# RenderMan 21.3

## Welcome to RenderMan 21.3!

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

### New Features

- PxrSurface's new Diffuse Exponent (when roughness is 0.0) is a power exponent that controls the diffuse falloff.
- PxrMarschnerHair now supports Glow and improved controls with the addition of new parameters.
- PxrSurface's interior scattering introduces a Multiscatter parameter. This is required to be *on* for use with the PxrVCM and PxrUPBP integrators.
- LPE system now handles complex sets of LPEs more efficiently.
- Per lobe LPE has been added to PxrMarschnerHair, these will output results from: HairSpecularR, HairSpecularTT, HairSpecularTRT, HairSpecularGLINTS.
- New built-in primvars "curvature_u" and "curvature_v" -- the principal surface curvatures in the u and v directions. (As before, the "curvature" primvar is the mean surface curvature, ie. the average of curvature_u and curvature_v.)

### Miscellaneous Changes

- Scenes with frequent use of EnableLightFilter may see a reduction in the amount of memory used.
- Support for changing light filter topology during IPR.
- Improved light sampling in volumes.
- Less noise is visible when using presence values less than 1.
- Time to first pixel in scenes with millions of lights has been greatly reduced.
- Added support for writing holdout shadows to the beauty alpha channel
- Some error messages for RiCurves now contain more info.
- We now allow layering of subsurface transmit gain in PxrLayer and PxrLayerMixer.
- PxrOcclusion useAlbedo now requires Bxdfs to return a valid albedo user lobe otherwise black is assumed. This may result in a change the behavior for custom Bxdfs, visible lights, and hair geometry.
- Denoiser layer groups can now override the main images variance channels using custom settings.
- Improved selection of lights when rendering anisotropic volumes.

### Bug Fixes

- Fixed a bug that could cause sporadic crashes early in a multi-threaded render.
- Fixed a bug with UNC Windows file path parameters for "it"
- Fixed a bug in PxrMarschner that may cause crashes.
- Spaces in file paths can be passed to "it"
- Fixed a possible stack overflow when rendering curves.
- Fixed missing results from baking one-sided geometry.
- Fixed an error message indicating bad nesting of structs when the problem was actually just incompatible struct connections in shader graphs.
- The orientation of Nn on a sphere now matches Ngn -- there was an inconsistency for transforms with negative determinant (both for Orientation "rh" and "lh") when Option "quadric" "orientspheretotransform" is 1.
- Fixed incorrect output space for bake3d normals.
- Fixed incorrect transmission term for motion blurred volumes.
- Python RIF's now run in python3.
- Fixed a bug that could cause OSL shaders to get dropped from objects with long "identifier" attributes.
- Fixed crash in watertight dicing that could occur with some maxgridsize settings.
- Fixed single-scatter nested volumes writing direct lighting and emission results to LPE AOVs. The results are now correctly classified and written to the correct AOV.
- Geometric light patches no longer potentially get disabled when they happen to have a high enough tessellation rate.
- Matte objects now work correctly in Z-depth AOVs. Toggle off the "respects matte" property of the Z AOV if you wish to emulate the previous behavior.
- Fixed potential for crashes during direct-linked renderer shutdown at the completion of a render that used OSL.
- Fixed potential end-of-render crash when using OSL shading nodes.
- Fixed incorrect derivatives in PxrBumpManifold2D.
- Fixed out-of-sync PxrLayerSurface.args from PxrSurface.args where diffuse tranmission didn't work with PxrLayerSurface.args.
- Added missing subsurface transmit gain in PxrLayerSurface.args and PxrLayer so we can now layer subsurface transmit gain as well.
- Fixed a bug that caused textures made with pattern "single" to return black when used with bilinear and bslpine filters.
- Fixed a crash that would occur when using holdouts in scenes with no light sources.
- Fixed a bug accessing the wrong MIP level for non-square textures resulting in spurious "T10004 {WARNING} Can't find res for xxxx" warnings and random texture read values.
- Fixed correlation artifacts in PxrDirt

# Known Limitations

## RenderMan Pro Server

- The PxrAovLight does not work properly with PxrUPBP.
- 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 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.