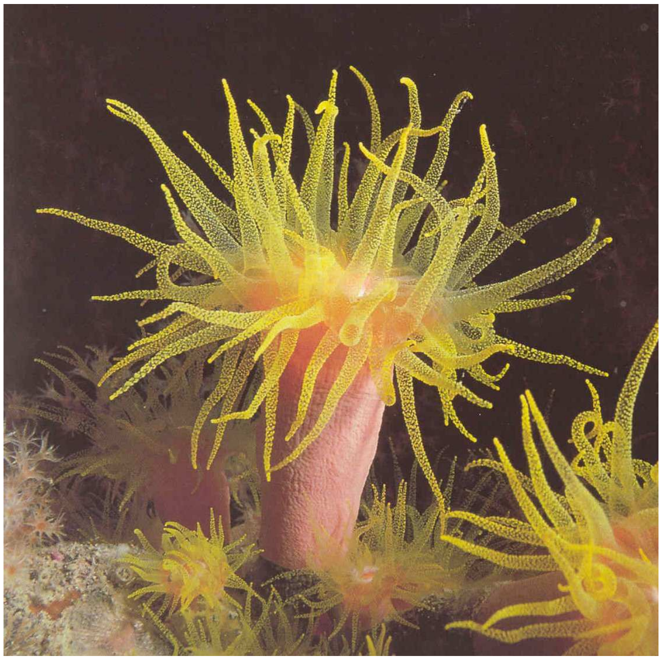
# Underwater Sea Anemone Scene Project

Paul Timpson
e9508113

Msc Computer Animation Thesis 2006



Contents

# Underwater Sea Anemone Scene Project

Paul Timpson
e9508113

## 1 Introduction

The Project was carried on from term three with the aim of improving the quality and accuracy of depictions of the underwater subjects as well as the atmosphere they inhabit. To achieve this, research was carried out of both a technical and visual nature. The visual research mainly consisted of consulting the “Blue Planet” book and DVD series for visual cues for the look of the scene as well as searching the internet for articles regarding ocean structures and Anemone-type life forms. Other texts consulted include the work entitled “British Anthozoa” by R.L Manuel which includes detailed

descriptions of Anemones found in British waters (Ref 19).
Visually the paramount desire was to give the scene a more organic feel. To do this an illusion of depth and tangibility had to be given to the subjects in the scene. This “Depth and tangibility” applies to the various areas of the scene;

a) The Terrain.
b) The Rocks.
c) The Anemone.
d) The Cloudy Atmosphere of the Water.

As well as the above, attention was given to the arrangement of primary light sources. A classic three light setup was used for the beauty pass; the multifaceted Uberlight was used for the light interaction with the water plane above the scene; however a differently arranged three direction light set up was used for the Subsurface scattering (to give more light from behind the Anemone) .
There was also an effort made to introduce more aspects of control into the scene and also to add a greater element of animation. This was achieved by the addition of a flock of fish and the greater use of user defined control textures (for the terrain and Fog).

In the detailed “Solution” section below all of the above elements of the scene will be discussed under the headings of Modelling, Shading and animation in an effort to cover all the disparate processes which went into production of the scene.

## 2 Previous Work

A major problem identified was the depiction of organic realism, the major organic creature in the scene is the Anemone. A decision was made to research the anatomy of these creatures to gain a better grasp of the textural appearance and the movement of their bodies.
The sea Anemone is part of the Zoantharia family of sea creatures (which also includes coral). They are named after the terrestrial flower and are filter feeding members of the Actnaria order. An Anemone is basically a polyp attached to the ocean floor by an adhesive foot. They feed by using tentacles which surround the mouth. Their bodies are soft and have an amount of translucency. “The hollow cylindrical body of a polyp is the column which terminates at its upper, oral ends in a transverse oral disc” a detailed diagram of the structure of an anemone can be seen below (ref 19- page 5)
The can reproduce asexually and sexually; on occasion actually tearing themselves apart to reproduce.

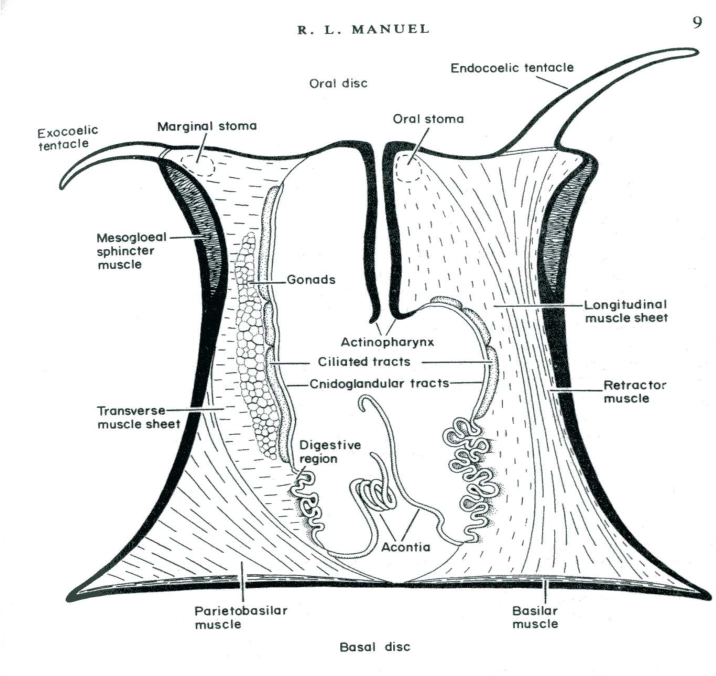
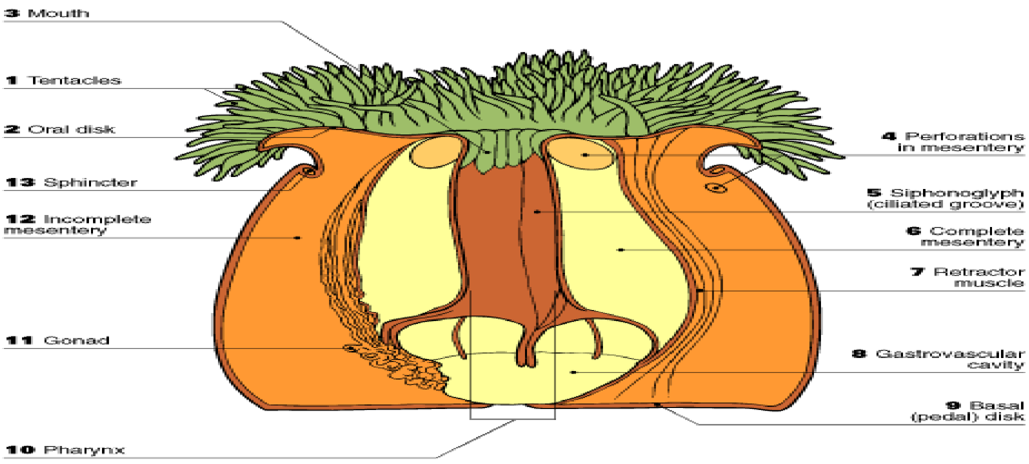


Fig 1 Anemone internal structure example a. (Ref 19)

Image from “British Anthozoa” R.L Manuel “Synopses of the British Fauna” editors Doris M Kermack and RSK  Barnes ( )



© Diagram Visual Information Ltd.

Fig 2 Anemone internal structure example b. (see ref 20)

The body cavity is called the coelenteron “the internal surface of the folded edge of the actinopharynx is formed into a densely ciliated groove, the siphonoglyph, which directs a water current into the coelenteron. This water current is used both for respiratory purposes and to inflate the polyp by hydrostatic pressure.”(Ref 19)

The above information on how an Anemone keeps its rigidity was key to decisions taken during the project regarding the movement of the Anemone and led to the use of movement akin to a related type of animal who uses water for both rigidity and movement, the Jellyfish.

The central challenge in capturing the surface of the anemone was its translucent soft texture. A decision was made to utilise subsurface scattering as a path to achieving this translucent softness. As subsurface scattering was to be one of the mainstays of the project of the project it received a substantial amount of research time.

A Siggraph paper (2001 ref 12) was used as a starting point for research as it included a chapter on “layered media for surface shaders”. This paper was suitable as it begins by describing some of the first research into light and its reaction to media of a thick or layered nature. The paper began by citing research carried out by Kubelka and Munk (KM) (1931). This research outlines a model for designing the appearance of layered Surfaces and specifically how light reflects and scatters through each layer.

They (KM) take the example of different layers of coloured paints on a white wall and state that “As light enters the top layer , it interacts with particles in that layer, causing some of it to be absorbed and some to be scattered" " this process continues until the incident light either been absorbed or scattered back out of the medium"(ref 12). Kubelka and Munk state that as thickness of a layer increases then the colour of the wall will consist of the tint of the thickest layer of paint without any of the white "subsurface" particles affecting it. The figure below shows how light pass through one medium and scatters between the two layers ,finally exiting from the top layer having gathered colour information from colour particles in both layers.

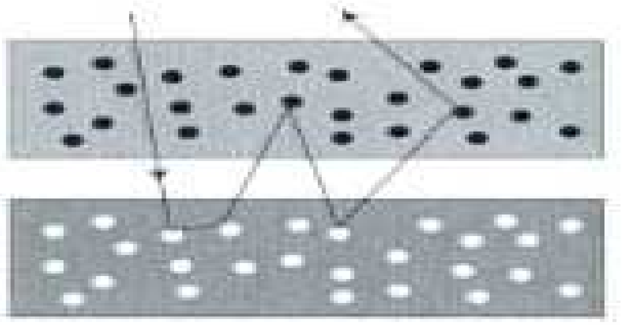


Fig 3(p.42 ref 12)

For the purposes of the improving the Anemone, the Kubelka Munk work is not entirely suitable as it does deal with 'layered' media and as such light scattering has to be computed for each layer. They also make assumptions such as the layer (or layers) involved are homogeneous in terms of thickness and material consistency and that the incident light hits the top layer "uniformly from all directions" (ref 12).

To attain our subsurface scattered look for the Anemone, we needed to focus more on the theory behind the scattering of light through one material or layer. The siggraph paper cited (ref 12)"Advanced Renderman 3" builds upon the Kubelka Munk(KM) and uses the transmission and reflection values that KM provides to further expand how the simulated light values scattering off particles can be attained and used. These light values are the "attenuation" (how much light is absorbed) and the "scattering coefficient" (how much light is scattered) The paper also examines the effect thickness (of the layer) has on these values of attenuation and scattering .

Whilst the KM Model does help to build a solid understanding of the calculations and pitfalls involved in calculating accurate light values passing through and reflecting off a media's surface, A reading of the Henrik Jensens book (Realistic Image Synthesis using Photon Mapping(ref 5)) as well as the 2001 Siggraph paper entitled "A Practical Model for Subsurface Light Transport"(Jensen, Marschner, Levoy, Hanrahan) was needed to gain a real picture of how to model subsurface scattering under "realistic lighting conditions and reflections functions"(ref 12) where "light just comes from just a few directions"(ref 12) and "isn't reflected with a uniform distribution"(ref 12) .

Moving away from light entering the surface for a moment to improve the overall look we cannot ignore surface reflection i.e we must also look at the light which is scattered at the surface and not inside it. The penultimate section of the aforementioned paper, Advanced Renderman 3 "Render harder" (ref 12) aids our research by giving us an overview of light reflection and absorption theory using "bidirectional reflectance distribution function" (BRDF) which models reflected light from a surface.

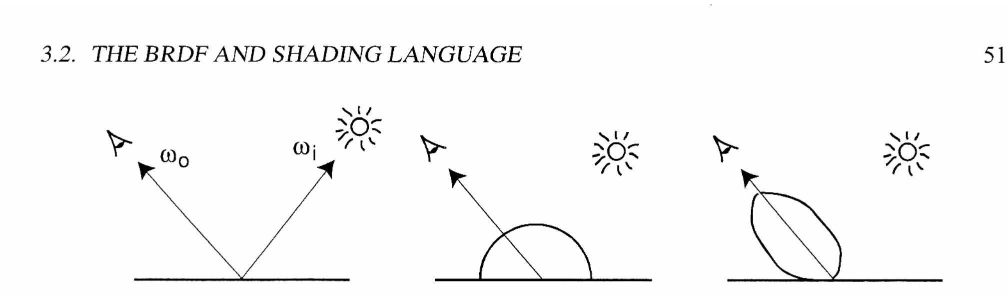
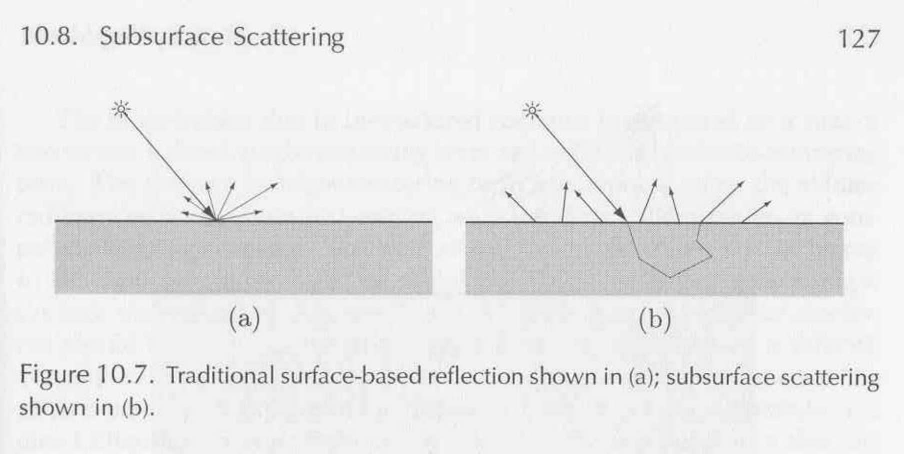

3.2. THE BRDF AND SHADING LANGUAGE 51

Fig 4 (ref 12. page 51 layered media paper 3 types of BRDF (basic , Diffuse, Mirror)

The BRDF function basically models how much light is coming from a certain direction and it determines the outgoing direction to the viewer. The rays of light hit the surface at a spot and leave the material at the same location in a certain direction. The BRDF is "a function of two directions"(ref 12).What controls a surface's BRDF fr value "and information about the distribution light arriving at the surface Li the reflected light to the viewer Lo"(Ref 12) is given by the integration of "incoming light over the hemisphere above the point being shaded" "(Ref 12).
In terms of a mechanism in Renderman to provide the above light calculations they are provided by the 'illuminance' function which gathers all light contributions in turn and using the 'hemisphere' value feeds values to be calculated by the BRDF equation.

The second part of modeling subsurface scattering is to simulate how the light scatters under the surface. This is achieved by "following rays through the layer that sample the layers aggregate scattering behaviour" (Ref 12). This means light enters the surface itself , interacts with particles suspended in the "volume" (Ref 12) and then exits.

To fully attain a picture of how we arrived at complete model of subsurface light scattering simulation, we must examine Henrik Wann Jensen's paper (Ref 13). As well as providing us with a way to scatter light in the surface. Jensen's model develops an extension to the BRDF and phase functions methods (which produce a diffusion approximation of light scattering on and scattering within a surface) and continues on to depict how to allow for light to enter a surface and leave at a randomly scattered point which is different from the entry position and in a random direction.
Thus producing an actual sub surface reflection of light (light is transported and reflected out of the surface at a different point to the entry spot).


10.8. Subsurface Scattering 127

(a) (b)

Figure 10.7. Traditional surface-based reflection shown in (a); subsurface scattering shown in (b).

Fig 5. The picture above compares traditional surface and subsurface scattering.(P.127 ref 5).

Jensen, with his colleagues Stephen R Marschner, Marc Levoy and Pat Hanrahan produced a model for "Subsurface Light Transport" (Ref 13) which covered four areas. Firstly it is an "exact solution for

single scattering with a dipole point source diffusion approximation for multiple scattering". (Ref.13), secondly the model also has an image based measurement technique for "determining the optical properties of translucent materials"
(Ref.13).
Finally Jensen et al describe the sampling techniques that allow the model to be used with a conventional raytracer.The Jensen model improves on the BRDF( Bi-directional reflectance distribution function) which was introduced by Nicodemus (Ref.18). Nicodemus's model relied on single scattering and used a "Lambertian component"(Ref.13) as an "approximation" for subsurface scattering. He does cite previous successful attempts at modelling subsurface scattering (for example Dorsey et al using Photon Mapping) but maintains that "all BRDF models ultimately assume light scatters at one surface point and they do not model subsurface transport from one point to another" (Ref.13).

His model (the BSSRDF-Bi-directional sub-surface reflectance distribution function) unlike the BRDF which assumes that "light enters and leaves at the same point" (Ref.13) uses a method whereby the "outgoing radiance is computed by integrating the incident radiance over incoming directions and area" (Ref 13)

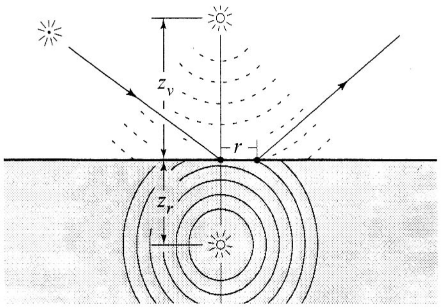


*Figure 3: An incoming ray is transformed into a dipole source for the diffusion approximation.*

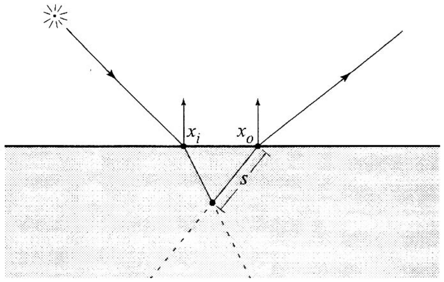
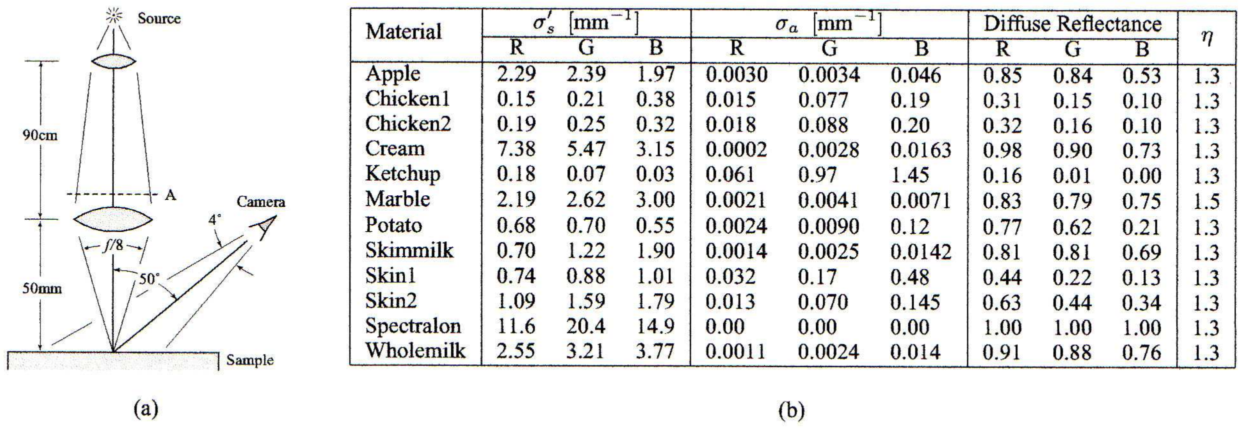


*Figure 4: Single scattering occurs only when the refracted incoming and outgoing rays intersect, and is computed as an integral over path length s along the refracted outgoing ray.*

Fig 6. The operations which are combined in Jensens model for subsurface scattering with BSSRDF

The BSSRDF ultimately is a sum of both the "dipole" multiple scattering (see figure above) element and the single scattering element. Jensen's models works due to the fact that light leaves the medium diffusely, this gives the surface an organic translucent feel. As this is the implementation which is used in Renderman Subsurface scattering tool it was possible to use measurement apparatus table supplied in the paper directly aids the user in choosing a level of translucency which suits his subject

*To appear in the SIGGRAPH conference proceedings*



| Material | $\sigma'_s$ [mm$^{-1}$] | | | $\sigma_a$ [mm$^{-1}$] | | | Diffuse Reflectance | | | $\eta$ |
|---|---|---|---|---|---|---|---|---|---|---|
| | R | G | B | R | G | B | R | G | B | |
| Apple | 2.29 | 2.39 | 1.97 | 0.0030 | 0.0034 | 0.046 | 0.85 | 0.84 | 0.53 | 1.3 |
| Chicken1 | 0.15 | 0.21 | 0.38 | 0.015 | 0.077 | 0.19 | 0.31 | 0.15 | 0.10 | 1.3 |
| Chicken2 | 0.19 | 0.25 | 0.32 | 0.018 | 0.088 | 0.20 | 0.32 | 0.16 | 0.10 | 1.3 |
| Cream | 7.38 | 5.47 | 3.15 | 0.0002 | 0.0028 | 0.0163 | 0.98 | 0.90 | 0.73 | 1.3 |
| Ketchup | 0.18 | 0.07 | 0.03 | 0.061 | 0.97 | 1.45 | 0.16 | 0.01 | 0.00 | 1.3 |
| Marble | 2.19 | 2.62 | 3.00 | 0.0021 | 0.0041 | 0.0071 | 0.83 | 0.79 | 0.75 | 1.5 |
| Potato | 0.68 | 0.70 | 0.55 | 0.0024 | 0.0090 | 0.12 | 0.77 | 0.62 | 0.21 | 1.3 |
| Skimmilk | 0.70 | 1.22 | 1.90 | 0.0014 | 0.0025 | 0.0142 | 0.81 | 0.81 | 0.69 | 1.3 |
| Skin1 | 0.74 | 0.88 | 1.01 | 0.032 | 0.17 | 0.48 | 0.44 | 0.22 | 0.13 | 1.3 |
| Skin2 | 1.09 | 1.59 | 1.79 | 0.013 | 0.070 | 0.145 | 0.63 | 0.44 | 0.34 | 1.3 |
| Spectralon | 11.6 | 20.4 | 14.9 | 0.00 | 0.00 | 0.00 | 1.00 | 1.00 | 1.00 | 1.3 |
| Wholemilk | 2.55 | 3.21 | 3.77 | 0.0011 | 0.0024 | 0.014 | 0.91 | 0.88 | 0.76 | 1.3 |

*Figure 5: (a) Measurement apparatus, (b) measured parameters for several materials.*

Fig 7 Jensen’s table of material parameters

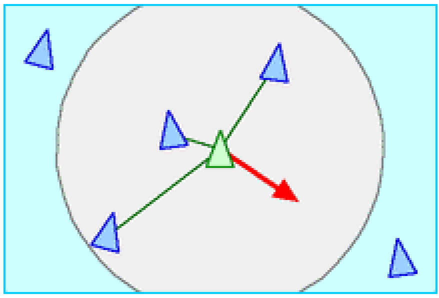
Fig 8. results from uses of Jensen’s model
BRDF(right hand picture)
Traditional shading models gives a
hard computer generated look. (Ref 23)
BSSRDF (left hand picture)
Our new shading model captures the soft appearance
of many natural materials such as skin.(Ref 23)

As mentioned above renderman uses the Jensen model to implement subsurface scattering. The process involves a number of steps which lends itself to experimentation with a single RIB file in order to achieve good positioning of lights for the desired effect. This will be examined in more detail in the shading section regarding the anemone below

## 3 Technical background

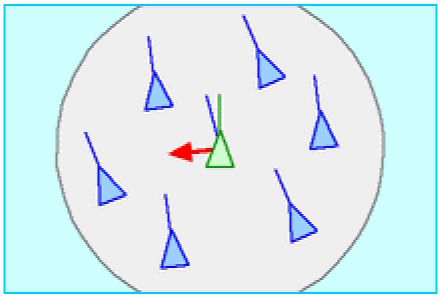
A decision was taken to add a greater element of animation into the scene through the addition of a flock of fish. The algorithm chosen to execute the flock was Craig Reynolds (ref 22) flocking algorithm and was combined with a use of an overview of possible implementation algorithm by Conrad Parker (Ref 22) In the aforementioned source Reynolds outlines the parameters for a true flocking system. These consist of three rules, each member of the flock or “boid”(Ref 21) must obey to maintain a flocking movement. These rules (Separation, Cohesion, Direction ) are outlined below supported by images from Reynold’s site

Fig 9. Separation
This rule states that each “boids”- (fish in this case) must avoid contact with other members of the “flock”



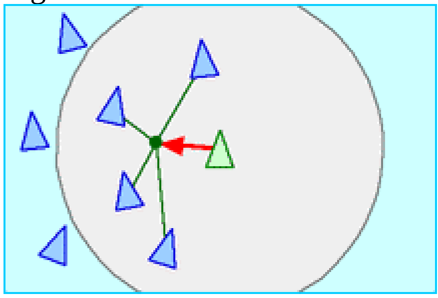
Separation: “steer to avoid crowding local flockmate” (Ref 21).

Fig 10.Cohesion
This rule states that all members of the flock must stay within a certain distance of each other,



Alignment: “steer towards the average heading of local flockmates”(Ref 21).

Fig 11.Direction



Cohesion: “steer to move toward the average position of local flockmates”(Ref 21)

In addition to the Reynolds source ,Conrad Parkers Site contains solid Algorithmic pseudocode which outlines each rule in a straight forward fashion, an example of which(rules 1-3) is shown below

```
Rule 1:
      PROCEDURE rule1(boid bJ)

            Vector pcJ

            FOR EACH BOID b
                  IF b != bJ THEN
                        pcJ = pcJ + b.position
```

```
            END IF
      END

      pcJ = pcJ / N-1

      RETURN (pcJ - bJ.position) / 100

END PROCEDURE


Rule 2:
      PROCEDURE rule2(boid bJ)

      Vector c = 0;

      FOR EACH BOID b
            IF b != bJ THEN
                  IF |bJ.position - b.position| < 100 THEN
                        c = c - (bJ.position - b.position)
                  END IF
            END IF
      END

      RETURN c

END PROCEDURE


Rule 3:

PROCEDURE rule3(boid bJ)

      Vector pvJ

      FOR EACH BOID b
            IF b != bJ THEN
                  pvJ = pvJ + b.velocity
            END IF
      END

      pvJ = pvJ / N-1

      RETURN (pvJ - bJ.velocity) / 8

END PROCEDURE
```

Fig 12. pseudoCode upon which to base a flocking system (Ref 22)

The rules outlined in the pseudocode basically follow the images(fig 9,10,11) shown above. The major difficulty in implementing the algorithms shown above was the translation into MEL code and the exportation of animation information into the next stage of the pipeline, namely the Houdini environment for application of shaders and output to RIB for rendering. It was noted that the learning curve involved in switching between a compiled language such as C++ and using an interpreted language such as MEL (with all is individual idiosyncrasies) was quite steep. As will be discussed below in the “Fish” animation section, the above pseudo code sources were implemented, added to and extended to suit the projects needs and the requirement of software such as Maya.

## 4 Solution

To restate the objective of the project. The aim was to achieve a realistic surface for the Anemone creature, this aim primary aspect is replicating the way the Anemone’s flesh allows light to partially pass through it. Other aspects are the stippled nature off the anemone surface and it’s “Jellyfish” -like undulating motion. The other aspects of the scene support the Anemone as the main focus of the composition. The rock Fish terrain present challenges in shading methods and were an ideal way of learning variety of shading mechanisms The choice of an underwater scene was partially because of the variety of colours and textures found in ocean scenes. The main thrust of the project was not entirely on realism but more effective use of the right shading mechanisms for the variety of shading challenges encountered.

### Anemone

#### *Modelling*

The Anemone was modelled in Maya due to familiarity with the modelling toolset which ensured a faster construction of the creature’s body. The picture below provided the guide whereby the creature was modelled

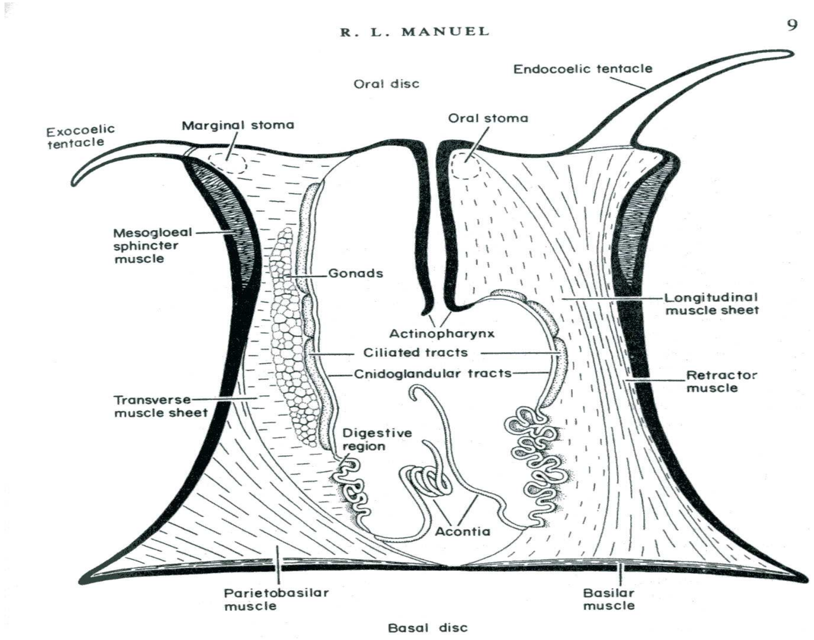


Fig.13 Picture from British Anthozoa (RL Manuel)

The Anemone itself was positioned as much as possible in the screen area so as to be the main focus but not obtrusively dominating the scene; it draws the viewer’s eye to it whilst not drowning out other subjects in frame. Use of the technique known as the “golden triangle” was employed to obtain an ideal position on the screen.

The colour scheme was chosen to match as closely as possible the image below (fig .14)

Fig.14 Sea Anemone

RGB values used closely adhere to the mixture of pink white and blue that constitutes the creature’s surface

*Shading*

The colour basis for the Anemone was formed from subtracting various levels of pink from the base pink colour (which was derived from the derived from the source image creating tones of above and below the base colour. These levels were then fed into a spline function and combined with a shade of blue and white (from a mix function with a higher proportion given to the white element). The pattern of pink and white/blue dots was built using a function sourced from “Essential Renderman Fast” (ref.8) with the ratio of “disk” patterns increased to match the number of dots on the source image. The main focus of the shader as mentioned before was an effort to attain the slightly translucent nature of a soft tissue organism like the Anemone. The first method experimented with is the method proposed by Felipe Esquivel (Titus) which basically takes into account in shading calculations the normal on the other side of a thin surface (-nf) and multiplies it by a second “Kd” factor –namely Kd2 see ref (17) which is

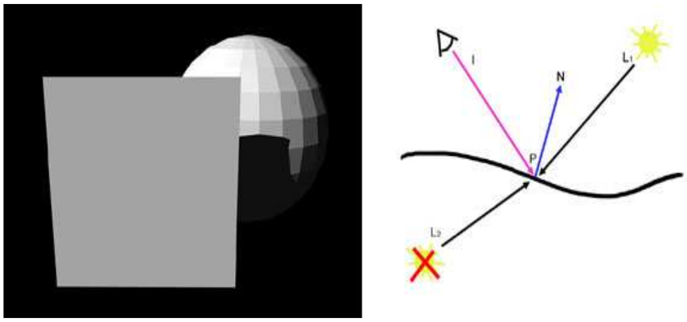


Fig.15 no light is taken into consideration from the opposite side of the plane

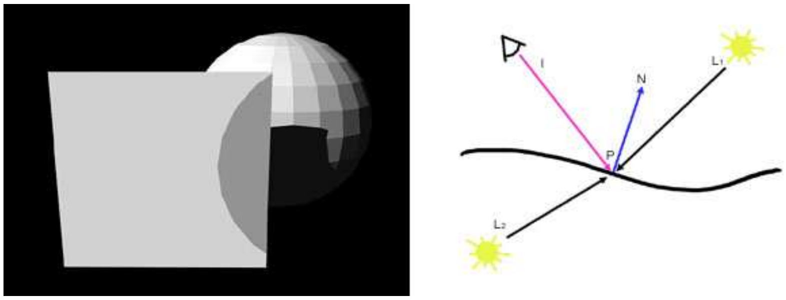


Fig.16 now with the inclusion of the “-nf” normal

Fig.17 the “–Nf” method of depicting light through a thin plain in Action in “A bugs life” (Ref 17)

However it was found that this method suited primarily thin planes and was ineffectual in producing a translucent effect. The fresnel function was also applied (using an ior (incidence of refraction) factor of 1.33 (the setting for water. This however became obsolete as fresnel is a factor which can be applied during use of Renderman’s implementation of subsurface scattering.
A last method used was a technique which controls the opacity of the surface depending on the angle of view. This was achieved with the creation of a vector (vector II) and a new Normal (NN) and controlling the Opacity of the Surface using the absolute value of the dot product of the two elements. However as mentioned above
Subsurface scattering proved to be the method which most suited the project’s needs and this was the method finally settled on to provide the desired effect.
The process for producing Subsurface scattering involves several steps, the first being the production of a point cloud which is in turn utilised by a “brick” map. The aforementioned point cloud is produced after a “bake_radiance_t” is applied to the geometry. This shader “bakes the diffuse surface transmission of direct illumination into a point cloud file”(Ref 24) It is paramount to choose the right options when applying “bake_radiance_t” to geometry ,most notably to set the “dicing” parameter to the area function to get the face of each micropolygon instead of each shading point(which would be undesirable). The colours “irrad” (computed direct illumination, ambient and diffuse) and “rad_t” are produced (rad_t is a product of irrad and Kdt-the surface colour) then “radt” can be passed into the bake3d function to produce our point cloud. What is actually occurring is that at the point that the program “ptfilter” is run a diffuse approximation is performed on the point cloud. This diffuse approximation is derived from the model discussed earlier which provided by Jensen. Using Jensen’s parameter table (see fig. 7) the user can choose scattering values and absorption values which match

some generic materials. The values chosen for the project are shown below

```
ptfilter –ssdiffusion –scattering 0.894, 0.655,0.957 –absorption 0.0002, 0.0028, 0.0163 –ior 1.5
direct_rad_t.ptc ss_diffusion.ptc.
```

Fig 17.The Anemone after application of “read_ssdiffusion” shader and rendering

As can be seen from the figure above a strong back light was used to provided  the light used in the subsurface calculation. It was necessary for lights (with pointLight_Rts light shader applied - an adapted light shader which check the diffuse depth and shoots rays based on the sample level –if it 0 shoot only one ray, if it is more than 0 shoot as many rays as allowed by the samples parameter) to be placed in different positions than the three point arrangement used for the beauty pass. Intensities were experimented with and distances changed until an intensity of 28 for the front light and an intensity of 120 was chosen for the main back light (the third light present was turned off as it was found to be creating a sharp white line on the right side of the model).

An adaptation was made to the read ss_diffusion shader (suggested by course tutor Jon Macey) whereby noise was applied to the direct illumination calculated from Ka multiplied by ambient and Kd multiplied by diffuse. This noise consisted of noise of PP(the shading point) pushed by 0.5 and noise of PP multiplied by 2.
This had the effect of producing an octopus like dappled colour to the subsurface effect which lent a more organic less ordered feel to the subsurface colour produced.

*Displacement*
Initially during the major animation section of development the displacement followed the even arrangement of depressions on the surface of the anemone. Two separate Fractal Brownian motion (fBm) functions with different amount of cycle levels (one 6 and one 9) were added to the smoothstep function which made the border between those areas displaced and not more jagged and realistic in its slightly chaotic nature.

*Animation*

The animation of the Anemone body was achieved using a “soft body transform” in Houdini. The effect of the transform is animated in the “Y” field using an expression to give it a predictable motion (See Appendix 1). To give an undulating action to the Anemone body, the top of the Anemone body was given an opposite action timing wise to the midsection of the anemone. To achieve this, the multiplication value by which the midsection is guided is changed and re-applied to the top part offsetting it just enough to give a motion similar to the jellyfish propulsion movement. This adds to the illusion of an organic living entity. The two equations for the soft transforms are shown below the first controlling the mid section animation and the second controlling the top part of the Anemone body or “Column”. A “soft transform” is a tool which is part of the Houdini toolset and it applies a lattice type transformation to the geometry, this lattice action had the equations applied to its scale parameter to give the swelling motion.

1st soft transform 0.1*((sin($F*6)/2)+9.8) , 2nd soft transform 0.1*((sin($F*7)/2)+9.8)
Equation 1. The two soft transform equations

As can be seen the second soft transform equation is offset($F*7) instead of ($F*6) to give complimentary movement of undulation

## Terrain

*Modelling*

The Terrain was modelled in Maya using an intuitive “Polygon Sculpt” tool and exported as an OBJ file. An attempt was made to have the contours of the surface gather towards and around the Anemone, adding to the framing effect and drawing the viewer’s eye to it.

*Shading*

The Shading scheme chosen consisted of two pairs splines (one of tones of blue and greens and one of tones of blue) these two splines which were then mixed using a voronoi function to create subtle patchwork. A noise factor over the top was added over the top to give some random quality to the overall colour.

*Displacement*

A higher level of control was needed to facilitate greater artistic freedom in applying the different frequency levels of noise provided by the Fbm function. These frequencies were organised into set clamped divisions that is upper, middle and lower. These divisions are clamped by the RGB values passed into the shader by a image produced by the user. This constitutes a more flexible solution to displacement application than the previous approach taken which consisted of an application of displacement all over the surface. A pleasing mix of displacement frequencies are produced as the displacement fluctuates depending on the purity and mix of the RGB values. A sample of the imaged used is shown below. It is a simple circular pattern which was repeated and blurred (it was chosen due to the rich variety of RGB values in it).

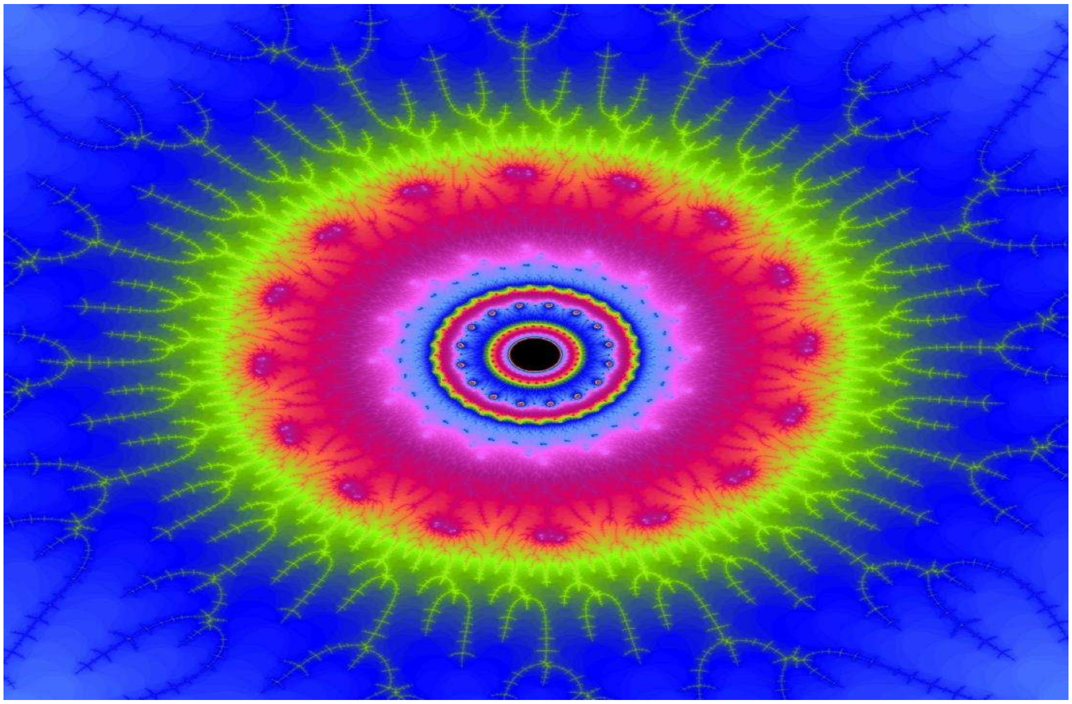


Fig 18. Basic pattern used in the Terrain Displacement

## Tentacles

### *Modelling*

The tentacles were modelled in Houdini using circles which were skinned to appear like tentacles.

### *Animation*

The circles were animated using a repeating equation to simulate a writhing like movement.

### *Shading*

The shader itself was shaded with a simple ramp in s faded with a smoothstep. As discussed above varied translucency methods were experimented with to gain a soft translucency (the "kd2" method and IOR mentioned above) ultimately it was decided to apply the read ss_diffusion shader produced by the Renderman subsurface scattering tool and include it as a subsurface scattered object.

### Displacement

No displacement was added to the Tentacles as it would have detracted from its soft tissue like look.

## Rock left of frame

### *Modelling*

The rock was modelled in Maya and exported with the rest of the scene in an OBJ.

### *Shading*

The original design for the rock that existed during the Major animation project consisted of rows of rings of alternating colours, the ring shape being perturbed by turbulence.

The break up of the rings was deemed not natural enough and it was decide to apply repeated levels of noise to the "s" value to break the pattern in a radical way, making it resemble more real rock formation. The shader values in "t" are also affected by a noise value which was multiplied by 8 to heighten its impact. The colour values consisted of a noisy overlay which is a mixture of blues and greens. The lower colour is a spline which uses a noise filtered cycling value which is based on a sin value of( 2*PI *s ) multiplied by 2*t this ensures a noisy but cycling colour value.

*Displacement*
A simple modulus value in t was used to create a subtle undulating effect, this was scaled back to a very subtle level for the final render.

Rock -right of frame

*Modelling*
As stated above the rock was modelled in Maya and exported with the rest of the scene. in an OBJ format
*Shading*
The design used in the Major Animation project was changed completely. Whereas it was an even light spotted colour before, a decision was taken to utilise the voronoi function's ability to synthesise cell patterns found in rocks naturally (see fig below)

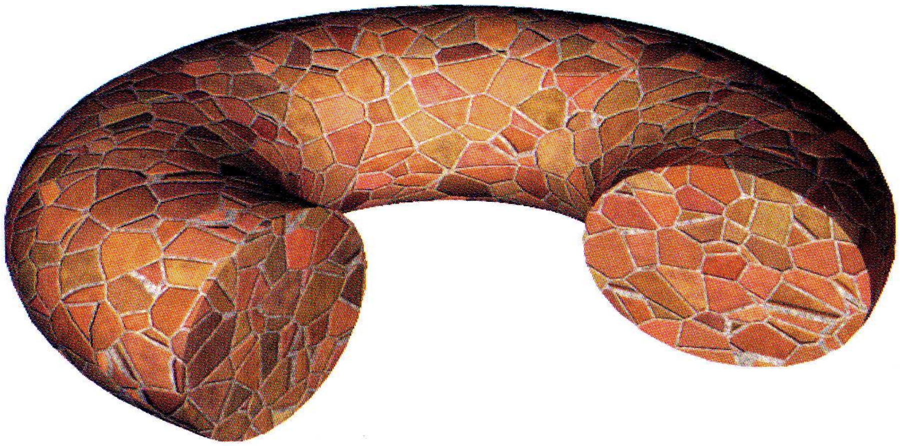


FIGURE 4.7 3D flagstone texture shows Voronoi cells.

Fig 19 Voronoi Pattern image from "Texturing and modelling – A procedural approach"(Ref 7)

The voronoi pattern is defined starkly by dark borders shows the sections where the distance between the first closest point and the second closest point arrive at a crossover point these tow points (f1 and f2) are subtracted leaving the difference which defines the black border. The colours within the cells are two sets of spline colours one made of several tones of green the other of several tones of orange and brown. The moss is controlled by the hump value passed in from the displacement value which is assigned to the rock. The black voronoi cell border value is offset from the displacement voronoi cell border value (and its accompanying moss) to give the rock a less planned synthetic feel. There were initially problems with the fBm noise values on the borders of the voronoi cells aliasing quite a lot these problems were solved by using filtered fBm function which keeps the levels of noise within the Nyquist region so that unpredictable reactions do not occur. The shader contains an if statement which check the height of the displacement value passed in to set the "green moss" value over the surface. Once the height of the displacement is ascertained the surface colour "surfcolor" and the colour Ct are blended , with the amount of moss allocated by the displacement height. If the displacement is set to a very high level the "moss" will cover the entire surface. A scaler value was also included so that the width of the border for the voronoi pattern could be set.

### The Fish.

*Modelling*

Nurbs were used to construct the overall shape and then sculpted and smoothed once converted to polygons. The Figure shown below was used as a guide for modelling purposes

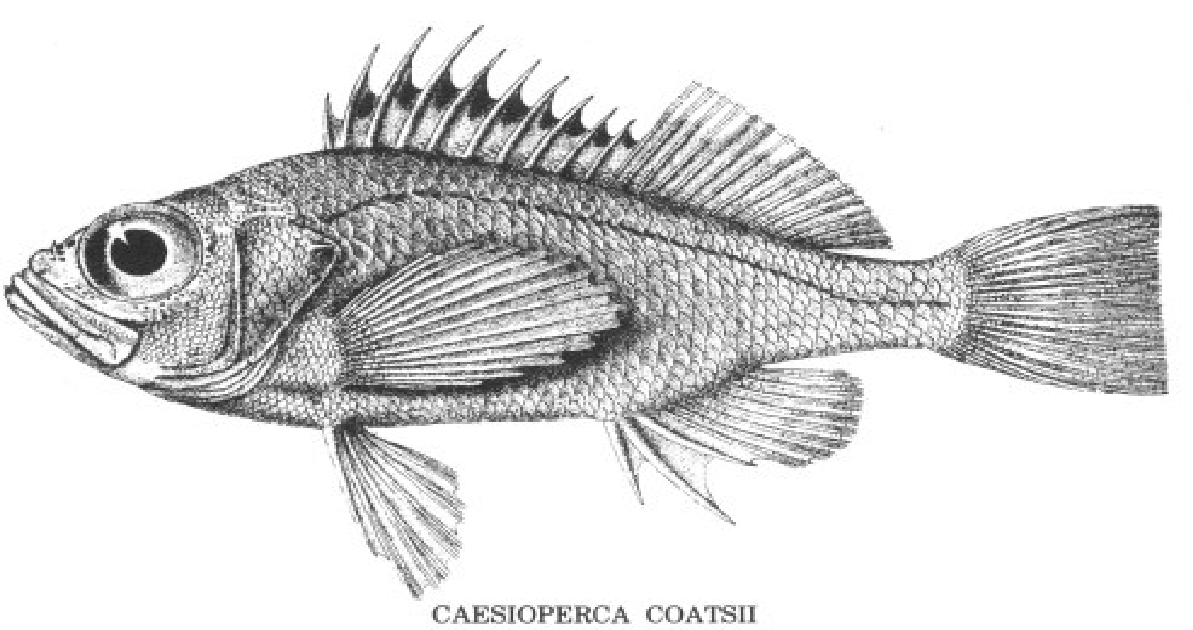


Fig 20. The image used as a reference guide for the fish model.

*Animation*

In addition to the compositional devices such as the orientation of the terrain and the rock orientation, the scene overall needed some movement which helps to give more cohesion to its composition and makes for a more dynamic viewing experience. The addition of the fish literally drags ones view to the anemone. The fish, in compositional terms "point" to the Anemone.

From a technical perspective adding the flock of fish was also an opportunity to learn a larger amount regarding the MEL scripting language and also Maya's implementation of expressions to simplify animation producing the flock gave a good insight into how MEL and expressions can be used in concert to produce automation in animation. The flocking algorithm originates from the three algorithmic rules set down by Craig Reynolds (namely separation, cohesion and direction).

The expression itself creates three vectors (PCX,PCY,PCZ) (CX,CY,CZ) and (PVX,PVY,PVZ)and fills them with values which encapsulate the three rules of flocking.
PCX…will be the guiding vector which makes all the Fish "try to fly towards the mass of the neighbouring boids"(Ref 22).
CX…will be the guiding vector which makes all the fish "try to keep a small distance away from other

objects (including other Boids)(Ref 22)
PVX ….. is the vector which makes all boids try to match each others velocity.

In addition to this a goal was added (meaning the Locator in the scene) which the flock of boids would generally try to move towards as the tried to obey all other rules.
Some difficulty was encountered in making the boids turn towards the locator as it moved around the screen which was overcome by using a “atan” calculation regarding the locators vector position (see included code) to give the Boid part of the fish hierarchy the proper “Y” rotation (to point towards the locator).

All the above vector values were added to an attribute called “boid_velocity” was added to the attributes of the fish’s hierarchy and thus gave the flock an automated yet controllable movement (using the locator as a guide).

The animation was produced initially in the original scene in Maya subjects were modelled, which mirrors the Houdini layout to maintain continuity between the fish and their surroundings. The resulting flocking motion was then exported to Houdini as “chan” file animation
A quick pipeline overview
The pipeline used to produce the animation of the fish was in general terms .
*Stage 1*
Fish modelled in Maya and skinned using a soft Bind
*Stage 2*
A simple interface was designed whereby the user enters in the number of Fish to be made. This is called using the command Fish_UI_i; in the command line. Or script editor.
*Stage 3*
Once the Button is pushed the fish are duplicated to the amount specified and arranged in a random fashion.
*Stage 4*
Once the Fish were duplicated the remainder of the script executes and selects a part of the skeleton (namely a joint called boid) and sets a lengthy expression onto the joint the expression encapsulates the three flocking rules referenced above and controls the flock’s movements according to these rules.
*Stage 5*
The flock of fish follows the locator in the scene named as locator1.
*Stage 6*
Once the user is happy with the flocking movement two scripts are utilised to export the geometry in OBJ format.
*Stage 7*
In Houdini the Obj Files are taken in using a file SOP and re-exported to a BGEO format for the reason of Houdini’s ability to read BGEO files more quickly.
Stage 8
Shader are applied and Renders take place.

The animation on the fishes tail itself was produced using a simple system of applying keys to the rotation of the y axis for each joint then delaying the effect over time to each consecutive joint , this resulted in a tail whipping effect similar to the sinusoidal effect that fish use to propel themselves.

*Shading*

The Fishscales represented a challenge regarding the mathematical arrangement of the scales, each row being offset from the one above. This issue was solved by using a two separate equations which produced an ellipsoidal shape and then manipulated that equation

```
float ss_a =(-sqrt(1-4*((ss-0.5)*(ss-0.5)))+1);
```

The code above (if used with straight values of s and t) produces an ellipsoid, so when combined with a modulus it repeats that ellipsoidal shape in a scale fashion.

```
float ss_b =(-sqrt(1-4*((ss_t-0.5)*(ss_t-0.5)) ) + 1);
```

The code above produces an alternate half ellipsoidal shape (so that when placed end to end the produce an interlocking type scale similar to the alternate arrangement of scales below
Combining these two ellipsoidal equations with modulus functions and then utilising a floor function to serve as a way of choosing which configuration of scales to lay down in each row leaves us with a rather pleasing scale pattern (see included shader code).

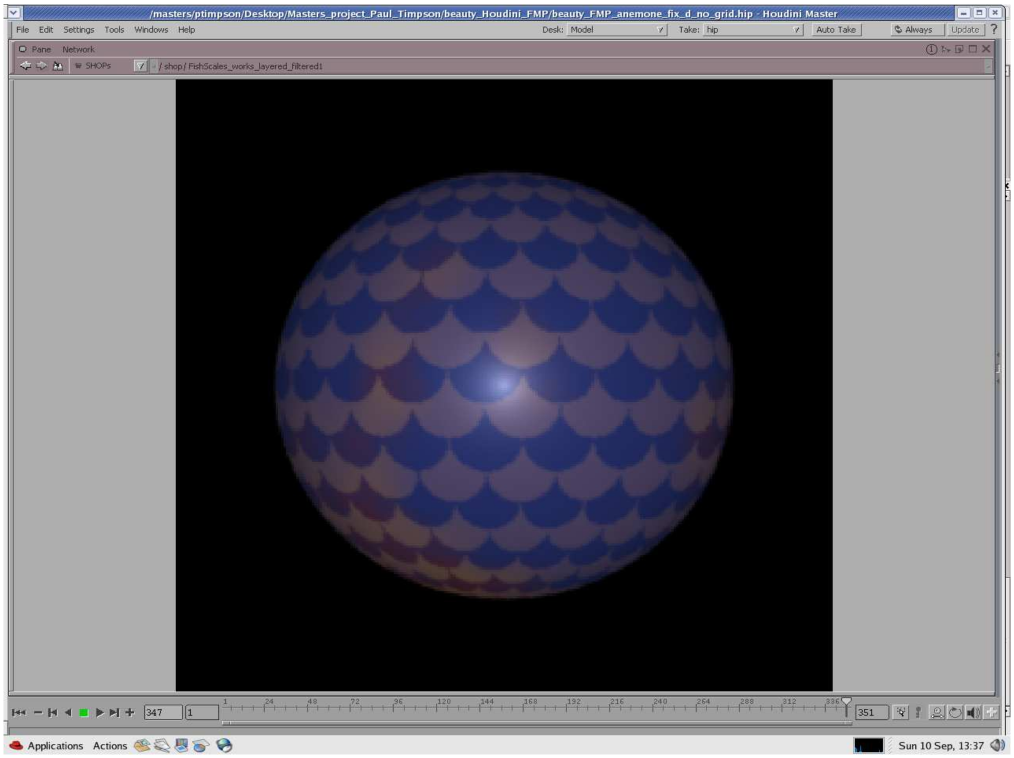


Fig 21 A screenshot of the finished procedural scale shader

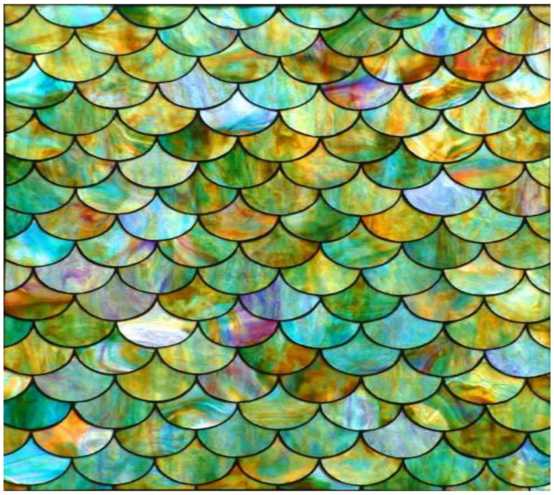


Fig 22 the arrangement reference for the fish scales

## 5 Atmospheric and lighting effects

### a) Fog

In the original version used in the major animation project, the use of static fog was influenced by time constraints as to rectify this it became one of the focal points of the overall project.  Upon viewing of underwater footage the need for “broken” up “moving” fog was deemed paramount if a realistic atmosphere was to be attained

A solution utilising particles was hit upon to provide a controllable spread of moving broken fog. Particles were produced in Houdini and a sprite shader was attached to supply a texture guide to be used in the fog shader. The method of how the fog moves is an illusion as basically consists of where ever the particles are rendered the opacity of the fog is at its most transparent giving an effect of movement as these “particle holes” moves through the otherwise dense fog. This was precisely the main problem with fog, it denseness. A series of experiments were tried to improve the situation one was to increase the particle amount to a large level but then the smoke effect was almost lost entirely) , the solution settled on was to multiply the effect of the texture value. This however only cleared clearly defined holes in the fog which were too obviously spherical to be natural. Time permitting more experiments will be tried in the future to improve on the fog model, whilst a lot was learned, in using renderman camera projection and utilising Houdini’s Particle capabilities, the overall result too obtrusive into the scene and occluded most of the subjects too completely.

### b) Uberlight

An Uberlight was used to produce flickering effect on the water above the scene, the texture was provided by projecting the voronoi texture (see below) onto a plane in Houdini and then animating the cells of the voronoi to produce a flickering effect. The moving voronoi texture sequence was then attached to the Uberlight to be projected onto the water plane and then composited in as a separate pass.

### c) Ambient Occlusion - Straight AO and Image based Illumination

Both straight forward Ambient Occlusion and Image based ambient Occlusion were used in the project adding soft shadows to the terrain and giving a lot of atmosphere to the piece. The Ambient Occlusion was set to 64 samples with a Pi/2 range. A shader which efficiency performs AO was downloaded from the internet (see Ref 25) and utilised to produce the occlusion pass. The ambient passes it was noted took more time than any other pass. Image based Illumination was slightly more involved in the fact that shots of the environment had to be taken from 6 different directions from a central point in the scene and once suitable shots had been acquired use was made of Txmake to produce an environment texture. This was applied to all geometry in the scene and a colour bleed pass was attained.

Fig 23 Voronoi texture which was animated and projected onto the water plane

Finishing touches

1) As was suggested by fellow students, some camera movement was added to give the viewer the impression that the viewers viewpoint is not stable as the viewer would be underwater where one would be buffeted by tides and currents

2) Sounds which resemble bubbling gurgling noises were layered over the images using to give an extra atmosphere to the scene.

3) A small script was written to automatically render files in sequence using renderman (as problems arose rendering whole sequences using renderman) - included on cd.

## 5 Problems:

The main problem areas encountered were in getting the Subsurface scattering mechanism in Renderman to work properly. As the model began life as a multi layered polygon model it obstructed the subsurface operation at first. It was not until the model was culled of inner polygons that the Subsurface Appearance began to come into full effect.

The second problem concerning the Anemone model was the choice of parameters given in Jensen’s guide as to varied translucent qualities of different materials( such as milk ,skin) . The variety of parameters available to the Subsurface mechanism in Renderman caused some confusion and lengthy experimentation until an absorption value of 0.0002, 0.0028,0.0163 and a scattering value of 0.894,0.655,0.957(this is the RGB value which gives the anemone Subsurface pass its pink tinge. An IOR value of 1.5 was also chosen.

Another problem area encountered was concerning the flocking system which required a lengthy time to implement. At first a straight procedural approach was adopted and the three rules implemented in this Fashion using the advice of tutors and basic code from an approach created by Adam Vanner. A setback was encountered when it was discovered that a lack of per frame responsiveness even when the command “scriptJob” was used with the “timeChange” flag was employed to monitor when the frame changed, it was found that several frames passed without an update and only upon the animation ceasing action that the scene updated. Upon discovering this, a decision was made to develop a purely expression based approach to the flock system which will be described below.
An additional problem area was the displacements employed in Houdini. Any displacement more than the most minimal proved to be a major issue causing unsightly blotches all over the surface (The “terrain” object was the main problem area). This was resolved by placing any problem displaced objects in their own object, setting the displacement bounds to suit the individual displacement on the object.

The flock posed problems due to the automated nature of the motion of the fish which wasn’t entirely predictable causing some intersections between fish and terrain. A lengthy period of time was spent adjusting the influence of the guide locator on the fish until a happy medium could be found whereby fish stayed at an acceptable distance from all obstacles
The most challenging procedural shader was the scale shader as it involved a combination of several offset setting for s and t until interlocking ellipsoidal.
As well as the above the “moss” on the rock right of screen was initially the source of much aliasing (the border of the voronoi pattern flickering on a per frame basis). After much manipulation of the frequency values this problem was solved.

The Fog proved to be a technical success in the fact that the method to produce it involved a large amount of time to research and implement (the method being the camera projected shader) Also the approach of using particles rendered from Houdini proved a success(a lot of knowledge regarding the implementation of . The particle rendered images were effectively used to control the opacity of the fog produced by the shader .However the resulting fog (although patchy as planned ) was too heavy in consistency and occlude too many of the primary pieces of the scene the main one being the translucence of the Anemone body.

A last minute problem was movement in the filtered noise values used to add to the Fish scale colour, even though filtered and the point used to create the fBM was placed in the correct space the fBM continued to behave erratically, this will be examined in the coming days

## 6 Conclusion

The project posed several problems due to pipeline issues with Houdini notably with getting geometry into Houdini and getting the same exact shape produced in the original modelling package(hole would appear in the model and levels of faceting had to be adjusted in a tedious manner. Upon reflection it would seem to have been wiser to remain within the Maya environment regarding producing the shader (use MtoR) but once forced to write shaders by hand for import into Houdini it forced the necessity of gaining a greater understanding of the mechanics of shaders which will be beneficial in the future. Also upon reflection time could have been spent adding controls to the interface which was left at stage of functionality rather than a complete user friendly state.
The Anemone subsurface scattering which took a large amount of project time ended with a pleasing result. The interaction of the Anemone body displacement stretching movement and the variety of tissue over the anemone body gives it a convincing organic feel. This and the success in improving both rock shaders, terrain as well as implementing a flocking system in MEL proved to be the most

gratifying aspects of completing the project.

## *7 References*

1) The Blue Planet, Authors: Andrew Byatt, Alastair Fothergill and Martha Holmes
BBC Books. Publisher-BBC Worldwide Limited ISBN 0563384980: Published 2001.

2) Advanced Renderman “Creating CGI for Motion Pictures”, Authors Anthony A. Apodaca, Larry Gritz . Publisher-Academic Press ISBN 1-55860-618-1: Published 2001.

3) The Renderman Companion, “A Programmers guide to Realistic Computer Graphics” Steve Upstill Publisher-Addison Wesley ISBN 0-201-50868-0: Published 1990.

4) Siggraph Course Notes 25 , Chairs Larry Gritz, Tony Apodaca; Monday August 9th 1999 copyright Pixar 1999.

5) Realistic Image Synthesis Using Photon Mapping, Author Henrik Wann Jensen
(foreword by Pat Hanrahan). Publisher-A K Peters ISBN 1-56881-147-0. Published 2001

6) Rendering For Beginners Image Synthesis using Renderman, Author Saty Raghavachary. Publisher-Elsevier ISBN 0240519353 Published 2005.

7) Texturing and Modelling “A Procedural Approach” Third Edition, Authors David S Ebert, F Kenton Musgrave, Darwyn Peachey, Ken Perlin, Steven Worley.Publisher Morgan Kaufmann
ISBN 1-55860-848-6 Published 2003.

8) Essential Renderman Fast, Author Ian Stephenson ISBN 1-85233-608-0, Publisher Springer; Published 2003.

9) The Creation and Simulation of Jellyfish Swimming , Thesis Author David Walker b1368495 NCCA Bournemouth.
10) Anthozoa Sea Anemones, coral, Sea Pens Http://www.tolweb.org Authors Daphne G Fautin, Sandra Romano posted 04 October 2000.
11) Anatomy of Coral, Http://www.marinereef.org/reports Author Unknown ,Posted date Unknown

12) Advanced Renderman 3 “Render harder” Siggraph 2001. Course 48 notes; Course Organizer Larry Gritz Exluna Inc August 2001.
Lecturers
Tony Apodaca, Pixar Animation Studios, Larry Gritz, Exluna, Inc, Matt Pharr,
Exluna, Inc. Christophe Hery, Industrial Light + Magic, Kevin Bjorke, Square USA
Lawrence Treweek, Sony Pictures Imageworks.
Chapter 3 Layered Media for Surface Shaders- Matt Pharr.
Chapter 2 A Practical Guide to Global Illumination

13) A Practical Model for Subsurface Light Transport , Henrik Wann Jensen, Stephen R Marschner Mark Levoy Pat Hanrahan Stanford University Siggraph 2001 Proceedings papers.

14) David Gould Mel scripting for maya animators ISBN

15) David Gould complete Maya Programming ISBN

Thanks:
Thanks to Jon Macey Ian Stephenson, Phil Spicer and all the staff at the NCCA ,
also thanks to Mario Auseloos, Pete Bowmar and

16) The Mel Companion Maya Scripting for 3D Artists – David Stripinis published 2003 ISBN 1-58450-275-4

17) “Translucent objects in Renderman” By Felipe Esquivel (Titus)
http://www.deathfall.com/tutorials.php?op=showcontent&id=73 accessed July 25th 2006

18) “Geometric considerations and nomenclature for reflectance” F.E Nicodemus, J.C Richmond, J.J Hsia,I.W.Ginsberg and T Limberis.
Monograpah 161, National Bureau of Standards (US) October 1977.

19) “British Anthozoa” R.L Manuel “Synopses of the British Fauna” editors Doris M Kermack and RSK Barnes Publishers – E.J Brill/Dr.W.Backhuys for the Linnean Society of London & the Estuarine and Brackish-Water Sciences Association.
ISBN 9004 085963

20) http://www.fofweb.com/Electronic_Images/onfiles/SciAniAnat2-15c.gif accessed Ausust 24th 2006 Author :Unknown

21)http:://www.red3d.com/cwr/boids/ :accessed july 20 2006, “Boids Background and Update” Author Craig Reynolds Accessed July 1 2006.

22) http:://www.vergenet.net/~conrad/boids/pseudocode.html : accessed 19 july 2006
Author Conrad Parker Accessed July 1 2006.

23) http://graphics.ucsd.edu/~henrik/papers/bssrdf/ -Authors Henrik Wann Jensen, Steve Marschner, Marc Levoy, Pat Hanrahan. Accessed 15 August 2006.

24) Renderman Translucency and Subsurface Scattering (application Note #37) October 2002 (Revised August 2005) Author, Unknown.

25) www.fundza.com “Global Illumination” Author Malcolm Kesson

Appendix
1. Equation for Anemone body movement.
Sin (FF)/100