Computer Graphics Techniques for Realistic Modeling, Rendering and Animation of Water. Part I: 1980-88

A. Iglesias

Department of Applied Mathematics and Computational Sciences, University of Cantabria, Avda. de los Castros, s/n, E-39005, Santander, Spain
iglesias@unican.es
http://personales.unican.es/iglesias

To Jack Bresenham, with admiration

Abstract. The realistic representation of natural phenomena has a long tradition in computer graphics. Among the natural objects, one of the most interesting (and most difficult to deal with) is water. During the last two decades a number of papers on computer graphics techniques for modeling and rendering of water have been published. However, the computer graphics community still lacks a survey classifying the vast literature on this topic, which is certainly unorganized and dispersed and hence, difficult to follow. This is the first of a series of two papers (both included in this volume) intended to fill this gap. Our aim is to offer a unifying survey of the most relevant computer graphics techniques for realistic modeling, rendering and animation of water. In this paper we focus on the methods developed during the period 1980-88.

1 Introduction

Modeling natural phenomena has always been among the most challenging problems in computer graphics. Because natural objects are frequently quite asymmetric and nonrigid, they have an inherent complexity far beyond that of most artificial objects. Of all the natural objects, one of the most interesting (but most difficult to deal with) is water. In fact, the modeling and rendering of water has been a traditional problem in computer graphics that has been addressed by many authors during the last two decades. However, the computer graphics community still lacks a survey classifying the vast literature on this topic. This is the first in a series of two papers (both included in this volume) whose aim is to offer a unifying survey of the most relevant computer graphics techniques for realistic modeling, rendering and animation of water. The whole survey has a chronological character; the two papers are devoted to the methods developed during the periods 1980-88 and 1989-97, respectively. This (somewhat artificial) classification has some obvious pros and cons. A major advantage is the opportunity to gain a clear perspective on how computer graphics advances have improved the rendering and animation of water over time. On the contrary, a

P.M.A. Sloot et al. (Eds.): ICCS 2002, LNCS 2330, pp. 181–190, 2002.

major shortcoming is that models from different years might be organized into different sections, even though they are based on the same principles and/or methods. Finally, we should remark that both papers have been strongly influenced by the limitations of space. Some references have had to be omitted and many explanations have been reduced to the minimum. However, we hope that we have included enough comments and references to make the papers useful to our readers.

2 Earlier works: 1980-85

The early 80s marked the starting point of research on the computer modeling and rendering of natural phenomena. However, relatively little time was spent on modeling the appearance and behavior of water. At that time, research on this topic was basically concentrated on representing a large mass of water without boundaries, such as the ocean. On the other hand, water was seen as a compact fluid rather than a mixture of individual droplets.

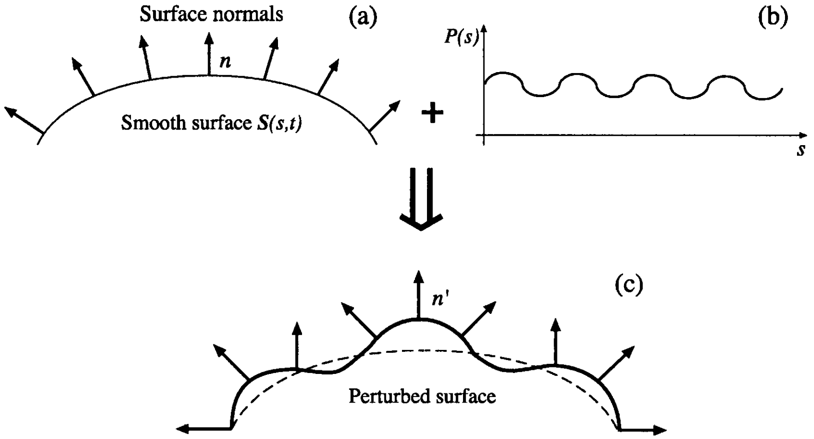

Fig. 1. 2D-scheme of the bump mapping technique: a smooth surface (a) is perturbed by adding a *bump mapping* function $P(s)$ (b) to the surface in the direction of the normal vector. This process allows a realistic rough-textured surface to be obtained (c)

The first attempts to render waves in water were based on the *bump mapping* technique developed by J. Blinn in 1978 [4]. This method allows realistic rough-textured surfaces to be obtained by perturbing the surface normal. Blinn realized that true rough-textured surfaces have a small random component in the surface normal and hence in the light reflection direction, which can be well reproduced by adding a perturbation function to the surface in the direction of the normal vector, as shown in Fig. 1. This solution is better than adding texture patterns to smooth surfaces because, in this last case, the resulting surfaces also appear smooth.

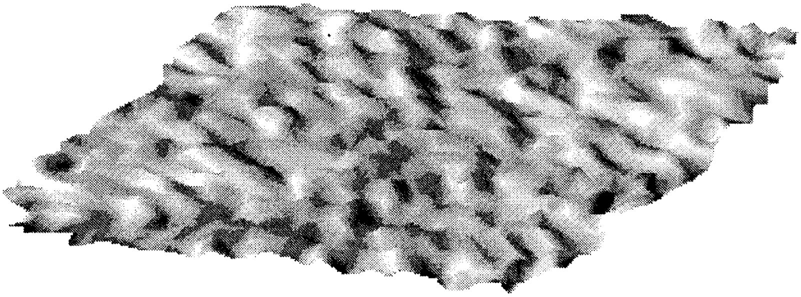

Fig. 2. Random waves perturbation function to simulate a choppy ocean through the bump mapping tecnique

Of course, different perturbation functions can be used to produce different effects. For instance, Schachter [32] proposed a model for fields of random waves (similar to those shown in Fig. 2) that involved a table look-up of precomputed narrow band noise waveforms (a technique intended for real-time applications, and which was implemented in hardware). Other early references were two sets of Siggraph slides not described elsewhere. The first one is the "Pyramid" slide by Gary Demos *et al.* in 1981 [27], where the waves in the sunset are obtained by bump mapping the flat surface with cycloidal waveforms. The second one is the "Night Castles" slide, by Ned Green, in the 1982 collection which used sine waves for the bump mapping technique.

These beautiful images lack some important realistic effects, such as the reflection of objects in the water. To overcome these limitations, Witthed's film "The Compleat Angler" combined bump mapping with *ray tracing*, a well-established technique to render transparencies taking into account reflection and refraction of light (see [14] for a nice introduction to the field). In that film, ray tracing[1] was applied to animate realistic reflections from ripples in a small pool, where the ripples were created by applying a single sinusoidal function to perturb the surface normal [37].

Although based on ray tracing too, Max used a different approach to render the wave surfaces for his famous film "Carla's Island" [20]. His scheme [19] was a ray tracing model in which ocean waves and islands were rendered by different but related algorithms. His work was based on the Fishman and Schachter algorithm [8] for rendering the raster images of the *height field*, i.e., single valued functions of two variables. Max modified the algorithm (one vertical scan line at a time) to reproduce the detail of small ripples near the eye without wasting time near the horizon [19]. The same height field was used for the islands, which were represented as elliptical paraboloids with superimposed cosine terms to give the rolling hills. The wave model was represented by an approximate solution (valid

[1] It should be remarked here that as early as in the 70s there were some references to procedural models for ray tracing, such as [17, 23]. See also [31].

only for waves of small amplitude) in which the wave velocity v is proportional to the square root of the wavelength λ; that is, $v \propto \sqrt{\lambda}$. He also assumed a first linear approximation for the wave surface, meaning that the wave trains pass through each other without modification (clearly, a non-realistic situation). Under this assumption, the model for the wave surface was given by a sum of cosine functions corresponding to individual trains of waves of low amplitude. In the model, additional terms from Stokes' approximation [34] of the wave equation might be added to large-amplitude waves. Unfortunately, the author considered the second-order term should be added to the largest wave only, so these (basically) linear waves exhibited a notable deficiency: they formed a self-replicant pattern when viewed over any reasonably large area. On the other hand, his linear small-amplitude theory was restricted to deep water, but it is in shallow water where waves break and this theory fails to predict this phenomenon. Finally, as pointed out by the author, this scheme did not address the problem of rendering ocean scenes with clouds, later analyzed in [11,12,21].

To solve these problems, Perlin applied bump mapping through a set of 20 cycloidal waveforms, each radiating in a circular fashion from a randomly placed center point [26]. Then, the appearance of the ocean surface was improved by combining bump mapping with a rich texture map based on nonlinear functions, the so-called *solid texture.* This technique, which is independent of the surface geometry or the coordinate system, defines a texture throughout a three-dimensional volume, where the object to be textured is embedded. The textured surface is obtained as the intersection of the object and the three-dimensional texture volume [7]. The solid texture was applied by Perlin to generate realistic images of clouds, which could be hereby incorporated into the ocean scenes. Another approach used clamped analytical functions to antialias textured surfaces [24]. These schemes work well when the viewer is some distance away from the water so that the ocean surface appears flat.

Although these initial developments were able to produce very beautiful images, they still needed further research to be suitable for realistic rendering and animation. This is due to the techniques the previous methods were based on, mostly bump mapping (such as [15,26,32,37]) and ray tracing (such as [19,20, 37]). Bump mapping is inexpensive, but it is not sufficient to simulate and animate waves in general. Since the actual surface is flat, bump mapped waves do not exhibit realistic silhouette edges or intersections with other surfaces. In addition, these waves cannot shadow one another or cast shadows on other surfaces. Thus, these methods could not produce realistic scenes containing a seashore; they are forced to restrict themselves to simple images of the ocean surface as one might see it from an aircraft well out the sea. However, when the viewer is near the surface, the three-dimensional nature of the surface becomes more important. On the other hand, ray tracing also exhibits some remarkable disadvantages, such as its tendency to aliase arising from point sampling distant waves near the horizon, the difficult of rendering complex procedural models and the most important one, its immense requirements of calculation time, making animation unapproachable. In fact, although many improved methods have been proposed since then [1], ray tracing is still not adequate for computer animation.

3 Mid Period: 1986-88

During this period the realism of water animations was improved in several directions. On one hand, considerable effort was devoted to simulating the interaction between the fluid and a solid. For example, in [10, 25] the authors proposed a combination of particle systems and hydrodynamics. The *particle systems* were introduced by Reeves [29] (see also [30]) to describe natural phenomena which cannot be well described by means of polygons or curved surfaces (trees, grass, wind, etc.). Their ability to model complex phenomena is given by the fact that these particles do not have smooth, well-defined surfaces but rather irregular, complex surfaces with variable shapes. In addition, the characteristics and shape of the particles change with time under the action of physical or stochastic models. For example, in [30] stochastic wind functions and wind maps were employed for shading and rendering particle systems representing both gusts of wind and the motion of wind-blown blades of grass. Finally, the particles are generated, move within the system and die or leave it over time.

These particle systems were generally applied to model the foam and the spray generated by wave breaking and collisions with obstacles. In [25] a particle system for each breaking wave was generated, where the initial position of each particle is at the crest of the wave and its initial velocity is in the same direction as the wave motion. Then, to avoid excessively uniform particle behavior, stochastic perturbations [9] with a Gaussian distribution were added to the velocity of the particles. Finally, another particle system model was generated to simulate the spray from the collision of waves with partially sumerged obstacles (rocks, piers, etc.). A similar approach can be found in [10]. There, the authors derived a rule for the generation of spray and foam: spray is generated when the difference between the particle speed and the surface speed projected in the direction of the normal to the surface exceeds a threshold; otherwise, foam is generated. When generated, spray is sent in the direction of the normal, whereas the foam is sent sliding along the wave surface.

On the other hand, the theory of hydrodynamics was applied to model the ocean waves and many of their associated effects. It should be remarked, however, that there is no hydrodynamic model able to fully and realistically describe the behavior of any real ocean waves. In addition, a model that might be completely useless to a physicist or an oceanographer might be very good for computer graphics and vice versa. Therefore, these approaches must be understood as simple (although often dramatic) approximations carried out for rendering and animation purposes only, with no physical meaning in general.

The general model of wave motion is highly nonlinear, and has no convenient solutions, so many simplified and idealized models have been usually applied instead. For example, Peachey [25] considered the Airy (linear) model of sinusoidal waves of small amplitudes. In his model, the velocity and the wavelength of a wave depend on the depth of the water. On the other hand, the ocean surface was modeled with a height field, thus preventing waves whose crests curl forward. From this point of view, Peachey's model seems to be closely related to Max's model (see Sect. 2). However, Peachey also dealt with wave refraction due to wave velocity changing with depth. *Wave refraction* implies that when the waves approach the shore from deep water, their crests tend to become parallel to the shoreline regardless of their initial orientation. This effect has important consequences in water rendering and animation: the models ignoring wave refraction may produce implausible situations, such as the crests running perpendicular to the beach in [19]. As remarked by several authors, in Max's film "Carla's Island" the ocean waves appear to cut right through the islands as if the islands were made of air.

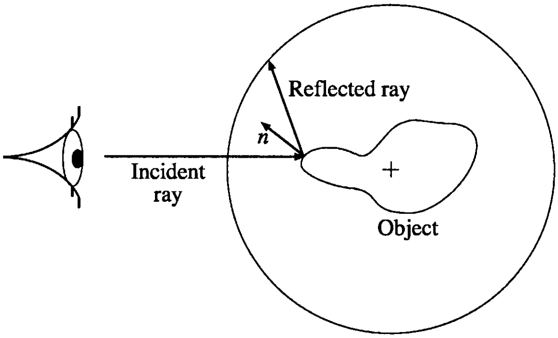

Fig. 3. Scheme of the environment mapping technique

Wave refraction was also considered by Fournier and Reeves [10], whose scheme was based on the Gerstner-Rankine model, proposed long ago in oceanography [13, 28]. Roughly speaking, this model establishes that particles of water describe circular or elliptical stationary orbits. Realistic wave shapes and other needed effects, such as those due to the depth (like wave refraction and surf) and the wind can be easily reproduced by simply varying some parameters of the orbit equations. In [10] the ocean surface was modeled as a parametric surface which obviously allows traditional rendering techniques, including ray tracing [16, 35] and adaptive subdivision [6]. The color and shading of the ocean surface were simulated through an *environment map*, a technique introduced by Blinn and Newell in 1976 [5]. Their basic assumption is that the scene environment is composed of objects and lights distant from the object to be rendered (the wave's surface, in this case). Then, each image of the surrounding environment is mapped onto the inside of a gigantic sphere with the rendered object at its center. The color of a point on the wave is determined by tracing a ray from the camera to the object, reflecting it over the surface normal and tracing the reflected ray outward to the sphere, thus giving us the index of the image and hence the color to be reflected (see Fig. 3).

To control the shape of the ocean the authors introduced several *trains of waves*, i.e. groups of waves sharing the same basic characteristics (heights, periods and wavelengths) and the same phase origin. Of course, each train of waves has its own set of parameters and optional stochastic elements. They allow the "variability" and "randomness" characteristics of the sea to be introduced via a combination of small variations within a train and large variations between trains. Since these variations were basically applied to the height and the wavelength, which can be made a function of time, animation could be easily incorporated in this way. Finally, bump mapping was applied to obtain some additional effects, such as water shimmering and water chop. For animation purposes, the bump map was translated over time just like trains of waves were translated over time.

Almost at the same time, other methods for modeling water surfaces based on Fourier synthesis [22], generalized stochastic subdivision [18], etc., were suggested. These methods worked well for producing still images, but were unsuitable for animation. The reason is that they did not include realistic models for the evolution of the surface over time, so actual phenomena such as wave refraction and depth effects were totally passed over. For example, in [22] the authors used an empirical wind-driven sea spectrum model to filter white-noise images. The ocean images were generated from these white-noise images by using the forward FFT (Fast Fourier Transform) and the inverse FFT, thus representing a fully developed sea in nature. The images were then rendered with a ray tracing algorithm to produce realistic ocean scenes. This work focused on synthesizing fully developed seas in deep water rather than seas in shallow water (where effects such as wave refraction and wave breaking exist). The result has a good appearance although it is valid only for deep water waves and difficult to animate correctly.

Another earlier[2] remarkable reference dealing with wave refraction is [36]. In this approach, Snell's law was applied to control the changes in direction of the ocean waves due to wave refraction. The wave orthogonals (the normals to the wave front) were wave-traced similarly to ray tracing. A ray representing a wave's front direction of propagation was followed and tested for the nearest intersection with the contour lines of the ocean floor. The angle of refraction was obtained from Snell's law and the resulting ray was tested for intersection, and so on. Then this process was simplified taking into account that wave refraction is independent of the viewer's viewpoint (hence, it must be computed only once) and it is essentially a two-dimensional effect. To improve realism, the ocean surface was treated as a true three-dimensional surface. It was represented by Beta-splines [2], the tension shape parameter $\beta 2$ (see [3] for a description) being used to add more complexity to the surface: higher values correspond to sharper edges within the surface. This parameter was also changed locally to create smooth troughs and sharper crests. Instead of using ray tracing, two texture

[2] Although finally published in 1987, most of the work in this paper was essentially completed before the appearance of the papers [10, 22, 25] (see Guest's Editor Introduction, ACM Transactions on Graphics **6**(3) (1987) 165-166).

mapping schemes (both including viewer orientation, surface normal orientation and some laws of optics) were employed to render the ocean waves. Both texture maps were created by hand and contained the reflected and the refracted color information, respectively. These schemes were not as physically accurate as ray tracing but were computationally quicker.

4 Conclusions

Table 1 summarizes the main conclusions of this paper. The left column includes

Table 1. Main water models (left column) and computer graphics techniques (right column) developed during the 80s for water modeling, rendering and animation

Water Models	*Computer Graphics Techniques*
EARLY WORKS: 1980-85 (*Simple models, Good still images, Oceans without boundaries*)	**Bump mapping, Ray tracing** **Height fields, Solid texture**
1980: Schachter [32]	Bump mapping (with random waves)
1980: Witthed [37]	Bump mapping (with sine waves) Ray tracing
1981: Pyramid [27]	Bump mapping (with cycloidal waves)
1981: Max [19],[20]	Height fields Ray tracing
1982: Night Castles	Bump mapping (with sine waves)
1985: Perlin [26]	Bump mapping (with cycloidal waves) *Solid texture*
MID PERIOD: 1985-88 (*Hydrodynamics, Wave refraction, Solid-fluid interaction, Animation*)	**Particle systems, Wave tracing** **Environment map, Fourier filter**
1986: Peachey [25]	Particle systems Height fields Ray tracing, A-buffer
1986: Fournier & Reeves [10]	Particle systems Environment maps, Bump mapping Ray tracing, Adaptive subdivision, etc.
1987: Mastin *et al.* [22]	*Fourier-domain filter* Ray tracing *Wind-driven spectrum model*
1987: T'so & Barsky [36]	Beta-splines *Fresnel texture mapping* *Wave-tracing*

the references of the most relevant computer graphics techniques for modeling and rendering of water developed during the 80s. They have been organized for clarity into two different periods, 1980-85 and 1986-88, each associated to a set of techniques listed in bold in the right-hand column. Italics in this column are used to indicate that the corresponding technique was originally described in the reference on the left. We should finally remark that, in spite of these significant advances, there were still a number of unsolved problems. Computer graphics techniques to solve them will be treated in the second part of this survey.

Acknowledgements

The author would like to acknowledge the CICYT of the Spanish Ministry of Education (project TAP98-0640) for partial support of this work. Many thanks are also given to the three referees for their careful reading of the initial version of the manuscript and their helpful suggestions which led to a substantial improvement of the paper.

References

1. Arvo, J., Kirk, D.: A survey of ray tracing acceleration techniques. In: A. Glassner (ed.): An Introduction to Ray Tracing, Academic Press, London, San Diego (1989)
2. Barsky, B.A.: Computer Graphics and Geometric Modelling Using Beta-Splines. Springer-Verlag, Heidelberg (1987)
3. Barsky, B.A., Beatty, J.C.: Local control of bias and tension in Beta-splines. Proc. of SIGGRAPH'83. Computer Graphics **17**(4) (1983) 193-218. Also in: ACM Transactions on Graphics **2**(2) (1983) 109-134
4. Blinn, J.F.: Simulation of wrinkled surfaces. Proc. of SIGGRAPH'78. Computer Graphics **12**(3) (1978) 286-292
5. Blinn, J.F., Newell, M.E.: Texture and reflection in computer generated images. Communications of the ACM **19** (1976) 542-547
6. Dippé, M., Swensen, J.: An adaptive subdivision algorithm and parallel architecture for realistic image synthesis. Proc. of SIGGRAPH'84. Computer Graphics **18**(3) (1984) 149-158
7. Eber, D.S., Musgrave, F.K., Peachey, D., Perlin, K., Worley, S.: Texturing and Modeling. A Procedural Approach, Academic Press, Boston (1994)
8. Fishman, B., Schachter, B.: Computer display of height fields. Computers and Graphics **5** (1980) 53-60
9. Fournier, A., Fussell, D., Carpenter, L.: Computer rendering of stochastic models. Communications of the ACM **25**(6) (1982) 371-384
10. Fournier, A., Reeves, W.T.: A simple model of ocean waves. Proc. of SIGGRAPH'86. Computer Graphics **20**(4) (1986) 75-84
11. Gardner, G.Y.: Simulation of natural scenes using textured quadric surfaces. Proc. of SIGGRAPH'84. Computer Graphics **18**(3) (1984) 11-20
12. Gardner, G.Y.: Visual simulation of clouds. Proc. of SIGGRAPH'85. Computer Graphics **19**(3) (1985) 297-303
13. Gerstner, F.J.: Theorie der wellen. Ann. der Physik **32** (1809) 412-440
14. Glassner, A. (ed.): An Introduction to Ray Tracing, Academic Press, London, San Diego (1989)

15. Haruyama, S., Barsky, B.A.: Using stochastic modeling for texture generation. IEEE Computer Graphics and Applications **4**(3) (1984) 7-19. Errata: IEEE Computer Graphics and Applications **5**(2) (1985) 87
16. Kajiya, J.T.: New techniques for ray-tracing procedurally defined objects. ACM Transactions on Graphics **2**(3) (1983) 161-181
17. Kay, D.S.: Transparency, Refraction and Ray Tracing for Computer Synthesized Images. Master's Thesis, Cornell University, Ithaca, New York (1979)
18. Lewis, J.: Generalized stochastic subdivision. ACM Transactions on Graphics **6**(3) (1987) 167-190
19. Max, N.L.: Vectorized procedural models for natural terrain: waves and islands in the sunset. Proc. of SIGGRAPH'81. Computer Graphics, **15**(3) (1981) 317-324
20. Max, N.L.: Carla's island. Issue 5 of the Siggraph Video Review (1981)
21. Max, N.L.: The simulation of natural phenomena. Proc. of SIGGRAPH'83 (panel). Computer Graphics, **17**(3) (1983) 137-139
22. Mastin, G.A., Watterberg, P.A., Mareda, J.F.: Fourier synthesis of ocean scenes. IEEE Computer Graphics and Applications **7**(3) (1987) 16-23
23. Newell, M.E.: The utilization of procedure models in Digital Image Synthesis. Ph. D. Thesis, University of Utah, Salt Lake City, Utah (1975)
24. Norton, A., Rockwood, A.P., Skolmoski, P.T.: Clamping: a method for antialiasing textured surfaces by bandwidth limiting in object space. Proc. of SIGGRAPH'82. Computer Graphics **16**(3) (1982) 1-8
25. Peachey, D.R.: Modeling waves and surf. Proc. of SIGGRAPH'86. Computer Graphics **20**(4) (1986) 65-74
26. Perlin, K.: An image synthesizer. Proc. of SIGGRAPH'85. Computer Graphics **19**(3) (1985) 287-296
27. Pyramid Catalogue: Pyramid, Box 1048, Santa Monica (1981)
28. Rankine, W.J.W.: On the exact form of waves near the surfaces of deep water. Phil. Tran. Roy. Soc. A **153**(4), (1863) 127-138
29. Reeves, W.T.: Particle systems - a technique for modeling a class of fuzzy objects. Proc. of SIGGRAPH'83. Computer Graphics **17**(3), (1983) 359-376; also in ACM Transactions on Graphics **2**(2), (1983) 91-108
30. Reeves, W.T., Blau, R.: Approximate and probabilistic algorithms for shading and rendering structured particle systems. Proc. of SIGGRAPH'85. Computer Graphics **19**(3), (1985) 313-322
31. Rubin, S., Whitted, T.: A 3-dimensional representation for fast rendering of complex scenes. Proc. of SIGGRAPH'80. Computer Graphics **14**(3) (1980) 110-116
32. Schachter, B.: Long crested wave models. Computer Graphics and Image Processing **12** (1980) 187- 201
33. Sims, K.: Particle dreams (Video). Segment 42, Siggraph Video Review 38/39 (1988)
34. Stokes, G.G.: Mathematical and Physical Papers, Vol. 1. Cambridge University Press, Cambridge (1880)
35. Toth, D.L.: On ray tracing parametric surfaces. Proc. of SIGGRAPH'85, Computer Graphics **19**(3) (1985) 171-180
36. Ts'o, P., Barsky, B.A.: Modeling and rendering waves: wave-tracing using beta-spline and reflective and refractive texture mapping. ACM Transactions on Graphics **6**(3) (1987) 191-214
37. Witthed, T.: An improved illumination model for shaded display. Communications of the ACM **23**(6) (1980) 343-349