# Realistic Underwater Visualisation

Matt Aranha*

Computer Graphics Group
University of Bristol
Bristol / UK

## Abstract

It is increasingly possible to render high fidelity 3D computer scenes in a reasonable time on modern computers. Accurate models and high fidelity visualisations of many natural environments are now possible for a wide variety of applications, including computer games and training simulations. This paper discusses a proof-of-concept of a pipeline for modelling and rendering of physically accurate virtual underwater environments. The following work is based on the three-dimensional modelling application *Maya*, and the *Radiance* suite to provide high fidelity lighting visualisation of underwater environments for possible training scenarios.

**Keywords:** Global Illumination, Rendering, Ray-tracing, Underwater

## 1 Introduction

Due to the costs associated with deep-sea underwater exploration and engineering, it can be beneficial to understand, using a computer simulation, an underwater environment prior to undertaking any operation. In such a computer simulation it is essential that the computer graphics are physically accurate as failure to do so could lead to significant problems due to the dangerous nature of such environments. Authentic underwater rendering could be used to allow divers and submersible ROV[1] pilots to predict underwater visibility and analyse optimum equipment and procedures for a range of underwater conditions. Furthermore, recent attempts at underwater scenes within the entertainment market, although believable, are of limited physical accuracy and are often time consuming to create. A streamlined system for creating underwater scenes for this market would allow accurate underwater scenes to be created efficiently.

Alias' *Maya* is a 3D modelling package used to produces complex 3D models. It has been widely used in many high profile films, games and visualisations [25]. *Radiance* produces final rendered images, it uses ray-tracing to create highly accurate lighting visualisations [26]. This paper presents a method of faithfully recreating underwater environments using these two applications as well as other advanced rendering techniques. The system's input is to be based on a set of observable parameters. The initial focus will be purely on static images, with a view to developing a later interactive version.

The rest of this paper is organised as follows. Section 2 covers a few of the main considerations in accurately recreating an underwater environment. Existing software solutions for creating underwater scenes are looked at in Section 3. Section 4 concentrates on the implementation of the proposed system. Results of the works are presented in Section 5 and conclusions of the system are given in Section 6.

## 2 Underwater Environments

The density of water is approximately 800 times greater than air. The consequence of this is that light interacts differently in this environment and thus underwater environments require a special rendering approach. Interaction with water molecules and suspended particles cause effects, including loss of contrast, diffusion of rays, change in colour and reduction of intensity. The perceived colour of the sea is largely to do with the biological state of the water [19] [20]. Light scattering causing optical effects such as caustics and shafts of light due to atmospheric particles is also an essential consideration for any underwater scene. The water surface has a great effect on light below the surface. At the surface, light can be both refracted and reflected. The gentle waves in a water surface produce the caustic effects seen at the bottom of the ocean where light has been focussed to produce a variation of light intensity. In addition, flat surfaces cause a mirror effect with more light reflected than during rough conditions.

When photons enter water, a complex chain of absorption and scattering takes place. This is ruled by the *radiative transport equation* (1) which characterises the radiance change along a path in the form of an integro-differential equation as a function of radiance $L$ at point $x$ in the direction $\overrightarrow{\omega}$ with absorption $\alpha$ and scattering coefficient $\sigma$ [12]. An accurate simulation needs realistic and efficient water surface and light interaction models. These account for caustics, light absorption and intensity and the effect of the bio-optic state. We will use photons maps to accurately model these effects.

*aranha@cs.bris.ac.uk

[1] Remote Operated Vehicle

$$\begin{aligned}\frac{\partial L(x,\overrightarrow{\omega})}{\partial x} = \quad & \alpha(x)L_e(x,\overrightarrow{\omega}) + \sigma(x)L_i(x,\overrightarrow{\omega}) \\ & -\alpha(x)L(x,\overrightarrow{\omega}) - \sigma(x)L(x,\overrightarrow{\omega})\end{aligned} \tag{1}$$

# 3 Previous Work

One of the major uses of Computer Graphics is to simulate environments. The use of interactive simulation provides an imitation of a state of affairs or real device. It is often valuable in environments where it may be prohibitively expensive or dangerous to venture without previous training. Simulators have been used to replicate aircrafts, space flight, trains and automobiles. In order to create a realistic simulator, it is important that the graphics are as accurate as possible.

One of the most recent significant demonstrations of underwater effects was in *Finding Nemo*. To create these effects, Pixar's *Renderman* was adapted to blur objects based on ocean depth and distance from the viewpoint. Water interaction was created by modifying *Fitz*, Pixar's fur and cloth simulator[1]. Obviously these methods are not physically real.

Existing advanced ray-tracing solutions, including *Radiance*, can accurately render scenes, recreating effects such as reflections, refraction and absorption. They also are able to produce physically accurate specular reflections and shadows. Our solution adapts these methods to produce authentic renderings. In addition to a physically based rendering solution, accurate participating media, surface and lighting simulation need to be incorporated.

Research into realistic participating media is a complicated problem which requires the solution of the *radiative transport equation*. A solution using a volumetric photon map demonstrated the ability to model silty water[12]. The Lucifer system used a curved photon mapping solution to model effects caused by inhomogeneous media[10]. It is important that our volumetric media is related to data on water content to validate its accuracy[20].

Generating accurate renderings of water surfaces has been a significant area of research[4][24], with many proposed solutions each with their own pros and cons. In general solutions are split according to the depth of the sea environments. Although there are built-in methods of generating a sea surface within programs such as Maya, these are based on generating shaders and displacement maps. They are not physically based and do not enable us to alter the effects under the surface[2]. For the purpose of this project, we need a solution based on the physical theory of waves. There are several suitable solutions including one of the original models, Stokes Waves as well as Gerstner Waves[15][14]. Both, however, have weak points and are not directly related to any quantifiable evidence. A suitable system was recently described in a paper by Premoze and Ashikhmin[23]. This system is based on oceanographic research and controlled by observable parameters such as wind speed and direction.

# 4 Implementation

As stated previously, the goal of the project was to create physically accurate results, the definition of this is somewhat vague. The ideal solution would be to be capable of recreating a scene as if the environment actually existed, being visually and measurably indistinguishable from real-world images[8]. As this work concerns a proof-of-concept, we will take physically accurate to mean perceptually indistinguishable, and not necessarily measurably precise due to the modelling inconsistencies to the real-world scenes. The system is to be capable of reproducing effects as described in Section 2.

Our research is based on existing modelling and rendering methods. These methods are however largely concerned with rendering of scenes above water, which is significantly different to underwater. One of our main considerations is thus the need for an accurate method of simulating the volumetric nature of water, taking into account light scattering/absorption. *Radiance* alone is not adequate for rendering such scenes due to its reverse ray-tracing implementation. Photon mapping can be used to overcome these shortfalls and accurately model effects such as refraction of light causing caustics [11] [13]. The *Fraunhofer ISE* photon map which has been implemented to work with *Radiance* can be used for this purpose [3].

## 4.1 Volumetric Nature

Implementation of a volumetric shader has to be capable of replicating the scattering of light within the ocean. This scattering varies depending upon the type of biological material in the ocean. Different types and concentration alter the perception of the environment. The system also needs to be capable of recreating murk, that is, the filtering of light with depth. To model this, an implementation of multiple scattering and absorption in anisotropic media with non-uniform density distributions would be the most accurate, however, it would also be extremely time consuming to process. We thus face a trade off between the accuracy and complexity of the volumetric shader and the time it will take to produce meaningful images.

The compromise we chose was to implement our system using *Radiance's* single-scatter approximation. This uses a Heyley-Greenstein eccentricity model for forward-scattering (2) [16] which is most suited to thin and low-albedo scatter media. This material is controlled by properties for the in-scatter (light scattered into view direction), out-scatter (light scattered away from view direction), and absorption. The rate of absorption is modelled using the *absorption coeffi cient* which is a function of wavelength(3). The *scatter function* is based on the wavelength of light and the integrated average of angle between the incident and scatter direction(4). Total loss of radiation is a summation of both of these (5).

$$f(\theta) = p_H G(\theta, g) = (1-g^2)/(1+g^2-2g\cos(\theta))^{1.5} \quad (2)$$

$$k_a(\lambda) = \text{absorbed fraction per unit distance} \quad (3)$$

$$k_s(\lambda) = k_s(\lambda) f(\theta) \quad (4)$$

$$k_t(\lambda) = k_a(\lambda) + k_s(\lambda) \quad (5)$$

A crude layering strategy was used to create a feeling of murk within the scenes allowing greater depths to appear darker than that of water closer to the surface. This layering is currently only discrete rather than a more suitable analogue approach. The interface between the water surface and the volumetric material representing the water is also currently somewhat coarse due to the complexity required to develop an accurate blending between the two. Both these limitations will be addressed in the future.

### 4.2 Lighting

A key factor to the realism of the scene is to accurately model the distribution of lighting of the Sun and Sky. In particular we need to consider the intense radiation of the sun and the redistributions within the sky. A Radiance tool, *gensky* is used to generate this, based on IES luminaire data, with parameters to specify the time and location. To help the user analyse suitable equipment for exploration, a tool to experiment with different types of underwater lights was implemented. This tool allows users to specify parameters to accurately create lights based on manufacturer's information of standard lights [21].

### 4.3 Surface Conditions

Due to the difficulty of creating the surface of the water, a balance is needed between accuracy, computational efficiency and complexity. Waves can be assumed to be composed of a linear combination of sinusoidal waves as a result of wind [4]. Experimental statistical measurements of waves have shown they have Gaussian distributions [7]. We have implemented wave generation using Fourier transform, based on the work by Tessendorf [24] and Mastin et al [4].

Implementation of the wave generator is split into the following sections: main module, interface unit, conversion module and GUI. The main module is for key technical implementations including the Fourier transform. To convert into the specific modelling package commands the interface unit is used. The conversion unit converts units and measurements. The Interface Unit and GUI are dependent upon the modelling package, the other two sections can be developed independent of any modelling package. Implementation of the overall system is similarly divided into two sections; scene generation, and the rendering engine.

### 4.4 Scene Generation

Underwater scene generation can be carried out using a variety of packages. For the purpose of this work, we developed a package based around *Maya* since it allows easy development of tools to increase its functionality. In the future our system could include integration with CAD packages so that blueprints for submersed structures can be directly imported into our system.

Once the scene is generated, we use our developed extension to generate waves to act as the sea surface. This extension is developed using MEL[2] script for platform independence. The script creates a surface to represent the sea surface based on a set of specified parameters. A future generation could be carried out using, for example C++, to give a better performance.

### 4.5 Rendering Engine

Although this could be implemented in many languages, the main program was developed in Java for easy prototyping and portability. In a future version this could be implemented in C++. The rendering engine was specifically designed to prevent the user from having to learn how to use the rendering tools, *Radiance* and *Fraunhofer ISE* photon map. Within the rendering engine program, parameters are specified for the original scene file, including view point and parameters to specify the scene properties (participating media, sky light distribution, secondary lighting). The rendering engine then generates new scene file(s), along with another file providing *Radiance* with instructions as to how to render the scene. The new scene files to include the scene parameters are created and the *Gensky* and *Lampcolor* programs are used to generate the Sun/Sky light distribution and lighting properties respectively. To specify a viewpoint from which to render the scene, a script developed by Gillibrand was used to export a camera view point [6]. *Radiance* is then run in the background to render the desired image.

## 5 Results

In order to test the performance of the system, photographs of a real scene were taken, and a model of it created using *Maya*. Generated results were then compared to that of the photograph. Comparisons were also made between the aesthetic *Maya* renderer and our physically based system. The performance of the system to accurately capture factors such as bio-optic state, depth, light intensity and position, and secondary lighting was also evaluated.

[2] Maya Embedded Language


Figure 1: Photograph of scene to be rendered, the Kormoran in the Red Sea

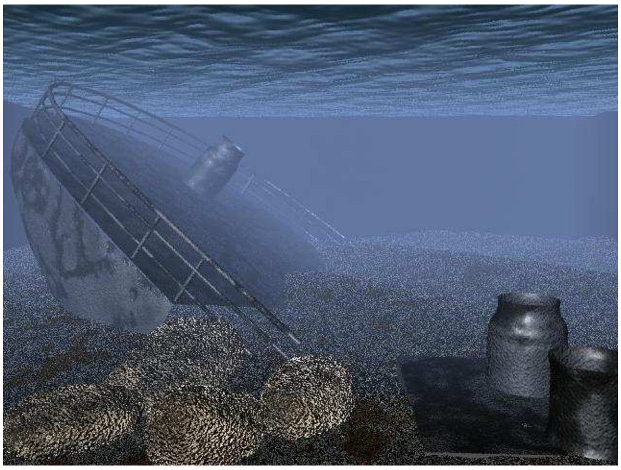

Figure 2: Maya render of scene

Figure 3: Physically based render

The selected scene was of a ship wreck in the Red Sea. The scene includes effects such as caustics, light shafts and the underwater phenomenon, Snell's window. The water is relatively low in marine life, providing a predominantly blue environment. The original scene can be seen in Figure 1. The scene rendered in Maya is shown in Figure2 and the physically based render, Figure3.

A number of techniques have been developed to compare image fidelity[22][5][17] using numerical and perceptual methods. Perceptual methods of analysis are most suited to comparing our images due to the geometric misalignments within the scene. Although neither the Maya or physically based render significantly resemble the original photograph, this is primarily due to the quality of the modelling. It can, however be seen that our rendering system produces effects which have not been produced by the Maya image. The increased accuracy of our system to reproduce the volumetric nature of the water and light interaction is apparent.

# 6 Conclusions

As the figures show, it is possible to compute good approximations of underwater scenes, but there is still some way to go before we can achieve the high fidelity required by the underwater industry. A number of improvements to our system are being considered.

To reduce the cost of computation, only homogeneous media and non spectral rendering were considered. Accounting for inhomogeneous media and spectral rendering would allow a system to be developed incorporating inelastic scattering, accounting for energy transfers between wavelengths and curved ray paths as a result of Fermat's Principle. The Lucifer system incorporating these was developed by Gutierrez et al[9].

To minimise processing complexity, whitecaps were not included in wave generation. Whitecaps are the foam on waves, caused by a breaking wave dependent upon temperature difference and chemistry of the water. For increased realism these should be included, a suitable approximation has been developed by Monahan [18]. Another feature also currently absent from our system is particulate matter in the water.

An important development required for an interactive system is to implement a model of the surge and swell present in the sea. Improved accuracy of the surface model and volumetric renderer needs to be considered, extending our model to shallow water environments. Non-uniform scattering and particles would also increase accuracy of the system.

# 7 Acknowledgements

Thanks to my supervisor, Alan Chalmers for his great guidance and enthusiasm. To Tomas Plachetka and

Richard Gillibrand for their patience and helping me understand some of the concepts used in this project. I am grateful for the Radiance online discussion group for their advice and John Liddiard for his support, suggestions and photographs.

Figure 4: Demonstration of volumetric rendering from a distance

Figure 5: Example of refraction from sea surface

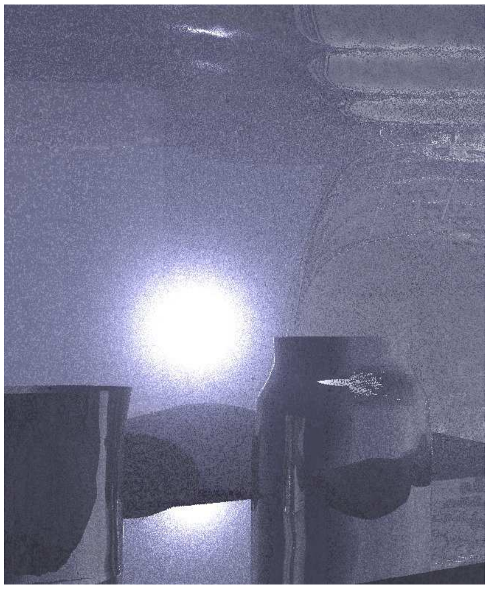

Figure 6: Effect of underwater lighting

## References


[1] B. Desowitz. Depth perception. *Computer Graphics World*, May 2003.

[2] Maya Fluid Effects. http://www.alias.com/eng/products-services/maya/technical_features/maya_fluid_effects/index.shtml, 2004.

[3] Fraunhofer Institute for Solar Energy Systems. http://www.ise.fhg.de/alt-aber-aktiv/radiance/photon-map, 2004.

[4] J.F. Mareda G.A. Mastin, P.A. Watterberg. Fourier synthesis of ocean scenes. *IEEE Computer Graphics and Applications*, pages 16–23, Mar 1987.

[5] A. Gaddipati, R. Machiraju, and R. Yagel. Steering image generation with wavelet based perceptual metric. *Computer Graphics Forum*, 16:82–91, Sep 1997.

[6] R. Gillibrand, A. Chalmers, G. Ellis, and P. Ledda. Maya to radiance. *In Proceedings of Radiance Workshop*, 2003.

[7] C.M. Goral, K.E. Torrance, D.P. Greenberg, and B. Battaile. Modelling the interaction of light between diffuse surfaces. *Computer Graphics*, 18(3):213–222, 1984.

[8] D.P. Greenberg. A framework for realistic image synthesis. *Commun. ACM*, 42(8):44–53, 1999.

[9] D. Gutierrez, F.J. Seron, A. Muoz, and O. Anson. Inelastic scattering in participating media using curved photon mapping. *ACM SIGGRAPH 2004 Full Conference, Los Angeles, CA (EEUU), 8-12 August 2004*, 2004.

[10] D. Gutierrez, F.J. Seron, A. Muoz, and O. Anson. Rendering ghost ships and other phenomena in arctic atmospheres. *The 13th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2005*, 2005.

[11] N.J. Christensen H.W. Jensen. Photon maps in bidirectional monte carlo ray tracing of complex objects. *Computers and Graphics*, 19(2):215–224, Mar 1995.

[12] P.H.Christensen H.W. Jensen. Efficient simulation of light transport in scenes with participating media using photon maps. *In Proceedings of SIGGRAPH'98*, pages 311–320, Jul 1998.

[13] H.W. Jensen. Rendering caustics on non-lambertian surfaces. *In Proceedings of Graphics Interface '96, Toronto*, pages 116–121, 1996.

[14] P.D. Komar. *Beach Processes and Sedimentation*. Prentice-Hall, 2 edition, 1997.

[15] B. LeMehaute. *An Introduction to Hydrodynamics and Water Waves*. Springer-Verlag, 1976.

[16] J.L. Greenstein L.G. Henyey. Diffuse radiation in the galaxy. *Astrophysical Journal*, 93:70–83, 1941.

[17] A. McNamara, A. Chalmers, T. Troscianko, and I. Gilchrist. Comparing real and synthetic scenes using human judgements of lightness. *In Proceedings of the Eurographics Workshop in Brno*, Jun 2000.

[18] E. Monahan and G. MacNiocaill. *Oceanic Whitecaps: Their Role in Air Sea Exchange Processes*. D. Reidel, 1986.

[19] Nasa SeaWiFS Project. http://seawifs.gsfc.nasa.gov/seawifs.html, 2004.

[20] K.S. Baker R.C. Smith. Optical classification of natural waters. *Limnology and Oceanography*, 23(2):260–267, Mar 1978.

[21] M.S. Rea. *Lighting Handbook, Reference and Application*. Illuminating Engineering Society (IES) of North America, 9 edition, 2000.

[22] H. Rushmeier, G. Ward, C. Piatko, P. Sanders, and B. Rust. Comparing real and synthetic images: Some ideas about metrics. *Sixth Eurographics Workshop on Rendering, Dublin, Ireland*, pages 82–91, 1995.

[23] M. Ashikhmin S. Premoze. Rendering natural waters. In *Eighth Pacific Conference on Computer Graphics and Application*, Oct 2000.

[24] J. Tessendorf. Simulating ocean waves. In *SIGGRAPH '99 Course Notes and SIGGRAPH 2000: Course Notes 25: Simulating Nature: From Theory to Practice*, pages 3.1–3.18, 2000.

[25] Alias Website. http://www.alias.com, 2004.

[26] Radiance Website. http://radsite.lbl.gov/radiance, 2004.