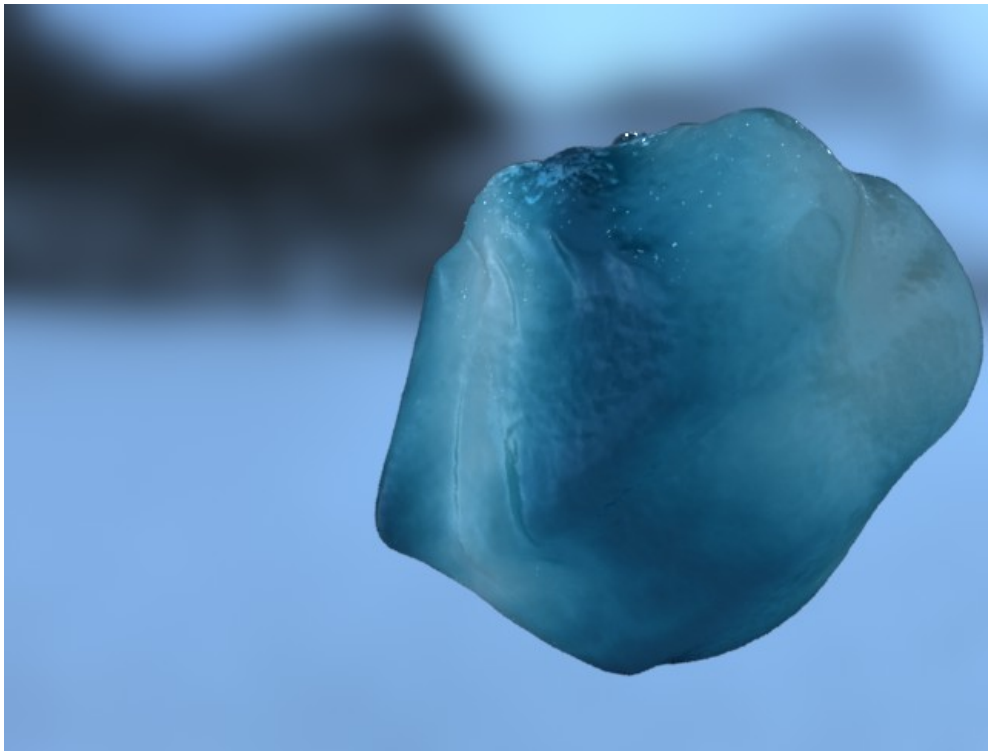
# Mr Cool Ice Shader Help File

**Vanessa Salas Castillo**
**i7832424**



## i. Summary

“**mr_cool_ice.sl**” is a Renderman shader that renders deep blue ice material using raytracing.

“**mr_cool_ice**” replicates the main characteristics of blue ice: scattering of light through the volume, reflections, refractions, fresnel effect, specularity.

The shader is completely procedural, it does not use painted textures. I offers you the possibility of using noises to control its patterns.

Ray tracing is used, however if the settings are optimised, it is a very efficient shader.

## ii. Pieces

As “**mr_cool_ice**” uses subsurface scattering, a previous pass is needed. This pass bakes the radiance of your model and creates an albedo noise to simulate the lack of uniformity of blue ice.

Two options are offered:

**1.** Baking the albedo, diffuse mean free path, radiance and the area of every micropolygon on a point cloud with “**bake_radiance_noise.sl**”, and then ecified in Renderman Application Notes for Translucency and Subsurface Scattering [PX]. This process has been automatised with a python script named that does the call for all these steps. This script can render more than one frame as well.using *ptfilter* and *brickmake* as sp

After this, “**mr_cool_ice**” is called with the resulting brickmap as a parameter. As the diffuse simulation (see Renderman Application Notes for Translucency and Subsurface Scattering [PX]) has been done, the shader just reads the result from the brickmap.

**2.** Baking the area and transmitted direct illumination at each point on a point cloud with “**bake_info.sl**”. Though not essential, it is recommended to generate an organised point cloud out of the resultant point cloud. This point cloud is a parameter for “**mr_cool_ice_info.sl**”.

ATTENTION: for both of the prepasses, it is important to remember to set the attributes “cull” “hidden” and ”cull” “backfacing” to 0 in the rib file.

## iii. “mr_cool_ice”

### 1. Parameters

**Coefficients to modulate the effect of each of the components of the shader**

| Name | Type | Function |
|---|---|---|
| Ko | float | Opacity coefficient |
| Ksss | float | Subsurface scattering coefficient |
| Ks | float | Specularity coefficient |
| Kwet | float | Specularity coefficient for wet effect |
| Kr | float | Reflection coefficient |
| Kt | float | Refraction coefficient |
| Kibl | float | Image based lighting coefficient. Env. Diffuse |
| Kd | float | Diffuse coefficient (for frost) |
| Kb | float | Displacement coefficient (for frost) |

**Global parameters**

| Name | Type | Function |
|---|---|---|
| env_map | string | Environment map for the image based lighting calculations. |
| diffuse_map | string | Blurred map for the image based lighting calculations. |
| coord_system | string | Coordinate system to do the environment look ups. |
| premult | float | If its value is one, layers are multiplied by the opacity |

**Specularity parameters**
This parameters control the behaviour of a Cook Torrance function to model the specularity of ice.

| Name | Type | Function |
|---|---|---|
| spec_flag | float | 1: Compute specularity |
| cook_roughness | float | Roughness of the surface |
| cook_specular_colour | color | Colour of the specular |
| gaussConstant | float | Gauss constant |

**Wet effect parameters**
This parameters control a wet effect produced with a Blinn specularity.

| Name | Type | Function |
|---|---|---|
| wet_flag | float | 1: Add wet effect |
| wet_eccentricity | float | |
| wet_rolloff | float | |
| wet_reflectivity | float | |
| spec_colour | color | Colour of the specular |

**Fresnel effect parameters**
Parameters to activate and control the Fresnel effect. When active, automatically affects the reflections and refractions calculated in the shader.

| Name | Type | Function |
|---|---|---|
| fresnel_flag | float | 1: do fresnel effect |
| ior | float | Index of refraction. The actual index of refraction of ice is 1.31 |

**Reflection oclussion parameters**
This is an image based lighting pass. It uses the environment map provided under “env_map”. This is the most expensive pass. Set carefully the settings of the samples. Using values that are multiples of 4 is more efficient.

| Name | Type | Function |
|---|---|---|
| rocc_flag | float | 1: Compute reflection occlusion pass. |
| reflectionocc_samples | float | Number of samples. Acts as a quality knob. |
| reflectionocc_blur | float | This value is later transformed to radians an used in the gather call as the cone angle. |

**Refraction parameters**
If computed uses a trace call and Fresnel if it has been activated.

| Name | Type | Function |
|---|---|---|
| t_flag | float | 1: Compute refraction. |

**Subsurface Scattering parameters**
The subsurface scattering pass gives translucency and most of its colour to ice. A brickmap with the result of the diffuse approximation done with *ptfilter* must be provided.

| Name | Type | Function |
|---|---|---|
| sss_flag | float | 1: Read subsurface scattering |
| sss_bkm | string | Name of the brickmap. |

**Frost parameters**
These are parameters to add frost to the superior part of the ice.

| Name | Type | Function |
|---|---|---|
| snow_flag | float | 1: Add frost to the ice |
| snow_height | float | Defines how low can the snow spread on the volume. This value exists, because icebergs are 80% underwater. It doensn't make sense that icebergs are covered of snow underwater. |
| snow_over | float | How horizontal should the surface be to be covered by snow. |
| spark_eccentricity | float | Blinn eccentricity parameter |
| spark_rolloff | float | Blinn falloff parameter |
| | | |

**Diffuse IBL and Ambient occlusion parameters**
These parameter is used to modulate the result of the snow, not for the ice.

| Name | Type | Function |
|---|---|---|
| ibldiff_flag | float | 1: Do image based diffuse |
| ibldiff_samples | float | Defines how low can the snow spread on the volume. This value exists, because icebergs are 80% underwater. It doensn't make sense that icebergs are covered of snow underwater. |
| occ_sample | float | Samples quantity to create the ambient occlusion pass. |
| occ_maxvar | float | |

### 2. Outputs

The pre pass shader and the actual shader, both produce “arbitrary output values” or AOVs. There is an AOV acting as a secondary channel for every different pass. Each of these secondary channels has to be made available in the RIB file first. By using this strategy, even though a single shader does all the calculations, it is still possible to have access to the different passes. Afterwards manipulate them inside a compositing tool as desired and achieve different results is quite easy. These secondary outputs are saved as 'tif' images. The final image is saved as a 'tif' as well.

| Type | Name |
|---|---|
| output varying color | _diffuse_colour |
| output varying color | _diffuse_ibl |
| output varying color | _opacity |
| output varying color | _occlusion |
| output varying color | _reflection_occlusion |
| output varying color | _refraction |
| output varying color | _snow_mask |
| output varying color | _sparkles |
| output varying color | _specularity |
| output varying color | _specularity_blinn |
| output varying color | _subsurface |

## iv. “mr_cool_ice_info”

### 1. Parameters

**Coefficients to modulate the effect of each of the components of the shader**

| Name | Type | Function |
|---|---|---|
| Ko | float | Opacity coefficient |
| Ksss | float | Subsurface scattering coefficient |
| Ks | float | Specularity coefficient |
| Kwet | float | Specularity coefficient for wet effect |
| Kr | float | Reflection coefficient |
| Kt | float | Refraction coefficient |

| | | |
|---|---|---|
| Kibl | float | Image based lighting coefficient. Env. Diffuse |
| Kd | float | Diffuse coefficient (for frost) |
| Kb | float | Displacement coefficient (for frost) |

### Global parameters

| Name | Type | Function |
|---|---|---|
| env_map | string | Environment map for the image based lighting calculations. |
| diffuse_map | string | Blurred map for the image based lighting calculations. |
| coord_system | string | Coordinate system to do the environment look ups. |
| premult | float | If its value is one, layers are multiplied by the opacity |

### Subsurface Scattering parameters

The subsurface scattering pass gives translucency and most of its colour to ice. A brickmap with the result of the diffuse approximation done with *ptfilter* must be provided.

| Name | Type | Function |
|---|---|---|
| sss_flag | float | 1: Read subsurface scattering |
| sss_pointcloud | string | Name of the point cloud. |
| sss_ior | string | Index of refraction |
| albedo_value | color | A colour to multiply the value of the albedo |
| dmfp_value | color | The value of the diffuse mean free path |
| sss_smooth | float | Activate smooth results for subsurface scattering |
| sss_unitlength | float | The unit length by default is mm |
| sss_maxsolidangle | float | The accuracy of the subsurface diffusion simulation. |

### Albedo parameters

Through a varying albedo there are many visually interesting options. Marble noise is used alter it

| Name | Type | Function |
|---|---|---|
| veining | float | 1: Read subsurface scattering |
| ice_tint | string | Name of the point cloud. |
| vein_colour | string | Index of refraction |
| albedo_value | color | A colour to multiply the value of the albedo |

| | | |
|---|---|---|
| dmfp_value | color | The value of the diffuse mean free path |
| sss_smooth | float | Activate smooth results for subsurface scattering |
| sss_unitlength | float | The unit length by default is mm |
| sss_maxsolidangle | float | The accuracy of the subsurface diffusion simulation. |

**Specularity parameters**
This parameters control the behaviour of a Cook Torrance function to model the specularity of ice.

| Name | Type | Function |
|---|---|---|
| spec_flag | float | 1: Compute specularity |
| cook_roughness | float | Roughness of the surface |
| cook_specular_colour | color | Colour of the specular |
| gaussConstant | float | Gauss constant |

**Wet effect parameters**
This parameters control a wet effect produced with a Blinn specularity.

| Name | Type | Function |
|---|---|---|
| wet_flag | float | 1: Add wet effect |
| wet_eccentricity | float | |
| wet_rolloff | float | |
| wet_reflectivity | float | |
| spec_colour | color | Colour of the specular |

**Fresnel effect parameters**
Parameters to activate and control the Fresnel effect. When active, automatically affects the reflections and refractions calculated in the shader.

| Name | Type | Function |
|---|---|---|
| fresnel_flag | float | 1: do fresnel effect |
| ior | float | Index of refraction. The actual index of refraction of ice is 1.31 |

**Reflection oclussion parameters**

This is an image based lighting pass. It uses the environment map provided under “env_map”. This is the most expensive pass. Set carefully the settings of the samples. Using values that are multiples of 4 is more efficient.

| Name | Type | Function |
|---|---|---|
| rocc_flag | float | 1: Compute reflection occlusion pass. |
| reflectionocc_samples | float | Number of samples. Acts as a quality knob. |
| reflectionocc_blur | float | This value is later transformed to radians an used in the gather call as the cone angle. |

**Refraction parameters**
If computed uses a trace call and Fresnel if it has been activated.

| Name | Type | Function |
|---|---|---|
| t_flag | float | 1: Compute refraction. |

**Frost parameters**
These are parameters to add frost to the superior part of the ice.

| Name | Type | Function |
|---|---|---|
| snow_flag | float | 1: Add frost to the ice |
| snow_height | float | Defines how low can the snow spread on the volume. This value exists, because icebergs are 80% underwater. It doensn't make sense that icebergs are covered of snow underwater. |
| snow_over | float | How horizontal should the surface be to be covered by snow. |
| spark_eccentricity | float | Blinn eccentricity parameter |
| spark_rolloff | float | Blinn falloff parameter |
| | | |

**Diffuse IBL and Ambient occlusion parameters**
These parameter is used to modulate the result of the snow, not for the ice.

| Name | Type | Function |
|---|---|---|
| ibldiff_flag | float | 1: Do image based diffuse |
| ibldiff_samples | float | Defines how low can the snow spread on the volume. This value exists, because icebergs are |

| | | |
|---|---|---|
| | | 80% underwater. It doensn't make sense that icebergs are covered of snow underwater. |
| occ_sample | float | Samples quantity to create the ambient occlusion pass. |
| occ_maxvar | float | |

### 2. Outputs

The pre pass shader and the actual shader, both produce “arbitrary output values” or AOVs. There is an AOV acting as a secondary channel for every different pass. Each of these secondary channels has to be made available in the RIB file first. By using this strategy, even though a single shader does all the calculations, it is still possible to have access to the different passes. Afterwards manipulate them inside a compositing tool as desired and achieve different results is quite easy. These secondary outputs are saved as 'tif' images. The final image is saved as a 'tif' as well.

| Type | Name |
|---|---|
| output varying color | _albedo_noise |
| output varying color | _diffuse_colour |
| output varying color | _diffuse_ibl |
| output varying color | _opacity |
| output varying color | _occlusion |
| output varying color | _reflection_occlusion |
| output varying color | _refraction |
| output varying color | _snow_mask |
| output varying color | _sparkles |
| output varying color | _specularity |
| output varying color | _specularity_blinn |
| output varying color | _subsurface |

## v. Dependencies

To be able to compile the shaders, the following files are necessary:

vsl.utils: A library with utility functions is used with the purpose of storing a variety of noise functions, conversions, and any other method that provides reusable functionalities. Some of the noises and other functions were taken and slightly modified from the ones implemented by Slim.

The **“noises.h”** library from Advanced Renderman [AP] is also included.

**“vsl_pass.h”**: This library contains the functions that create the passes needed by each shader.

blinn.h
filterwidth.h

## vi.References

[PX] Pixar Animation Studios. 2009. *Pixar's Renderman User's Manual*. Available from: http://nccastaff.bournemouth.ac.uk/jmacey/Renderman/prmandocs/RPS_14.0/ (Accessed 12 May 2009).