Eze ShaderSuite

Contents

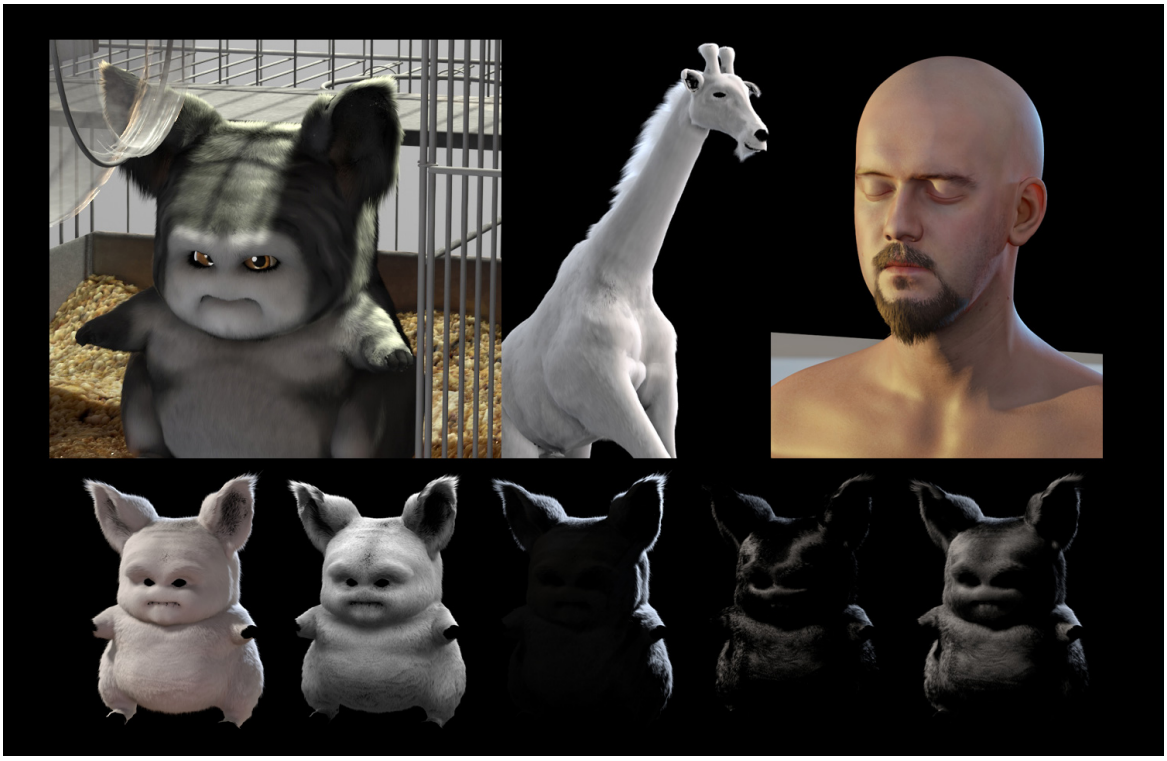
1

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

-Full raytrace shader
-marchsner hair diffuse model

-3delight studio 11 raytrace environmental lights, occlusion and indirect diffuse ready
-2 legacy specular layers (artist friendlier)
-extensive root/tips controls including color, darkening, alpha and ambient and maps for each
-all maps slots support uDim maps type

1.3 compiling

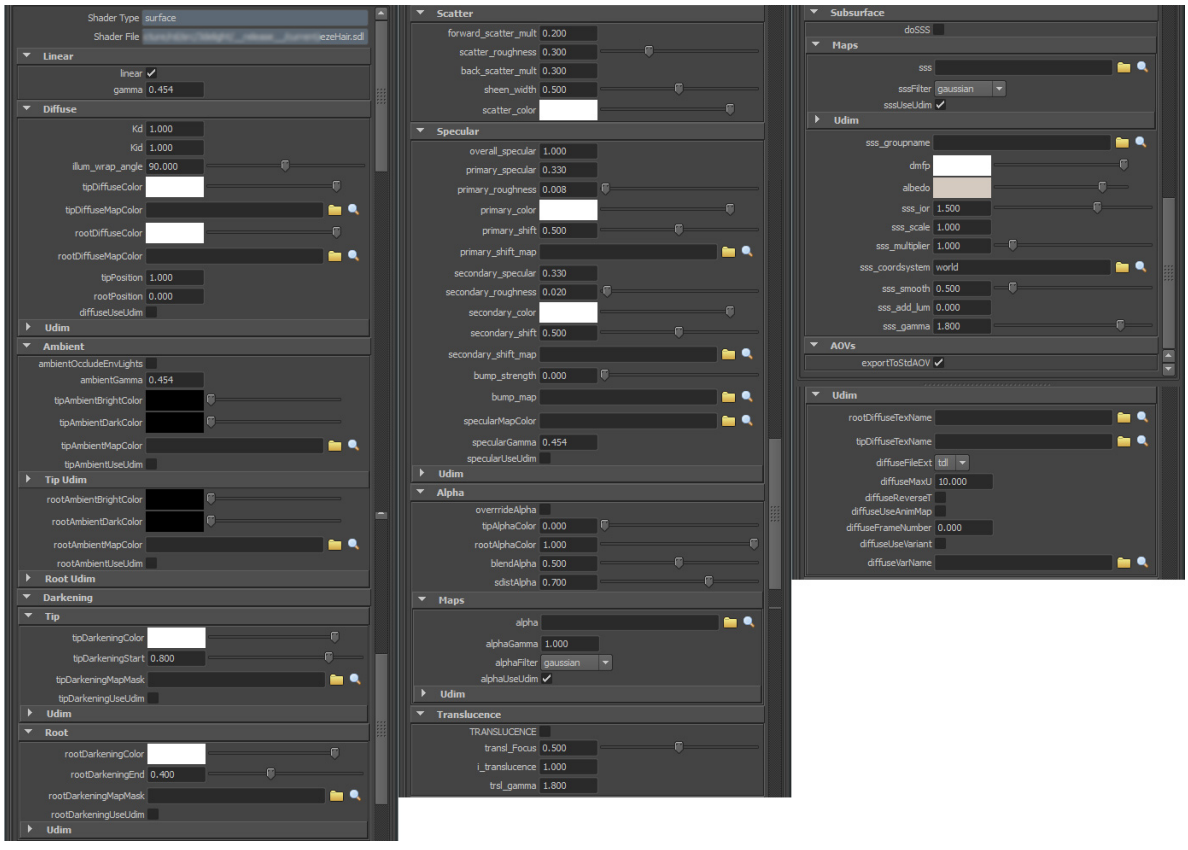
1.3.1 all platforms

-shaderdl -I¡directory¿ ¡shaderName¿.sl

In directory, supply your $3DELIGHT/maya/rsl directory, where global_illumination.h and fluid_utils.h are found

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 the direct lighting multiplier.

Ki

the indirect diffuse lighting multiplier (any light contributing to the indirectDiffuse, and environments).

i_color

XXX

i_thickness

hair thickness

i_tint

environment lighting tint

i_samples

trace samples for environment lighting

i_xxx_r (roughness)

shift of the primary specular toward the the root.

i_xxx_tt (forward scattering)

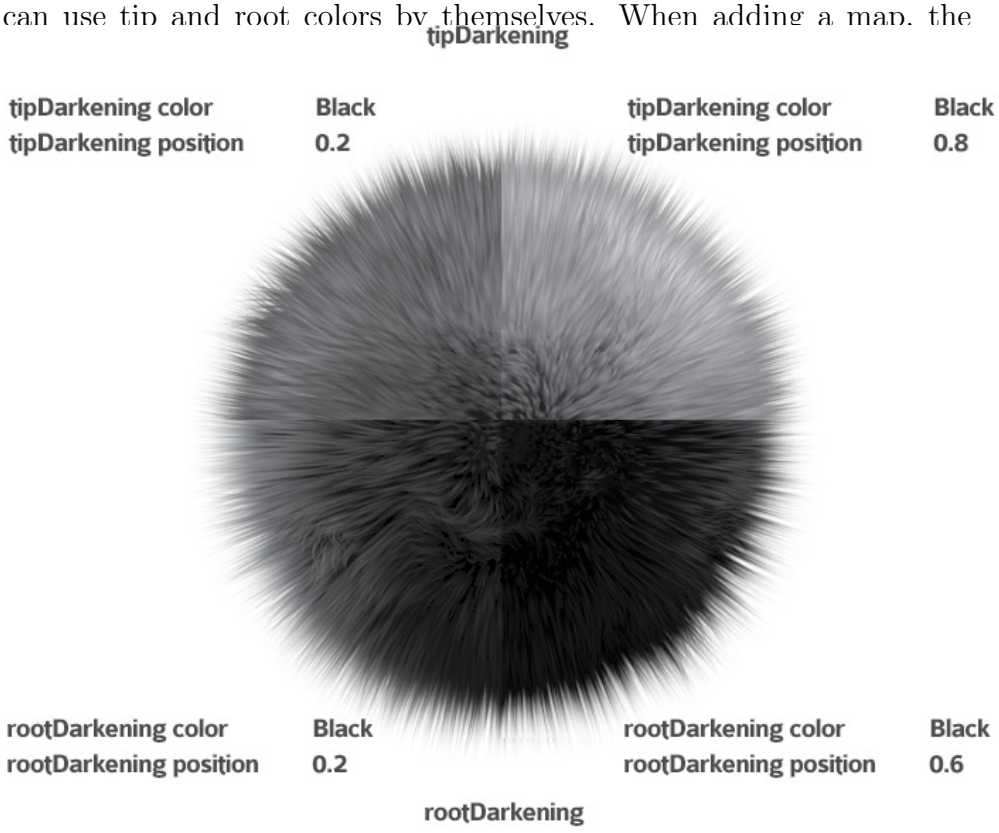
a strong forward scattering component from the light coloured hair. This causes blond, brown, gray and white hair to look very bright when lit from behind

i_xxx_r (TRT)

-coloured secondary peak shifter toward the top from the white primary specular peak. In a head of hair this leads to the secondary highlight that is visible just above the primary, sometimes appearing more as a coloured fringe on the primary than a separate feature

Tip and root Colors, multipliers, controls and masks

You can use tip and root colors by themselves. When adding a map, the ma
and
the
tip
If y
whi

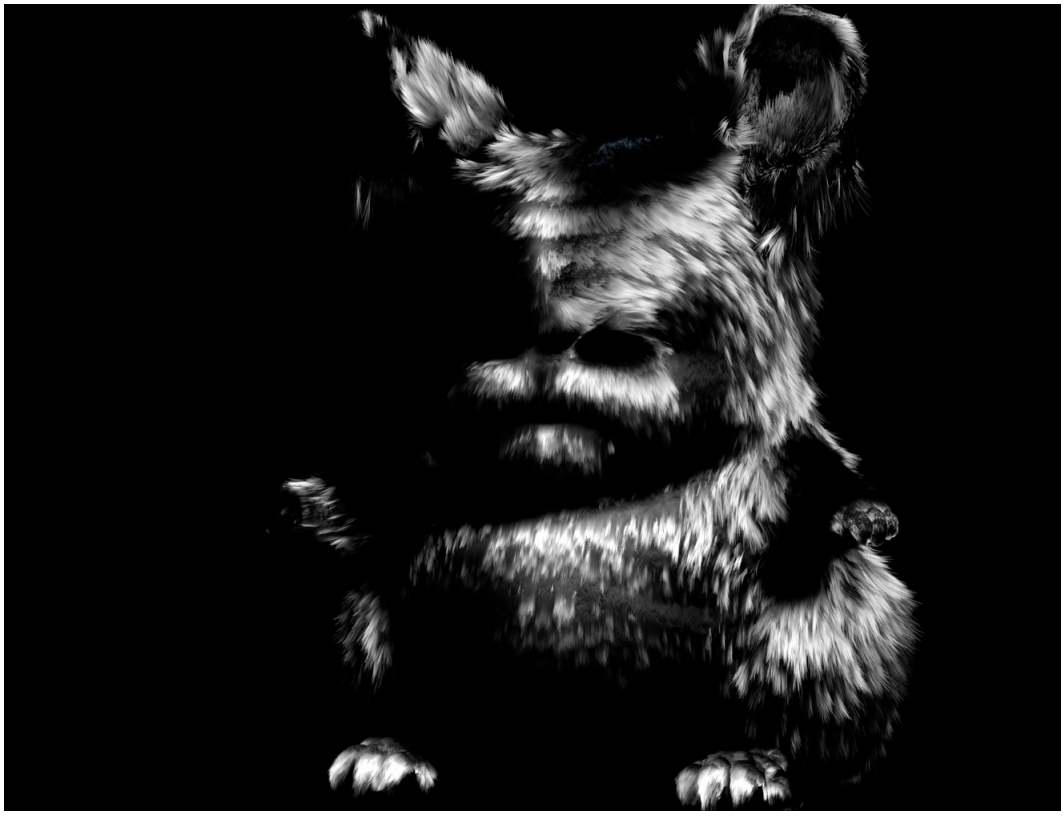


In Figure 2.2, ezeHair shader parameters.

2.3 legacy specular

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 Get the maya sample scenes file from the website to get the shift map that you can use for the specular shift.



In Figure 2.3, 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.

2.4 alpha

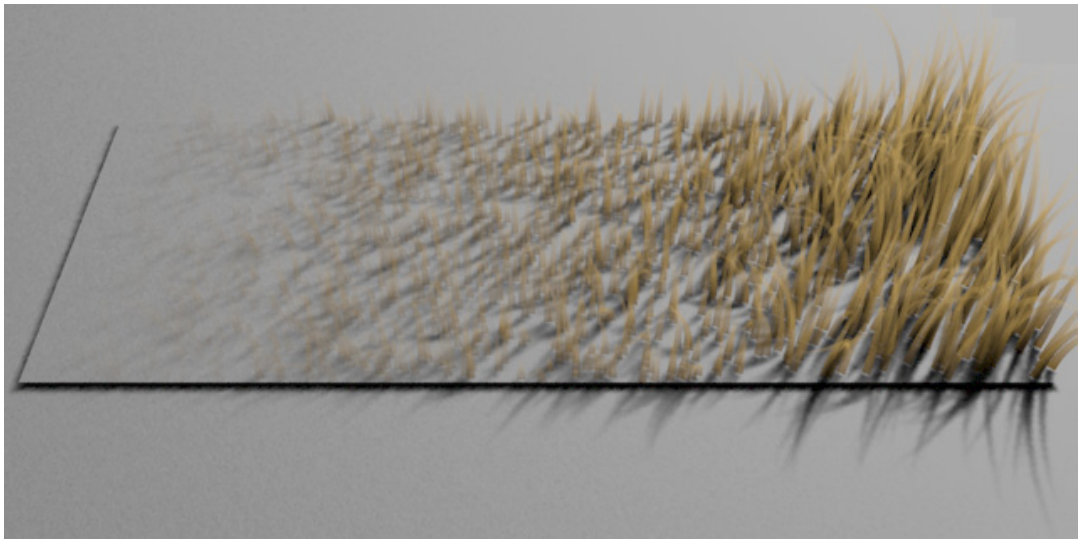
If using full tracing, leave the default parameters (override geometry opacity On, and opacity 1 in root and tip for perfomance!!)
Either you overriding the Oi, using shaves root/tip transparency, or using a map remember to check Use Surface Shaders in shadow pass!
Note:ezeShaders will skip any lighting/sss 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 2.4, ezeHair shader using alpha overrides and a alpha gradient black & white map_hair_translucence.

2.5 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.

2.5.1 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).

2.5.2 aov listing

output varying color aov_hair_diffuse
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_hair_id
output varying color aov_indirect

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

ezeDisplacement

In Figure 3, ezeDisplacement used to mimic short fur

3.1 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

4

ezehairDisplacement

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

4.1 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 $-\frac{worldSpaceValue}{2}$

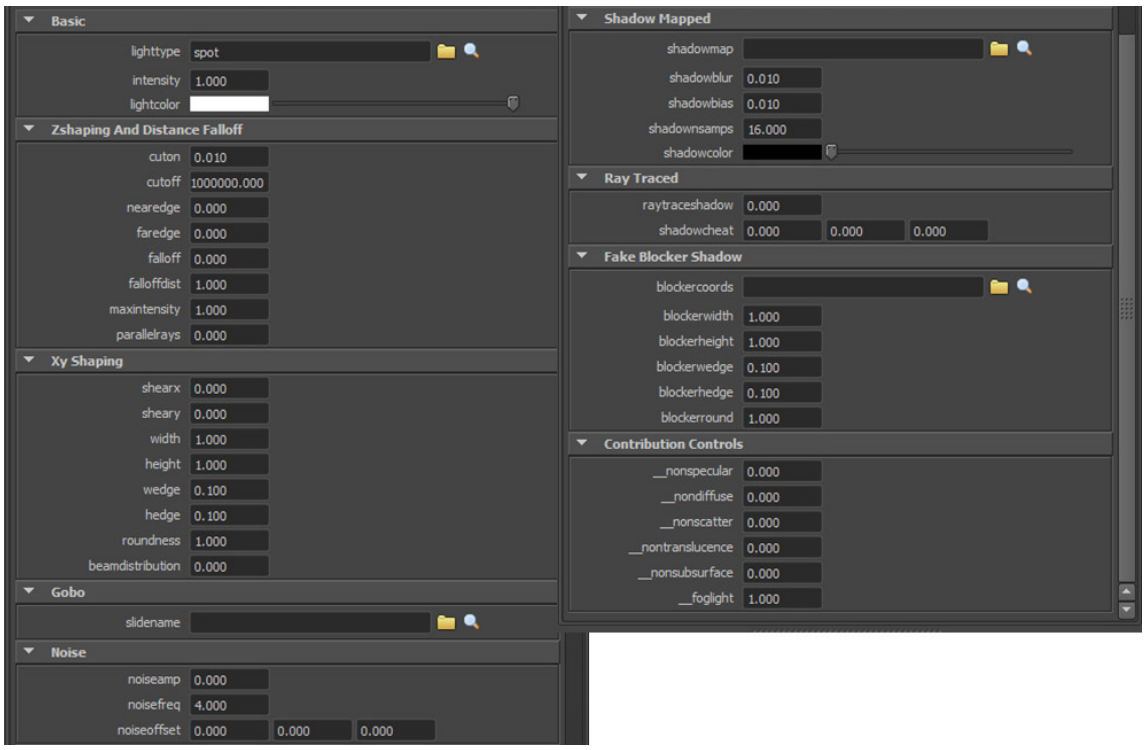
multiplier

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

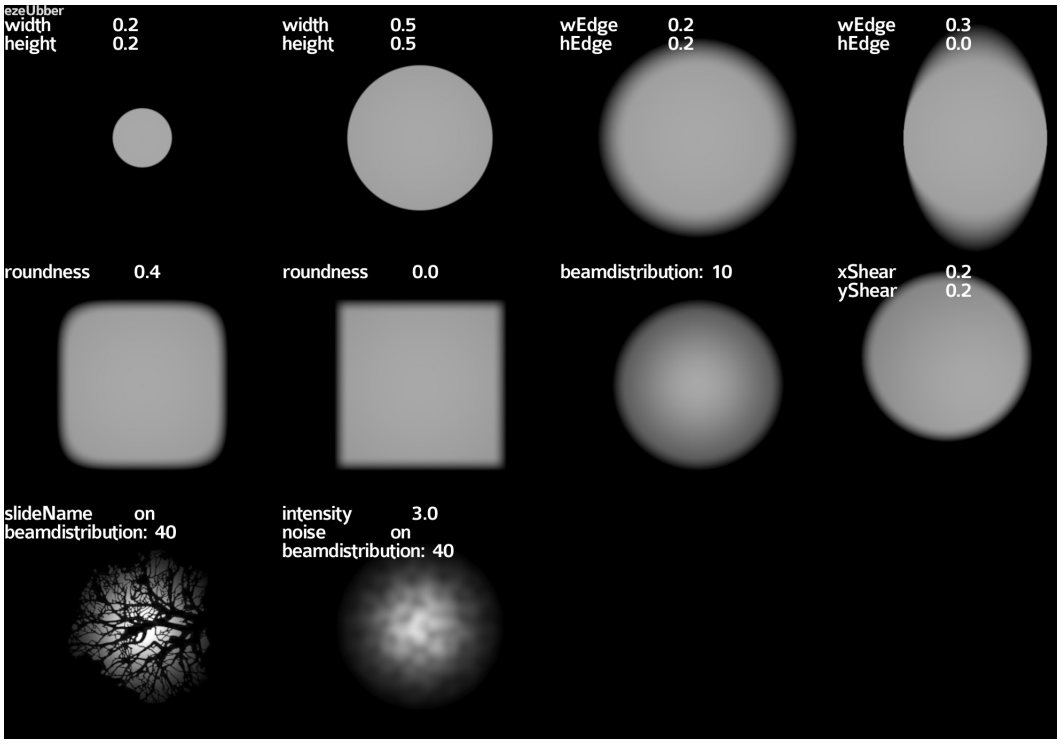
Note: refer to your sculpting software manual on how to get the maximum displacement Value in worldSpace

5

ezeUbber



In Figure 5, ezeHair shader parameters



In Figure 5, ezeHair shader parameters

5.1 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

5.2 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).

5.3 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!

5.4 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.

5.5 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.

5.6 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.

5.7 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.

5.8 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.

5.9 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).

6

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.

6.1 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.

7

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

7.1 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

8

Maya tips

8.1 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

8.2 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 renderi
<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 displ 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