RenderMan 23.5

Welcome to RenderMan 23.5!

This release introduces improvements to RenderMan. A new method for temporal blur of volumes is included and can improve performance in fantastic ways on some scenes. More savings can be found in using meshlights when it comes to memory consumption and convergence.

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

New Features in 23.5

- DisplayChannels now have an optional "shadowthreshold" control that can be used to tweak shadow pass AOVs where shadows could be cut off in darker areas. The default value of 0.01 gives the same result as before and you can lower that value to make missing shadows re-appear. *Warning: Setting this threshold too low may create fireflies*
- A new algorithm for deformation motion blurring volumes is now used by default. The advantages of the new "Eulerian" motion blur versus the previous method ("Reves") are vastly reduced preprocessing time, reduced memory consumption, and lack of (or at least greatly reduced) bias. The downsides are slightly slower convergence time, and the requirement that the velocity data exists throughout all regions of the volume along the intended path of motion. The new Attribute "volume" "int temporalmethod" can be used to revert to the previous behavior (by setting the value to 1).
- Two new volume attributes have been added to simplify the retiming of volume velocity data. Attribute "volume" "float fps" specifies an inverse scale on the velocity values computed by the PxrVolume shader, and can be used instead of the velocity multiplier on that shader. Attribute "volume" "float shutteroffset" specifies how the velocity should be applied to the volume relative to the current shutter: a value of 1 means that the volume positions are at shutter open, and the velocity is applied to move those positions to shutter close; a value of -1 means that the volume positions are at shutter close, and the velocity is applied to move those positions from shutter open.

Fixes

- Fixed a threading bug that would cause a crash when many alembic archives were rendered together
- Previously, the PxrSurface bxdf's user AOVs would be black when all its lobe gains are 0. This is no longer the case
- Adjusted the precision of ray/curves intersections to prevent very small curves (1e-4 in length) from fragmenting/disappearing partially
- Fixed several bugs where the adaptive sampler may interact poorly with checkpointing and not recover all of its state. Validation is now more thorough when warning if any channels driving the adaptive sampler are not going to be included in any checkpoint images
- A bug that could lead to incorrect results for partially opaque RiPoints close to each other has been fixed
- BXDFs with "Volume" in their plugin name that are intended to behave like PxrVolume will now actually behave like PxrVolume when setting up their internal visibility when used as light sources
- Volumes that are removed from a light's shadowing subset now correctly avoid self-shadowing themselves
- The limitation in PxrVolume where the densityFloatPrimVar and densityColorPrimVar parameters could not be used with deformation motion blur has been removed.

Miscellaneous Changes

- Mesh lights and emissive volumes are now importance sampled according to relative power over the surface/volume. For volumetric light sources in particular, this can greatly improve the convergence
- The memory consumption of volumetric meshlights has been substantially reduced, by at least 2X and up to 30X or more depending on the data

Known Limitations

Interactive/Live Rendering Limitations

- Bucket size cannot be changed 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
- Objects are not re-diced during interactive camera edits
- Mesh lights cannot be interchanged as geometry without a restart

RenderMan Pro Server

- When baking illumination sample filters and display filters are not currently supported and may lead to a crash.
- PxrUnified integrator is currently experimental as it does not yet support all the standard rendering features.
- 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 lights.
- Primvars attached to geometry may not have colons in their names.