Eze ShaderSuite

Contents

1

What is this about?

1.1 goals

-a beauty render to rul' it all
-enclose the whole lighting pipeline in only a few multipurpose shaders in-

troducing a whole renderman lighting suite ready for your furry needs
-simplify the lighting of characters with fur, including occlussion and gi, to work with 3d environments as well as location footage
-smart raytracing

In Figure 1.1, from left to right:
full beauty render with hair: envMap, occlusion, and global illumination
hair diffuse aov pass
full beauty pass with hair: envMap, occlusion 2 lights, sss
Bottom row: some of the many hair AOVs outputs

1.2 key features

1.2.1 ezeHair key features

-marchsner hair diffuse model
-environmental lights, occlusion and indirect diffuse ready

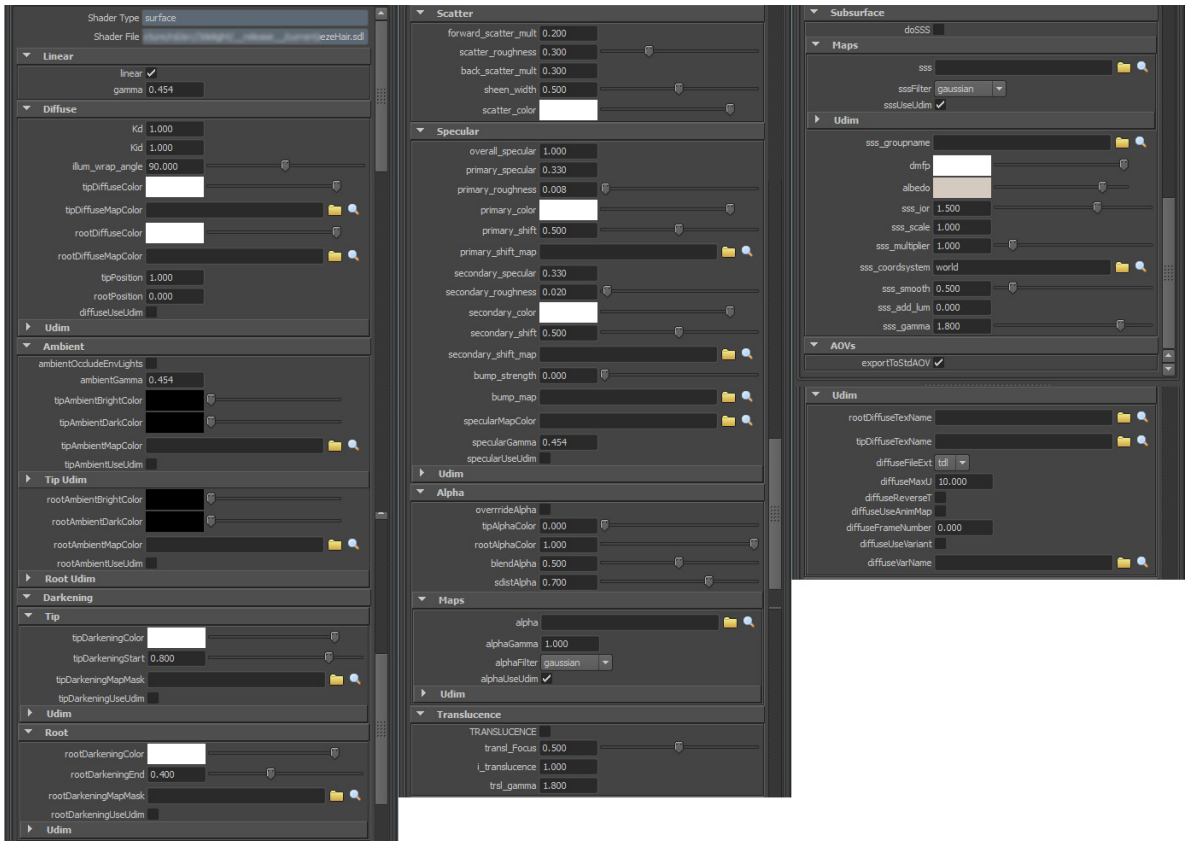
-baked maps capabilities to skip heavy lighting calculations
-2 specular layers
-extensive root/tips controls including color, darkening, alpha and ambient and maps for each
-all maps slots support uDim maps type

1.2.2 ezeSurface key features

-multiple diffuse and specular colors
-environmental lights, occlusion and indirect diffuse ready
-specular mask shape for alpha masking the specular highlights

2

ezeHair



In Figure 2, ezeHair shader parameters Interface.

2.1 linear

on/off

check this on when working in linear space

NOTE!: it is recommended to linearize the textures before hand using tdlmake. You using tdl files? Why not linearizing the texture when converting them? Refer to 3delight help on why not! using this setting.

gamma

If working In linear space and want to linearize your textures at render time, you can set this to 0.454. But you've been warned not to.

2.2 diffuse

In Figure 2.2, ezeHair shader AOV_hair_diffuse, with a high value wrapIllum Angle (notice the back lighted hair on the back of the neck) and ezeEnvironment. Note: Diffuse AOV contains the contribution from all direct lights and ezeEnvironment/ezeIndirect

Kd

Kd is a diffuse lighting multiplier. This will of course affect all kind of lighting, except if using the Ambient override to use a prebaked map for fur.

Ki

is the indirect diffuse lighting multiplier (any light contributing to the indirectDiffuse ie ezeIndirect). Keep in mind There is also global multiplier on the ezeIndirect shader.

Ka

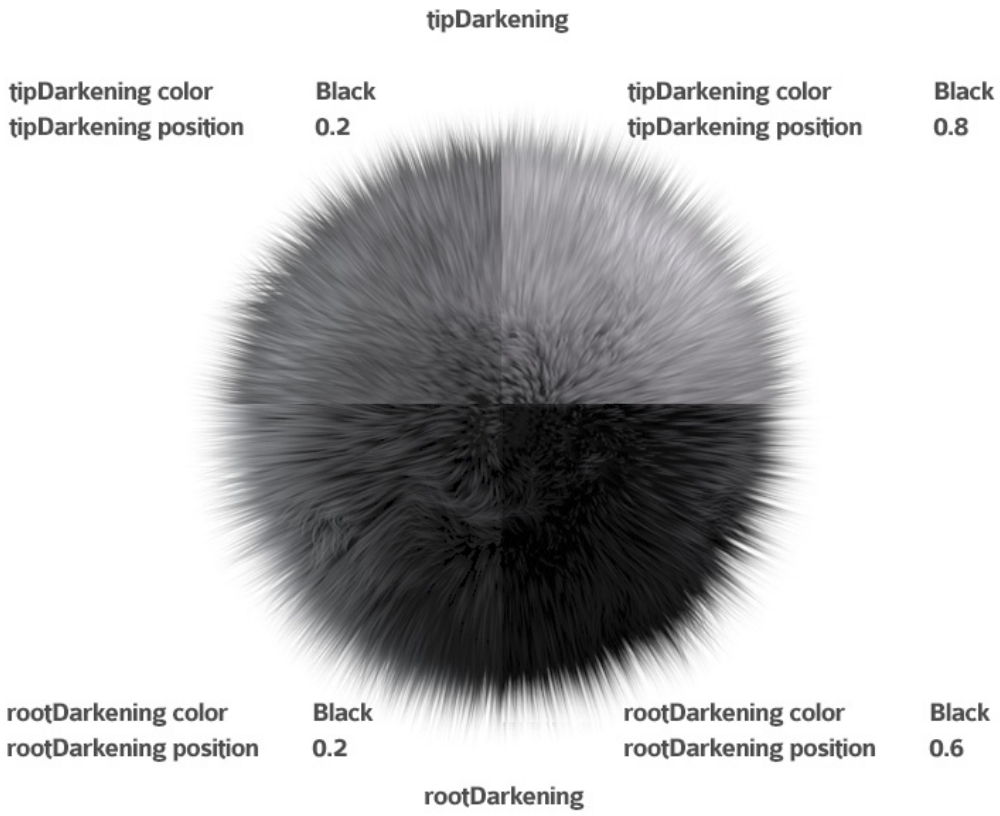
Ka is an ambient multiplier

Illum Wrap Angle

Illum Wrap Angle refers to the light wrapping angle of the light on the hair strand as a cylinder. The bigger the angle the more illumination it will show from side and back lights

Tip and root Colors, multipliers, controls and masks

You can use tip and root colors by themselves. When adding a map, the map will be multiplied by the tip and root colors according with the root and tip position. The position will control the ramp start and end points for the texture tip and root blend, saying that the hair strand root is 0 and the tip is 1.
If you want to use shaves root/tip colors,leave the maps blank, the colors to white, and check the export root/tip colors in your Shave Globals node.



In Figure 2.2, ezeHair shader parameters.

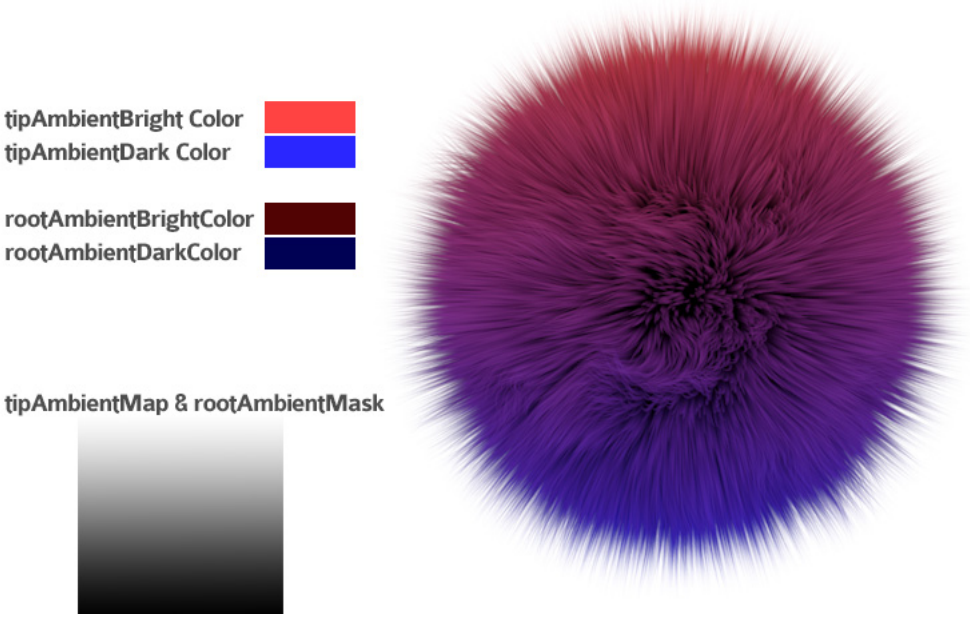
2.3 ambient

Ambient tip root controls

Control the ambient light fake for the tip and root using tipAmbientLightColor and rootAmbientLightColor.

Tip root Ambient Control maps

If you want to avoid calculating occlusion on the fly or basically want to have different ambient values, you specify a map in here, and you will get the full control over tip bright/dark colors and root bright/dark colors according to the map! An extra color multiplier for the tip and the root with ramp start/end points control that can be mask with a image.



In Figure 2.3, ezeHair fake ambient Map coloring.

ambientGamma

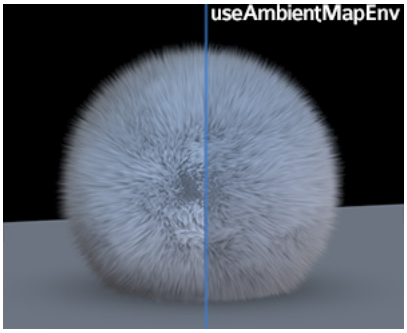
Use this parameter to control the gamma of the texture Mask.
The actual blended colors will retain the gamma value from the diffuse tab.

darkening controls and darkening masks

Controls the root and tip darkening, start/end points of the darkening and an optional mask map for the effect.

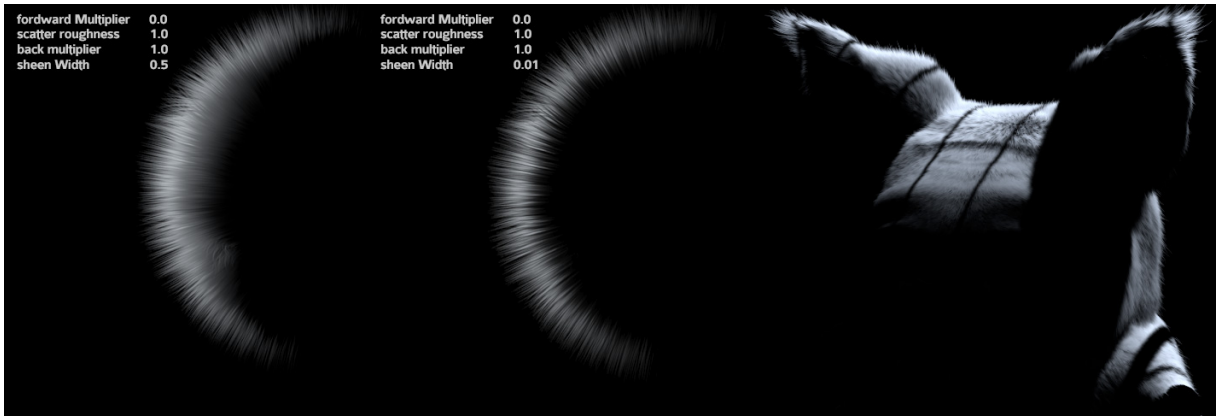
ambientOccludeEnvLights

If On, ezeEnvironment light will skip occlusion calculation and use the ambient map instead to occlude the sampled environment map. This is handy to speed up renders by skipping occlusion on the hair, or to fake the occlusion with a baked map but with some drawbacks & limitations, see below



In Figure 2.3, As you can see on the comparison, the occlusion is lost on the tips as the Ambient map here was calculated from the geometry. A quick fix for this would be to bake the hair occlusion as described on Pixars paper on Ratatouille.

2.4 scatter



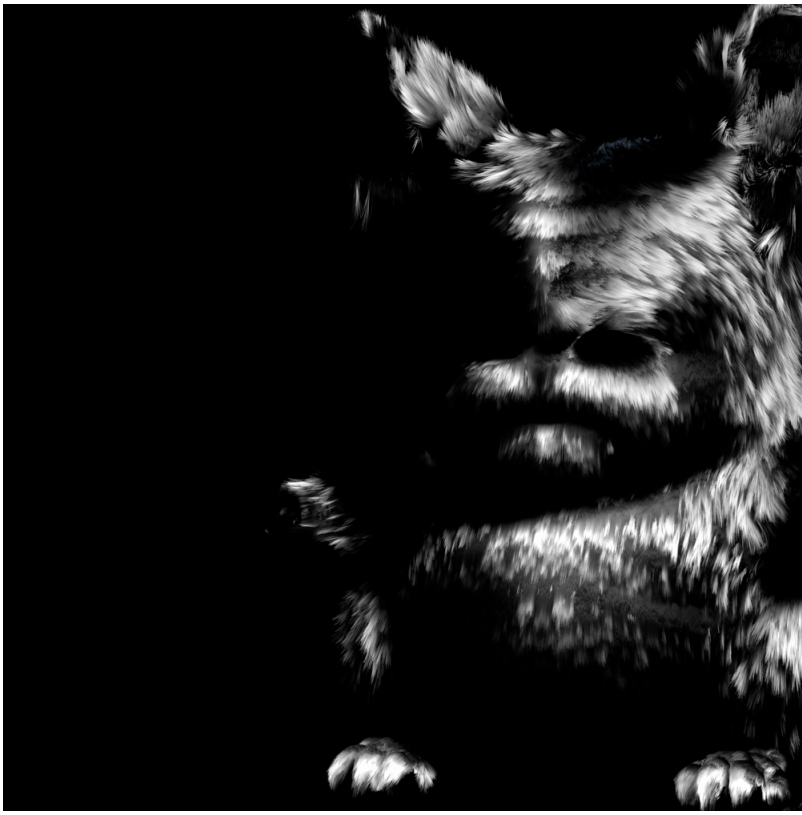
In Figure 2.4, ezeHair backScatter shader parameters.

settings

-Forward scatter multiplier
-Scatter roughness
-Back scatter multiplier
-Seen width
-Scatter Color
-Scatter Map

2.5 specular

Get the maya sample scenes file from the website to get the shift map that



you can use for the specular shift.
In Figure 2.5, ezeHair shader wet look aov_hair_primaryspecular

overall Specular

Turns on/off all specular highlights

primary specular

multiplies the specular values

primary specular roughness

controls the specular roughness

primary specular Color

color for the specular (map slot coming soon)

primary shift

shifts the specular along the heir lenght

Using Specular Shifts Maps

Use the provided map for this parameter.
This map is included in the maya scene samples texture/ folder.

2.6 alpha

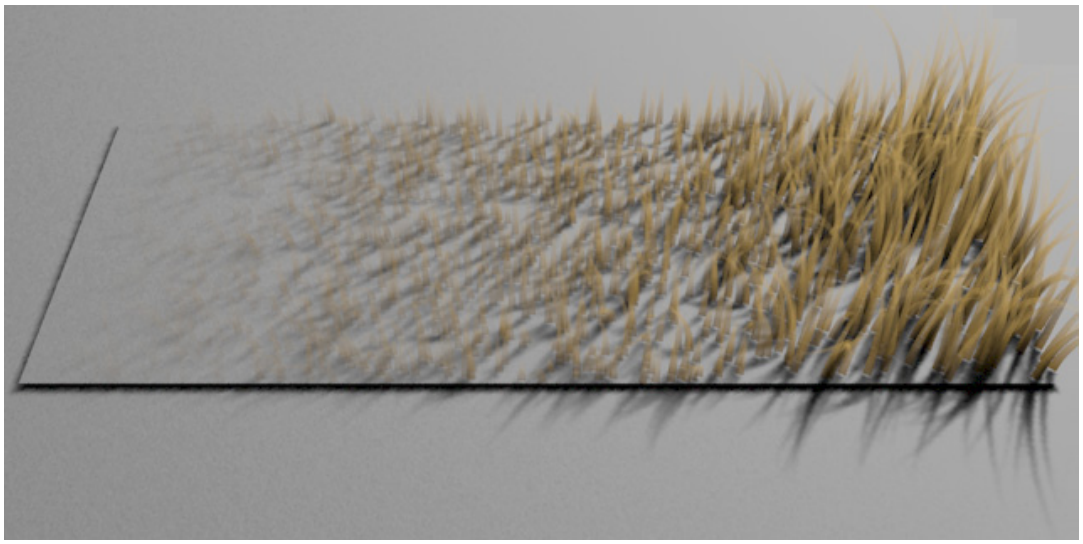
Either you overriding the Oi, using shaves root/tip transparency, or using a map remember to check Use Surface Shaders in shadow pass!
Note:ezeShaders will skip any lighting/sss calculation while not in "finalRenderPass".
If you are using any other shaders you might need to use a shaderCollection override for the shadow pass to avoid unnecessary calculations.

Overriding shave Surface Oi

You can either use shaves opacity (remember to check tip Fade and export opacity at the shaveShape and shaveGlobals), or override it with the controls under this tab.

alphaMap

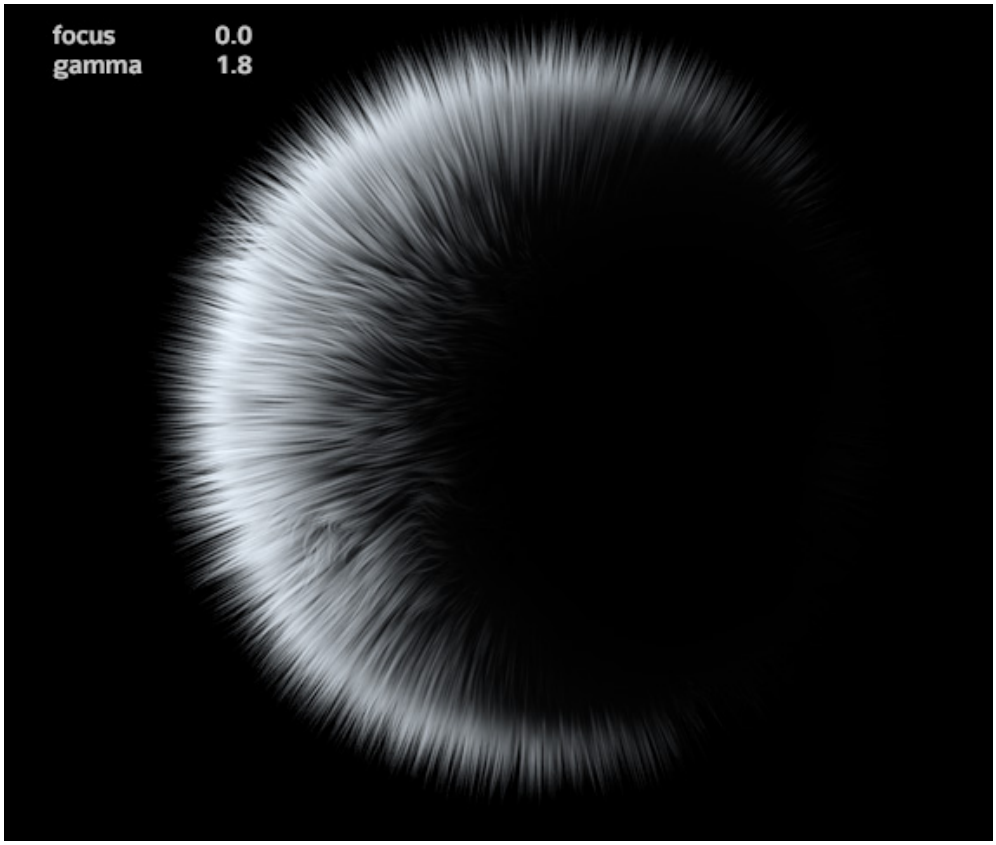
this map will multiply the opacity values wether you use the Oi or the alpha override parameters.



In Figure 2.6, ezeHair shader using alpha overrides and a alpha gradient black & white map_hair_translucence.

2.7 translucense

In Figure 2.7, ezeHair shader AOV_hair_translucence.



In Figure 2.7, ezeHair shader AOV_hair_translucence.

Focus

depth of the translucence effect

translucence

multiplier for the effect

translucence color

multiplier (coming soon)

translucence map

multiplier (coming soon)

gamma

controls the gamma for the effect

2.8 sss

For SSS to work under 3delight, attach a geometryAttr node to the hair and add the sss attributes and specify a group. The same group should be specified in the shader.
Recommended parameters for the sss attributes:
-Hair does not need dicing, you can add a shading rate multiplier and set it to 20 and add a DiceHair attribute and check it on.
-If you are baking the hair on a separate pass to a ptc to later read, you can scale the dice size to get a smoother look (ezeBake shader includes this parameter!)

maps

Cutout: This map will control where to skip sss calculation. This will speed up your renders quite a lot by isolating the calculation of the effect. Pure black means skip (check the threshold parameter). It is recommended to use the color map IF using a color map, and theres no need for a custom cutout map (Still not implemented)
cutout Threshold:
Use to set the threshold for the sss skip.
color: Multiplier for the sss color

2.9 AOVs

If you plan to use a composition software as Nuke to make your final image, you might want to take advantage of the fact that the multipliers for each effect are not taken into account on the aovs outputs. For ie: Setting the translucence multiplier to 0.01, it will make it virtually invisible on the final render, but the aov will contain the proper values as calculated in the shader prior to the multiplication.

2.9.1 output to standard aovs

This option will make the shader to output the aovs ALSO to the standard ones (this applies to the ezeSurface Shader).

2.9.2 aov listing

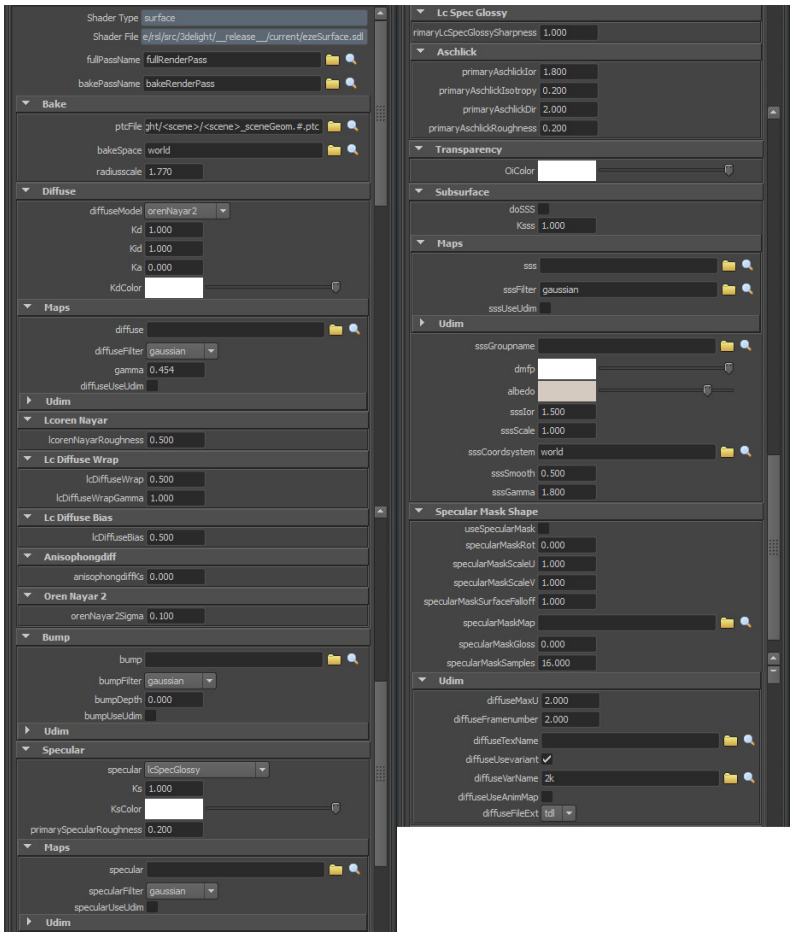
output varying color aov_hair_diffuse
output varying color aov_hair_indirect
output varying color aov_hair_totalscatter
output varying color aov_hair_frontscatter
output varying color aov_hair_backscatter
output varying color aov_hair_ambient
output varying color aov_hair_translucence
output varying color aov_hair_surfaceColor
output varying color aov_hair_totalspecular
output varying color aov_hair_primaryspecular
output varying color aov_hair_secondaryspecular
output varying color aov_hair_subsurface
output varying color aov_hair_v
output varying color aov_indirect

if output to standard aovs is checked, the shader will also contribute to the common ezeSurface aovs

output varying color aov_specular = aov_hair_totalSpecular
output varying color aov_diffuse = aov_hair_diffuse
output varying color aov_ambient = aov_hair_ambient
output varying color aov_surfaceColor = aov_hair_surfaceColor
output varying color aov_subsurface = aov_hair_subsurface
output varying color aov_indirect = aov_hair_indirect

3

ezeSurface



In Figure 3, ezeSurface shader parameters Interface.

In Figure 3, ezeSurface finalRenderPass and some AOVs outputs

3.1 diffuse

Passes configuration

fullPassName

This parameter is for specifying the pass name in which the beauty render takes place. In this particular pass everything is calculated and sampled. If not in this pass the subsurface, specular, indirect, and occlusion contributions are skipped (not needed for occlusion, or indirect lighting when baking ptc/bkm, etc.)
For skipping occlusion/indirect lights It specifically skips the one belonging to _cat: envLight.

bakePassName

In this section you can specify the bake pass and the target ptc file to avoid using a shader collection override on the objects at the bake pass. Use the same ptc file in several shaders and they will all concurrently write the information to the file. You can also leave this passName parameter blank nd use ezeBake in your bake pass instead.

Kd

diffuse lighting multiplier

Ki

the indirect diffuse lighting multiplier any light contributing to the indirectDiffuse_AOV, ie ezeIndirect. Theres also a multiplier in ezeIndirectShader

Ka

ambient value (independent from the ambientLights on the scene

KdColor

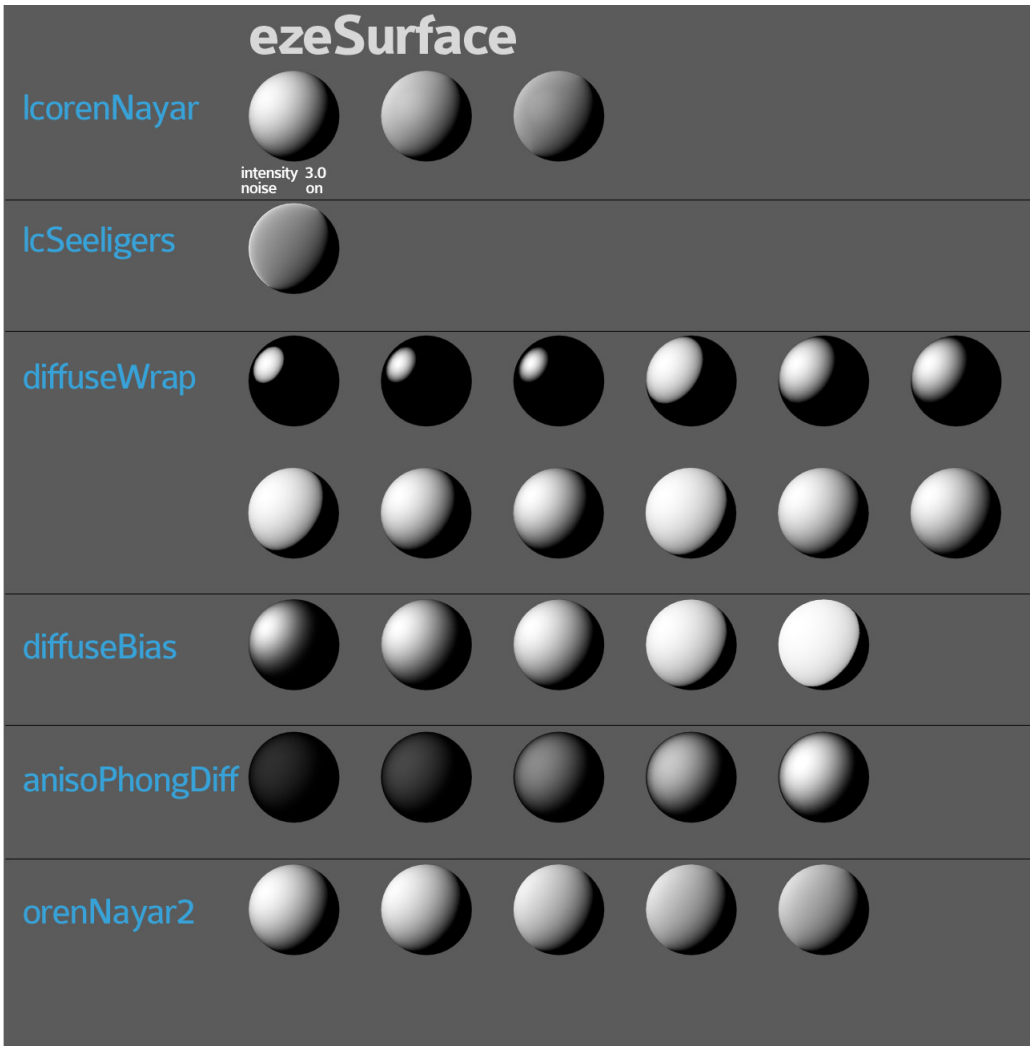
diffuse color to use if the map is not specified. If there is a map, the map will be multiplied by this color.

diffuse

color map

Gamma

Gamma for the diffuse map and the KdColor If working In linear space and want to linearize your textures at render time, you can set this to 0.454.
NOTE!: it is recommended to linearize the textures before hand using tdlmake. You using tdl files? Why not linearizing the texture when converting them? Refer to 3delight help on why not! using this setting.



In Figure 3.1.1, ezeSurface diffuse shader parameters.

3.2 bump

Map

Map to use for the bump (normal map option coming soon)

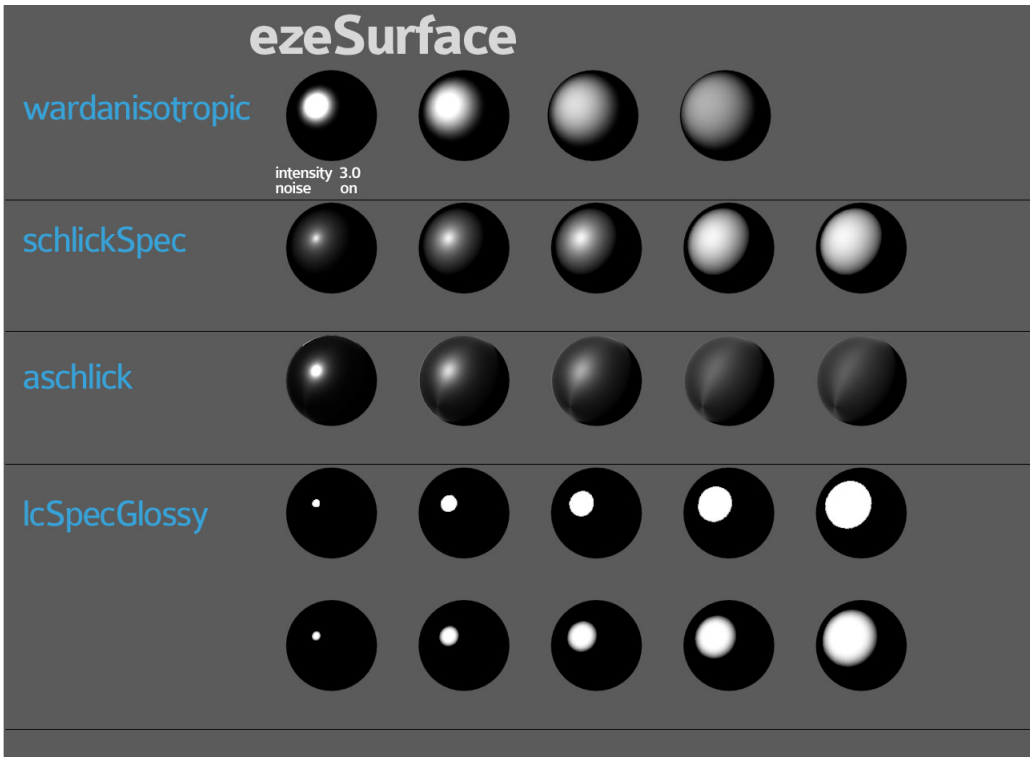
Depth

depth of the bump effect

3.3 specular

3.3.1 specularModels

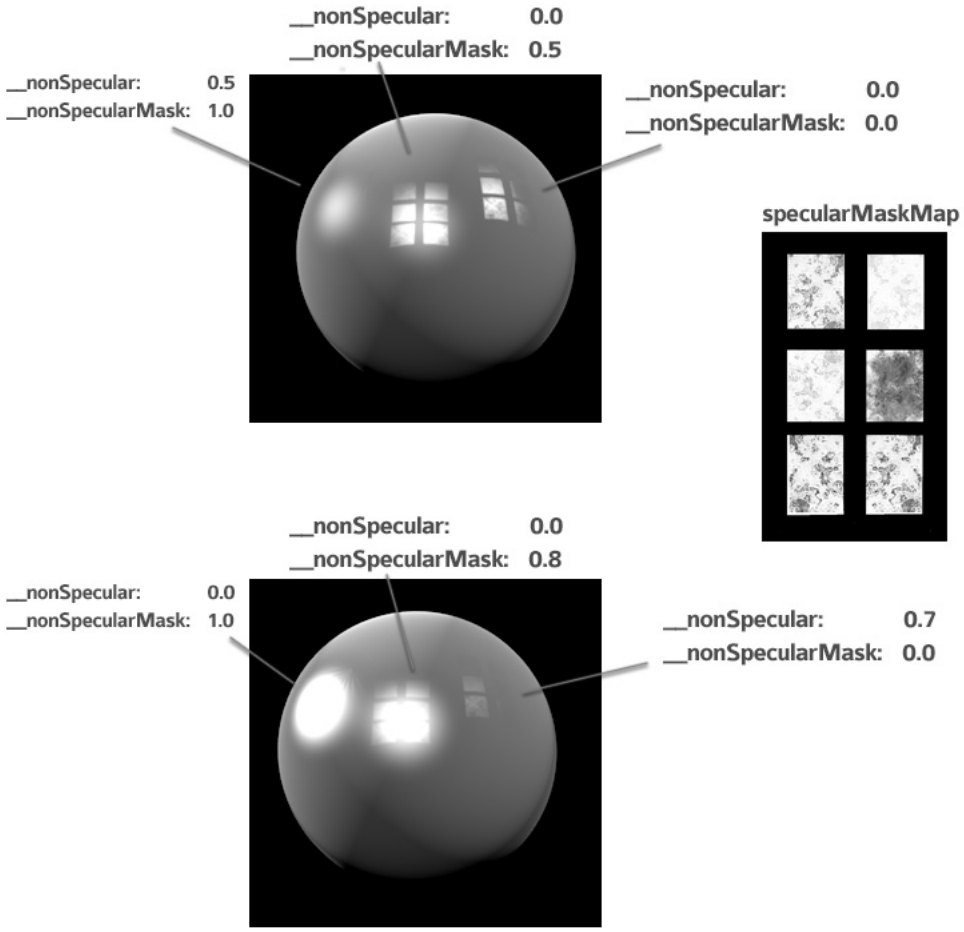
Here you can choose between a variety of Specular models for achiving different looks



In Figure 3.3.1, ezeSurface specular shader parameters.

NOTE: Some of this specular models are intended to use the surface normals disregarding the bump.

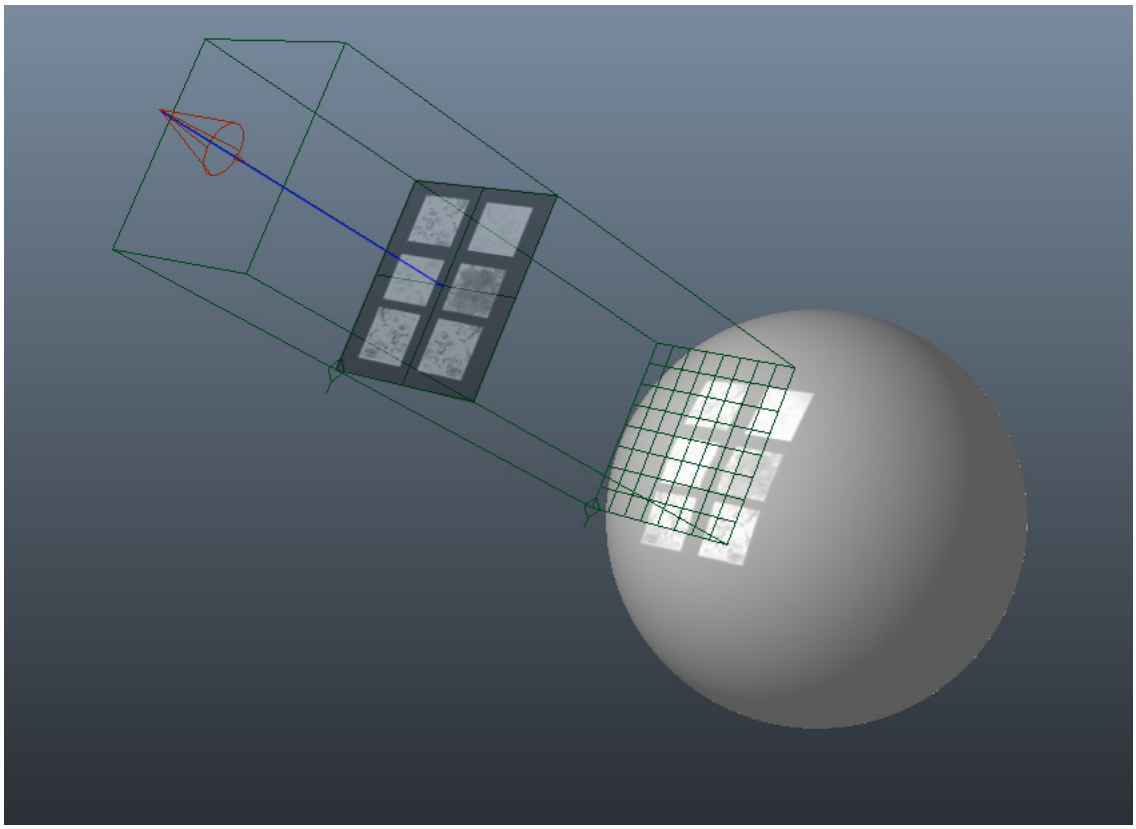
3.4 specularMaskShape



In Figure 3.4, ezeSurface specularMaskShape sample values.

This parameter is used for masking specular highlights with a map. The map is projected in 3d space from the light incident vector onto the surface to mask its specular contribution.

For further control ezeUber light shader has a specular mask contribution slider to attenuate or disable this effect on any light.



In Figure 3.4, ezeSurface specularMaskShape demostrative diagram.

useSpecularMask

Turns on/off the masking

specularMaskRot

Rotates the map projection on the light incident axis

specularMaskScaleU

scales the projection width

specularMaskScaleV

scales the projection height

specularMaskSurfaceFalloff

Fades the maps over the surface glazing angle (from the light point of view that is)

specularMaskMap

Specify an rgb map here.

specularMaskGloss

Leave this at 0 to get the mask without blurring.
Expect higher rendering times from this! This parameters uses the gather function,

specularMaskSamples

Samples for the gloss setting

3.5 sss

For SSS to work under 3delight, attach a geometryAttr node to the hair and add the sss attributes and specify a group. The same group should be specified in the shader.

-If you are baking on a separate pass to a ptc to later read, you can scale the dice size to get a smoother look (ezeBake shader includes this parameter)

cutout Threshold

Use to set the threshold for the sss skip. This value compared against the texture value or the sss color if not texture is used.
0.07 is a good start point as at that value the sss effect will probably go unnotice.
color: Multiplier for the sss color

3.6 shaderParameters

Oi

Surface Opacity multiplier (more parameters on transparency coming soon)

3.7 AOV

If you plan to use a composition software as Nuke to make your final image, you might want to take advantage of the fact that the multipliers for each effect are not taken into account on the aovs outputs. For ie: Setting the translucence multiplier to 0.01, it will make it virtually invisible on the final render, but the aov will contain the proper values as calculated in the shader prior to the multiplication.

3.7.1 AOV listing

output varying color aov_specular
output varying color aov_diffuse
output varying color aov_ambient
output varying color aov_surfaceColor
output varying color aov_subsurface
output varying color aov_indirect
output varying color aov_refraction

4

ezeDisplacement

In Figure 4, ezeDisplacement used to mimic short fur

4.1 displace

Using shave on a displaced geometry? You can easily copy the values from your shave displacement alphaGain and offset map here and it should match!!!

displacement

Overall displacement multiplier (usually you would leave this value at 1)

textureName

Displacement texture file

offset

Here you would usually put your displacement $-\frac{worldSpaceValue}{2}$

multiplier

Here you would usually put your displacement $\frac{worldSpaceValue}{2}$

Note: refer to your sculpting software manual on how to get the displacement worldSpaceValue

5

ezehairDisplacement

To avoid shaves displacement map, and be able to use them also with maya Fur and Yeti this shader will be available soon!

5.1 hairDisplace (soon)

Although not recommend for obvious reasons, this shader might be handy for matching exactly the hair underlying geometry displacement.It has the same attributes as the common displace, but this one will displace the hair geometry on N_Srf and only update N_Srf, not N. Beware of using just the right amount of displacement bound to avoid a render overkill!

displacement

Overall displacement multiplier (usually you would leave this value at 1)

textureName

Displacement texture file

offset

Here you would usually put your displacement $-\frac{worldSpaceValue}{2}$

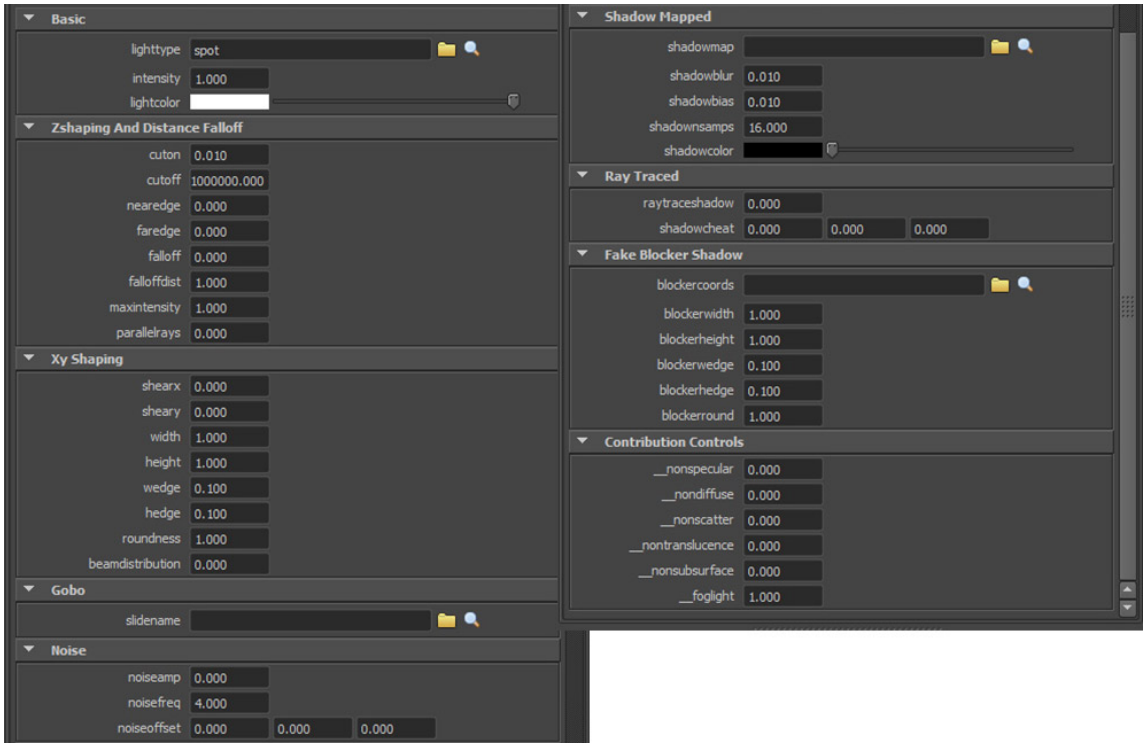
multiplier

Here you would usually put your displacement $\frac{worldSpaceValue}{2}$

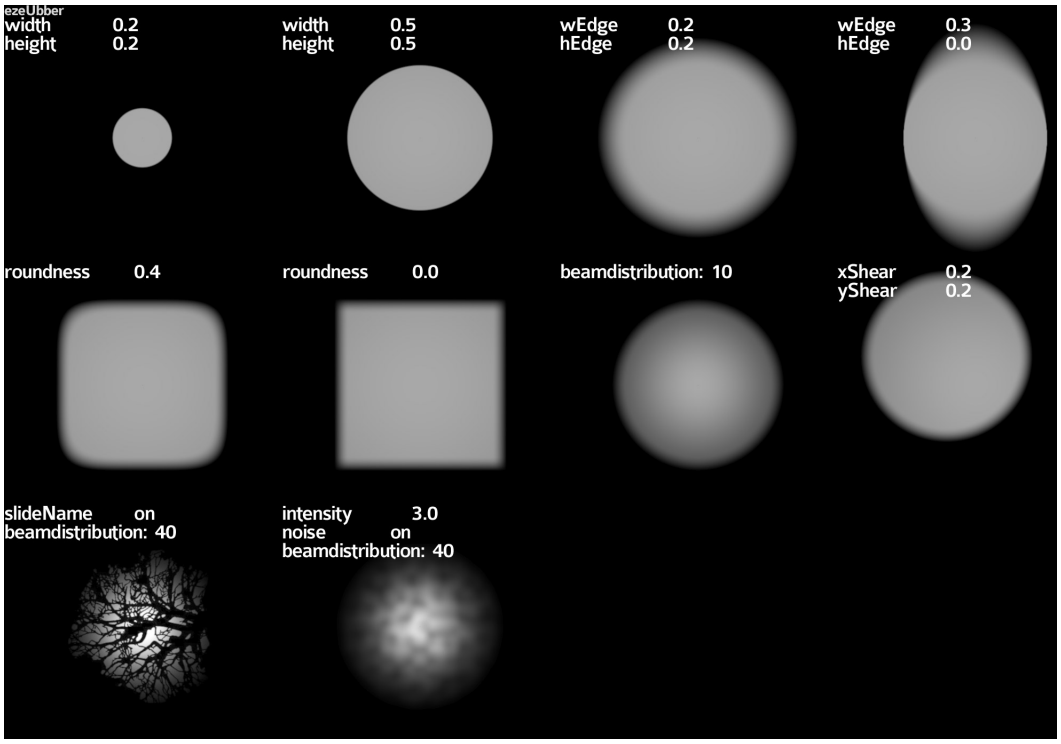
Note: refer to your sculpting software manual on how to get the maximum displacement Value in worldSpace

6

ezeUbber



In Figure 6, ezeHair shader parameters



In Figure 6, ezeHair shader parameters

6.1 Basic

lighttype

one of "spot", "omni", or "arealight". Spot lights are those that point in a particular direction (+z in local light space, for this light. Omni lights throw light in all directions. Area lights are emitted from actual geometry (this only works on BMRT area lights for the time being).

NOTE: when using spot type, mayas coneAngle value is discarded and width height parameters in the shader are used instead!

NOTE2: if a shadow map is to be used, you need to point to the map (see shadow Mapped subsection), add a attributes node to the light with shadow Maps attributes.

See maya tips for tokens or refer to the 3delight docs for a full list of tokens.

exposure

overall intensity scaling of the light

lightcolor

overall color filtering for the light

6.2 Zshaping and Distance fallOff

cuton, cutoff

define the depth range (z range from the origin, in light coordinates) over which the light is active. Outside this range, no energy is transmitted.

nearedge, faredge

define the width of the transition regions for the cuton and cutoff. The transitions will be smooth.

falloff

defines the exponent for falloff. A falloff of 0 (the default) indicates that the light is the same brightnessregardless of distance from the source.
Falloff==1 indicates linear (1r) falloff
falloff==2 indicates $1/r\hat{2}$ falloff (which is physically correct for point-like sources, but sometimes hard to use)

falloffdist

the distance at which the incident energy is actually equal to intensity*lightcolor.
In other words, the intensity is actually given by: I = (falloffdist / distance)
ˆ
falloff

maxintensity

to prevent the light from becoming unboundedly large when the distance ¡ falloffdist, the intensity is smoothly clamped to this maximum value.

parallelrays

when 0 (the default), the light appears to eminate from a single point (i.e. the rays diverge). When nonzero, the light rays are parallel, as if from an infinitely distant source (like the sun).

6.3 Zy Shaping

Shaping of the cross-section. The cross-section of the light cone is actually described by a superellipse with the following controls:

shearx, sheary

define the amount of shear applied to the light cone direction. Default is 0, meaning that the center of the light cone is aligned with the z axis in local light space

width, height

define the amount of shear applied to the light cone direction. Default is 0, meaning that the center of the light cone is aligned with the z axis in local light space

wedge, hedge

the amount of width and height edge fuzz, respectively. Values of 0 will make a sharp cutoff, larger values (up to 1) will make the edge softer.

roundness

controls how rounded the corners of the superellipse are. If this value is 0, the cross-section will be a perfect rectangle. If the value is 1, the cross-section

will be a perfect circle. In between values control the roundness of the corners in a fairly obvious way.

beamdistribution

controls intensity falloff due to angle. A value of 0 (the default) means no angle falloff. A value of 1 is roughly physically correct for a spotlight, and corresponds to a cosine falloff. For a BMRT area light, the cosine falloff happens automatically, so 0 is the right physical value to use. In either case, you may use larger values to make the spot more bright in the center than the outskirts. This parameter has no effect for omni lights.
Note:When using this parameter with a hair backLight,no matter the wrapIllum Angle of the ezeHair diffuse, It will draw the light useless!

6.4 gobo

Cookie or slide filter

slidename

if a filename is supplied, a texture lookup will be done and the light emitted from the source will be filtered by that color, much like a slide projector. If you want to make a texture map that simply blocks light, just make it black-and-white, but store it as a RGB texture. For simplicity, the shader assumes that the texture file will have at least three channels.

6.5 Noise

Projected noise on the light

noiseamp

amplitude of the noise. A value of 0 (the default) means not to use noise. Larger values increase the blotchiness of the projected noise

noisefreq

frequency of the noise.

noiseoffset

spatial offset of the noise. This can be animated, for example if you are using the noise to simulate the attenuation of light as it passes through a window with water drops dripping down it.

6.6 Shadow Mapped

shadows are mainly computed by shadow maps. Please consult the PRMan documentation for more information on the meanings of these parameters

shadowmap

the name of the texture containing the shadow map. If this value is "" (the default), no shadow map will be used.

shadowblur

how soft to make the shadow edge, expressed as a percentage of the width of the entire shadow map.

shadowbias

the amount of shadow bias to add to the lookup.

shadownsamps

the number of samples to use.

6.7 RayTraced

These options work only for BMRT: raytraceshadow - if nonzero, cast a ray to see if we are in shadow. The default is zero, i.e. not to try raytracing.

Ray-traced shadows

check this on when working in linear space

shadowcheat

add this offset to the light source position. This allows you to cause the shadows to eminate as if the light were someplace else, but without changing the area illuminated or the appearance of highlights, etc.

6.8 fake Blocker Shadow

"Fake" shadows from a blocker object. A blocker is a superellipse in 3-space which effectively blocks light. But it's not really geometry, the shader just does the intersection with the superellipse. The blocker is defined to lie on the x-y plane of its own coordinate system (which obviously needs to be defined in the RIB file using the CoordinateSystem command).

blockercoords

the name of the coordinate system that defines the local coordinates of the blocker. If this is "", it indicates that the shader should not use a blocker at all.

blockerwidth, blockerheight

define the dimensions of the blocker's superellipse shape.

blockerwedge, blockerhedge

define the fuzzyness of the edges.

blockerround

how round the corners of the blocker are (same control as the "roundness" parameter that affects the light cone shape.

shadowcolor

Shadows (i.e. those regions with "occlusion" as defined by any or all of the shadow map, ray cast, or blocker) don't actually have to block light. In fact, in this shader, shadowed regions actually just change the color of the light

to "shadowcolor". If this color is set to (0,0,0), it effectively blocks all light. But if you set it to, say (.25,.25,.25), it will make the shadowed regions lose their full brightness, but not go completely dark. Another use is if you are simulating sunlight: set the lightcolor to something yellowish, and make the shadowcolor dark but somewhat bluish. Another effect of shadows is to set the __nonspecular flag, so that the shadowed regions are lit only diffusely, without highlights.

6.9 Contribution controls

Allows you to control the effect amount a light will have on the shader.
0 means full amount, 0.2 = 80%, 0.9 = 10%, 1 = None,

__nonspecular

when set to 1, this light does not create specular highlights! The default is 0, which means it makes highlights just fine (except for regions in shadows, as explained above). This is very handy for lights that are meant to be fill lights, rather than key lights. NOTE: This depends on the surface shader looking for, and correctly acting upon, this parameter. The builtin functions diffuse(), specular() and phong() all do this, for PRMan 3.5 and later, as well as BMRT 2.3.5 and later. But if you write your own illuminance loops in your surface shader, you've got to account for it yourself The PRMan user manual explains how to do this.

__nondiffuse

the analog to __nonspecular, if this flag is set to 1, this light will only cast specular highlights, but not diffuse light. This is useful for making a light that only makes specular highlights, without affecting the rest of the illumination in the scene. All the same caveats apply with respect to the surface shader, as described above for __nonspecular

__nonScatter

scatter Effects, works for both ezeSurface and ezeHair

__nonTransluncence

only used in ezeHair

__nonSubsurface

subSurface effect, works for both ezeSurface and ezeHair

__nonSpecularMask

blends the specular masks, any value between 0-1 will have an effect on the mask transparency. See ezeSurface Specular masks section for more information on this

__foglight

the "noisysmoke" shader distributed with BMRT will add atmospheric scattering only for those lights that have this parameter set to 1 (the default). In other words, if you use this light with noisysmoke, you can set this flag to 0 to make a particular light *not* cause illumination in the fog. Note that the noisysmoke shader is distributed with BMRT, but will also work just fine with PRMan (3.7 or later).

7

ezeBake

Although ezeSurface contains all the baking parameters and handles the baking skipping complex lighting calculation, this shader is supplied for the sake of having a separated shader to fulfil all your baking needs.

7.1 Bake

PtcFile

ptc file to bake

bakeSpace

Coordinate System in which to bake the points. Refer to 3delight docs for coord system help.

radius Scale

Scale multiplier for the microPolygon radii, usefull when baking lower shading rate ptcs. Refer to 3delight docs for more on this subject.

8

ezeEnvironment

In Figure 8, ezeHair hairBall and ezeSurface ball iluminated by 1 ezeEnvLight latitude longitude hdri map (notice the hair contributing to the occlusion).

In Figure 8, ezeSurface iluminated by a ezeEnvLight latitude longitude hdri map (generated with tdlmake default values) with occlusion shadows (ptc based), subsurface, ward anisotropic specs and 2 ezeUbbers.

8.1 Multipliers

Ligth Color

Multiplier for the envMap

Saturation

saturation multiplier for the envMap

KEnv

exposure control for the envMap

8.2 maps

envMap

is the map for the env, it should be a lat/long map in 32Bits float linear space. Please note that this map is necessary and if not supplied the light will not work.

envSpace

world

samples

number of samples for convultion, set to 0 to use ?????????

maxDist

bias

coordSyst

Use World or Create a 3delight coordSystem and set the shape name here to be able to rotate the map.

8.3 occlusion

KOcc

turns on/off the occlusion calculation

Filename

point to the scene ptc file for the occlusion

hitSides

options: front/back/both, usually you want to keep this setting to both

maxDist

maximum search distance for surrounding light occluding geometry, if greater than this distance the environment map kicks in

int

intensity of the occlusion shadow, 0 means pure shadows are black, 0.5 means middle gray, and 1 is white (same effect as turning it off)

coneAnle

the search cone angle from the sampled point

falloff

0 or 1, turns it on and off

falloffMode

1 linear, 2 quadratic., 3 square

bias

offset the sample point from the geometry along the normal, usually helpful when using displacements

clamp

this parameter should be on by default, avoids overshadowing.

maxSolidAngle

a quality/speed parameter, 0.04 should be a good start point (0.2 will speed things up at quality expense, 0.01 will be better looking but slower)

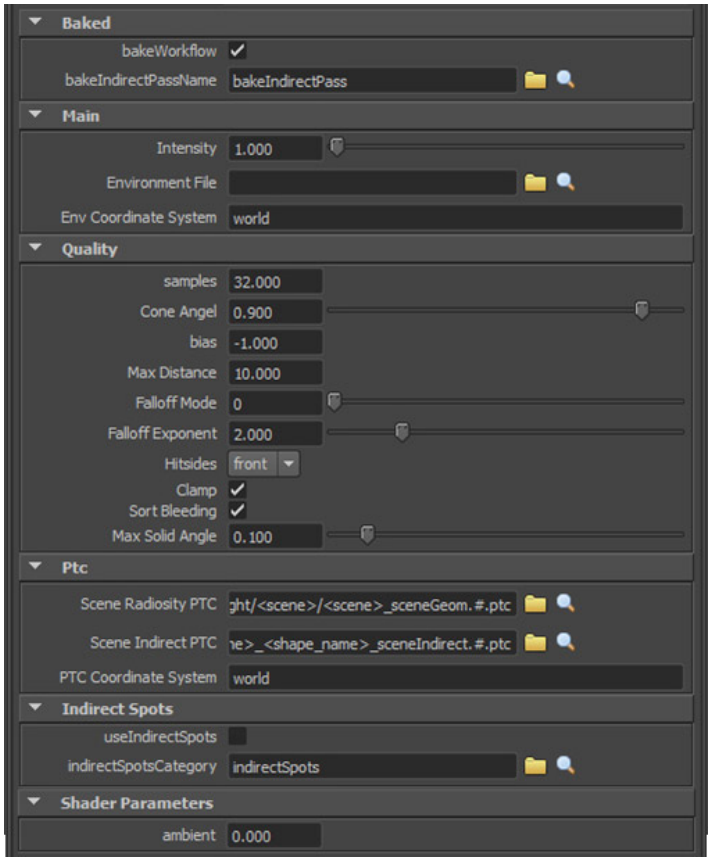
8.4 Category

__category

a string with the category name this light belongs to. envLight is the default value as the hair shader will treat this lights differently and turns off occlusion (Kocc) when occlusion baked maps are used.

9

ezeIndirect



In Figure 9, ezeIndirect parameters

9.1 baked

bakeWorkFlow

This is the Recommended way to do indirect Diffuse. If On, it will pre process a ptc file with indirect lighting information and sample the lighting from there in the final pass. If Off it will do oldSchool raytrace type indirect lighting, feeding from the Radiosity PTC file.

bakeIndirectPassName

If bakeWorkFlow is on, this passName will be compared to the current one to switch betweend baking and reading modes.

9.2 Main

Intensity

Intensity multiplier for the light effect, this multiplier does not affect the baked light intensity, only the readed from ptc and on the fly.

environmentFile

(this parameter is obsolete at this time and will be removed) to use as a background sample, leave this blank if using ezeEnvironment

9.3 Quality

Samples

turns on/off the occlusion calculation

ConeAngle

bias

offset the sample point from the geometry along the normal, usually helpful when using displacements

maxDist

maximum search distance for surrounding light bleeding.

falloff

0 or 1, turns it on and off

falloffExponent

1 linear, 2 quadratic., 3 square

hitSides

options: front/back/both, usually you want to keep this setting to both

clamp

this parameter should be on by default, avoids overshadowing.

sortBleeding

Will sort geometry from the sampled point to calculate indirect light occlusion from other geometry.

maxSolidAngle

a quality/speed parameter, 0.04 should be a good start point (0.2 will speed things up at quality expense, 0.01 will be better looking but slower)

9.4 PTC

Only used when bakedWorkFlow is selected

scene Radiosity PTC

Specify here the ptc contaning the micro polygon size/radiosity channels to use for indirect calculations.

scene Indirect PTC

If bakeWorkFlow is On, this file will be created when in the bakeIndirectPass name specified and read on the finalRenderPass.

PTC coordinate system

world

9.5 indirect Spots

Not implemented yet, This parameter actives the spotlight isolation for the indirect effect.

10

uDim reference

All texture map paramters can use a common texture map, or uv spanned textures like the ones mari does when working with different uvspaces.

10.1 parameters

on/off

This will override the usage of the texture map and use the uDim instead

texName

This is the actual path+filename without uv space coord, frameNumber, nor extension.

diffuse Maximun U

This is the number of uv spaces to use accross u, if reached u will be 0 again and v=(v+1), see example images.

diffuse frame number

useVariant

Your file name may be something like this fileName_variant . uvSpace N. ext, where variant is a version, a type, or a resolution count of the file.

variant name

Here you can specify the variant name to use if needed. for Ie "2k" (diffuse_2k.0001.tdl), "8k" (diffuse_8k.0001.tdl), or even a date like "20120107" (specular_20120107.0001.tdl)

useAnimMap

Inserts the frame number with a 4 digit padding between the complete file name and the extension like diffuse_8k.0054.0001.tdl

diffuseFileExt

file extension to be used, either tdl of tiff. Remember tiff files can have a dramatic impact on your render times specially with big files! i mean seriously dramatic.

11

Maya quick start setup

Some Object sets are suitable for speeding up rendering of the passes. Not required though.

The passes you will need to quickly get started are:

Shadow pass

11.1 Passes

Shadow pass

one line	another column
second line here	and here

Object Sets	pass_ShadowGeo	all geometry you want it to cast shadows
	pass_shadowLights	all lights you want to calculate shadows from
render Settings		Basic pass, + this attributes: -display sets: geo and lights -render shadows: On -use displacement shaders: On -full volumetric shaders: On -export all AOVS: On

Bake Pass

Object Sets	pass_bakeGeo	all geometry you want it to occlude ambient lighting
Shader collection overrides	bakeGeoAttributes ezeBake shader	-two sided -raster oriented dice Off
render Settings		Basic pass, + this attributes: -display sets: geo and lights -render shadows: Off -use displacement shaders: On -full volumetric shaders: On -export all AOVS: On

indirectBakePass

Object Sets	pass_bakeIndirectGeo	all geometry you want to have indirect illuminationthis should NOT include the hair. The hair will sample this 3d texture bake from the strand root position in 3d pace to get the correct values
Shader collection overrides	bakeGeoAttributes ezeBake shader	-two sided -raster oriented dice Off
render Settings		Basic pass, + this attributes: -display sets: geo and lights -render shadows: Off -use displacement shaders: On -full volumetric shaders: On -export all AOVS: On

fullRendePass

Object Sets	pass_fullRenderGeo	all geometry you want to render
	pass_fullRenderLights	all lights you want to be included while render
Geometry Attributes		hairGeoAttributes -dice type Hair -primitive Hair -shading rate: 40 (this will speed up renders when sampling, as micropolygon dicing is not needed for hair primitives -raster oriented dice Off geometryAttributes: —
render Settings		Full pass with this settings: renderShadows: Off

12

Maya tips

12.1 ezeUber maya visualization plugin

Check this python plugin to interactively see the uberLight parameters in the viewport
http://www.creativecrash.com/maya/downloads/scripts-plugins/rendering/renderman/c/uberlighthelper

12.2 file path expressions

most used tokens

Refer to File Path Expressions section in 3delight for maya help for complete token listing

$VAR or %VAR%	The value of the environment variable VAR will replace $VAR in the path.
@	The current frame number will replace the @ character.
#	The current frame number, padded to form a 4 digits number, will replace the # character.
‘melCommand‘	This string will be replaced with the path to the current Maya project directory.
<project>	This string will be replaced with the path to the current Maya project directory.
<pass>	This string will be replaced with the name of the render pass node used for rendering
<scene>	This string will be replaced with the name of the scene
<camera>	This string will be replaced with the name of the camera shape being used for rendering
<aov>	This string will be replaced with the name of the variable being output in the display. Only valid in render pass display filename attributes

References

[1] blackbody. Based on Mitchell Charitys rgb blackbody color mappings
http://www.vendian.org/mncharity/dir3/blackbody/
http://www.vendian.org/mncharity/dir3/blackbody/intensity.html
http://www.vendian.org/mncharity/dir3/blackbody/UnstableURLs/bbr_color.html

[2] uDim. Code based on William J. Earl
http://earlyworm.org/2011/udim-constant-shader/

[3] hair specular and scatter Code based Sachin Shrestha from 3delight forums

[4] surface specular masks Code rewritten but based on Charles Trippe ct_surf_ceramic shader

[5] ezeUber light based on Larry Gritz uberLight shader

other Resources

[6] http://www.renderman.org/RMR/Publications/sig06.course25.pdf