RenderMan 22.2

Welcome to RenderMan 22.2!

Welcome to RenderMan 22.2. 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!

New Features in 22.2

A New Analytic Cylinder Light – A new PxrCylinderLight is included for rendering things like fluorescent tubes and lightsabers.

Added support for a Deep Matte File – Using the PxrCamera, you can specify a previously rendered EXR with deep or shallow data to then control the propagation of rays in the scene either by depth (deep) or as a render mask (shallow EXR with Alpha)

Miscellaneous Changes

- Added procedural paths to all render paths for third parties
- Geometric meshlights can now be created by attaching the special parameter "int __islight" to a Bxdf that implements the EmitLocal method, and setting the value of the parameter to be 1
- Improved handling of NaNs in albedo channels when denoising
- Terminal stats will once again show the full channel list as the display mode for each render output
- PxrWireframe now has a background color parameter
- Added a new flag to the denoiser to help select the GPU index to use. Calling the denoiser with `denoise --list-gpus` will now produce a result like:
0: Quadro M6000, 1114MHz, 12212MB
on stdout with one line per device. Note this may not list any device where CUDA is disabled despite being capable
- Attribute "dice" "string offscreenstrategy" now accepts "planarprojection" (the default regular dicing strategy) as a way of explicitly opting out of the default "viewfrustumdistance" offscreen strategy
- Corrected PxrAttribute type parameter default to be "float"
- The default number of images that "it" will store in memory has been increased from 40 to 100. Using high numbers of AOVs would rapidly degrade performance. The setting remains in the Preferences dialog if further tuning is required
- OSL isconnected() shadeop will now return 2 for down-connections to non-OSL nodes (e.g. C++ BXDF's)
- Many simple EXR viewers can't recognize that Ci.{r,g,b} should be displayed as red, green and blue. The "EXR no AOV" option in 'it' now also substitutes the name Ci.r' for 'R' etc. Also provide a Float and Half choice for "no AOV" exports. The regular Export EXR options maintain the layer names as rendered for beauty and AOVs. This should make exporting IPR images and sending them to your (insert mobile device or preview software) easier
- The "exitat" checkpoint option will now work even if the "interval" checkpoint option isn't set
- Exposed premultipliedAlpha parameter on PxrCookieLightFilter and PxrGoboLightFilter

Fixes

- Fixed a bug where editing a geometry master instanced inside a group would yield a crash
- The normal N on brickmap gprims has been fixed
- Addressed a bug with checkpointing that could corrupt tiled openexr file output
- Fixed an issue with PxrSurface internal volumes and sample weighting
- Error codes returned by RixTexture::GetLastError() would sometimes erroneously refer to an unrelated texturing call. This meant texture shading plugins (such as PxrTexture) would not use missing colors to fill result entries when they should have done, yielding unitialized data
- There was a bug in the automatically computed trace bias for non-uniformly scaled objects. This has been fixed
- Fixed a bug where in some cases, the renderer would be provided with invalid values in transform matrices, yielding invalid instances bounds
- A problem has been fixed with inconsistent normals on highly detailed, pretessellated, undisplaced subdivision meshes, particularly near extraordinary features such as creases
- Fixed instability following mesh light deletion
- (To Tractor) Fix an issue that prevented rendering to "it" when spooling batch render jobs to Tractor
- Fixed bugs in recovering from checkpoints, particularly when an EXR uses asrgba
- Ray hit footprints and hence texture filter sizes on pretessellated displaced Catmark subdivision surfaces were wrong. This has been fixed and they now match RenderMan 21
- Fixed an issue with grouping that forced incorrect workflow for Manifold Walk
- Fix missing shadowSubset shadowing from nested instancing
- Fixed a rare bug where the combination of deformation and rotation motion could result in clipped motion blur trails when the number of motion samples varied across objects in the scene
- Fixed memory leak on multi-threaded scene ingestion
- A bug in the PxrDispTransform node resulting in incorrect displacement directions for Zbrush and Mudbox vector displacement has been fixed
- A bug where dPdu and dPdv in flat curves had wrong magnitude and may have resulted in wrong bump mapping or texture filtering in flat curves geometry
- Fixed an issue where oversmoothing of bump maps produced artifacts
- Fixed an issue where the renderer may crash when rendering Alembic files containing errors

- A fix for volumes/emission lifetime taking too long is included
- Various fixes for issues with AOVs, displays, denoise options reverts their behavior to 22.1
- Fix conditional visibility for thinShadow on Lights

Interactive/Live Rendering Limitations

- Crop window edits are restricted to fall inside the original crop window
- Bucket order or size cannot be changed during live rendering
- min and max samples settings cannot be altered during live rendering
- Changes to Presence do not update when using the opacity cache option
- Motion Blur will disappear during interactive rendering with scene changes
- Displacement does not update on changes
- Objects are not automatically re-diced during interactive sessions
- Mesh lights cannot be interchanged as geometry without restart
- Altering density of volumes does not update during IPR
- PxrUnified may crash while making interactive edits when using Manifold Walk and/or Indirect Guiding.

RenderMan Pro Server Limitations

- PxrUnified integrator does not yet support all the standard rendering features
- Meshlights cannot be instanced
- Load on demand procedurals are not supported anymore, all procedurals are now loaded immediately
- 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 bxdf
- 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 new mesh light on existing geometry during IPR results in double geometry
- Motion blurred polygons do not motion blur normals when deformed. Use Subdivision meshes instead
- 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.