Using Displacement

Using PxrDisplace to create added geometric details is a common workflow when combined with packages like Mudbox or Zbrush to export maps. We recommend vector displacement as this makes the most sense for layering but we render both scalar and vector maps with the same node.

You'll need to make sure that the object with displacement has a non-zero displacement bound. For performance reasons this value defaults to zero so one must always increase this for the object where displacements are needed.

- Create a PrmanObjectStatements node and assign it to your object (using the CEL parameter)
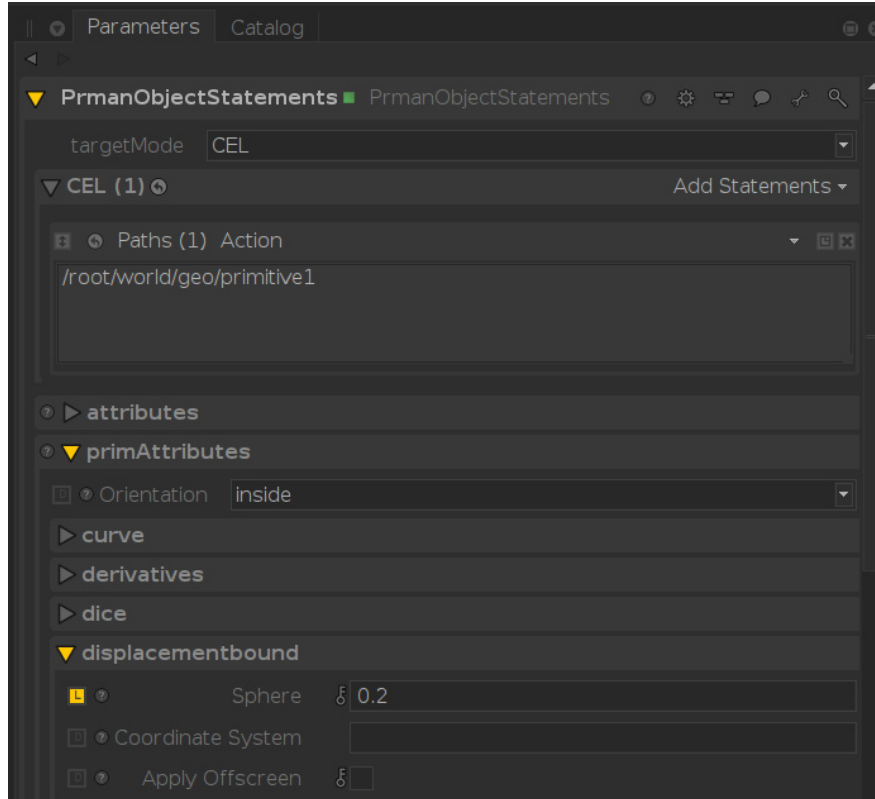
- Find the displacement bound in the PrmanObjectStatements node under "primAttributes.displacementbound"
- Set "Sphere" to a value greater than zero, something large enough to encapsulate your displacement but small enough that you won't cause the renderer to think that surface extends beyond it's intended final bound. If the bounds are too small you will see clipping or missing pieces. If too large, it may displace the object before it's necessary. We will automatically tighten those bounds afterwards to be more efficient. So better to err on too large than too small. A value of 0 (zero) disables displacement for the object.
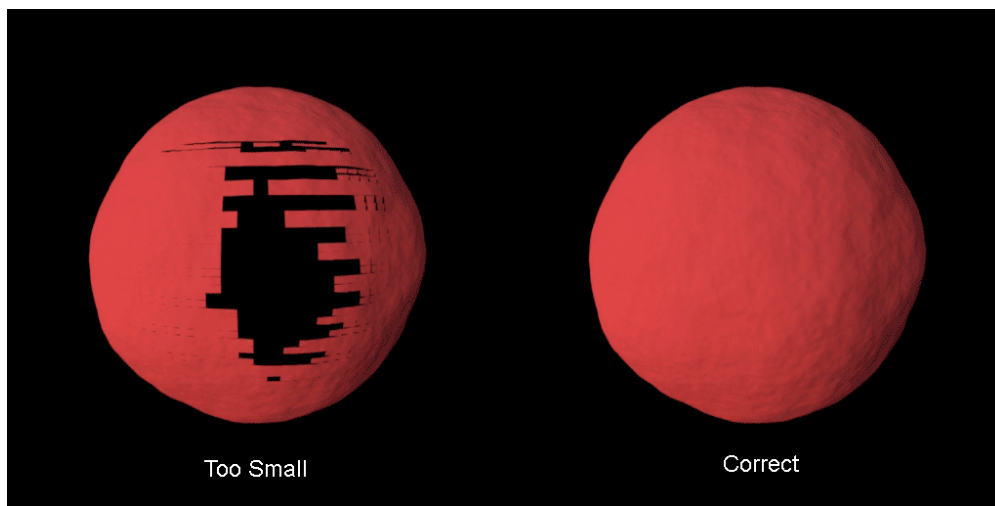
RenderMan will print a warning for bounds that are too large or too small by a minimum factor of 10 to help you adjust

Below is a typical network for adding displacement to your object:
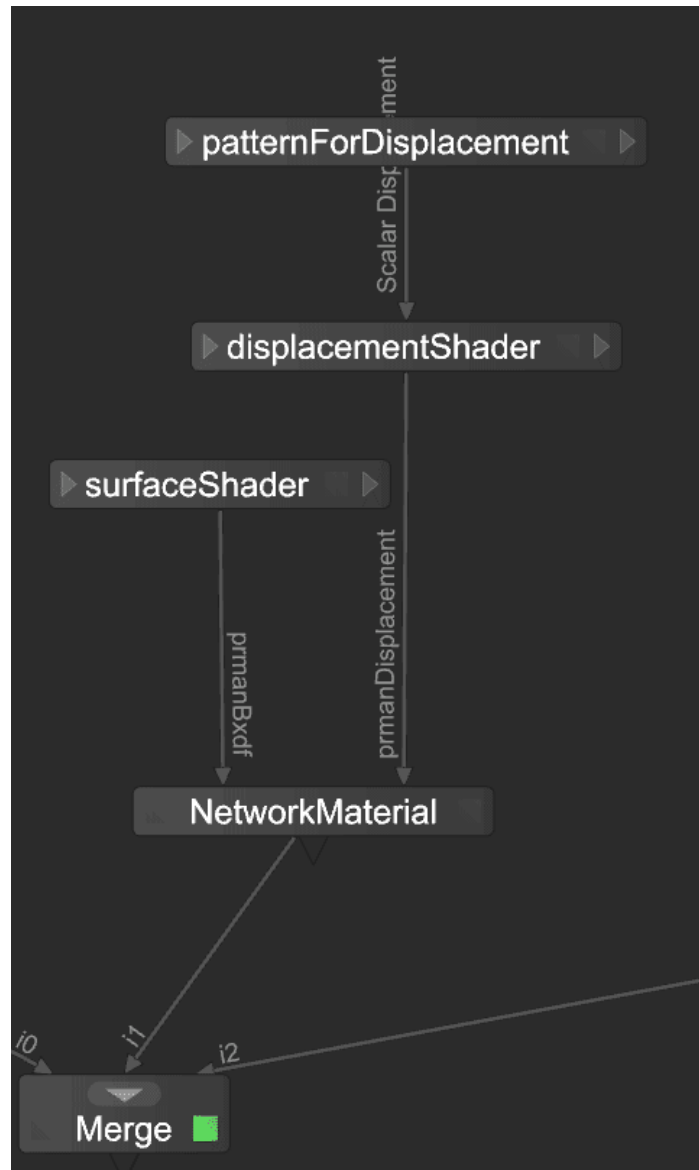
- Create a Network Material
  - Make a port for the Bxdf or surface shading using Add Terminal in the node
  - Create a second port for Displacement using the same menu
- A PrmanShadingNode is used as the surfaceShader, typically PxrSurface type
- A second PrmanShadingNode is a PxrDisplace type
- A pattern out (float for scalar displacement or RGB for vector) is connected to the displacementShader

Render to see your result. Be sure you have a MaterialAssign after the Merge node with the correct CEL path to your object(s) and the network material location set to the NetworkMaterial