2.5.0

Release Date

March 20, 2019

Download links

- Solid Angle Downloads

This version uses the **Arnold 5.3.0.0** core, and it includes the beta version of Arnold GPU. See the Release Notes.

> **Important information about Arnold GPU (beta)**
> - Check the system requirements before using Arnold GPU. If you don't have a supported card or the required drivers, GPU rendering will not work.
> - Review the list of supported features and known limitations before you start using Arnold GPU.
> - If you have any technical problems, questions, or feedback on Arnold GPU, use the beta forum on Arnold Answers

FEATURES

- GPU rendering (BETA): You can now switch between CPU and GPU render devices interactively and expect visually similar results. NVIDIA® GPUs from Turing™ to Maxwell™ architectures are supported, and Arnold will take advantage of multiple GPUs, NVLink™ and NVIDIA® RTX™ hardware accelerated raytracing if available. Note that due to beta status of this feature, a number of features are missing, performance is not final, and use in production is not advised. Unsupported lights, shaders, parameters and AOVs are marked with an asterisk (*) in the label. We plan to gradually improve this in subsequent releases and will appreciate your feedback. For a complete description of requirements, features and caveats, see Getting Started with Arnold GPU.
- GPU rendering from the command line: The render device (CPU or GPU) can be controlled from the command line via the **-arnoldDevice [cpu|gpu]** flag.
- Pre-populate GPU cache: The OptiX™ disk cache, which is automatically generated during JIT compilation prior to GPU rendering, contains a store of previously compiled OptiX™ programs, so that subsequent renders start faster. The pre-population is a heavy process and it is recommended that the pre-population is triggered after installing a new Arnold version, updating to a new NVIDIA® driver, or changing the hardware configuration of GPUs on the system.
- Visible lights: Quad, disk, cylinder and point lights now have a `Camera` and a `Transmission` attribute, allowing these lights to become visible to camera and transmission rays. These attributes are left to 0 by default, thus not changing the default behaviour of the non-visible lights.
- Improved random-walk SSS: A new `randomwalk_v2` SSS mode has been added that scatters more accurately and deeply through highly-transparent/optically-thin objects, which produces SSS with more saturated colors around fine surface detail and heavily backlit regions of an object. Note that renders will be more costly and noisier than with the original method, since random walks will be on average longer and more random.
- MaterialX export: Bakes the shader and other look properties for one or more shapes (e.g. Arnold Procedural) to a .mtlx file along with the description of the shaders and shading graphs.
- Added uv_projection shader
- Added matrix_interpolate shader
- Anisotropy controls for coat in standard_surface shader
- Added include_graph operator
- Added denoise_albedo built-in AOV

ENHANCEMENTS

- Improved skydome sampling
- Improved adaptive sampling
- Smart opaque: built-in shaders now set the Opaque flag automatically based on whether or not the shader settings would require disabling the opaque flag on the object to render correctly.
- Improved bump and normal mapping
- Instantaneous shutter option: This new option in the motion blur render settings now keeps motion keys and only sets the camera ray times to be equal to the reference time. Use this with motion vectors instead of setting shutter intervals to zero.
- Support for negative transmission extra roughness in standard_surface shader
- Added time mode in ramp shaders: The ramp and ramp_rgb shaders now have an additional time mode that computes the input based on the current time and the camera's start and end shutter interval.

- Use implicit UVs added to the ramp_rgb shader
- Append frame number to the driver path automatically
- Added volume scattering to the shape visibility flags
- Expose autobump visibility
- Expose ignore list in the render settings
- Assign operator network to procedurals

FIXES

- Polygon selection does not work when the name has unicode characters in it
- Problem with special character in the uv_transform UV set name
- Wrong bump mapping with custom projection
- Bump mapping is too soft when UVs are tiled in the texture tag. Note that the look has changed following this fix and now it is (correctly) stronger when a custom UV projection or tiles are set in the texture tag.

API

> There are modifications in the C4DtoA API which require third-parties to recompile their C4DtoA extensions.