Desalination and Water Treatment
www.deswater.com

doi: 10.5004/dwt.2018.22377

121 (2018) 134–138
July

Large-scale lake simulation techniques applied to water resource management system

Junjian Tang

Beijing International Studies University, Beijing, China, email: townjean@126.com

Received 24 February 2018; Accepted 15 April 2018

ABSTRACT

Optimized water resource management depends on reliable and detailed information which can be described in virtual waterscape simulation. Addressing virtual lake-waterscape reality is important for a water resource management system. Many researchers use the same methodology as being used in ocean simulation for lake simulation. However, there is a big difference between the real lake and the effect of lack simulation. This paper is aimed at using random noise to improve large-scale lake simulation. The procedures are going to be taken as follows: first, using Perlin noise to model lake waves. Second, in order to create appropriate dense mesh, mesh points generated in screen space are going to be projected into world space. Third, using efficient parallel computing power of the CUDA to calculate height field generation, vertex transformation and rendering in graphics processing unit. The rendering result shows that the CUDA method can reduce the computational complexity, meeting the requirements of real-time simulation.

Keywords: Virtual basin; Fluid simulation; Virtual reality; Perlin noise; CUDA

1. Introduction

Water resources are basically natural resources and strategic economic resources. For a better use of water resources, virtual basin water resource management system [1] was developed. It is also one of the important research topics in fluid simulation which is to express virtual basin's information in a realistic form. Real-time and efficient fluid simulation is always a challenging problem in computer animation, virtual reality and other fields. There is no single method to represent the physical laws of fluids and to capture large amount of detail in fluids. Although physically based fluid simulation can achieve physical effects of fluid, it is mainly applied to small-scale fluid simulation [2].

Taking large-scale fluid simulations in ocean as reference, based on different ocean characteristics, there are two kinds of common research methods: geometric methods for describing ocean surface in the space domain, statistical methods for describing the spectral characteristics of ocean surface in frequency domain.

The principle of large-scale lake simulations is similar to ocean simulation. However, the lake's wavelength and wave height are obviously different from that of the ocean's. If directly using simulation method of ocean, the simulation result is not reasonable because it is far away from that of the real lake. The aim of the presented research is to do some improvement to address this issue.

This paper is structured as follows. In Section 2, the conventional simulation model and the improved simulation model are presented. Section 3 shows a simulation process based on CUDA architecture and the simulation results and the comparison of the running performances between different platforms considered. In Section 4, the paper ends with a conclusion.

Presented at the 3rd International Conference on Recent Advancements in Chemical, Environmental and Energy Engineering, 15–16 February, Chennai, India, 2018.

2. Methodology

2.1. Ocean wave simulation spectral model

In the wave theory of oceanography, waves can be seen as superposition of waves with infinite directions, frequencies, amplitudes and phases. These waves constitute the spectrum of ocean waves defined by the energy and spectral forms of the waves [3]. Ocean wave spectrum not only contains a large number of two-order wave information but also describes the distribution of wave's internal energy. There are many ocean wave spectrums, such as Phillips, Joswap, which are widely used in oceanography [4]. The basic principle is based on the ocean statistics and empirical model, using a large number of superimposed sine wave to model the sea surface, then use the fast Fourier transform (FFT) to construct a finite region height field which is similar to the distribution of the wave spectrum, then through stitching to form a large area water and add a variety of light and shadow effect and other marine characteristics [5]. Because the Philips spectrum can better realize the surface change effect by wind, so many researchers use Philips spectrum to simulation. The Philips spectrum function is defined as follows:

$$P_h(K) = A\left[\exp\left(-\frac{1}{kL}\right)^2\right]^2 / k^4\left(\left|\hat{k}\right|\cdot\hat{w}\right)^2 \tag{1}$$

where $A = 0.00025$, $L = V^2/g$, is the maximum amplitude, V is the wind speed, g is the gravitational acceleration, $\hat{k}$ is the direction of wave motion, $\hat{w}$ is the direction of wind [6].

For the wave simulation based on spectral statistical model, wave height field is often expressed by function $h(x, y, t)$. In order to match the surface grid established in graphics rendering, the values of the obtained height field are usually assigned to the grid vertices of the corresponding water surface. At any moment in time, the height values of the grid vertices which can be matched with the horizontal position (x, y) can be represented by the sum of all sine functions on the complex domain.

$$\mathrm{h}(\mathrm{x},\mathrm{t}) = \Sigma_k h(x,t)\exp(ik\cdot x) \tag{2}$$

where t represents time, $k = (2\pi m/L_x, 2\pi m/L_z)$, L_x and L_z refer to the distances between two adjacent vertices of X and Z direction. From Eq. (1), using FFT, the height values of discrete points on the surface of the wave can be obtained. The wave shape on the sea surface is determined by the height amplitude obtained by the fast Fourier transform, and $h(x, t)$ is an independent Gauss wave, its motion can be described by the following form of spatial spectrum function:

$$P_h(k) = \langle\left|\bar{h}^*(k,t)\right|^2\rangle \tag{3}$$

As Eq. (1) described the motion law of $h(x, t)$, the Fourier amplitude formula of wave shape height field can be obtained:

$$\tilde{h}_0(k^2) = 1/\sqrt{2}(\zeta_r + i\zeta_i)^*\sqrt{P_h(k)} \tag{4}$$

where ζ_r and ζ_i are two independent Gauss random numbers. For the given wave propagation frequency $w(k)$, the Fourier amplitude of the height field of the water surface grid at moment t can be expressed as follows:

$$\tilde{h}(k,t) = \tilde{h}_0(k)^*\exp\{iw(k)t\} + \tilde{h}_0(-k)\exp\{-iw(k)t\} \tag{5}$$

The variation in the height of wave grid be calculated according to Eq. (5), and thus the wave model's height field is generated. Because the shape of the actual wave changes not only in the direction Z but also in the horizontal direction. This offset makes the wave become curved wave shape [7], as shown in Fig. 1.

In order to simulate the curved wave, the vertex coordinates in the plain should be modified, that will produce some offset. The offset can be described as:

$$X(x, z) = X(x, z) + \lambda D(X, t) \tag{6}$$

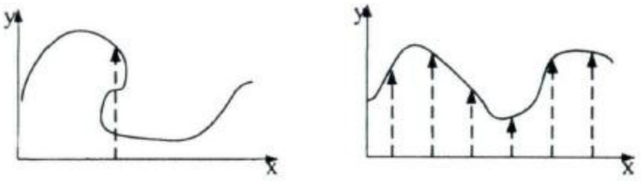
where X is the coordinate vector on the plane of X–Z, λ is used to control the magnitude of the offset amplitude, D is the horizontal offset vector. D can be described as:

$$\mathrm{D}(\mathrm{x},\mathrm{t}) = \Sigma_k(-i)\frac{k}{k}\tilde{h}(k,t)\cdot\exp(ik\cdot x) \tag{7}$$

The main process of calculating the height field is shown in Fig. 2.

2.2. Improved lake wave simulation method

For a large-scale lake simulation, many researchers use the same method as the method used in ocean simulation. They often use the spectral statistical model of ocean simulation and adjust the parameters of the spectrum to generate



(a) curved wave (b) calm sea surface wave

Fig. 1. Shape of curved wave.

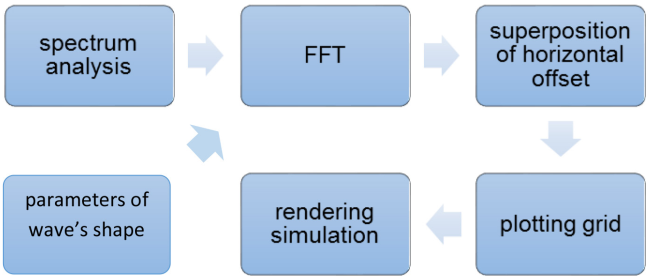


Fig. 2. Process of generating height field.

relatively smooth water surface, so as to achieve the effect of simulating lake surface. Although similar results can be obtained, there is still a gap between the actual lake and the simulation effect, because the spectral statistical model is the result of statistics of ocean waves, not lake's waves [8].

Lake waves are influenced not only by the wind but also by the width and depth. When wind blew over lake surface, its speed, direction, duration and range will bring different lake waves. On the other hand, the lake's width and depth also affect lake waves.

So there are many differences between lake wave and sea wave. The lake wave's shape is obviously different from the sea wave, its self-interference normally cause triangular waves. The wind may blew lake water easily than ocean water, due to its small water volume and the surface maybe more peaceful than that of ocean. There is no surge in the lake while there is a lot on the sea. The lake wave's height is usually lower and its wavelength is usually smaller. As a result, when lake waves are considered as a random phenomenon, it is possible to use random noise to simulate lake waves.

2.2.1. Create the grid of lake surface

When calculating the grid model, the orthographic projection algorithm is mostly used, which is divided into the following steps: (1) calculating the coordinates of the lake surface region in the world space and dividing it into a grid. (2) Transform the grid from world space to screen space. (3) Permutation mapping on the screen space grid. (4) Mesh renderer. This method transforms grid from the world coordinate system to the screen space coordinate system, so there will be no breakage and serious flickers. The disadvantage is that it needs another algorithm to extend it to the lake surface.

In this research, an improved method is used, which is divided into the following steps: (1) generate a regular grid in the screen space, facing the viewpoint. (2) Project the grid's points on the lake surface. (3) Transform the effective grid matrix on the lake surface to the grid in the world space coordinate system. (4) Permutation mapping on the grid in the world space coordinate system. (5) Mesh Renderer. This method has better spatial measurability and can achieve different levels details. It is suitable for generating large-scale lake surface grid.

2.2.2. Building a high field based on Perlin noise

The Perlin noise, proposed by Ken Perlin is continuous and smooth, is mostly used for simulating the natural landscape such as meadow [9–11].

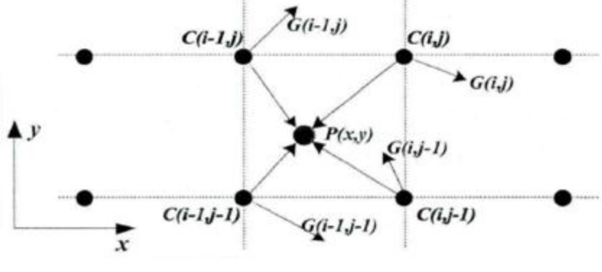


Fig. 3. Generating Perlin noise.

For any point $P(x, y)$ on a two-dimensional plane grid, $\text{Noise}(x, y) = z$. as shown in Fig. 3, Corresponding to each control point $C(x, y)$, $G(x, y)$ is C's gradient value, whose direction and length are generated randomly.

$\text{Noise}(x, y)$ can be calculated according to the following steps:

- Calculating the contribution value of the surrounding four control points to the noise value of point P

$$\begin{aligned} D_1 &= \left(P(x,y) - C(i-1,j-1)\right)\cdot G(i-1,j-1), \\ D_2 &= \left(P(x,y) - C(i,j-1)\right)\cdot G(i,j-1), \\ D_3 &= \left(P(x,y) - C(i-1,j)\right)\cdot G(i-1,j), \\ D_4 &= \left(P(x,y) - C(i,j)\right)\cdot G(i,j), \end{aligned} \tag{8}$$

- Based on the improved interpolation formula $w = 6t^5 - 15t^4 + 10t^3$, interpolate D_1, D_2, D_3, D_4 in the X direction.

$$\begin{aligned} S_1 &= D_1 - \left(6u^5 - 15u^4 + 10u^3\right)\cdot\left(D_1 - D_2\right), \\ S_2 &= D_3 - \left(6u^5 - 15u^4 + 10u^3\right)\cdot\left(D_3 - D_4\right) \end{aligned} \tag{9}$$

where $u = x - (i - 1)$, $v = y - (j - 1)$, then interpolate S_1, S_2 in the Y direction.

$$\text{R} = S_1 - \left(6v^5 - 15v^4 + 10v^3\right)\cdot\left(S_1 - S_2\right) \tag{10}$$

- changing the amplitude and frequency of the noise function

$$\text{amplitude} = \text{persistence}^i, \text{frequency} = 2^i \tag{11}$$

where persistence defines that the value of amplitude changes with the frequency.

- The final Perlin noise function can be constructed by using superimposition of these changed noise functions:

$$\text{Noise}(x,y) = \textstyle\sum_{i=1}^{N} \text{Noise}_i(x,y) \tag{12}$$

If the noise value of Eq. (12) is taken as the height of the projective grid points, a smooth height field can be obtained. To obtain fluctuating height field, using coordinate transformation on the points of lake surface grid:

$$\begin{aligned} T_i &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ i\times\Delta x & i\times\Delta y & 1 \end{bmatrix}, \\ \text{T} &= \begin{bmatrix} \alpha & \beta & 0 \\ -\beta & \alpha & 0 \\ (1-\alpha)\times x_f + y_f\times\beta & (1-\alpha)\times y_f - x_f\times\beta & 1 \end{bmatrix} \end{aligned} \tag{13}$$

where $\alpha = \cos(\theta + i \times \Delta\theta)$, $\beta = \sin(\theta + i \times \Delta\theta)$, Δx and Δy are the increments on the X axis and the Y axis, $\Delta\theta$ is the angle increment. Then the modified noise function can be expressed as follows:

$$\text{PerlinNoise}(x,y) = \text{Noise}\left(P(x,y) \times T_i \times T_r\right) \tag{14}$$

where $P(x, y)$ is expressed in homogeneous coordinates. After correction, the lake surface projective grid can be used to express the dynamic fluctuating lake surface due to the superimposition of the height field obtained by Perlin noise [12].

2.2.3. Rendering

The simulating effect of the light on the water surface plays an important role in the realistic sense of the water surface. The effect of the light includes refraction, reflection, diffuse and so on. This paper will discuss two of them which are reflection and refraction, respectively.

The illumination model satisfies the Fresnel law [13].

$$\rho = \frac{1}{(1 + E \cdot N)} \tag{15}$$

where E is the reflection direction, N is the normal direction. Fresnel reflected textures are read from a pre-calculated texture, and then using cube mapping technology to obtain texture color to generate reflective texture objects, then mixed with the water surface color to get the light reflection image. The calculation of refraction is similar, which need rendering objects that do not have transparent refraction under the actual viewpoint, then save the final calculation result in a texture [14–16].

Through using various rendering methods, realistic lake simulation is realized. Fig. 4 shows the rendering process.

3. Simulation and result

NVIDIA CUDA provides a good comprehensive development environment for programmers based on graphics processing unit (GPU) acceleration applications. The operation efficiency of parallel computing architecture built with CUDA is at least 10 times higher than the simple CPU [17–20].

This paper pays much attention to improving the real-time simulation of a large area of lake through using the CUDA architecture. In order to make the simulation system and the CUDA program run orderly, the osgCUDA class for the management of the CUDA program is designed. The big rectangles generated by the projective grid are divided into small rectangular segments, transfer to GPU, and realize parallel processing by CUDA kernel. Generating height field according to Perlin noise can also be realized by CUDA kernel [21,22]. Fig. 5 shows the simulation process based on CUDA.

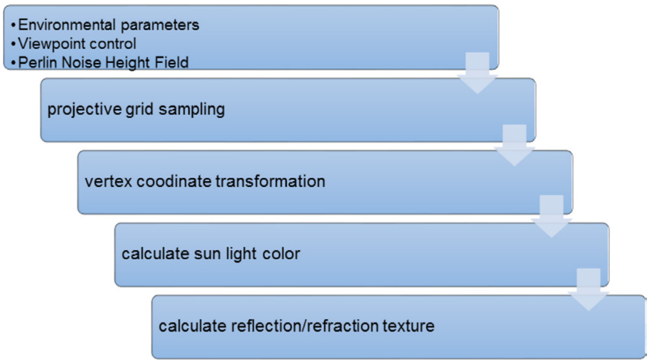


Fig. 4. Rendering process.

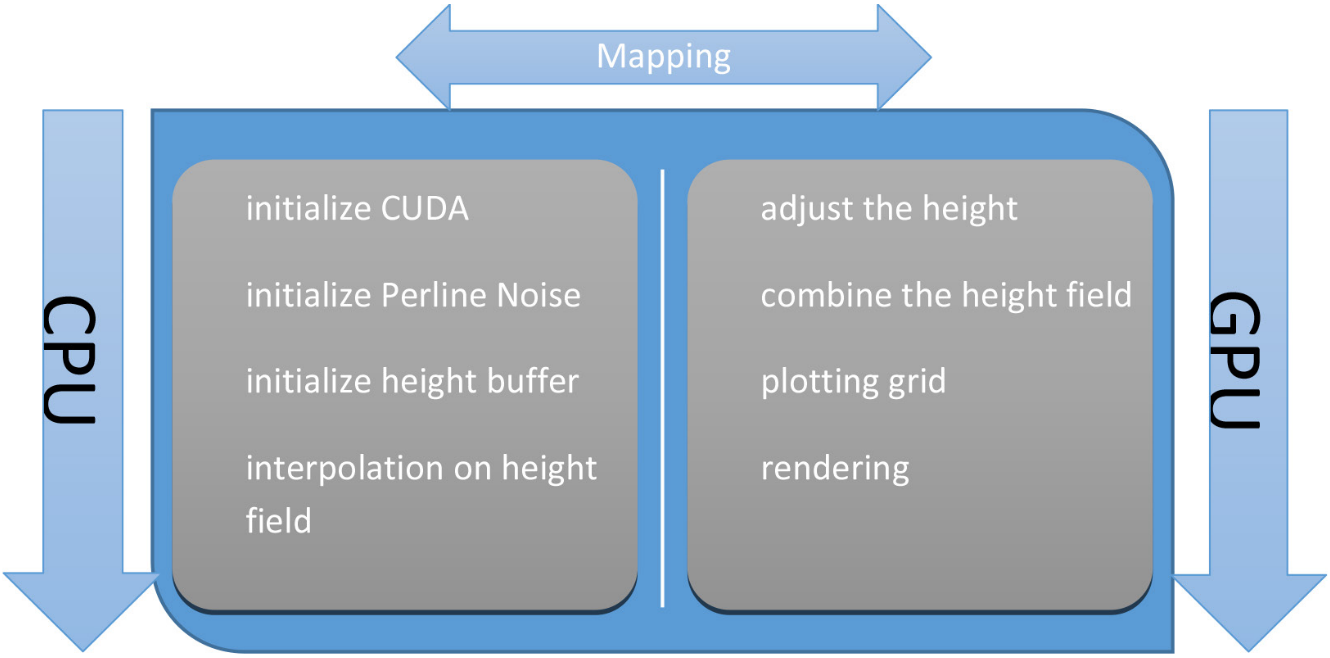


Fig. 5. Simulation process based on CUDA.

Fig. 6. Simulation result based on CUDA.

Table 1
Comparison of running results between CPU and CUDA platform

Grid resolution	CPU(ms)	CUDA(ms)	Speedup ratio
128*128	28.36	3.81	7.44
512*512	101.14	7.06	14.33
1,024*1,024	187.53	9.16	20.47
2,048*2,048	290.21	15.31	18.95

The environment of the above simulation test is: Windows 7, Visual Studio 2015, CUDA5.5, 16GB memory card, video card GTX560, CPU Intel Core i3 4core 3.3 GHz. The last generated scene is shown in Fig. 6. The running results are shown in Table 1.

Data in Table 1 show that using CUDA, the frame rate can reach 65 fps, when the resolution is 2,048*2,048. This can meet the requirement of real-time simulation.

4. Conclusion

By studying the large-scale fluid simulation method, the following conclusions can be drawn: (1) the old simulation results are not reasonable because when using spectrum statistics ocean models to simulate large-scale lake, the different characteristics between the ocean wave and the lake wave are not taken into consideration. (2) Basing on improved Perlin noise, by using the projection grid method, the large-scale lake simulation can produce reasonable lake surface wave effect according to the given wavelength, amplitude and frequency. (3) The CUDA architecture puts the calculation of height field generation, vertex transformation and rendering in GPU, which make full use of the high-speed computing power of video card and improve the performance of real-time simulation.

There is a limitation in the research. The reflection in the water varies with change of the angles between the reflection vector and the water surface. It is an issue deserving a strict attention in the future research. Furthermore, only one environment condition is taken into consideration in this study while more environmental factors could be considered to see whether there is different effect in simulation performance.

References

[1] Y.Z. Jiang, Y.T. Ye, H. Wang, Discussion on intelligent regulation technology architecture for river basin based on internet of things, Water Resour. Informat., 4 (2010) 1–5.

[2] X.F. Duan, H.X. Ren, Overview of physically-based ocean waves simulation, Comp. Sci., 41 (2014) 1–6.

[3] C.T. Pozzer, S.R.M. Pellegrino, Procedural solid-space techniques for modelling and animating waves, Comp. Graph., 26 (2002) 877–885.

[4] P.Y. Ts'o, B.A. Barsky, Modeling and rendering waves: wave-tracing using beta-splines and reflective and refractive texture mapping, ACM Trans. Graphics, 6 (1987) 191–214.

[5] L.B. Yan, Real-time generation of ocean wave surface, J. Comput.-Aided Des. Comput. Graphics, 12 (2000) 715–719.

[6] L. Pan, J. Zhou, J. Xiao, Three-dimensional digital lake simulation based on Openscenegraph and C++/CLI, ICIC Express Lett., 7 (2013) 3402–3408.

[7] P. Nugjgar, N. Chiba, Markov-Type vector field for endless surface animation of water stream, Visual Comput., 29 (2013) 959–968.

[8] J.D. Shi, Y.M. Jianf, Simulation of 3D shallow waves based on OpenGL, Microelectron. Comput., 23 (2006) 137–140.

[9] K. Perlin, in B.A. Barsky, Ed., An Image Synthesizer in Computer Graphics, SIGGRAPH '85 Proc., 1985, pp. 287–296.

[10] K. Perlin, E.M. Hoffert, in J. Lane, Ed. Hypertexture in Computer Graphics, SIGGRAPH '89 Proc., 1989, pp. 253–262.

[11] K. Perlin, Improving noise, ACM Trans. Graphics, 21 (2002) 681–682.

[12] Y. Xiang, S. Xu, Real-time rendering on dynamic water surface based on Perlin noise, Comp. Eng. Des., 34 (2013) 3966–3970.

[13] K.S. Zhang, E.J. Liang, Fresnel's law and phase relationship between reflected and incident light, J. Wuhan Univ. Technol., 27 (2005) 116–118.

[14] Y.F. Li, T.T. Cheng, W. He, Simulation method for lake surface wave, J. Syst. Simul., 21 (2009) 7507–7510.

[15] T.T. Cheng, L.L. Chen, W. He, Modeling algorithm for realistic travelling lake surface simulation in virtual environment, Comp. Eng. Appl., 44 (2008) 184–187.

[16] K.E. Niemeyer, C.J. Sung, Recent progress and challenges in exploiting graphics processors in computational fluid dynamics, J. Supercomput., 1 (2013) 1–37.

[17] X.J. Hu, H.Y. Yang, Y. Wan, Research and implementation of screen space based ocean simulation, Comp. Eng. Des., 36 (2015) 452–457.

[18] Z. Cao, T.Y. Su, G.Y. Wang, Multidimensional visualization technology research about the large-scale oceanographic environment, Period. Ocean Univ. China, 47 (2017) 132–138.

[19] L.L. Zhao, S.B. Zhang, M. Zhang, T. Yao, High performance FFT computation based on CUDA, Appl. Res. Comp., 28 (2011) 1556–1559.

[20] H.J. Wang, S.C. Zhou, C.Y. Ma, G. Chen, Paralleled dynamic multi-resolution physical simulation model of water surface, Comp. Eng., 38 (2012) 286–290.

[21] F. Du, Y. Zhanf, Visual simulation of atmospheric scattering on earth based on GPU, J. Syst. Simul., 21 (2009) 148.

[22] S.L. Wang, F.J. Kang, J.H. Xu, Research on general ocean simulation technology, J. Syst. Simul., 29 (2017) 381–386.