

On Modeling and Rendering Ocean Scenes

Jean-Christophe Gonzato and Bertrand Le Saëc

LaBRI**
351 cours de la libération,
33405 Talence
France
`[gonzato|lesaec]@labri.u-bordeaux.fr`

This paper is devoted to ocean images. First, we propose a complete geometrical model accounting for refraction, diffraction, reflection, transmission and multi-waves trains. Then, we describe a specific algorithm for the rendering of coastal scenes using a particular illumination model for the scene and displacement mapping texture techniques to deal with multi-waves trains.

keywords: Ocean, dynamic wave tracing, optical phenomena, displacement mapping, illumination, ray tracing.

1 Introduction

The goal of this paper is to generate photorealistic images of coastal scene intended for animations. This implies, first of all, the construction of a specific geometrical model providing wireframe images of the sea. Moreover, this model must offer continuous sequences of such frames. Secondly, an inexpensive rendering model has to be proposed to generate the corresponding images.

First, we introduce a first geometric model detailed in [8]. The shape of waves is computed by parametric functions derived from the Gertsner and Biesel models. Plunging breaker near the shore are modelised. In this first model, we generate wave crest line by using a ray-tracing based algorithm called Dynamic Wave Tracing. To generate the scene, we cast wave rays which are orthogonal to the wave crests in the scene. With this method, all optical phenomena like refraction, diffraction and reflection can be computed.

In this article, we present rapidly the shape of wave computation. Then, refraction, diffraction and reflection are introduced. The refraction is computed by the well-known Snell-Descartes law. For the diffraction, an adaptation of the method proposed by Larras [13] to simulate wave propagation in ports is used.

In order to increase the realism of the images, multi-waves trains and capillary waves are introduced. A tessellation of each wave trains are computed. Our method treats these waves with an algorithm based on displacement mapping [2][14] to capture the real surface of the oceans.

Finally, a simple rendering algorithm for these scenes is done. This algorithm is based on the Incremental Ray Tracing algorithm. Our illumination model is an adaptation of the one proposed by Whitted [19]. The differences between the original algorithm and ours lies in the fact that the sea is treated as a particular case. Fresnel laws are applied to compute the contribution of reflection and refraction parts of light and the color of the sea is computed by its full spectral absorption coefficients.

2 Shape of Waves

A first theory was proposed by the physicist Gerstner in 1804. It is rigorous for an infinite depth. Each particle of water revolves around a fixed point $M(x_0, z_0)$ describing a circle of radius R . This circle is included in a disc whose radius is $\frac{1}{K}$. K is the wave number, ω is the angular speed.

² Laboratoire Bordelais de Recherche en Informatique (*Université Bordeaux I and Centre National de la Recherche Scientifique*). The present work is also supported by the *Conseil Régional d'Aquitaine*.

The (x, z) coordinates of each particle are :

$$\begin{cases} x = x_0 - R \sin(Kx_0 - \omega t) \\ z = z_0 + R \cos(Kx_0 - \omega t) \end{cases} \quad (1)$$

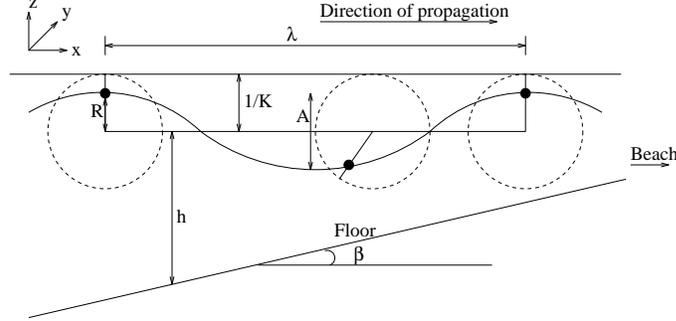


Fig. 1. General definition of wave

Gerstner [12] theory is usable with a great depth, but the influence of the bottom on waves is essential near the shore. When waves arrive on the beach, only the period seems to be invariant [12], the wavelength decreases. All variables marked with an ∞ are used at an infinite depth. In the sequel, h is the depth at the current point x .

$$\frac{K_\infty}{K} = \frac{\lambda}{\lambda_\infty} = \tanh(Kh) \quad (2)$$

Biesel [3] improves the previous model showing that the circles have to be progressively replaced by ellipses with the major axis oriented along the ground slope when the depth is taken in account. This change entails the formation of breaking waves near the beach.

In the Gerstner model, the circles described by the particles of water are restricted to the disc (equivalent to $R \leq \frac{1}{K}$). When $R > \frac{1}{K}$, loops in the wave shape may occur, which does not occur in nature (fig 2.left). An other point is that the major axis of the ellipses, in Biesel theory, becomes greater than the main disc radius. This increase simulates the beginning of plunging waves at medium depth but, near the shore, the shape wave goes below the ground and the model is inoperative. Therefore, we propose to limit the major axis of the ellipses to the disc. Since the main disc tends to 0 when h goes to 0, this modification guaranties a predictable arrival of waves on the beach. The limit shape of breakers we obtain is presented in fig 2.right.

In order to produce plunging breakers in our model, we propose to add two functions to the previous basic parametric equations. The first one consists in reproducing Biesel waves by stretching progressively the waves on the crest while making no modification in the trough (fig. 3.Left). The second function is a combinaison of an *Orientation* and *Displacement* functions in order to simulate

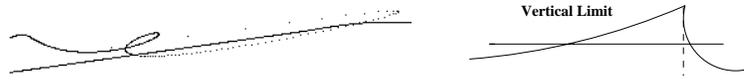


Fig. 2. Left: A problem with Fournier model ; Right: A first limit

the gravity. On the crest, particles of water fall down (fig. 3.Right). These functions are linked to the water depth in order to simulate plunging breakers only near the shore.

For further details of this 2D model, we refer the reader to our previous publication [8].

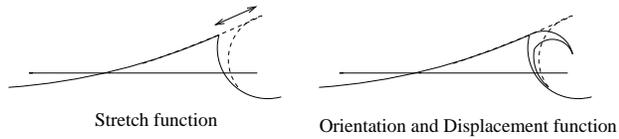


Fig. 3. Towards a plunging breaker

Finally, we present here an example of our model to produce waves and plunging breakers (fig. 4).

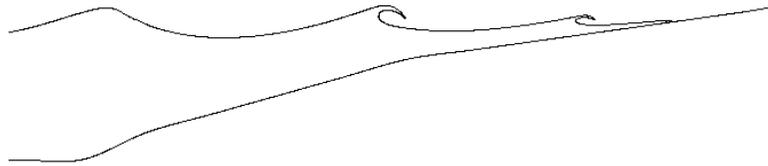
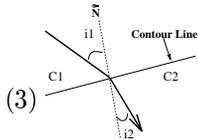


Fig. 4. A example of waves

3 Refraction of Waves

In order to have a complete description of the waves near the shore, we have to position the wave crest lines. The shape of a crest line depends on the form of ocean floor it passes over. Refraction of waves is the principal phenomenon of the depth effect [12][13] . Waves deflect, like light rays through different media. They tend to align themselves slowly to the contour line and, by extension, to the beach. We can apply the refraction given by Descartes law to wave trains by using wave orthogonals and the velocity C_1 and C_2 of waves before and after the contour line.

$$\frac{\sin(i_1)}{C_1} = \frac{\sin(i_2)}{C_2} \quad (3)$$


In [8], an algorithm called Dynamic Wave Tracing (DWT) providing a geometrical definition of a coastal scene with refraction is described. The aim of this algorithm is to keep the same accuracy all over the scene. DWT traces orthogonal wave rays from the open sea along the direction of wave propagation. Step by step, the refraction deflection is calculated. In order to preserve the precision, we send a new ray between two consecutive rays when they diverge too much. The shape of waves is computed along these rays by an improved version of the Fournier-Reeves wave [6]. This algorithm allows to generate many types of scene like baies and islands.

In order to simplify the presentation, the computation of the wave amplitude taking in account refraction will be done only in §4.2.

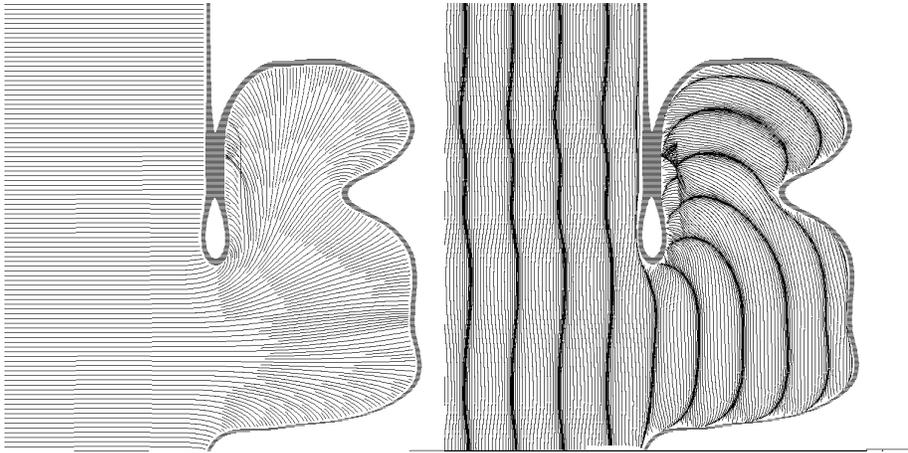


Fig. 5. Wave refraction sample

4 Diffraction of waves

Light diffraction is the set of phenomena that occur when the light has not a linear propagation, in a homogenous environment. Wave diffraction is similar to the light diffraction. This phenomenon is really visible behind jetties and breakwaters.

4.1 Theory of diffraction

A first solution to compute light and wave diffraction was described by Penney and Price in 1944 for a semi-infinite breakwater [3]. Afterwards, several other

solutions have been proposed by physicists, but they all involve expensive computation. An empirical solution was proposed by Larras [13] :

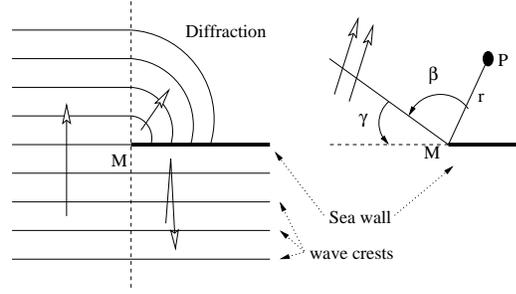


Fig. 6. Wave diffraction theory

This method allows us to quickly find the amplitude of a wave at any point $P(x, y)$ situated behind a breakwater. The cartesian coordinates $P(x, y)$ are transformed to polar coordinates $P(r, \beta)$ computing from the pierhead M . Larras calculates that the relative elevation at point P as the combination of a diffracted incident wave H_i and a reflected wave H_r . These two waves depend on two geometrical parameters U_i and U_r . U_i and U_r are lower than zero when P is behind the jettie ($\beta > \frac{H}{2}$). On this part, the waves are damped.

$$\begin{cases} U_i = 2\sqrt{\pi\frac{r}{L}} \sin(\frac{\pi}{4} - \frac{\beta}{2}) \\ U_r = 2\sqrt{\pi\frac{r}{L}} \sin(\frac{\pi}{4} - \frac{\beta}{2} - \gamma) \end{cases} \quad (4)$$

In these formula, L is the wave length of the incident wave, γ is the angle between the wave train and the pier (fig 6).

In [12] H_i (resp. H_r) is a function of U_i (resp U_r) (fig 7.) by :

$$H_i = \left[\left(\int_{U_i}^{\infty} \cos \frac{\pi}{2} r^2 dr \right)^2 + \left(\int_{U_i}^{\infty} \sin \frac{\pi}{2} r^2 dr \right)^2 \right]^{\frac{1}{2}} \quad (5)$$

To simplify, we propose to approach the previous function by the following empirical equation :

$$\begin{cases} \text{When } U_{i/r} \geq 0.75 \\ H_{i/r}(U_{i/r}) = 0.014e^{-0.4U_{i/r}+3} \times \sin(32 \ln(1 + \frac{U_{i/r}}{15})(U_{i/r} - 0.75)) + 1 \\ \text{otherwise} \\ H_{i/r}(U_{i/r}) = 0.03 + 0.47e^{0.96U_{i/r}} \end{cases} \quad (6)$$

$H_{i/r}$ (resp. $U_{i/r}$) being H_i or H_r (resp. U_i or U_r).

The relative amplitude H is computed by :

$$H = \sqrt{H_i^2 + \alpha^2 H_r^2 + 2\alpha H_i H_r \cos \Phi}$$

where

$$\Phi = 4\pi \frac{r}{L} \sin \gamma \cos \beta + \gamma \quad (7)$$

and α is a coefficient of reflection defined for the breakwater varying from 0 to 1.

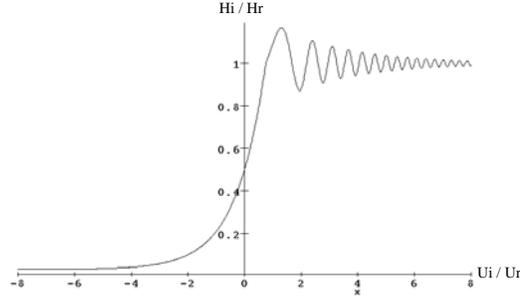


Fig. 7. H_i, H_r as a function of U_i, U_r

4.2 Our method

Diffraction. All rays are traced step by step from the open sea, the DWT algorithm [8] computes the refraction deflection, the local celerity, the wavelength and the amplitude in order to determine the surface of the sea. To include diffraction in our algorithm, the pierhead M of the sea wall is computed. So, when a ray touches a sea wall (point 1 on fig 8.a), we flag its neighbours. Hence, we store the first ray (R1 on fig 8.a) that does not touch the sea wall. We insert new rays flagged DIFFRACTED at the pierhead of the sea wall from R1 direction to the sea wall direction. Step by step, we compute the relative amplitude H of the water on each diffracted ray. The real amplitude at any point P on a diffracted ray is $A_P = HA$ where A is the amplitude at the pierhead. The DWT algorithm continues and creates automatically new diffracted rays.

Refraction in Diffraction. Due to variation of depth, the amplitude of waves is modified by the refraction. This modification is done by the law :

$$A_1 = A_0 \sqrt{\frac{L_0}{L_1}} \quad (8)$$

where L_0, L_1 (Resp. A_0, A_1) are the distances between two neighbour rays (Resp. the amplitudes) before and after a step of progression (fig 8.b). Since we have to

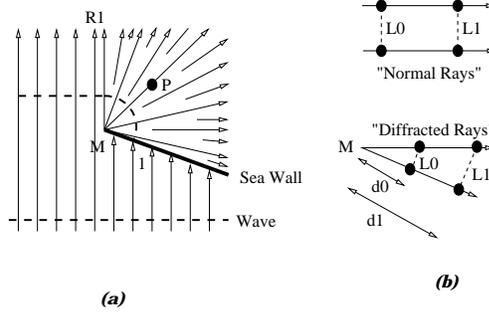


Fig. 8. Our method to compute diffraction, refraction

take in account simultaneously refraction and diffraction phenomena, we use the following equation for the rays behind the pier :

$$A_1 = A_0 \sqrt{\frac{d_1 L_0}{d_0 L_1}} \quad (9)$$

where d_0 and d_1 are the distances from the point to the pierhead before and after a step of the progression (fig. 8.b). If there is no refraction on the diffracted rays, $\frac{L_0}{L_1} = \frac{d_0}{d_1}$ and so the amplitude is constant ($A_1 = A_0$).

4.3 Results

The two next figures present the result of our algorithm on a simple flat ground. A semi-infinite sea-wall is inserted in the scene. Fig 9.a shows how rays evolve in the scene, and Fig 9.b is the corresponding wave train. A discontinuity is noticeable in the wave train near five o'clock. In fact, the angle between the first ray near the wall is lower than between two other diffracted rays. So, the over-sampling does not start at the same time on the scene. This has no effects on the surface computation and continuity.

This scene is 100 meters long and 100 meters large. We send 70 rays and sample along rays each 0.4 meter. New rays are cast when the distance between two consecutive rays is greater than 1.0 meter. This algorithm generates, for this scene, 125 diffracted rays to define the back of the sea wall. It runs on a Silicon Graphics O2 with R5000 processor and takes less than 5 seconds to compute the entire scene (Dynamic wave tracing, diffraction computation and tessellation).

5 Reflection

Wave reflection is similar to the light reflection. When a wave hits a breakwater, it is reflected following the well-known Snell-Descartes law. The reflected wave is quickly damped.

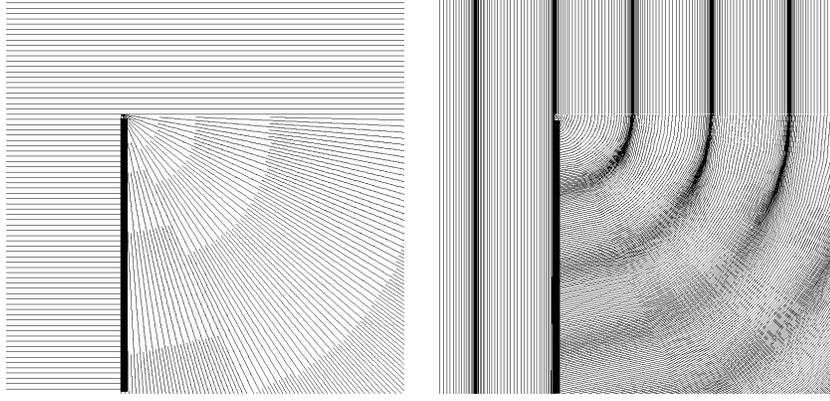


Fig. 9. a. DWT algorithm, b. Waves

5.1 Application in our algorithm

Dynamic Wave Tracing algorithm is based on the classical ray-tracing and so when a wave ray hits a pier, it is automatically recast in its reflected direction. These new rays are called REFLECTED. We notice H_{M1} the amplitude of the wave at the pier. The reflected wave is automatically damped along its propagation on the reflected direction. We compute the relative amplitude by using an exponential function :

$$H_r(d) = e^{-K_r d} \quad (10)$$

Where K_r is a damping coefficient and d is the distance from the point of evaluation to the pier (point $P1$ fig 10). The amplitude at the point $P1$ is :

$$H_{P1} = H_{M1} \times H_r(d) \quad (11)$$

When the amplitude of a ray decreases under a given threshold closed to zero, its progression is stopped. Indeed, when the amplitude of a wave is weak, it has no effect on the other wave trains and on the geometry.

5.2 A particular point : the pierhead

In order to produce a continuous phenomenon between reflected waves and waves going straightforward near the pierhead, we propose to add new rays around the pierhead from the direction of reflection to the direction of propagation without pier. So, we add a new coefficient called H_α which decreases continuously from 1 in the direction of reflection to 0 in the straightforward propagation of the waves. H_α is done by the following empirical formula:

$$H_\alpha(\alpha) = \cos\left(\frac{\pi(\alpha + \beta)}{4\beta}\right) \quad (12)$$

With H_α , we modify locally the amplitude of the wave at the pierhead ($M2$) in function of the new casted ray. So the amplitude at the point $P2$ is :

$$H_{P2} = H_\alpha \times H_{M2} \times H_r(d) \quad (13)$$

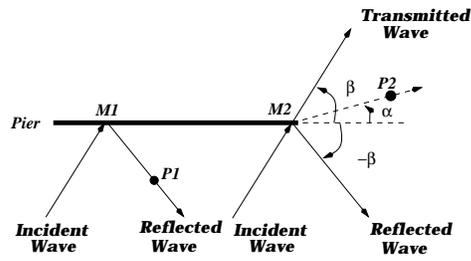


Fig. 10. Pierhead computation

5.3 Results

In this section, we present two results dealing with reflection (fig. 11). The left picture is a scene composed of one wave train reflected and diffracted. The view is produced by a simple Z-buffer algorithm and so the mix between incident and reflective wave is not visible. The second picture presents the tessellation at the pierhead and how we finish smoothly the reflection on the pier. These images has been obtained using Open Inventor.

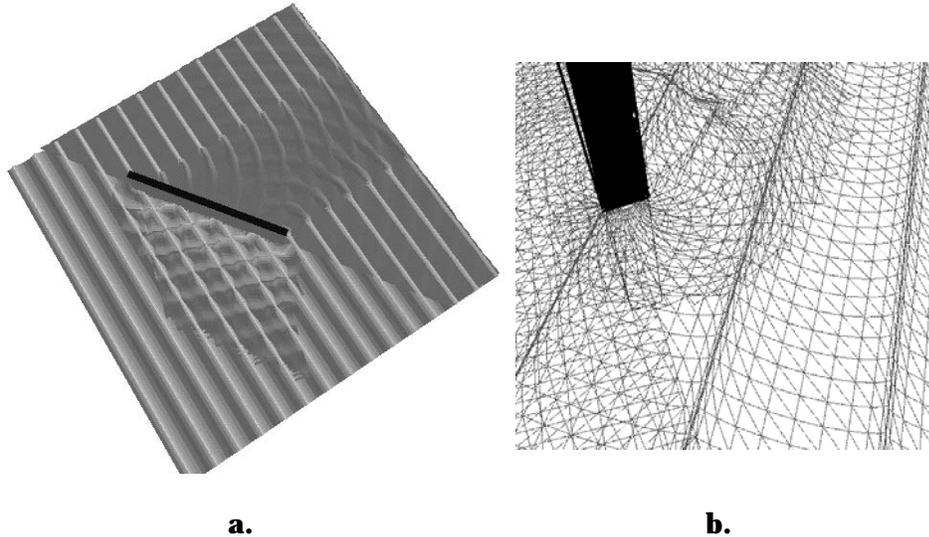


Fig. 11. a. Inventor view of reflection, b. reflection on the pierhead

6 Surface tracking

The surface of the ocean is the combination of different wave trains (direction of propagation, wave length, amplitude, speed, ...).

Peachey [15] and Ts'o-Barsky [17] proposed to use a predefined height field to store the ocean surface composed of the sum of several wave trains. But this solution does not allow plunging breakers. In our more general model, wave trains are treated one by one and the modeler generates all the faces associated with each wave train. So, we have a superposition of surfaces defined by faces. The real surface of the sea is actually computed during the rendering step based on ray-tracing. More precisely, our scene surface is viewed as a flat surface on which we apply displacement maps associated with each wave trains as well as a noise map reproducing capillary waves. The resulting surface is the sum of the different displacement maps.

6.1 Wave trains as displacement mapping

Displacement mapping was introduced by Cook [5]. This technique is used to texture objects when shortcuts of computation as BRDF [16] and bump mapping [4] become noticeable to eye. In displacement mapping, each point of the object is moved along its normal to generate a true deformation of the surface.

A comparison of bump algorithms is made in [2]. The displacement mapping appears to be efficient when the angle between the viewer and the surface is small that is usually the case for coastal scenes.

In our algorithm, wave trains are generated independently and a specific tessellation is computed for each one. So, each point of the real surface of the sea involves several faces : the ocean surface height is computed by addition of the wave trains i.e. faces during the rendering pass. The different tessellations are embedded in a bounding box H_{box} which height depends on the amplitude of the different wave trains. H_{box} can be easily computed during the generation of wave trains.

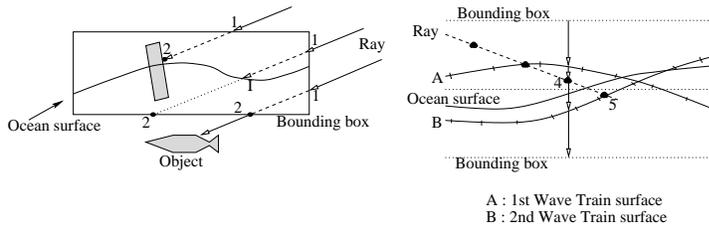


Fig. 12. Ocean surface tracking

To render our scenes, we use the classical incremental ray-tracing algorithm outside “sea bounding boxes”. These objects exist geometrically in the scene, but have no optic influence on the rendering.

When a viewing-ray intersects the bounding box (Points 1. in Fig 12), we send a new ray in order to search the second intersection between this ray and the bounding box or the intersection with an object (different of the perturbed objects) (Points 2. in Fig 12). By this way, we can compute the interval where we have to identify the surface.

The viewing-ray progresses step by step until we are sure that the implicit surface is crossed. After that, the identification of the intersection is computed

by bisection search (Points 4 and 5 in fig 12). At each step, the height of the implicit ocean surface is computed by sending down a vertical ray containing the current step point from the top of bounding box. This ray intersects the different wave trains and so we obtain the height of the real surface by addition. The normal to the surface is computed as the following way :

$$N = (1.0, 0.0, -\sum_{i=1}^n \frac{x_i}{z_i}) \wedge (0.0, 1.0, -\sum_{i=1}^n \frac{y_i}{z_i}) \quad (14)$$

where $N_i(x_i, y_i, z_i)$ is the normal of wave train number i to the point of intersection by the DOWN-ray. To avoid discontinuities on the normal definition of the surface, N_i is computed by the classic Phong interpolation.

6.2 Capillary waves

Capillary waves are modeled using fractal noise. These capillary waves are added to our model and treated as previously. Fractal noise is obtained by a function F :

$$F(n, x, y, t) = \sum_{i=1}^n \frac{\text{noise}(x.k^i, y.k^i, t)}{k^i} + D \times \frac{\text{noise}(x.k^{n+1}, y.k^{n+1}, t)}{k^{n+1}} \quad (15)$$

Where *noise* is a classical function obtained using cardinal spline interpolation [18]. x, y are the coordinates of the point on the sea surface, t is the time.

The fractal noise can be controlled by the distance between the viewer and the surfaces in efficient way. The parameter n of the function manages this precision of the fractal noise. For the part of the sea closed to the viewer, the needed accuracy is maximal and n is equal to six. Far from the viewer, the value of n is zero. The parameter D is used to avoid discontinuities between two fractal levels. This parameter increases progressively from 0 to 1 as a function of distance to the next fractal level.

Plate 1. presents a sample of image of multiple wave trains and capillary waves generated by our algorithm.

7 Rendering

7.1 Illumination model

Our implementation uses a classical ray-tracing illumination model [19] :

$$I = I_a k_d C_\lambda + K_\alpha I_s (k_d C_\lambda (N \cdot V') + k_s (N \cdot H)^p) + K_r I_r + K_t I_t \quad (16)$$

- I_a : Intensity of ambient light
- I_r : Intensity of reflected light
- I_t : Intensity of transmitted light
- k_d : Diffuse part of the object
- k_s : Specular part of the object
($k_d + k_s \leq 1$)

- C_λ : Intrinsic color of the object
- K_α : Absorption coefficient of direct illumination
- K_r : Part of energy reflected
- K_t : Part of energy transmitted
- ($K_r + K_t \leq 1$)

7.2 Reflected and refracted contribution at the sea surface and sea absorption

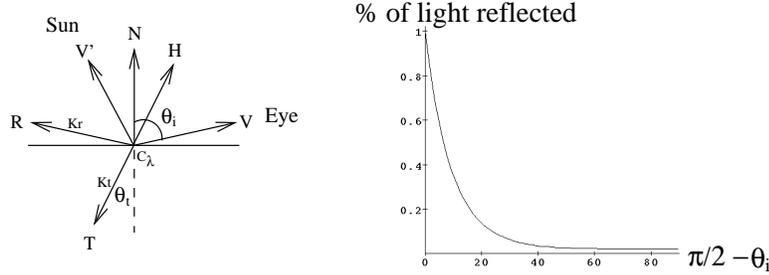


Fig. 13. a. Global illumination, b. Contribution of reflexion

In the formulation of the illumination model, some specific treatment has to be done to correctly render the sea. First, we have to take in account the absorption of the sea. Actually when the refracted ray does not hit any object in the sea, we can directly use a predefined color of the sea. In [11], there is a list of the main colors I_{sea} of the different oceans and seas around the world. The retrodiffusion of the light is included in these colors.

Ocean/Sea	H	C
Pacific Ocean	61°	473
Indian Ocean	80°	474
Mediterranean Sea	74°	473
Atlantic Ocean	70°	483
Baltic Sea	55°	540
Panama Gulf	34°	487, 2
Senegal Coast	<i>Cloudy</i>	512

where H is the angle (degrees) between the sun and the horizon, C the main color (nm) of the sea.

The part of energy reflected and refracted are theoretically determined by the Fresnel laws. The reflected coefficient K_r and refracted coefficient K_t are functions of the incident angle θ_i and the refracted angle θ_r . But, only approximations of K_r , K_t can be established. In physics, two kinds of approximation are commonly used [10]. We choose the less expensive of them :

$$\begin{aligned}
K_{r1} &= \frac{n_1 \cos \theta_i - n_2 \cos \theta_t}{n_1 \cos \theta_i + n_2 \cos \theta_t} \\
K_{r2} &= \frac{n_2 \cos \theta_i - n_1 \cos \theta_t}{n_2 \cos \theta_i + n_1 \cos \theta_t} \\
K_r &= \frac{K_{r1}^2 + K_{r2}^2}{2} \\
K_t &= 1 - K_r
\end{aligned} \tag{17}$$

Absorption occurs when the viewing ray travels through the scene. We use always a coefficient $K_\alpha = e^{-\alpha d}$ where α is the extinction coefficient and d the distance between the object and the current point. For the sea, the absorption is treated as a particular way. If the transmitted ray does not hit any object, there is no absorption and we can directly return the color of the sea, else the absorption is taken in account and the intensity returned I_t is given by :

$$I_t = K_\alpha I_{object} + (1 - K_\alpha) I_{sea} \tag{18}$$

In Equation 16, I_{sea} will actually be C_λ (resp. I_t) when the ray does not hit (resp hit) any object under the sea.

7.3 Color model

We use a spectral model to compute the color of the refracted rays. When a ray touches an object under the water, we transform the color of this object, coded in the RGB format, into its spectral diagram as detailed Glassner method [7]. [11] gives the spectrum of absorption of the ocean water (Fig 14.). We then compute the new spectrum of the object after the absorption. And finally, this spectrum is recoded in RGB passing through the XYZ code format [9].

To compute the RGB value of I_{sea} , from Table 7.2, we construct a bell-shaped curve around main color of the chosen I_{sea} in the complete spectrum space. Finally, the spectrum is translate into equivalent RGB values.

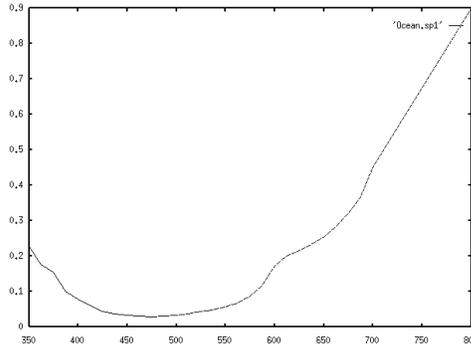


Fig. 14. The Ocean water absorption spectrum

8 Results

Plate 1. presents a part of multi-wave trains ocean. Plates 2. show the diffraction phenomenon on a small rectangular ocean section. The waves come from the background to the foreground. The main part of the pictures represents the waves diffracted after the jetty. Plates 2.Left presents a non perturbed surface and Plate 2.Right shows the addition of capillary waves. Finally, Plate 3. presents a more general scene with capillary waves, reflection and transmission. Its rendering (1050x675, 8 ray per pixels) takes 30 minutes on a Silicon Graphics O2 with an R5000 processor.

9 Conclusion

This paper has presented two contributions to the rendering of coastal scenes. The first concerns improvements of the geometrical modeling of the sea proposed in [8] (diffraction, multi-wave trains, capillary waves). The last is an illumination model adapted to this kind of scenes. The simplicity of the illumination model combined with the dynamic geometrical modeling of the scene allows efficient animations.

References

1. ARISTAGHES, C., AND ARISTAGHES, P. *Théories de la houle, houle réelle, propagation de la houle*. Service technique central des ports maritimes et voies navigables, 1985.
2. BECKER, B. G., AND MAX, N. L. Smooth transitions between bump rendering algorithms. In *SIGGRAPH'93* (1993), pp. 183–190.
3. BIESEL, F. Study of wave propagation in water of gradually varying depth. In *Gravity Waves* (1952), U.S. National Bureau of Standards Circular 521, pp. 243–253.
4. BLINN, J. F. Simulation of wrinkled surface. In *SIGGRAPH'78* (1978), vol. 12(3), pp. 286–292.
5. COOK, AND ROBERT, L. Shade trees. In *SIGGRAPH'84* (1984), vol. 18(3), pp. 223–231.
6. FOURNIER, A., AND REEVES, W. T. A simple model of ocean waves. In *SIGGRAPH'86* (1986), vol. 20, pp. 75–84.
7. GLASSNER, A. S. How to derive a spectrum form an rgb triplet. *Computer Graphics and Applications* 8(3) (1988), 60–70.
8. GONZATO, J. C., AND LE SAËC, B. A phenomenological model of coastal scenes based on physical considerations. In *8th Eurographics Workshop on Computer Animation and Simulation* (1997), Springer-Verlag, Ed., pp. 137–148.
9. HALL, R. *Illumination and Color for Computer Generate Imagery*. Springer-Verlag, 1989.
10. HARDY, A. C., AND PERRIN, F. H. *Principles of Optics*. McGraw-Hill, 1932.
11. IVANOFF, A. *Introduction à l'Océanographie*, vol. II. Vibuert, Paris, 1975.

12. KINSMAN, B. *Wind Waves, their generation and propagation on the ocean surface*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1965.
13. LACOMBES, H. *Cours d'Océanographie Physique*. Gauthier-Villars, Paris, 1965.
14. LOGIE, J. R., AND PATTERSON, J. W. Inverse displacement mapping in general case. In *Computer Graphics Forum* (1995), vol. 14(5), pp. 261–273.
15. PEACHEY, D. R. Modeling waves and surf. In *SIGGRAPH'86* (1986), vol. 20, pp. 65–74.
16. SCHLICK, C. An inexpensive brdf model for physically-based rendering. In *EUROGRAPHICS'94* (1994), vol. 13, pp. 233–246.
17. TS'O, P. Y., AND BARSKY, B. A. Modeling and rendering waves : Wave-tracing using beta-splines and reflective and refractive texture mapping. In *ACM Transactions on Graphics* (July 1987), vol. 6, pp. 191–214.
18. WATT, A., AND WATT, M. *Advanced Animation and Rendering Techniques*. Addison-Wesley, 1990.
19. WHITTED, T. An improved illumination model for shaded display. *Comm. ACM* 26(6) (1980), 342–349.



Plate 1: Multiple wave trains

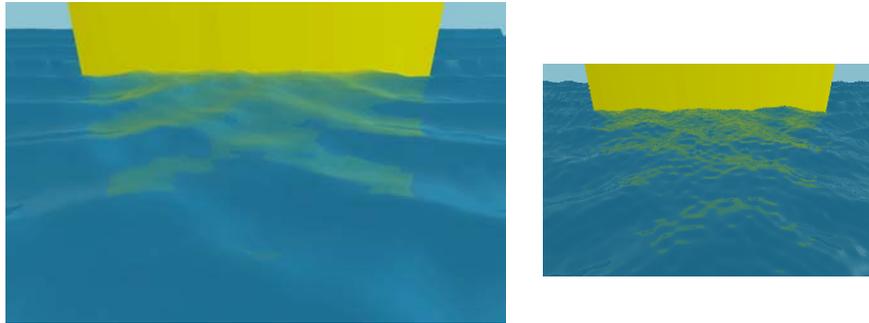


Plate 2: Simple diffraction after a jettie and same scene with additionnal noise

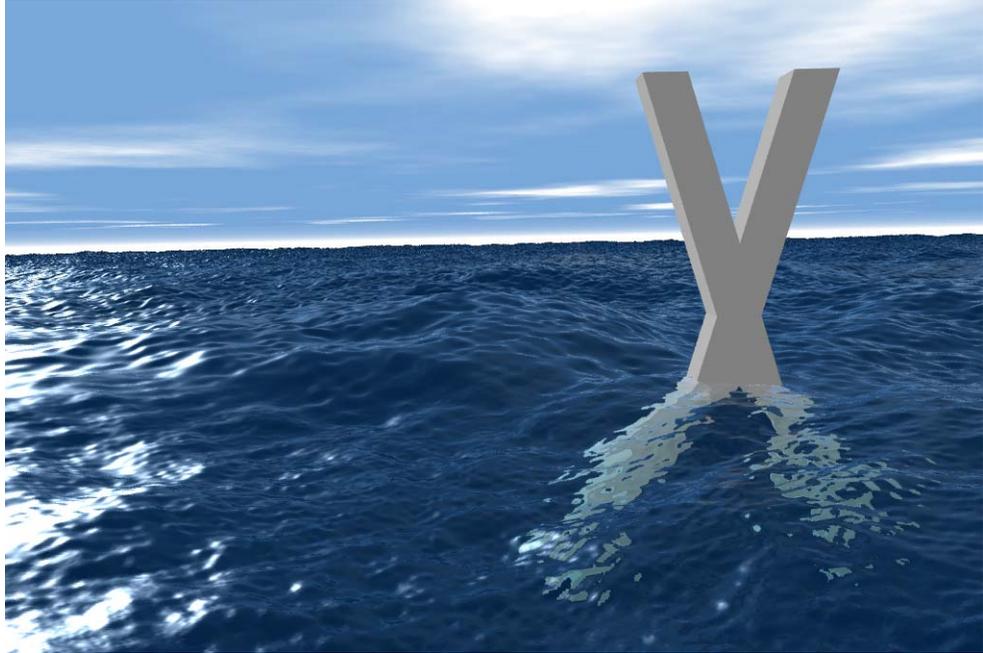


Plate 3: A more general scene with capillary waves, reflection and transmission



Plate 4: Ocean view by night