

RenderMan 21.5

Welcome to RenderMan 21.5!

This release introduces the following improvements, fixes, and miscellaneous changes.



RenderMan 21.5 ships with many changes and improvements to the API. You will need to recompile plugins to be compatible with these changes. You can find the Developer Documentation and Doxygenated Files [here](#). You must be logged in to the forum.

New Features

- Path traced (exponential and non-exponential) subsurface scattering has been introduced into [PxrSurface](#) as an option. This technique uses the same controls while providing an accurate result with a single set of controls instead of combining other techniques with Single Scatter.
- It is now possible to run the [Denoise tool](#) on EXRs exported after a compositing process if the channel names and data are preserved.
- Added color remapping features to [PxrHairColor](#) as a new Artistic mode. We default to the previous Physical mode for backward compatibility.



Scene courtesy of [Krystal Sae Eua](#) for *The Mill*

Miscellaneous Changes

- Substantial improvements made to Denoiser performance on the CPU, speed improvements of 2x to 4x are possible.
- Default settings for PxrMarschnerHair have changed.
- Improved sampling results when using mesh lights.
- Improved default Shadow LPE with improved accuracy and now works with light filters. Previous behavior is preserved by using the 'shadows214;' LPE prefix, i.e. lpe:shadows214;CDL

- PxrSurface now has more accurate behavior when thin shadows are enabled on glass-like surfaces; the improved behavior more accurately models the energy loss due to multiple reflection/refraction bounces. *This change may cause some darkening of shadows compared to earlier versions.*
- The GGX specular option for PxrSurface should see reduced noise as well as a slight change at grazing angles on anisotropic materials.
- Dramatic improvement in PxrMarschnerHair rendering speed, this might result in a slight brightness change when specularcompensation was active.
- Updated the light selection algorithm. We expect, especially in situations with very many lights, that the new algorithm will demonstrate faster iterations times and/or improved convergence per iteration.
- Max continuation length is now at least 256. Unless you have a significant number of overlapping volumes, you should not need to worry about this parameter.
- Displace shaders now receive Synchronize messages.
- Minor improvement to OSL texture access.
- Multiscatter now supports multiple samples, just like single scattering does. Also, equiangularWeight is only controlling single scatter now. Multiscatter ignores this.
- We've added velocityMultiplier to [PxrVolume](#), it could be helpful when converting velocity per frame to per second, etc.
- The max resident memory for [PxrVCM](#) has been reduced by around 30%.
- The time to first pixel for scenes with volumes undergoing deformation motion blur has been greatly reduced, especially when the motion blur is large and many threads are used.
- Fixed a bug where Denoise would not take advantage of all possible hardware capability on Linux machines in either CPU or GPU mode.

Bug Fixes

- Fixed a bug in AccumOpacity with Volumes.
- Fixes in the fresnel computation on PxrMarschnerHair provide stricter energy conservation across lobes.
- Fixed an issue where energy compensation was not applied correctly to non-Clearcoat lobes in PxrSurface.
- Fixed a crash when using -recover 1 with very large display resolution and/or many displays.
- Fixed bug causing incorrect matches in LPE results on aggregate volumes.
- Ray differentials computed for objects in or behind volumes are now correctly computed, eliminating discontinuities in texture filtering and improving the performance of same.
- Support width attribute for points in AlembicProcPrim.
- Fixed occasional indexing error in surface normals for SpSSDiffusion (ie. PxrSkin and PxrSubsurface bxdfs) when some albedos are black.
- Display "int matte" has been fixed to once again work as expected. Integrator developers should call RixDisplayServices::FlagAsMatte() on samples coming from matte objects to indicate to the renderer when to ignore splatted values. Plugin developers using RixDisplayServices will need to recompile their plugins.
- A bug that caused the shadow collector LPE to vary in intensity with a different number of fixedSamples to a light has been addressed.
- Fixed a random number biasing bug in hair widening. *This will change noise patterns on widened hair due to better sampling.*
- Fixed memory allocation bug in deep EXR output drive that would cause a crash when writing out deep data using crop windows.
- Fixed inaccurate tangents and normals computed along a semi-sharp subdivision surface crease edge which could occur at very coarse shading rates.
- Fixed possible moire patterns with the PxrRectLight

Known Limitations

RenderMan Pro Server

- Full LPE support is only available with the PxrPathTracer and PxrVCM integrators.
- Light Filters in the PxrVCM integrator do not affect or shape photon emission.
- The Cone Angle parameter for lights is incorrect, it does not match the input angle, this will be fixed in a future version.
- The PxrAovLight does not work properly with PxrUPBP.
- Analytical lights placed inside volumes may yield artifacts when made visible to the camera. As a workaround, the light camera visibility should be turned off, and a geometry with a similar shape should be used (visible to the camera, invisible to transmission and indirect rays), with the proper emissive bxdf.
- Using the '.' character in the handle for an OSL shader could cause unpredictable results during re-rendering.
- Instances are not supported for baking.
- 3d baking: no direct bake-to-ptex support.
- PxrBakePointCloud cannot directly render ptex.
- No RixPTC/pointcloud API (so PxrBakePointCloud cannot read ptc files).
- Sample/Display filter plug-ins do not have access to lighting services for light dependent effects, e.g. lens flare.
- Adding new mesh light on existing geometry during IPR results in double geometry.
- Camera visibility changes are not respected during Live Rendering.
- For PxrUPBP, If the light source is inside a volume, that volume needs to be defined as Volume "box"
- For PxrUPBP, To get a volume caustic, the object casting the caustic needs to have higher intersectpriority than the volume.
- For PxrUPBP, Overlapping heterogeneous volumes are not working yet. (However, overlapping homogeneous volumes do work.) This will be resolved in the future.
- PxrPortalLight may yield artifacts when generating photons for PxrVCM / PxrUPBP. This will be fixed in a future release.
- 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 have mesh lights attached to them.
- 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.