RenderMan 24.2

Release date: November 10, 2021

Welcome to RenderMan 24.2!

RenderMan 24.2 brings some new functionality to artists, as well as several bug fixes.

These release notes build on the the release notes for 24.0 and 24.1, so please see those notes first.

For each of the bridge product integrations, please see the Release Notes within each of their respective sections.

Dive right in using the table of contents below!

What's New

RIS and XPU

- hdPrman: Consume MaterialX through hdPrman. Pattern networks are compiled to OSL. BxDFs are transcoded to PxrSurface.

RIS

- Volumes: PxrPathTracer now supports "Aggregate Volumes".   Aggregate Volumes is a new technique allowing you to render overlapping volumes more efficiently and lower the time to first pixel. It also allows PxrSurface to use a heterogeneous volume in its interior integrator. See details here: Aggregate Volumes and in the DCC specific documentation sections.
- Volumes: PxrVolume has a new velocityPrimVar parameter. This is more efficient than reading the velocity via a PxrPrimvar or PxrVariable pattern
- Volumes: Adding additional new parameters to impl_openvdb: densityScale, densityRolloff, filterWidth, velocityScale
- Lights: Light shaders now have the ability to opt into multiscattering approximation calculations for the new aggregate volume feature
- Integrators: Added support for Path Guiding. It can be activated in PxrPathTracer and the PxrUnified integrator via the new risGuiding integer parameter. Path Guiding is currently incompatible with integrator settings with more than one indirect sample, and with PxrPathTracer's 'manual' bxdf sampling mode. Requires light learning to be enabled
- OSL raytype() is now implemented. The supported ray types are "camera", "shadow", "reflection", "refraction", "diffuse", "glossy", "subsurface", and "displacement"
- Statistics: More stats are available in the JSON report.

XPU

- Rendering: Crop Window support is now available on XPU.
- Rendering: Add support for computing curvature on polymesh/subdiv/curve/point geometry
- Statistics: More stats from XPU's lighting subsystem are now available in the JSON report.

Changes in Stylized Looks

PxrStylizedLines:

- New feature: Dilate + Sort for signals going off geo
- Master Line Thickness & Line Thickness Scaling
- Daisy Chain Comp Modes: Over, Plus, Multiply
- Overwrite Data AOV option

PxrStylizedHatching:

- New AOV: NPRNtriplanar
- Support custom surface normal coordinates for blended triplanar (ex: __Nref)
- Hatching Mask (set in PxrStylizedControl) to mask Hatching ON/OFF per object/shader

PxrStylizedControl:

- Support custom surface normal coordinates for blended triplanar (ex: __Nref)

- Cam Depth Scale at object/shader level for all features in Lines & Hatching that rely on camera distance
- Hatching Mask control at object/shader level
- Triplanar Scale: XYZ scales for triplanar projections at object/shader level

Other Bug Fixes

RIS and XPU

- Curves: Fixed a crash when a very large amount of curves are present in a scene
- Statistics: Better robustness during startup of the stats system
- Textures: Fixed a bug where txmake would fail on images whose largest mip layer is 4Gb or larger

RIS

- Lighting: Fix equiangular clipping in presence of infinite lights
- Lighting: Several fixes related to light selection
- Lighting: Fixed a live render bug where the ramp would render backwards initially if begin > end
- Integrators: A crash in the Path Guiding feature of PxrUnified has been fixed
- BxDFs: Added labels for glass parameters to PxrLayerSurface to match PxrSurface: Shading Tangent, Manifold Walk IOR, and Anisotropy
- BxDFs: Fixed presets in LamaConductor
- OSL: Silenced a debug message that was slowing down regex_search()
- OSL: Improved OSL metadata tokenization to correctly handle <primvar:varname>
- OSL: Fix out-of-bounds vector access when a spline is provided with only a single control point
- Alembic: Fix for incorrect renders that could show up with polygonal input data with normals
- Geometry: Fixed a memory leak for the case of general polygons which also occurred for pretessellated subdivision surfaces
- Volumes: indirect visibility of volumes now correctly takes into account scattering events that occur within the volume. Previously indirect visibility was only checked at volume interfaces, which meant that surfaces within volumes that also enclosed the camera could wrongly reflect indirect invisible volumes
- Checkpointing: When using the new variance metric, fixed a problem that prevented resuming from a checkpoint
- Rendering: Fixed an issue that if you edited the shutter value, the framing and sampling unexpectedly changed
- Rendering: Add support for FloatVectorAttribute when setting metadata for EXR output
- macOS: Don't crash on Macs with AVX512 instruction set support

XPU

- Shading: Fixed a known look difference between CPU and GPU portions of XPU
- BxDFs: Fixed the potential for NaNs when using the fuzz lobe of PxrSurface
- Textures: Fixed issue with primitive variable substitutions into texture filenames
- Textures: Fixed artifacts that arise in XPUGPU mode when texture cache size is small
- Textures: XPU now shares the same minimum ptex and non-ptex texture cache sizes RIS. In the case the user sets a minimum texture cache that is lower than our internal minimum we use it and do not display any warning for both ptex and non-ptex texture caches
- OSL: Robustly handle the case if memory allocation fails when setting up the color system for OSL
- Patterns: Fixed a crash that would occur if '-' was used in the name of a pattern
- Performance: Fixed cases where CPU + GPU would sometimes render slower than the GPU alone
- Geometry: Fixed a problem that could result in a crash when editing materials on curves
- Lighting: Correctly initialize default values for light parameters in XPU

"it"

- "it": Fixes for supporting the status HUD when viewing AOVs

hdPrman - the RenderMan Hydra Renderer Delegate

- Fixes for a case where PxrDisplacement would not work when being passed to XPU
- Make volumes respond to underlying changes in the field prim
- Fixed a bug that could sometimes hang hdPrman when processing materials with loops in the node graph
- Updated calculation of primvar calculations in curves to match RenderMan
- Fixed a camera framing issue
- Support array inputs within hdPrman's material parser
- New warning when a renderer delegate cannot be created
- Fixed issues with the fallback shader
- Add support for UsdPreviewSurface's opacityThreshold param in HdPrman
- Metallic materials should have an F0 equal to their base color in UsdPreviewSurface
- Support .args and .oso files embedded in .usdz files
- Updated UsdPreviewSurface clearcoat calculations to better match expected inputs
- Support UsdTransform2d
- Fix HdPrman support for USD Preview Material texture coordinate conventions when using a RenderMan tex format file. Inserts a UsdTransform2d if necessary to invert the T axis
- Optimize framebuffer updates for faster frame rate
- Cleaned up warnings from the stats system that would appear during interactive rendering
- Adding support for missing light parameters to hdprman (msApprox, msApproxBleed, msApproxContribution, emissionFocusNormalize)
- Updated our renderer definition with settings to allow for better integration with Solaris features
- Added support for dataWindowNDC and aspectRatioConformPolicy
- Adjust HdPrman's fallback volume shader to match Storm's. Visually, this fallback approximates something like smoke: low albedo, isotropic (i.e. scattering has no directional bias), single-scatter
- Fix HdPrman's fallback maxSamples to match the RenderMan fallback (64)
- Adds support for UsdUvTexture's "sourceColorSpace" input and fixes UsdPreviewSurface's normal mapping

- We want the ability to pass asset paths that refer to things other than .tex files on disk to prman for the lightColorMap property on lights
- Support additional clipping planes
- Add support for UsdPreviewSurface's displacement

Miscellaneous Changes

RIS

- Integration: A few enhancements in light selection that will provide improvements in convergence time.
- Integration: Significant speedup for RIS PxrOcclusion integrator for ambient occlusion (with "useAlbedo" off) when objects have complex pattern networks.
- OSL: simplified semantics for calling debug(). Simplify to debug(aov, "writemode", "always") and allow conditionals around the call to replace the flagval itself. Default behavior of debug(aov) is equivalent to debug(aov, "writemode", "camerahit")
- OSL: PxrNormalMap willi now check for primvars Tn and Bn for the tangent and bitangent before using "builtin"
- Lights: Startup time improvements when many unique mesh lights are present
- Lights: Performance improvements during light selection
- API: Support for modifying data within nested mutable shading contexts
- API: If you have a custom integrator, you may need to recompile it due to an API change that was required to support Volume Aggregates.

XPU

- Significant speedups for XPU PxrVisualizer (with styles "shaded", "normals", or "st") and XPU PxrOcclusion integrators (for ambient occlusion with "useAlbedo" off) when objects have complex pattern networks.
- Error Message cleanup
- XCPT style error handling added to several subsystems