On Modeling and Rendering Ocean Scenes Diffraction, Surface Tracking and Illumination

Jean-Christophe Gonzato and Bertrand Le Saëc

LaBRI[1]
351 cours de la libération,
33405 Talence
France
[gonzato|lesaec]@labri.u-bordeaux.fr

ABSTRACT

This paper is devoted to ocean images. First, we propose an improvement of our model [Gonza97] accounting for diffraction, 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, water waves, dynamic wave tracing, diffraction, transmission, displacement mapping, illumination, ray tracing.

1 Introduction

The goal of this paper is to generate photorealistic images of coastal scene intended for animations. Our main goal is to obtain images 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.

In [Gonza97], we presented an algorithm called Dynamic Wave Tracing (DWT) providing a geometrical definition of a coastal scene. The aim of this algorithm is to keep the same accuracy all over the scene and to account for refraction. 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. This algorithm is the base of the present paper. The shape of waves is computed along these rays by an improved version of the Fournier-Reeves wave [Fourn86].

Here, we present two enhancements of this previous paper. The first one is the addition of jetties in the scenes and so the pos-

[1] 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.*

sibility to deal with diffraction. For this, we use an adaptation of the method proposed by Larras [Lacom65] that is used to simulate wave propagation in ports. The second improvement is a generalisation to multi-waves trains and capillary waves in order to increase the realism of the images. A tessellation of each wave trains are computed. Our method treats these waves by using an algorithm based on displacement mapping [Becke93][Logie95] to define the real surface of the oceans.

Finally, we propose a simple rendering algorithm for these scenes. This algorithm is based on the Incremental Ray Tracing algorithm. Our illumination model is an adaptation of the one proposed by Whitted [Whitt80]. 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 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.

2.1 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 [Biese52]. Afterwards, several other solutions have been proposed by physicists, but they all involve expensive computation. An empirical solution was proposed by Larras [Lacom65] :

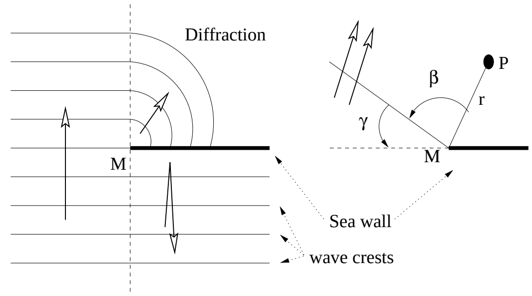

Figure 1: Wave diffraction theory

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 ($\beta > \frac{\Pi}{2}$). On this part, the waves are damped.

$$\begin{aligned} U_i &= 2\sqrt{\pi \tfrac{r}{L}} \sin(\tfrac{\pi}{4} - \tfrac{\beta}{2}) \\ U_r &= 2\sqrt{\pi \tfrac{r}{L}} \sin(\tfrac{\pi}{4} - \tfrac{\beta}{2} - \gamma) \end{aligned} \quad (1)$$

In these formula, L is the wave length of the incident wave, γ is the angle between the wave train and the pier (fig 1.).

In [Kinsm65] H_i (resp. H_r) is a function of U_i (resp U_r) (fig 2.) by :

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

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

when $\mathbf{U_{i/r} \geq 0.75}$

$$\begin{aligned} H_{i/r}(U_{i/r}) = 0.014 e^{-0.4 U_{i/r} + 3} \times \\ \sin(32 \ln(1 + \tfrac{U_{i/r}}{15})(U_{i/r} - 0.75)) + 1 \end{aligned} \quad (3)$$

otherwise

$$H_{i/r}(U_{i/r}) = 0.03 + 0.47e^{0.96U_{i/r}} \tag{4}$$

$H_{i/r}$ (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

$$\Phi = 4\pi\frac{r}{L}\sin\gamma\cos\beta + \gamma \tag{5}$$

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

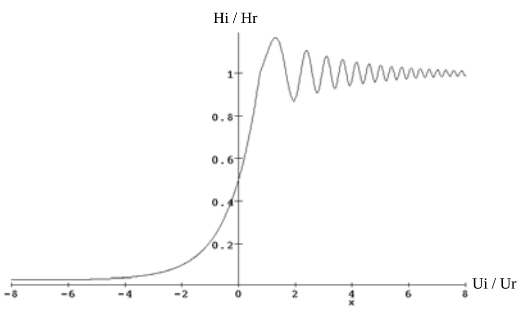

Figure 2: H_i, H_r as a function of U_i, U_r

2.2 Our method

2.2.1 Diffraction.

All rays are traced step by step from the open sea, the DWT algorithm [Gonza97] 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 3.a), we flag its neighbours. Hence, we store the first ray (R1 on fig 3.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.

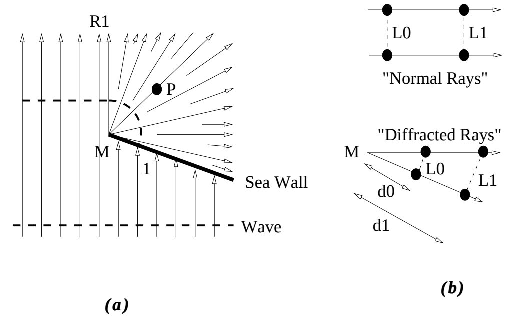

Figure 3: Our method to compute diffraction, refraction

2.2.2 Refraction in Diffraction.

Due to variation of depth, the amplitude of waves is modified by the refraction. This modification is doned by the law :

$$A_1 = A_0\sqrt{\frac{L_0}{L_1}} \tag{6}$$

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 3.b).Since we have to take in account simultaneously refraction and diffraction phenomena we use the following equation for the rays behind the pier :

$$A_1 = A_0\sqrt{\frac{d_1}{d_0}\frac{L_0}{L_1}} \tag{7}$$

where d_0 and d_1 are the distances from the point to the pierhead before and after a step of the progression (fig. 3.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 ($A_1 = A_0$).

2.3 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 4.a shows how rays evolve in the scene, and Fig 4.b is the corresponding wave train. A discontinuity is noticable 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 tesselation).

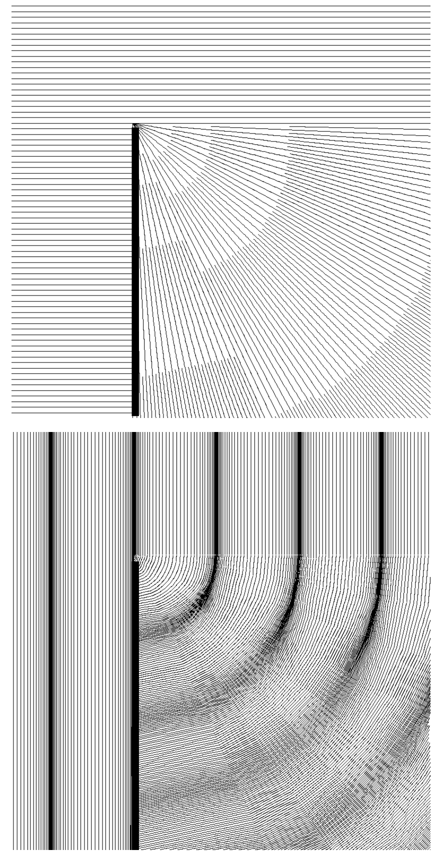

Figure 4: a. DWT algorithm, b. Waves

3 Surface tracking

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

Peachey [Peach86] and Ts'o-Barsky [Tso87] 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.

3.1 Wave trains as displacement mapping

Displacement mapping was introduced by Cook [Cook84]. This technique is used to texture objets when shortcuts of computation as BRDF [Schli94] and bump mapping [Blinn78] become noticable 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 [Becke93]. 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 independentally and a specific tesselation 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 tesselations are embedded in a bounding box H_{box} which height depends on the amplitudeof the different wave trains. H_{box} can be easily computed during the generation of wave trains.

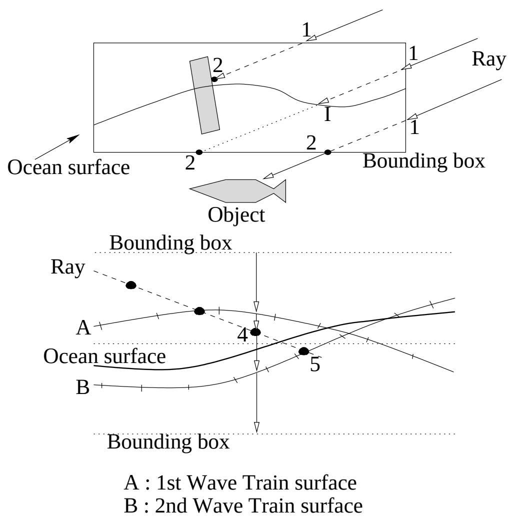

Figure 5: Ocean surface tracking

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 5), 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 5). 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 5). 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 :

$$N = (1.0, 0.0, -\sum_{i=1}^{n} \frac{x_i}{z_i}) \wedge (0.0, 1.0, -\sum_{i=1}^{n} \frac{y_i}{z_i}) \tag{8}$$

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.

Figure 6: Multiple wave trains

3.2 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 :

$$F(n,x,y,t) = \sum_{i=1}^{n} \frac{noise(x.k^i, y.k^i, t)}{k^i} + D \times \frac{noise(x.k^{n+1}, y.k^{n+1}, t)}{k^{n+1}} \tag{9}$$

Where *noise* is a classical function obtained using cardinal spline interpolation

[Watt90]. 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.

Figure 6 presents a sample of image of multiple wave trains and capillary waves genrated by our algorithm.

4 Rendering

4.1 Illumination model

Our implementation uses a classical ray-tracing illumination model [Whitt80] :

$$I = I_a k_d C_\lambda + K_\alpha I_s (k_d C_\lambda (N \cdot V') + k_s (N \cdot H)^p) + K_r I_r + K_t I_t \tag{10}$$

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

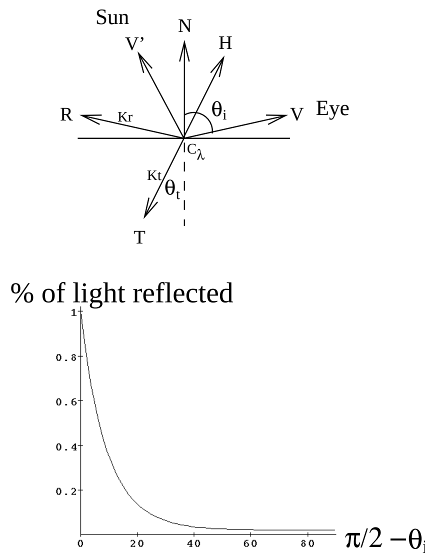

Figure 7: a. Global illumination, b. Contribution of reflexion

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

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 [Ivano75], 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^o	473
Indian Ocean	80^o	474
Mediterranean Sea	74^o	473
Atlantic Ocean	70^o	483
Baltic Sea	55^o	540
Panama Gulf	34^o	487, 2
Senegal Coast	*Cloudy*	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 [Hardy32]. We choose the less expensive of them :

$$\begin{aligned} K_{r1} &= \frac{n_1 \cos\theta_i - n_2 \cos\theta_t}{n_1 \cos\theta_i + n_2 \cos\theta_t} \\ K_{r2} &= \frac{n_2 \cos\theta_i - n_1 \cos\theta_t}{n_2 \cos\theta_i + n_1 \cos\theta_t} \\ K_r &= \frac{K_{r1}^2 + K_{r2}^2}{2} \\ K_t &= 1 - K_r \end{aligned} \tag{11}$$

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 :

$$I_t = K_\alpha I_{object} + (1 - K_\alpha) I_{sea} \tag{12}$$

In Equation 10, I_{sea} will actually be C_λ (resp. I_t) when the ray does not hit (resp hit) any object under the sea.

4.3 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 detailled Glassner method [Glass88]. [Ivano75] gives the spectrum of absorption of the ocean water (Fig 8.). 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 [Hall89].

To compute the RGB value of I_{sea}, from Table 4.2, 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.

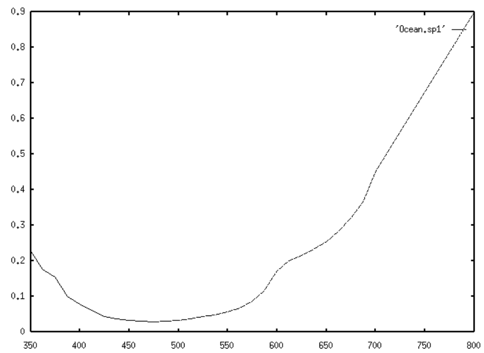

Figure 8: The Ocean water absorption spectrum

5 Results

Plate 1. presents a simple flat face perturbed by the linear fractal noise. 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 cappilary 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.

6 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 [Gonza97] (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

[Arist85] 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.

[Becke93] Becker, B. G., and Max, N. L. Smooth transitions between bump rendering algorithms. In *SIGGRAPH'93* (1993), pp. 183–190.

[Biese52] 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.

[Blinn78] Blinn, J. F. Simulation of wrinkled surface. In *SIGGRAPH'78* (1978), vol. 12(3), pp. 286–292.

[Cook84] Cook, and Robert, L. Shade trees. In *SIGGRAPH'84* (1984), vol. 18(3), pp. 223–231.

[Fourn86] Fournier, A., and Reeves, W. T. A simple model of ocean waves. In *SIGGRAPH'86* (1986), vol. 20, pp. 75–84.

[Glass88] Glassner, A. S. How to derive a spectrum form an rgb triplet. *Computer Graphics and Applications 8(3)* (1988), 60–70.

[Gonza97] 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.

[Hall89] Hall, R. *Illumination and Color for Computer Generate Imagery.* Springer-Verlag, 1989.

[Hardy32] Hardy, A. C., and Perrin, F. H. *Principles of Optics.* McGraw-Hill, 1932.

[Ivano75] Ivanoff, A. *Introduction à l'Océanographie*, vol. II. Vibuert, Paris, 1975.

[Kinsm65] Kinsman, B. *Wind Waves, their generation and propagation on the ocean surface.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1965.

[Lacom65] Lacombes, H. *Cours d'Océanographie Physique.* Gauthier-Villars, Paris, 1965.

[Logie95] Logie, J. R., and Patterson, J. W. Inverse diplacement mapping in general case. In *Computer Graphics Forum* (1995), vol. 14(5), pp. 261–273.

[Peach86] Peachey, D. R. Modeling waves and surf. In *SIGGRAPH'86* (1986), vol. 20, pp. 65–74.

[Schli94] Schlick, C. An inexpensive brdf model for physically-based rendering. In *EUROGRAPHICS'94* (1994), vol. 13, pp. 233–246.

[Tso87] 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.

[Watt90] Watt, A., and Watt, M. *Advanced Animation and Rendering Techniques.* Addison-Wesley, 1990.

[Whitt80] Whitted, T. An improved illumination model for shaded display. *Comm. ACM 26(6)* (1980), 342–349.

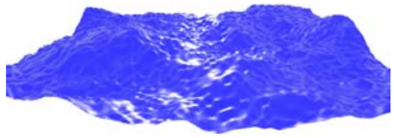

Plate 1: A simple perturbed face

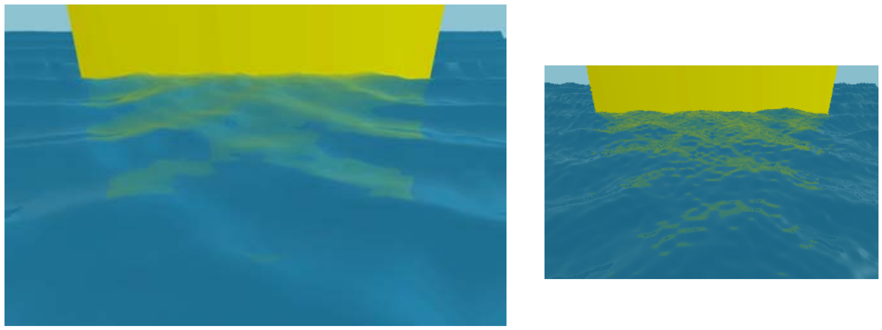

Plate 2: Simple diffraction after a jettie and same scene with additionnal noise

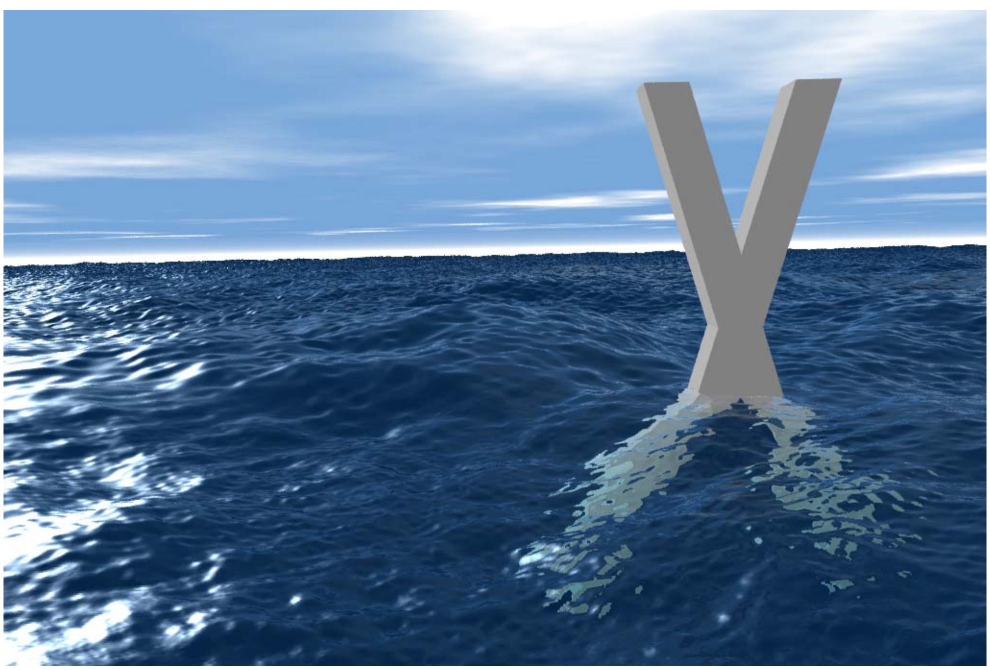

Plate 3: A more general scene with capillary waves, reflection and transmission