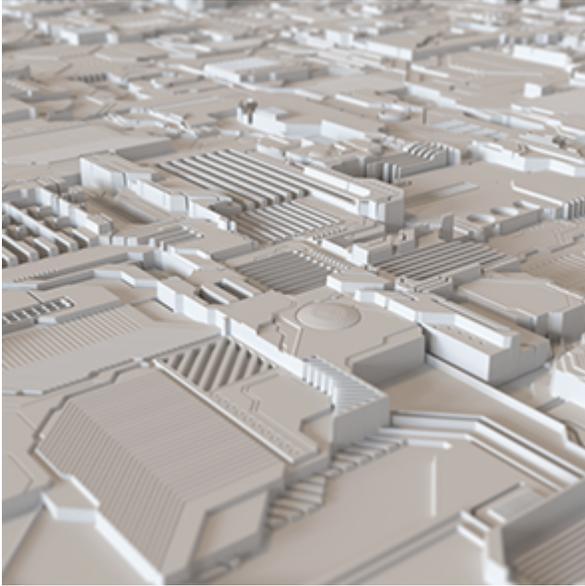


# Displacement Shading



Displacement texture map from [JSplacement](#)



Roll over image to view without displacement

Displacement maps can be an excellent tool for adding surface detail that would take far too long using regular modeling methods. Displacement mapping differs from bump mapping in that it alters the geometry, and therefore will have a correct silhouette, and self-shadowing effects. Depending on the type of input, the displacement can occur in two ways: Float, RGB & RGBA inputs will displace along the normal while a vector input will displace along the vector.

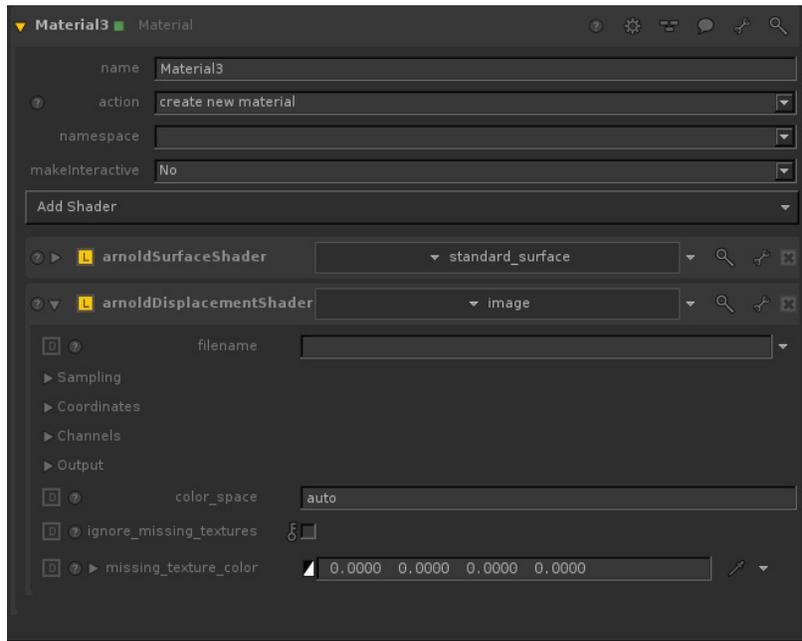
The example above shows how a simple plane, with the addition of a displacement map, can produce an interesting looking simple scene.

You should ensure that your base mesh geometry has a sufficient number of polygons otherwise subtle differences can occur between the displaced low-resolution geometry and the high-resolution mesh from which it was generated.

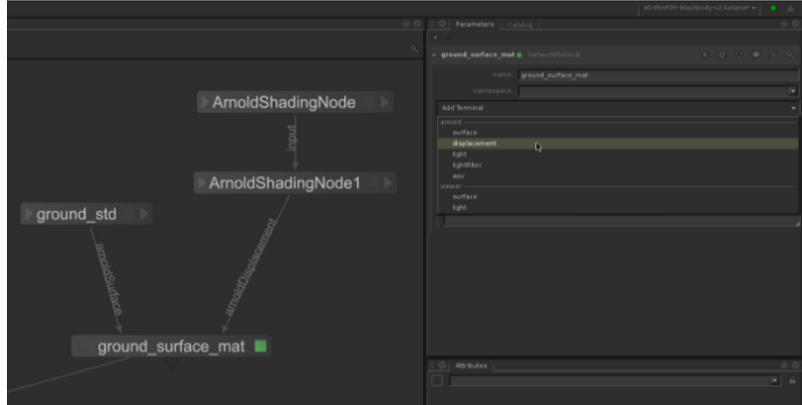


Make sure that you use a 32-bit or 16-bit floating-point format to store your image, and not an integer format. An integer format will not work correctly. This is because integer formats do not support negative pixel values, which are used by floating-point displacement maps.

There are two ways to set up displacement in KtoA. The first, if you only need one shader node (such as an image node for height displacement) is to create a **Material** node. You can then add an *arnoldDisplacementShader* (and also an *arnoldSurfaceShader* for the surface shading):



The other way is to create a network. Create a **NetworkMaterial** node, and add a terminal for surface shading (Arnold => surface) for your surface shading network, and also add a terminal for displacement shading (Arnold => displacement). You can then create **ArnoldShadingNode** nodes and set the shader type to whatever you need (such as *image*, *noise*, plug them through a *range* shader, etc). The last shader in the displacement network connects to the *arnoldDisplacement* port on the **NetworkMaterial** node:



After that, you can use a **MaterialAssign** node to assign the material (network or regular) to the *polymesh* or *submesh* locations you want to have the displacement.