Appendix C – Shaders Reference

This reference provides information about the basic shaders that support RenderMan. The data has be compiled from “The RenderMan Companion” by Steve Upstill and the PIXAR document, “MacRenderMan Shaders” published (August 1990) as part of “MacRenderMan Developers Stuff”.

Like ‘plug-ins’ for the Adobe image processing software PhotoShop, an almost unlimited number of (potential) shaders can be added to a RenderMan environment – this document only describes the essential ones.

Shader Summary

LightSource
ambientlight
distantlight
pointlight
spotlight
pointnofalloff
shadowdistant
shadowpoint
shadowspot

Atmosphere
depthcue
fog

Displacement
cloth
dented
diaknurl
droop
emboss
filament
sinknurl
threads

Surface
blue_marble
carpet
checker
cmarble
glass
glassbal
glow
matte
metal
plastic
rmarble
rsmetal
rubber
screen
show_st
show_xyz
sinknurl
spatter
stippled
stone
texmap
txtplastic
wood
transparent_texture
eroded

blue_marble "blue_marble" "Ks" "Kd" "Ka" "roughness" "txtscale" "specularcolor"

"Ks" 0.4 "Kd" 0.6 "Ka" 0.1
"roughness" 0.1
"txtscale" 1
"specularcolor" [1 1 1]

This shader gives a surface a very delicate marble-like appearance. It gives the visual impression of turbulent fluid flow, as if the marble had been formed by molten coloured rocks. The txtscale parameter scales the turbulence.

carpet "carpet" "Ka" "Kd" "scuff" "nap"

"Ka" 0.1 "Kd" 0.6
"scuff" 1 "nap" 5 "swirl" 1

This shader produces a carpeted surface, complete with scuff-marks. The scuff parameter controls the “amount of scuff”, or the relative frequency of intensity variations. Higher values produce more frequent scuffing. nap describes the “shagginess” of the carpet. Higher values make a more coarse-looking carpet.

The carpet shader makes a reasonable stab at anti-aliasing, so the actual grain of the carpet fades away with distance.

There are no specular reflections from real carpet (at least on a macroscopic scale), so the only lighting parameters are Ka and Kd, which have the usual meanings of ambient and diffuse reflective intensities, respectively.

This way anti-aliasing is performed can cause linear artifacts in some cases.

checher "checker" "Kd" "Ka" "frequency" "blackcolor"

"Kd" 0.5 "Ka" 0.1
"frequency" 10
"blackcolor" [0 0 0]

This shader imposes a checker-board pattern over a surface. The frequency parameter sets how many times the pattern is to repeate itself within the texture space of the surface. Blackcolor sets the colour to be used for the pattern.

cmarble "cmarble" "Ka" "Ks" "Kd" "roughness" "specularcolor" "veining"

"Ka" 0.1 "Ks" 0.4 "Kd" 0.6
"roughness" 0.1
"specularcolor" [1 1 1]
"veining" 1

This shader produces a marble that consists of coloured veins on a white background with some greyish mottling. The vein colour is determined by the current surface colour. The parameter veining controls the frequency of the veins in the marble; higher values produce more and narrower veins.

The parameters Ka, Ks and Kd have the usual meanings of ambient, specular and diffuse reflective intensities, respectively. Roughness and specularcolor control the sharpness and colour of the specular highlight.

glass "glass" "Ka" "Ks" "Kd" "roughness" "specularcolor" "envname" "Kr"
glassbal "glassbal" "Ka" "Ks" "Kd" "roughness" "specularcolor" "envname" "Kr" "eta"

"Ka" 0 "Ks" 0.6 "Kd" 0
"roughness" 0.025 (for glass) or 0.2 (for glassbal)
"specularcolor" [1 1 1]
"envname" "your environment map"
"Kr" 0.5 "eta" 0.6

The glass shader makes an object appear to be made out of transparent and possibly coloured glass. No refraction is attempted, so the glass appears to be thin, but reflections are simulated using the environment map given with envname. The environment map's reflective intensity Kr can be controlled to fine-tune the appearance; a lower-intensity reflection may look better on very dark glass. Although you can use the shader even if you don't have an environment map (just ignore envname), it will look more like transparent plastic then glass.

You can use the shader for either clear glass or coloured glass by setting the surface colour (clear glass has color [1 1 1]). The transparency of the glass is controlled only by the surface colour; if you want to make the glass less transparent you should make the colour darker.

The glassbal shader can be used to make some objects look like solid glass. It simulates the refraction seen through a sphere turning objects seen through the glass upside-down and backwards. The shader needs the environment map given with envname to this refraction. It will not work without an environment map. Because the shader simulates refraction as if the object is a sphere, it works well on curvy objects like teapots (and spheres), but will not look correct on flat objects or cylinders.

The colour of the object can be set with the surface colour, just as with glass. The transparency is somewhat different here, however, because the camera is not really seeing "through" the object. Therefore, the surface opacity should always be set to [1 1 1], and the relative intensity of the "refraction" will again depend on the surface colour.

The parameter eta is the relative index of refraction of the atmosphere to the glass. By default, this is the standard value of air (1.0) relative to glass (1.66).

The parameters Ka, Ks and Kd have the usual meanings of ambient, specular and diffuse reflective intensities, respectively. Roughness and specularcolor control the sharpness and colour of the specular highlight.

glassbal needs an environment map; glass looks bad without one.

glow

"glow" "attenuation"

"attenuation" 2

This shader imparts a glow to a surface. The glow is brightest when looking directly at the glowing object, and falls off rapidly nearby.

matte

"matte" "Ka" "Kd"

"Ka" 1 "Kd" 1

A matte surface exhibits only diffuse reflection, because it scatters light uniformly with no preferred direction. That makes the apparent brightness of such a surface independent of the direction from which it is viewed.

metal

"metal" "Ka" "Ks" "roughness"

"Ka" 1 "Ks" 1
"roughness" 0.25

Very similiar to the matte surface shader except that this surface allows specular reflections to occur ie. reflections are concentrated around the mirror direction. Roughness controls the concentration of the specular highlight, a high roughness value giving a more diffuse reflection.

plastic

"plastic" "Ka" "Ks" "Kd" "roughness" "specularcolor"

"Ka" 1.0 "Ks" 0.5 "Kd" 0.5

"roughness" 0.1
"specularcolor" [1 1 1]

This shader models a plastic material as a solid medium with microscopic coloured particles suspended within it. The specular highlight is assumed to be reflected directly off the surface, and the surface colour is assumed to be due to light entering the medium, reflecting off the suspended particles, and re-emerging. This explains why the colour of the specular reflection is different from the surface.

rmarble "rmarble" "Ka" "Ks" "Kd" "roughness" "specularcolor" "veining"

"Ka" 0.1 "Ks" 0.4 "Kd" 0.6
"roughness" 0.1
"specularcolor" [1 1 1]
"veining" 1

This shader produces a marble that consists of red veins on a white background with some greyish mottling. The parameter veining controls the frequency of the veins in the marble; higher values produce more and narrower veins.

The parameters Ka, Ks and Kd have the usual meanings of ambient, specular and diffuse reflective intensities, respectively. Roughness and specularcolor control the sharpness and colour of the specular highlight.

rsmetal "rsmetal" "Ka" "Ks" "Kr" "roughness"

"Ka" 1.0 "Ks" 1.0 "Kr" 1.0
"roughness" 0.1

This shader uses random data for the reflection instead of requiring an environment map. The parameter Kr controls the intensity of the reflection. This shader is not recommended for simple spheres, it produces a chrome-plated look on objects with more complex curvature.

rubber "rubber" "Ka" "Kd" "txtscale"

"Ka" 1.0 "Kd" 1.0
"txtscale" 1.5

This shader is similiear to a matte shader except that it includes small amounts of white dust in the surface of the rubber. The amount of dust is controlled by the parameter txtscale.

screen

"screen" "Ks" "Kd" "Ka" "roughness" "density" "frequency" "specularcolor"

"Ks" 0.5 "Kd" 0.5 "Ka" 0.1
"roughness" 0.1
"density" 0.25 "frequency" 20
"specularcolor" [1 1 1]

This shader produces a wire-frame appearance. The frequency parameter controls how many grid lines there are in the surfaces texture space; the default produces 20 grid lines per surface. The density parameter controls the portion of the surface that is opaque. The default 0.25 means that the "wires" will cover 25% of the texture space.

show_st

"show_st"
no parameters

For each point on a surface, this shader sets its red and green colour equal to the texture coordinates at that point. Transparency cannot be set with this shader - all surfaces are set to be opaque.

show_xyz

"show_xyz" "xmin" "ymin" "zmin" "xmax" "ymax" "zmax"

"xmin" -1 "ymin" -1 "zmin"-1
"xmax" 1 "ymax" 1 "zmax" 1

This shader converts points within a bounding box, given by the parameters, into red, green and blue values.

spatter

"spatter" "Ka" "Ks" "Kd" "roughness" "specularcolor" "basecolor"
"spattercolor" "specksize" "sizes"

"Ka" 1 "Ks" 0.7 "Kd" 0.5 "roughness" 0.2
"specularcolor" [1 1 1] "basecolor" [0.1 0.1 0.5] "spattercolor" [1 1 1]
"specksize" 0.01 "sizes" 5

This shader makes objects look like blue camp cookware with white paint spatters. Actually, both the blue basecolor and the white spattercolor can be changed if you desire.

The paramter specksize controls the size of the paint specks as you would expect. However, there are a range of sizes of paint specks controlled by the parameter sizes. Lower (integer) values produce smaller and more uniform specks. Higher values produce some larger blotches and specks of many different sizes.

The parameters Ka, Ks and Kd, have the usual meanings of ambient, specular and diffuse reflective intensities, respectively. roughness and specularcolor control the sharpness and colour of the specular highlight.

This shader can have problems with aliasing.

stippled "stippled" "Ka" "Ks" "Kd" "roughness" "specularcolor" "grainsize" "stippling"

"Ka" 0.1 "Ks" 0.3 "Kd" 0.8
"roughness" 0.3
"specularcolor" [1 1 1]
"grainsize" 0.01
"stippling" 0.2

This shader makes objects appear to be made of plastic with lots of little bumps, as computer keyboards, camera surfaces, stucco and many other objects. This is done by making the surface appear to have intensity variation in small grains or granules. The parameter grainsize controls the size of these granules, and stippling controls the relative variation in intensity of the granules; larger values produce a rougher looking surface.

This shader makes a fairly good attempt at anti-aliasing itself, so the granules should appear to fade out with distance in a way similiar to a 'real' stippled surface.

The parameters Ka, Ks and Kd have the usual meanings of ambient, specular and diffuse reflective intensities, respectively. Roughness and specularcolor control the sharpness and colour of the specular highlight.

stone "stone" "Ka" "Ks" "Kd" "roughness" "specularcolor" "scale" "nshades" "exponent" "graincolor"

"Ka" 0.2 "Ks" 0.9 "Kd" 0.8
"roughness" 0.3
"specularcolor" [1 1 1]
"scale" 0.02 "nshades" 4
"exponent" 2
"graincolor" [0 0 0]

This shader makes objects look like they are carved our of grainular stone, like granite, by making "crystals" of varying intensity and colour. The parameter scale controls the size of the "crystals", or grains; larger values make larger grains. This is the only parameter that most users will want to change.

The parameter nshades is the number of unique intensity levels found in the

grains. Higher values of this will produce less "simplistic" looking stone. Setting nshades equal to 3 will produce stone that looks remarkably like the spattered-paint fake stone Zolatone. The exponent parameter controls the distribution of intensity levels; higher values push the intensities toward the darker end (more toward graincolor, as described below).

The "intensity" levels are actually levels of mixing two colours, the surface colour and the graincolor parameter. Since graincolor is black by default, the different colours normally are in fact different intensities of the surface colour. However, if you want red-and-green speckly stone for some reason, you could do this by setting the surface colour and graincolor appropriately, but you should probably set nshades to something pretty low to avoid getting lots of weird colours between red and green.

The parameters Ka, Ks and Kd have the usual meanings of ambient, specular and diffuse reflective intensities, respectively. Roughness and specularcolor control the sharpness and colour of the specular highlight.

This shader can have problems with aliasing.

texmap

"texmap" "Ka" "Ks" "Kd" "roughness" "specularcolor" "texname" "maporigin" "xaxis" "yaxis" "zaxis" "maptype "s1" "t1" "s2" "t2" "s3" "t3" "s4" "t4"

"Ka" 1.0 "Ks" 0 "Kd" 1.0
"roughness" 0.25
"specularcolor" [1 1 1]
"texname" "name of texture file"
"maporigin" [0 0 0]
"xaxis" [1 0 0] "yaxis" [0 1 0] "zaxis" [0 0 1]
"maptype" 3 (ie. no projection)
"s1" 0 "t1" 0 "s2" 1 "t2" 0
"s3" 0 "t3" 1 "s4" 1 "t4" 1

This shader texture maps a surface. The name of the texture file is given by texname. The parameters maporigin, xaxis, yaxis, zaxis, maptype, s1-4 and t1-4 are passes directly

Note that because the t component of a texture map is displayed on a monitor as increasing downward, textures mapped onto surfaces can easily appear to be upside-down. You should be careful to orient your coordinate system correctly when using projections for texture mapping. For example, if you are using planar projection you may want to have the y axis of the projection plane pointing down.

The maptype parameter indicates the following types of projection:
0 planar,

1 cylindrical,
2 spherical,
3 no projection, and
4 automap.

In the case of spherical mapping the parameters maporigin, xaxis, yaxis and zaxis describe the coordinate system of the projection sphere. The maporigin is naturally the center point of the sphere. The xaxis, yaxis and zaxis are points describing the 3 coordinate axes relative to maporigin. The texture map wraps around the "equator" of the sphere such that the seam is located on the positive x-axis of the sphere.

The surface is texture-mapped as if it were painted with the image in the file. Normal shading techniques are then used to render the surface, as specified in the normal way.

The parameters Ka, Ks and Kd have the usual meanings of ambient, specular and diffuse reflective intensities, respectively. Roughness and specularcolor control the sharpness and colour of the specular highlight.

txtplastic

"txtplastic" "Ks" "Kd" "Ka" "roughness" "specularcolor" "mapname"

"Ks" 0.5 "Kd" 0.5 "Ka" 1
"roughness" 0.1
"specularcolor" [1 1 1]
"mapname" "name of a texture file"

This shader is based on the "plastic" surface shader. The parameter mapname allows an image previously converted to a texture file to be mapped onto a surface.

wood

"wood" "Ka" "Ks" "Kd" "roughness" "specularcolor" "grain" "swirl" "swirlfreq" "c0" "c1" "darkcolor"

"Ka" 1 "Ks" 0.4 "Kd" 0.6 "roughness" 0.2
"grain" 5 "swirl" 0.25 "swirlfreq" 1
"specularcolor" [1 1 1]
"darkcolor" [dependent on the surface colour]
"c0" [0 0 0] "c1" [0 0 1]

This shader creates a realistic-looking wood. The frequency of the wood grain can be changed with the grain parameter. The relative amount or amplitude of the turbuent swirl in the grain is controlled by the swirl paramater, and swirfreq controls the frequency of this turbulence. Low values of swirl produce more uniform looking wood, while low values of swirlfreq make the wood

appear to be more knotty. Obviously these two parameters interact to a large extent. You should be careful not to set swirl too high or swirlfreq too low or the wood will become a jumbled mess.

The wood is simulated by creating a grain that is essentially composed of differently coloured concentric “cylinders” around a central axis defined by the two points c0 and c1. This axis is the z axis by default. Note that the orientat- ion of this axis can be varied either by changing these two parameters or by doing some transformations between the call to the shader and the definition of the geometry. Either one of these approaches may make more intuitive sense in different applications.

The colour of the wood will normally consist of bands of different intensities of the surface colour. This is the most generally useful way of invoking the shader. However, for special appearances this can be changed by changing the darkcolor parameter, which controls the colour of the dark grain of the wood. The different intensity levels are actually levels of mixing between this colour and the surface colour, so setting the surface colour to red and darkcolor to white will produce red wood with white grain and various shades in between.

The parameters Ka, Ks and Kd have the usual meanings of ambient, specular and diffuse reflective intensities, respectively. roughness and specularcolor control the sharpness and colour of the specular highlight. This shader can have problems with aliasing.

transparent_texture "Ks" "Kd" "Ka" "roughness" "specularcolor" "texname" "traname"

"Ks" 0 "Kd" 1.0 "Ka" 1.0 "roughness" 0.1
"specularcolor" [1 1 1]
"texname" ["name of the texure file to be used for texturing"]
"traname" ["name of the texure file to be used for transparency"]

This shader uses two texture files. Like the shader texmap, the file associated with "texname" is used as a texture map – except that this shader provides no control over the type of projection used. The other texture file, associated with "traname", controls the transparency of the surface(s) to which this shader is assigned. The gray scale values of "traname" alter the level of transparency of the shaded surface. Black pixels of the image make the corresponding parts of a surface fully transparent while white pixels renders the surface opaque.

eroded "eroded" "Ks" "Ka" "Km" "roughness"

"Ks" 0.4 "Ka" 0.1 "Km" 0.3
"roughness" 0.25

This shader erodes a "plastic-like" surface in such a way that parts of it are worn down to be transparent. The Km parameter controls the magnitude of the erosion.

LightSource Shaders

ambientlight "ambientlight" "intensity" "lightcolor"

"intensity" 1
"lightcolor" [1 1 1]

An ambient light source supplies light of the same colour and intensity to all points on all surfaces

distantlight "distantlight" "intensity" "lightcolor" "from" "to"

"intensity" 1
"lightcolor" [1 1 1]
"from" [0 0 0] "to" [0 0 1]

Unlike an ambient source, a distant source casts its light in only the direction defined by the from and to parameters. Otherwise the output light is the same as an ambient light.

pointlight "pointlight" "intensity" "lightcolor" "from"

"intensity" 1
"lightcolor" [1 1 1]
"from" [0 0 0]

A point light source is the converse of a distant light. It radiates light in all directions, but from a single location. It has a from parameter, but no to.

spotlight "spotlight" "intensity" "lightcolor" "from" "to" "coneangle" "conedeltaangle" "beamdistribution"

"intensity" 1.0
"lightcolor" [1 1 1]
"from" [0 0 0] "to" [0 0 1]
"coneangle" randians (30) "conedeltaangle" radians (5) "beamdistribution" 2

This shader reproduces the lighting effect of a spot light.

The intensity of the light varies from 0 (off) to any positive value (usually 1) representing the light at full intensity. The lightcolor parameter is an RGB triple representing the colour of light emitted by the source.

The from and to parameters specify the direction in which the light is shining. The coneangle and conedeltaangle parameters specify the distribution of the light as a cone-shaped beam, whose intensity falls off with the angle from the center to the cone. The falloff from the cone center is a "square-law" falloff (cosine of this angle raised to the power of 2) by default, but can be changed to a higher (or lower) power by setting the beamdistribution parameter.

pointnofalloff "pointnofalloff" "intensity" "lightcolor" "from"

"intensity" 1.0
"lightcolor" [1 1 1]
"from" [0 0 0]

This is a point light shader without an inverse square intensity falloff. It is useful for lighting a scene when you don't want to go through the process of calculating the large intensity usually required to get a normal point light to look the way you want it. It is also useful for producing uniform lighting from objects inside a scene, without the "pooling" normally associated with point light sources.

In addition, some modelling systems, for example, work only with point light sources. In such a system some point lights may be placed far away from the scene to simulate distant lights, and this shader is a good choice in such circumstances.

The intensity of the light varies from 0 (off) to any positive value (usually 1) representing the light at full intensity. The lightcolor parameter is an RGB triple representing the colour of light emitted by the source.

The from parameter represents the position of the light source in space.

shadowdistant "shadowdistant" "intensity" "lightcolor" "from" "to" "shadowname" "samples" "width"

"intensity" 1.0 "lightcolor" [1 1 1]
"from" [0 0 0] "to" [0 0 1]
"shadowname" (name of a shadow map)
"samples" 16
"width" 1

This is a normal distant light with an optional shadow map parameter given with shadowname. If a shadow map is not supplied the light behaves like a normal distant light source.

The paramter samples controls the sampling rate for filtering the shadow map. Higher values will produce less noisy-looking shadows, but will take

significantly longer. You can produce a (very noisy) test shadow very rapidly by setting samples to 1.

The width parameter controls "overfiltering" in the s and t directions. Higher values will give shadows more blurry edges, which can be used either as an effect or to hide the jagged edges or a low-resolution shadow map.

The intensity of the light varies from 0 (off) to any positive value (usually 1) representing the light at full intensity. The lightcolor parameter is an RGB triple representing the colour of light emitted by the source.

The from and to parameters specify the direction in which the light is shining.

shadowpoint

"shadowpoint" "intensity" "lightcolor" "from" "sfpx" "sfnx" "sfpy" "sfny" "sfpz" "sfnz" "samples" "width" "shadowname"

"intensity" 1.0
"lightcolor" [1 1 1]
"from" [0 0 0]
"sfpx" "sfnx" "sfpy" "sfny" "sfpz" "sfnz" (6 shadow maps)
"samples" 16
"width" 1
"shadowname"

This is a point light source that can cast shadows in all directions. To do this, you must supply 6 shadow maps, sfpx, sfnx, sfny, sfpz and sfnz for the positive and negative x, y and z directions respectively. This is very similiar to the idea of creating an environment map from 6 cube-face images. If any of the cube faces are not supplied, the shader will behave as a normal point light in those directions. For best results, the field of view for shadow images should be greater than 90 degrees (95 recommended).

The paramter samples controls the sampling rate for filtering the shadow map. Higher values will produce less noisy-looking shadows, but will take significantly longer. You can produce a (very noisy) test shadow very rapidly by setting samples to 1.

The width parameter controls "overfiltering" in the s and t directions. Higher values will give shadows more blurry edges, which can be used either as an effect or to hide the jagged edges or a low-resolution shadow map.

The intensity of the light varies from 0 (off) to any positive value (usually 1) representing the light at full intensity. The lightcolor parameter is an RGB triple representing the colour of light emitted by the source.

The from parameter specifies the position of the light source in space.

shadowspot "shadowspot" "intensity" "lightcolor" "from" "to" "coneangle" "conedeltaangle" "beamdistribution" "shadowname" "samples" "width"

"intensity" 1.0
"lightcolor" [1 1 1]
"from" [0 0 0] "to" [0 0 1]
"coneangle" randians (30) "conedeltaangle" radians (5) "beamdistribution" 2
"shadowname" "name of the shadow file"
"samples" 16 "width" 1

This light is a spotlight with an optional shadow map parameter shadowname. If a shadow map is not used the light is a normal spotlight.

The paramter samples controls the sampling rate for filtering the shadow map. Higher values will produce less noisy-looking shadows, but will take significantly longer. You can produce a (very noisy) test shadow very rapidly by setting samples to 1.

The width parameter controls "overfiltering" in the s and t directions. Higher values will give shadows more blurry edges, which can be used either as an effect or to hide the jagged edges or a low-resolution shadow map.

The intensity of the light varies from 0 (off) to any positive value (usually 1) representing the light at full intensity. The lightcolor parameter is an RGB triple representing the colour of light emitted by the source.

The from and to parameters specify the direction in which the light is shining.

The coneangle and conedeltaangle parameters specify the distribution of the light as a cone-shaped beam, whose intensity falls off with the angle from the center to the cone. The falloff from the cone center is a "square-law" falloff (cosine of this angle raised to the power of 2) by default, but can be changed to a higher (or lower) power by setting the beamdistribution parameter.

Displacement Shaders

cloth

"cloth" "freq" "depth"

"freq" 500
"depth" 0.02

This shader produces a cloth-like perpendicular weave pattern. The freq parameter changes the frequency of the "threads" (higher values mean the threads are closer together), and depth controls the height of the threads. The surface aliases pretty fiercely, but real cloth actually produces a somewhat similiar effect, so it looks fairly realistic.

dented

"dented" "Km"

"Km" 1.0

This shader produces a dented surface. The amount of denting is controlled by Km.

diaknurl

"diaknurl" "maporigin" "xaxis" "yaxis" "zaxis" "freq" "depth" "width" "radius" "zmin" "dampzone"

"maporigin" [0 0 0]
"xaxis" [1 0 0] "yaxis" [0 1 0] "zaxis" [0 0 1]
"freq" 10 "depth" 0.25 "width" 0.05
"radius" (refer to notes)
"zmin" "zmax" (refer to notes)
"dampzone" 0

This shader cuts a diamond knurl pattern into a cylindrical object. The parameters maporigin, xaxis, yaxis, and zaxis are used to do a cylindrical projection.

The freq parameter gives the number of grooves to cut per unit length along the z axis; higher values give closer grooves. The depth parameter controls the depth of the grooves, and the width parameter controls the width of the grooves. In order to render a correctly diamond-shaped pattern, the radius of the cylindrical object must be given with the radius parameter.

By default, the diamond knurl will be rendered along the entire length (along the z axis in shader space) of the cylinder. However, you can set minimum and maximum bounding z values with zmin and zmax parameters. The surface will not have knurl pattern cut outside these boundaries. In addition, you can make the knurl smoothly fade out instead of abruptly stopping at

these boundaries by setting the dampzone parameter. This parameter controls the width of the zone in which the depth of the grooves goes to zero. This zone is inside the zmin and zmax boundaries.

Remember to set the displacement bounds attribute when using this shader.

This shader can experience severe aliasing.

droop

"droop" "Km"

"Km"

This shader droops or sags a surface downward as if under the influence of gravity. 'Downward' for this shader means moving a surface in negative y.

emboss

"emboss" "Km" "texname"

"Km" 0.03
"texname" "name of a texture file that will control the embossing"

This shader embosses a surface according to an image given with the parameter texname. Pale areas of an image used for the texture file will push the surface "inward". The magnitude of the displacement is controlled by the paramter Km.

filament

"filament" "frequency" "phase" "width"

"frequency" 5.0
"phase" 0
"width" 0.3

This shader turns a cylinder into a spiral light-bulb filament. The filament can be brought to a point by applying the same shader to two cones at the ends of the cylinder. The frequency and phase parameters are identical to those of the "threads" shader.

sinknurl

"sinknurl" "maporigin" "xaxis" "yaxis" "zaxis" "freq" "depth" "zmin" "zmax" "dampzone"

"maporigin" [0 0 0]
"xaxis" [1 0 0] "yaxis" [0 1 0] "zaxis" [0 0 1]
"freq" 100 "depth" 0.005
"zmin" "zmax" (see notes)
"dampzone" 0

This shader cuts a sinusoidal knurl grooves along the length of a cylindrical object. The paramters maporigin, xaxis, yaxis and zaxis are used to do a cylindrical projection.

The freq parameter gives the number of grooves to cut around the circumference of the object. By default this number is quite high, but low numbers produce a shape like a classic Greek column. The depth parameter controls the depth of the grooves.

By default, the diamond knurl will be rendered along the entire length (along the z axis in shader space) of the cylinder. However, you can set minimum and maximum bounding z values with zmin and zmax parameters. The surface will not have knurl pattern cut outside these boundaries. In addition, you can make the knurl smoothly fade out instead of abruptly stopping at these boundaries by setting the dampzone parameter. This parameter controls the width of the zone in which the depth of the grooves goes to zero. This zone is inside the zmin and zmax boundaries.

Remember to set the displacement bounds attribute when using this shader.

This shader can have problems with aliasing.

threads

"threads" "maporigin" "frequency" "depth" "phase" "zmin" "zmax" "dampzone"

"maporigin" [0 0 0]
"frequency" 5 "depth" 0.1 "phase" 0
"zmin" "zmax" (refer to notes)
"dampzone" 0

This shader cuts a right-handed thread into a cylindrical object using the parameters maporigin, xaxis, yaxis, and zaxis to do a cylindrical projection.

The parameter freq gives the number of threads per unit length along the cylinder. The depth parameter controls the depth of the thread, and phase rotates the threads around the z axis of the cylinder. A value of 0 means no rotation and 1 means 360 degree rotation. This can be used to match threads from different cylinders.

By default, the threads will be rendered along the entire length (along the z axis in shader space) of the cylinder. However, you can set minimum and maximum bounding z values with zmin and zmax parameters. The surface will not have a thread pattern cut outside these boundaries. In addition, you can make the knurl smoothly fade out instead of abruptly stopping at these boundaries by setting the dampzone parameter. This parameter controls the width of the zone in which the depth of the grooves goes to zero. This zone is inside the zmin and zmax boundaries.

Remember to set the displacement bounds attribute when using this shader ie.

```
Attribute "bound" "displacement" [1.5]
```

This shader has problems with aliasing. The ShadingRate usually needs to be set to quite a low number.

depthcue "depthcue" "mindistance" "maxdistance" "background"

"mindistance" 0
"maxdistance" 1
"background" [0 0 0]

This atmospheric shader linearly adds the background colour according to the distance between the camera and a surface. No background colour is added if the surface is less than mindistance away. The background colour eliminates the surface colour entirely for points farther than maxdistance. In between, the two colours are mixed.

fog "fog" "distance" "background"

"distance" 1
"background" [0 0 0]

This atmospheric shader is somewhat more realistic for emulating atmospheric absorption. It assumes that the attenuation of the surface colour in the fog is never complete, as it is in the depth-cue shader.