

RenderMan 22.5

Welcome to RenderMan 22.5!

Welcome to RenderMan 22.5. This release introduces improvements to the previous RenderMan.

Please dive right into the release notes below for more detailed information on the latest version of your favorite renderer!



Users of custom Integrators will need to recompile their plugins.

What's New

- **Render masks/mattes in indirect effects** – Using [PxrSurface](#), you can now render mattes in a reflection or through a refraction/transmission including points/position in world space
- **New Live Edits** – We've enabled live displacement and volume edits in IPR (note this only applies to tuning displacement, not creating or deleting a displacement)
- **Enhance** – PxrCamera now has an option to zoom into a render without affecting any view-dependent features like dicing or mipmapping, allowing you to inspect your scenes easily without changing your camera

Miscellaneous Changes

- Batched [OSL](#) processing is now *on* by default
- OSL messages are not emitted more than once
- The Alembic procedural now correctly supports velocity values (when exported with the Alembic), in order to allow the rendering of motion blur correctly on deforming geometry with varying topology
- [PxrProjector](#) has a new "planar" mode
- There is a new parameter under the [PxrSurface](#) Globals called "User Color" that the user can use to feed data into the user lobe 4 (U4) of PxrSurface. Then by using LPEs we can output that data as masks/mattes using LPE
- Volumes
 - Setting velocityMultiplier to 0 will now disable any expensive velocity pre-processing in PxrVolume
 - The renderer now has improved support for interactive editing of volume shading parameters. In particular, editing of parameters which do not affect transmittance remain fast, while editing of density parameters which affect transmittance will correctly trigger a (potentially expensive) rebuild of the volume acceleration structures
- The renderer now more gracefully handles attempts to load shading plugins of the wrong type (i.e. trying to load PxrDiffuse as anything other than a Bxdf)
- "it"
 - The HeatMap function in "It" now remembers the last used font
 - There is a new histogram viewing option in the Heat Map command
- Added a new "enhance" parameter to [PxrCamera](#) for magnifying the region around a pixel without moving the camera or affecting dicing and MIP map levels. This feature is intended as an aid for trouble-shooting small areas of the image where you may see noise or changing topology during animation
- PxrTexture, PxrPtexture, PxrMultiTexture, PxrFractal, PxrVoronoi and PxrWorley have a new "Adjust Output" section to scale, offset RGB and alpha/float outputs. PxrTexture, PxrPtexture, PxrMultiTexture also have a saturation control
- To conform with the original d_openexr display driver, d_deepexr now accepts two new flags: "exrcompression" (same as "compression") and "exrpixeltype" (same as "type"). "compression" and "type" are still valid flags and referenced in d_deepexr.args for backward compatibility
- Added support for colored shadow AOVs using PxrShadowDisplayFilter and PxrShadowFilter with added support to PxrImagePlaneFilter
- Exrinfo tool now reports keycode, timecode, string vector attributes, and supports multi-part images (multi-part images are not output by RenderMan in this release)
- Better scene translation pruning of hidden geometry
- Quantized png, targa, and tiff display drivers now convert linear to sRGB
- Improved derivatives for RiBlobs under motion
- Variables of type "mpoint" (used by RiBlobby) are once more functional again
- LPE custom handle not operator (e.g. [^'foo']) now accepts unnamed LPE groups
- When using the Chaikin creasemethod on subdivision surfaces, we no longer average semi-sharp creases with a neighboring crease of infinite strength. This fixes the potential for holes in the surface when a semisharp crease is near a boundary or an infinite crease. This change also increases compatibility with the OpenSubdiv behavior
- The Option "trace" "maxdepth" is now ignored. It was a left-over from older versions of RenderMan, and its default value was 25 so it rarely did anything. In modern RenderMan usage, the Integrator "maxPathLength" parameter can be used instead as a similar control. Also note that the "trace" "maxdiffusedepth" and "trace" "maxspeculardepth" Attributes are still valid, and used by most Integrators (the big exception is the PxrUnified integrator that has a unified trace depth.)
- The fallback default material PxrDiffuse now allows non-networked presence shading

Fixes

- Fixed a regression introduced in 22.4 where deformation motion blur describing pronounced rotation was producing some noticeable darkening in surfaces with reverse orientation
- Denoise
 - Various improvements to Denoise stability and filtering results
 - A bug in the Denoiser where the quality of filtering of a lighting image was affected by the name of the channels in said image has been fixed
 - The Denoiser will now attempt to correct for negative values in the albedo channels and will issue a warning if it finds any
- PxrPathTracer now correctly computes the alpha channel result for holdout objects inside volumes
- Curves
 - Fixed an issue in curve rendering that could lead to crashes with large numbers of curves
 - Fixed a bug in curves geometry resulting in wrong derivatives during evaluation of presence. The bug manifested as presence not responding properly to connected patterns
- Fixed a potential crash with deep checkpointing enabled when a render aborts early
- Removed Catmull-Rom and Mitchell filter options from PxrBakeTexture, PxrNormalMap, PxrLayeredTexture, PxrMultiTexture and PxrProjectionTexture. These filter were never supported for non-ptex textures
- Improve shadows LPE results to be more stable with different light sampling
- The renderer now correctly supports interactive editing of presence and opacity, whether it is cached or not
- Oversampling in pixels on bucket edges when adaptively sampling has been fixed. This can provide up to a 15% reduction in number of rays and runtime for multithreaded non-incremental rendering, and a small reduction in the number of rays and runtime for multithreaded incremental rendering. (No change for single-threaded rendering or multithreaded incremental rendering with checkpointing)
- PxrDisney, this legacy material has received some fixes
 - Bucket artifacts when PxrDisney "specular" input references a pattern input that switches between zero and non-zero
 - Missing specular Fresnel reflection when the "specular" input is simply 0
- Fixed a bug that prevented SIMD OSL from running on some processors that only support AVX
- Fix an issue with importing "black_oxidized_steel" and "hematite" presets from the preset browser
- Fixed a bug that could lead to blocky artifacts when viewing a light with "visibleInRefractionPath" set inside a surface with interior volumetric extinction
- Fixed bug causing potential crash when an OSL shader can not be opened
- Visible noise along volume envelopes (in pixels partially covered by a volume) has been fixed. This noise was especially visible with pixelfiltermode "importance"
- Fixed a precision issue in PxrRandomTextureManifold and PxrRamp that was preventing randomization with the object ID
- Fixes to LPE parsing and the carat token
 - Fix S75000 parsing warning for NOT operator i.e. `[ ^ ]`
 - Fix LPE user lobe inclusion into NOT operator i.e. `[ ^ ]`
- Fixed failed scoped coordinate system lookup from procedurals
- Issues with wrong LPEs when rendering volumes that overlap surfaces in PxrPathTracer with sampleMode = "manual" have been fixed
- Fix zero trace depth with high trace:maxdiffusedepth/maxspeculardepth values
- Fixed a bug where using multiple surfaces with colored opacities would yield wrong composited results
- Fixed a bug where frame user options were sometimes ignored
- A bug that would cause the renderer to crash if the smoothnormals attribute was turned on and the geometry was undergoing deformation motion blur has been fixed
- Fixed a precision issue which could cause volume rendering to hang indefinitely

API Changes

- Added projection DOF query from RixRenderState::GetOption as Ri:DepthOfField
- k_incidentLobeSampled and k_launchShadingCtxId lookups for RixShadingContext::GetBuiltinVar are correctly implemented in cached presence, opacity, displacement, and meshlight contexts. Note however that the returned values will always indicate invalid incident lobe or invalid launch shading context
- Change to RixLPEInline.h requires rebuild of any integrator plug-ins
- The RtFloat2 and RtFloat3 variants of RixShadingContext::GetPrimVar() now correctly allocate and initialize storage with the requested fill value in all cases when the requested primvar is not found

Interactive/Live Rendering Limitations

- Crop window edits are restricted to fall inside the original crop window
- Displacement cannot be added or removed during IPR currently and expected to update. Only changes to displacement are supported

RenderMan Pro Server Limitations

- PxrUnified integrator does not yet support all the standard rendering features
- We do not read point data from OpenVDB files
- PxrSurface back diffuse color is not output to the albedo color AOV
- Analytical lights placed inside volumes may yield artifacts when made visible to the camera. As a work around, the light camera visibility should be turned off, and a geometry with a similar shape should be used (visible to camera, invisible to transmission and indirect rays), with the proper emissive bxdif
- Using the `.` character in the handle for an OSL shader could cause unpredictable results during re-rendering
- Per-Instance baking is not supported, only the reference instance
- 3d baking: no direct bake-to-ptex support
- PxrBakePointCloud cannot directly render ptex
- Sample/Display filter plug-ins do not have access to lighting services for light dependent effects, e.g. lens flare
- Adding a new mesh light on existing geometry during IPR results in double geometry
- Motion blurred geometry does not motion blur normals when deformed

- When attempting to access an array primvar, you must first check the size of the array primvar and allocate the appropriate space. Not doing so may lead to a crash
- Points and curves cannot be used as geometric (mesh) lights
- Deformation motion blurred volumes don't currently work with densityFloatPrimVar or densityColorPrimVar. You will need to use a PxrPrimVar node connected to densityFloat and densityColor instead



The Centos KDE style "Oxygen" installs a version of Qt and sets the user's environment variable QT_PLUGIN_PATH forcing "it" and LocalQueue to attempt to load an incompatible Qt library. Either avoid installing the Oxygen theme or unset QT_PLUGIN_PATH before running "it" or LocalQueue. Other KDE styles may also install this theme.