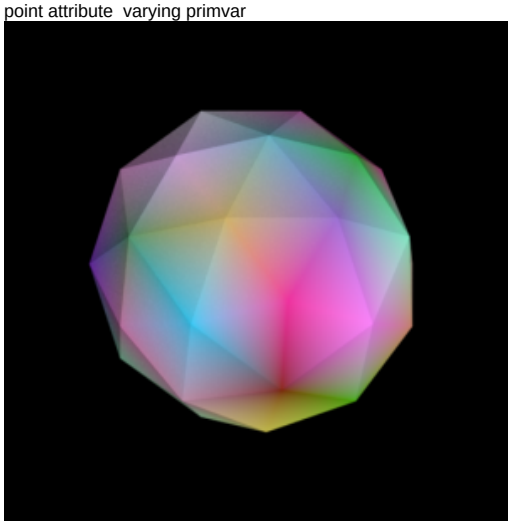
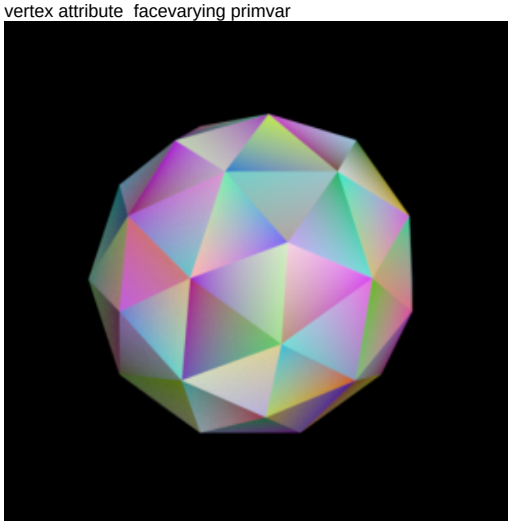
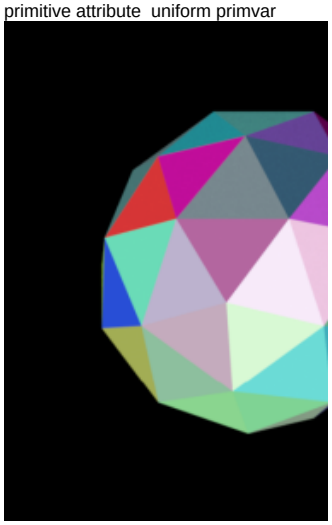
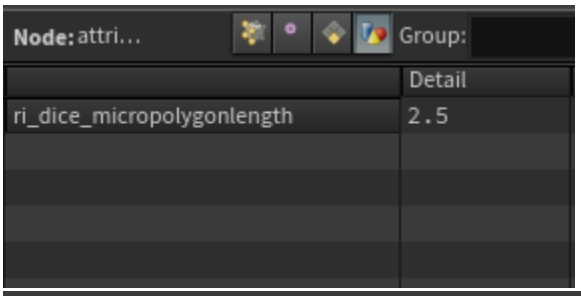
Using Primvars and Attributes

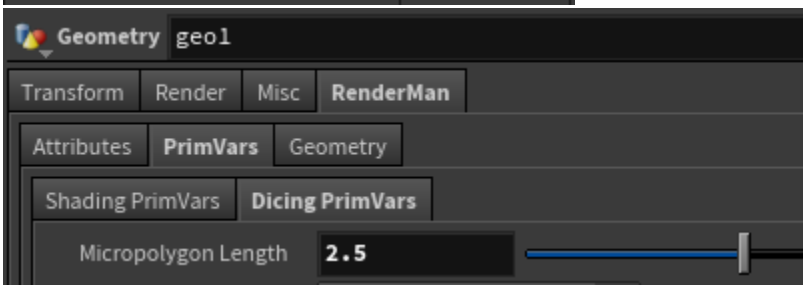
Primitive Variables (often referred as "primvars") are data attached to the geometry. This is useful when a shader needs to reference custom data from the geometry. There is no need to remap Houdini primvars to RenderMan primvars. All geometry attributes are passed to RenderMan as primvars with specific variation depending on attribute type.

The examples below show the difference between the four Houdini attribute types (point, vertex, uniform, detail) and how they translate to RenderMan primvars. All the Houdini attributes regardless of type are all RenderMan primvars so can be accessed via the PxrPrimvar VOP.

point attribute varying primvar	vertex attribute facevarying primvar	primitive attribute uniform primvar
		

Houdini attributes can also be used to control aspects of rendering such as shading, dicing, displacement, or geometry specific settings by creating primitive variables that are recognized by RenderMan. These can be set via a houdini detail attribute in SOPs or by adding RenderMan spare parameters to the OBJ. If both parameter and attribute are set, the detail attribute will override the OBJ parameter.





primvars.hip

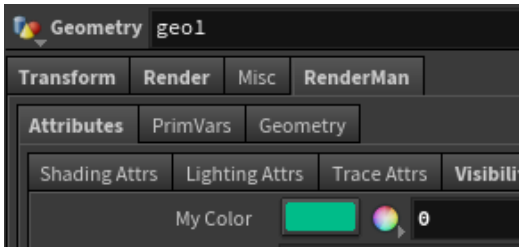
Instance Attributes are data attached to the instance. This useful for varying attributes across instances. Like primvars, instance attributes can also be added via SOP attributes or via OBJ parameters.

	via SOPs	via OBJs

user

Node: attri... Group:

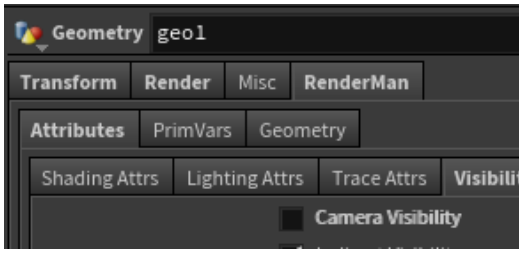
	Cd[r]	Cd[g]	Cd[b]
0	0.641601	0.20504	0.904597
1	0.800464	0.522061	0.401968
2	0.510895	0.885056	0.881155
3	0.775474	0.59574	0.7169
4	0.879506	0.828006	0.397275



rman

Node: attri... Group:

	ri_visibility_camera
0	0
1	1
2	1
3	0
4	1



attrs.hip