GPU-based Ocean Rendering

Yung-Feng Chiu Chun-Fa Chang
National Tsing Hua University
{yfchiu, chang}@ibr.cs.nthu.edu.tw

Abstract

We present efficient algorithms for real-time rendering of ocean using the newest features of programmable graphics processors (GPU). It differs from previous works in three aspects: adaptive GPU-based ocean surface tessellation, sophisticated optical effects for shallow water, and spray dynamics for oscillating waves. Our tessellation scheme not only offers easier level-of-detail (LOD) control but also avoids the loading of vertex attributes from CPU to GPU at each frame. The object-space wave sampling approach allows us to produce sophisticated optical effects for shallow water and to implement a state-preserving particle system for simulating spray motions interactively without uploading the attributes for all particles from CPU to GPU per frame. This practical method can render realistic-looking water surfaces at the speed of about 10 frames per second on a PC (with a 2.8GHz Pentium 4 processor, 512 MB RAM, PCI Express, and an NVIDIA GeForce 6600 GPU).

Water surface representation

We use GPU to tessellate the visible region of the water surface according to the current viewpoint (See Fig. 1).

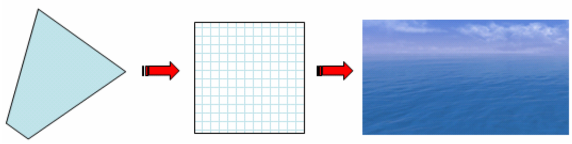

Fig. 1 The left is the visible ocean region in object space. The middle is the corresponding screen-space rectangle discretized to a regular grid which is represented as a vertex texture. The right is the corresponding region with waves applied on it.

Object-space wave sampling

The grid points of the visible water surface provide the positions where the height field of wave is evaluated. So, we subsequently sample a set of height maps at different scale to get a different spatial resolution of waves as follows:

$$H_w = \sum^{n} tex2D(wave_i, V_{pos}.xz / scale_i)$$

where *tex2D()* is an intrinsic sampling function of GPU, V_{pos} is the grid point of water surface, and $scale_i$ controls the filtering that is done during sampling.

Optical effects

We take into account the per-pixel height of ocean waves instead of roughly using the plane y = 0 to clip those objects across the water surface accurately for optical effects (Fig. 2 Right). Fig. 2 Left shows artifacts caused by using static height y=0 to clip across objects.

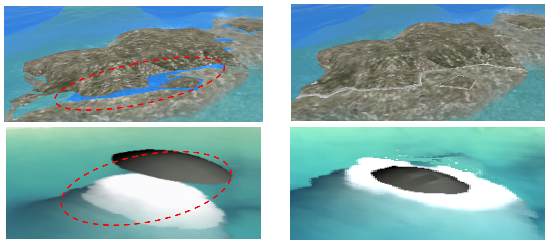

Fig. 2

Results

Fig. 3 demonstrates six screenshots captured from our real-time rendering system.

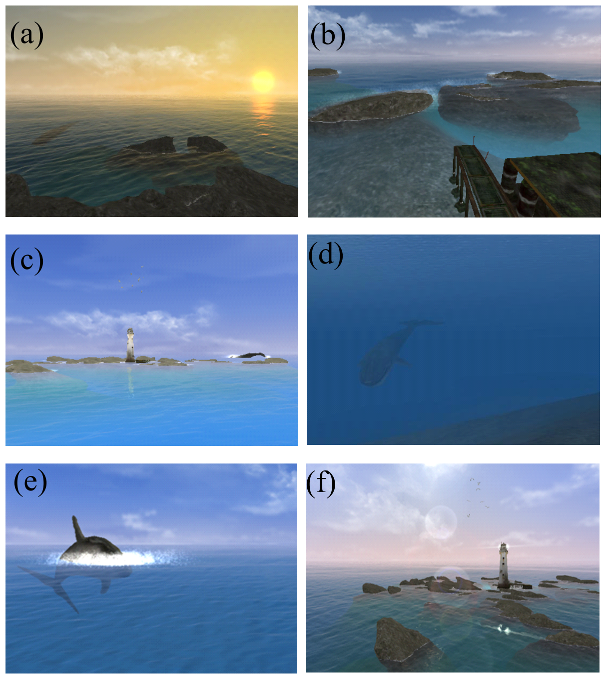

Fig. 3 (a) The sunset and sky light illumination. (b) The spray dynamics around rocks. (c) Local reflection and depth-dependent water color. (d) The underwater scattering effect. (e) Spray dynamics around the whale which is falling into water. (f) Refraction and Fresnel effects.