Realistic water rendering

Yuan Tian, Yufei Lin, Jiao Zhang

University of Southern California

***Abstract* – One of the biggest problems in computer graphics is creating and rendering realistic water. Realistic water rendering requires that the wave of water, the water refraction and reflection are correct. In this paper, an implementation for rendering realistic water is presented. The rendering first simulates water waves by using normal map. Then, in order to improve the level of realism, Render to Texture (RTT) and Projective Texturing technology are used to achieve the water reflection and refraction.**

***Index Terms* – *water, rendering, reflection,* refraction**

I. INTRODUCTIONS

In computer graphics, rendering the realistic natural phenomena is an important part since the last two decades [1]. Among all the tasks that simulating natural phenomena, rendering realistic water is one of the most important challenges [2]. In this paper, we are going to rendering realistic water. In order to create realistic images of water, there are three important components we need to address, the wave of water, reflection of water and refraction of water.

The wave generation is the basis of the realistic water implementation because it significantly improves the level of realism in a scene. A very common method for rendering realistic water is to simulate water waves in a small area and repeat the wave motion on the surface. This approach provides a great simulation, but it cannot give enough realism because of the change of water depth.

Peachey [3], and Tso and Barsky [4] have used different shapes to model shallow water waves. Perlin described noise synthesis approach [5] to render the ocean surface which has different view with different distance. Mastin [6] has used technique oceanography community to simulate deep ocean waves, and Fournier and Reeves had more discussion about water wave. In this paper, we address the water wave generation by using the DirectX 11 framework which employ normal maps to create visual details.

Water has many different visual properties, two of those are that it reflects light from the environment and that it refracts light that passes through it. This realism can be achieved if reflections and refractions are computed. Reflections and refractions contribute the most to the perceived realism of the simulation of water surfaces. Refraction is the change in direction of a wave, such as light, in relation to a change in its speed. Reflection is the change in direction of light, or more general of a wave, on the interface of two different substances so that the light returns into the media it comes from. For realistic water rendering, it is essential to handle the interaction between the water surface and light correctly.

This paper describes the process for rendering realistic water. First, the necessary basics to do the simulation will be presented. The basics will describe the two most common simulation approaches and their impact on rendering techniques due to different data structures. After the basics, a section about implementation shows a few methods about how water can be rendered. After that, there are some pictures to show results of simple rendering of a water surface.

II. BASICS

For simulation and rendering of water, sophisticated physical models are needed. This section will cover two basic technologies which are used to do the realistic water rendering. One is the render to texture (RTT) technology, and the other is the projective texturing.

A. *Render to texture (RTT)*

RTT is a rendering technology that renders a 3D model to a 2D texture and renders the 2D texture to a rectangle image. It can be used to manufacture reflection texture and refraction texture, and it is important to make the water surface reflection and refraction phenomenon.

Rendering to texture, or "texture baking," allows you to create texture maps based on an object's appearance in the rendered scene. The textures are then “baked” into the object, that is, they become part of the object via mapping, and can be used to display the textured object.

B. *Projective Texturing*

Projective texturing map is a method of texture mapping that allows a textured image to be projected onto a scene as if by a slide projector. Projective texture mapping is useful in a variety of lighting techniques. It is one of the most important technologies to render 3D graphics in real-time. It achieves many advanced effects, such as Soft Shadow, Projective Light Map, and the Water Reflection. It makes a projection between a 2D texture and a 3D model. In other word, it uses a special camera to obtain the 3D object and projects it on a 2D texture. In this paper, this technology will be used to get the refraction and reflection of the object. It uses a camera view point to render a 3D scene onto the 2D back buffer.

III. Implementation

Several rendering algorithms are used in Our implementation, such as the normal maps, RTT and the projective texturing map.

A. *The wave generation*

The importance of plausible modeling of any water surface is that the visual characteristics of water surfaces are very distinct.

In our model we use a normal map to create a static water ripple. Normal mapping is a mapping technique that uses RGB to record the displacement of surface normal. In normal mapping, R and B represent two directions parallel to the horizontal plane, and G represents the direction perpendicular to the horizontal plane. In a pixel of normal mapping, the larger its G component, the more vertical it faces. This applies equally to R and B components. According to this principle, we can automatically generate normal maps through some plug-ins. However, considering that the normal mapping of the water surface is usually generated from actual measured data, we got a suitable normal map through search engine, and then we modified it with the Nvidia normal map plug-in in photoshop to make the water surface have more roaring waves. In our modification, we added the G component of each pixel of the normal map and reduced the R and B components, so as to make the water surface fluctuate more vertically. The final version of the normal map is shown in Figure 1.

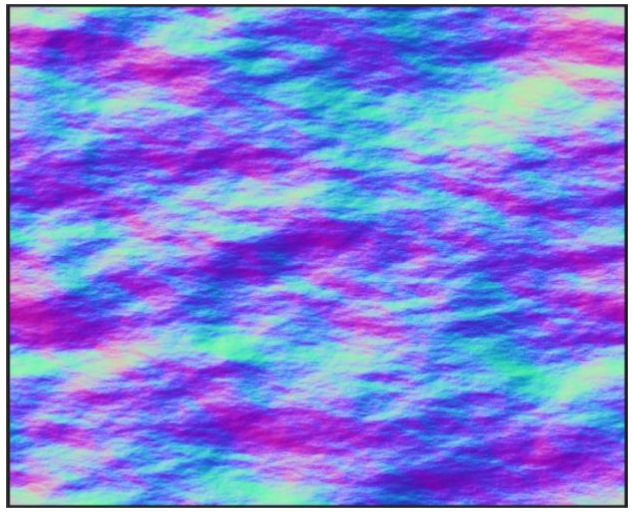

Fig. 1. A Normal Map Used for Water Surface

However, normal mapping can only be used to create static surface ripples. In order to make the water surface produce dynamic ripple, we need to use a common texture processing technology-moving texture technology. We use time as a variable in the shader to control the sampling coordinates in each frame.

B. *The water reflection*

Reflection is the change in direction of light, or more general of a wave, on the interface of two different substances so that the light returns into the media it comes from. There are two types of reflection: Specular Reflection and Diffuse Reflection. Only specular reflection is covered in this section [1].

The law of reflection states, that the angle θ between the incoming light ray r_i and the surface normal $\vec{n}$ is equal to the angle θ_r between the reflected light ray r_r and the surface normal. Let $\vec{v}$ be the inverse direction vector of the incoming light ray and $\vec{v}_r$ the direction vector of the outgoing light ray, then the direction of the outgoing ray can be calculated in vector form as followed, assuming that $\vec{n}$ and $\vec{v}$ are normalized:

$$\cos\theta = \vec{n} \bullet \vec{v}$$

$$\vec{w} = \vec{n}\cos\theta - \vec{v}$$

$$\vec{v}_r = \vec{v} + 2\vec{w}$$

In this paper, we firstly get the water reflection view texture by using RTT. In order to do that, we need to find out the projection camera which is the symmetrical of the observe camera about the water surface. Secondly, RTT is used to obtain the reflection view texture by using the projection camera. Then, this reflection view texture is projected on the water surface so each vertex on the water surface can acquire the UV of the reflection view texture. Finally, all UV coordinates are used to mixed the reflection view texture with water surface in the shading stage and the result can be generated. See Fig 3. The result of reflection of water.

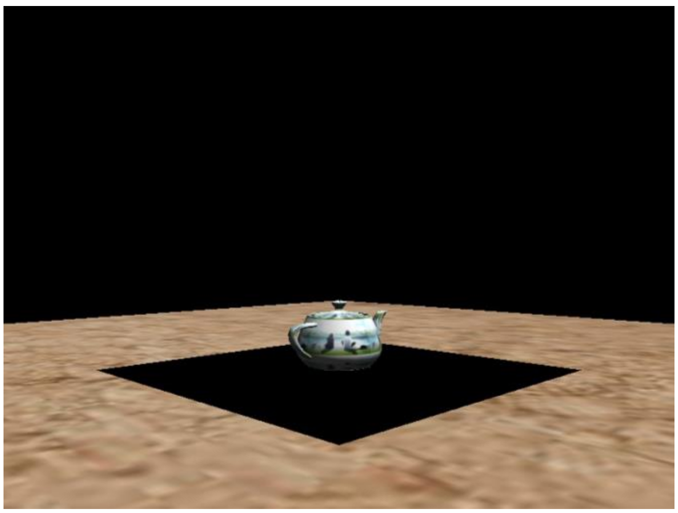

Fig. 2 Realistic water rendering without reflection.

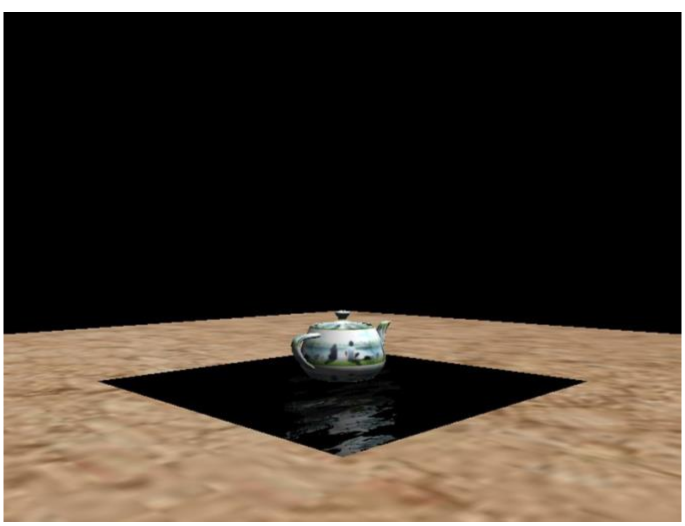

Fig. 3 Realistic water rendering with reflection

C. The water refraction

Refraction is the change in direction of a wave, such as light, in relation to a change in its speed. Such changes in speed happen if the media in which the wave travels changes [1].

Snell's law describes this behavior and states that the angle θ be- tween the incoming ray r_i and the surface normal $\vec{n}$ is related to the angle θ_t between the refracted light ray r_t and the inverse normal $\vec{n}_t$.

$$\frac{\sin\theta}{\sin\theta_t} = \frac{v_1}{v_2} = \frac{n_2}{n_1}$$

where v1 and v2 are the wave velocities in the corresponding media and n1 and n2 are the indices of refraction depending on the media. To get the refracted angle θ_t we can use Snell's law:

$$\begin{aligned} \cos^2\theta_t &= 1-\sin^2\theta_t && Pythagoreanidentity \\ &= 1-\eta^2\sin^2\theta && Snell'slaw, where \ \eta = \frac{n_1}{n_2} \\ &= 1-(\eta^2-\eta^2\cos^2\theta) \end{aligned}$$

The direction of the refracted light ray r_t with its direction vector $\vec{v}_t$ can be calculated as followed:

$$\begin{aligned} \cos^2\theta_t &= \sqrt{\cos^2\theta_t} \\ \vec{n}_t &= -\vec{n}\cos\theta_t \\ \vec{v}_t &= \eta\vec{w} + \vec{n}_t \end{aligned}$$

The last equation is due to:

$$\vec{\omega}_t = \frac{\vec{\omega}}{|\vec{\omega}|}\sin\theta_t = \vec{\omega}\frac{\sin\theta_t}{\sin\theta} = \eta\vec{w}$$

In our implementation, in order to see all the object below the surface through the water surface, we need to generate the corresponding view texture in the same way with reflection view texture. In the implementation of obtaining refraction view texture, the observer camera plays the role of projection camera instead of the symmetrical one in the reflection view texture. Beyond that, the rest of the steps are basically the same as reflection one.

Firstly, we get a refraction view texture by rendering the object of water surface onto a 2D texture. Then, the refraction view texture is projected on the water surface so that all vertexes get their sample coordinate. Finally, the refraction view texture is mixed with the water surface and the shader can produce the refraction effect . See Fig.4. and Fig 5.

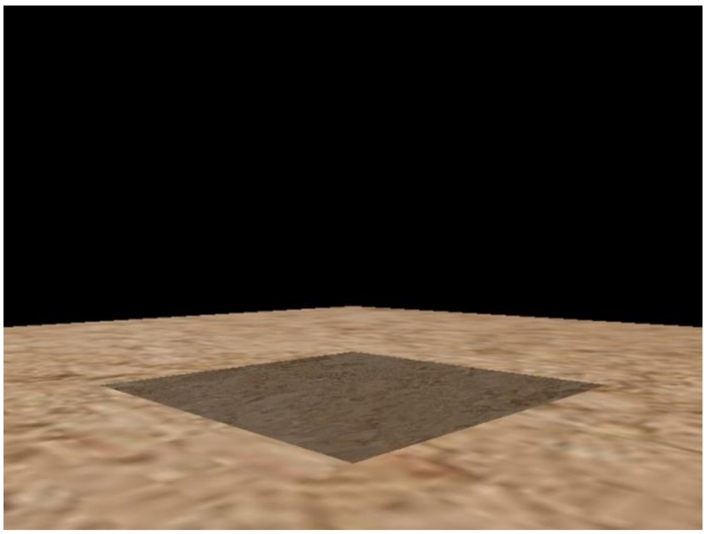

Fig. 4 Realistic water rendering with refraction.

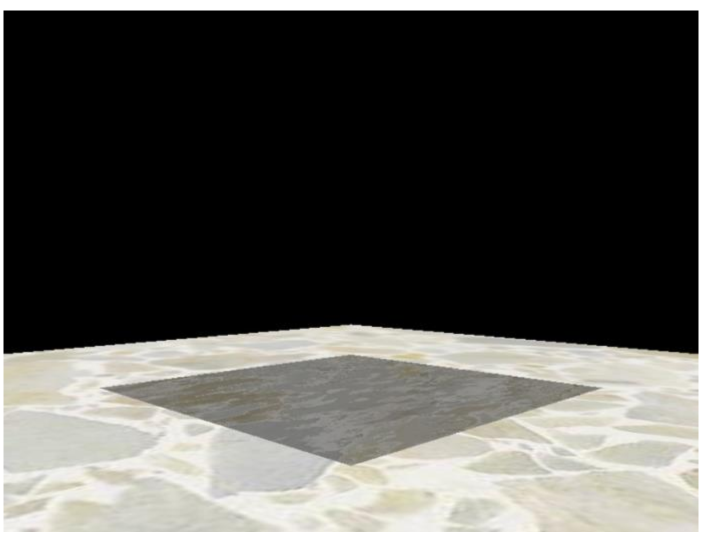

Fig. 5 Realistic water rendering with refraction.

IV. RESULTS

The performance of the realistic water rendering is as follow. See Fig. 6 for screenshots of the viewpoint representative of the visual quality that can be achieved when applying the wave generation, water refraction and reflection.

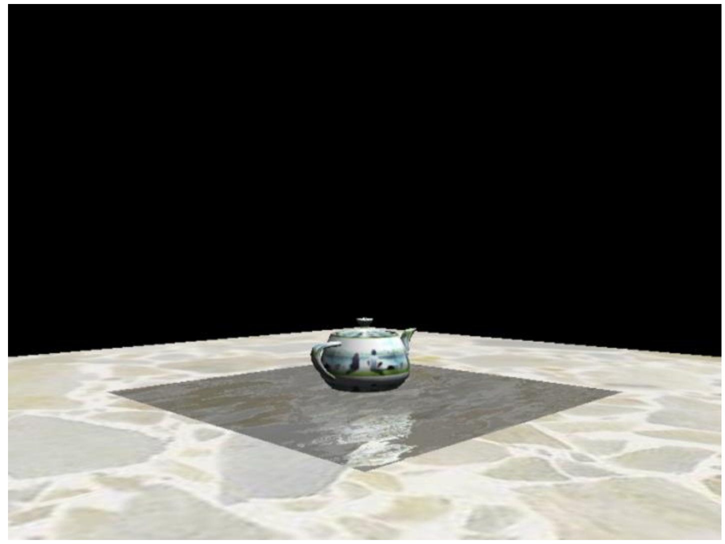

Fig. 6 Final result of realistic water rendering.

V. CONCLUSION

This paper mainly covers an implementation of water surface rendering. Among the implementations covered in this paper, normal mapping technology and texture moving technology are mainly used to create dynamic ripple effect for the water surface, in which normal texture is used to create static waves on the water surface, and texture moving technology is used to move static waves created by normal texture toward certain direction in each frame. In addition, the combination of RTT and projective texturing technology is mainly used to implement the reflection and refraction of water surface. In this implementation, RTT technology is mainly used to obtain the texture of reflection view and refraction view and projective texturing technology is mainly used to apply the texture of reflection view and refraction view onto water surface correctly. With the help of the all the technology mentioned above, realistic water surface effect can be implemented successfully. At the same time, this method is efficient and cost-saving.

REFERENCES

[1] B. Fleck, "Real-time rendering of water in computer graphics," Unpublished Manuscript, 2007.

[2] S. Premože and M. Ashikhmin, "Rendering natural waters," in Computer Graphics Forum, 2001.
[3] D. R. Peachey, "Modeling waves and surf," in ACM Siggraph Computer Graphics, 1986.
[4] P. Y. Ts'o and B. A. Barsky, "Modeling and rendering waves: wave-tracing using beta-splines and reflective and refractive texture mapping." ACM Transactions on Graphics (TOG), vol. 6, (3), pp. 191-214, 1987.
[5] K. Perlin, "An image synthesizer," ACM Siggraph Computer Graphics, vol. 19, (3), pp. 287-296, 1985.
[6] G. A. Mastin, P. A. Watterberg and J. F. Mareda, "Fourier synthesis of ocean scenes," IEEE Comput. Graphics Appl., vol. 7, (3), pp. 16-23, 1987.
[7] M. Nätterlund, "Water surface rendering," Unpublished Master' s Thesis, Umeå University, Umeå, Sweden, 2008.