Ripple Simulation Based on Mass-Spring Model and Image Space Computation

Xinghua Zhang
Beijing Forestry University
Beijing, China

*Gang Yang**
Beijing Forestry University
Beijing, China

***Abstract*—Dynamic water surface simulation is one of the most challenging issues in computer graphics. This paper implement an effective method for simulating dynamic water ripple effect in real-time. In the method, the water surface is constructed as a mass-spring grid, and the motion status of each mass point can be computed with Hooke's law. Instead of computing the mass points' motion in three dimensional geometric space, we adopt a image-space computation approach: simulating the image deformation on each point caused by the water fluctuation to represent various dynamic water effects, including the refraction effect on water surface, wave propagation and wave attenuation. Experimental results show that the method can simulate water ripple effects realistically, and the real-time rendering performance can be achieved.**

Keywords-component; water ripple simulation;mass-spring model;image deformation

I. INTRODUCTION

In computer graphics, the simulation of natural scenes is one of the major research topics. As one of the most common natural materials, the realistic representation of water has great impact in the simulation of natural scenes. While, the simulation of water is always a challenging problem as water has dynamic and changeful surfaces and the lighting effects over water surfaces is complex.

In recent years, computer graphics researchers had proposed many methods in water simulation. Some methods compute the water animation based on physical models, such as Foster et al [1-2] solve the Navier-Stokes equation by finite difference approach to obtain the velocity field and pressure field of water to produce the height field of wave, achieving realistic simulation results; Kass et al [3] used simplified numerical method to solve the Navier-Stokes equation for water simulation; Premoze et al [4] simulated ocean based on the oceanologic theories and physical model, getting realistic simulation results of ocean surfaces.; Qian-hua Chen et al [5] adopted finite volume approach to solve 2-dimensional shallow water equations, and realizing the simulation of water ripple effect by using particle systems and implicit surfaces. These methods can get realistic simulation results, but it is difficult to achieve real-time simulation efficiency by these methods due to the complexity of solving physical models. In order to improve the simulation performance, some researchers adopt mathematical functions or process-based method to simulate the dynamic process of water approximately. Peachey [6] constructed different phases of sine function to simulate the wave shape; Fournier et al [7] synthesized the parametric surfaces to simulate the complicated waves; Murta et al [8] simulated the movement of water flow by using particle systems and implicit surfaces; Huai-ping Yang et al [9] put forward a wave simulation algorithm based on the cellular automata theory and neighborhood spread process.; Jinhui Yu et al studied the generation of cartoon water by using random sine wave. These process-based methods can obtain high performance, but due to the lack of physical model, their simulation results are often sort of simple, not as diverse as the real dynamic water. In this paper, we simulate the dynamic water by considering the stress relationships among the points of water surfaces, and introduce the Mass-spring model. The water surfaces are described as a mass-spring grid, and Hooke's law can be used for computing the motion of each point. Furthermore, in order to improve the computation efficiency, we adopt an image-based computation manner, which can simulate the water refraction effect, wave propagation and wave attenuation in the 2-dimensional image space, and achieving real-time simulation efficiency.

II. CONSTRUCTION OF WATER SURFACE BASED ON MASS-SPRING MODEL

When water is calm, the stress on each point of water surface is balanced, and all the points are in balance positions. If one point is acted upon by a force, it will produce acceleration, and deviate from the balance position, producing vibration in the position; this disturbance can affect the neighbouring points, making the vibration propagating around and forming the ripple effect. Base on this observation [10], we describe the stress situations over water surface by mass-spring model. We consider the water surface as a series of discrete particles connected by springs. All the particles and springs form a mass-spring grid. The stress and deformation of spring is followed the Hooke's law. So we can calculate the stress and motion of each particle in the grid according to the Hooke's law.

In order to avoid solving complex differential equations in simulation process, we make the following simplification: (1) The movement process of water surface is divided into discrete time steps (a time step is denoted as dt), and the stress acted on a particle in a time step is the same; (2) The stress of one particle is only relevant with its eight neighbouring particles (Fig.1).

*Corresponding author

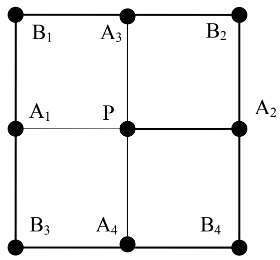

Figure 1. Particle P and its eight neighbouring particles.

Based on the simplification, the stress and movement of a particle can be calculated fast according to the state of its eight neighbouring paticles. Assuming in a moment, the particle P's displacement is S, speed is Vs. The stress of particle P can be calculated as:

Fs = k * b * [(B1+B2+B3+B4) - 4 * S] + k * a * [(A1+A2+A3+A4)-4*S].

Where k is the Hooke's coefficient; a and b are two influence weights used for the neighbouring particles with different distances and a>b. A1， A2， A3， A4 are closer than B1， B2， B3， B4 to particle P, so they have more influence on particle P. After obtaining the stress on particle P according to the above formula, we can calculate the acceleration of particle P as: a = Fs * dm (dm is the quality of the water, we set dm as 1g here). Then the speed of particle P in the next time step is: Vs2 = Vs - a * dt (dt is the interval of time); and the displacement of P in the next step is S2 = S + Vs2*dt.

According to the above calculation process, we can compute the stress and displacements of all the particles iteratively, and simulate the wavy effects of water under disturbing force.

III. WATER SIMULATION IN TWO-DIMENSIONAL IMAGE SPACE

Using above method, the three-dimensional water surface model can be constructed, and the dynamic water effects, such as wave propagation, wave attenuation and refraction effect can be computed by the three-dimensional model. For achieving higher performance, we don't implement the computation in three-dimensional space, but adopt a image-based computation method: processing the deformation in two-dimensional image space to represent the dynamic ripple effects over water surface.

We set a picture as the background of water, just like using a pattern to decorate the bottom of a pool. Each pixel of the picture corresponds to one particle of the mass-spring grid. When the water surface is fluctuated, because of refraction, the background picture will be deformed with the dynamic wave of water, presenting the realistic ripple impression of water. According to this idea, we simulate the refraction effect, wave propagation and wave attenuation in the two-dimensional image space, achieving good simulation effects and high performance. In the following sectons, we will introduce the simulation of refraction, wave propagation and wave attenuation respectively.

A. The simulation of Refraction

Fig.2 gives the principle of light refraction in water. At time step t0, the light of point P1 shoots from water into air, and since the refraction angle α is greater than β, the refractive light in air will deviate from normal. So when the refractive light shoots into eyes, people will feel the light is coming from P0. At the next time step t1(t1=t0+ dt, dt >0), because of the wave propagation, point P2 may moved to the position of point P1; then people will feel that the image color in P0 should be the image color of P2. So, in simulation process, because of light refraction, the image color of P1 and P2 should be displayed in point P0 in time step t1 and t0 respectively.

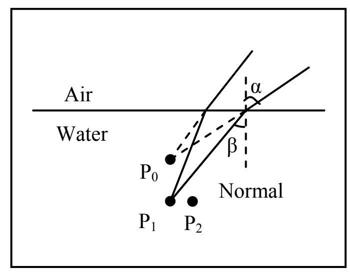

Figure 2. Light refraction.

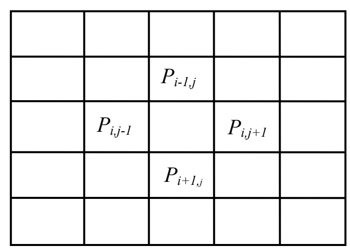

Figure 3. The relationship of Pi,j and points around.

B. The simulation of wave propagation

When water is calm, the image in water is also calm. But when water fluctuates and the water wave propagates around, the image will deform continuously and regularly, giving people the wonderful impression of wave effects. At the same time, different points on water surface usually have different height because of the water fluctuation, hence causing the refrection effects on different points are different, forming image deformation. To simulate the deformation effects, the key is to compute the amplitude of each point at each time. The most direct method is of computing the amplitude and wave at each time step with Hooks' Law and calculating the refraction effect with refraction equation. But the computing complexity of the method is very high, especially there are many computation of sin (x) and cos (x) in the solving of refraction equation. In this paper, a very fast and linear computation method for wave generation is given. This method can compute the amplitude of a point at the next moment just by certain linear formula according to the amplitudes of its four neighbouring points (the adjacent up, down, left, right points) at current moment.

In Fig 3, we set the amplitude of Pi,j at t0 is Xi,j, the amplitude at t1 is X'i,j, assume the X'i,j is relevant with not only Xi,j but also the adjacent up, down, left, right points of Pi,j. In order to simplify calculation, the influence weights of the four points are the same, and the influences of other points are omitted. Therefore, a wave amplitude linear calculation formula can be written:

$$X'_{i,j}=\alpha\cdot(X_{i,j-1}+X_{i,j+1}+X_{i-1,j}\ X_{i+1,j})+b\cdot X_{i,j} \qquad (1)$$

In formula (1), the amplitude of Pi,j at t0 is X'i,j; a，b are the influence weights; the amplitude values of Pi,j and its four neighbouring points at moment t0 are Xi,j ，Xi,j-1，Xi,j+1，Xi-1,j and Xi+1,j.

Assuming the resistance of water is 0. According to the energy conservation law, the total potential energy of water should remain unchanged, that means the sum of amplitude of all the points at any moment is a constant. Therefore, formula (2) can be established for t0 and t1:

$$\sum_{i=1}^{m}\sum_{j=1}^{n}X'i,j=\sum_{i=1}^{m}\sum_{j=1}^{n}Xi,j \quad (m，n>0) \qquad (2)$$

Where m is the row number, and n is the column number; there are m×n pixels in the picture for computation. Computing X'i,j according to formula (1) , and substituting it into formula (2); the result can be gotten:

$$a\bullet\left(\sum_{i=1}^{m}\sum_{j=1}^{n}X_{i,j-1}+\sum_{i=1}^{m}\sum_{j=1}^{n}X_{i,j+1}+\sum_{i=1}^{m}\sum_{j=1}^{n}X_{i-1,j}+\sum_{i=1}^{m}\sum_{j=1}^{n}X_{i+1,j}\right)+$$
$$b\bullet\left(\sum_{i=1}^{m}\sum_{j=1}^{n}Xi,j=\sum_{i=1}^{m}\sum_{j=1}^{n}Xi,j\right) \qquad (3)$$

The corresponding points of Xi,0 and Xi,n+1 (i=1，2，…，m) are beyond the range of picture, so we use the amplitudes of boundary points as their approximate values; that means we set:

$X_{i,0}\approx X_{i,1}$ and $X_{i,n+1}\approx X_{i,n}$ *(i=1，2，...，m)*，similarly：

$X_{0,j}\approx X_{1,j}$ and $X_{m+1,j}\approx X_{m,j}$ *(j=1，2，...，n)*

According to above assumption, formula (3) can be rearranged as:

$$(4a+b)\bullet\sum_{i=1}^{m}\sum_{j=1}^{n}Xi,j=\sum_{i=1}^{m}\sum_{j=1}^{n}Xi,j \qquad (4)$$

Generally, the value of $\sum_{i=1}^{m}\sum_{j=1}^{n}Xi,j$ is not 0, so we can get:

$$(4a+b)=1 \qquad (5)$$

From formula (5), we can get one very simple solution: *a = 1/2, b = –1*. The calculation of "1/2" can be realized by shift operation, which is very fast, so the solution *a = 1/2, b = –1* is the best solution for simplifying the calculation.

Substituting the solution into formula (1), the linear wave amplitude formula can be obtained eventually:

$$X'_{i,j}=\frac{X_{i,j-1}+X_{i,j+1}+X_{i-1,j}+X_{i+1,j}}{2}-X_{i,j} \quad (6)$$

From formula (6) we can get the approximate method for calculating the amplitude of each point: after knowing the amplitude of a certain point at current moment, the amplitude of the point at the next moment can be calculated by dividing the sum of the four amplitudes of its neighbouring points by 2, and minus its amplitude at current moment.

C. The simulation of Wave attenuation

In formula (6), the energy attenuation is not considered, making the fluctuation of water preserve forever. In fact, the water wave should be attenuated gradually because of the energy attenuation. In order to simulate the wave attenuation effect, we decrease the amplitude according to a certain rate each computation step. The attenuation rate is set as 1/16 or 1/32 in this paper, and hence the attenuation computation can be implemented with shift operations in very high performance.

IV. THE IMPLEMENTATION OF SIMULATION ALGORITHM

We define two two-dimensional arrays corresponding to two pictures, bufl[Height][Width] and buf2[Height][Width], to save the amplitude of each point at two continue moments. "Height" is the row number of the picture, and "Width" is the column number. At first, the initial state of water is flat, and the amplitude of each point is set as 0 (that means the bufl and buf2 are initialized to 0). In simulation process, the data of the two buffers are exchanged and displayed continuesly, realizing the simulation of dynamic water.

The computation process of wave simulation can be represented as following code:

```
void RippleSpread( )
{
    for(int i = 0; i < Height; i++)
    for(int j = 0; j < Width; j++)
    {   // Corrugated energy diffusion
        buf2[i][j]=((buf2[i-1,j]+buf2[i,j-1]+buf2[i,j+1]
        +buf2[i+1,j]) >>1)+buf2[i][j];
        // Energy attenuation. Attenuation rate is 1/32
        buf2[i][j]-=buf2[i][j]>>5;
    }
    // Exchange the data buffer
    int **ptmp = buf1;
    buf1 = buf2;
    buf2 = ptmp;
```

}

In the computation process, we only use the simple addition, subtraction and displacement operations in two-dimentional image space. Compared with calling large number of trigonometric functions or simulating in three-dimensional geometric meshes, this method has obvious advantage in speed.

In order to produce ripples on the calm water surfaces, the trigger energy should be added into water. This is just like throwing a stone into water to produce ripples in the real life. The size and energy of ripples are related to the weight and volume of the stone. In the algorithm, we give the trigger energy by modifying the buffer data. Executing an assignment statements buf[x][y] = -n is equivalent to set a "sharp pulse" in point (x,y). According to our experiments, it is appropriate to set the value of n in 32-128. The trigger energy can have certain radius, that's giving "sharp pulse" to all the points in the circular area that center at the entry point and have certain radius. The value and radius of trigger energy are just like the weight and volume of the stone thrown into water.

V. EXPERIMENTAL RESULTS

We implement the water simulation with Java3D programming. Users can trigger water ripples by clicking mouse or sliding mouse in water surfaces (as shown in Fig 4). Users also can interactively adjust the 3D position and orientation of water scene (Fig 5). In addition, randomly generated raindrops can be added on water surfaces to simulate the water ripple effects in the rain (Fig 6).

The resolution of image space in Fig.4, 5 and 6 is 512*512, which means there are 512*512 particles in the mass-spring grid for water simulation. The simulation performance can reach 36 frames per second, satisfying the requirement of real-time application.

Figure 4. Mouse slides over the water surfaces.

Figure 5. Changing the viewing orientation.

Figure 6. Rippled water in the rain.

VI. CONCLUSION

In this paper, the mass-spring model is used to construct the water surface, and an image based approach is adopted to simulate the wavy effects of water. With the method, the refraction effects, wave propagation and wave attenuation can be simulated realistically, and the simulation performance is real-time. The method of this paper can be applied effectively for dynamic water simulation in virtual scene representation.

Because the simulation process of our method is executed in two-dimensional image space, the larger wavy effects, such as spray and spoondrift still can't be simulated. In the future, we can consider simulating these more complex wavy effects by combining the computation in the three-dimensional mass-spring model.

ACKNOWLEDGMENT

This work was supported by National Higher-education Institution General Research and Development Foundation in

Beijing Forestry University (No.YX2010-24) and National "948" Program of China (No.2009-4-41)

REFERENCES

[1] Foster , Metaxas D, "Realistic Animation of Liquids. Graphical Models and Image Processing" , 1996, 58 (5), pp. 471-483.

[2] Foster, RFedkiw, "Practical Animation of Liquids." Proc of the 28th Annual Conf on Computer Graphics and Interactive Techniques. New York : ACM Press, 2001.

[3] Kass, M.Miller, G Rapid, "stable fluid dynamics for computer graphics" 1990(04).

[4] Premoze, S. M Ashikhmin, "Rendering Natural Waters". Computer Graphics Forum, 2001, 20 (4), pp. 189-200.

[5] Qian-hua Chen, Jian-song Deng, Fa-lai Chen. "The computer animation of water ripples". Journal of Computer Research and Development. 2001,05, pp.524-528.

[6] D Peachey, "Modeling waves and surf." Computer Graphics.1986,20(4), pp. 65-74.

[7] A Fournier, W T Reeves, "A simple model of ocean waves." Computer Graphics, 1986,20(4),pp. 75-84.

[8] Alan Murta, James Miller, "Modeling and rendering liquids in motion." WSCG1999,Plzen: University of West BohemiaPress,1999.194 201.

[9] Huai-ping Yang, Shi-min Hu, Jia-guang Sun, "A New Algorithm for Water Wave Animation". Chinese Journal of Computers. 2002,25(6).

[10] Li-liang Pan, "Ripple Simulation Based on Hooker law" http://dev.gameres.com/Program/Visual/Effects/DMyRippleII.doc. 2002-07-21/2003-08-25.

[11] Guang-xin Li, Zhen-guo Ting, Hai-sheng Zhan, Li-hua Zhou, "A Modeling Algorithm for Realistic Water Surface Simulation in Virtual Environment". Journal of Computer Research and Development. 2004 41(9).

[12] Yin Tian, "Designing for a Animation-procedure Based on Java". Computer Knowledge and Technology(Academic Exchange). 2006,02, pp.165-166.

[13] Xiao-dong Wang, Lei Guo, Hong Luo,Yin-wen Dong, "Design and Implementation of Network Topology Display by Waves Animation Based on Java.". Application Research of Computers.2005,22(3).

[14] G. Hopkins, "Java3D Tutorial".
http://java.sun.com/developer/onlineTraining/java3d/ .2007.

[15] S. Fujii , H. Aoyama, M. Sota, J. Iwata, and T. Mizuno, "A system for Learning Ecology with 3D graphics and Local Materials Database". Current Developments in Technology-Assisted Education. 2006.

[16] Matthew W. Becker, "An Introduction to Ground-Water Modeling Using Virtual Reality Modeling Language (VRML)". Journal of Geoscience Education, 2003-11.

[17] D. Selman, "Java.3d Programming". USA: Oreilly & Associates Inc 2005-07.

[18] Sun Microsystems, Inc, "The Java 3D™ API Specification Version 1.2". USA: Sun Microsystems, Inc. 2000-03.