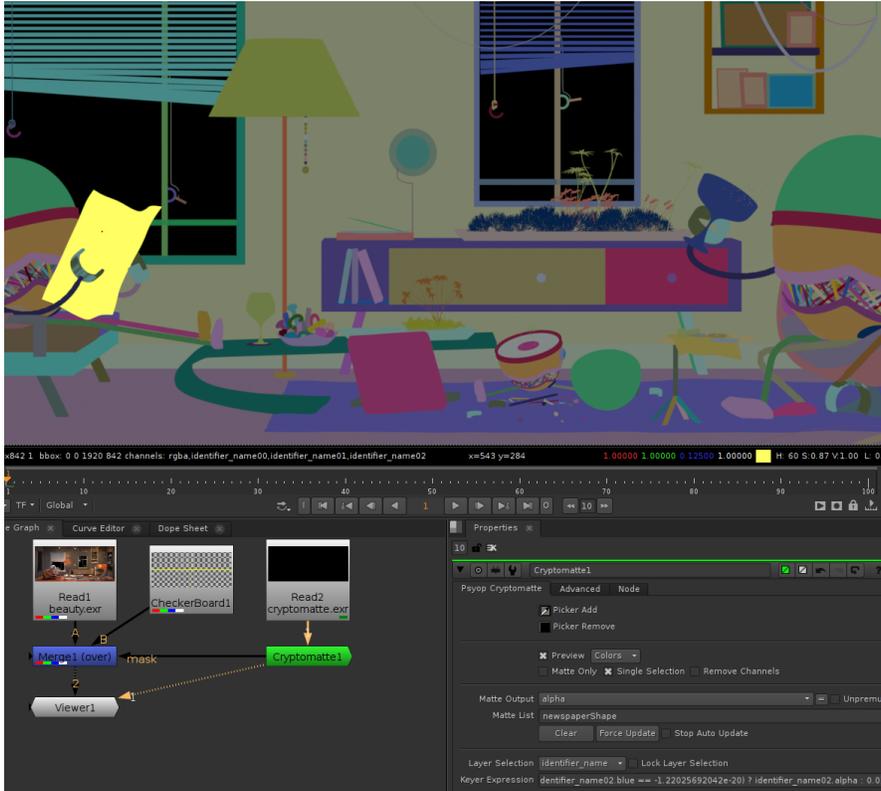# RenderMan 21.7

Welcome to RenderMan 21.7!

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

New Features

- Compatible with the Katana 3.0 Beta Release
- Cryptomatte support is added. Find example workflows in the Maya and Katana documentation. Volumes are not yet supported and will be opaque.

Cryptomatte output in Nuke

- 'it' now has expanded exposure controls allowing better viewing of HDR images.
- Mesh lights now support the same shadow controls and shadow linking as the analytic lights.
- Improvements to light selection on scenes with many lights should improve render times.
- Performance improvements to volume rendering, especially very sparse volume fields and/or single scatter volumes.
- OpenEXR texture reads (.tex as EXR) are faster.
- Added more controls to texture patterns:
    - This includes:
        - PxrTexture
        - PxrProjectionLayer
        - PxrLayeredTexture
        - PxrMultiTexture
        - PxrNormalMap
        - PxrBump
    - Added 3 new parameters to texture patterns:
        - int mipBias: bias mip selection, negative numbers bias lower resolution while positive numbers bias to higher resolution textures.
        - float maxResolution: clamp mip selection, useful to prevent loading of higher resolution mipmap levels.
        - optimizeIndirect: sets to an inexpensive Box filter on indirect hits, except on PxrBump where the filter is fixed.
    - The user attributes, "int texturemipbias" and "float maxtextureresolution", will override these settings.
- Improvements to Display Filters allow for easier construction of non-physical effects.

Miscellaneous Changes

- GPU Denoise requires CUDA 7.5 and compatible hardware.
- For refraction, extinction and ssAlbedo are unconnectable on PxrSurface
- Added support for baking with displacement

- Improved precision of PxrRamp inputs
- PxrPortalLight: now supersamples the input texture, leading to less potential pixelization on low resolution textures
- PxrSphereLight's intensityNearDist is now more correct/precise.
- Path Traced Subsurface options in PxrSurface now converge more quickly for high phase values
- Lighting Services:
  - Minor changes to horizon culling of lights and clusters to the light selection algorithm
  - Additional code robustness for lights: Fixed crash in cases where mesh lights were instanced
- PxrPortalLights, added more protection against slightly negative dot products in the portal sample function.
- PxrMarschner lobe picking improvements, this reduces the variance, especially for high albedo hair
- Memory optimizations in KD tree representation of volumes
- Optimizations for volumes in the case of low density volumes
- Updates to remove subsurfaceZeroScatterAttenuation to make sure that PxrLayerSurface reflects the new approach in PxrSurface
- Prevent PxrOcclusion from calculating occlusion for infinite lights.
- Add new RixShadingContext::GetNearestHits method with excludeSubset parameter.
- Reduced memory usage when baking complex scenes with PxrBakePointCloud.
- PxrDisney "ClearCoat" lobe has been renamed "Clearcoat" for output to standard LPEs used by other shipped materials.
- Thread contention has been reduced when emitting many RiObjectInstances from multiple threads.
- Added new API calls that Display and Sample filters can use to perform image processing (e.g., for NPR) across bucket boundaries.

Bug Fixes

- Fixed a possible race condition when plugins in multiple threads access the same subdivision mesh via the RixSubdivisionSurfaceEvaluatio n interface.
- Fixed a light selection bug when a scene was made up of all infinite lights
- Fixed a possible crash when using Denoise on Windows.
- A crash caused by rendering alembic files with large number of points has been addressed.
- Fixed issue that 0% progress was reported after hours of rendering.
- Fixed a crash with PxrVCM and volumes
- Fixed bug causing occasional incorrect normals on displaced Loop subdivision surfaces.
- Avoid crashing if using the Bake Hider on an empty scene.
- A crash observed when enabling watertight dicing on degenerate subdivision faces has been fixed.
- Fixed a problem where PxrVolume applied to NURBs geometry sometimes created incorrect shadows.
- Fix problems when rendering the VLen, PWorld, and other geometric AOVs when using volumetric glass shaders.
- Fix an interaction between PxrPathTracer with Russian roulette enabled at depth 1 and PxrVolume with single scattering leading to incorrect alpha values.
- Handling of volumes with camera visibility 0 has been made significantly more robust when dealing with complex situations including overlapping volumes and scattering from surfaces inside the volume.
- A bug causing multiscattered volumes to evaluate non-varyingprimvar lookups for emission has been fixed.
- When the extrememotiondof hider option is enabled, frame number is now incorporated into DOF sampling, preventing "shower door" (constant sliding noise across the frame) artifacts on out-of-focus objects.
- Fixed an issue where autocrop would not work correctly in openexr displays
- Fixed bug causing potential for nan results from edge-on sampling of disk or rect lights when using emissionFocus.
- A bug that caused issues with clearcoat layering with PxrLayerSurface has been addressed.
- A bug that prevented dispScaleSpace from applying on the PxrDispTransform node has been fixed.
- Fixed an issue where the adaptive sampler would finish an image before exitat checkpoint was reached.

Known Limitations

RenderMan Pro Server

- Rendering an alembic file with any instances will only render the original geometry. The procedural does not call ObjectBegin/ObjectEnd or ObjectInstance so all instances are ignored.
- Full LPE support is only available with the PxrPathTracer and PxrVCM integrators.
- Cryptomatte
  - Crop renders write the whole image with black padding. These will still be correctly aligned with the main render.
  - Single-scatter volumes show as an opaque volume envelope. Multi-scatter volumes work correctly.
  - Checkpointing is not currently supported.
- 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", RiVolume
- 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.

- 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.