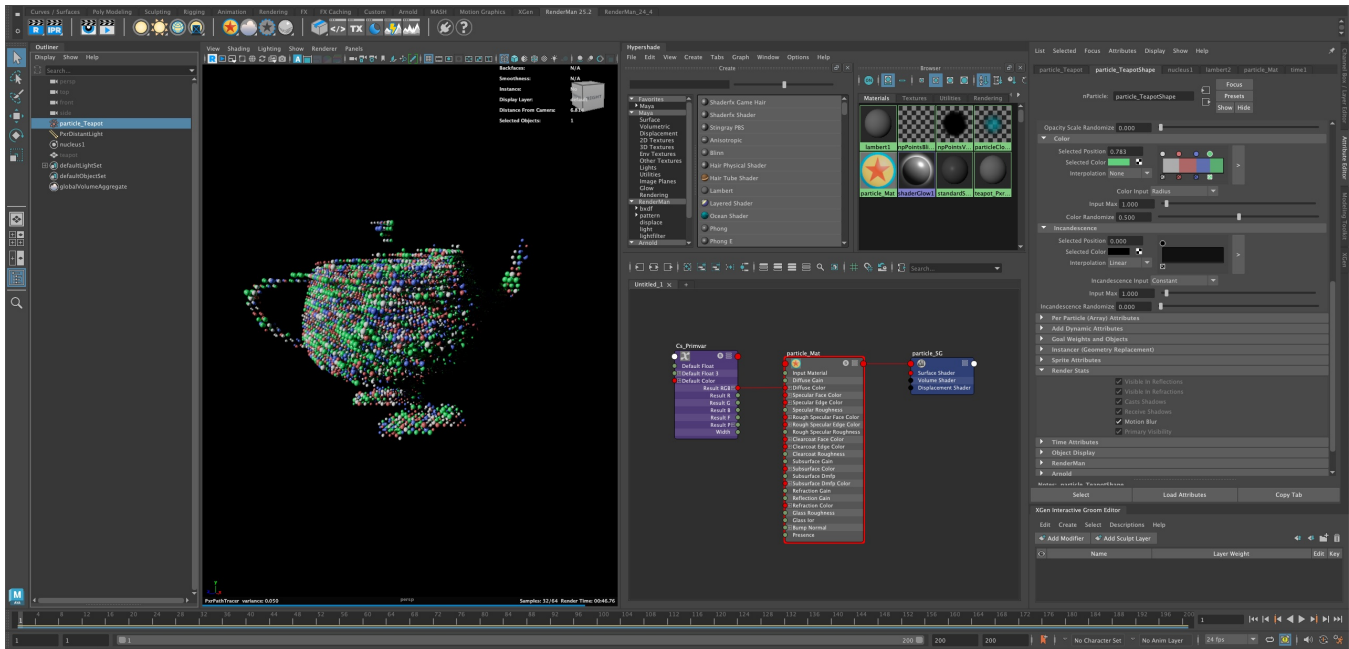


Particles in Maya



A RenderMan Walking Teapot full of particle spheres

RenderMan for Maya provides automatic support for Maya particles, including Maya particle instancing.

The following particle types are supported.

- Points
- Spheres
- Streak
- Multistreak
- Sprites



Real Primitives

RfM renders particles as real geometric primitives. RenderMan's highly optimized particle primitives are fast and efficient and carry many benefits, including:

- Antialiasing, motion blur, and full integration into the scene.
- Particles can cast shadows
- Arbitrary shaders can be attached

Also, when rendering particles through RenderMan you have control over the physical size of the particles in world space units. This is due to the fact that Pixar's RenderMan is resolution-independent, while hardware rendering modes commonly work in pixel units.



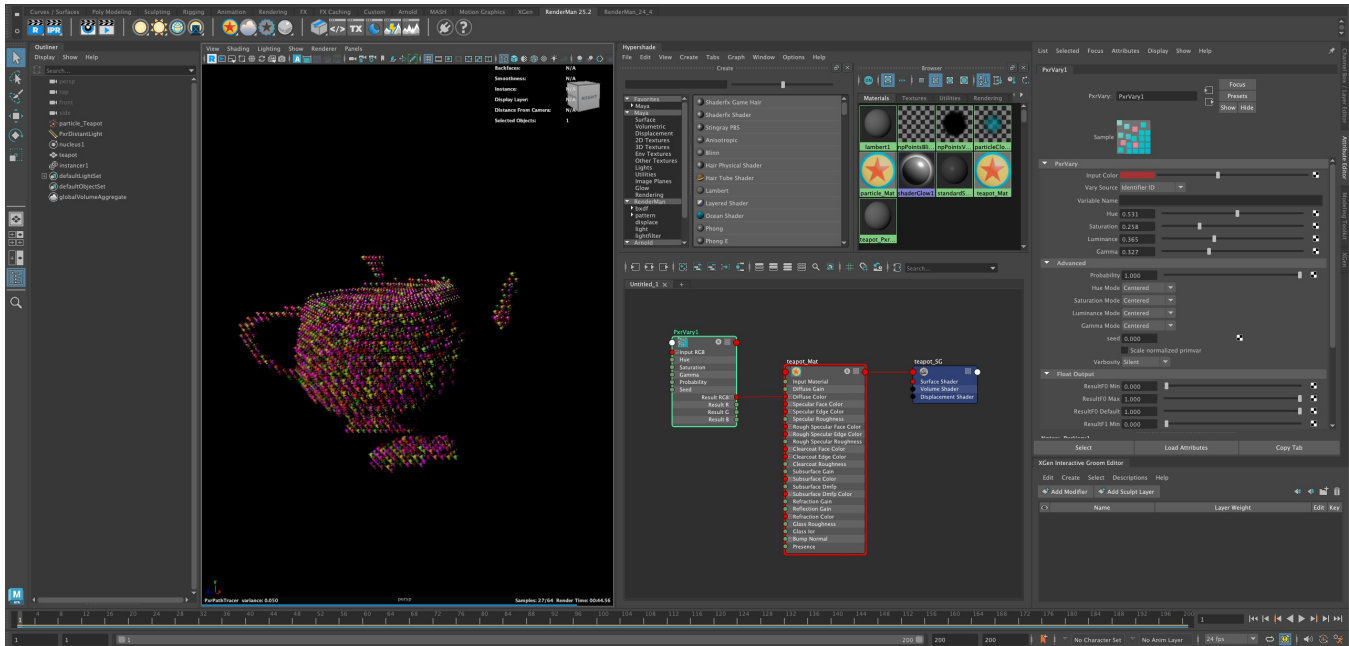
Ray Visibility

Make sure to force the particles to be visible for indirect rays, as Maya turns this off by default. You can use Maya's attribute spreadsheet to turn on "visible in refraction" and "visible in reflection" in your particle's Render Stats.

Particle Instancing

RenderMan for Maya supports Maya particle instancing. For complete details on how to set up your scene for particle instancing, please refer to the Maya documentation.

There are some cases where RenderMan for Maya does not correctly translate Maya's configuration. Generally, if you stick to the prescribed steps and avoid obscure orientation configurations, you'll get good results.



A RenderMan Walking Teapot full of particle spheres ... now instanced with tiny RenderMan Walking Teapots!

Per-Particle Effects & Arbitrary Attributes

RenderMan for Maya automatically establishes the correct rendering context for variations in particle color, opacity, and radius. A useful way to attach the particle's color and opacity, is to use the Cs and Os primvars. You can attach the primvars to the material by using a PxrPrimvar pattern.

You can write MEL expressions to calculate per-particle radii (**radiusPP**), color (**rgbPP**), opacity (**opacityPP**), and lifespan (**lifespanPP** and **useLifespanPI**). You can also use Maya to setup procedural ramps which remap these values.

Finally, you can create arbitrary attributes and associate them with the RenderMan representation of your particle objects. This might be of use if you plan to write a custom shader to perform calculations using per-particle or per-particle-object attributes. For more on the general mechanism for transmitting arbitrary primitive attributes through RenderMan, please refer to the primitive variables reference.