were prepared with a previous version (e.g. to finish an existing project), you can disable the new physically_based option, which is ON by default. We *strongly*recommend staying in physically-based correctness moving forward with new shows as this will produce better balanced shading overall. (trac#2010, trac#2023)
- **Adaptive subdivision has changed**: In adaptive flatness mode, subdiv_pixel_error is now vastly different. *ALL* old settings in existing scenes are worthless now, and there's no way (nor any point) to make a compatibility mode. However, the polygon count for existing scenes that use the default settings (regular non-adaptive subdivision, i.e. pixel_error = 0) will not change.
- **Quad lights are single sided**: The sidedness parameter on quad lights introduced too many complexities for volumetrics and textured quad support, while offering very few advantages. It has been removed. Quad lights are now always single-sided. To achieve double-sided quad lights, simply duplicate the light and flip it. (trac#2027)
- **Renamed KICK_SEARCHPATH environment variable**: This relatively unknown kick-specific environment variable has been renamed to the more generic-sounding ARNOLD_PLUGIN_PATH, which we'll try to standardize on for external Arnold apps and plugins. Note that multiple paths can be specified, using the path separators native to each Operating System: (trac#2021)

```
Linux/OSX: /this/is/path/one:/this/is/path/two
Windows:   C:\this\is\path\one;D:\this\is\path\two
```

Bug fixes

#2161	NaNs in transmittance effects of the standard shader		
#2141	NaNs/Infs at low roughness settings in the standard shader		
#2121	AiPixelLength is numerically unstable	6 weeks	6 weeks
#2112	Drivers use additional ~4GB when image/bucket size exceeds ~2^24 pixels	6 weeks	6 weeks
#2111	Shortcut syntax for single elem arrays is not working for constant user data	6 weeks	6 weeks
#2107	bright specks at grazing angles with the standard shader using the stretched-phong brdf	7 weeks	6 weeks
#2105	kick should not render when given an unknown command line option	7 weeks	7 weeks
#2090	Specular and Reflection Fresnel are broken in standard shader when Ks or Kr are connected to a shader network	2 months	2 months
#2088	fix bump mapping noise/precision issues	2 months	8 weeks
#2085	NaNs in volume scattering when integrating very short segments	2 months	2 months
#2084	modify ray->mindist to prevent rays from getting stuck between two nearly coincident surfaces	2 months	2 months
#2079	SSS computation doesn't obey affect_diffuse parameter from lights	2 months	6 weeks
#2076	Bad intersection test on disk points	2 months	2 months
#2074	Setting light.filters array to NULL causes a crash	2 months	2 months
#2061	lights that don't contribute to volumetrics should not be attenuated by atmosphere	3 months	3 months
#2059	AiRadiance and AiIrradiance are not thread safe	3 months	3 months
#2055	node_update is not called in the nodes of a procedural when load_at_init is set to false	3 months	2 months
#2051	Receiving a signal after AiEnd() can cause strange behavior	3 months	6 weeks
#2050	custom plugin filters crash with negative AA samples	4 months	3 months
#2049	Crash when a hair material has diffuse cache on and a SSS cache flush is done after a render	4 months	2 months
#2047	Linking a shader component (such as .r) discards the static value for the other components	4 months	3 months
#2045	Modified textures are being permanently invalidated	4 months	3 months
#2042	Standard shader produces bright dots at grazing angles	4 months	3 months
#2034	Problem loading .obj files with Windows-style EOLs	4 months	3 months
#2033	kick -info crashes with parameter names > 32 chars	4 months	3 months
#2032	Plugins in '.' are not loaded when "-l" option an/or plugin searchpath is specified	4 months	3 months
#2019	Writing instance parameters to .ass is broken	4 months	3 months
#2010	Correct BRDFs for energy conservation	4 months	2 months
#1935	kick overwrites the same image file with every frame (with -turn <n> option)	6 months	2 months
#922	kick doesn't override the output file with -o	3 years	2 months
	or		