Realistic Rendering of Blue Ice

Masters Project

Vanessa Salas Castillo
i7832424

i. Table of Contents

Table of Contents

ii. Introduction

Blue ice icebergs are captivating and visually attractive. Many photographers have captured beautiful and dramatic images of them in different places of the globe. The intriguing blue colour of iceberg is due to the fact that they act as filters that absorb light from the red end of the spectre.

The aim of this work is to replicate the appearance of blue ice icebergs. The most important characteristics that are wanted to simulate are the translucency of blue ice and its inner variations. Other important and distinctive properties that will be modelled are reflectiveness, refraction and specularity.

The problem addressed in this project is efficient shading and rendering of realistic blue ice. The problem is approached with a subsurface scattering based solution.

iii. Technical Background

1. Blue ice

Ice is a very weak absorber of light. As a consequence, when sunlight passes through a thin piece of ice, the volume looks almost white.

However, a block of ice thicker than one metre appears blue to the eye [DU]. Ice can be thought of as a filter that gradually absorbs red, and in a minor degree, green light. The absorption of red light is six times bigger than that of blue light. Blue light passes through untouched. Pure ice is inherently blue because its wavelength of minimum absorption is 470 nanometres. This is illustrated in Fig 1.

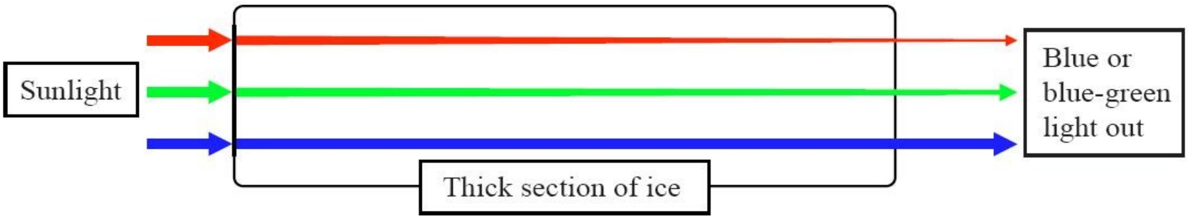

Fig 1
Image by General Atomics Sciences Education Foundation 1999 (DR)

Air bubbles inside the ice volume scatter light in different directions as portrayed in Fig 2. When ice has a very high concentration of air bubbles it looks white and opaque. This is because when the light goes through the ice and finds a bubble, the light is scattered back. Therefore, not much light is absorbed by the ice and when this leaves the ice is white. The albedo of white ice is surprisingly high due to the large number of bubbles.

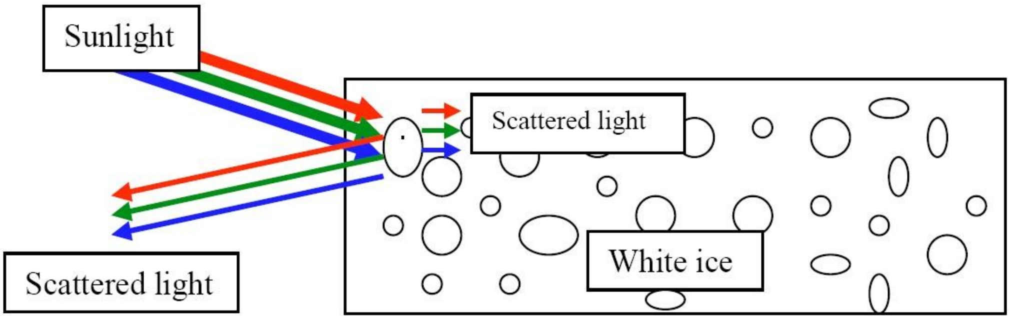

Fig 2
Image by General Atomics Sciences Education Foundation 1999 (DR)

In the case of blue ice, the ice is very well consolidated and the air bubble concentration is very low [DR]. This allows sunlight to travel a long distance inside the ice before being scattered out (see figure 3 below). As explained before, this causes the ice to be nearly transparent. This also allows for the absorption of light in the red end of the spectrum and so, the light leaving the volume is blue or green-blue. [WO] The albedo of pure ice without bubbles would be just the Fresnel reflectance for an air-ice interface. The albedo of ice increases by 7 from 400 to 250 nm.

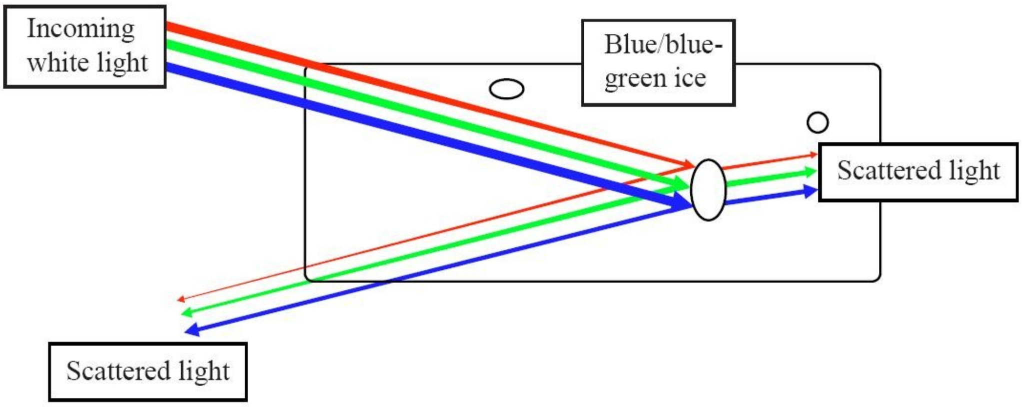

Fig 3

Image by General Atomics Sciences Education Foundation 1999 (DR)

a. Some reference images

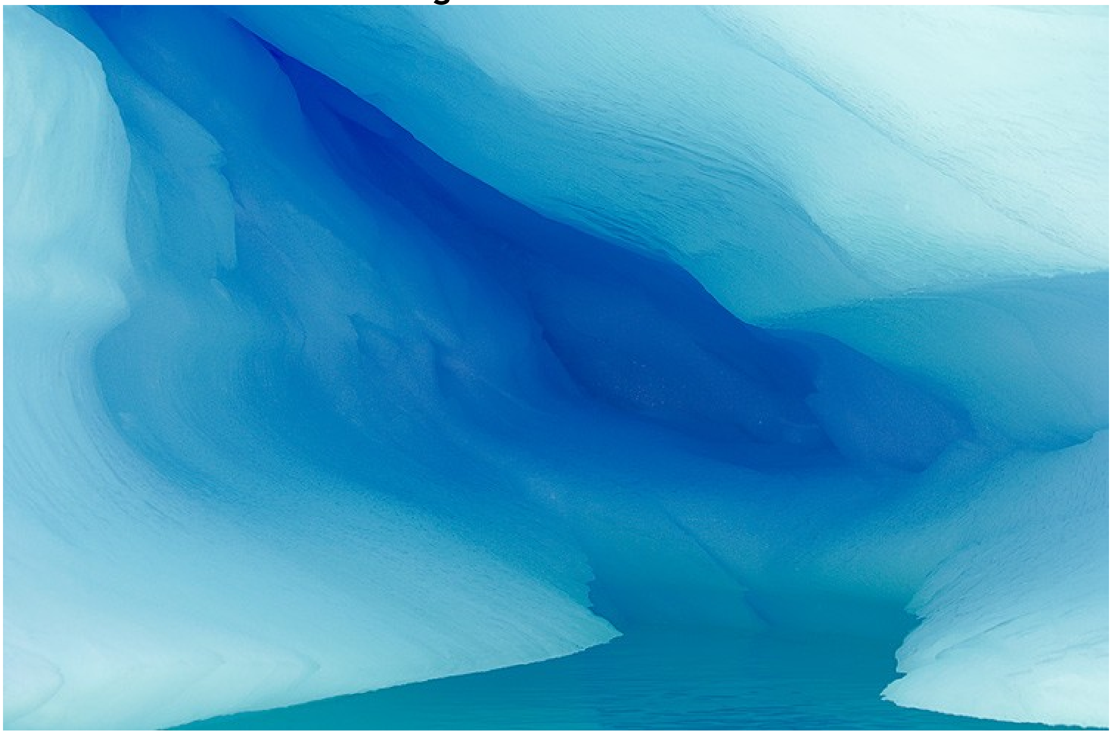

Fig 4
Photograph by Morris A 2007 (MO)

Fig 5
Photograph by Stanzel M 2007 (ST)

2. Dielectrics

Ice is classified as a dielectric, a material that does not conduct electricity. A characteristic of dielectric materials is that a relatively small percentage of the incident light is reflected and that the light that is reflected by pure specular reflection is not selective [DO].

The Maxwell Equations explain the relationships for the directionality of reflection and transmission. These equations describe the electromagnetic waves striking on a surface.

In dielectrics, a change on the speed of light in the material causes a change in the direction of the transmission. This change of direction is known as refraction [DO]. The index of refraction of ice is 1.31. [BI]

In ice the fraction of light reflected is higher at grazing angles. The Fresnel effect describes the fact that the amount of light reflected from a surface depends on the viewing angle. Fresnel equations are used to predict the reflectance of smooth surfaces through its refractive index and the angle of incidence [WE].

3. Subsurface Scattering, BRDF and BSSRDF

The light reflected by a surface in each direction can be described by the radiance in each direction. The effect of a material redirecting light is then given by a function that is the ration of the radiance reflected in a particular direction as a result of the total incident flux per unit area from another direction. This ration is known as BRDF, bidirectional reflectance distribution.

Dielectric materials always have some degree of translucency. It is said that a material is translucent when light scatters inside of it before either being absorbed or leaving the material at a different point [JE06]. Subsurface scattering is responsible for effects like colour bleeding inside materials, or the diffusion of light across shadow boundaries [DO]. The white box shown in figure 4 is translucent, notice its waxy feel and the light bleeding through its volume.

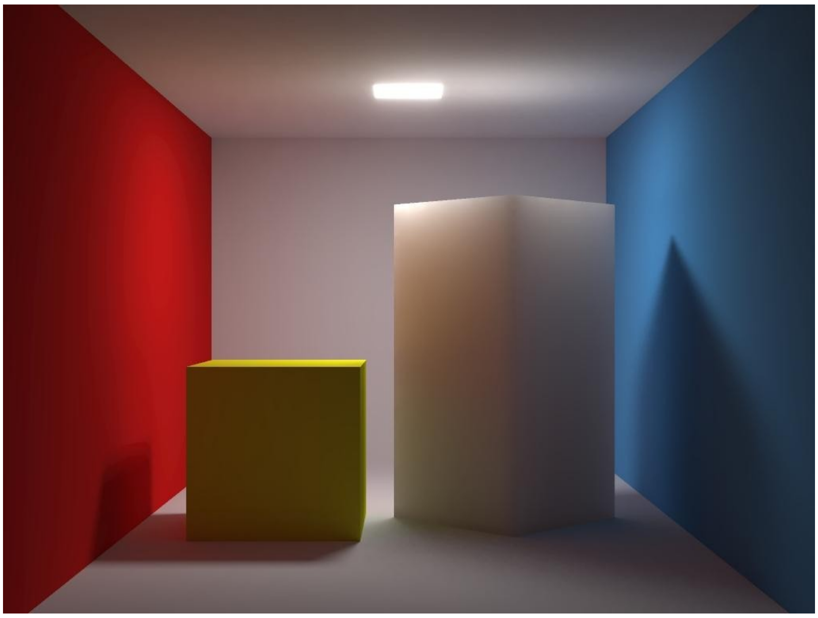

Fig 6
Image by Jensen H. (JE02)
A translucent box.

The previously mentioned BRDF, is a simplification of the more general bidirectional surface scattering distribution function (BSSRDF). BRDF assumes that light entering a material leaves at the same position. This approximation is not valid for dielectrics [JE01]. See figure 5 for an illustration of the sampling process for each function.

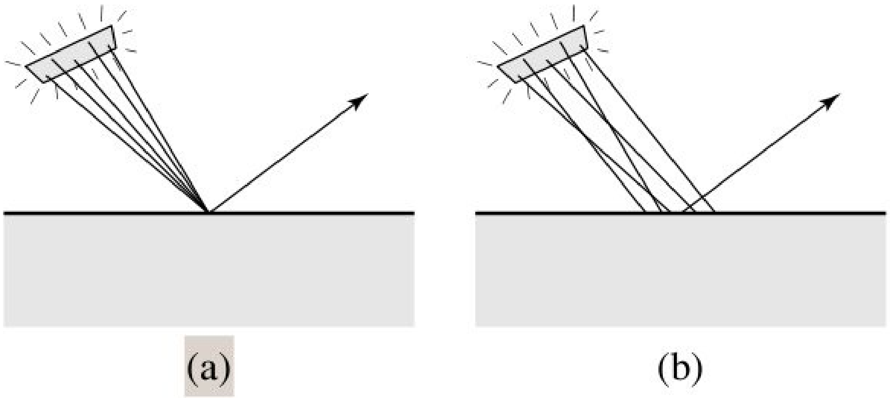

Fig 7
Image by Jensen H. (JE01)
Image a) illustrates the sampling of a BRDF and image (b) sampling a BSSRDF.

4. The Diffusion Approximation

The light distribution in highly scattering media tends to become isotropic. Each scattering event blurs the light distribution, as a result the light distribution becomes quite uniform. This was established by Jensen in [JE01].

Based on this, a diffuse approximation is made. The model combines a dipole diffusion approximation with an accurate single scattering computation. This diffuse approximation is the method used by the *ptfilter* from Renderman to compute subsurface scattering, as mentioned in the Renderman Application Notes for Translucency and Subsurface Scattering [PX].

Photon mapping is good at simulating subsurface scattering but it becomes very costly. The diffuse approximation method is faster and Jensen [JE06] says it is particularly suited for highly scattering translucent materials.

a. Ptfilter

Ptfilter is a utility provided by Renderman that performs point based calculation on point cloud files. *Ptfilter* takes a point cloud file as an input and outputs a point cloud with new data. [PX09a]

b. Brickmake

Brickmake is a utility provided by Renderman that creates a multi-resolution 3D voxel structure named brick map from a point cloud file. [PX09a]

5. Image Based Lighting

Image based lighting is a 3D rendering technique that involves the process of using images of light from the real world to illuminate CG scenes.

According to Paul Devebec, the basic steps in Image Based Lighting are:

- Capturing real-world illumination as a high dynamic range image or HDRI.
- Mapping the Illumination to the environment
- Placing the 3d scene inside this environment
- Simulating the light from the environment [DE]

In order to calculate the colour and irradiance received by a point, rays are sent randomly in different directions to calculate how much light arrives to it. When a ray hits the environment, it samples the image and gets the colour of the pixel intersected. If it hits another part of the object, then iterates to compute the light coming from this point of the object (see figure 8). [DE]

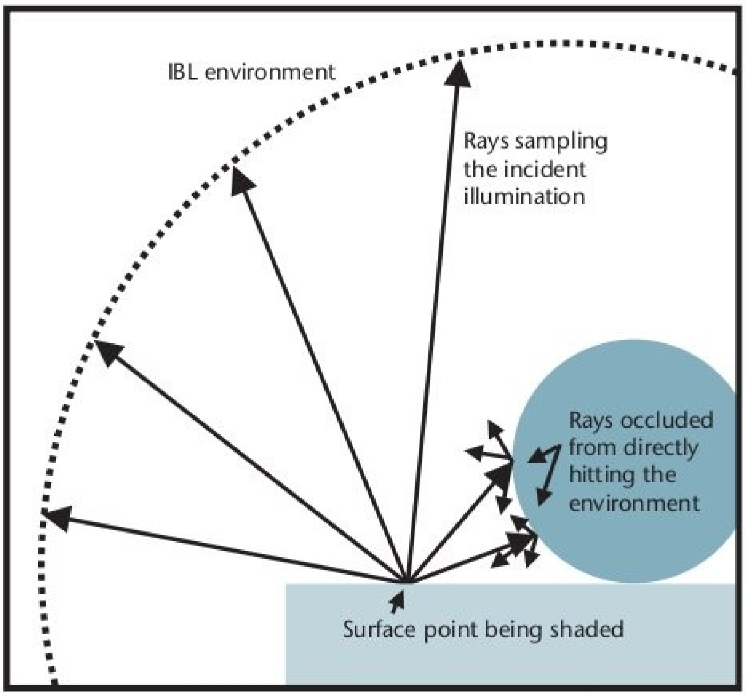

Fig 8
Image by Devebec P. (DE)
Tracing rays to determine the illuminationa point receives from an IBL environment.

Within Renderman, a *gather* loop can be used to collect light from other surfaces via ray tracing. A call to *occlusion* returns the percentage of the hemisphere that is occluded from the vector view by the environment. [PX]. This two functions will become very important at the moment of creating our image based lighting passes.

iv. Previous Work

In many films that have used iced backgrounds, these have being tricked to avoid the need of translucency effects. In Pirates of the Caribbean III, for instance, Digital Domain matte paint supervisor Wei Zheng comments about the team worries about some shots of a ship sailing into a ice cave, with icebergs and ice pieces floating in the sea. As they saw the pictures of the backplates they noticed that the ice and snow was lit from the front, and not from the back, so no translucency had to be replicated. They used matte painting in most of the cases and textures based on Greenland plates. [FO07]

Even though not much is revealed, it seems like a similar technique is used by Digital Domain in The Golden Compass to generate an ice bridge. Bryan Grill says they used an ice shader and digital painted textures to create such effect. [FO08]

The diffusion approximation technique was successfully used in the latest G.I. Joe movie to simulate deep ice. According to Aherne [AH], this technique was very efficient and effective. Ray marching inside the volume to recreate the effect of the varying density of blue ice is not necessary and implies very long processing times.

Knowing this, is interesting to analyse how the effect of subsurface scattering has been used by others to achieve different effects.

The diffussion approximation was used successfully to simulate the skin of Dobby a domestic elf in the Harry Potter movie. Also in The Lord of the Rings to simulate the skin of the character Gollum [JE06].

For Pirates of the Caribbean, Industrial Light and Magic decided to use subsurface scattering to shade various characters and elements. They ran into some obstacles to set the input parameters to the subsurface scattering calculations. Usually, illumination calls use intuitive tuning parameters that return a float and can be modulated multiplying it against a colour map. Albedo and free mean path length, are not intuitive parameters and can not be easily manipulated. To approach this problem, Industrial Light and Magic developed what they called the “texture inversion trick”. [ES]

This technique was developed by Hery C. [HE]. It consisted in using the diffuse texture maps that texture artist usually paint for BRDF surfaces, to determine the parameters for the subsurface scattering. The trick invented by Hery, was to assume that the texture maps where the result of the scattering equations made under uniform lighting conditions and then work backwards to the input parameters. [ES].

In one of the latest Pixar shorts, Lifted, the main character is made out of a green jelly-like translucent material, as can be seen in the figure above. To accomplish this effect, the irradiance was computed as the sum of the illumination along ray-marched steps inside the surface. So, for every point on the surface, a certain step distance was marched to calculate the irradiance at the point for each light. This value was then attenuated by a factor related to the distance the light traveled. This effect is very nice, but also very expensive. [PX09c]. It can be appreciated in figure 9.

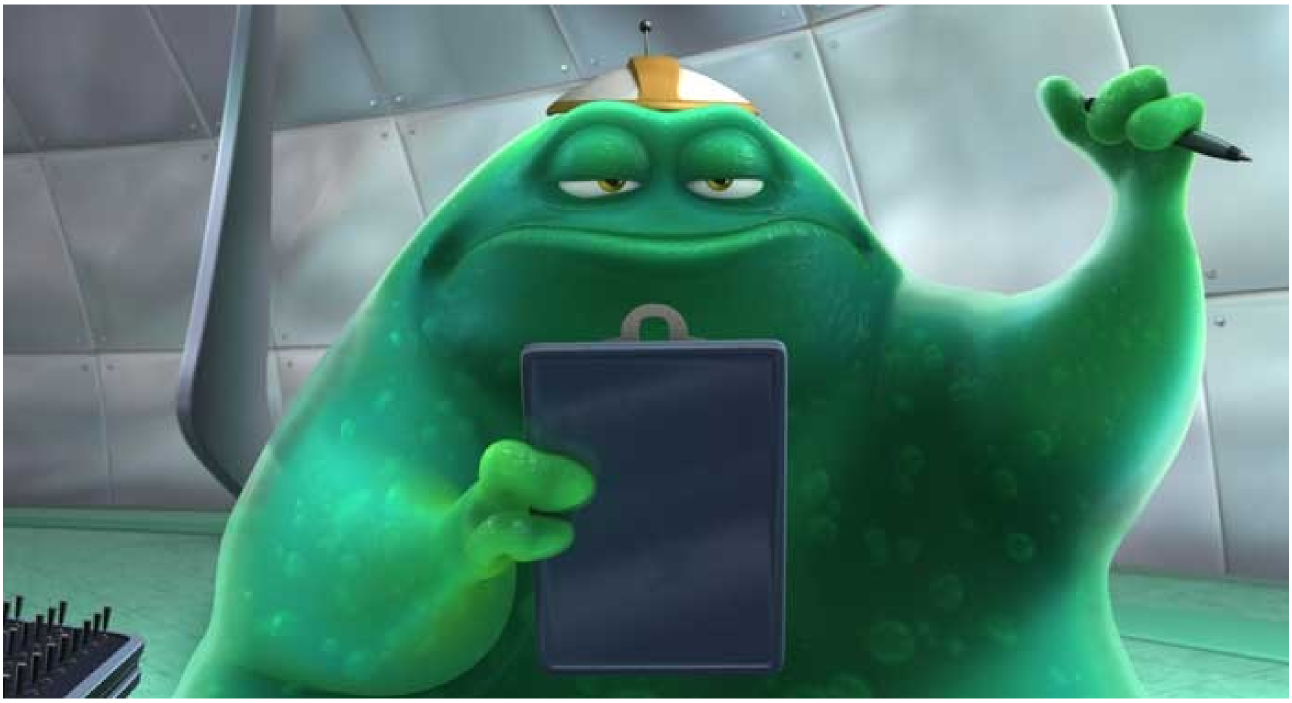

Fig 9
Image by Pixar Animations (PX09c)
Mr. B from Lifted.

v. Solution

1. Defining the problem

The problem addressed in this project is how to procedurally shade and render realistic blue ice icebergs while maintaining low rendering times.

The main characteristics of blue ice that are to be replicated are the scattering of light through the volume and the beautiful effects produced by the absorbency of the material. Other characteristics that are important to achieve the appearance of blue ice are its reflections, refractions and specularity.

One of the constraints is the rendering time. It is because of the high cost and number of resources necessary to handle long rendering times in the industry that obtaining satisfying results in the shortest time possible is so important.

2. Describing the solution

Renderman was chosen due to its status as an industry standard and its high programmability [PX]. Therefore, Renderman Shading Language will be used to write the shader and PRman to do the rendering.

The shader will be completely procedural, no painted textures will be used. The reason behind this is that, even though using textures gives control and very accurate details in most occasions, the characteristics of this particular material were considered interesting to achieve using noises and other patterns.

Ray tracing will be used, however efforts will be made to optimise the settings as much as possible.

a. General design of the shader

The shading process is done in two steps. The first step is a pre pass to bake a point cloud with some information required by the main shader. The second shader generates the final image. Both steps will be described in more detail in the following section.

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.

The main shader and the pre pass shader that creates the point cloud, both use a library named “vsl_pass.h”. This library contains the functions that create the passes needed by each shader.

At the same time 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.

In the shaders, there are flags to turn on and off the computation of each pass. This is not only a very useful feature for the final user but it makes developing very comfortable.

There is also a coefficient to modulate the effect of every contribution to the shader.

The shader receives a reference to a coordinate system that will be used to do the environment look ups. The value of this variable is “world” by default.

b. Methodology

- Maya was used to model the iceberg, create the scene and position the light and cameras.

- A rib was then produced from the Maya scene using RibExport.

- This rib was restructured and modified to be used directly with PRman to apply the shaders and do the tests.
- The pre pass for the subsurface scattering is done. It creates a point cloud.
- The point cloud is used to do the diffusion approximation with the *ptfilter*. This produces other point cloud with the results.
- The point cloud is transformed into a brick map with the *brickmake* programme.
- The brick map is used by the main shader to compute the subsurface scattering pass. The rest of the passes are generated by this shader.

Additionally, a python script was written to create the final animation. The subsurface scattering generation process was also automatised with a python script.

The set up of the image based lighting with an HDRI was very empirical. The HDRI image was first projected using a sphere. It was rotated conveniently, and then the transformations applied to the sphere, were defined as the coordinate system for the environment (plus the coordinate system transformation from Maya to Renderman).

3. The Shader

The first step was to model a very simple iceberg to start doing the test. This model evolved during the process because a more complex geometry was required. As advised by [TA] most by modelled using the sculpt tool in Maya. The details were added using a displacement shader.

a. Displacements

The displacement shader added detail to the geometry using three layers of noise: a brownian and a fractal for the overall look and an extra fractal for some more detail. To reproduce the effects of ablation a worley noise was added, but the results were not very successful. The displacement used is very small.

The displacement shader is not integrated to the main shader as it is necessary to use it during the pre pass for the subsurface scattering as well.

b. Shading

As it was mentioned previously, the shading process is done in two steps. The shader to do the pre pass for the subsurface scattering will be described under the subsurface scattering section.

“mr_cool_ice”

This is the principal shader and produces the final image. In the image is presented visually how it is composed.

- **Subsurface Scattering**

Blue ice gets most of its colour from the scattering of light and selective absorption of wavelengths as described previously. As this is emulated with the subsurface scattering pass, it will constitute the beauty pass of the shader.

The multiple scattering that characterises blue ice is approximated as a diffusion process following the Renderman Application Notes for Translucency and Subsurface Scattering [PX]. The subsurface scattering pass is our main pass.

Testing Stages

While doing tests the following procedure was applied. During a pre pass a shader would bake the direct illumination onto a point cloud, necessary values to calculate the subsurface scattering.

Following the Renderman Application Notes for Translucency and Subsurface Scattering [PX], a diffusion simulation would be produced *as a* second point cloud using the *ptfilter* .

The *ptfilter* uses the algorithms described in [Jensen01], [Jensen02], and [Hery03] . Its inputs are the index of refraction and either the albedo and diffuse mean free path, or the scattering and absorption of the material.

After the point cloud was produced, the *ptviewer* was used as advised by Aherne [AH] instead of waiting for the final image. It is quicker to see the results of the tests by examining the resulting point cloud. If more detail is needed, then the corresponding brick maps were created and the outcome was examined with the *brickviewer*.

At some point it was thought that it was possible to acquire the right values for the parameters the *ptfilter* receives by using the same technique used by Industrial Light and Magic in Pirates of the Caribbean developed by Hery [HE]. However, even after exhaustive research, it was not possible to find some of the values required by such method.

The approach then was to test different scattering and absorption values. Soon it was realised that varying albedos were required and therefore switched to testing albedos and diffuse mean free paths values. Many tests were done before finding the right parameters. Confirming what was said before, these parameters are not very intuitive. The values of the albedo were tested using the information about ice explained in the first section of this document.

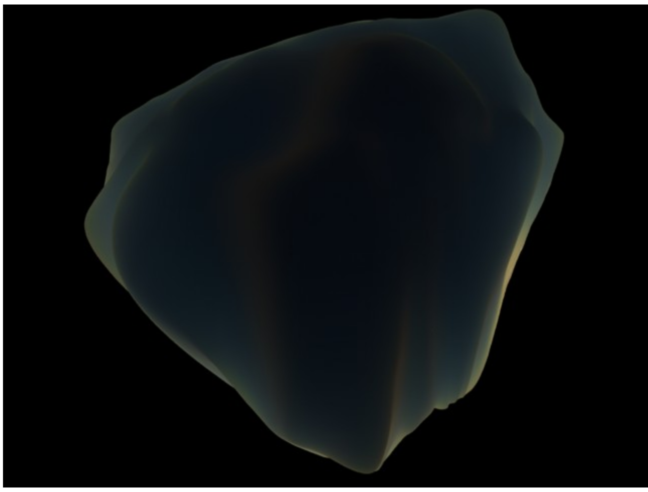

Fig 10a
Albedo: 0.8730, 0.730, 0.503
DFMP: 6.7, 8.7, 10.7
Own image

Fig 10b
Albedo: 0.510, 0.510, 0.510
DFMP: 6.7, 8.7, 10.7
Own image

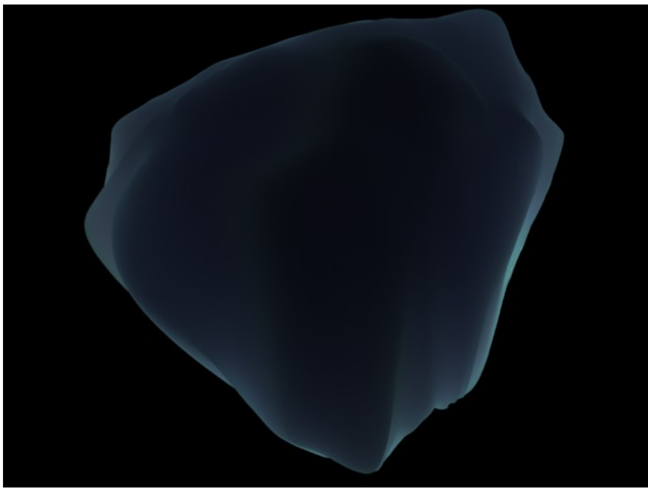

Fig 10c
Albedo: 0.530, 0.730, 0.730
DFMP: 4.0, 4.8, 7.15
Own image

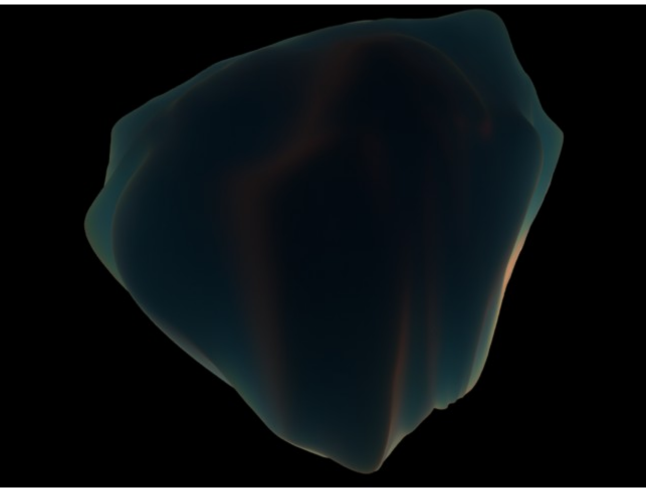

Fig 10d
Albedo: 0.930, 0.730, 0.503
DFMP: 4.0, 4.8, 7.15
Own image

Initially, the values for the albedo and the diffuse mean free path were tested taking into account the information about how ice works discussed in the beginning of this document.

In the following images, subsurface scattering with different unit length variations and the results can be appreciated.

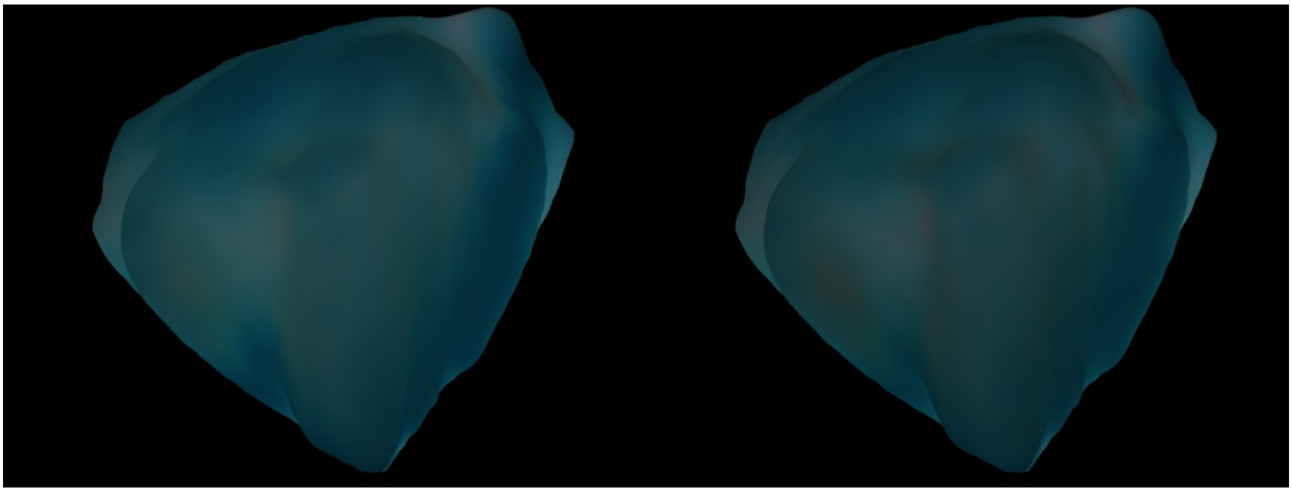

Fig 11
Both objects were shaded using the same values for subsurface scattering except for the DMFP. The DMFP for image on the right is 1. The DMFP for image on the right is 3.
Own Image

Lighting was another big variable in the equation. The next images are product of studying the difference of adding different lights.

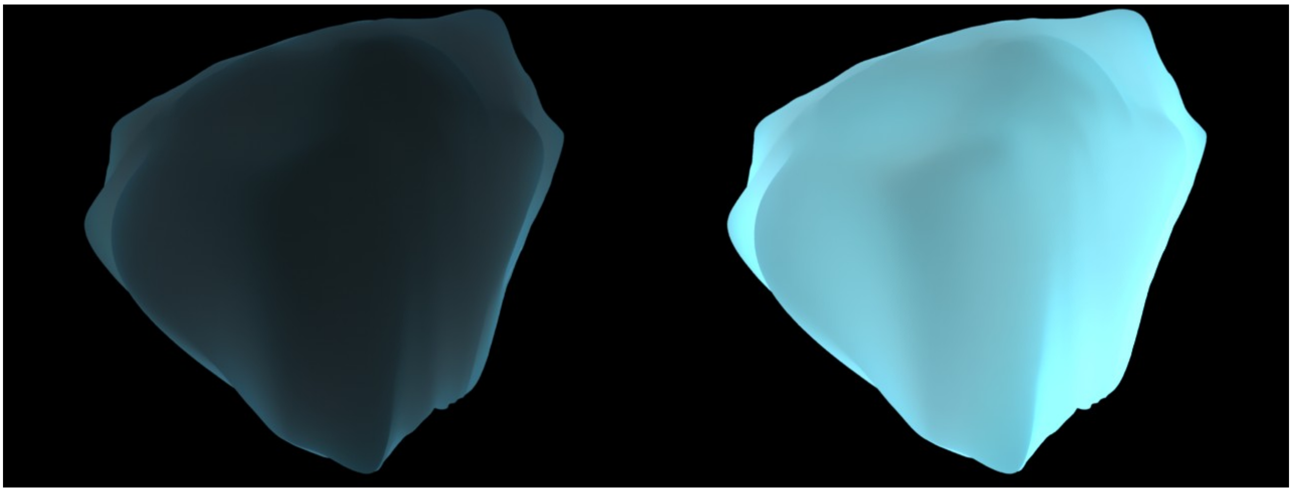

Fig 12
Both objects were shaded using the same values for subsurface scattering. The object on the left has a 3 lights scheme. The one on the right has also a back light.
Own Image

Different noises were added to the variate the albedo. Results were not very satisfying initially, as the figure 13 shows.

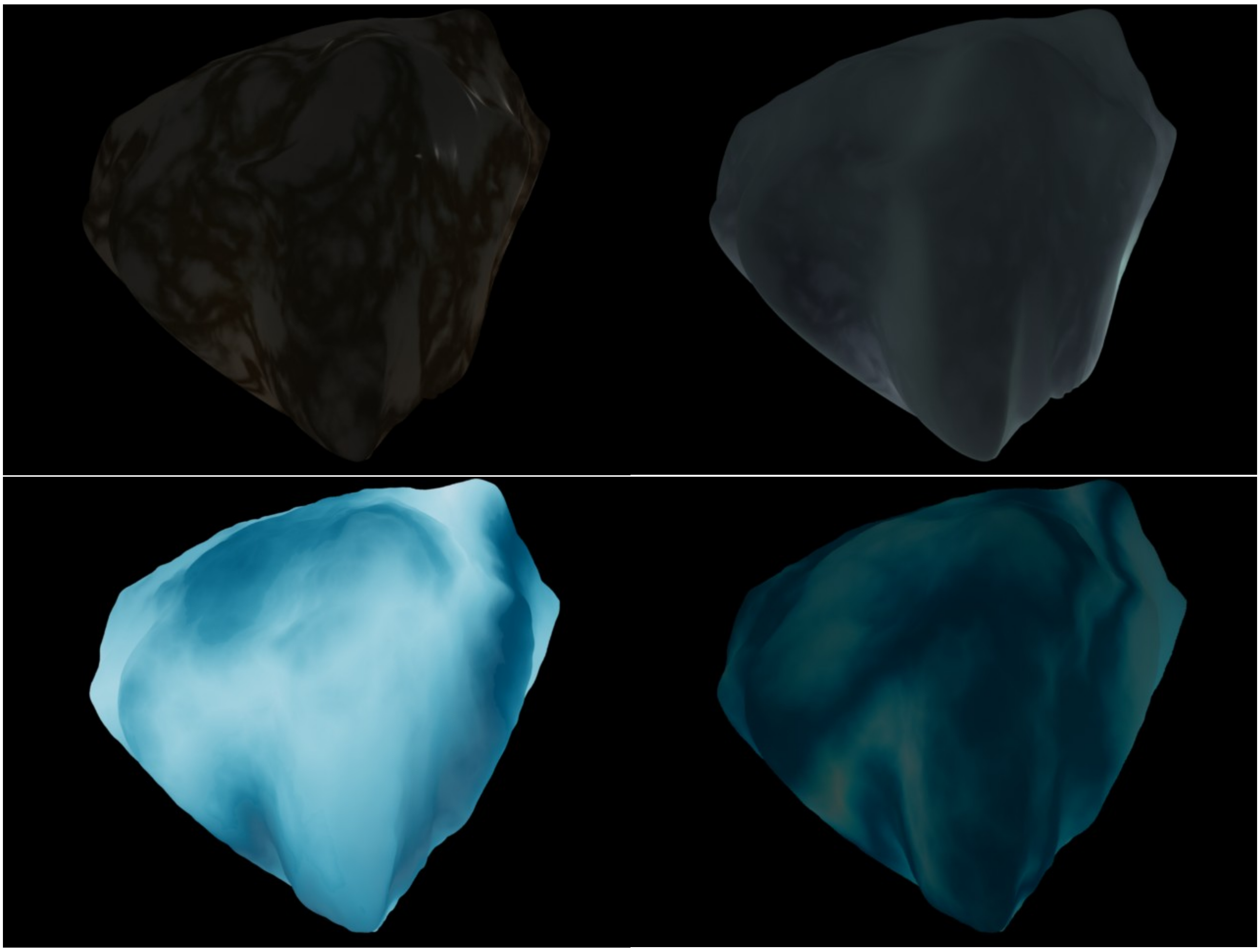

Fig 13

Noises added to the albedo. Initial results were far from the blue colouration obtained before.

Own Image

Results

“bake_info” (subsurface scattering pre pass)

The purpose of doing this fist step, is to bake onto a point cloud the necessary information to later produce the subsurface scattering pass. In other words, this pass serves the subsurface scattering pass.

To produce the point cloud, two lights are used. One illuminates the volume from behind and the other from the front up at the left as can be appreciated in the following image.

This was done in this way because as explained by Apodaca [AP] when dealing with translucent objects is important to consider how back lighting affects them. This was also pointed out in both articles “Shading Food: Making

It Tasty for Ratatouille” and “Lighting Food” [PX09b]. There is a very close relation between lighting and scattering.

Since the *subsurface()* function is going to be used, the point cloud must have the micropolygon area and the transmitted direct illumination at each point. This information is then stored in the “radiance” and “area” channels. The point cloud is generated using the *bake3d()* shadeop.

While creating the pre pass, backface, hidden and culling must be turned offso that first bounce reflections can easily find faces that are backfacing or hidden to the final render camera. [PX]

Radiance

The radiance[1] is calculated by multiplying the irradiance[2] and the surface colour, this last might be provided as a parameter.

The irradiance is being calculated considering all the non camera dependant information: the image based diffuse, the ambient occlusion as its complement and the indirect diffuse, according to the following equation:

irradiance=((Kibl * diffuseibl) + ((1-Kibl) * occlusion)) * Kocc + indirectdiffuse * Kind

- Ambient Occlusion (occlusion)
- Image Based Lighting Diffuse (diffuse_IBL)
- Indirect Diffuse or Colour Bleeding (indirect_diffuse)

These passes will be explained in detail next.

Area

When baking the area it is important to enable the “dicing” option. Otherwise, the shader would receive the smoothed version of the geometry, instead of the micropolygons, producing incorrect results.

Even thought the *ptfilter* was used directly during the tests, the *subsurface()* function is used to create this pass to avoid the creation of point clouds and brick maps to the user. Nevertheless, it is advised to generate an organised point cloud from the point cloud obtained in the pre pass.

---

[1] This term describes the amount of light that passes through or is emitted from a particular area and falls within a given solid angle in a specified direction. Characterises total emission or reflection.[DO]

[2] The radiant energy per unit time and area at a surface.[DO]

“mr_cool_ice” (subsurface scattering pass)

Albedo

The varying albedo of ice, was modelled using a marble noise. Marble noise was chosen because it is a three dimensional noise that resembles the appearance of blue ice. After the albedo was calculated using do_Albedo, it was saved as a colour variable and exposed as one of the shader outputs (“_albedo”) for control.

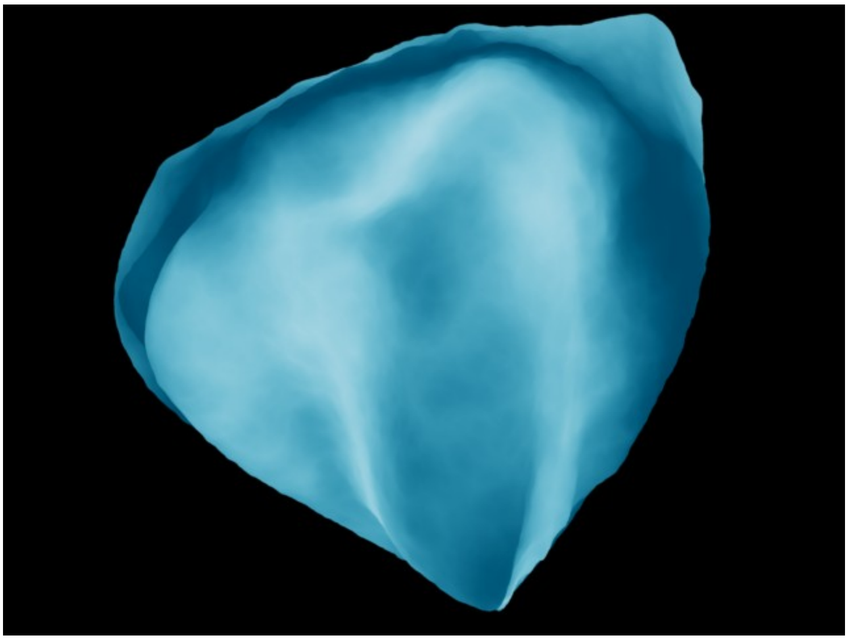

Fig 14
Varying albedo
Own Image

Diffuse Mean Free Path

The value of the diffuse mean free path[3] was altered several times trying to find an appropriate value. When the diffuse mean free path is too long, the ice looks jelly like. Under the lighting conditions described before while baking the radiance point cloud, the value used for the diffuse mean free path was 1.8.

Index of Refraction

The value used was 1.2. The physic value of ice was not used because it was not visually pleasing.

---

[3] The average distance a ray travels before it encounters an obstacle and reflects.

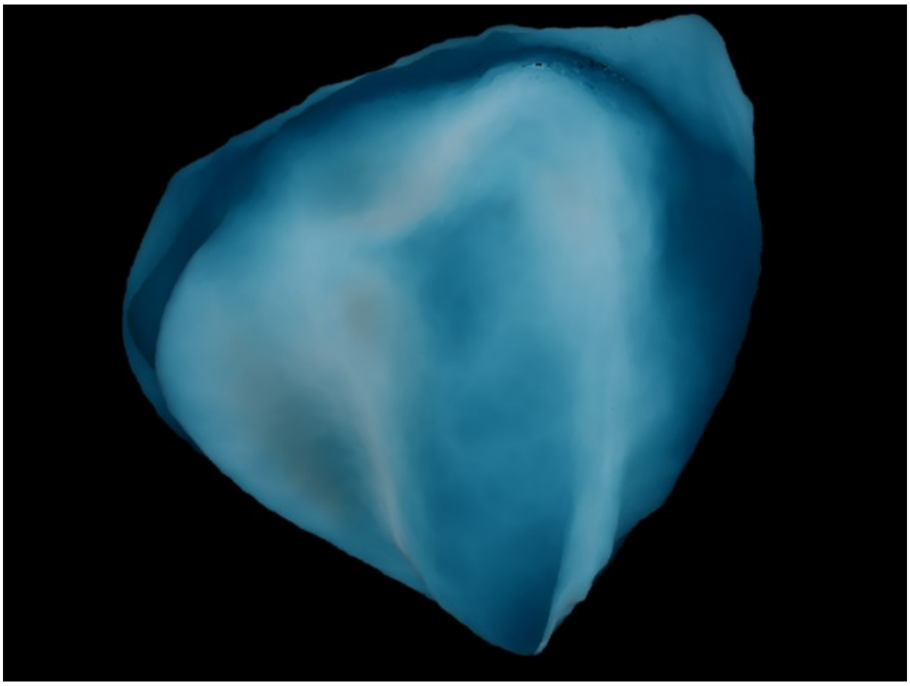

Fig 15
Subsurface scattering pass
Own Image

- **Ambient Occlusion and Image Based Lighting Diffuse (ambient_occlusion and diffuse_IBL)**

These two passes were generated using the *occlusion* function and doing the environment look ups at the same time to economise, as advised by [PE].

The PRman *occlusion* function is called using the normal normalised and facing forwards, also called shading normal. The function is also provided with a blurred version of the environment map and it is asked to return the information of the environment look up in a colour variable.

The environment diffuse is the result of multiplying the colour from the environment look up at that point by the occlusion value.

The *occlusion* call uses a reference to a coordinate system to do the environment look ups. This resulted to be important due to the fact that first, the geometry was imported from Maya, and second, the environment map was rotated. Without specifying the appropriate coordinate system, the lighting information would not have matched the environment map used.

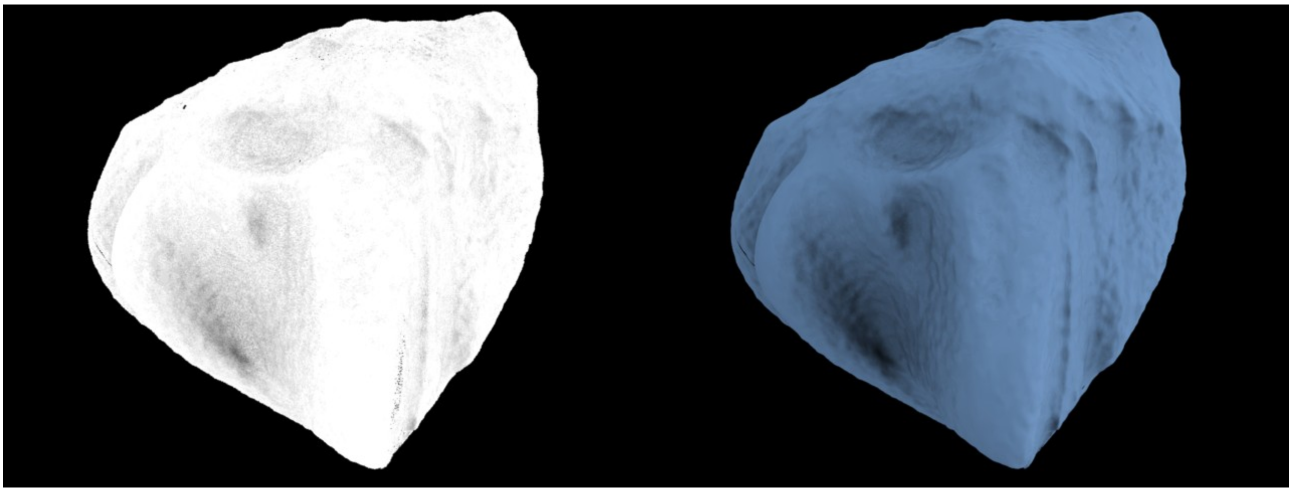

Fig 16
Ambient occlusion and Image Based Lighting diffuse passes.
Own Image

■ Indirect Diffuse or Colour Bleeding (indirect_diffuse)

This pass is included to account for colour bleeding. It is done with a call to *indirect diffuse*. Currently is not being used because there are no more objects in the space so, it is not necessary. Time wise it is an expensive pass.

■ Reflection Occlusion

The reflection occlusion function, receives the R vector as either the product of *reflect(In, Nn)* or as the reflection vector obtained from the Fresnel function. This depends on whether the Fresnel flag of the shader is set to on or not. The reflection occlusion pass, uses a reference to a coordinate system to do the environment look ups as well.

It is executed with a call to *gather()* around the vector R. It uses an angle to blur the reflection and specifies the number of samples which is by default a low, but good enough value. Expects to receive the colour of the surface intersected and the ray direction.

When there has been an intersection, the colour of the surface is added up to a variable. When there is no hit, then an environment look up is done utilising the ray direction and this information is added to the same variable. The ray direction is transformed to the provided coordinate system before doing the environment look up.

Finally, the result of dividing the colour contribution and the number of samples is returned.

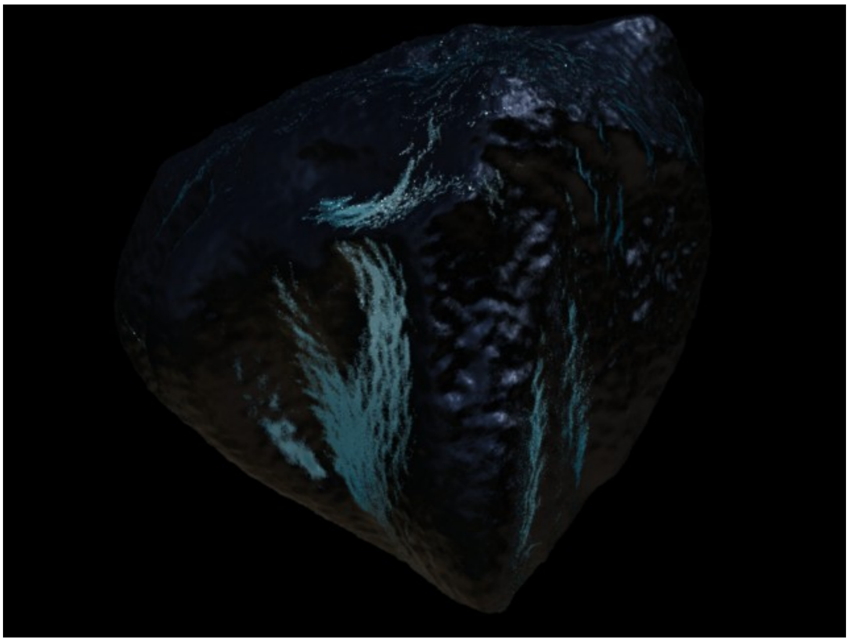

Fig 17
Reflection occlusion pass
Own Image

- **Refraction**

The refraction function receives a vector that is either the result of the *refract()* function or is the refraction vector obtained from the Fresnel function. As in the case of the reflection occlusion function, this depends on whether the Fresnel effect was activated or not by the user. The refraction pass is achieved with a call to *trace.*

- **Specularity**

An specific BRDF for icebergs was not available, only tables with the corresponding values. Going to an iceberg and calculate the BRDF was not a viable option either.

Therefore, the specularity had to be approximated. To do so, a Cook Torrance function was used. Cook Torrance was preferred because it uses the Fresnel effect, which is a characteristic of ice, it has being already contributing to other passes. The Cook Torrance model also takes into account the micro imperfections of the surface and evaluates how, depending on their orientations, contribute to the specular reflections. [ES]

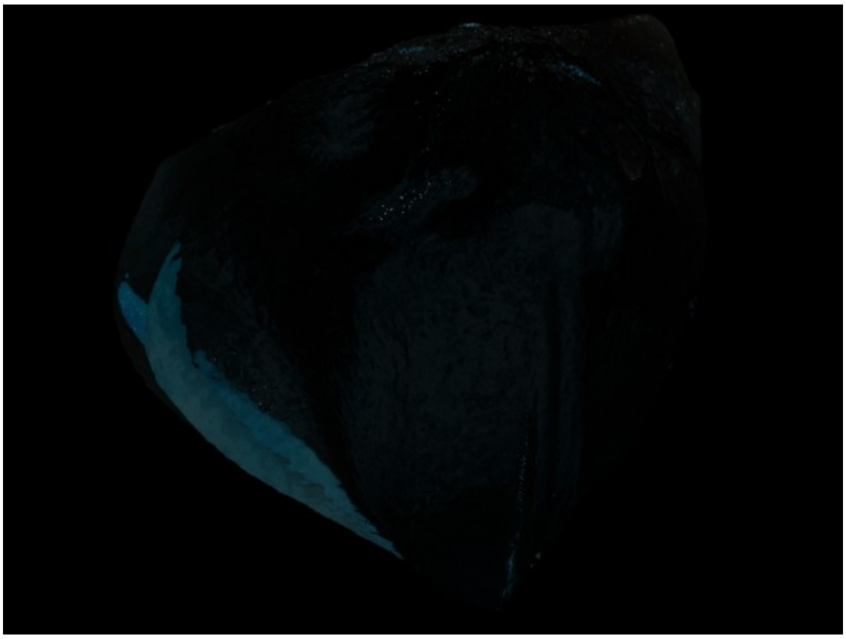

Fig 18
Refraction pass
Own Image

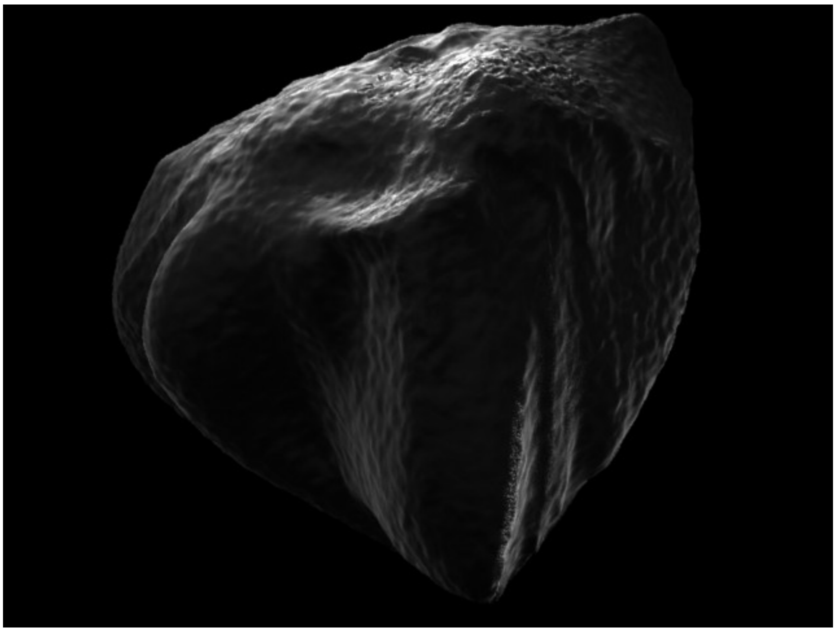

Fig 19
Specularity pass. Cook Torrance model.
Own Image

■ Wet Effect

A blinn specularity was added on top of the Cook Torrance to create a wet effect as an optional feature. The effect was accomplished by following the technique explained by [FE]. The specular colour uses the subsurface scatering value as input. This effect is not necessary to simulate every kind of ice.

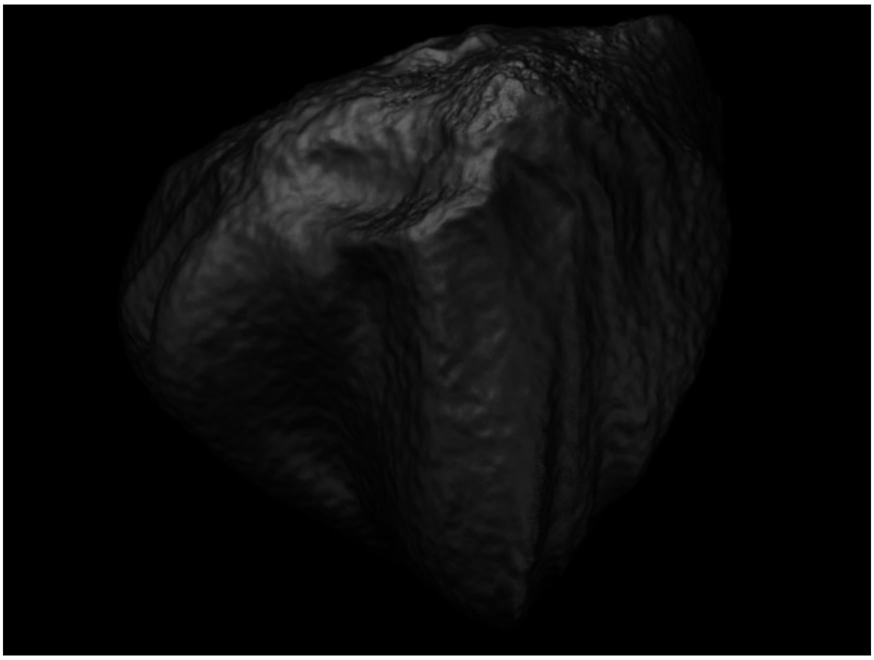

Fig 20
Wet Effect. Blinn model.
Own Image

- ### Snow

As an optional feature, there is the possibility to add frost to the ice. To do so, the following elements were included:

Snow Mask:

The aim of this mask, is to select the places of the geometry where the snow or frost could accumulate. To do so it first checks the areas where the component y of the normal is greater than a value specified by the user.

After that, is necessary to decide the level of the snow. Since 80% of the volume of icebergs is underwater, is not very possible that these areas were covered by snow.

Then a turbulent noise might be added by choice to randomise a bit the placement.

Diffuse:

The white colour of the snow is modulated by a fractal to add more variations. The final diffuse of the snow is obtained by multiplying this value by the snow mask. It also considers the contribution of the diffuse occlusion and the ambient occlusion.

Displacement:

The value is obtained by subtracting the snow mask to the diffuse of the snow.

Sparkles:

This effect intends to replicate the specularity of the small ice crystals present in snow. This is done based on [IN]. To do so a blinn specular model with a large roll off and a eccentricity of around 0.5 is used. A brownian noise with very sharp and small noises is used as the input of the specular colour.

As the sparks should not look fixed on the surface, the “fourth dimension” field of the brownian noise uses a function based on the x, y and z components of the I vector, multiplied by a factor to control the speed of the sparkles movement.

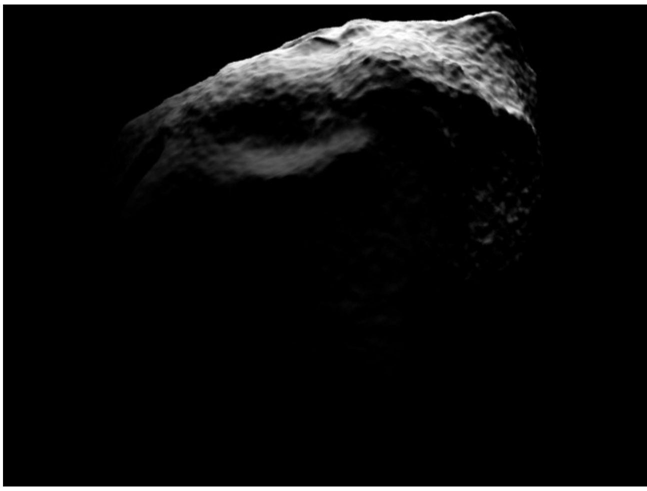

Fig 21c
Albedo: 0.530, 0.730, 0.730
DFMP: 4.0, 4.8, 7.15
Own image

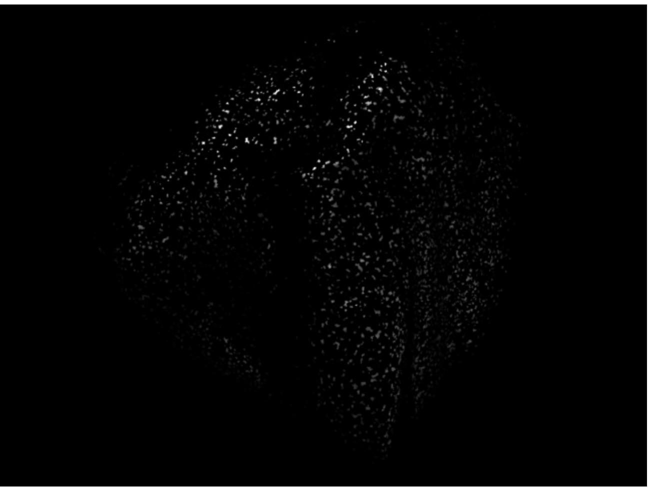

Fig 22d
Snow Sparkles
Own image

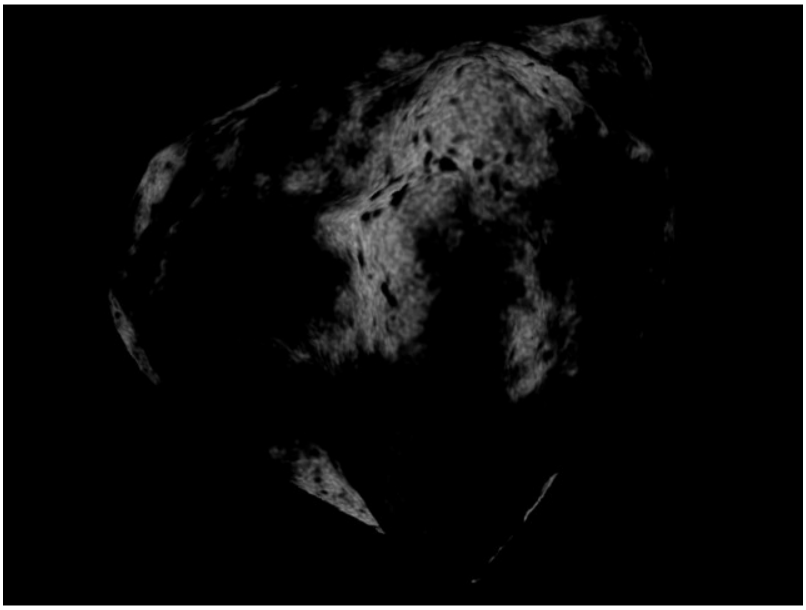

Fig 23
Snow Diffuse **Own Image**

Recipe to assemble the final result:

```
subsurface * Ksss
+
diffuse_colour * ((Kibl * diffuse_ibl)+(1.0-Kibl * occlusion)) * snow_mask
+
sparkles * snow_mask
+
reflection_occlusion*Kr
+
refraction*Kt *fresnelKt
+
specularity*Ks * fresnelKr
+
specularity_blinn * Kwet *_subsurface
```

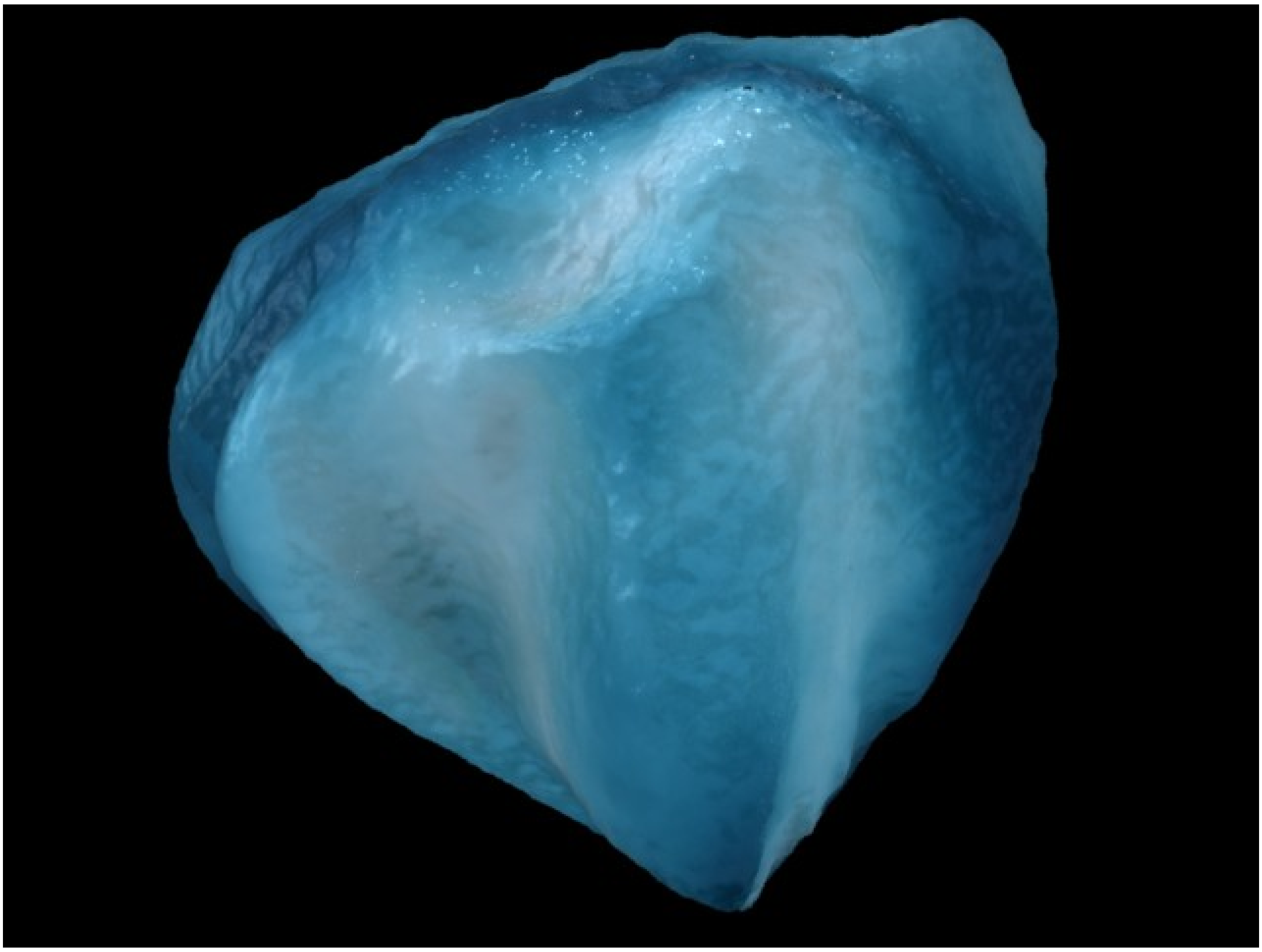

Fig 24
Final Image

vi.Conclusion

Regarding the primary objective of the project, which was to render blue ice efficiently, it can be said that it was achieved.

It would have been desirable to use a specific BRDF for icebergs. However, since only tables with values and no actual BRDF was found an approximation was made layering a Cook Torrance and a Blinn specularity.

Some aspects could be improved. The ambient occlusion is not taking into account that the surface of the ice is very reflective, specular and refracted, being so, it should not be possible to any point of the surface to be largely occluded. To solve this problem secondary rays should have been taken into account.

Also, regarding ambient occlusion, point based ambient occlusion was not used. The efficiency achieved in the process of creating the ambient occlusion pass with this technique, is lost when going through the trouble of creating an extra pass to shade a single object. If an entire scene was being shaded, then point cloud based occlusion would be recommended.

Perhaps implementing the solution using Slim would have been interesting as a learning experience. Also, it would have made the testing process faster and easier.

vii.References

[AH] Aherne, S. (shanea3d[at]gmail.com), July 8, 2009. *Re: icebergs ice icebergs.*e-Mail to Salas Castillo V. (vanessitaenlaluna[at]gmail.com).

[AP] Apodaca A., Gritz L and Barzel R. 2000. *Advanced RenderMan: creating CGI for motion pictures*. Burlington: Morgan Kauffman Publishers.

[BA] Baek D., Davis A., Jacobs D. 2009. *Glacier Cave*. Stanford University Available from: https://graphics.stanford.edu/wikis/cs348b-09/GlacierCave (Accessed 30 May 2009).

[BI] Birn J. 2000. Digital Lighting and Rendering. United States of America: New Riders Publishing.

[DE] Debevec P. 2002. Image-Based Lighting. IEEE Computer Graphics and Applications. Available from: http://www.debevec.org/CGAIBL2/ibl-tutorial-cga2002.pdf (Accesed 12 August 2009)

[DO] Dorsey J., Rushmeier H., Sillion F. 2008. Digital Modeling of Material Appearance. Burlington: Morgan Kauffman Publishers.

[DU] Douma M, curator. 2008. Snow and Ice. *Causes of Color*. (Accessed 12 July 2009). Available from: http://www.webexhibits.org/causesofcolor/5C.html

[DR] Drummond B. 2002. Why is blue ice blue. *About blue Ice.* Available from: http://www.carleton.edu/departments/GEOL/Links/AlumContributions/blueice/blue.html (Accessed 25 June2009).

[ES] Escape Studios. 2009. *Pixar's Renderman Certified Courseware*. Available from: http://nccastaff.bournemouth.ac.uk/jmacey/Renderman/rmancourse/twiki/bin/view/Main/WebHome (Accessed 12 July 2009).

[FE] Fennell S. 2005. How to Create a Wet Shader in Maya. *Highend3d*. Available from: http://www.highend3d.com/maya/tutorials/rendering_lighting/shaders/186.html (Accessed 10 August 2009).

[FO07] Fordham J. Pirates of the Caribbean 3. *Cinefex*. 110, 62.

[FO08] Fordham J. The Golden Compass. *Cinefex*. 112, 62.

[HE] Hery C. 2003. *Implementing a Skin BSSRDF (or Several...)*. Industrial Light and Magic. Lucasfilms*. SIGGRAPH 2003.*

[JE01] Jensen H, Marshner S, Levoy M and Hanrahan P. 2001. *A Practical Model for Subsurface Light Transport.*

[JE02] Jensen H and Buhler J. 2002. *A Rapid Hierarchical Rendering Technique* *for Translucent Materials. In Proceedings of SIGGRAPH 2002, 576 -581*

[JE06] Jensen H. 2006. Subsurface Scattering. *Images with Subsurface Scattering*. Available from: http://graphics.ucsd.edu/~henrik/images/subsurf.html. (Accesed 10 May 2006)

[IN] Inti. 2005. Realistic Snow Tutorial. *Highend3d*. Available from: http://www.highend3d.com/maya/tutorials/rendering_lighting/shaders/Realistic-Snow-198.html (Accessed 10 August 2009).

[MA] Malcolm K. 2009. CG References and Tutorials. Savanah Collegue of Art and Design. Department of Visual Effects. Available from: http://www.fundza.com/index.html (Accessed 01 July 2009).

[MC] McCoy D. and Bashforth B. Shading Food: Making It Tasty for Ratatouille. 2009. *Pixar's Renderman Certified Courseware*. Available from: http://nccastaff.bournemouth.ac.uk/jmacey/Renderman/rmancourse/twiki/bin/view/Main/WebHome (Accessed 12 July 2009).

[MO] Morris, A. 2007. *Blue ice cave in iceberg.*[photograph]. Birds as Art. Available from: http://www.birdsasart.com/bn221.htm. (Accessed 15 May 2009).

[PE] Perez O. (osirisp[at]gmail.com), July 14, 2009. Re: hielitos. e-Mail to Salas Castillo V. (vanessitaenlaluna[at]gmail.com).

[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).

[PX09a] Pixar Animation Studios. 2009. Utilities RenderMan Pro Server. *Pixar's Renderman Pro Server*. Available from: https://renderman.pixar.com/products/tools/rps-utilities.html (Accessed 19 August 2009).

[PX09b] Pixar Animation Studios. Ratatouille Lighting Team. 2009. Lighting Food. *Pixar's Renderman Certified Courseware*. Available from: http://nccastaff.bournemouth.ac.uk/jmacey/Renderman/rmancourse/twiki/bin/view/Main/WebHome (Accessed 12 July 2009).

[PX09c] Pixar Animation Studios. 2009. Inside Pixar's Shorts. *Pixar's Renderman Certified Courseware*. Available from:http://nccastaff.bournemouth.ac.uk/jmacey/Renderman/rmancourse/twiki/bin/view/Main/WebHome (Accessed 15 July 2009).

[ST] Stanzel M. 2007. National Geographic. Available from: http://lh3.ggpht.com/_oRMGJP0hCtE/R3OrK8j8mqI/AAAAAAAAAVI/kA9cq0mrHXc/blue_iceberg.jpg. (Accessed 15 May 2009)

[TA] Taylor N. (noahbtaylor[at]gmail.com), June 30, 2009. Re: Iceberg texture. e-Mail to Vanner A. (avanner[at]bournemouth.ac.uk).

[WA] Warren S., Brandt R. and Boime R. 1993 (Review 2003). Blue Ice and Green Ice. *Antarctic Journal*.. Vol 28. p. 255-256. Available from: http://www.atmos.washington.edu/~sgw/PAPERS/1993_MtHowe.pdf (Accessed 02 June 2009).

[WE] Westin S. 2007. *Fresnel Reflectance*. Cornell University. Available from: http://www.graphics.cornell.edu/~westin/misc/fresnel.html (Accesed 02 August 2009).

[WH] Whitehurst A. 2009. Lighting VFX Shots the IBL Way. *Andrew-Whitehurst.net.* Available from: http://www.andrew-whitehurst.net/howTo2009.html (Accessed 05 July 2009).

[WO] Woolf L. 1999. The Color of Ice. General Atomics Sciences Education Foundation. San Diego California.

viii.Appendices

Other images obtained in the way

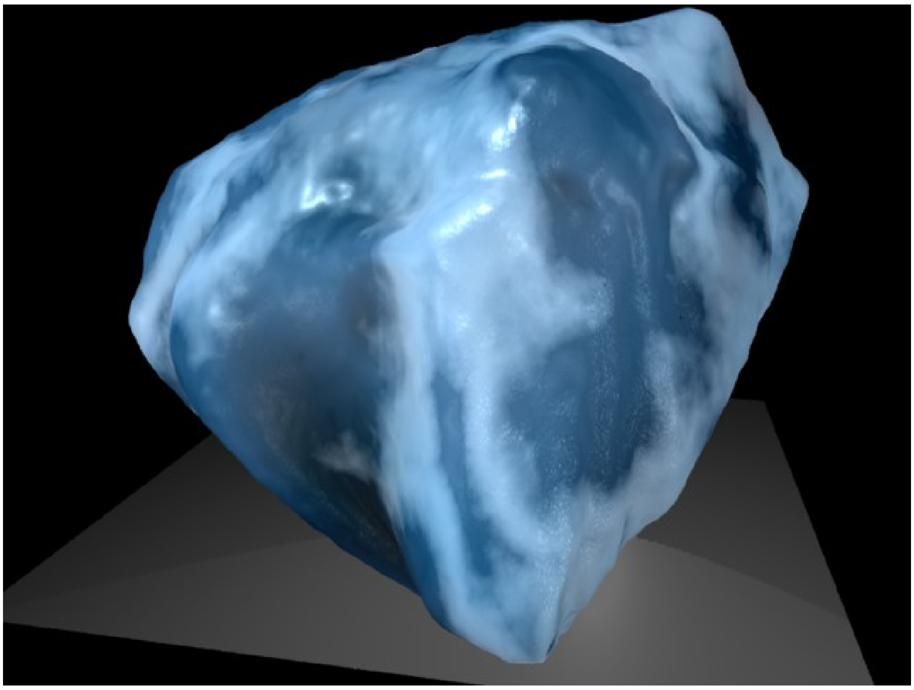

Fig 25

Fig 26

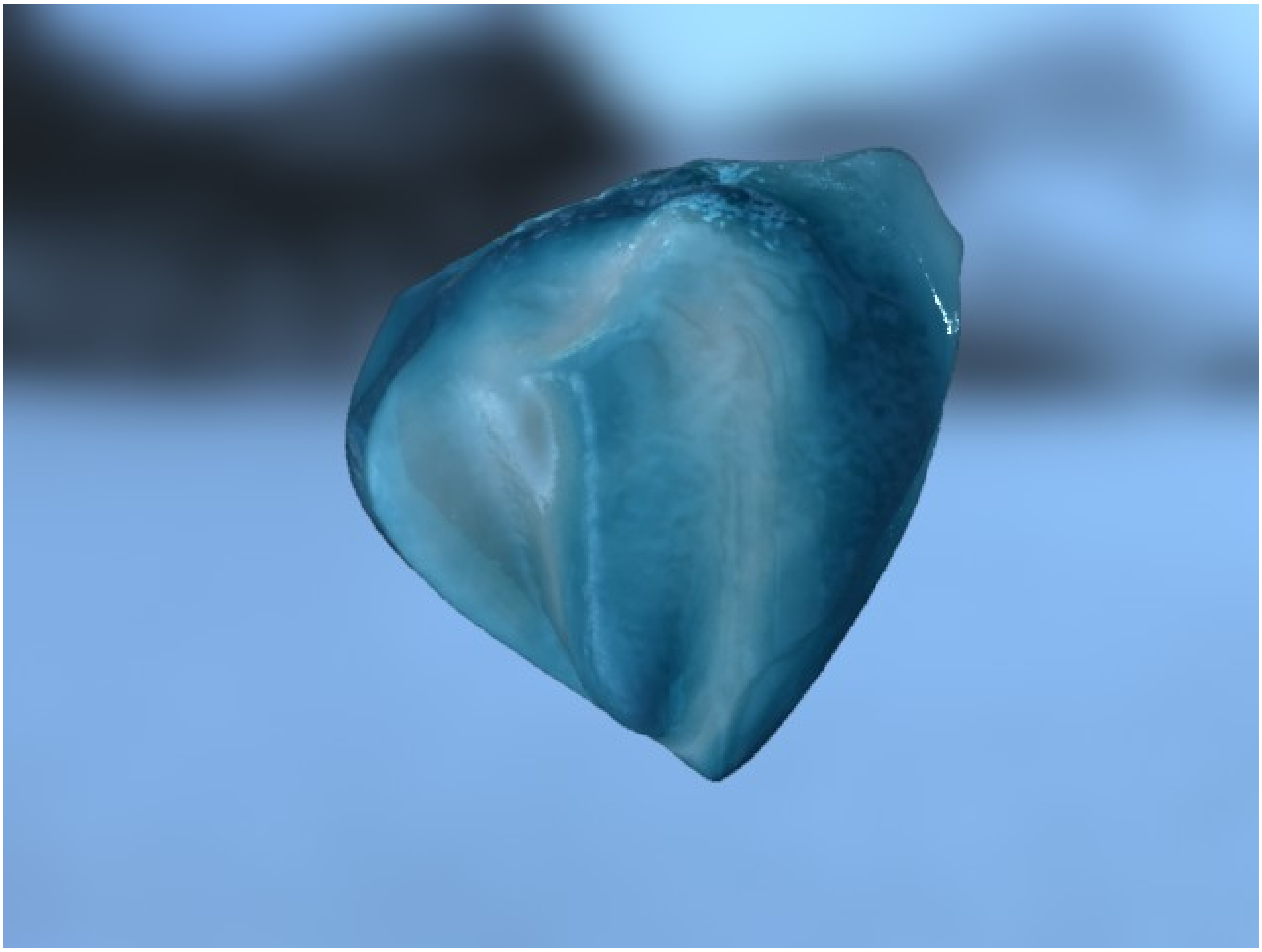

Fig 26 Composed with the HDRI used for the environment