

4.0.2.0

Milestone 4.0.2

Enhancements

- **Multi-threaded displacement:** When two or more buckets require the result of a mesh displacement in order to continue they may now collaborate in the displacement computation to speed it up when the `enable_threaded_displacement` render option is active (*default ON*). (#2212)
- **Displacement of un-subdivided meshes:** It is no longer required to set `polymesh.subdiv_type` to other than none before displacing a polygon mesh. This is useful for dense meshes that already have enough polygons to capture the detail in the displacement map. (#2212)
- **Per-face displacement shaders:** `polymesh.disp_map` has been changed from an `AtNode` parameter to an array of `AtNode`'s. So now, not only does it accept a `NULL` or a single shader as input, but also an array of displacement shaders. Please note that when it contains more than one shader, `polymesh.disp_map` will share the indexing scheme provided by `polymesh.shidxs` and as a result must have the same number of elements as `polymesh.shader`. (#2669)
- **Faster abort during displacement and ray accel build:** The renderer will now more frequently poll for a render abort condition during both displacement and ray accel builds, which increases interactivity when working with big meshes and high subdiv settings. (#2709, #2710)
- **Faster scenes with many lights:** Scenes with thousands of lights cause quite long render times simply due to the time it takes to loop over the lights themselves. Some of those lights probably won't have any influence for a given shading point. We have introduced an acceleration structure to classify the lights by their volumes of influence and then loop over the important lights only. `sg->lights` and `sg->nlights` no longer return all the lights in the scene, and instead they now return the (hopefully small) set of lights which can have any influence in `sg->P`. The speedup will depend on both the number of lights and the number of overlapping influence volumes. We have seen speedups of 1.5x in empty scenes with 4 sparse lights all the way up to 20x in empty scenes with a few hundred sparse lights. (#2542, #2543)
- **New nlights color mode in utility shader:** The `nlights` mode shows the relative number of lights considered at the shading point, which is helpful when debugging scenes with many lights. This is visualized with a "heatmap" color gradient that goes from red (all lights in the scene are considered), passing through yellow and green, to blue (very few lights are considered) and finally black (no lights considered). (#2664)
- **Auto-cropped EXR output:** We now optionally embed `DataWindow` information when saving EXR files. This data window, or ROI in Nuke, stores a tight bounding box around non-empty pixels in the image, which can greatly accelerate post-processing. To enable this feature, use the `theautocrop` attribute in the `driver_exr` node (*default OFF*). (#2660)
- **Removed default node name:** When creating a new node, instead of this node receiving a default name based on a combination of timestamp, node type and count, etc... we now just give the node an empty name. This results in memory savings and speedups when loading or writing `.ass` files. (#2649)
- **Better info messages for licensing:** We have updated the warning messages for the licensing subsystem, making them more informative. This also includes messages coming from the `kick -licensecheck` command. (#2638, #2657)
 - No server running:
00:00:00 5MB WARNING | [rlm] could not connect to license server on 5053@localhost
 - Wrong license version:
00:00:00 5MB WARNING | [rlm] wrong license version, found 1 license for arnold 303
00:00:00 5MB WARNING | [rlm] please contact licensing@solidangle.com
 - Expired license:
00:00:00 5MB WARNING | [rlm] could not find any license for arnold 400, the license may be expired
00:00:00 5MB WARNING | [rlm] please contact licensing@solidangle.com
 - No license files:
00:00:00 5MB WARNING | [rlm] could not find any license files, please check the license server
 - Up-to-date license file:
00:00:00 5MB | [rlm] checkout of arnold (version 400) OK

API additions

- **AiArrayInterpolateRGB/RGBA():** In Arnold 4.0.0 we introduced several `AiArrayInterpolate*`() functions but we somehow forgot the RGB and RGBA versions. (#2666)

Incompatible changes

- **Transfer control about aborting on texture error to the calling code:** By setting a valid pointer for the return (success) value in `AiTextureAccess()`, the current error message is disabled and the render process is not aborted. It is left for the client code to handle the error as appropriate. (#2671)
- **Deprecated some 2D vector functions:** The old-style `AiV2Add`, `AiV2Sub` and `AiV2Scale` functions have been deprecated, the operators `+`, `-`, `*` should be used instead. The signature of the `AiV2Lerp` and `AiV2Clamp` functions has changed so that they return a value directly rather than through an argument; the old-style functions have been kept for now but marked as deprecated. (#2648)
- **disp_height no longer affects displaced mesh bounding box:** Both `disp_height` and `disp_padding` are separate and independent now. The former sets scale for displacement, while the latter extends the bounding box. If `disp_height` is set without setting `disp_padding`, it could result in clipping of the displaced mesh. (#2722)

Bug fixes

Ticket	Summary	Component	Owner	Priority	Version	Created
#2678	arnold 4 and auto_transparency_threshold	arnold	alan	blocker	3.3	3 weeks
#2656	always_linear metadata doesn't work when writing to .ass	arnold	angel	critical	3.3	5 weeks
#2659	Weird shadowing with transformed instances of procedurals	arnold	oscar	critical	3.3	5 weeks
#2662	Halo artifacts with 'importance_sampling' in 'volume_scattering'	arnold	oscar	critical	3.3	5 weeks
#2665	Bug when writing to .ass shaders with component links	arnold	angel	critical	3.3	4 weeks
#2479	Some geometry in procedurals with SSS is not rendered	arnold	oscar	major	3.3	4 months
#2641	Threading Issue when generating SSS pointclouds	arnold	oscar	major	3.3	6 weeks
#2646	intersection precision problems in cone primitive	arnold	oscar	major	3.3	6 weeks
#2652	Handle ABS(int)	arnold	thiago	major	3.3	5 weeks
#2655	Linking to a component in an array element not working	arnold	angel	major	3.3	5 weeks
#2661	missing Doxygen docs in AiSSSEvaluateIrradiance()	arnold	thiago	major	3.3	5 weeks
#2675	missing return value from AiRender() function in Python bindings	arnold	angel	major	3.3	3 weeks
#2676	missing return value from AiASS*() functions in Python bindings	arnold	oscar	major	3.3	3 weeks
#2677	Bad composition of inverse XForm matrices in procedural networks	arnold	oscar	major	3.3	3 weeks
#2680	Memory leak on array element linking	arnold	angel	major	3.3	3 weeks
#2681	Wrong shader override in instanced procedural networks	arnold	oscar	major	3.3	3 weeks
#2682	Wrong xform matrix recovering in instanced procedural networks	arnold	oscar	major	3.3	3 weeks
#2683	missing return value from AiNodeGetLink() function in Python bindings	arnold	angel	major	3.3	3 weeks
#2685	AiFormatTime() is not thread-safe	arnold	marcos	major	3.3	3 weeks
#2698	IPR memory leak in utility node in ambocc mode	arnold	angel	major	3.3	2 weeks
#2701	Nodes with empty names cause problems when saving to .ass	arnold	angel	major	3.3	10 days
#2704	AI_LOG_SSS missing from python bindings	arnold	angel	major	3.3	8 days
#2711	Misaligned temporary SIMD variables in MSVC cause memory crashes	arnold	xo	major	3.3	4 days
#2713	Gamma is not being applied to RGB(A) array elements	arnold	angel	major	3.3	3 days
#2714	log memory stamps shift by 1 above 10Gb	arnold	marcos	major	3.3	3 days
#2715	wireframe returns black in indirect	arnold	marcos	major	3.3	3 days
#2679	P color mode in utility shader is broken	arnold	oscar	minor	3.3	3 weeks