An Efficient and Stable Water Surface Animation for Interactive Applications

Sang-Min Song, Young-Min Kang, and Soo-Yol Ok†

TONGMYONG University
535, Yongdang-Dong, Nam-Gu, Busan, Korea, 608-711
{songsmir,ymkang,sooyol}@tu.ac.kr

Abstract. In this paper, we propose a realtime approach to water surface animation based on shape matching. The method proposed in this paper is based on shape matching method. The water surface is modeled as a mesh structure which is divided into multiple clusters composed of mesh vertices. The clusters are overlapped with adjacent clusters, and each cluster is animated with shape restoration forces. The shape restoration property of each cluster produces wave effect on the water surface. We exploited GPU parallelism to animate the water surface in realtime, and the water surface can plausibly interact with external objects.

Key words: shape matching, wave equation, wave, cluster based shape matching, GPU parallelism

1 Introduction

[1] As the recent graphics hardware is being rapidly developed, more and more realtime applications are trying to employ realistic water representation. The previous methods can be classified into two major categories: approximation with parametric wave functions and physics animation with wave equation. Although the approximation approach can efficiently generate the water surface, the interaction between the water surface and other objects cannot be easily implemented. In contrast, the simulation based on the wave equation requires heavy computation but the interaction between water and external objects can be easily represented. In order to employ the water animation in an interactive application such as game, the interaction between the water and other object should be taken into account. In this paper, we propose an interactive water surface animation method based on shape matching. The method produces realistic water animation with plausible interaction with external objects. Moreover, the method is stable enough to be employed in interactive applications.

2 Previous work

In the early techniques, the water surface was represented as a height map computed with continuous periodic functions[1]. The method was improved to repre-

[1] †: Corresponding Author

sent more realistic water surface by considering the water depth[2]. Such methods based on approximation can efficiently produce the water surface. However, a large amount of periodic functions is required to represent a complex and realistic water surface. Chen[3] exploited GPU parallelism to efficiently process the periodic functions. The method produces a bump map to represent the water surface. LOD approach was also exploited to alleviate the computational cost[4–6].

The motion of water surface can be easily modeled and animated with the periodic functions. However, the periodic function cannot easily produce complex surface. In order to represent the complex surface, fast Fourier transform (FFT) was also exploited[7, 8]. However, the method with FFT cannot effectively control the water surface. Therefore, the methods cannot be employed interactive application.

In order to enable the interaction with external objects, physically based approach should be employed. The major categories for physics based water simulation are grid-based Euler method [9, 10] and particle-based Lagrange method[11]. However, the physics based approaches require too heavy computation to be integrated into realtime applications.

Recently, an efficiently method based on 'wave particles' were proposed[12]. However, the method cannot produce the wave valleys so that the surface is not sufficiently realistic.

3 Basic Concept of Water Surface Animation

Our method links the deformable object to generate wave effect. The physical property of each object is dominated by restoration force computed by shape matching. Therefore, each cluster element of water surface is constrained to maintain original shape. Since each element is overlapped with adjacent element, the motion of each element is transported to the adjacent elements. Therefore we can obtain the wave effect along the adjacent object chain.

The shape matching approach produces force to move deformed vertices back to the target position. The deformable objects try to maintain the original shape, but the locations of vertices in the objects are not strictly constrained. The deformable object is deformed when external force is exerted as shown in Fig.**??**. In this paper, V denotes the set of vertices. A vertex i is an element of the set V. The properties of i include the current position x_i, the original position x_i^0, goal position g_i, acceleration a_i, velocity v_i, and the mass m_i. If the object is rigid, the position of vertex i can be expressed with a rotation matrix $\mathbf{R}$ and a translation matrix $\mathbf{T}$. Once we know the matrix $\mathbf{R}$, the goal position can be easily computed as follows:

$$x_i^{'} = m_i R(x_i^0 - t_0) + t - x_i \tag{1}$$

where t is the current center of mass and t_0 denotes the original center of mass.

In order to compute the rotation matrix **R** in Eq.1, we employed a optimal linear transformation matrix as follows:

$$x_i^{'} = m_i A(x_i^0 - t_0) + t - x_i \tag{2}$$

Least square method is applied to compute the matrix **A** in Eq. 2. The change of vertices can be described as follows:

$$\varepsilon_i = Aq_i - p_i \tag{3}$$

We should minimize the following:

$$S = \sum_i \varepsilon_i^2 = \sum_i (Aq_i + p_i)^2 \tag{4}$$

Then the matrix **A** is obtained as follows:

$$\begin{aligned} A &= (\textstyle\sum_i m_i p_i q_i^{\mathbf{T}})(\sum_i m_i q_i q_i^{\mathbf{T}})^{-1} \\ &= ApqAqq \end{aligned} \tag{5}$$

The matrix **A** in Eq. 5 can be decomposed into $\mathbf{A}pq$ and $\mathbf{A}qq$. Since the symmetric matrix $\mathbf{A}qq$ involves no rotation, the matrix $\mathbf{A}pq$ is actually utilized. The rotation matrix **R** can be obtained as follows:

$$\begin{aligned} A_{pq} &= RS \\ R &= A_{pq}S^{-1} \end{aligned} \tag{6}$$

Each vertex in the object are forced to move to a goal position g_i, and the goal position can be obtained with the rotation matrix **R** as follows:

$$g_i = \mathbf{R}(x_i^0 - t_0) + t \tag{7}$$

4 Water Surface Model and Animation

The shape matching approach utilizes the rotation matrix **R**. However, the matrix restores the original shape of each object so that the wave vanishes too rapidly. Therefore, we employed the matrix $\mathbf{A}pq$ instead of **R** to produce more realistic water surface.

Our methods represent the water surface with the collection of small geometric elements called clusters which are essentially deformable objects. The objects share some vertices with overlapping adjacent clusters as shown in Fig. 1. The shape matching process is applied to each cluster. Total n clusters are denoted as c_1, c_2, ..., and c_n respectively, and C denotes the set of the clusters.

As shown in Fig.1, the cluster c_i is composed of 4 vertices, and Each cluster shares some vertices with adjacent clusters as shown in Fig. 2.

In this paper we employed distortion transportation between shape matching based deformable objects. The difference between the goal position of deformed cluster and undeformed cluster is transferred along the chained clusters, and the effect of the transfer produces wave motion.

When an external force is applied to x_{13} in Fig. 3, the clusters which share the vertex also deformed as shown in Fig. 3. The clusters are animated to restore the original shape by shape matching algorithm. The shape matching applied to each cluster produces different goal position of x_{13} such as g_{13}^6, g_{13}^7, g_{13}^{10}, and g_{13}^{11} as follows:

$$g_{13} = (g_{13}^6 + g_{13}^7 + g_{13}^{10} + g_{13}^{11})/N_{cluster} \tag{8}$$

where $N_{clusture}$ denotes the number of clusters that share the vertex.

Water surface boundary For the realistic animation of water surface, the wave should be reflected at the boundary of the water surface. However, the previous methods cannot easily produce the wave reflection. In our method, the reflection can be easily produced by simply constrain the boundary vertices. Fig. 4 shows the wave reflection result when the boundary vertices are constrained not to move. As shown in the Fig. 4 (a), the external force applied to a vertex produces wave shown in Fig. 4 (b). The wave is reflected at the constrained vertices as shown in Fig. 4 (c) because the goal positions of the vertices are constant. Our method, therefore, easily produce wave reflection and the boundary of the water surface can be easily specified.

Damping with original position The method proposed in this paper produces wave effect by keeping each cluster maintaining the original shape. However, the simple shape matching of each cluster produces long-lasting oscillation. The actual water surfaces easily become calm when no more external forces are exerted. In order to represent such property of water surface, we employed damping with original shape. Each vertex are animated not only with the shape matching force, but also with the restoration force back to the original position. As shown in Fig. 5, the vertices are accelerated to move back to the original

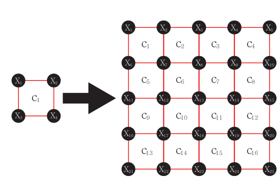


Fig. 1. Water surface structure with clusters

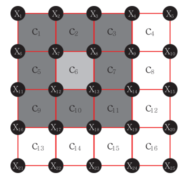
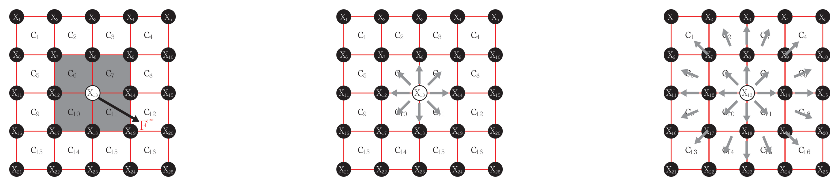


Fig. 2. Overlapping clusters: the cluster c_6 shares the vertices x_7, x_8, x_{12}, x_{13} with adjacent clusters $c_1, c_2, c_3, c_5, c_7, c_9, c_{10}$, and c_{11}.

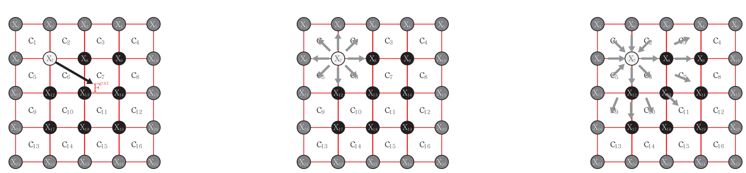


(a) shared vertex x_{13} (b) shared vertex propagates the wave (c) propagated waves

Fig. 3. Wave energy transportaion

position x_i^0, and the result animation shows plausible damping on the water surface. Such behavior can be regarded a damping process that dissipates the wave energy to restore the global water surface back to the energy equilibrium. The restoration force is of course proportional to the deformation amount, and the resulting behavior is naturally exponential function of the opposite direction of the deformation. In our case, the deviation tendency of each particle was measured by the velocity of the particle, and the actual acceleration for each particle i is computed as $a_i = (x_i^0 - x_i)\exp(-v_i)$.

Wave interference In order to produce realistic wave surface, interference of different waves is very important. In the previous methods, the wave interference was often ignored or required additional computation. However, our method produces plausible wave interference without any additional consideration or computation. figure to the left in Fig. 6 shows single wave while the figure to the right shows the interference of two different waves. As shown in the figure,



(a) external force (b) wave generation (c) wave reflection

Fig. 4. Wave reflection

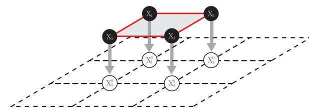


Fig. 5. Damping with original shape: the deformed vertices x_1, x_2, x_3, x_4 incur restoration force back to the original positions x_1^0, x_2^0, x_3^0, and x_4^0.

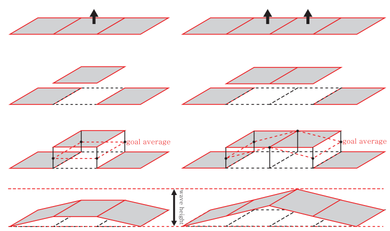


Fig. 6. Wave interference: the differences of goal positions produces plausible wave interference

the amplitude of the wave is computed by the average goal position of adjacent clusters. This property easily enables the wavy interference.

Improved cluster shape The simple rectangular clusters cannot represent various shape deformation so that the complex water surface cannot be produced. Therefore, we improved the shape of the clusters as shown in Fig. 7 to generate various deformations within the original rectangular cluster.

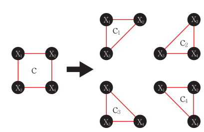
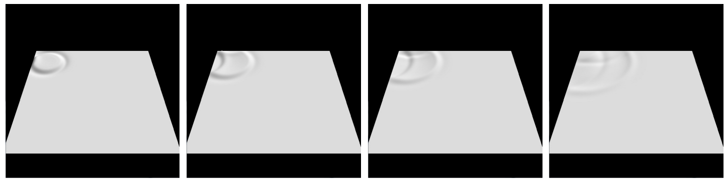


Fig. 7. The rectangular cluster C is subdivied into 4 clusters c_1, c_2, c_3, and c_4.

5 Experiments

The proposed method was implemented on a system with intel Core i7-3960X CPU system running on Windows 7 64bit OS with 32GB RAM and three NVidia Geforce GTX 580 graphics hardwares.

Fig. 8 shows the wave reflection at the surface boundary. The waves are plausible reflected without any additional consideration or computation. Fig. 9 also shows the wave reflection. The black-colored external object is located on the surface, and the wave is reflected at the boundary when it hits the object.



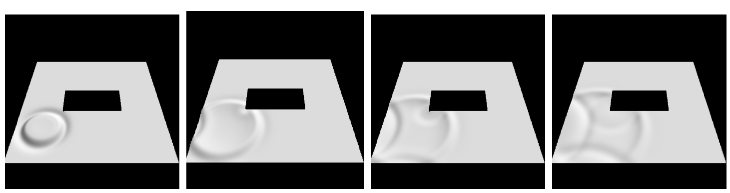
$t = 1$ $t = 2$ $t = 3$ $t = 4$

Fig. 8. Wave colliding with the water surface boundary

Fig. 10 shows the interference effect of different waves. The proposed method plausibly and efficiently expresses the wave interference. The figures to the left show the rendering results of interfering waves, and the figure to the right shows the amplitudes of wave surface in order to highlight the interfering effect. Fig. 11 shows the water animation and rendering result with our method.

6 Conclusion

In this paper, we proposed interactive animation techniques for water surface. The proposed method requires no special computation for water-object interaction. Moreover, the method produces stable and physically plausible animation results, and the waves are efficiently interferes each other. The water surface element objects are divided into clusters and efficiently processed with parallel GPU. The efficiency and plausibility of the method proposed in this paper make it possible to employ realistic water representation in realtime applications.

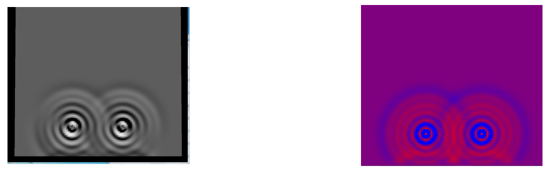


$t = 1$ $t = 2$ $t = 3$ $t = 4$

Fig. 9. Wave reflection at the external object boundary

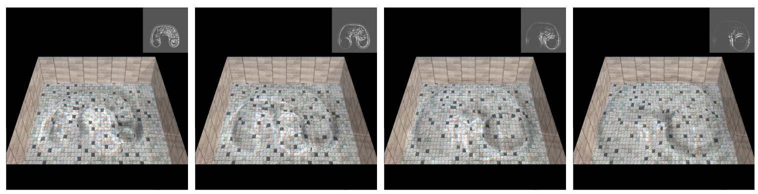
Acknowledgments

This work was supported in part by the Ministry of Knowledge Economy (MKE), Korea, under the Information Technology Research Center (ITRC) support program supervised by the National IT Industry Promotion Agency (NIPA) (NIPA-2011-(C-1090-1021-0006)), and this work (Grants No.C0033371) was supported



(a) Rendering of interfering waves (b) wave amplitude map

Fig. 10. Wave interference result



$t = 1$ $t = 2$ $t = 3$ $t = 4$

Fig. 11. Water animation result

by Business for Cooperative R&D between Industry, Academy, and Research Institute funded Korea Small and Medium Business Administration in 2012.

References

1. Max N. : Vectorized procedural models for natural terrain: Waves and islands in the sunset. ACM SIGGRAPH. 517–324 (1981)
2. Peachey D. : Modeling waves and surf. SIGGRAPH Comput, Graph. 65–74. (1986)
3. Chen H., Li Q., Wang G. : An efficient method for real-time ocean simulation. Technologies for E-Learning and Disital Entertainment 4469. 3–11. (2007)
4. Mark Finch. : Effective Water Simulation from Physical Models. Addison-Wesley. 5–29. (2004)
5. Lee H.-M., Go C., Lee W.-H. : An efficient algorithm for rendering large bodies of water. Entertainment Computing - ICEC. 302–305. (2006)
6. Lee H.-M., Go C., Lee W.-H. : A frustum-based ocean rendering algorithm. Agent Computing and Multi-Agent Systems. 584–589. (2006)
7. Masting G., Watterberg P., Mareda J. : Fourier synthesis of ocean scenes. IEEE Comput. Graph. Appl. 16–23. (1987)
8. Premoze S., Ashikhmin M. : Rendering natural waters. IEEE Computer Society. p.23. 4, 7, 10, 13, 14. (2000)
9. Kass M., Miller G. : Rapid, stable fluid dynamics for computer graphics. ACM SIGGRAPH. 49–57 (1990)
10. Foster N., Metaxas D. : Realistic Animation of Liquid. Graphical Models and Image Processing. 23–30 (1995)
11. Miller G., Pearce A. : Globular Dynamics: A Connected Particle System For Animating Viscous Fluids. (1989)
12. Yuksel C. : Wave Particles. ACM SIGGRAPH 2007 (2007)