# RenderMan 21.4

Welcome to RenderMan 21.4!

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

New Features

- A new pattern for the easy creation of realistic hair has been added, PxrHairColor
- A new manifold for the control and creation of random texture assignments has been added, PxrRandomTextureManifold
- PxrManifold2D has Invert S. Defaults to 0 or off.
- Lights (PxrRectLight, PxrDomeLight, and PxrPortalLight) have added parameters to control texture map saturation and gamma.
- PxrIntMultLightFilter has new parameter Saturation.
- PxrVCM now supports an additional parameter: 'specularCurveFilter' (float, default value: 1.0f). This allows the user to de-activate specular roughness filtering. Specular roughness filtering usually yields renders with less noise, but this may result in inaccurate caustics. Set this value to 0.0f when you want to render ground-truth caustics.

Filter default: 1.0
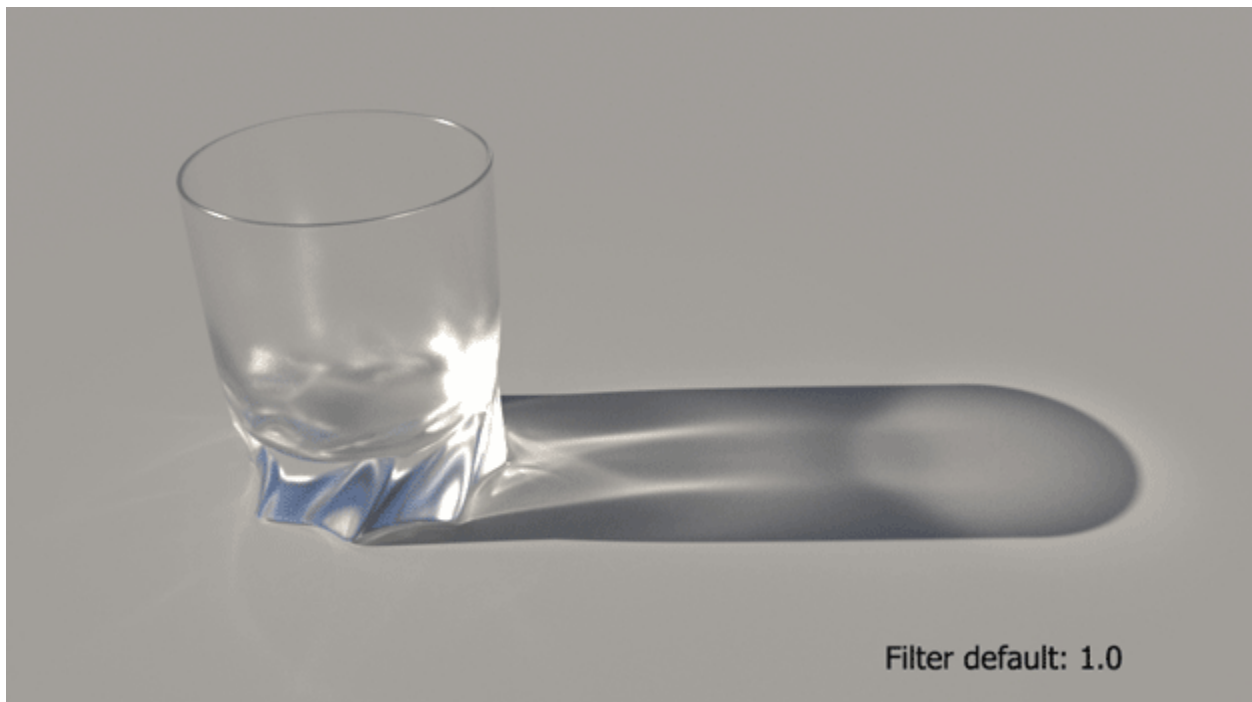
Scene courtesy of Julien Duval

- Single scatter volumes now work with deep images.
- PxrNormalMap has a new "orientation" parameter to either select between standard orientations (OpenGL or DirectX) or manual settings (Custom).

Miscellaneous Changes

- Volume rendering improvements can increase speeds on scenes up to 15% in volumes with large areas of zero density and rendered without motion blur.
- DeepEXR's storage changed from "tiled" to "scanline". Note that this may dramatically increase memory requirements to cache the frames as they are being rendered. We recommend Horizontal or Zigzag-X bucket order when rendering to help ameliorate the cost. Nuke requires scanline storage for deepEXR files.
- PxrSurface's Specular and Clearcoat energy compensation now applies to the physical fresnel mode.
- A change made to how photons and refractions are handled will change the brightness of scenes rendered with PxrVCM compared to previous versions.
- Volumes in deep images rendered using the PxrPathTracer integrator have been improved. In particular, single-scatter volumes now work with deep images and developers of custom integrators can now register splats at distinct multiple depths along a single ray for deep images using the new RixDisplayServices::AddSample() method. *Note that this may require developers to recompile their plugins*, but no source changes are needed.
- The RixDisplayChannel::ChannelType::k_IdChannel enum has been renamed to k_IntegerChannel. The old name remains available as an alias but will be removed in the next major release.

- The "float rawId" builtin AOV is now deprecated. Use the equivalent "int id" AOV instead.
- Made optimizations to the core raytracing speed.
- An incremental render that successfully uses recover mode to pick up from a previous render now does at least 'minextrasamples' new samples (Hider "raytrace" "int minextrasamples").
- Attribute "identifier" "id" of -1 (old behavior was < 0) signals the master ID wins, not the instance (i.e. no instance override desired).
- Speed improvements to txmake when dealing with large input images have been made. Large textures should use the OpenEXR format option.
- Reduced noise and improved convergence in the PxrOcclusion integrator.
- Texture cache memory statistics are now more accurate, particularly the texture entry in the overall memory summary.
- Custom integrator plugins are now responsible for splatting builtins; PxrGeoAovs.h has been added to help with this.
- Improved performance of OSL access to primvars and inputs from C++ shaders.

Bug Fixes

- Fixed bug in PxrLayerMixer for physical specular and subsurfaceTransmitGain of the layer 2, 3, 4.
- Fixed PxrSurface's subsurface albedo where there were artifacts when the subsurface gain was connecting to a pattern that has zero value.
- Fixed an issue with a possible race condition in the bake hider when cracking procedurals.
- PxrSurface's energy compensation now works for the physical specular mode as well as the artistic mode.
- PxrVCM integrator now correctly produces deep images of volumes
- Fixed a rare multi-threaded crash encountered when using many LPE's.
- Fixed a potential crash when baking from delayed read archives.
- Fixed bug causing potential for NaNs when using PxrMeshLight.
- Fixed an issue where there might be artifacts caused by light selection.
- Fixed bug in attempts to access primvars that don't exist via OSL'sgetattribute() call.
- Fixed a bug where the motionFore and motionBack AOV's for cross-frame denoising could have incorrect values on instanced geometry.
- Resolved an issue where refraction rays would be traced using the subsurface scattering interior integrator, resulting in missing refraction and/or bucketing artifacts in the case where there is subsurface scattering without volumes.
- Fixed a bug in PxrTee that was preventing it from working correctly with float inputs when using RfM.
- Correct P values now returned for single scatter volumes.
- Fixed potential crash when a Subdivision Surface shading network has multiple PxrBakeTexture outputs.
- The normal on brick mapprimswas sometimes in the wrong space; this has been fixed.
- Fix to OpenEXR image outputs whereby only the first increment would be output in some scene setups.
- PxrSeExpr now recognizes the "objectId" keyword.
- Fixed a memory corruption issue when rendering very large framebuffers.
- Fixed crasher in PxrSurface multi-lobe sss when the subsurface scattering color is black and has post-tint.
- A bug that caused theshadowcollectorLPE to have a different brightness when a different number of light samples to the integrator and a different number of the fixed samples on the light were being used.
- Fix holdout shadow look difference between alpha and separate AOV outputs
- Checkpointing and recovering from an EXR file that had the "asrgba" option turned on should now work as expected.
- Fixed a potential crash with nested dielectrics.
- Fixed a bug that could cause lost low-priority events when using intersection priority along with instancing.
- Fixed a bug where deforming geometry with duplicate motion samples could cause portions of unrelated geometry to disappear.
- Fixed a crash for the PxrVCM or PxrUPBP integrator with higher image resolution than 2K.
- Fixed a bug causing the actual size of the texture cache to be slightly bigger than intended.
- Fixed a bug causing the Lighting stats timer for tracing in Sample Evaluation to not work.
- Fixed bug causing overly sharp texture access when using Bspline/Bilinear filtering in both C++ and OSL shaders.
- Fixed bug causing incorrect texture filter widths when using builtin u and v as texture coordinates from OSL.

Known Limitations

RenderMan Pro Server

- Full LPE support is only available with the PxrPathTracer and PxrVCM integrators.
- The Cone Angle parameter for lights is incorrect, it does not match the input angle, this will be fixed in a future version.
- Display "int matte" does not work as expected, this will be fixed in a future release.
- 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.